BROADCOM SOFTWARE

DX Operational Intelligence - SaaS

Table of Contents

DX SaaS Network Requirements **10**

Upcoming Features/Enhancements **11**

Release Notes **13**

- **Content Reference** **13**
- **DX Operational Intelligence** **14**
  - 2023.9.1 Release Notes 19
  - 2023.8.2 Release Notes 21
  - 2023.8.1 Release Notes 30
  - 2023.7.1 Release Notes 31
  - 2023.6.1 Release Notes 31
  - 2023.5.1 Release Notes 35
  - 2023.4.2 Release Notes 40
  - 2023.4.1 Release Notes 51
  - 2023.3.2 Release Notes 54
  - 2023.3.1 Release Notes 57
  - RESTMon 2.2.1 Release Notes 62
  - 2023.1.1 Release Notes 63
  - 2022.12.1 Release Notes 71
  - 22.08 Release Notes 81
  - 22.06 Release Notes 85
  - 22.05 Release Notes 85
  - 22.04 Release Notes 88
  - 22.03 Release Notes 91
  - 22.02 Release Notes 95
  - 22.01 Release Notes 95
  - Release Notes 2021 102
    - 21.12 102
    - 21.11 104
    - 21.10 105
    - 21.09 106
    - 21.08 108
    - 21.07 111
    - 21.06 112
    - 21.05 114
    - 21.04 116
    - 21.03 117

21.02 120
21.01 121
Compatibility Matrix 122
Known Issues 122
Third Party Software Requirements 124
Overview of DX Operational Intelligence 125
Quick Start Guide for Tenant Administrators 128
Getting Started 129
Architecture Overview 129
Ports and Integration 130
Tenant Onboarding Process 132
DX Operational Intelligence User Interface 134
Launch Pad 134
Common Services - Settings Page 135
DX Operational Intelligence Home Page 137
Settings Page 138
Learning DX Operational Intelligence 139
Academy Courses 139
Video Tutorials 140
Blogs 141
Communities 141
Knowledge Base Articles 141
Tenant Profile Management 144
Features Management 145
User and Role Management 146
Role Management 146
Supported User Roles and their Access Privileges 146
Tenant Administrator 147
Security Administrator 153
Power User 155
User 161
Best Practices for RBAC Implementation 166
Create and Manage Custom Roles 167
Create Custom Role 167
Edit Custom Role 168
Copy Custom Role 169
Deactivate Custom Role 169
Delete Custom Role 170
Capabilities Access Privileges 170

DX Operational Intelligence....170
DX Platform....177
DX Dashboards....177
OI Universes....178
Create and Manage an OI Universe....179
Data Level Access Permission for Capabilities....182
Service Analytics Data Level Access Permission....182
Alarm Analytics Data Level Access Permission....182
Monitored Inventory Data Level Access Permission....183
Performance Analytics Data Level Access Permission....183
Predictive Insights Data Level Access Permission....184
Capacity Analytics Data Level Access Permission....184
User Management....184
Create and Manage Users....185
Manage SAML Users....188
Manage SAML-Defined Groups Mapped to DX Platform Role....190
Encrypt SAML Assertion....193
Update the IdP Certificate....193
Configure SiteMinder as SAML Identity Provider....193
Capability Configurations....196
Custom Situation Definitions....196
How Situation Clustering Works....196
Configure Custom Situation Definition....197
Create Custom Situation Definition....197
Define Custom Situation Name....200
Define Alarm Filter Criteria....201
Define Situation Stabilization Criteria....202
Define Alarm Clustering Criteria....203
Preview Results of Custom Situation Definition....206
Prioritize Custom Situation Definitions....207
View Situation Clusters of Type Custom....207
Integration....208
Token Management....210
Generate a Token as Tenant Administrator....210
Generate a Token as Security Administrator....211
Generate a Token as Power User, User, or User With Custom Role....211
Inbound Integration....212
Connector Parameters....212
Monitoring....213

Connector Health Monitoring....213
RESTMon Self-Monitoring....214
Integration with CA Products....214
Integrate DX Application Performance Management....214
Integrate DX App Synthetic Monitor....215
Integrate DX NetOps Performance Management....218
Integrate DX NetOps Spectrum....218
Integrate DX Unified Infrastructure Management....222
Integration with Third-Party Products....224
Ingestion APIs....224
RESTMon....245
Outbound Integration....571
DX Gateway....572
On-Prem Gateway....575
On-Prem ITSM....579
Channels....579
Prerequisites....580
Configure Email Channel....580
Configure Webhook Channels....582
Configure Ticket Management Channels....606
Troubleshoot Notifications....649
Policies Overview....650
Supported Alarm Types....650
Policy Filters....650
Create Policy....652
Create a Policy from Alert Queue....656
Edit Policy....657
Suppress Notifications During Maintenance Schedule....657
Supported Filter Attributes Reference....658
Message Templates....665
Default Message Templates....665
Custom Message Templates....679
Create a Copy of Existing Template....682
Message Template Variables....682
Proxy Configuration....698
Using....700
Service Analytics....700
Services User Interface....701
Service Overview Page....701
Service Creation Templates....721

Service Details Page.... 728
Service Level Indicator and Service Level Objective.... 755
Enable the SLIs & SLOs Tile.... 757
Create SLIs and SLOs.... 757
Configure Service Availability Using SLI and SLO.... 763
Manage Adjustments.... 768
Error Messages.... 772
Service Personalization.... 772
Personalizing Service View.... 773
Personalizing Service Details View.... 775
Service Personalization using Custom Metrics.... 777
Alarm Analytics.... 782
Alarm Severity Mapping.... 783
All Alarms.... 784
Alarms Inspector.... 812
Metric-Based Alert Configuration.... 820
Situation Alarms.... 825
Service Alarms.... 840
Anomaly Alarms.... 856
Maintenance Window.... 857
Audit Trail.... 863
Configure Automic Automation.... 864
Using Automic Automation.... 869
Insights.... 873
Performance Analytics.... 877
Troubleshoot Anomaly Detection.... 884
Manage Views.... 884
Anomaly Detection.... 885
Configure Monitoring.... 886
Metric Triage Use Cases.... 900
Capacity Analytics.... 901
Who can use Capacity Analytics.... 901
Capacity Analytics Process Flow.... 901
Access Capacity Analytics.... 903
Metrics for Capacity Analytics.... 903
vmware Probe Metrics.... 920
cdm Probe Metrics.... 928
ibmvm Probe.... 932
hpe_3par Probe.... 934
netap ontap Probe Metrics.... 935

Universal Monitoring Agent for Kubernetes....937
Services Configuration....937
Configure Services in Capacity Analytics....937
Groups Configuration....939
Configure Groups for Capacity Analytics....940
Metrics Configuration....942
Navigating Capacity Analytics....947
Capacity Analytics Overview Page....947
Health Chart....948
Top Capacity Consumers....953
Configured Groups and Services....958
Service Key Performance Indicators (KPIs)....971
Troubleshoot Capacity Analytics....972
Predictive Insights....972
Predictive Insights User Interface....973
Topology Tab....976
Alarm Actions for Prediction Alarms....977
Enable Predictive Definitions....980
Predictive Insights OOB Metrics....981
Monitored Inventory....986
Monitored Inventory User Interface....987
Monitored Inventory Workflow....1001
DX OI - Logs....1004
About DX OI - Logs....1004
DX OI - Logs Architecture....1005
Who Can Use DX OI - Logs....1006
Supported Log Ingestion Channels....1007
Supported Log Types....1007
Installation and Configuration for Log Ingestion....1008
Sizing Guidelines....1009
Verify System Requirements....1011
Download the Installer....1011
Install Standalone Log Collector....1012
Deploy Standalone Log Collector - Docker....1022
Agent-less Log Collection Methods....1023
Agent-based Log Collection Methods....1027
Authenticate Log Ingestion....1034
Configure Google Cloud Pub/Sub for Log Ingestion....1034
Configure Custom Configuration Files for Log Collector....1036
Set up Custom Log Ingestion....1041

Custom Logs Parsing Rules....1043
Using DX OI - Logs....1064
Authentication and Authorization....1065
Access DX OI - Logs....1065
DX OI - Logs User Interface....1065
Log Alarm Configuration....1093
Log Ingestion Throttling....1116
DX OI - Logs APIs....1121
Troubleshoot DX OI - Logs....1137
DX Dashboards....1137
Reference....1139
DX Operational Intelligence APIs....1139
Authentication and Authorization of APIs....1139
DX Operational Intelligence Query API....1142
Service Analytics APIs....1146
Write APIs....1147
Read APIs....1162
Service Templates APIs....1170
Get Services for an Inventory....1188
ServiceRepo APIs....1193
Situation Alarm Action APIs....1210
Situation Clustering Dimensions APIs....1229
Topology Processor APIs....1238
Get Correlation Information....1238
Get Tenant Correlation Rules....1240
Dry-Run Correlation Rules API....1245
Update Correlation Rules API....1247
Update Specific Fields API....1251
DX SaaS APIs....1253
DX Platform Catalog APIs....1253
Product Usage Collector API....1254
Add or Update the CI Attributes....1256
Authenticate User....1259
Create Email Channel....1259
Create Policy....1260
Delete Configured Mail Server....1262
Delete Email Channel....1262
Delete ITSM Channel....1263
Delete Message Template....1263
Delete Policy....1264

List All Policies....1264
List Channels With Filters Applied....1266
List Existing Channels....1268
List Message Templates....1270
List Specific Policy Details....1272
Retrieve Auth Config....1274
Retrieve Licensed SKUs....1274
Retrieve Linked Template Names....1277
Retrieve Mail Server....1279
Retrieve Template Details....1279
Retrieve Tenant Cohort ID....1280
Save Mail Server Configuration....1281
Test Mail Server Configuration....1281
Update Possible Values Mapping of Incident Fields....1282
Update Policy by Linking Channel and Message Template....1284
Global Maintenance APIs....1286
Device Lookup....1286
Service Lookup....1288
Create a Schedule....1289
Retrieve Schedules for a Tenant....1297
Retrieve Information for a Schedule....1299
Retrieve Information for Bulk Schedules....1301
Delete Schedules....1303
Edit a Schedule....1305
Stop a Schedule....1306
Retrieve the Maintenance Windows for the Specified Schedules....1306
Retrieve Maintenance Window Schedules for a Service....1309
Retrieve Schedules for a Configuration Item (CI)....1310
Get Overlap Sub Services....1314
Validate Maintenance Schedule Name....1315
Usage Data (Telemetry)....1317
Documentation Legal Notice....1319

DX SaaS Network Requirements

This section provides a list of IP addresses that must be allowed for the data centers you currently utilize to avoid any network-related security disruptions.

The DX SaaS infrastructure is upgraded to increase reliability with minimum downtime for maintenance windows. This optimization requires mandatory network configuration updates to your firewall policies to avoid network-related security disruptions and impacts on both the inbound (for example, webhook) and outbound (for example, agents, logs, alarms, RESTMon) communication. The IP addresses of the new hardware must be allowed in your deployment when the new hardware is activated. Failure to allow communication to the new IP addresses may affect the ability to ingest monitoring data, and, given the nature of some monitoring technologies, this could potentially impact the performance of the monitored system. Contact your Network Administrator to ensure that the following IP addresses are allowed:

US Datacenter (https://axa.dxi-na1.saas.broadcom.com)

- Inbound:
  - 34.145.151.0/24
- Outbound
  - 34.96.90.96/28
  - 34.150.194.136/29

EU Datacenter (https://axa.dxi-eu1.saas.broadcom.com)

- Inbound:
  - 34.141.238.0/24
- Outbound
  - 34.117.194.112/28
  - 34.141.162.16/29

Upcoming Features/Enhancements

> **NOTE**
> **Disclaimer!** This release note is forward-looking information and is subject to change or withdrawal by Broadcom at any time until the actual SaaS rollout date, which remains at Broadcom's sole discretion. Check the Release Notes for the updated release notes when the SaaS rollout happens. The DX Operational Intelligence SaaS status and rollout updates can be found here: DX SaaS Service Status

The Upcoming release includes the following new features and enhancements:

- Cross-Tenant Data Supported in DX Dashboards Reports
- Rotate Topology View
- App Synthetic Monitor (ASM) Enhancements
- Fixed Issues

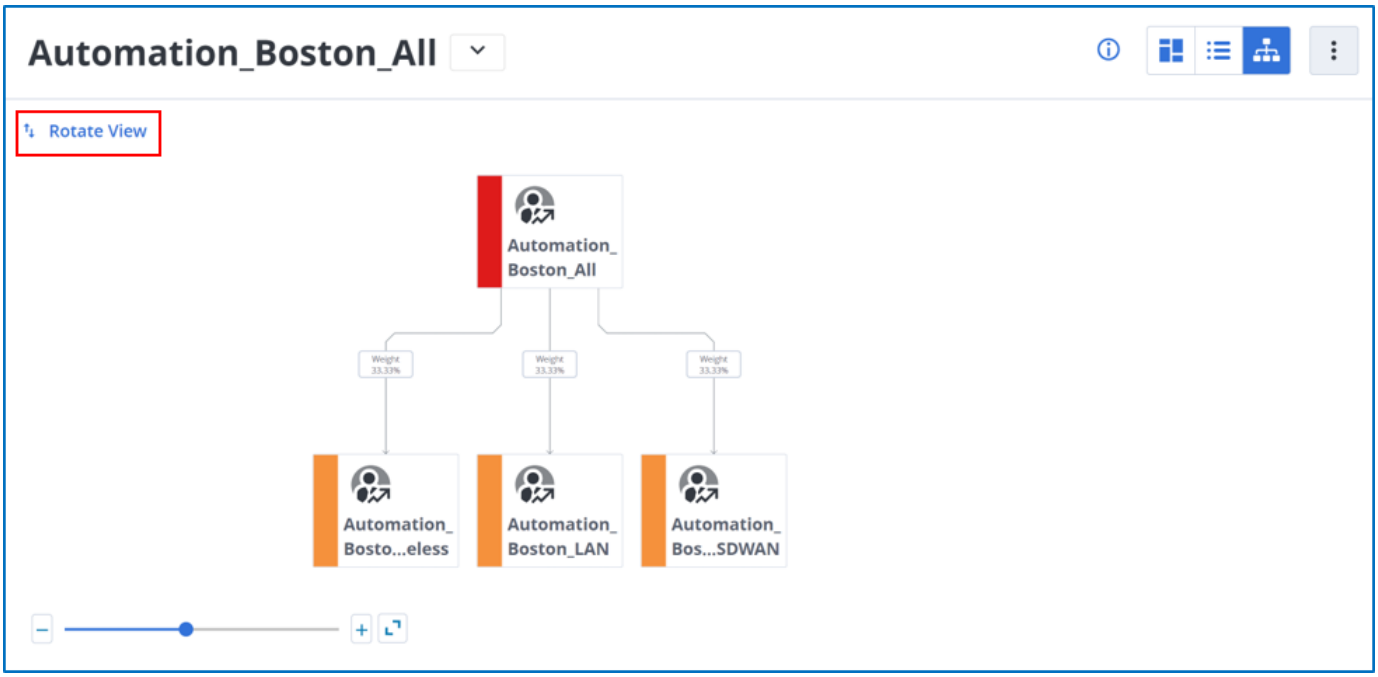
Cross-Tenant Data Supported in DX Dashboards Reports

Starting with this release, if a dashboard includes data from other tenants, the generated report includes this data as well. In the earlier version, data only from the current tenant was supported in the reports.

> **NOTE**
> When the dashboard has data from other tenants, the cross-tenant data panels display data for all the universes (All Access universe). However, the current tenant data panels display data only for the selected universe.

Rotate Topology View

Starting with this release, you can view the **Topology View** both horizontally and vertically on the **Service Details** page. Click **Rotate View** to rotate the view.



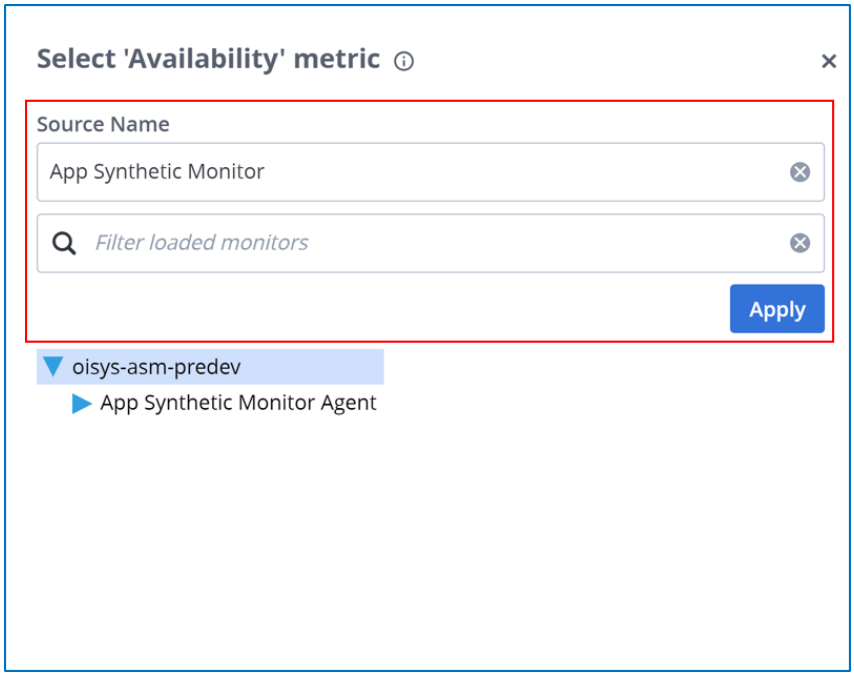
In the earlier version, the service topology was displayed only vertically.

App Synthetic Monitor (ASM) Enhancements

Starting with this release,

- You can provide any name for the `introscope.agent.agentName` property while configuring the ASM Agent settings in DX Application Performance Management to get the **Last Check Status** metric in DX Operational Intelligence.
  In the earlier version, only **App Synthetic Monitor Agent** was supported as a value for this property.

  NOTE
  For more information, see the Configure DX App Synthetic Monitor Agent section.
- The **Select 'Availability' metric** dialog in the **Service Details** panel provides a search filter to select the Availability metric. By default, the source name is selected as **App Synthetic Monitor**. You can change this value to the configured ASM source name.



NOTE
For more information, see the Verify ASM Data in DX Operational Intelligence section.

Fixed Issues

In this release, the following issue is fixed:

- DX Operational Intelligence - SaaS was not accessible post maintenance. This issue is fixed now.

Release Notes

This Release Notes provides information about the new features, defects fixed, and the known issues of DX Operational Intelligence.

- Content Reference
- DX Operational Intelligence
- Compatibility Matrix
- Third-party Software Requirements

Content Reference

The following table lists the sections where you can find the information in the DX Operational Intelligence documentation:

Topic		Section in DX Operational Intelligence
Channels		Integration > Outbound Integration
Connector Parameters		Integration > Inbound Integration
Custom Situation Definitions		Capability Configurations
DX Dashboards		Using
DX Gateway		Integration > Outbound Integration
Integration with CA Products		Integration > Inbound Integration
Integration with Third-Party Products (RESTMon)		Integration > Inbound Integration
Message Templates		Integration > Outbound Integration
OI Universes		User and Role Management
Policies		Integration > Outbound Integration
Proxy Configuration (On-Premise Only)		Integration
Role Management		User and Role Management > Role Management
Token Management		Integration
User Management		Role Management > User and Role Management
APIs	DX Platform APIs	Reference > DX SaaS APIs
	DX Operational Intelligence APIs	Reference
	Global Maintenance APIs	Reference > DX SaaS APIs
	Ingestion APIs	Integration > Inbound Integration > Third-Party Integration
	RESTMon APIs	Integration > Inbound Integration > Third-Party Integration > RESTMon > Using

DX Operational Intelligence

This section provides the list of new features, current and fixed issues for each of the DX Operational Intelligence SaaS releases.

Release	Description
2023.9.1 Release Notes	• Cross-Tenant Data Supported in DX Dashboards Reports • Rotate Topology View • App Synthetic Monitor (ASM) Enhancements • Fixed Issue
2023.8.2 Release Notes	• Manual Grouping of Alarms • Lifecycle Events Tab Includes Notifications and Alarm Actions Details • Exclude Lifecycle Events • Enhancements to Metric Group Alarms • Supportability Metrics for Notification Channels and Alarm Actions • Share Service Analytics Filters and Layouts • Search Filter Added on Manage Adjustments Page
2023.8.1 Release Notes	• Save Monitored Inventory Filters as Queues • Stable Situation Filter Attribute Changes • Log-Based Triaging for Alarms and Monitored Inventory • Out-of-the-box APM Metric View Dashboards
2023.7.1 Release Notes	• Support for Sharing of Alarm Queues
2023.6.1 Release Notes	• Templates for Service Creation • Adjust Historical Outages • Application Alarms Inspector for Deep Triaging • Display Annotations on All Alarms Page • Authenticate Log Ingestion • Support for Cross-Tenant Data Dashboards • OOTB Dashboards for IBM Websphere MQ • View Continuous Delivery Directory (CDD) Metrics in DX Dashboards • DX Dashboards - Fixed Defects • Known Issues
2023.5.1 Release Notes	• Metrics-Based Alert Configuration • OI Alarm Metrics Overview Dashboard • Launch OI Alarm Metrics Overview Dashboard from Alarm Analytics • Last Updated Column Now Displays Timestamp • Display Situation Details on All Alarms Page Situation • Ticket Details Include Link to First Alarm • Script Operator Support • Projection Filter Supports JSON Pointer Syntax • Theme Preferences Changes • Known Issues

Release	Description
2023.4.2 Release Notes	• Added Support for Integration with BMC Helix • Configure Service Health Status • Override Default Alarm Severity in Service Health Calculation • Calculate Service Health Based on SLI • Policy Filter Fixes for Notifications • Set Expiration Date for Tokens • Retrieve Maintenance Window Schedule for Service Using API • Retrieve Licensed SKUs Using API • Display User Name Instead of Tenant ID • Embed DX Dashboards in Other Applications • Known Issues
2023.4.1 Release Notes	• View Metrics in Context of Selected Entities from Monitored Inventory • Enable Time Range for Reports • Reports Page Enhancements • New Out-of-the-box Dashboards • Embedded Dashboards Changes
2023.3.2 Release Notes	• General Availability of Log Analytics • Syslog Log Events Enrichment • Launch Alarm Contextual Dashboards • Launch CI Contextual Dashboards • Reopen ITSM Ticket With Time Criteria
2023.3.1 Release Notes	• Alerts for Service Level Indicators and Objectives • Monitored Inventory Quick Filters • Alerts for Connector Health • Filter Monitored Inventory by Attribute • Insights Enhancements • DX Operational Intelligence SaaS to Automic Integration • DX Dashboards • DX Platform Improvements
RESTMon 2.2.1 Release Notes	• In-process Filters to Filter and Ingest Only Required Data • Monitoring Overview Dashboard Improvements • RESTMon Docker - Added Support for Installing on Existing PV/PVC • Performance Improvements • Vulnerabilities Fixes
2023.1.1 Release Notes	• Additional Alarm Filter Attributes for Maintenance Windows • Additional Filter Attributes for SLIs and Monitoring Groups • CI Attributes Supported for Incident Fields Mapping • CI Attributes Supported for Ticketing and Notifications • DX Dashboards New Features/Enhancements • Filtering Topology, Metrics, and Alarms Before Ingestion • Handling Alarm Updates from Source Products • HTML Format for Email Messages • New DX Gateway Version Released • Simplified Filter for All Alarms • Tenant Profile Management • Ternary Expressions Supported for SLIs/SLOs • Updating Specific Fields in Raw Alarms Using API

Release	Description
2022.12.1 Release Notes	• **New Features:** – Insights into Services and Alarms • **Enhancements:** – Additional Privileges for Alarms and Situations – Automate Alarm Actions for Historical All Alarms – Custom Situation Preview Enhanced – Customize Situation Definition Name – DX Dashboards – DX Gateway Enhancements – Enable or Disable Algorithmic Clustering – Embed Dashboards in SA Service Details – Exclude SLO Calculation During Maintenance Window – General Availability of SLI/SLO – Message Templates Enhancements – Mute Existing Open Alarms During Maintenance Window – ServiceNow Version Support – Timezone Support for SLO Calendar Window Calculation – User Tokens Support for Maintenance APIs – View Pipeline Errors • **Known Issues:** – New SLI Alarm is Not Created – Existing Alert is Not Closed – SLI Alarm is Not Generated

22.08 Release Notes	• **New Features:**
	– OI Universes
	– Role-Based Access Privileges Enhancements
	– Ticket Enrichment Rules
	• **Enhancements:**
	– Documentation Changes
	– Service Analytics Enhancements
	  • SLO Calendar Support
	  • Added Anomalous Information on Services Chart
	  • Enhanced SLI and SLO Creation
	– Alarm Analytics Enhancements
	  • Column Customization and Personalization Support
	– Performance Analytics Enhancements
	  • Anomaly Detection Improvement
	– Maintenance Windows Enhancement
	– Dashboard Enhancements
	– Vulnerability Fixes
	– Known Issues
	– Defects Fixes
	• **Fixes:**
	– Vulnerability Fixes
	– Known Issues
	– Defects Fixes

22.06 Release Notes	• Alarm Analytics Enhancements

Release	Description
22.05 Release Notes	• Situation Custom Definitions Enhancements – Added Stabilization TimeLine Options – Added Service Hierarchy Option in Alarm Clustering Criteria – Support for Out-of-the-Box Custom Situation Definitions – Customize Alarm Clustering Criteria Field – SLI Validation for Complex Aggregations and Metric Config APIs for Handling Multiple SLI/SLO • RESTMon 2.2 Download From Settings Page • Dependent Features Auto Selected • DX Dashboards Enhancements – RESTMon Self-Monitoring Dashboards Enhancements – UMA Dashboards Enhancements • Known Issues
22.04 Release Notes	• Service Creation Enhancement • Support for Lifecycle Event for Maintenance Alarms • Improved Capacity Monitoring and Capacity Forecasting Capabilities in Capacity Analytics • RESTMon: New Version RESTMon 2.2 Released – RESTMon Performance Improvement and Memory Optimization – Simplify 3rd Party Metric paths and Display on Performance Analytics – Polling Interval by Data Type for Polling Integrations – Token Authentication Support for Streaming Integrations – SCOM 2019 UR1 Support – Improved Alarm and Topology State Management – Alarms Auto-Retry – Dynatrace Integration using Webhooks – Deprecated Features in RESTMon 2.2 Release • Associate Policy with Message Template on Policy Page • Channel Page Changes • Message Template Page Changes • DX Dashboards – Enhancements to the Capacity Analytics Dashboards – Dashboard Sharing Enhancement

Release	Description
22.03 Release Notes	• Alarm Analytics – Support Configuration for Deviation Anomalies • Maintenance Window – Support for Duplicate Maintenance Window – Manage Active Maintenance Windows – Added Drop-down Filter Descriptions While Creating Maintenance Windows • Performance Analytics – Support for Show Entities Without Metrics – Added Unit on the y-Axis of Metric Chart • Service Analytics – Show Expired Topology Vertices – Added SLI/SLO Data in the Service Overview Page – Chart Preview for a Metric on SLI Metrics List – Support Automatic Anomaly Detection for SLIs – Enhancement to the Layer Attribute on the Topology Details Page – Added Topology Correlation Settings Access Privileges • Situation Alarms – Support for Topology tab Within Situation Alarm Template – Added Impacted Entity(s) Attribute in Overview Tab of Situation Details Page – Custom Situation Definitions for Situation Clustering – Support for Lifecycle Events tab for Situations • New Bi-directional ITSM Integration with WolkenSoft • DX Dashboards – New OOTB AXA Dashboards – Shared Dashboard Link Enhanced: – Use Existing Metrics Values for Threshold • Known Issues • Defects Fixed
22.02 Release Notes	• DX Dashboards Features and Enhancements – Pivotal Cloud Foundry (PCF) Dashboards – DX APM/AXA: Product Usage Dashboard – UMA Dashboards Changes – Connector Health Dashboard Changes • Fixed Defects

Release	Description
22.01 Release Notes	• Role-Based Access Control – Security Administrator Role Introduced • Alarm Analytics – Support for Creation of Alarm Queue Based on Status of Alarms and Age of the Alarms to Trigger Policy-driven Notifications • Connector Parameters Page Changed • DX Dashboards – APM-IBM ACE Dashboards – Anomalies Report – APM-Mainframe Dashboards – Alarm/Incident Monitoring Dashboard Changes • Maintenance Window: Create Maintenance Window based on Dynamic Grouping • Policies: Age-Based Alerting Using Policies • Service Analytics – Topology Groupby Support – Support for Service Level Indicator/Service Level Objective – Get Services for an Inventory API Support • Token Management • Fixed Defects • Known Issues

2023.9.1 Release Notes

The 2023.9.1 release includes the following new features and enhancements:

- Cross-Tenant Data Supported in DX Dashboards Reports
- Rotate Topology View
- App Synthetic Monitor (ASM) Enhancements
- Fixed Issues

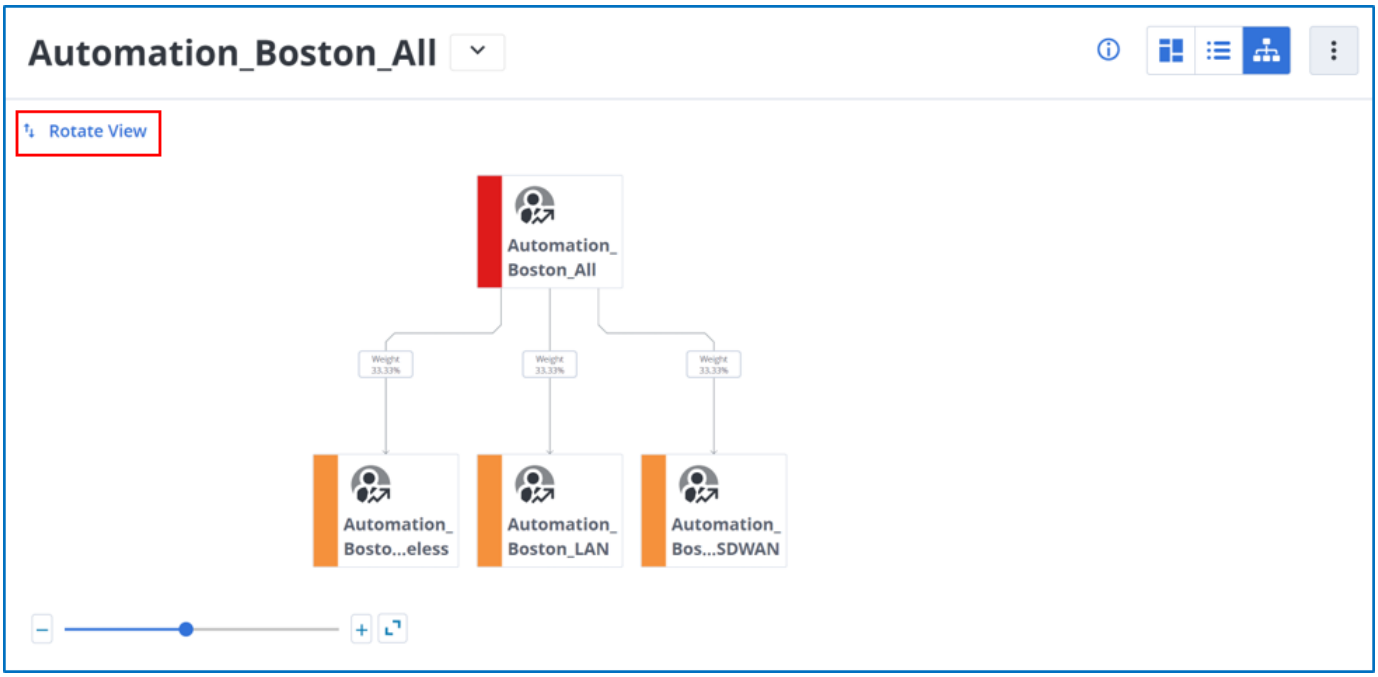
Cross-Tenant Data Supported in DX Dashboards Reports

Starting with this release, if a dashboard includes data from other tenants, the generated report includes this data as well. In the earlier version, data only from the current tenant was supported in the reports.

> **NOTE**
> When the dashboard has data from other tenants, the cross-tenant data panels display data for all the universes (All Access universe). However, the current tenant data panels display data only for the selected universe. For more information, see the Reports section in the **DX Dashboards** documentation.

Rotate Topology View

Starting with this release, you can view the **Topology View** both horizontally and vertically on the **Service Details** page. Click **Rotate View** to rotate the view.

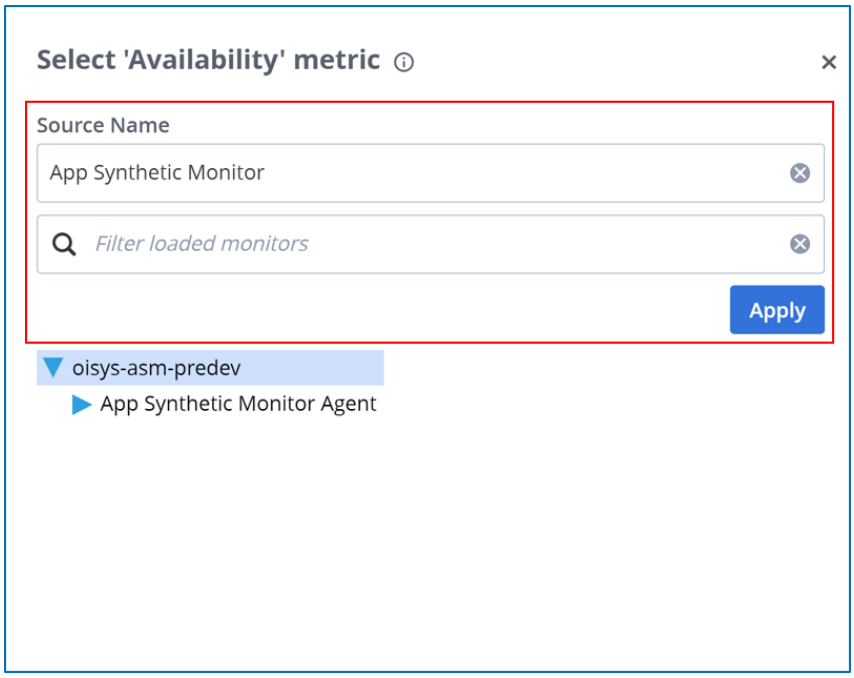


In the earlier version, the service topology was displayed only vertically.

App Synthetic Monitor (ASM) Enhancements

Starting with this release,

- You can provide any name for the `introscope.agent.agentName` property while configuring the ASM Agent settings in DX Application Performance Management to get the **Last Check Status** metric in DX Operational Intelligence.
  In the earlier version, only **App Synthetic Monitor Agent** was supported as a value for this property.
  NOTE
  For more information, see the Configure DX App Synthetic Monitor Agent section.
- The **Select 'Availability' metric** dialog in the **Service Details** panel provides a search filter to select the Availability metric. By default, the source name is selected as **App Synthetic Monitor**. You can change this value to the configured ASM source name.



NOTE
For more information, see the Verify ASM Data in DX Operational Intelligence section.

Fixed Issues

In this release, the following issue is fixed:

- DX Operational Intelligence - SaaS was not accessible post maintenance. This issue is fixed now.

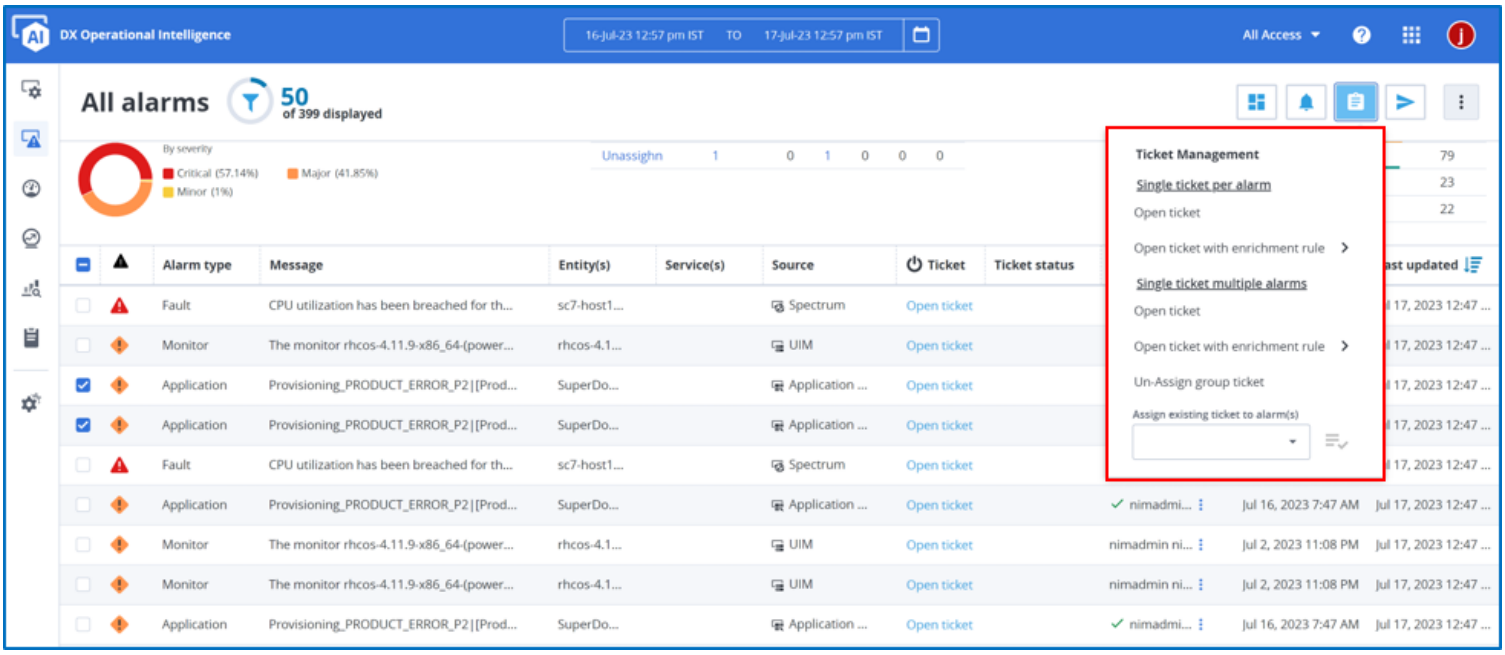
2023.8.2 Release Notes

The 2023.8.2 release includes the following new features and enhancements:

- Manual Grouping of Alarms
- Lifecycle Events Tab Includes Notifications and Alarm Actions Details
- Exclude Lifecycle Events
- Enhancements to Metric Group Alarms
- Supportability Metrics for Notification Channels and Alarm Actions
- Share Service Analytics Filters and Layouts
- Search Filter Added

Manual Grouping of Alarms

If multiple alarms with a similar root cause are ingested into DX Operational Intelligence, you can now create separate tickets for each of the alarms or you can create a single ticket for all those alarms. The **Ticket Management** icon on the **All Alarms** page provides the options as shown in this image:



- **Single ticket per alarm:** Creates a separate ticket for each of the selected alarms.
- **Single ticket multiple alarms:** Creates a single ticket for all the selected alarms.

> **NOTE**
> For more information, see the Ticket Management section.

Lifecycle Events Tab Includes Notifications and Alarm Actions Details

Starting with this release, the **Lifecycle Events** tab displays the following information as well:

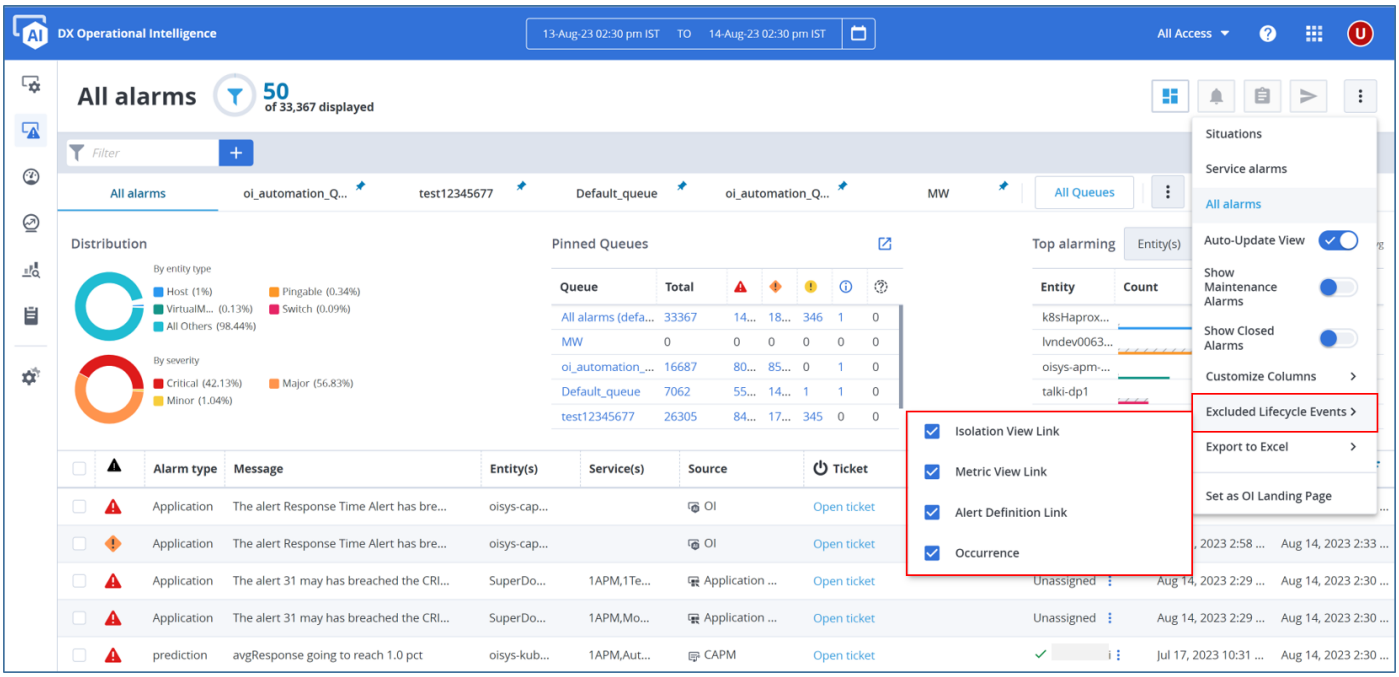
- Status (Success or Failure) of the automatic email notifications for both raw alarms and situations.
- Status (Success or Failure) of the automatic webhook notifications for both raw alarms and situations.
- Any updates to Southbound Gateway.

  > **NOTE**
  > The updates are displayed only for raw alarms and not situations.
- Any updates to **earliestSourceAlarmURL** and **mostImpactedSourceAlarmURL** in the incident.
- Any updates to the alarm fields using APIs.
- Any alarm updates to the raw alarm or situation actions through APIs.

> **NOTE**
> For more information, see the All Alarms section.

Exclude Lifecycle Events

Starting with this release, you can exclude updates from displaying in the **Lifecycle Events** tab on the **All Alarms** page. Click **Excluded Lifecycle Events** and select the required options so that related events are not displayed in the **Lifecycle Events** tab.



By default, all the four options are selected. To add more fields to this list, you may contact **Broadcom Support**.

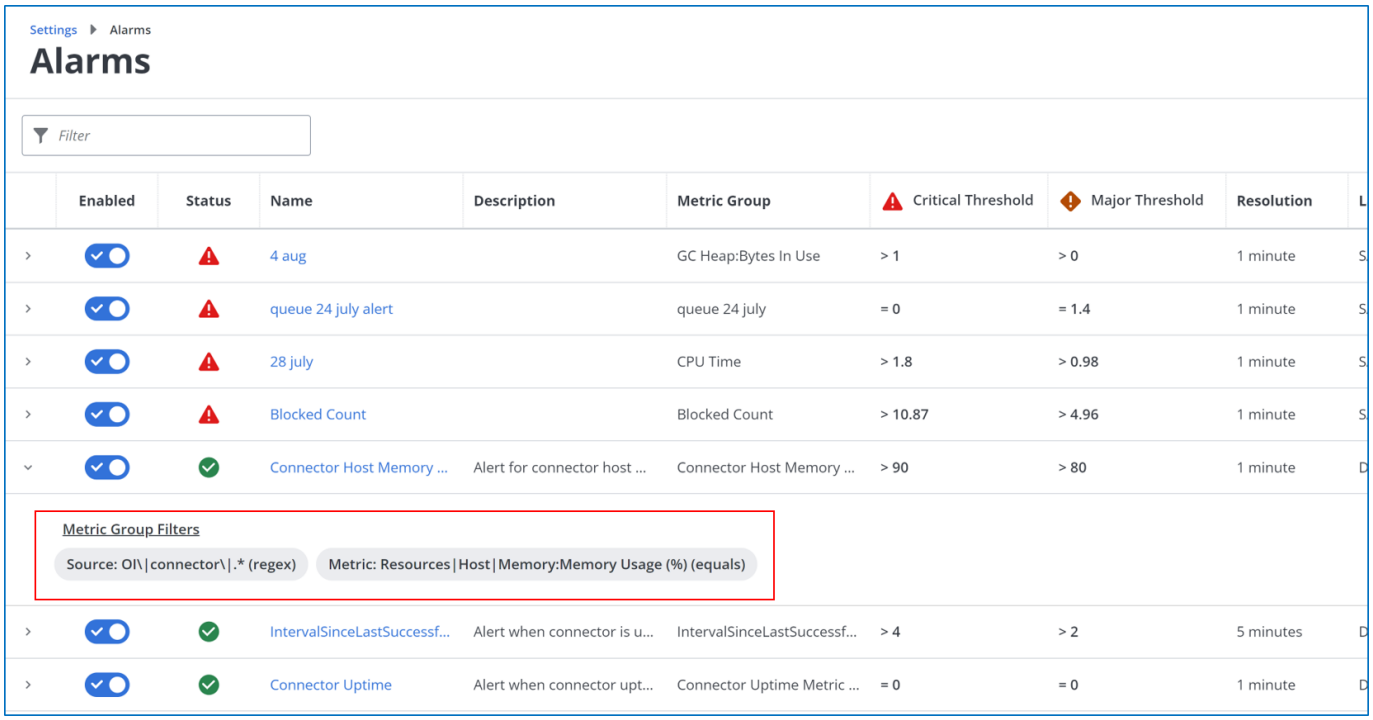
NOTE

- The information is excluded only for raw alarms and not for situation alarms. That is, the updates are not excluded in the **Lifecycle Events** tab on the **Situations** page.
- The selected options persist only for the current session.
- For more information, see the All Alarms section.

Enhancements to Metric Group Alarms

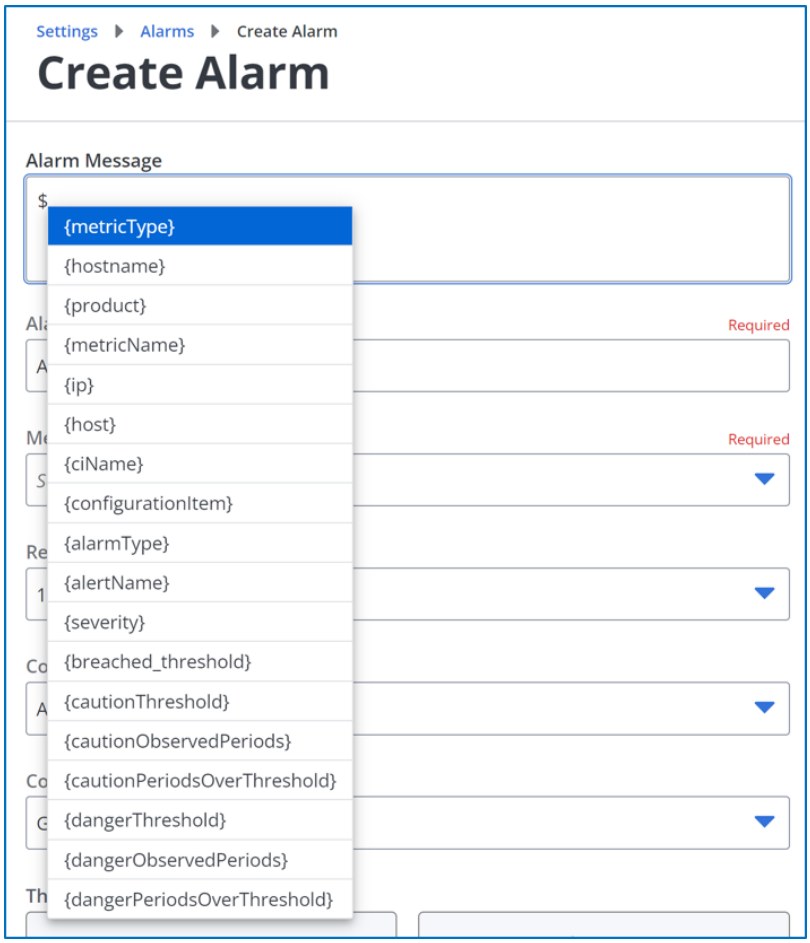
This release includes the following enhancements to the Metric Group alarms:

- The **Alarms** page for the Metric Groups (**DX SaaS Settings > Manage Alarms > Alarms**) now displays the following information as well:
  - **Status:** Displays the status of the alarm. **Values:** Major Threshold Breached, Critical Threshold Breached, and Normal
  - **Resolution:** Displays the resolution that is configured for the alarm.
  - **Metric Group Filters:** Displays the filter for the metric group. Expand the alarm to view the filter as shown:



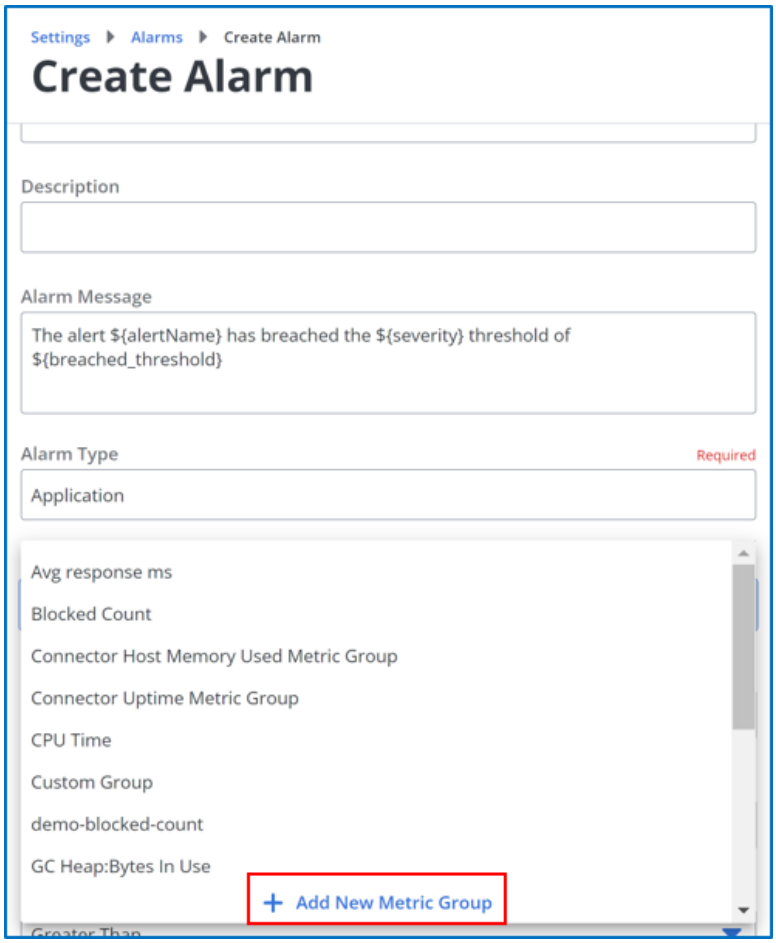
- When you create or edit an alarm,
  - You can now customize the alarm message using the metric attributes. For example, you can add the following message using the attributes. ${metricName} on host ${hostname} has breached the ${severity} threshold of ${thresholdValue}.

    To view the list of metric attributes, press **$** as shown:



In the existing version, the alarm message is set by default and you cannot change this message.

– You can select an existing metric group or you can create a metric group on the fly using the **+ Add New Metric Group** option.



– Also, you can also associate this alarm with an existing policy or you can create a policy on the fly using the **+ Add New Notification Policy** option.

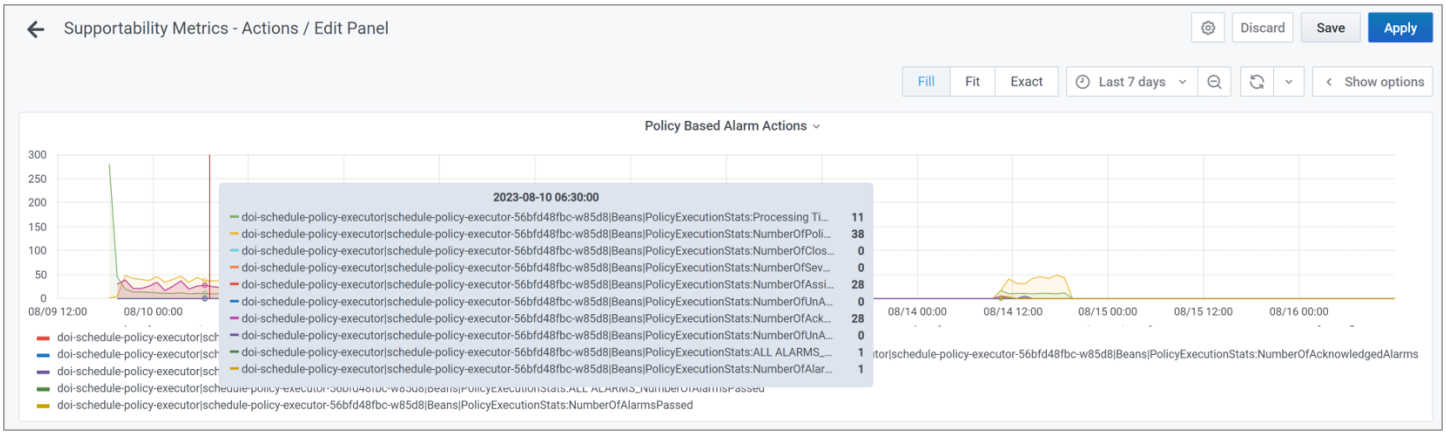
NOTE
For more information, see the Configure Alarms for Metrics Groups section.

Supportability Metrics for Notification Channels and Alarm Actions

Starting with this release, the following metrics are supported at the tenant level for channel-related notifications and also alarm actions. Using these metrics, you can audit the number of notifications that succeeded or failed. You can also audit the number of alarm actions that succeeded or failed.

Category	Metrics
Email and Webhook Notifications	• Delivered Message Count • Failure Message Count • WebHook Response Time (ms) (Only for Webhook)
Tickets	• Automatic Ticket Update Success Count • Automatic Ticket Update Failure Count • Close Ticket Success Count • Close Ticket Failure Count • Create Ticket Success Count • Create Ticket Failure Count • Manual Ticket Update Success Count • Manual Ticket Update Failure Count
Alarm Actions	• Number of Alarms Passed • Number of Closed Alarms • Number of Assignee Changed Alarms • Number of Severity Changed Alarms • Number of Assigned Alarms • Number of UnAssigned Alarms • Number of Acknowledged Alarms • Number of UnAcknowledged Alarms • Number of Policies Processed • Processing Time

You can visualize these metrics in DX Dashboards as shown in the sample dashboard:



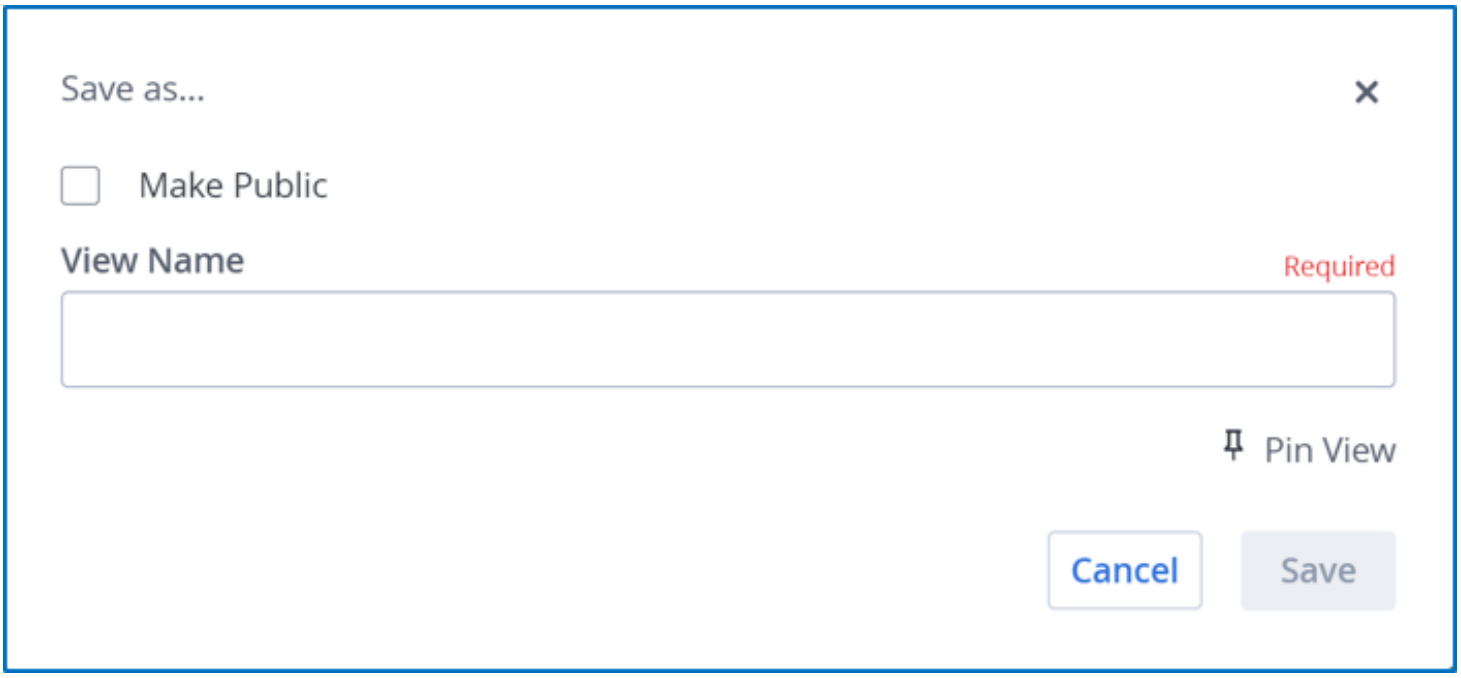
To visualize these metrics, you can create custom dashboards in DX Dashboards using the following information:

Category	Description
Email and Webhook Notifications	• **Data Source:** AIOps_Metrics (Advanced Query Builder) • **Query:** – **Source Name Specifier:** • Specifier: EXACT • Name: OI|Supportability|Metrics|All – **Attribute Name Specifier:** • Specifier: REGEX • Pattern: dxi-notify|.*|Beans\:.*
Tickets	• **Data Source:** AIOps_Metrics (Advanced Query Builder) • **Query:** – **Source Name Specifier:** • Specifier: EXACT • Name: OI|Supportability|Metrics|All – **Attribute Name Specifier:** • Specifier: REGEX • Pattern: doi-incidentmanagement|.*|TicketStats\:.*
Alarm Actions	• **Data Source:** AIOps_Metrics (Advanced Query Builder) • **Query:** – **Source Name Specifier:** • Specifier: EXACT • Name: OI|Supportability|Metrics|All – **Attribute Name Specifier:** • Specifier: REGEX • Pattern: doi-schedule-policy-executor|.*|PolicyExecutionStats\:.*

NOTE
For more information about creating custom dashboards using these metrics, see the DX Dashboards documentation.

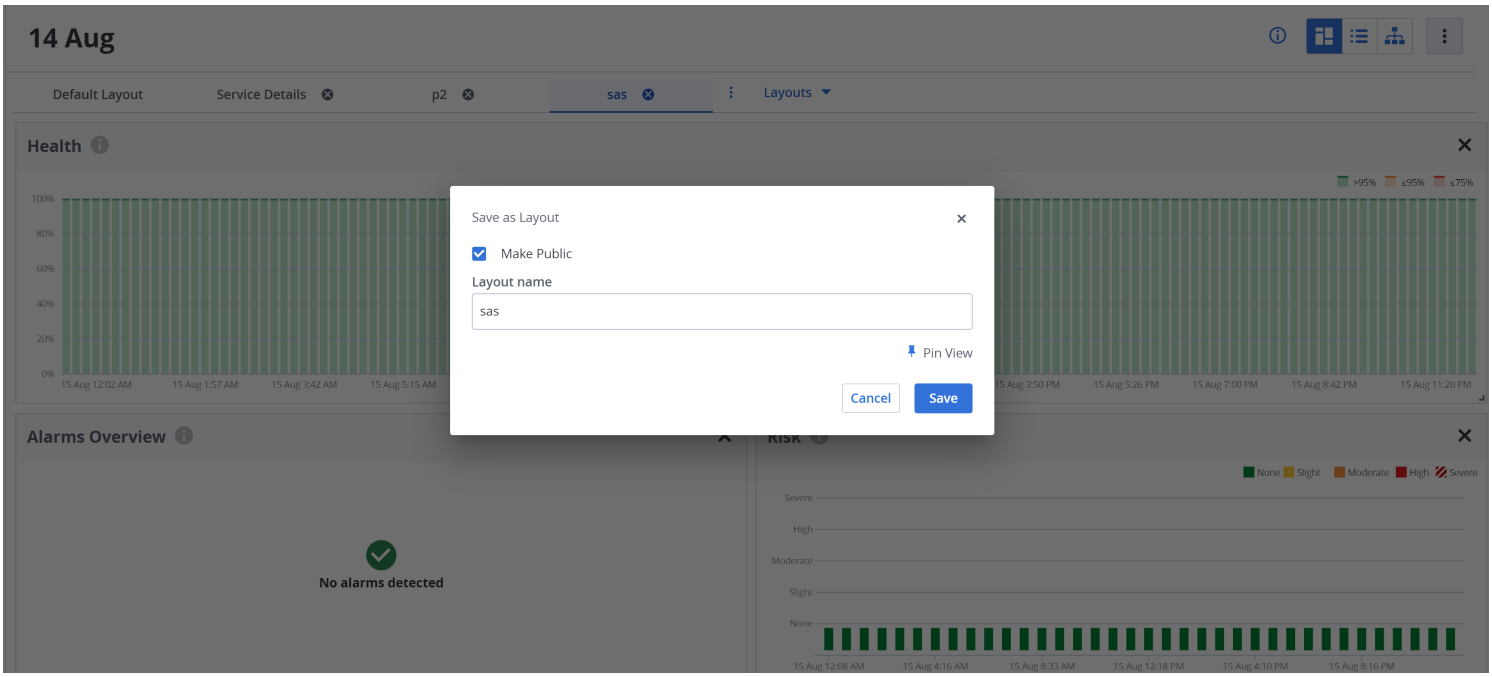
Share Service Analytics Filters and Layouts

Starting with this release, you can share the filters that are also known as **Views** on the **Service Analytics** page with other users in the tenant. Select the **Make Public** option to share the view.



The **Make Public** option for the default view is disabled because you cannot share the default view. However, you can make a copy of the default view and select the **Make Public** option to share it.

Similarly, you can share the layouts with other users using the **Make Public** option. To share the default layout, you must make a copy of the default layout and select the **Make Public** option.



NOTE
You cannot share the **Service Details** dashboard on the Service Details page. For more information, see the Service Overview Page section.

Search Filter Added on Manage Adjustments Page

The **Manage Adjustments** page now includes a search filter.

2023.8.1 Release Notes

The 2023.8.1 release includes the following new features and enhancements:

Save Monitored Inventory Filters as Queues

Starting with this release, when you filter the entities on the Monitored Inventory page, you can save the filter as a queue and you can also pin the queue that is frequently used. The pinned queue gets featured as a tab. You can save any number of queues but you can pin only a maximum of five queues.

> **NOTE**
> For more information, see the All Queues section.

Stable Situation Filter Attribute Changes

The **Is Stable Situation** filter attribute on the **All Alarms** and **Policy** pages is changed to **Situation State**. The changed filter attribute has the following values:

- **Situation State: Stable:** Filters all the situation alarms that are stable.
- **Situation State: Active:** Filters all the situation alarms that are active.
- **Situation State: Not Available:** Filters all the alarms that are not associated with any situation.

Log Based Triaging for Alarms and Monitored Inventory

Starting with this release, you can launch the logs in the context of alarms and monitored inventory for all the log types.

- **Alarms:** When an alarm is generated in DX Operational Intelligence by a source product for a particular host, you can launch the logs in the context of alarms in the OpenSearch dashboards for all the log types. This helps to narrow down the possible root cause of the raised alarm. The context is set by displaying the logs from 15 min prior to the creation time of the alarm and opening the dashboards for logs. You can subsequently change the filter criteria in the dashboard for further troubleshooting.
- **Monitored Inventory:** You can also launch logs in the context of monitored inventory for all the log types which helps in checking the logs that were being generated for the last 15 min for a given hostname. You can subsequently change the filter criteria in the dashboard for further troubleshooting.

> **NOTE**
>
> - In the earlier version, the contextual launch was supported only for syslogs.
> - If the host has multiple log types, then the **ao_itao_logs_*_<*>** index pattern is used to query all the log types.
> - Log-based triaging is not supported for custom logs ingested for the host. That is, the custom logs are not shown either in the context of alarms or monitored inventory.
> - Logs ingested for the host must have FQDN for them to participate in the host-based co-relation (log enrichment) and thus have the logs shown in the alarm and monitored inventory context.
> - For more information, see the Log-Based Triaging for Alarms and Monitored Inventory section.

Out-of-the-box Dashboards

This release of DX Dashboards includes the following APM dashboards out-of-the-box. These dashboards are available in the **APM-MetricView** folder.

- APM: GC Monitor
- APM: GC Heap
- APM: EM Overview
- APM: APIM Embedded Dashboard

You must view these dashboards in the Metric View of APM. Navigate to the folder in the tree and select to display the dashboard as a tab. Click the tab to view the data in the dashboard.

> **NOTE**
> For more information, see the APM-MetricView Dashboards section.

2023.7.1 Release Notes

The 2023.7.1 release includes the following new feature:

Support for Sharing of Alarm Queues

Prior to this release, the alarm queue that you save on the **All Alarms** page was accessible to all the users in the tenant. Any user could update or delete the queue.

Starting with this release, you can save the queue as private or you can share the queue with other roles. The private queues are available only to you. However, the shared queues are available to all the users assigned to those roles.

> **NOTE**
> For more information, see the All Alarms section.

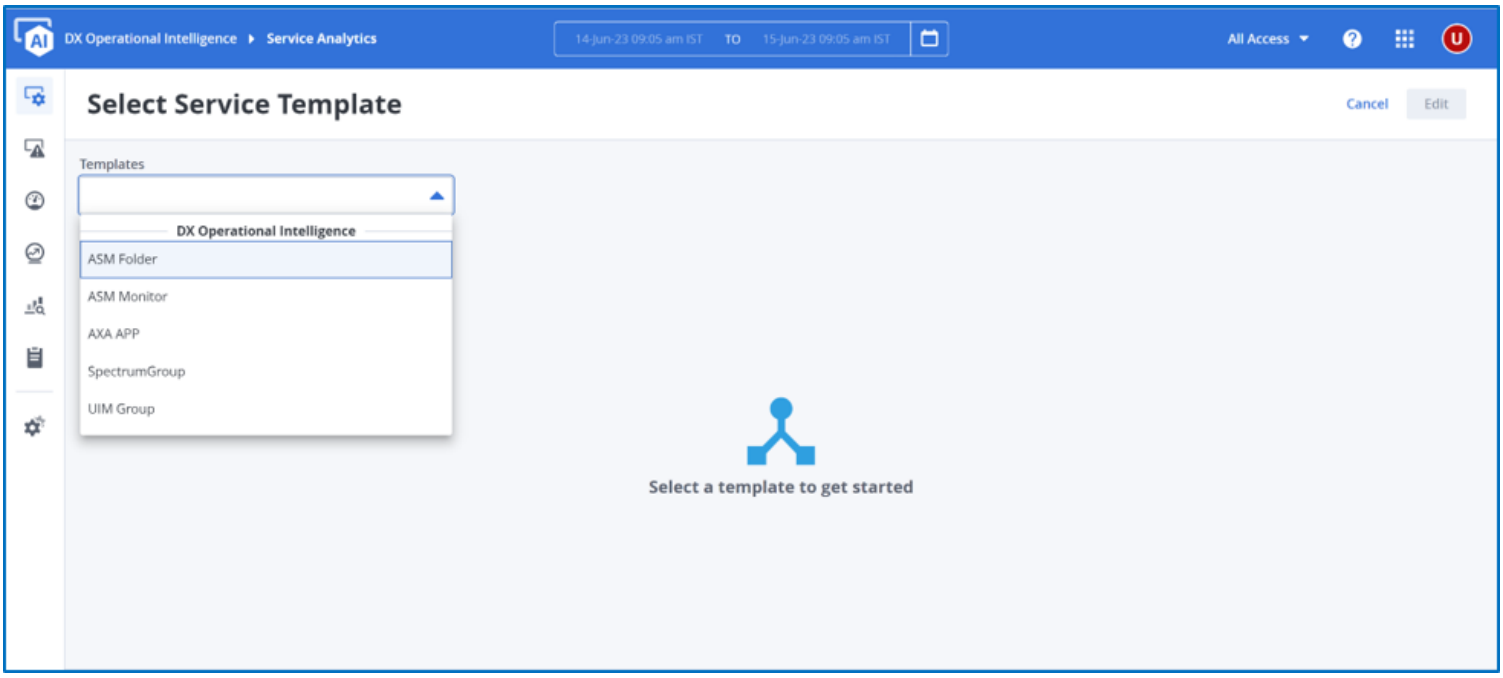
2023.6.1 Release Notes

The 2023.6.1 release includes the following new features, enhancements, and fixed issues:

- Templates for Service Creation
- Adjust Historical Outages
- Application Alarms Inspector for Deep Triaging
- Display Annotations on All Alarms Page
- Authenticate Log Ingestion
- Support for Cross-Tenant Data Dashboards
- OOTB Dashboards for IBM WebSphere MQ
- View Continuous Delivery Directory Metrics in DX Dashboards
- DX Dashboards - Fixed Defects
- Known Issue

Templates for Service Creation

Starting with this release, DX Operational Intelligence includes default service templates that serve as a starting point for creating a service. These templates are preconfigured with attributes that you can customize with the service content.



The following default service templates are available:

- **ASM Folder:** Use the ASM Folder template to create a service based on the ASM folders.
- **ASM Monitor:** Use the ASM Monitor template to create a service based on the ASM monitor (vertices).
- **AXA App:** Use the AXA App template to create a service based on the App Experience Analytics application. The application for which the service is being created must be integrated and monitored by Application Performance Management to be able to create the service.
- **SpectrumGroup:** Use the Spectrum Group template to create a service based on the Spectrum Groups.
- **UIM Group:** Use the UIM Group template to create a service based on the UIM group.

> **NOTE**
> For more information, see the Service Creation Templates section.

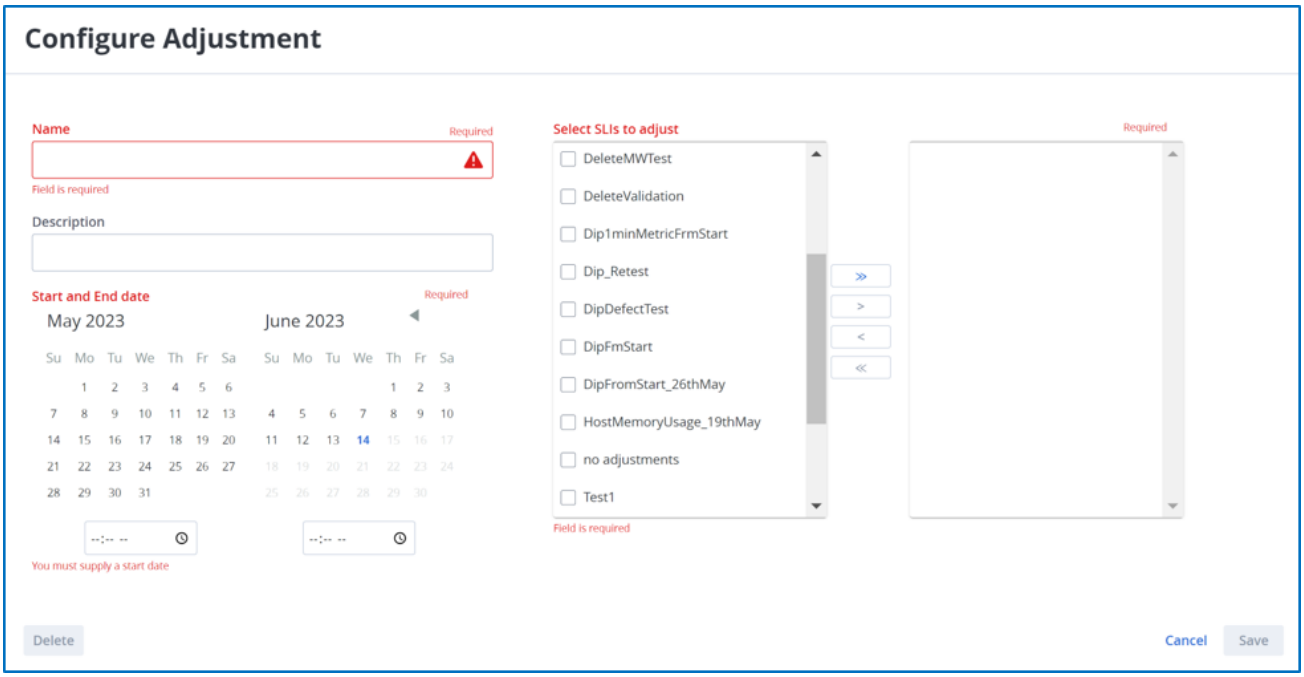
You can also create, update, and manage the service templates using APIs.

> **NOTE**
> For more information, see the Service Templates APIs section.

Adjust Historical Outages

The Service Owners are paid based on compliance with an SLI/SLO. To avoid any impact on compliance due to any outages, as a service owner, you can schedule maintenance windows for the services in DX Operational Intelligence. During those maintenance schedules, the Error Budget metric is not calculated for that service which ensures that you are not penalized for known maintenance schedules.

However, if any unexpected outages, the Error Budget metric dips and impacts compliance. To remove the impact of such outages on the Error Budget metric, starting with this release, you can make adjustments to the historical outages using the **Manage Adjustments** tab on the SLI/SLO page. You can add past timeframes to the SLIs where the metric should not be calculated.

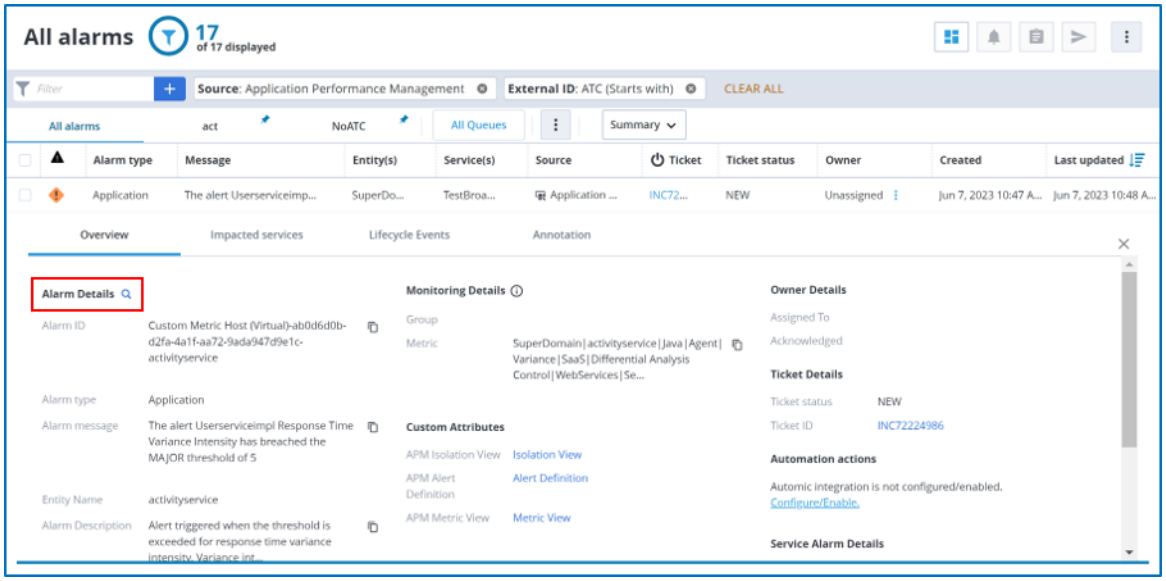


NOTE
For more information, see the Manage Adjustments section.

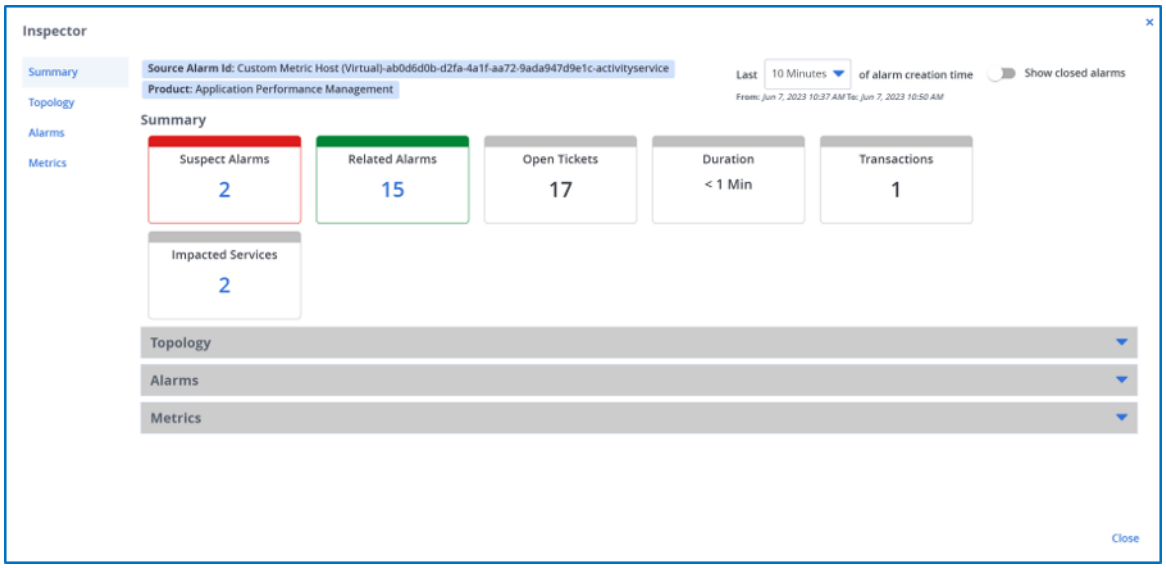
Application Alarms Inspector for Deep Triaging

When an application alarm is generated by DX Application Performance Management, there may be other alarms that are generated in the same time window which are related to the source alarm. Using the **Alarms Inspector**, you can view the associated alarms and possible suspect alarms, their details and also the metrics, and their corresponding topology from where the alarms are generated. The Inspector also ranks the alarms based on their topological structure which helps in determining the hotspot.

You can navigate to the **Inspector** view from the **All Alarms** page. Select the alarm filter as **ExternalId Starts with ATC** to display the alarms. Click the **Lens** icon to open the Inspector view for the required alarm.



The following image illustrates the Inspector view which is divided into the following sections: Summary, Topology, Alarms, and Metrics.



NOTE
For more information, see the Alarms Inspector section.

Display Annotations on All Alarms Page

Starting with this release, you can customize the columns on the **All Alarms** page to display the Annotation column. This column displays the message icon if any annotation is added to the alarm. Hover over the icon to view the added annotation. By default, this column is not displayed. Use the **Customize Columns** option in the Alarms View filter to select Annotation.

You can also filter the alarms by the following attributes on the All Alarms page:

- Annotation: Available
- Annotation: Not Available

NOTE
For more information, see the All Alarms section.

Authenticate Log Ingestion

Starting with this release, when the ingestion endpoint is APM Services Gateway, you must authenticate the log ingestion with the user token. You can generate the user token on the **Settings > Tokens** page. For more information, see the Tokens section.

Provide the user token as follows:

- **Standalone SLC:** Provide the user token as the authorization token at the time of the installation. For the Silent installation mode, add the user token in the config.json file.
- **Docker SLC:** Provide the user token as the value for the **LOGCOLLECTOR_REQUEST_AUTHORIZATION** environment variable.
- **Custom Logs Ingestion via Direct API:** Provide the user token as the Bearer Token in the HTTP Header.

Support for Cross-Tenant Data Dashboards

Before this release, you could visualize data in a dashboard only from one tenant. For large enterprises that have their data distributed across multiple tenants, triaging, monitoring, or reporting data across different tenants became challenging.

Starting with this release, you can create dashboards with data from multiple tenants. You can visualize all the data in a single dashboard in separate panels or in a single panel. Before you create the dashboard, provide the tenant token for the data sources to be used.

> **NOTE**
> For more information, see the For more information, see the Create Dashboards Using Cross-Tenant Data in the **DX Dashboards** documentation.

OOTB Dashboards for IBM WebSphere MQ

This release of DX Dashboards includes the following IBM WebSphere MQ dashboards. These dashboards are available in the **APM-IBM MQ** folder:

- IBM MQ Overview
- IBM MQ Queues
- IBM MQ Queue Managers
- IBM MQ Channels

> **NOTE**
> For more information, see the APM-IBM MQ Dashboards section in the **DX Dashboards** documentation.

View Continuous Delivery Directory (CDD) Metrics in DX Dashboards

Starting with this release, you can utilize the CDD data source plug-in to query and visualize your adaptive testing metrics. The CDD data source uses CDD REST APIs to query the underlying data services and allows you to configure dashboards to gain insight into your data. For more information, see the Continuous Delivery Director Integrations documentation.

> **NOTE**
> The CDD data source is not available out-of-the-box. To enable this data source, contact **Broadcom Support**.

DX Dashboards - Fixed Defects

The following defects were fixed:

- **Variable Not Fetching Historical Data for Time Range:** If you selected the time range, variables in the AIOps_Inventory data source were not fetching the historical data. This issue is fixed now.
- **Limited Email IDs Supported in Reports:** Before this release, there was a limit on the number of email addresses that you could provide in the report configuration. This issue is fixed now.

Known Issue

When a cross-tenant dashboard includes the AIOps_Metadata data source, the scheduled report displays errors.

2023.5.1 Release Notes

This release includes the following new features, enhancements, and fixed issues:

- Metrics-Based Alert Configuration
- OI Alarm Metrics Overview Dashboard
- Launch OI Alarm Metrics Overview Dashboard from Alarm Analytics
- Last Updated Column Now Displays Timestamp
- Display Situation Details on All Alarms Page
- Situation Ticket Details Include Link to First Alarm
- Script Operator Support
- Projection Filter Supports JSON Pointer Syntax
- Theme Preferences Changes
- Known Issues

Metrics-Based Alert Configuration

You can now configure alerts for metrics from APM, UIM, Spectrum, CA APM, or any third-party metrics that are ingested into DX Operational Intelligence. To configure the alerts, group the metrics by attributes and then set the Critical and Major thresholds for these groups. The **Metric Groups** tile on the **Settings** page enables you to group the metrics and the **Alarms** tile on the **Settings** page enables you to configure the alerts.

DX Operational Intelligence provides the following metric groups out-of-the-box for connectors:

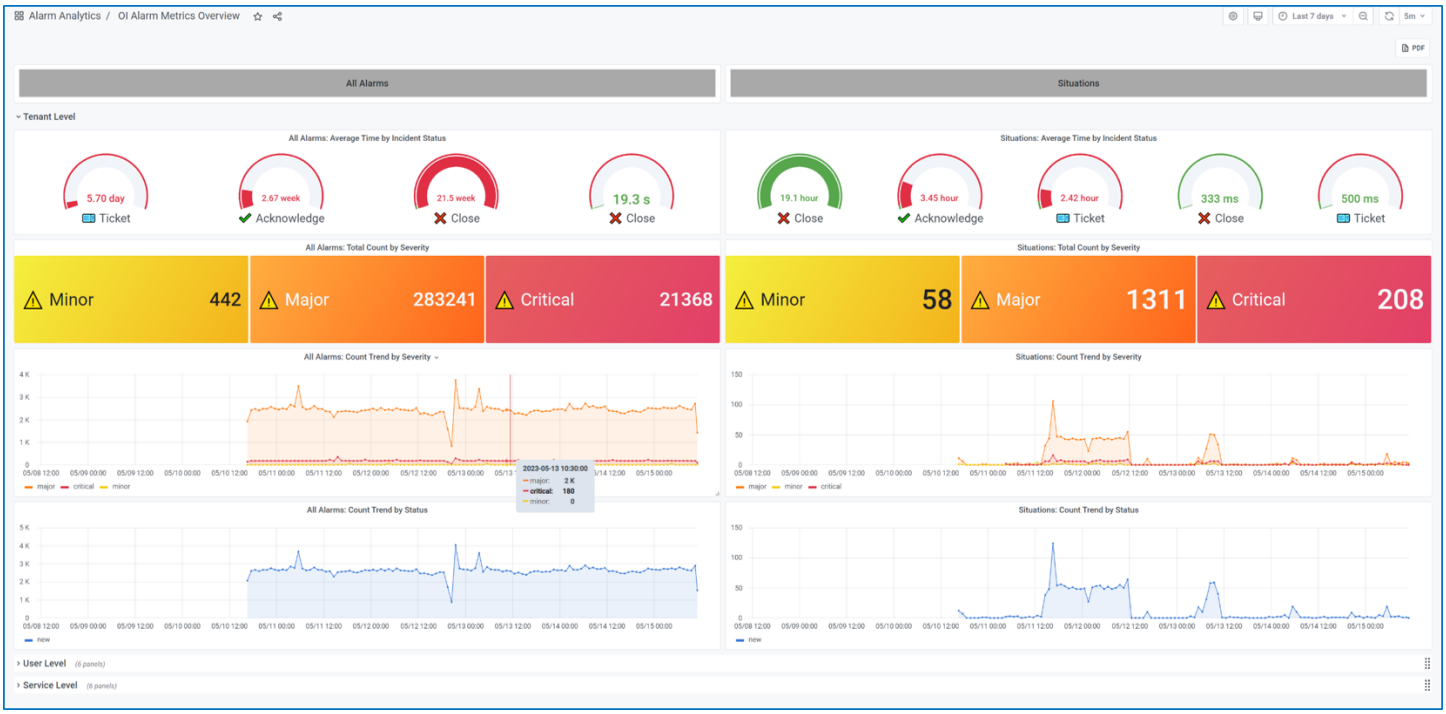
- Connector Host Memory Used Metric Group
- Connector Uptime Metric Group
- IntervalSinceLastSuccessfulPush Metric Group

You can edit these metric groups as required and you can also create metric groups for other metrics. After the metric group is created, set the Critical and Major thresholds for these groups to trigger alarms.

> **NOTE**
> For more information, see the Metric-Based Alert Configuration sections.

OI Alarm Metrics Overview Dashboard

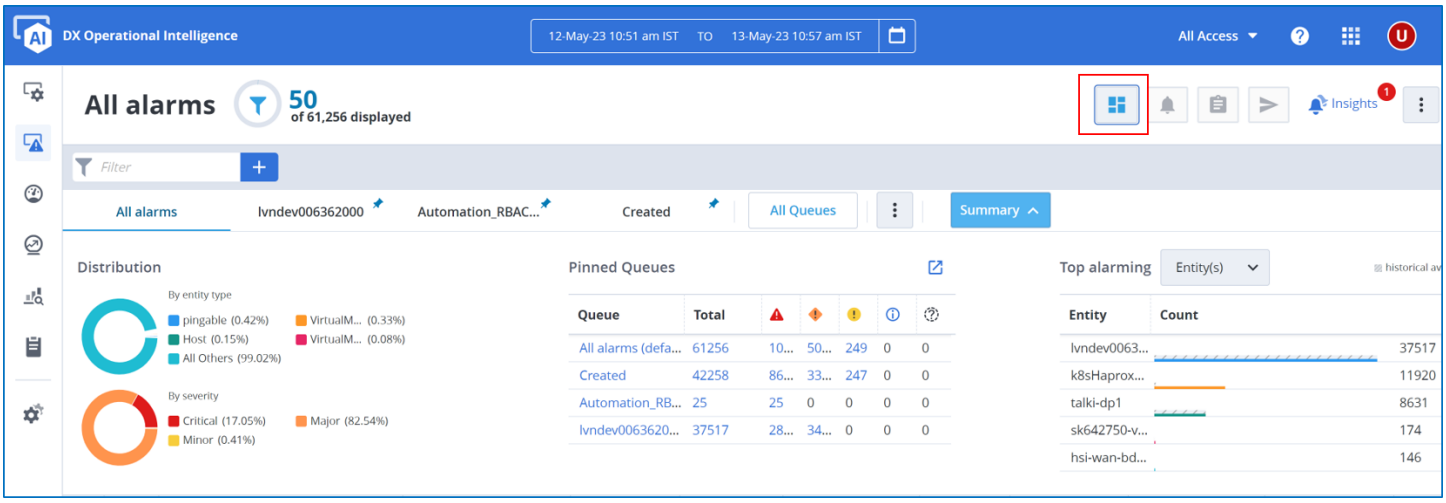
Starting with this release, DX Dashboards includes the **OI Alarm Metrics Overview** dashboard out-of-the-box. This dashboard provides the following information for all alarms and situations at the tenant level, user level, and service level. This dashboard is available in the **Alarm Analytics** folder.



NOTE
For more information, see the **OI Alarm Metrics Overview** dashboard in DX Dashboards documentation.

Launch OI Alarm Metrics Overview Dashboard from Alarm Analytics

You can now launch the **OI Alarm Metrics Overview** dashboard from the **All Alarms** and **Situations** pages. Click the **DX Dashboards** icon to launch the dashboard in a new tab.

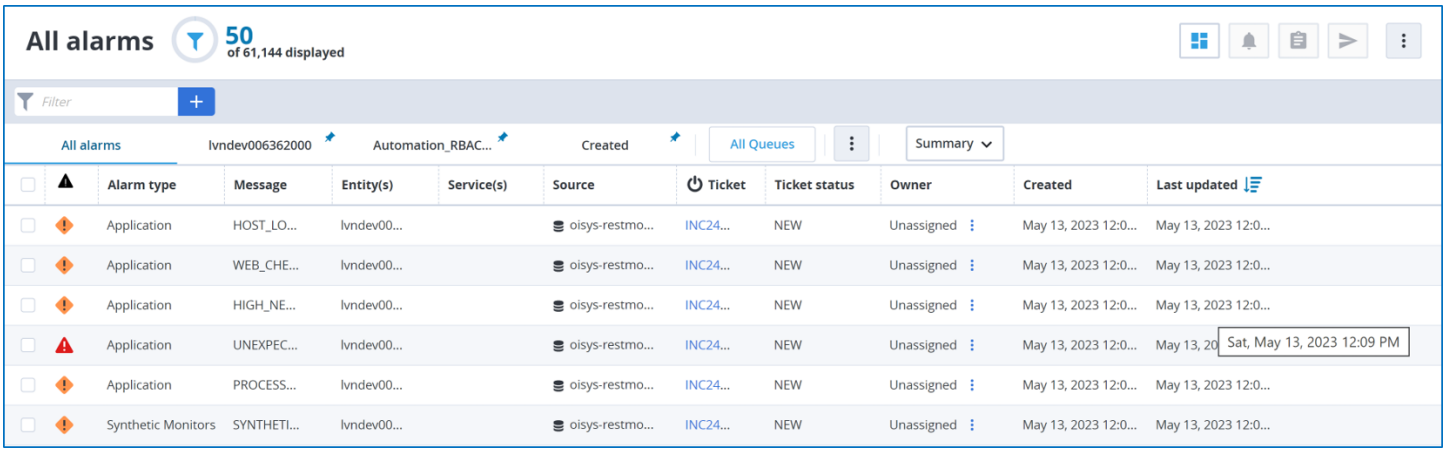


When you navigate to the dashboard, data is displayed for the time range that is selected on the **All Alarms** or **Situations** page and not the time range that is selected in DX Dashboards.

NOTE
For more information, see the **OI Alarm Metrics Overview** dashboard in the DX Dashboards documentation.

Last Updated Column Now Displays Timestamp

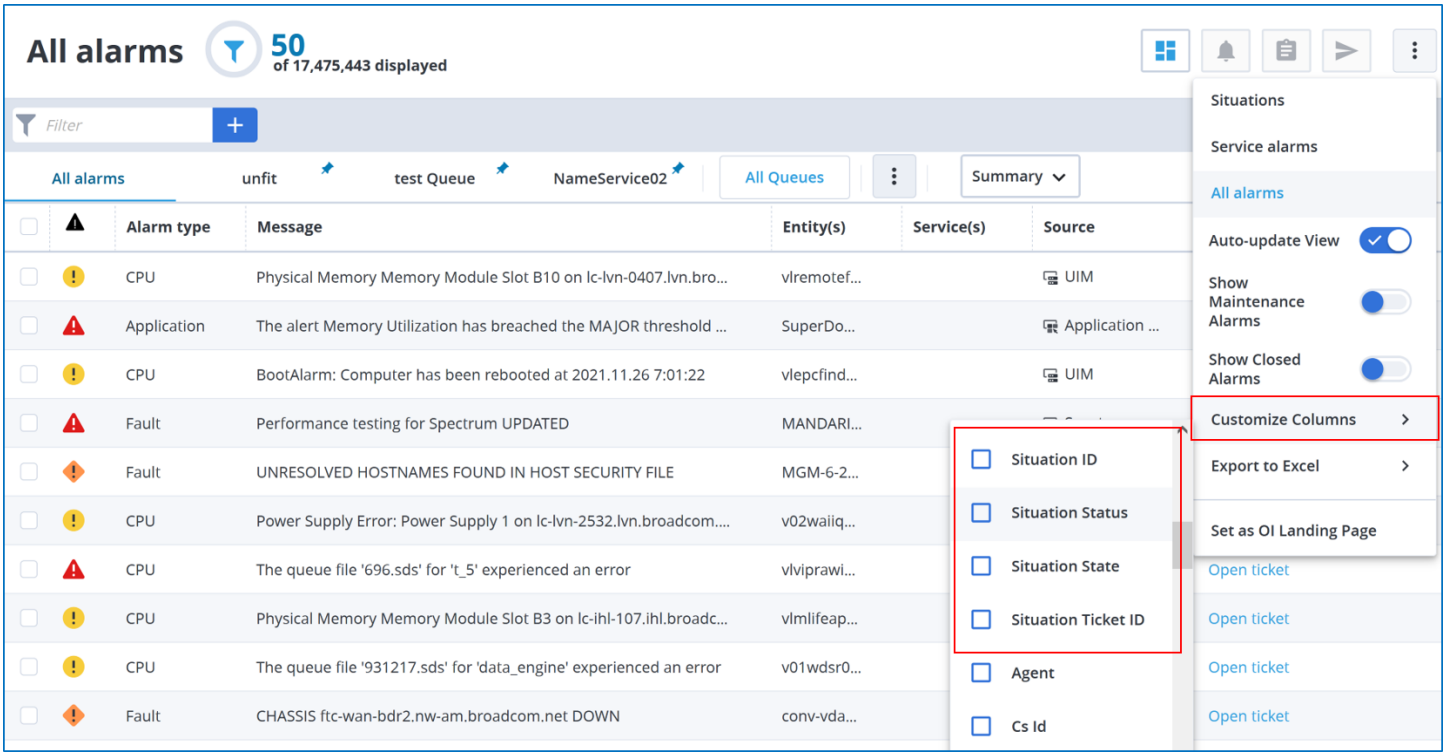
In the earlier version, the **Last Updated** column on the **Alarm Analytics** pages displayed as 1m, 2h 5m, and so on. Starting with this release, the **Last Updated** column displays the exact timestamp when the alarm was updated.



NOTE
For more information, see the All Alarms section.

Display Situation Details on All Alarms Page

Starting with this release, you can view the following situations details of raw alarms that are part of a situation on the **All Alarms** page.



- **Situation ID:** Click the ID to navigate to the corresponding situation details.
- **Situation Status:** Indicates the status of the situation. **Values:** New, Updated, or Closed.
- **Situation State:** Indicates if the situation is active or stable.
- **Situation Ticket ID:** Displays the ticket ID for the situation if present.

NOTE

- These columns are not displayed by default. Use the **Customize Columns** option in the **Alarms View** filter to select the required columns to be displayed.
- You can also filter by these attributes on the **All Alarms** page. For more information, see the All Alarms section.

Situation Ticket Details Include Link to First Alarm

When you click the Ticket ID of a situation, the ticket details now include the following details:

- **Earliest Source Alarm URL:** URL of the earliest source alarm that participated in the situation.
- **MostImpacted Source Alarm URL:** Of all the source alarms that participated in the situation, the URL of the source alarm that is impacting the situation the most.

NOTE
For these attributes to be included in the ticket details, ensure to add the following attributes to the message template. These attributes are available under **Add Variables > Situations**:

- **Earliest Source Alarm URL:${earliestSourceAlarmURL}**
- **MostImpacted Source Alarm URL:${mostImpactedSourceAlarmURL}**

Script Operator Support

The Script operator processes the upstream data in the NASSQL query, applies the Javascript functions, and generates new data. For example, the metric value for a metric named Average CPU Uptime is zero during the maintenance window. In the dashboard, the metric graph is displayed as zero during that period. Using this script function, you can check if the timestamp of the metric is in the maintenance window.

NOTE
For more information, see the Script Operator section in the DX Dashboards documentation.

Projection Filter Supports JSON Pointer Syntax

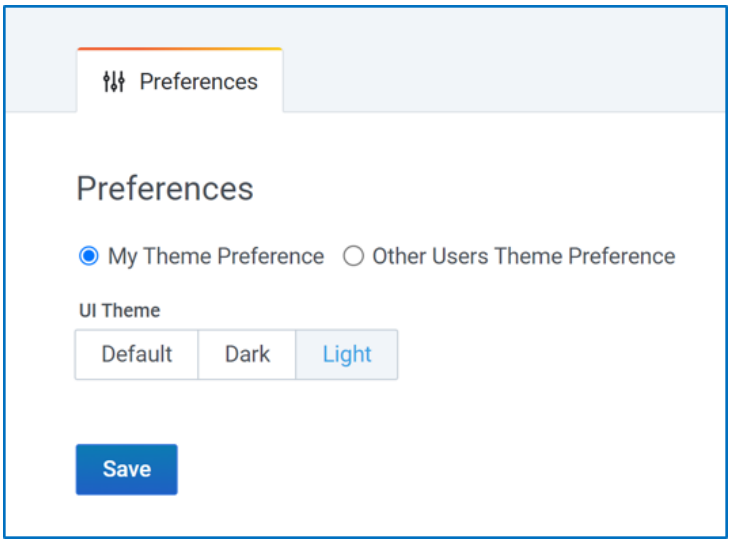
The Projection Filter that is used in TAS Query filters only the interested attributes. You can use this filter to get a specific value of an attribute value object as another attribute using JSON Pointer syntax in the results.

NOTE
For more information, see the Projection Filter section in the DX Dashboards documentation.

Theme Preferences Changes

In the existing version, if a Tenant Administrator changed the theme preference, the changes were applied to all the users in the tenant.

Starting with this release, DX Dashboards provides the following options to select the theme:



- **My Theme Preference:** A Tenant Administrator can select this option to set the theme preference for their account.
- **Other Users Theme Preference:** A Tenant Administrator can select this option to set the theme preference for all the users in the tenant. However, the user selection takes precedence.

  NOTE
  For more information, see the DX Dashboards documentation.

Known Issues

This release includes the following known issues:

- The existing alarms are not closed even after the metric group is deleted.
- When only DX APM is installed, a Tenant Administrator is unable to view or create metric groups or alarms for the metric groups.

2023.4.2 Release Notes

This release includes the following new features, enhancements, and fixed issues:

- New ITSM Channel - BMC Helix
- Configure Service Health Status
- Override Default Alarm Severity in Service Health Calculation
- Calculate Service Health Based on SLI
- Policy Filter Fixes for Notifications
- Set Expiration Date for Tokens
- Retrieve Maintenance Window Schedule for Service Using API
- Retrieve Licensed SKUs Using API
- Display User Name Instead of Tenant ID
- Embed DX Dashboards in Other Applications
- Known Issues

New ITSM Channel - BMC Helix

Starting with this release, you can integrate DX Operational Intelligence with BMC Helix for the incident management. This integration enables you to synchronize information across DX Operational Intelligence and BMC Helix. Any changes made to the alarms in DX Operational Intelligence are reflected in BMC Helix and vice versa.

Also, this integration supports:

- Automatic and Manual Ticket Creation
- Annotations for Alarms
- Clear Alarms and Close Tickets
- Associate the BMC Helix channel with mapping rules to enrich the ticket
- Launch BMC Helix directly from the ticket in DX Operational Intelligence

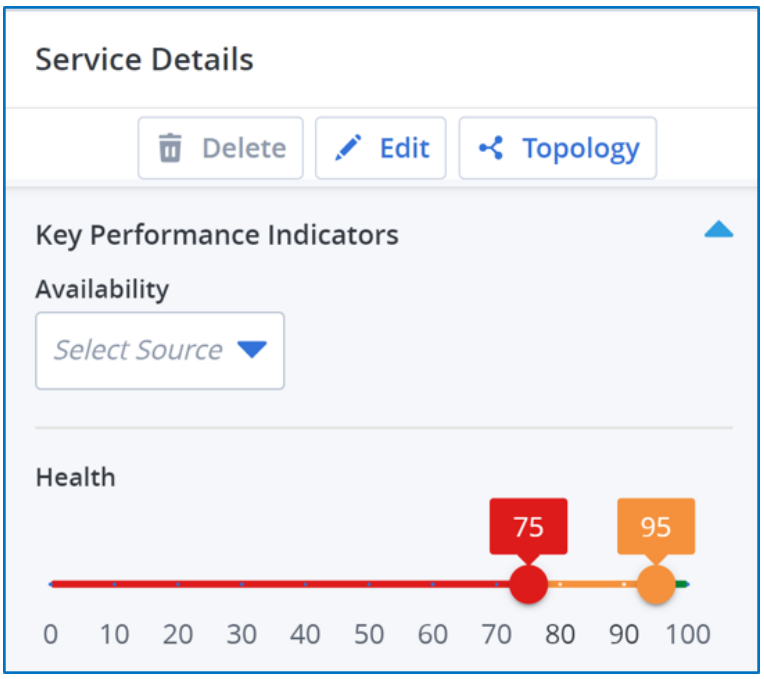
NOTE
For more information, see the Integration with BMC Helix section.

Configure Service Health Status

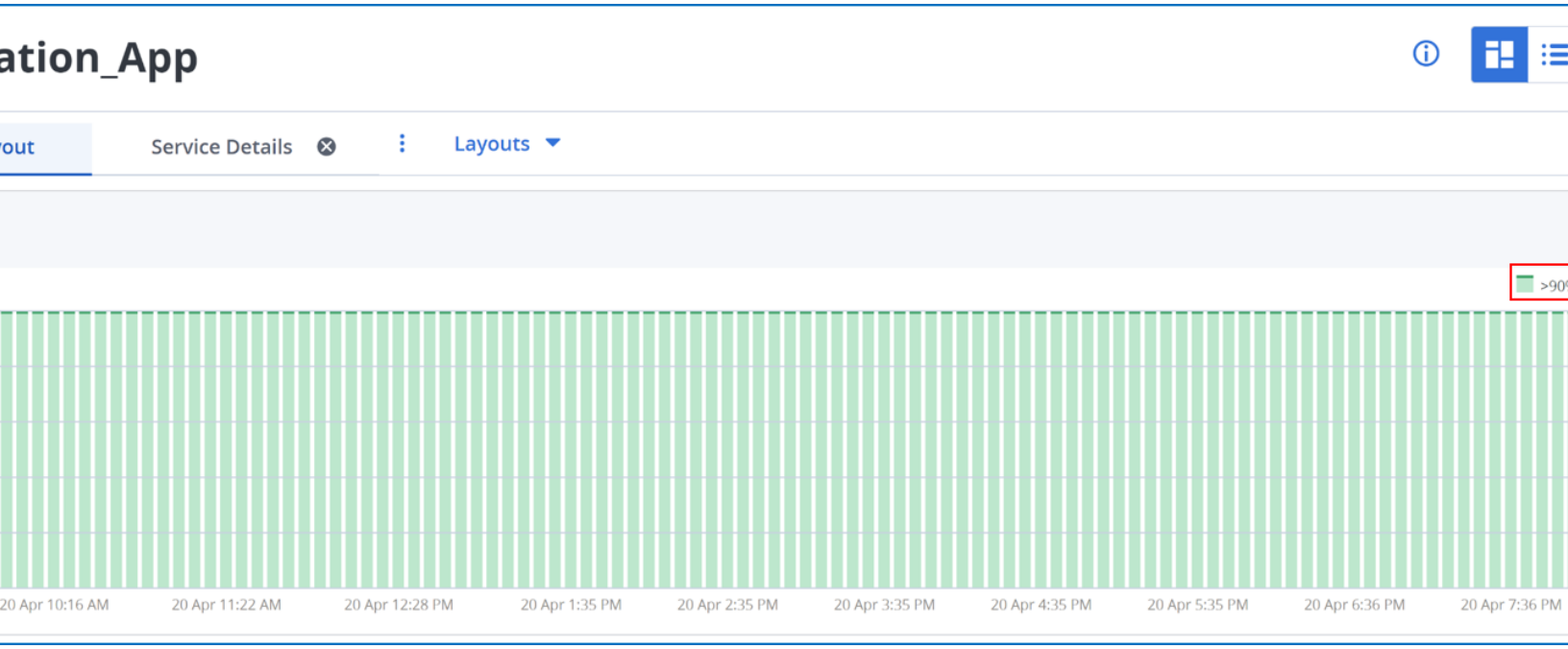
The Service Health indicates the percentage of devices that are running normally within the service. Prior to this release, the service health status on the **Services** page was based on the following predefined threshold values and the **Health** widget on the **Service Details** page reflected the health based on these values:

Health Category	Service Health
GOOD	95% - 100%
AVERAGE	> = 75% - < = 95%
BAD	0% - 75%

Starting with this release, you can reconfigure these values or ranges using the slider in the **Service Details > Health** section while creating or editing a service. The following image illustrates the default threshold values.



Now if you change the threshold values on the slider to 80 and 90, the **Health** widget displays the service health based on the new values as shown:



According to the reconfigured values, the service health is:

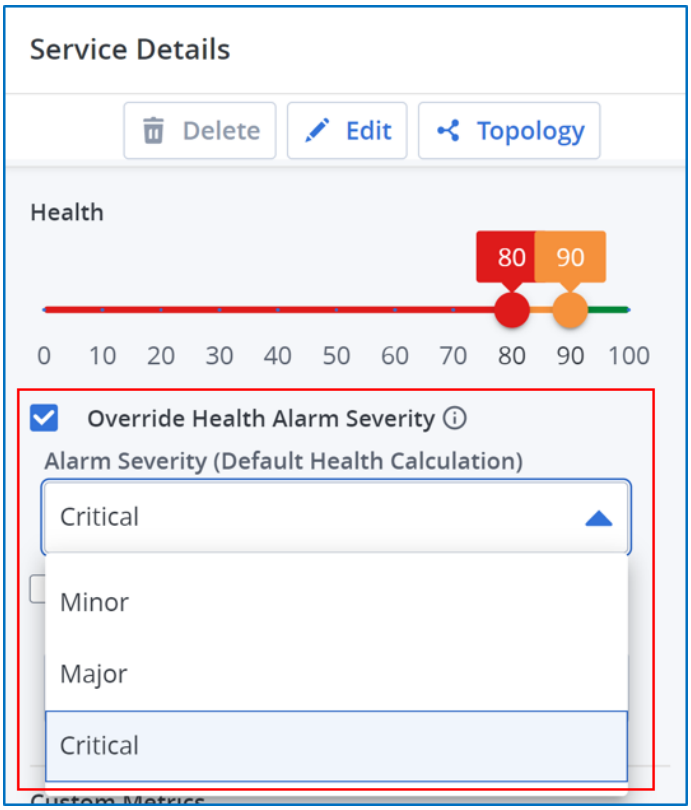
- Good if the percentage of devices that are running is 90% - 100%.
- Average if the percentage of devices that are running is > 80% but <90%.
- Bad if the percentage of devices that are running is 0% - 80%.

NOTE
For more information, see the Configure Service Details Properties section.

Override Default Alarm Severity in Service Health Calculation

The service health is calculated based on the devices or entities that are down. A device or entity is counted as down only if the critical alarms are triggered against it. The major and minor alarms are not considered in this calculation. For example, if a service has 10 devices with only three devices having the critical alarms, the service health is calculated based on that devices that are down. If the devices in the service do not have critical alarms, then the devices are counted as up and the service health is displayed as 100%.

Starting with this release, you can configure the service health of the existing and new services based on all the three alarms, critical, major, and minor.



To configure the service health based on major and minor alarms, select the **Override Health Alarm Severity** checkbox in the **Service Details** section and select the severity from the **Alarm Severity** drop-down list. Selecting this checkbox overrides the default service health calculation which is based on critical alarms only.

If you select:

- **Critical:** Only Critical alarms are considered for the service health calculation.
- **Major:** Major and Critical alarms are considered for the service health calculation.
- **Minor:** Minor, Major, and Critical alarms are considered for the service health calculation.

After this configuration, the Health widget on the Service Details page reflects the health based on the configuration.

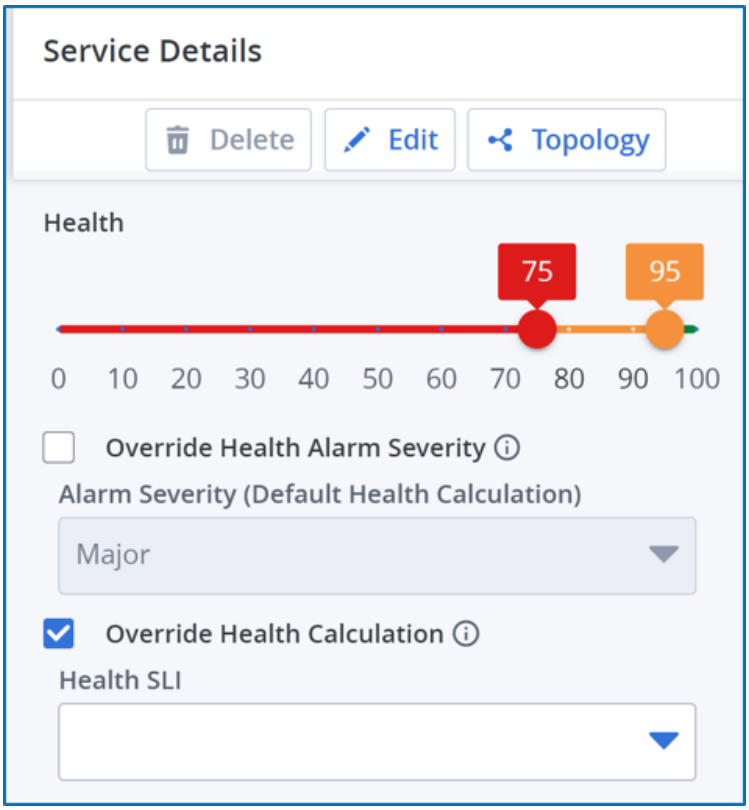
NOTE

- Only critical alarms are considered for the service health calculation if you do not select this checkbox.
- This configuration is specific to the service. If you select this option for a parent service, then this configuration is used in the service health calculation only for the parent service and not for the child services. The service health for the child services is calculated based on the default condition, which is the critical alarms only.
- If the health SLI is selected, the default health metric is calculated with this alarm severity. For more information, see the **Calculate Service Health Based on SLI** section.
- For more information, see the Configure Service Details Properties section.

Calculate Service Health Based on SLI

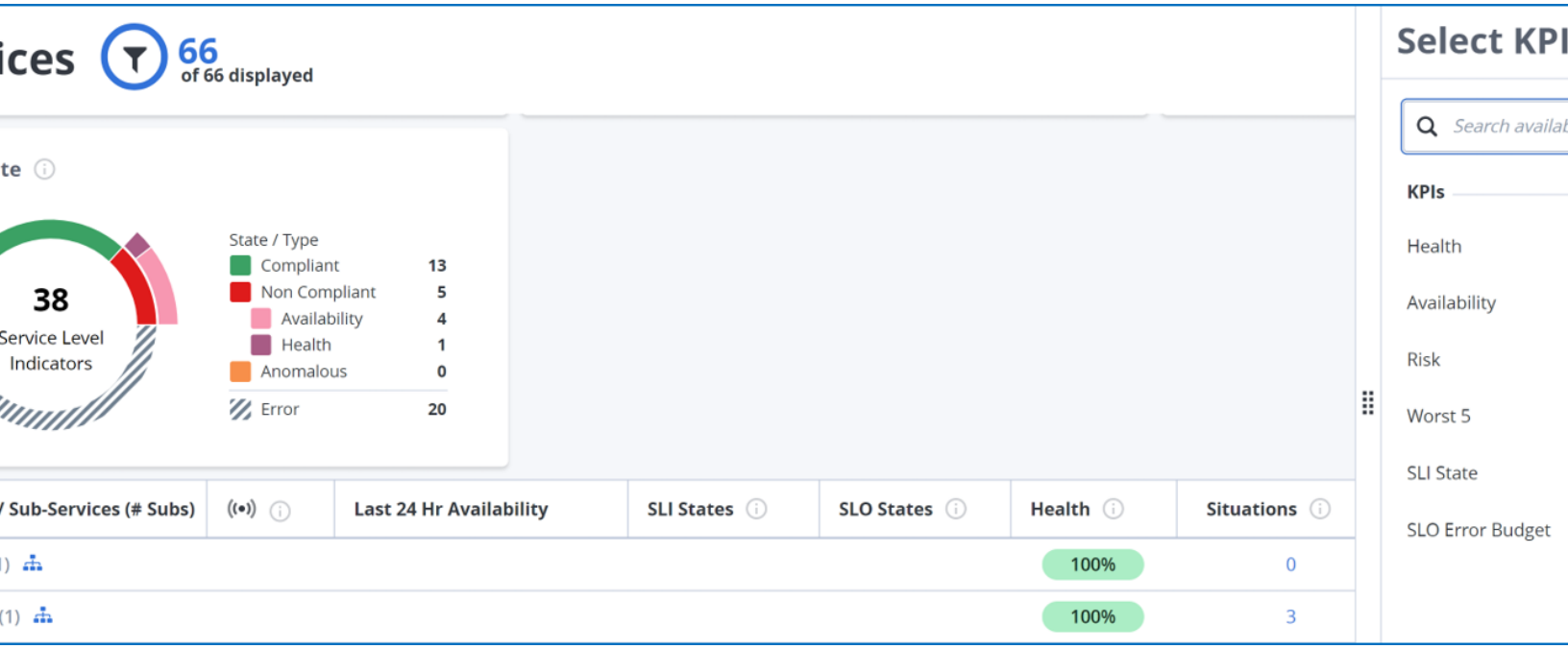
The service health is calculated based on the number of available devices and impacted devices. To calculate the service health based on metrics, you can create an SLI using the metrics. The SLI creation section now provides **Health** as the SLI Type to configure the health.

After you create the SLI, you can select the **Override Health Calculation** checkbox and select the **Health SLI** in the **Service Details** section.



This selection overrides the metric that is used to calculate the service health with the Health SLI. The **Health** widget on the **Service Details** page displays the service health based on the Health SLI.

> **NOTE**
> If the **Health SLI** is created, the **SLI State** widget on the **Services** page displays the **Health**. Click Health to view all the services that have Health SLI configured.



NOTE
For more information, see the Configure Service Details Properties section.

Policy Filter Fixes for Notifications

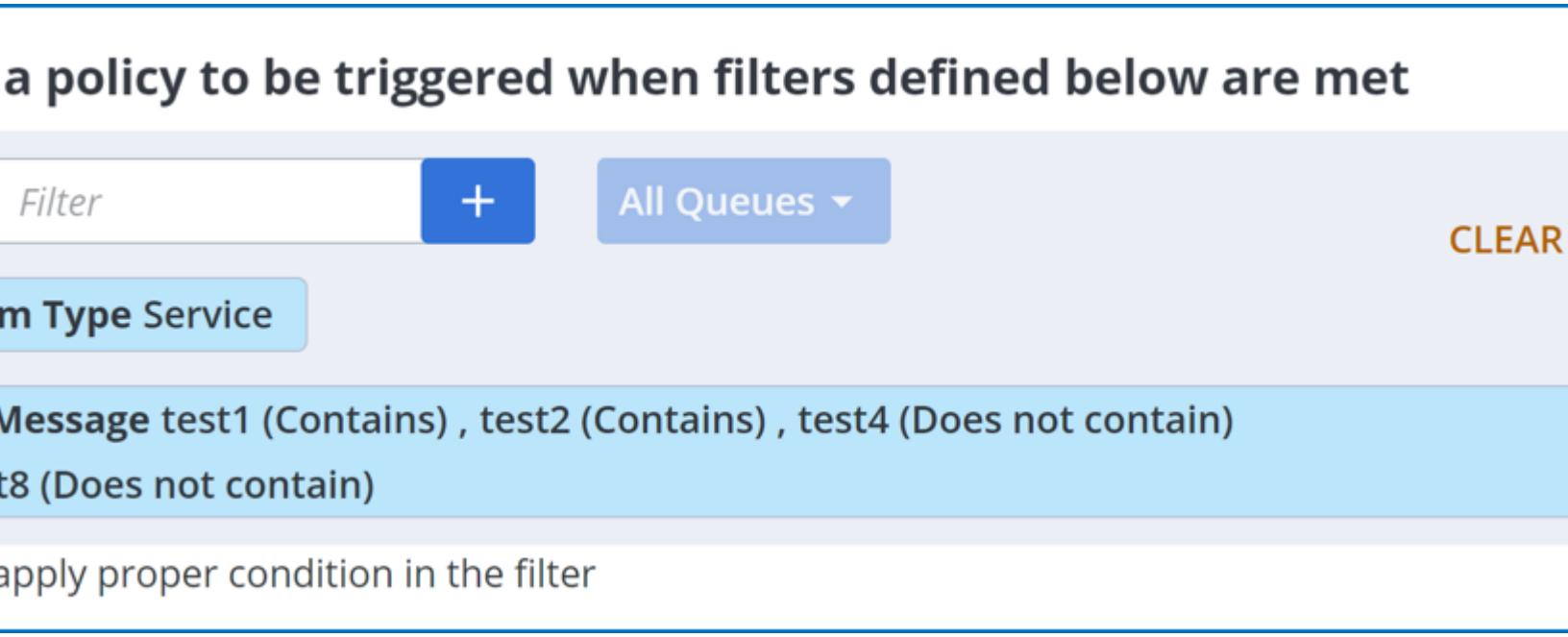
The policy filters on the **Settings > Create Policy** page and the **All Alarms** page behaved differently. Changes that were made to make the filters in line with the current alarm queue and filtering capabilities in DX Operational Intelligence are impacting the way policy filters work.

- Use Case 1: Positive and Negative Operators in the Filter
- Use Case 2: Services in the Hierarchy

Use Case 1: Positive and Negative Operators in the Filter

Prior behavior:

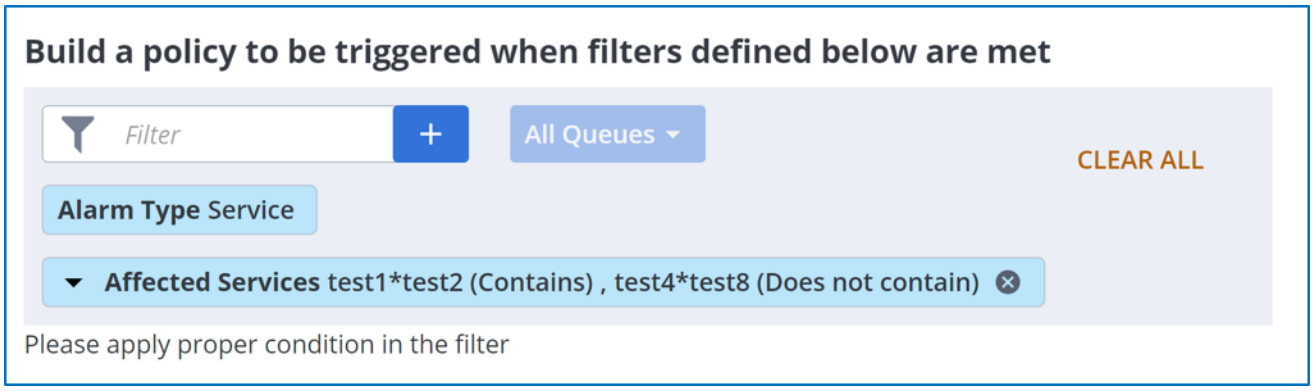
- If the policy filter contained only the **positive operators** such as **Equals, Contains, Starts With,** and so on, then **OR** was used to filter the attribute values.
- If the policy filter contained only the **negative operators** such as **Not Equals, Does Not Contain, Does Not Start With,** and so on, then **AND** was used to filter the attribute values.
- If the policy filter contained a combination of **positive** and **negative** operators, then **AND** was used to filter the attribute values for both positive and negative operators.
  For example, the following policy filter has two positive operators and two negative operators.



A notification was sent only if the alarm message met this condition: **((contains test1 AND test2) AND (does not contain test4 AND test8))**. That is, the message must contain test1 and test2 and must also not contain test4 and test8.

After the changes, if the policy filter has a combination of positive and negative operators, the positive and negative operators are grouped separately with **OR** and **AND** respectively and then evaluated with **AND**. For example, a notification is sent only if the alarm message meets this condition: **((contains test1 OR test2) AND (does not contain test4 AND test8))**. That is, the message must contain test1 or test 2 and must also not contain test4 and test 8.

Workaround: This implementation has an impact on the existing policy filters. To ensure that they work correctly, update the policy filter if you want in the following format. Enter the values of the message attribute in the same order as it is in the alarm field value as shown in this example.



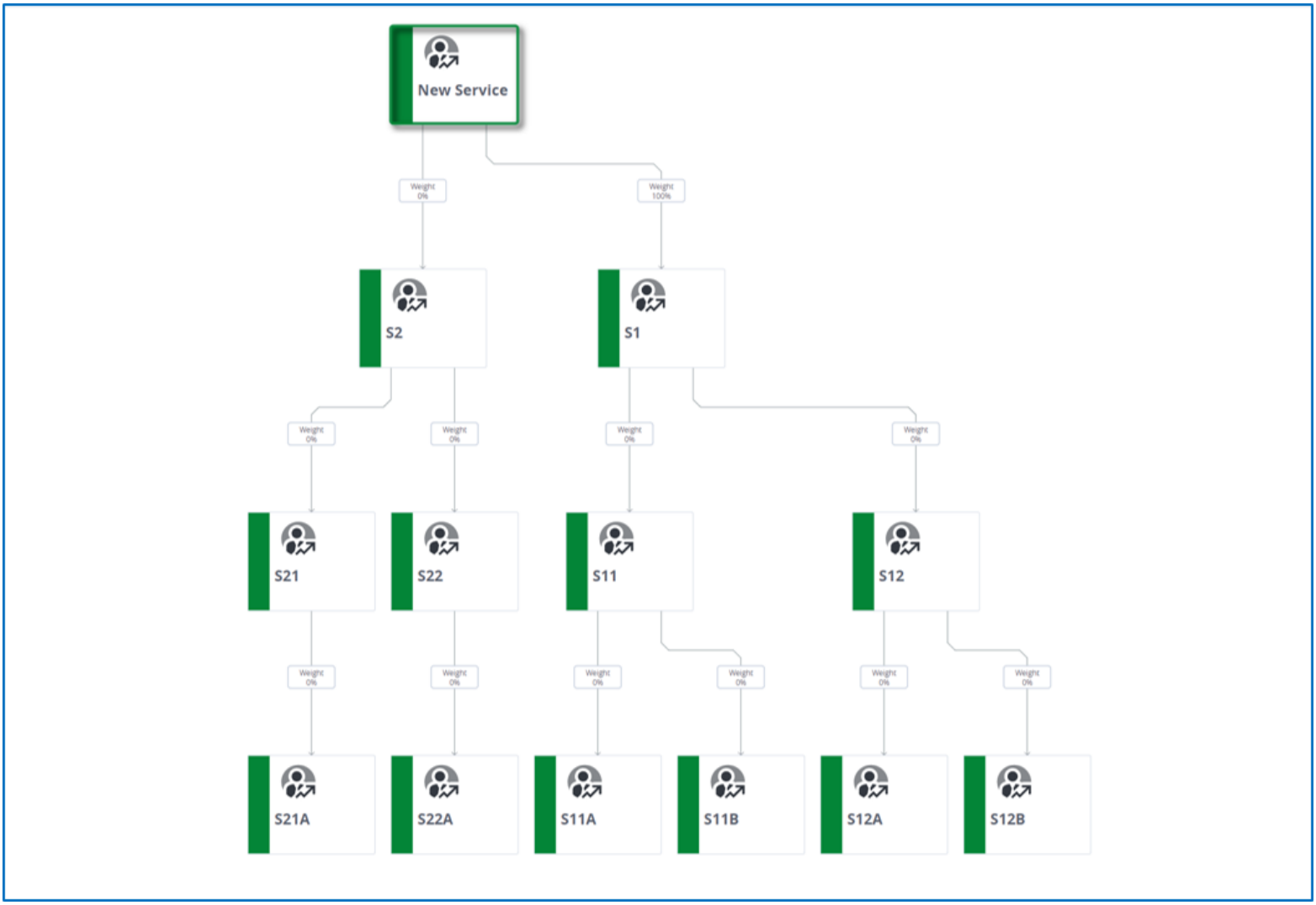
Use Case 2: Services in the Hierarchy

Prior to the changes, when you created a policy for a service from the alarm queue on the **All Alarms** page, the policy filter considered all the services in the hierarchy. However, on the **Settings > Create Policy** page, the filter considered only the service that was selected in the filter and not the services in the hierarchy.

After the changes, when you create a policy for a service from the **Settings > Create Policy** page, the filter considers even the child services in the hierarchy similar to the alarm queue in the **All Alarms** page to support notifications.

For example,

- New Service is a parent service.
- S1 and S2 are the child services.
- S11, S12, are child services for S1, and so on.

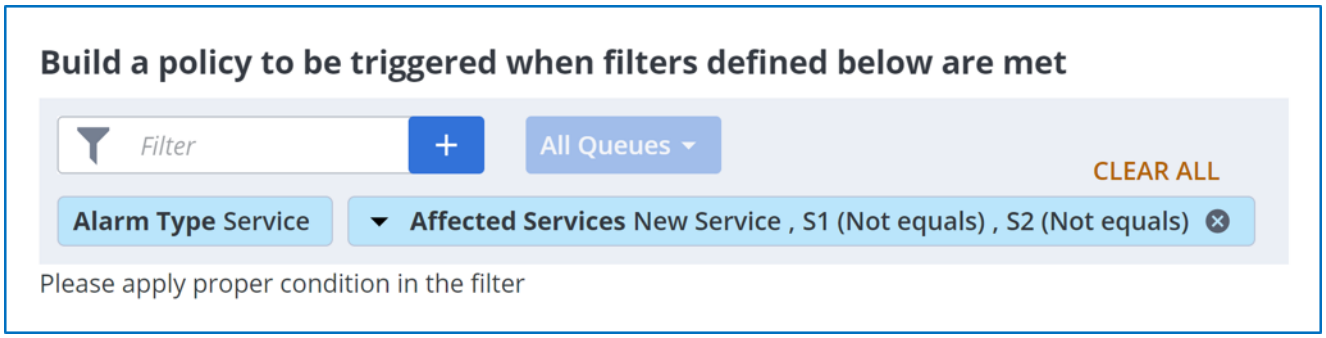


When you create a policy for New Service (parent service), a notification is generated even if the alarm is raised on any of the child services (for example, S2, S11, S11B) in the hierarchy.

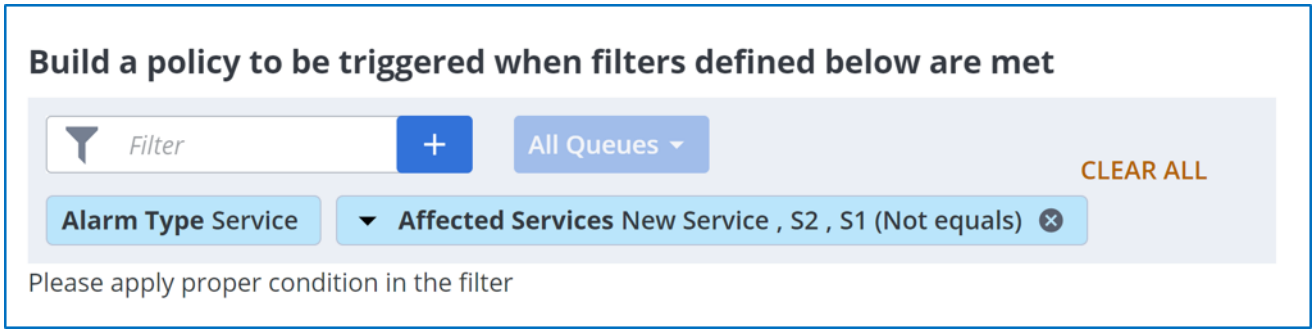
Workaround: If you do not want the child services to be considered by the policy that is created on the **Settings > Create Policy** page, update the policy filter using the **Affected Services** attribute with **Equals** and **Not Equals** operators.

> **NOTE**
> Only the **Equals** and **Not Equals** operators support service hierarchy.

For example, to consider only the parent service, enter **Affected Services EQUALS New Service**, **Affected Services NOT EQUALS S1**, and **Affected Services NOT EQUALS S2** as shown. A notification is generated only when the alarm is raised on New Service.



Similarly, in the following example, a notification is generated when the alarm is raised on New Service, S2, or any of the services in the S2 hierarchy.



Set Expiration Date for Tokens

Starting with this release, you can set the expiry date and time for the tokens that you create. After the token expires, the status of the token is changed to **Expired**. When you create a token, the **Generate** button is enabled only if the expiry date is valid.

- The local time is selected for the tokens.
- You cannot change the expiration date for the existing tokens.
- You cannot revoke or un-revoke an expired token.

> **NOTE**
> For more information, see the Token Management section.

Retrieve Maintenance Window Schedule for Service Using API

Starting with this release, you can use get the list of all the maintenance window schedules for a particular service in the last one year using an API.

> **NOTE**
> For more information, see the Retrieve Maintenance Window Schedule for Service section.

Retrieve Licensed SKUs Using API

You can retrieve the licensed SKUs using the following APIs:

- **Trigger recalculation of plaTenantsXmlfile content:** Use this API to recalculate the PLA tenants.
- **Download calculated plaTenantsXmlfile content:** Use this API to download the XML file that has the calculated PLA Tenants.

NOTE
For more information, see the Retrieve Licensed SKUs section.

Display User Name Instead of Tenant ID

The **Profile** section of DX Operational Intelligence now displays the user name instead of the tenant ID.

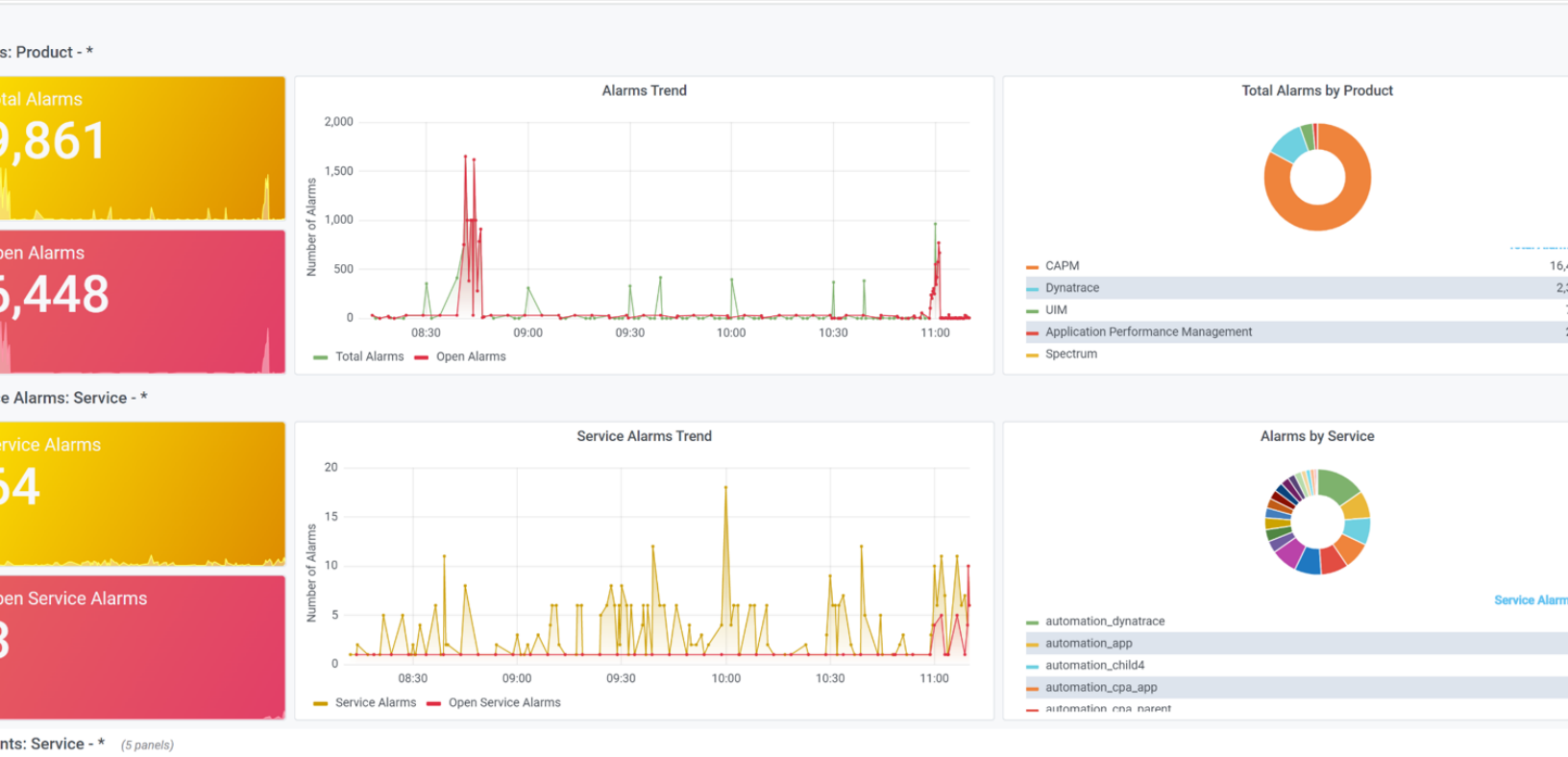
Embed DX Dashboards in Other Applications

You can now embed any DX Dashboard in other applications using the **embedded=true** parameter in the dashboard URL. For example, to embed the **Alarm/Incident Monitoring** dashboard in your application, open the dashboard, copy the dashboard URL, and append the parameter to the URL as shown:

```
https://dxi-dashboard.dxi-na1.saas.broadcom.com/d/jMfIrFBZkSM/alarm-incident-monitoring?orgId=2&embedded=true
```

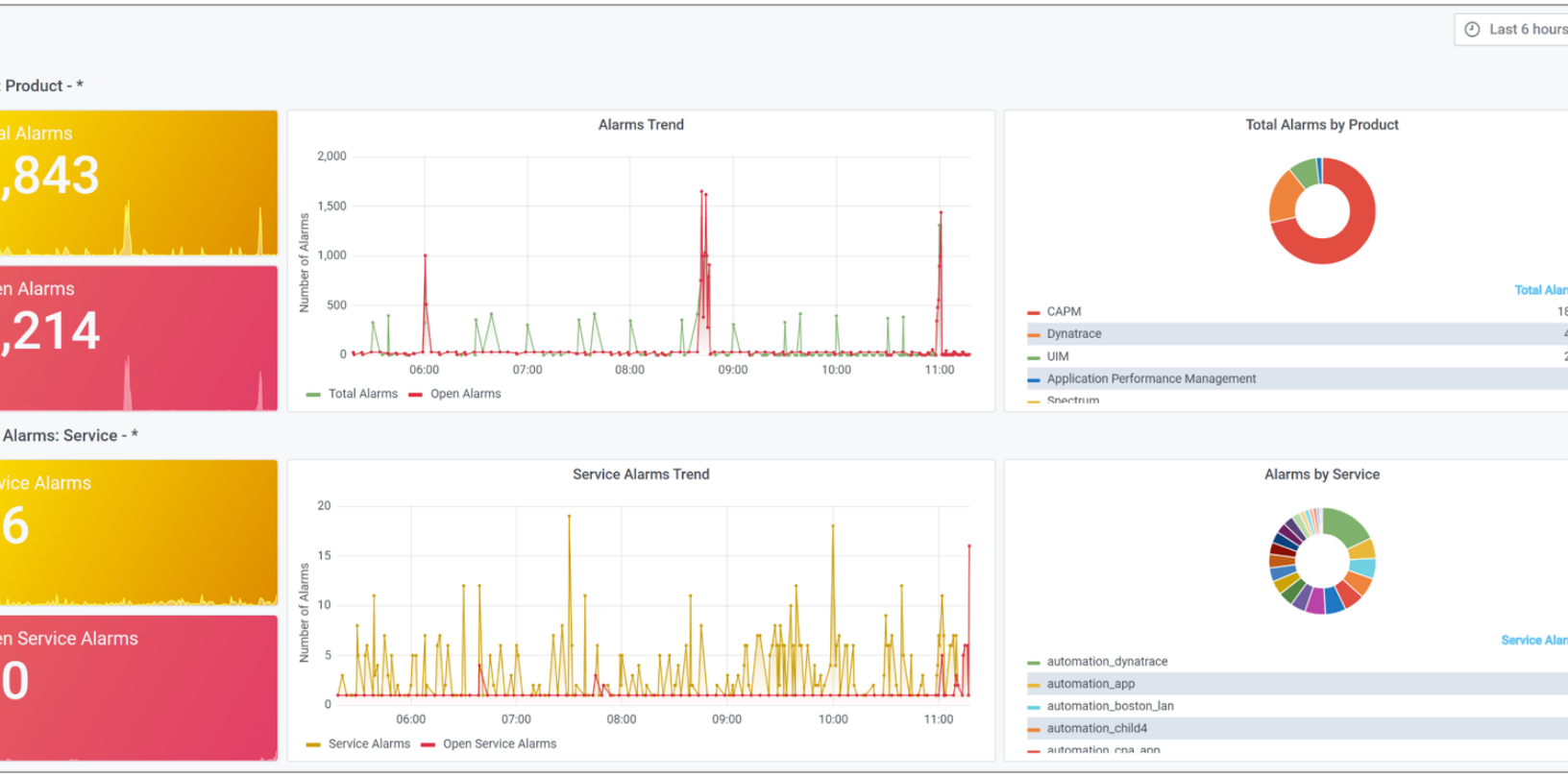
You can use this link in your application.

When you add the **embedded=true** parameter, only the visualizations are included in the dashboard but not the dashboard name, filters, settings, or the left navigation pane.



To include the time range in the dashboard, use the **enabledaterange=true** parameter as shown in the following example:

```
https://dxi-dashboard.dxi-na1.saas.broadcom.com/d/jMfIrFBZkSM/alarm-incident-monitoring?
orgId=2&embedded=true&enabledaterange=true
```



Known Issues

This release includes the following known issues:

- If you select the SLI Type as Health, then the SLI and the health metric are capped in the range between 0 to 100.
- If the service health is overridden by SLI/SLO, then the overridden service health values are displayed only on the Services pages and the Impacted Services tab of alarms. However, the Performance Analytics page, custom metrics on the Services page, and Service dashboards display the service health that is calculated from alarms.
- When a child service is overridden by the SLI health, the SLI health from the child service is not considered for the calculation nor the SLI health is not rolled up to parent service health.

  For example, a parent service S1 has two child services S2 and S3 and each of these services has a contribution of 50% weightage to the parent service. If the service health of S2 and S3 is 100% and 100% respectively, then the contribution of these services to the parent service is 100%.

  If the S2 health (40%) is overridden by SLI1 and S3 health (60%) is overridden by SLI2, then the parent health service is 100% (from the default alarm calculation) and the service health is not rolled up from S2 and S3.
- The child service displays two places after the decimal (for example, 99.89) while the parent service displays only one place (for example, 99.9) though the parent has 100% weightage from the child service.
- The widgets on the Service Details page that are not displayed by default are not retained on calendar refresh.
- The **Override Health Calculation** checkbox is not disabled even if the user does not have access to SLI.
- If the SLI is deleted or the SLI metric is not available, a warning message is displayed on the Service Details panel.
- If an SLI is attached to the service health and a user changes the SLI aggregation type, then the service health calculation will be stopped because of new metric creation for the same SLI. Also, the **Health SLI** dropdown in the **Override Health Calculation** section displays two SLIs for some time.

2023.4.1 Release Notes

This release includes the following new features, enhancements, and fixed issues:

- View Metrics in Context of Selected Entities from Monitored Inventory
- Enable Time Range on DX Dashboards Reports
- Enable/Disable DX Dashboards Reports From Index and More
- New Out-of-the-box DX Dashboards
- Embedded DX Dashboards Changes

View Metrics in Context of Selected Entities from Monitored Inventory

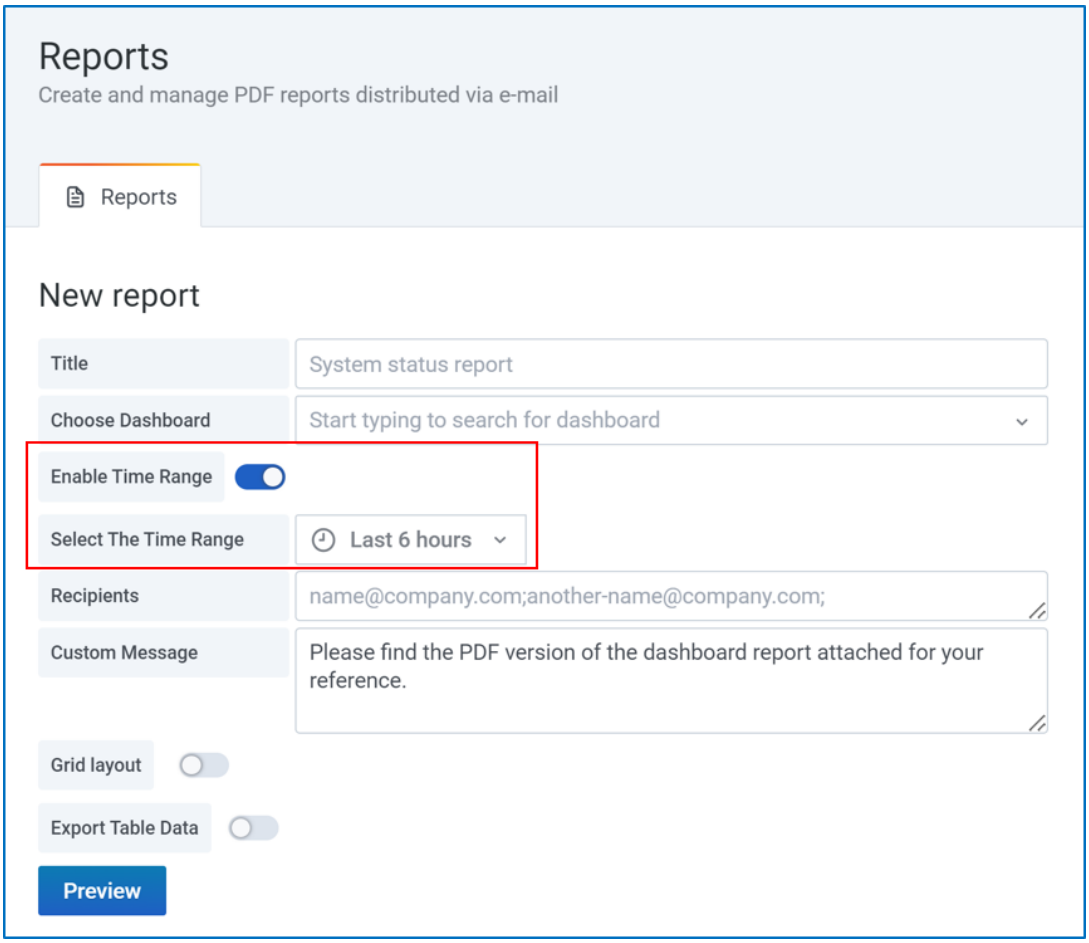
You can now triage performance issues directly within the **Monitored Inventory** page and in context. Understand issues better and solve problems faster by having performance analytics tooling in context to the user selection. Click the **Performance Analytics** icon under the **Type** column to launch the panel.

NOTE

- On the **Performance Analytics** panel, data is displayed only for **6 Hour**, **12 Hours**, or **1 Day** depending on your selection. You cannot select any other time period.
- You can resize the Performance Analytics panel.
- The Monitored Inventory page also displays the total count of entities and also the count of displayed entities above the widgets.
- For more information, see the Monitored Inventory section.

Enable Time Range on DX Dashboards Reports

In the existing version, reports are generated with data for the time range that is set in the dashboard. Starting with this release, you can specify the time range for the data to be captured in the report using the Enable Time Range option. Enable this option and select the time range as required. By default, Last 6 hours is selected.

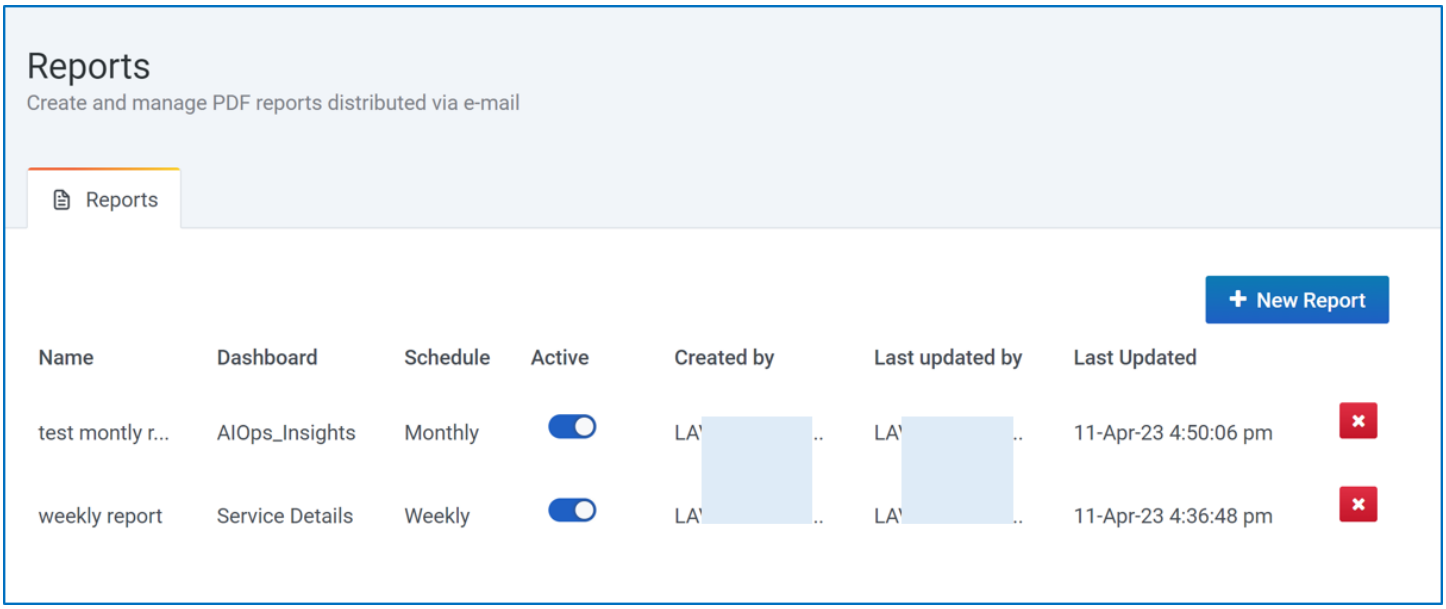


The **Enable Time Range** option is disabled and the time range that is set in the dashboard is considered by default. But if the time range is enabled for a report, the time range that is selected in the report overrides the time range set in the dashboard.

> **NOTE**
> For more information, see the Reports section in DX Dashboards documentation.

Enable/Disable DX Dashboards Reports From Index and More

Starting with this release, the Reports page now provides the following additional information about the reports:



- **Active:** Indicates if the report is active or not. You can enable or disable the report generation on this page.
- **Last Update By:** Displays the name of the user who last updated the report.
- **Last Updated:** Displays the timestamp of the last update to the report.

In the earlier version, information only about the name, dashboard, schedule, and created by was displayed.

> **NOTE**
> For more information, see the Reports section in DX Dashboards documentation.

New Out-of-the-box DX Dashboards

The following dashboards are now available out of the box in the **APM-MetricView** folder:

- APM: Blamepoint Overview
- APM: Frontend Overview

Both these dashboards contain the standard set of metrics to provide initial direction for triagers to identify the system experts who can assist with a problem.

DX APM reports these metrics wherever the Java methods are monitored. For example:

- Frontends
- Backends

> **NOTE**
> For more information, see the APM-MetricView Dashboards section in DX Dashboards documentation.

Embedded DX Dashboards Changes

Starting with this release,

- Annotations are not supported for embedded dashboards. You cannot add or view annotations.
  Reason: Data that is displayed in the dashboards varies depending on the source (agents or products), metrics, and so on. When you open an embedded dashboard in another application such as DX APM or DX Operational Intelligence, data that is displayed in the embedded dashboard may not be the same for which the annotations were added in DX Dashboards.
- The CSV button is no longer displayed in the embedded dashboards.

NOTE
For more information, see the Embed DX Dashboards section in DX Dashboards documentation.

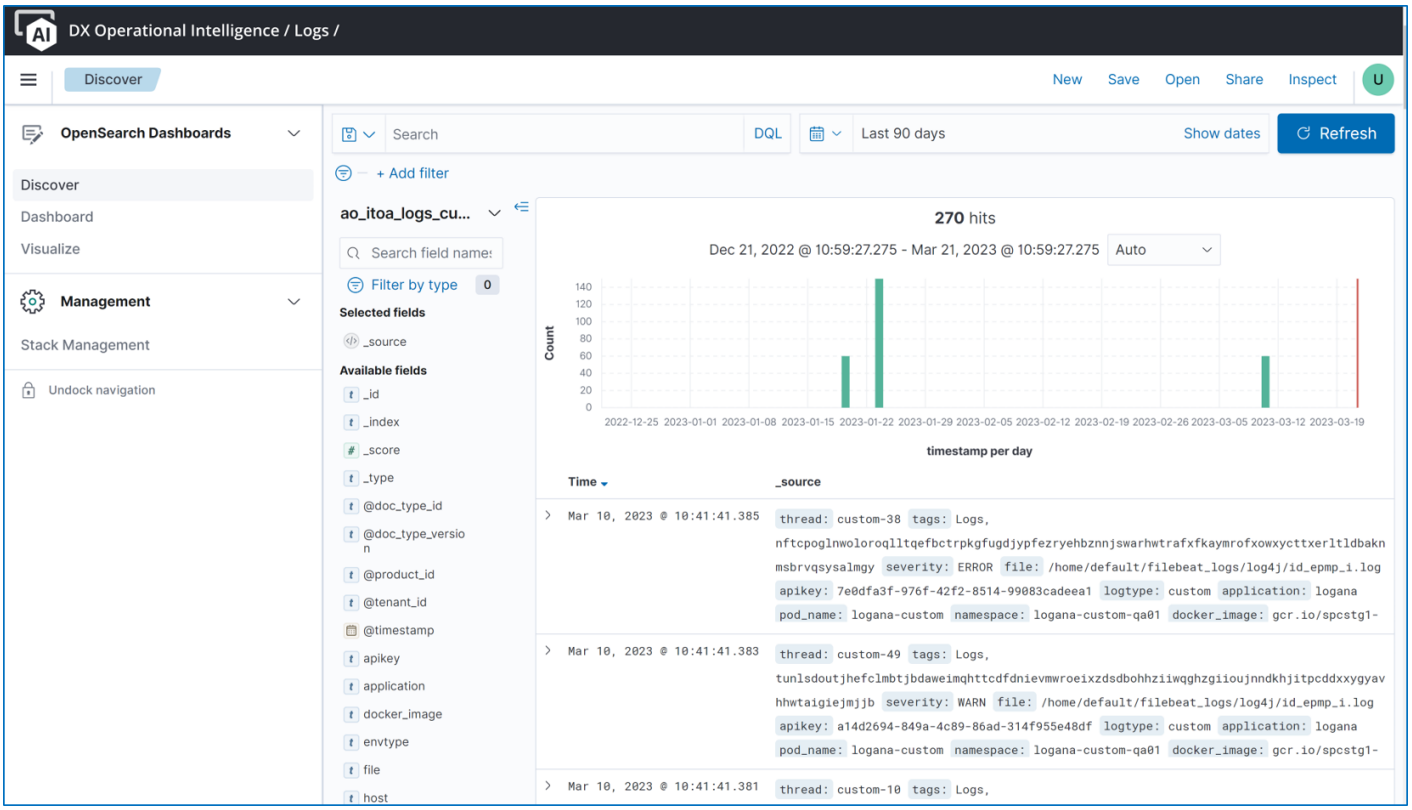
2023.3.2 Release Notes

This release includes the following new features, enhancements, and fixed issues:

- General Availability DX OI - Logs
- Launch Alarm Contextual Dashboards
- Launch CI Contextual Dashboards
- Reopen ITSM Ticket With Time Criteria

General Availability of DX OI - Logs

DX Operation Intelligence - Logs, is a purpose-built log management and monitoring capability that collects log data that is generated by your applications and IT infrastructure to gain operational insights. DX OI - Logs provides a single pane of glass to perform analysis, visualization, and get insight into the logs across application workloads, infrastructure, and network.



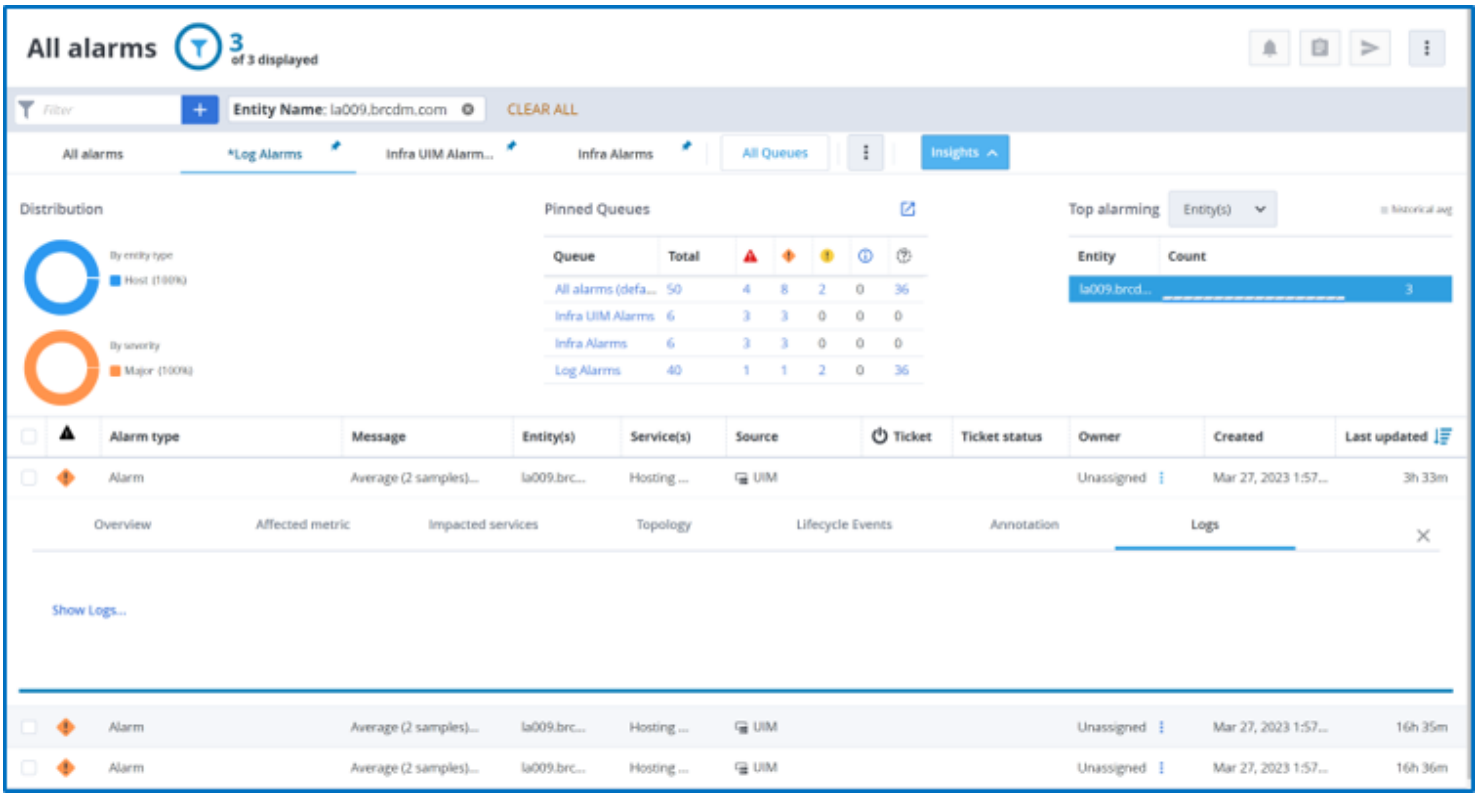
Key Features are:

- **Log Collection:** Route logs from any source to DX OI SaaS using one or more channels for ingestion
- **Log-based Alerting:** Specify rules to generate alerts from logs
- **Dashboards and Visualization:** Create custom dashboards to visualize logs and understand trends
- **Logs in the context of Alarms and CIs:** View syslog in context of Alarms and CIs

NOTE
For more information, see the Logs section.

Launch Alarm Contextual Dashboards

You can now view the logs for alarms from the **All Alarms** page. When an alarm is raised in DX Operational Intelligence by any product for a particular host and the given host is also sending syslog to Log Analytics, the **Logs** tab is enabled.



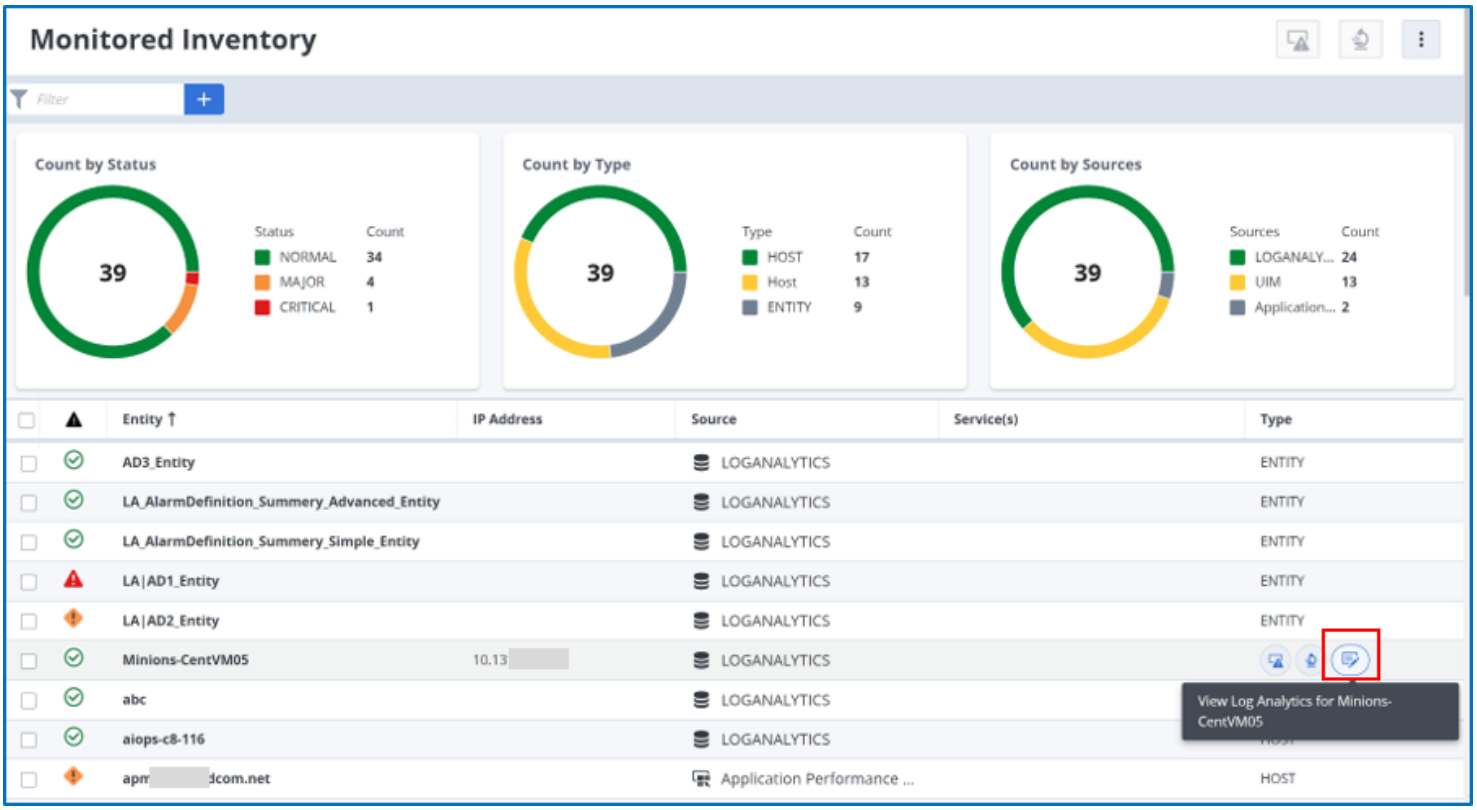
Click **Show Logs** to navigate to the **Log Analytics** page to visualize logs for the host. By default, logs for the last 15 minutes are displayed.

This tab is disabled if the log event is not enriched.

> **NOTE**
> For more information, see the All Alarms section.

Launch CI Contextual Dashboards

You can now view logs for an entity or CI from the **Monitored Inventory** page, when Log Analytics is one of the source products and the entity or CI is of the type **Host**. If Log Analytics is present for an entity, the **View Log Analytics** button is displayed under the **Type** column on the **Monitored Inventory** page. Hover over the entity for this button to be displayed.



When you click the **View Log Analytics** button, the Log Analytics page opens in a new tab providing the syslogs context for the selected entity. You can view logs for multiple inventory items.

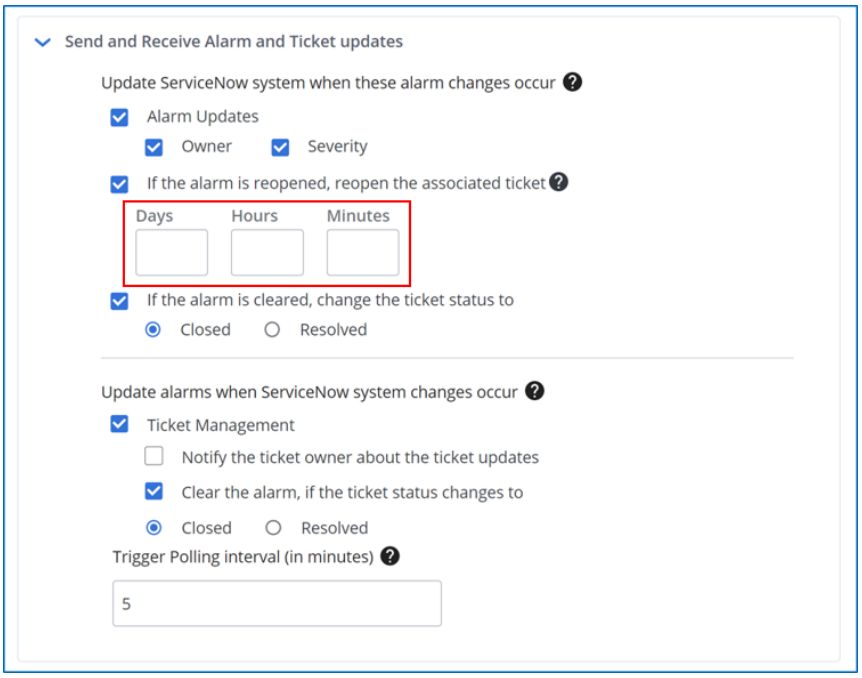
This button is unavailable if the entity is not of the type Host or Log Analytics is not in the Source Product list.

> **NOTE**
> For more information, see the Monitored Inventory section.

Reopen ITSM Ticket With Time Criteria

When an old alarm is reopened in DX Operational Intelligence, you can configure to reopen the associated ticket in the ticketing system. Starting this release, you can also configure to reopen only those tickets that were resolved in the last 30 days, which is the maximum allowed time.

The following image illustrates where you can configure this time criteria for ServiceNow:



Allowed Values for this configuration: Days: 0-30, Hours: 0-23, and Minutes: 0-59 respectively.

If you do not specify the time period, the duration is taken as 30 days by default.

> **NOTE**
> For more information, see the Automatic Synchronization of Alarm and Ticket Updates section.

2023.3.1 Release Notes

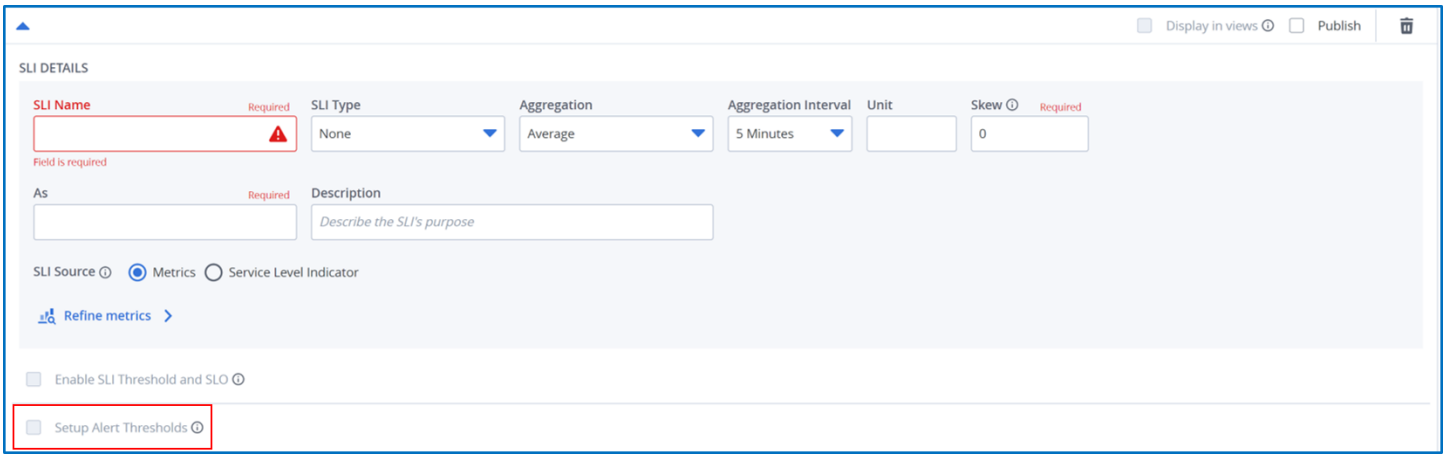
This release includes the following new features, enhancements, and fixed issues:

- Alerts for Service Level Indicators and Objectives
- Alerts for Connector Health
- Monitored Inventory Quick Filters
- Filter Monitored Inventory by Attribute
- Insights Enhancements
- DX Operational Intelligence SaaS to Automic Integration
- DX Dashboards
- Platform Improvements

Alerts for Service Level Indicators and Objectives

You can now monitor your key service performance indicators and their compliance through alerts. Use thresholds on the Service Level Indicator metrics to be alerted to the issue and correct the problem before it affects the Service Level Objective. Configure thresholds on the Service Level Objective metrics (compliance and error budget) to catch issues before the compliance or error budget is breached.

To add the alert, **set up alert thresholds** in the SLI configuration and enter the values for Threshold, Periods Over Threshold, and Observed Periods.



Known Limitation: An alert for a major alarm will not be generated for the same threshold value as a critical alarm even when the condition is met but the observed periods are different.

> **NOTE**
> For more information, see the Create SLIs and SLOs section.

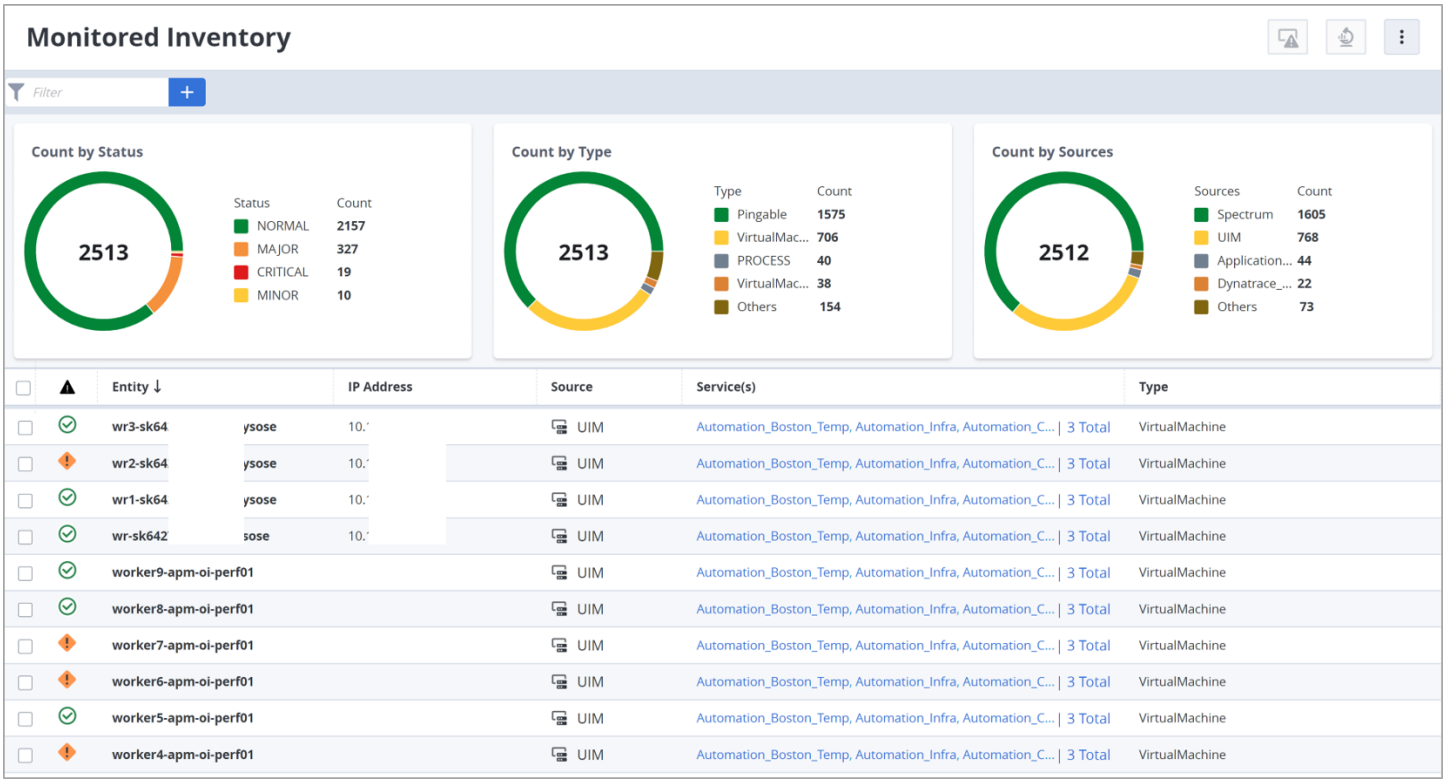
Alerts for Connector Health

DX Operational Intelligence now generates alarms related to the Connector health based on pre-configured rules. For example, when the Connector host has consumed 95% of the memory usage, DX Operational Intelligence generates a Critical alarm which is displayed on the **All Alarms** page.

> **NOTE**
> For more information, see the Connector Health Monitoring section.

Monitored Inventory Quick Filters

Understand the health of your entities on the Monitored Inventory page at a glance and filter quickly to the problematic entity using the newly added status, type, and source widgets.



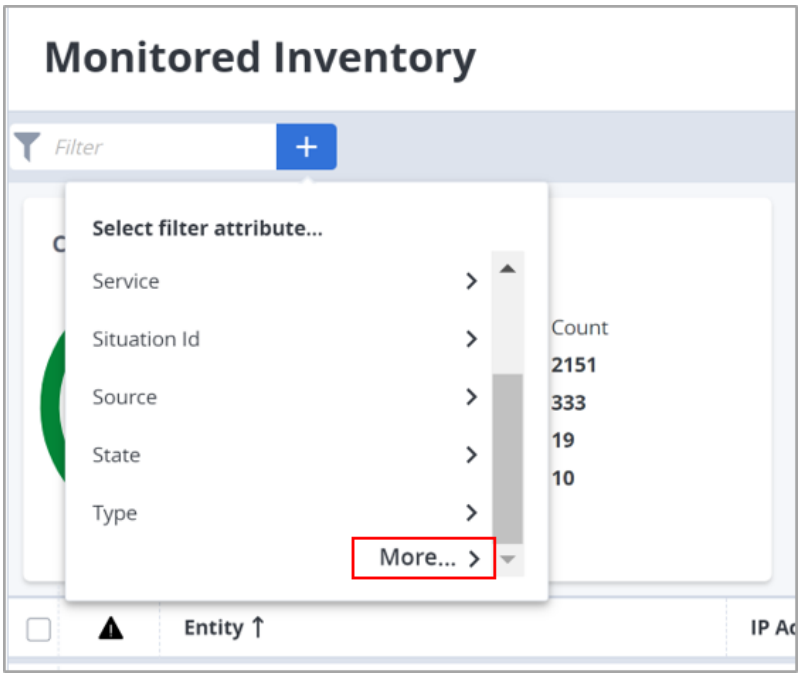
- **Count by Status:** Displays the count of the entities by status. **Status values:** Normal, Major, Critical, and Minor.
- **Count by Type:** Displays the count of entities by the entity type. The count is displayed for the top four types and the remaining entities are grouped under Others.
- **Count by Sources:** Displays the count of the entities by source. The count is displayed for the top four sources and the remaining entities are grouped under Others.

Click on the widget legend or the color on the donut chart to filter the data. To reset the view, click on **CLEAR ALL** to clear the filter.

> **NOTE**
> For more information, see the Monitored Inventory User Interface section.

Filter Monitored Inventory by Attribute

The Filter section on the Monitored Inventory page now includes additional attributes that you can use to filter the data. Click **More** to view the additional attributes.



This list also includes the option **Others** that you can use to filter by inventory attributes.

> **NOTE**
> For more information, see the Monitored Inventory User Interface section.

Insights Enhancements

The Insights page includes the following enhancements:

- **Label Change:** The Insights label for Alarms on the Insights page changed to **High Severity persistent alarms noticing the significant change**. In the earlier version, the label was **High priority policies having persistent alarms**.
- **Export Button Added:** You can export the insights information on the Services page to an Excel sheet.

  > **NOTE**
  > The Services without devices section displays only the service name. However, when you export this section, the Service ID is also included.
- **Pagination Added:** Pagination added to the Devices not in services and Devices shared across services sections.

> **NOTE**
> For more information, see the Insights section.

DX Operational Intelligence SaaS to Automic Integration

In this SaaS update, we have simplified the integration with an on-premises Automic environment. With this simplified support for hybrid deployment of Operational Intelligence SaaS and on-premises Automic, you can more easily perform automated diagnostics, triage, and intelligent remediation activities.

> **NOTE**
> For more information, see the Configure Automic Automation section.

DX Dashboards

This release of DX Dashboards includes the following enhancements:

- API Support Added for DX Dashboards
- New OOTB Dashboards
- Source Name Specifier and Attribute Name Specifier Variables Support for Graph Visualization
- Dashboards Limit Increased on Reports Page
- Connector Health Dashboard Enhanced

> **NOTE**
> For more information, see the DX Dashboards documentation.

Platform Improvements

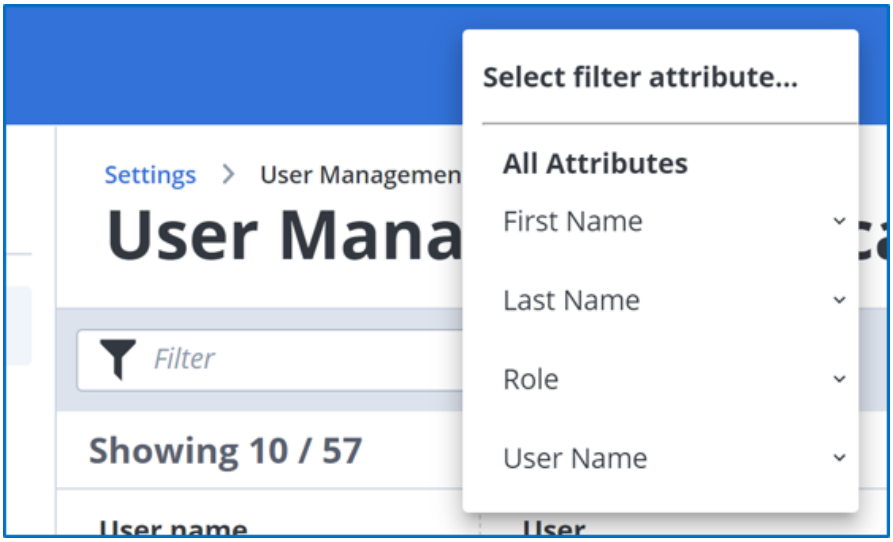
This release includes the following changes and enhancements:

- User Management Enhancements
- Tenant Level Features Management

User Management Enhancements

The User Management page includes the following enhancements:

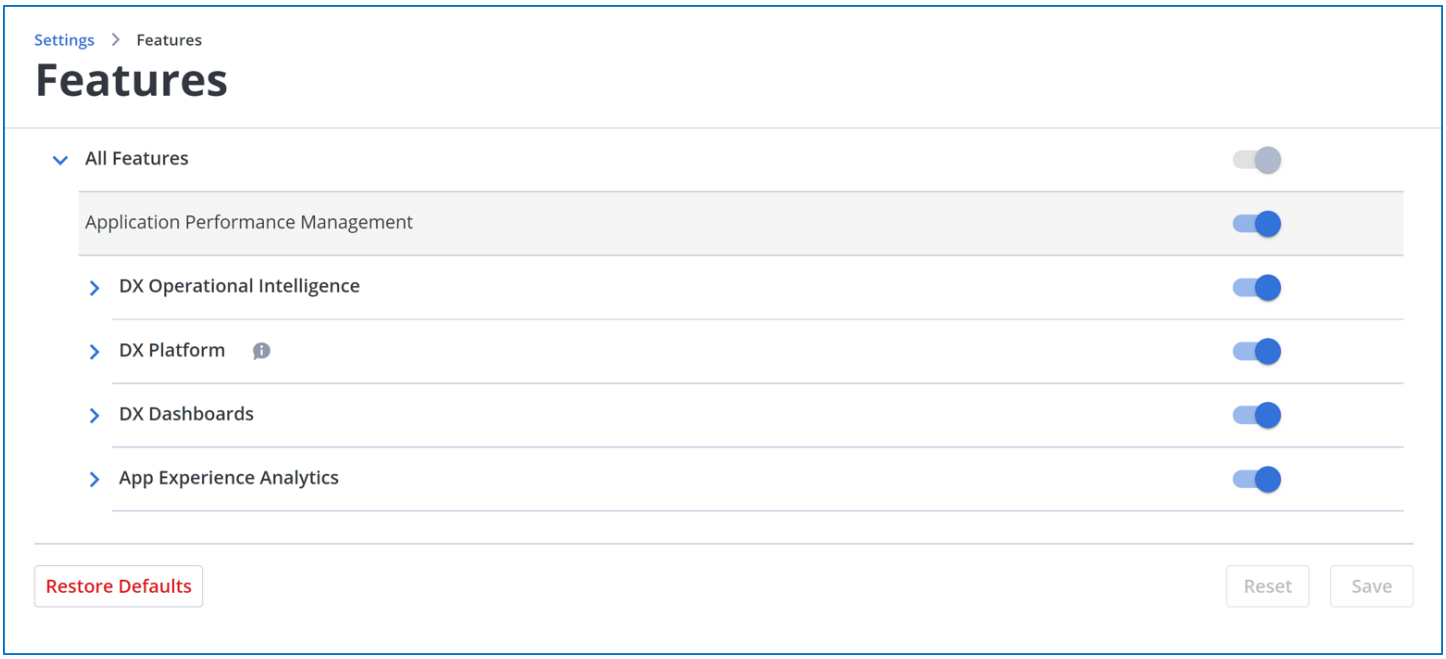
- Basic Authentication Enabled Tenants:
  - **Filter Attributes Added:** You can now filter the data by the following attributes: First Name, Last Name, Role, and User Name. In the earlier version, you could filter only by entering text.



- SAML Enabled Tenants:
  - **Added Filter Section:** The Users page now has the Filter section.

Tenant Level Features Management

Starting with this release, the **Settings** page includes the **Features** tile that enables you to manage features at the tenant level. You can enable or disable the features for each of the capabilities as required.



NOTE
For DX APM, you can enable or disable all the features at once.

NOTE
For more information, see the Features Management section.

RESTMon 2.2.1 Release Notes

The RESTMon 2.2.1 release includes the following features and improvements:

- In-process Filters to Filter and Ingest Only Required Data
- Monitoring Overview Dashboard Improvements
- RESTMon Docker - Added Support for Installing on Existing PV/PVC
- Performance Improvements
- Vulnerabilities Fixes

In-process Filters to Filter and Ingest Only Required Data

Now, ingest only the required CIs instead of everything. Many monitoring tools do not provide URL-based filters and therefore the amount of data that is returned can be enormous. Ingesting all this data means disturbing the sanity of your environment.

You can configure RESTMon to ingest only the required information using the attribute filter in the **profile.json** file. For example, in the attribute filter, define the attributes such as application name, hostname, and tags and RESTMon filters out the unnecessary data and ingests only what matches the criteria.

For example,

```
"attribute filter":[
   {
      "name" : "<attribute>",
      "value": [<value>]
   }
]
```

You can find the filter attributes in the Inventory and Topology sections of the schema. Alternatively, if the data is already ingested into DX Operational Intelligence, you can find the attributes in the **Entity Details** section on the **Monitored Inventory** page. For more information, see the Filter Entities and Related Data Before Ingestion section.

Monitoring Overview Dashboard Improvements

The Monitoring Overview dashboard in DX Dashboards includes the following changes:

- Merged multiple error panels into one for a better user experience. For example, all error-related panels are now merged into a single panel.
- Added units to all panels. For example, bytes and milliseconds.
- Added new panels:
  - Requests Submitted Time
  - Average Response Size (Byte)
  - Average Response Time (ms)
  - Total Requests Per Interval (number)
  - Total Success Requests Per Interval (number)
  - Total Failed Requests Per Interval (number)
  - Total Request Retries Per Interval (number)
  - Total Paginated Requests Per Interval (number)

RESTMon Docker - Added Support for Installing on Existing PV/PVC

Enhancements to Helm charts to support installing RESTMon on an existing PV/PVC instead of creating new. This is applicable to the docker version and can be especially useful when you want to run and install RESTMon on the same OSE cluster as DX Operational Intelligence.

Performance Improvements

The Average Process Time improved up to 12x for metrics and the overall Average Process Time improved up to 4x when ingesting all data types (topology, alarms, metrics) together.

2023.1.1 Release Notes

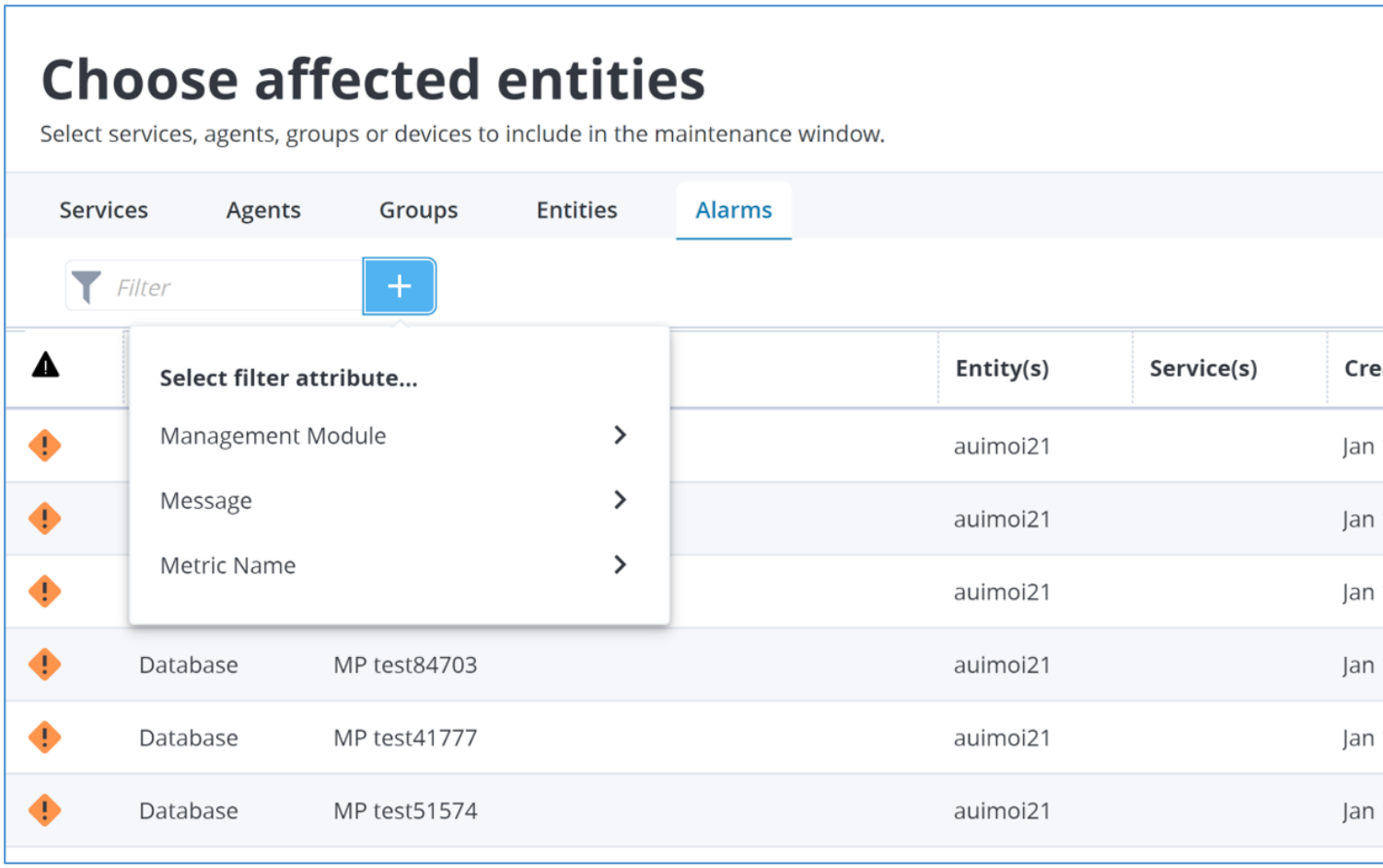
This release includes the following new feature:

- Additional Alarm Filter Attributes for Maintenance Windows
- Additional Filter Attributes for SLIs and Monitoring Groups
- CI Attributes Supported for Incident Fields Mapping
- CI Attributes Supported for Ticketing and Notifications
- Define Services with Transactions
- DX Dashboards New Features/Enhancements
- Handling Alarm Updates from Source Products
- HTML Format for Email Messages
- New DX Gateway Version Released
- Simplified Filter for All Alarms
- Tenant Profile Management
- Ternary Expressions Supported for SLIs/SLOs
- Updating Specific Fields in Raw Alarms Using API

Additional Alarm Filter Attributes for Maintenance Windows

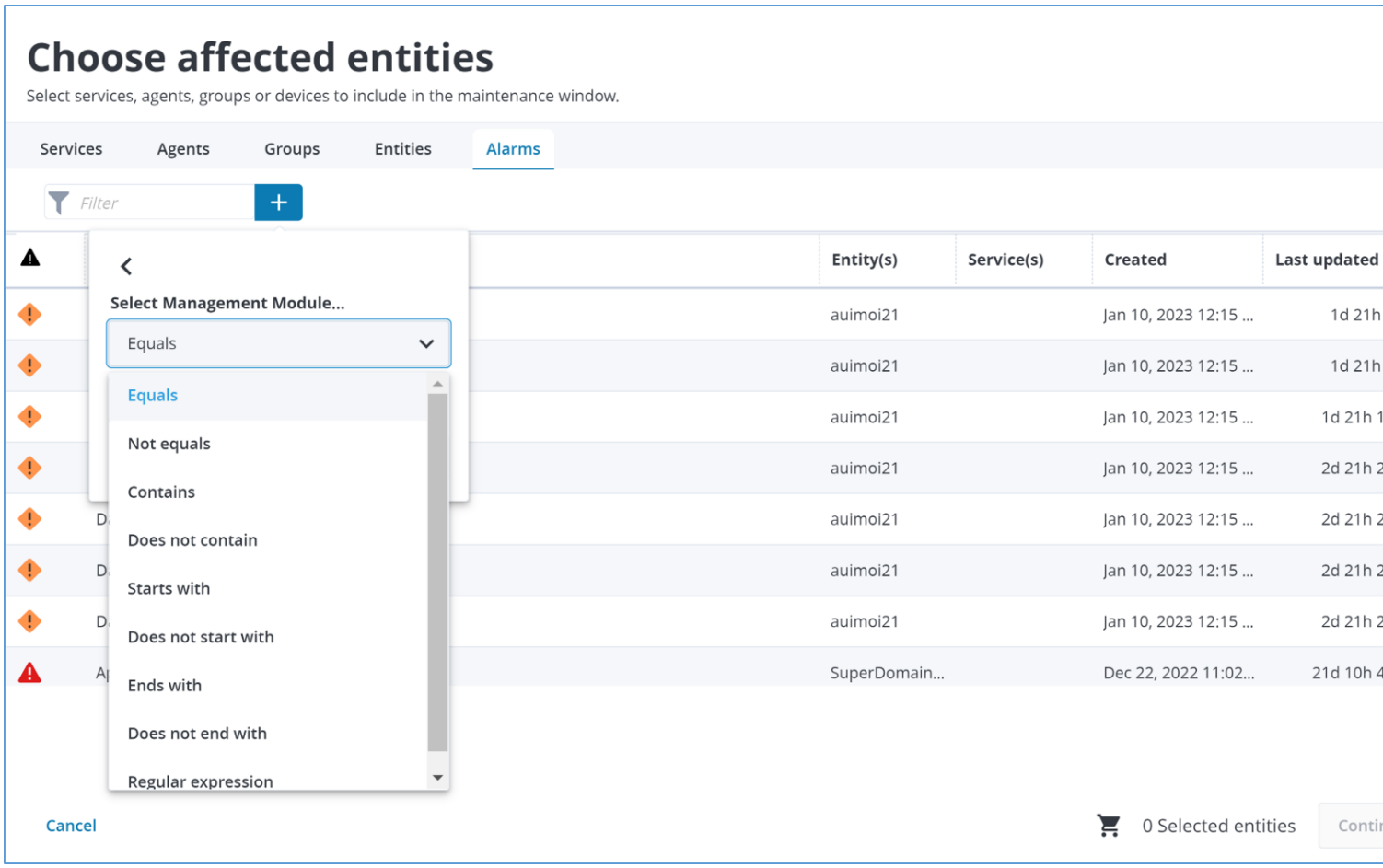
Starting with this release,

- You can create maintenance windows for alarms based on the management module, message, and metric name. In the earlier version, only metric name was supported for alarms.



You can filter by a single attribute or multiple attributes in a single maintenance window. When filtering by multiple attributes, the filter uses the AND operation to fetch the results. DX Operational Intelligence also uses the same filter criteria with the AND operation to mark the alarms for maintenance.

- The filter supports operations such as Equals, Not equals, Contains to filter the alarms. In the earlier version, only regular expression was supported.

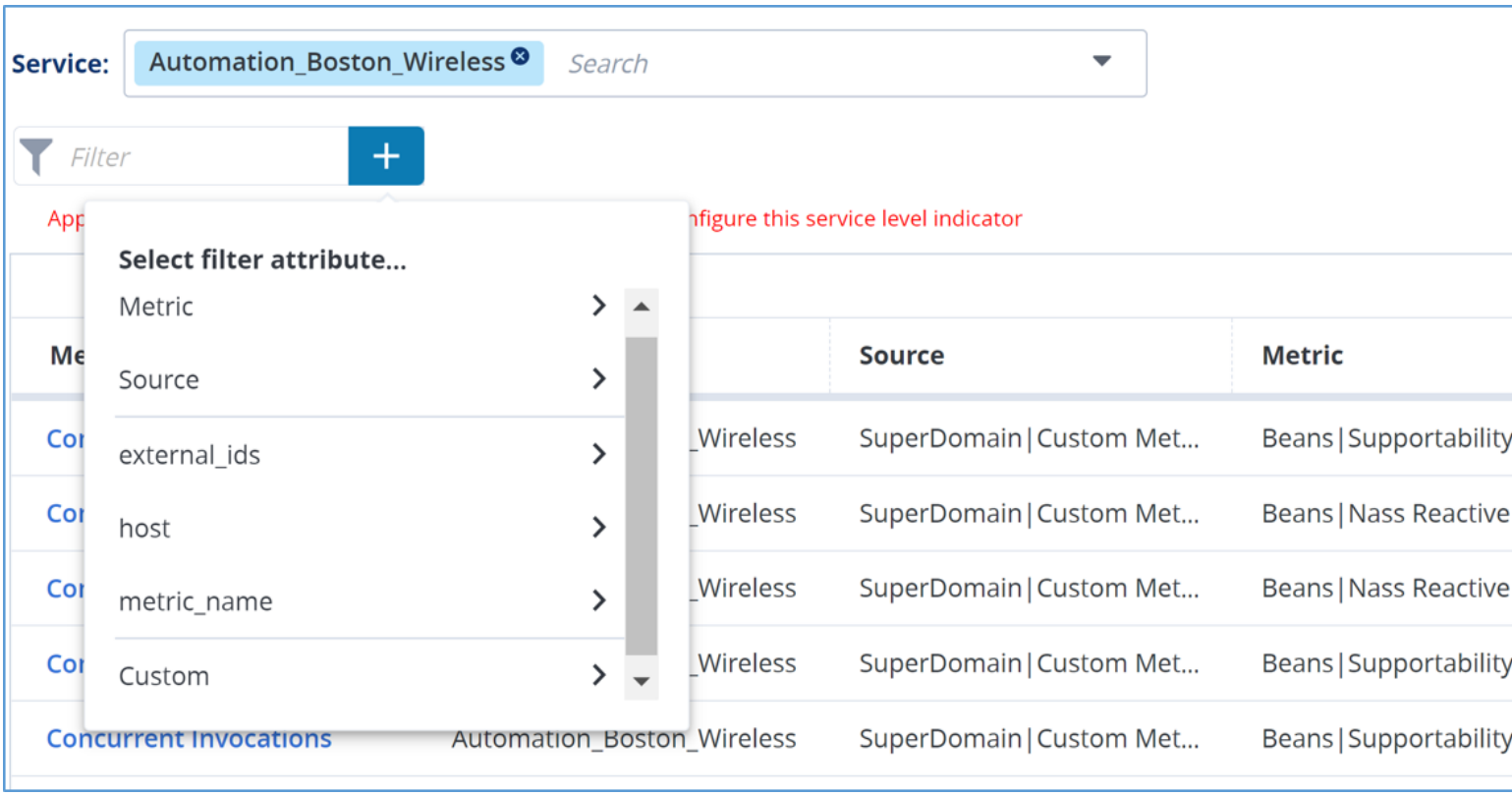


NOTE
For more information, see the Create a Maintenance Window section.

Additional Filter Attributes for SLIs and Monitoring Groups

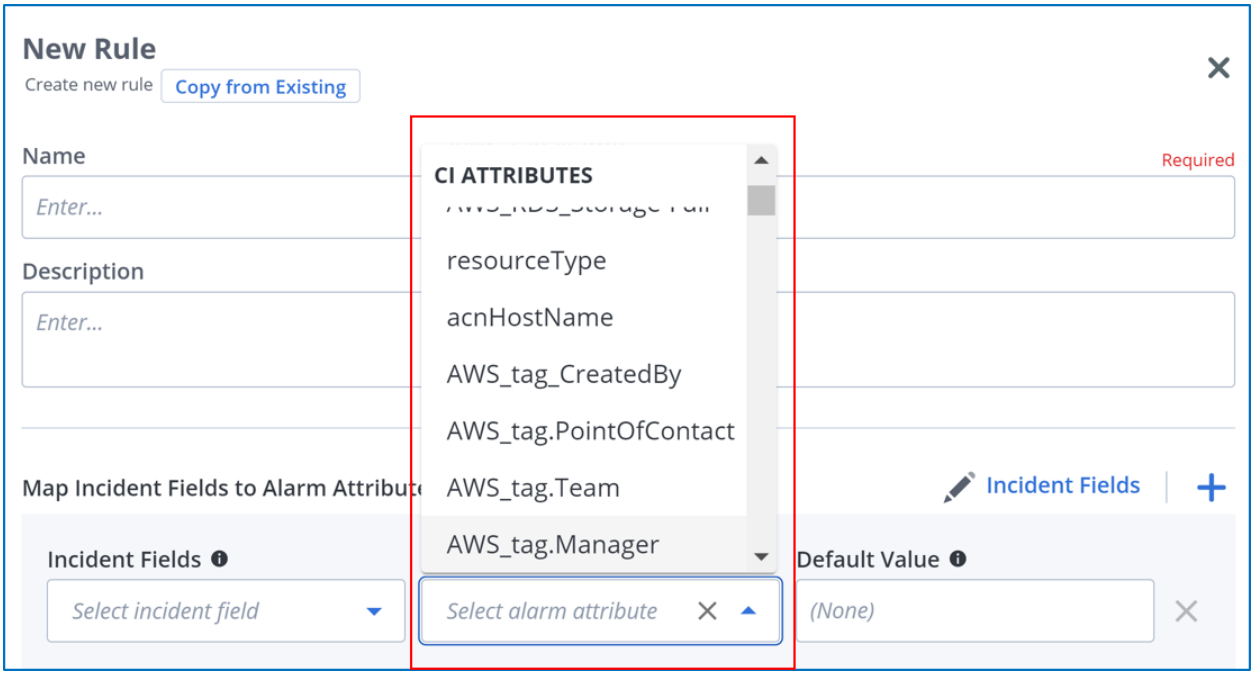
You can now filter the metrics for the SLIs and Monitoring Groups by their source name (Source), attribute name (Metric), metadata attributes, and custom attributes. If an attribute is not available in the metadata, you can use the custom filter. In the earlier version, only the source and metric filter attributes were supported.

The following image illustrates the filter attributes for SLI:



CI Attributes Supported for Incident Fields Mapping

In addition to the alarm attributes, you can now map the incident fields to the Configuration Item (CI) attributes in the ticket enrichment rule. When an alert is raised for a CI, the notifications and incidents include these additional details.



For the CI attributes to be available for selection, you must first register the required CI attributes using the API. After they are registered, the **All Alarms** option on the New Ticket Enrichment Rule panel displays these attributes under CI Attributes.

- The CI attributes are supported only for All Alarms.
- The CI attributes are available for selection only if they are registered.
- If the CI attribute does not have any value, then the default value is displayed for that attribute in the notification.
- If the default value is not defined, then the incident field value is blank or is not populated.

NOTE

- For more information, see the Ticket Enrichment Rules section.
- For more information about how to register the CI Attributes, see the Add or Update the CI Attributes section.

CI Attributes Supported for Ticketing and Notifications

You can now enrich notifications and tickets by adding CI attributes to your message templates, policies, tickets, and channels (Slack and Webhook). However, to make these attributes available on the UI, you must first register the attributes using the Add or Update CI Attributes API.

NOTE
For more information about how to add or update the CI attributes, see the Add or Update the CI Attributes section.

After the attributes are registered, these attributes are available in the following sections:

- **Message Templates:** On the Create and Edit pages, the registered attributes are listed under **Add Variables > CI Attributes**.

  NOTE
  For more information, see the Create Custom Template and Update Custom Template sections.
- **Policies:** On the Create and Edit pages, the registered attributes are listed in the **Filter** under **CI Attributes** only for All Alarms.

  NOTE

  - The CI attributes are not supported for other alarms.
  - For more information, see the Create Policy section.
- **Generic Webhook and Slack Channels**: On the Create and Edit pages, the registered attributes are listed under **Add Keys > CI Attributes.**

  NOTE
  For more information, see the Configure Slack Channel and Configure Webhook Channels sections.

Define Services with Transactions

Starting with this release, you can configure a service to follow the DX APM business transactions to other elements. You can also include these transactions in the service. The **Service Configuration** section in the **Topology View** includes the **Follow Transactions** option that you can select to display the transaction path for the service.

The **Topology View** also includes a filter named **Transactions Selected**. You can use this filter to display the view for a single transaction or multiple transactions.

NOTE
For more information, see the Service Configuration section.

DX Dashboards New Features/Enhancements

This release of DX Dashboards includes the following features and enhancements:

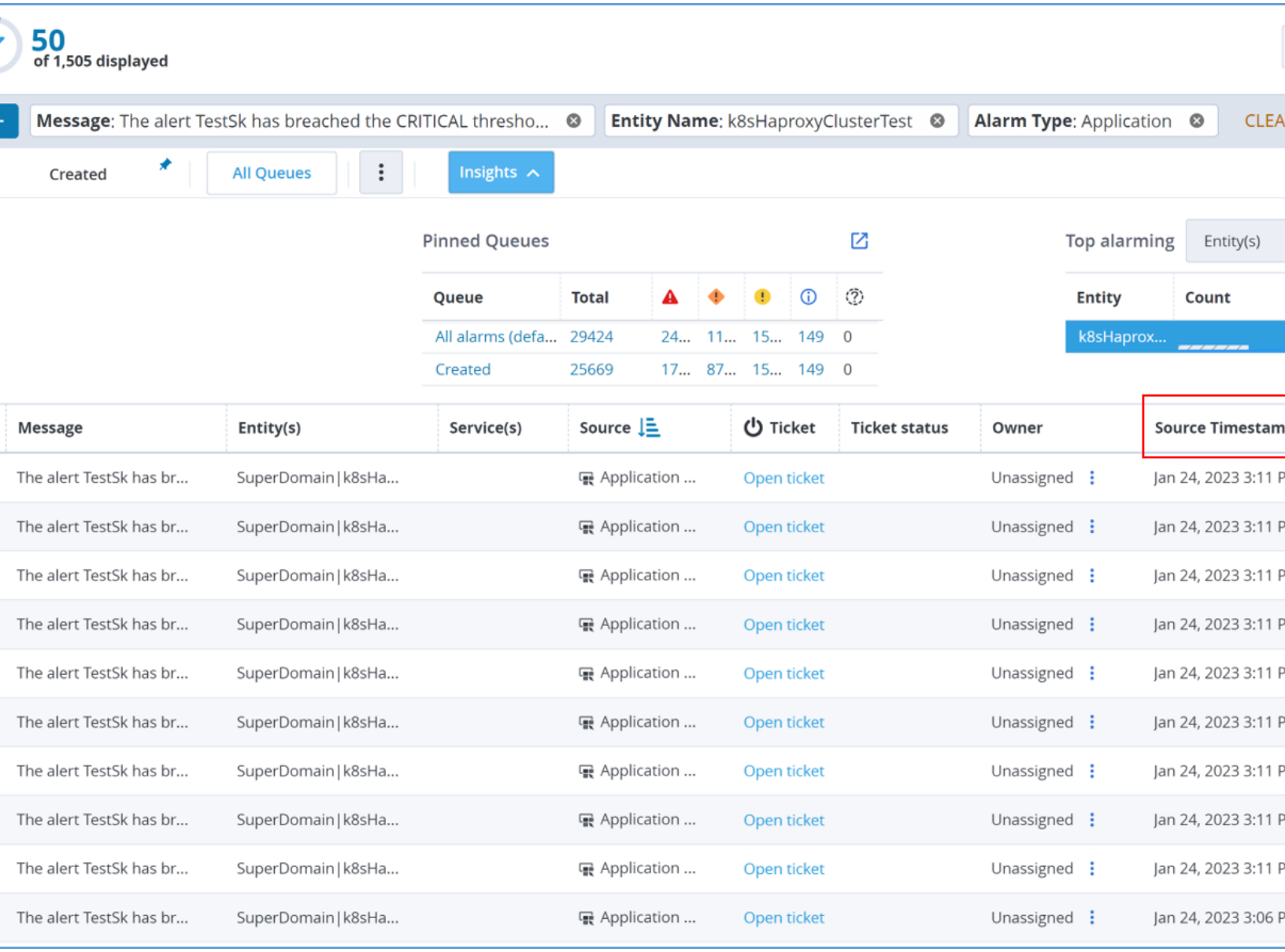
- Log Out Option Added
- New OOTB dashboards
- UI Theme Selection Enabled
- Known Issue

> **NOTE**
> For more information, see the DX Dashboards documentation.

Handling Alarm Updates from Source Products

In the existing version of DX Operational Intelligence, if the timestamp of the incoming alarm from a source product is older than the timestamp of the alarm update made in DX Operational Intelligence, then those incoming alarm updates are ignored and are not reflected in DX Operational Intelligence.

Starting with this release, the **sourceTimestamp** column is added on the **All Alarms** page. This column is updated with the source product timestamp along with the incoming updates, even if the timestamp is old.

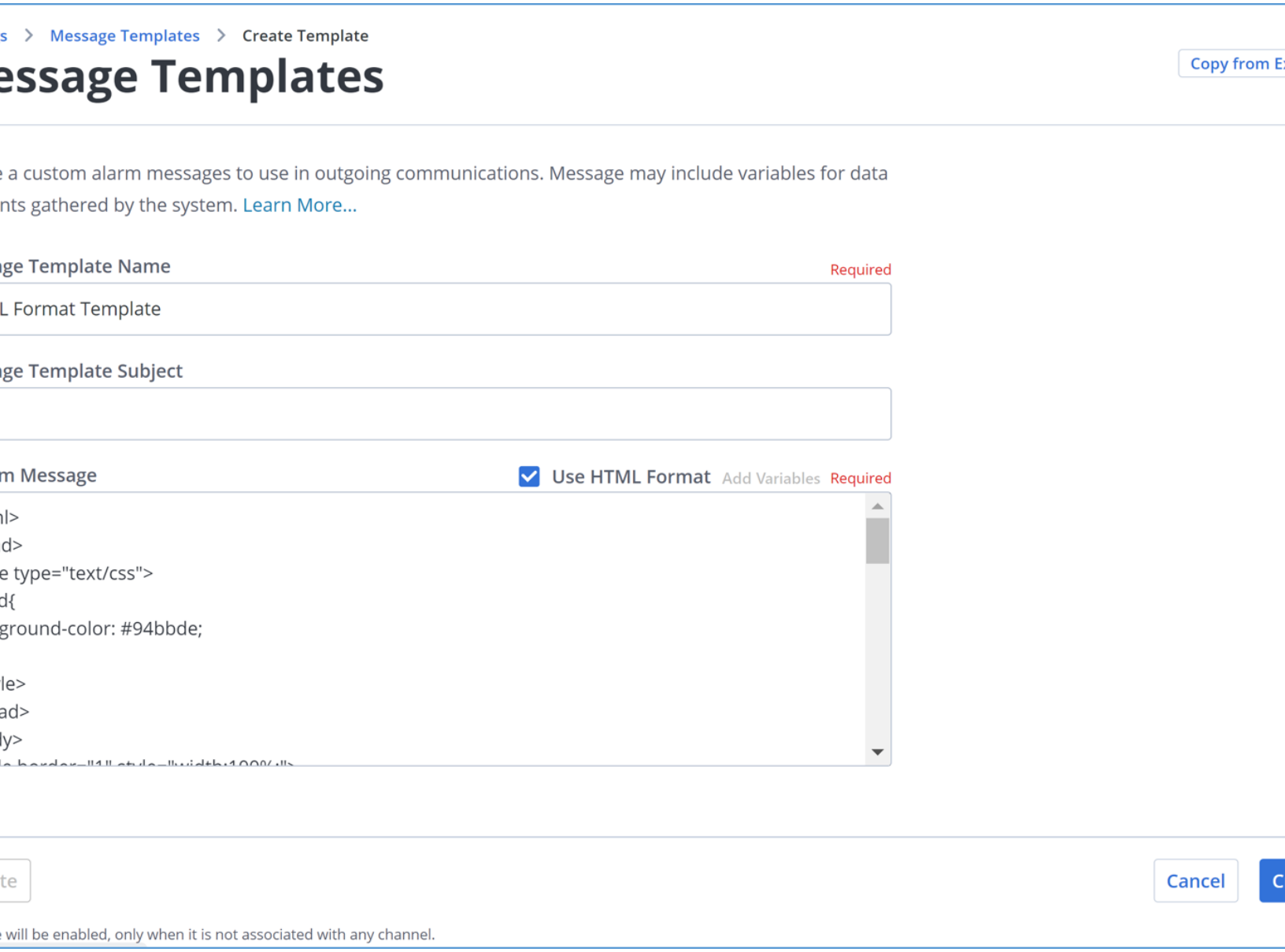


NOTE

- The **Source Timestamp** column is not displayed by default. To display this column, select **Source Timestamp** in the **Customize Columns** list.
- For existing alarms, the **Source Timestamp** column is populated only after there is an update to the alarm from the source product.
- The **Source Timestamp** column reflects the timestamp for the source product updates and the **Last Updated** column reflects the timestamp for alarms updated made in the DX Operational Intelligence.
- If no updates were made in DX Operational Intelligence, then both columns reflect the timestamp for the source product.
- For more information, see the Alarm Analytics section.

HTML Format for Email Messages

Starting this release, you can customize the content format in the message template using the **Use HTML Format** option. For example, to send the message content in a table format, you can select this option and can provide the HTML payload in the **Custom Message** section of the message template.



Consider the following points for the custom HTML payload:

- When you select this option, the **Add Variables** option is unavailable.
- All the HTML elements must have a closing tag.
- The HTML tags are case-sensitive.
- All the HTML tag-specific attribute values should be enclosed in double quotes.
- All the HTML elements must be nested properly.
- To use paragraphs or plain text, use the appropriate HTML tags.

NOTE
For more information, see the Create Custom Message Template section.

New DX Gateway Version Released

A new version of DX Gateway (23.1.3) is available with this release. Integration of DX Operational Intelligence SaaS with BMC Remedy and CA SDM is now REST-based instead of SOAP. To use this version of DX Gateway, ensure to update the port for REST instead of SOAP.

Simplified Filter for All Alarms

You can now filter the data on the All Alarms or Alarm Details pages using the **+** and **-** buttons instead of manually specifying the filter. On the All Alarms page, hover over the value you want to filter by and click + to add the filter and click - to remove the filter. In the Alarm Details section of the Overview tab, hover over the attribute value to display the filters. Click **+** apply a single filter attribute and click the **funnel** button to add multiple filter attributes.

The following illustration shows how to filter the alarms directly:

NOTE
For more information, see the Alarm Analytics section.

Tenant Profile Management

The **Settings** page in DX SaaS now includes a new tile named **Tenant Profile** to help you manage the tenant. The **Tenant Profile** page displays the tenant display name, products that are provisioned for the tenant, and their serial numbers. On this page, you can edit the tenant display name.

NOTE
For more information, see the Tenant Profile section.

Ternary Expressions Supported for SLIs/SLOs

DX Operational Intelligence supports ternary or conditional expressions for SLIs and SLOs. For example,

- ${pipeline:avg_metric1} >= ${pipeline:avg_metric2} ? Max(${pipeline:avg_metric1}, ${pipeline:avg_metric2}) : Min(${pipeline:avg_metric1} ,${pipeline:avg_metric2})
- (${pipeline:avg_metric1} >= ${pipeline:avg_metric2} && ${pipeline:avg_metric2}>0) ? Max((${pipeline:avg_metric1} + ${pipeline:avg_metric2}),10.0) : 0

Any number in the expression should be a decimal. For example, (${pipeline:max}>= ${pipeline:min} && ${pipeline:min}>0)?Max((${pipeline:min}+${pipeline:max}),500.0):100.0.

NOTE
For more information, see the Create SLIs and SLOs section.

Updating Specific Fields in Raw Alarms Using API

You can add or update fields in the raw alarms (all alarms) using the APM Gateway or NGINX endpoint that is available on the **Connector Parameters** page.

NOTE
For more information, see the Update Specific Fields API section.

2022.12.1 Release Notes

This section describes the new features, enhancements, known issues, and fixed issues in this release:

- **New Features:**
  - Insights into Services and Alarms
- **Enhancements:**
  - Additional Privileges for Alarms and Situations
  - Automate Alarm Actions for Historical All Alarms
  - Custom Situation Preview Enhanced
  - Customize Situation Definition Name
  - DX Dashboards
  - DX Gateway Enhancements
  - Enable or Disable Algorithmic Clustering
  - Embed Dashboards in SA Service Details
  - Exclude SLO Calculation During Maintenance Window
  - General Availability of SLI/SLO
  - Message Templates Enhancements
  - Mute Existing Open Alarms During Maintenance Window
  - ServiceNow Version Support
  - Timezone Support for SLO Calendar Window Calculation
  - User Tokens Support for Maintenance APIs
  - View Pipeline Errors
- Known Issues:
  - New SLI Alarm is Not Created
  - Existing Alert is Not Closed
  - SLI Alarm is Not Generated

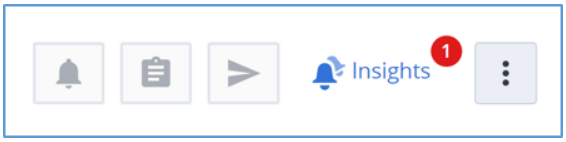
Insights into Services and Alarms

You can configure DX Operational Intelligence to provide insights into services and raw alarms using the **Settings** > **Insights** tile. The Insights page lists the insights that are provided out-of-the-box, and they are enabled by default. You can enable or disable them as required.

> **NOTE**
> By default, this feature is disabled. To enable this feature for your tenant, contact **Broadcom Support**.

The following image illustrates the Insights page:

Information for the enabled insights is displayed on the **Service Details** and **All Alarms** pages. Click **Insights** displayed on the top-right corner to view the details.



NOTE
For more information, see the Insights section.

Additional Privileges for Alarms Analytics

This release includes the following additional privileges for Alarm Analytics:

- Alarms Views
- Situations

NOTE
For more information, see the Alarm Analytics Access Privileges section.

Alarms Views

Category	Privileges
Unassociated Alarms	Allows to view alarms that are not associated with any service: • Unassociated Alarms

Situations

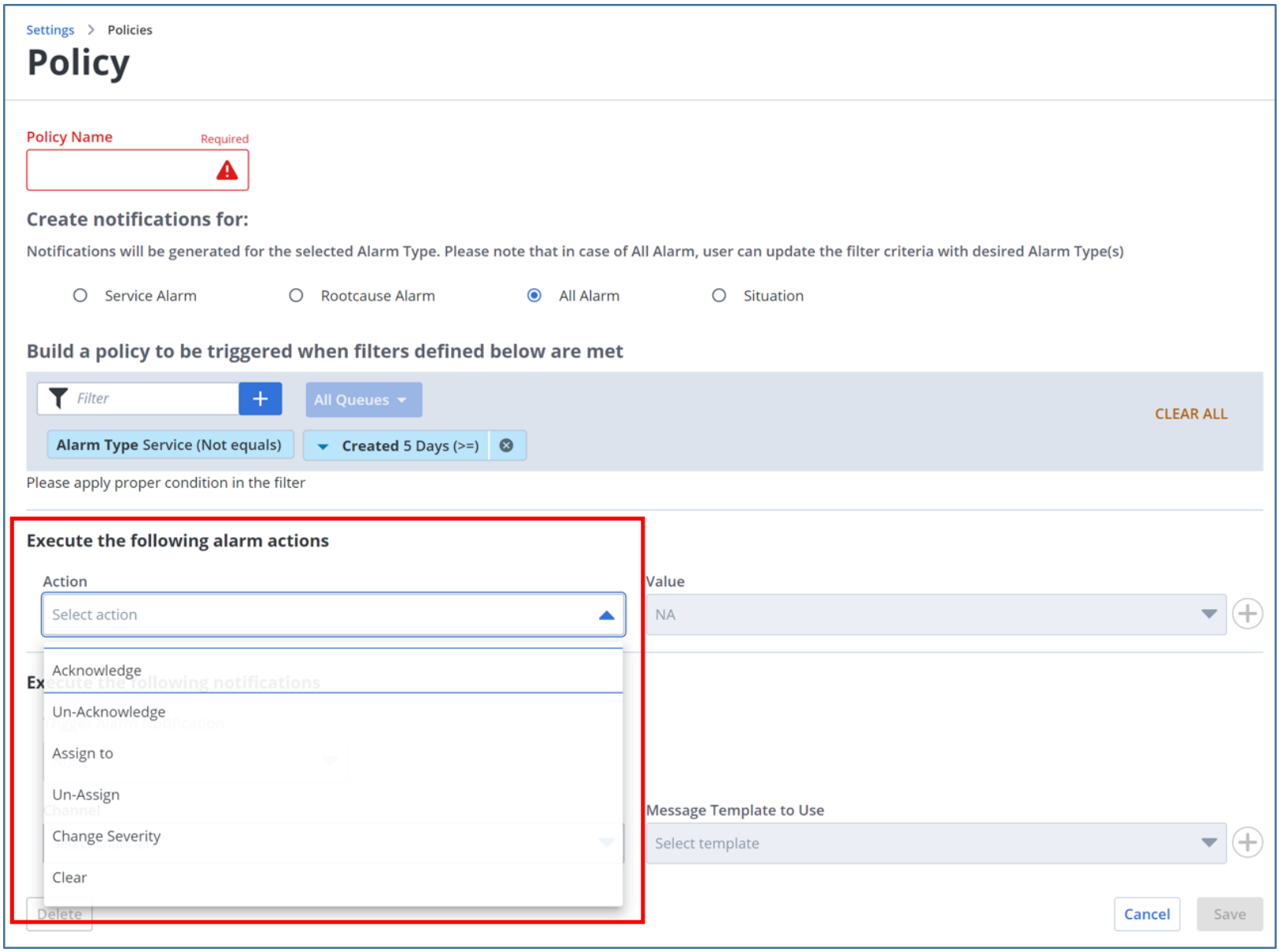
Category	Privileges
Situation Stable Window	Allows the following access privileges to situation windows: • View Situation Stable Window • Update Situation Stable Window
Situation Tenant Configuration	Allows the following access privileges for algorithmic clustering: • View Situation Tenant Configuration • Update Situation Tenant Configuration • Delete Situation Tenant Configuration
Situation Search Action	Allows the following access privilege for situation search actions. • Situation Search Action

Automate Alarm Actions for All Alarms

You can now configure a policy to automatically perform alarm actions on historical raw alarms when the filter condition is met.

For example, you can create or update a policy to automatically acknowledge all alarms that are older than five days. When the policy filter criteria are met, all alarms older than five days are automatically acknowledged.

The following image illustrates the alarm actions you can automate:



NOTE
For more information, see the Create Policy section.

Customize Situation Definition Name

You can define a custom name for a situation definition using operations. Each operation has a set of attributes (also referenced as supported fields) that are provided out-of-the-box. For example, you can add the operations (**Policy Name: {policyname()}** and **Count:{count(severity)}**) to customize the definition name. This custom name is appended to the situation name and is also displayed in the preview.

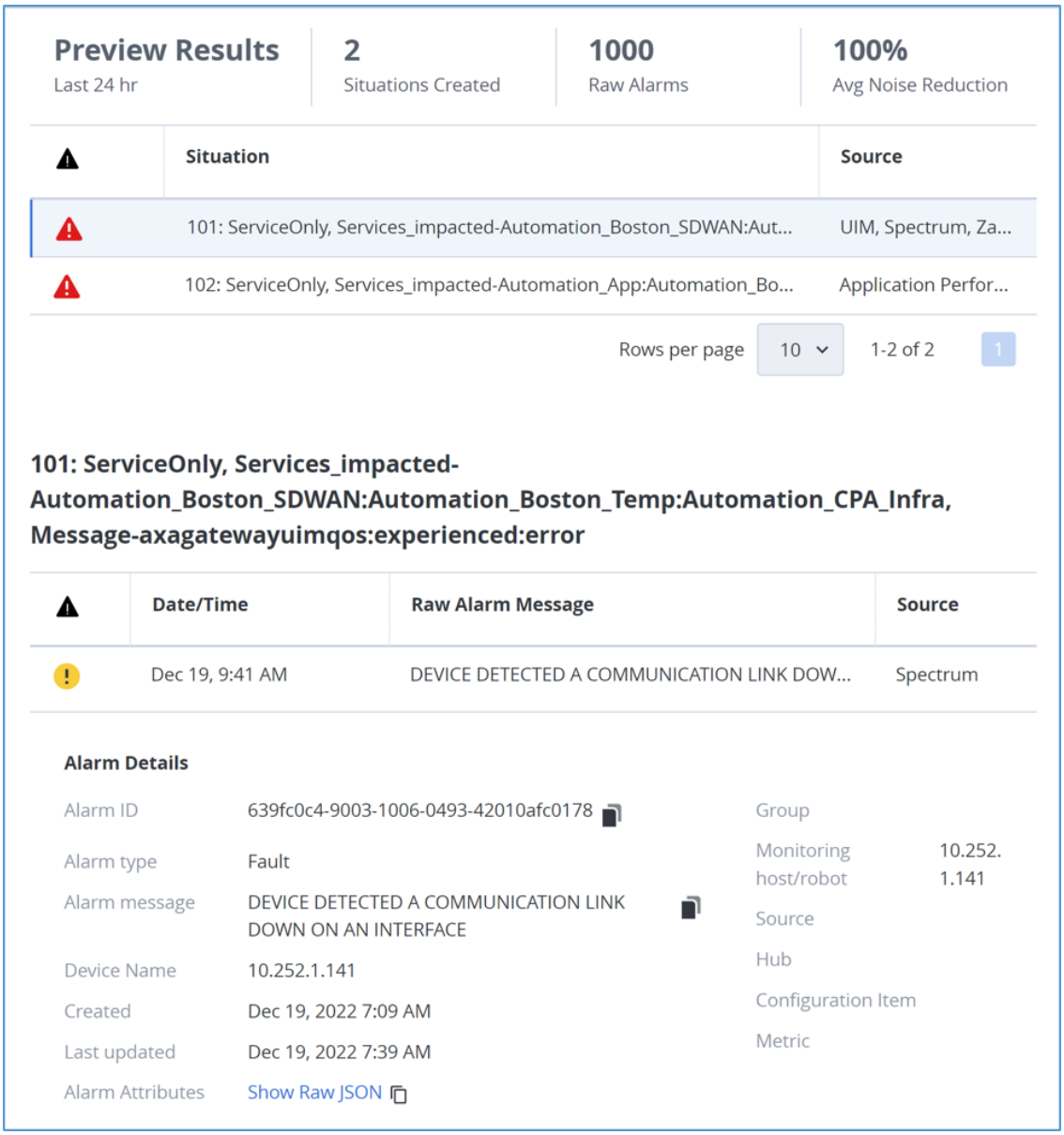
You can use the OOTB supported fields or add fields for these operations using the API.

NOTE

- For more information, see the Define Custom Situation Name section.
- For more information about how to add these operations using the API, see the Add or Remove Supported Fields for Custom Situation Name Operations API in the **Situation Clustering Dimensions APIs** section.

Custom Situation Preview Enhanced

The custom situation preview, which provides a quick view of the resultant clusters, is enhanced to display the alarm details. Click the alarm to view the alarm details.



NOTE
For more information, see the Preview Results of Custom Situation Definition section.

DX Dashboards

In this release, DX Dashboards includes the following new features and enhancements:

- New OOTB Dashboards
  - APM-WebMethods
  - AXA PLA Telemetry Dashboards
  - APM PLA Telemetry Dashboards
- Option to Schedule Reports Monthly
- Embed DX Dashboards in DX Operational Intelligence
- Discover Using URL Parameters

> **NOTE**
> For more information, see the DX Dashboards documentation.

DX Gateway Enhancements

DX Gateway includes the following changes:

- Tomcat is not included in DX Gateway
- Java 11 is required

Enable or Disable Algorithmic Clustering

You can now enable or disable algorithmic clustering for situations using the **Enable Algorithmic Situations** option in the alarms view filter. This option is enabled by default. When you disable this clustering, the existing active algorithmic type clusters become orphans, and no new algorithmic type clusters are formed.

> **NOTE**
> For more information, see the Alarms View Filter section.

Embed DX Dashboards in Service Analytics Page

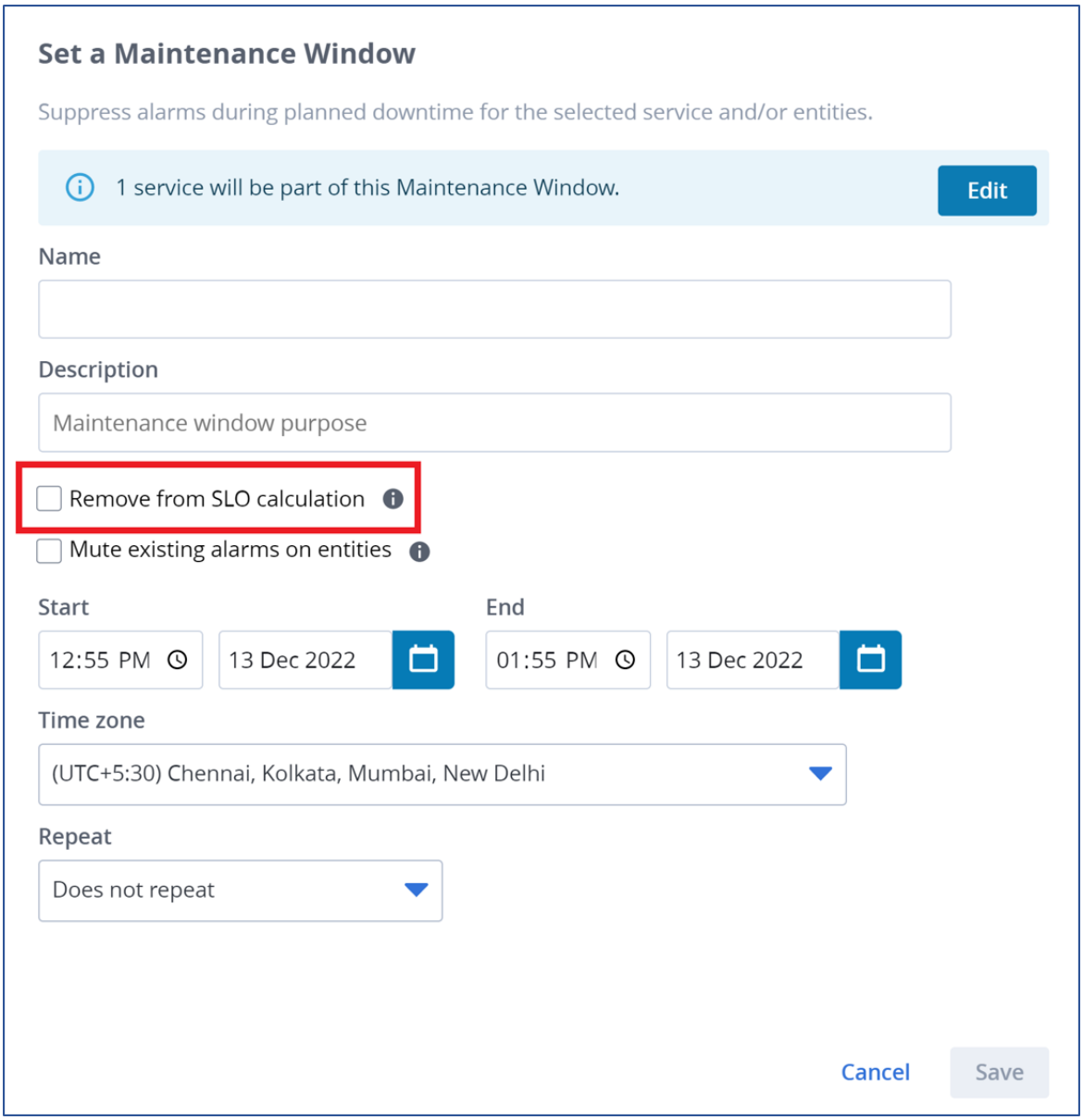
You can embed a DX Dashboard to be viewed on the Service Analytics Details page. By default, the **Service Details** dashboard, which is the OOTB dashboard, is tagged and embedded. However, you must pin this dashboard on the Service Analytics Details page to appear as a tab.

To embed other dashboards, add the **DX OI Service Details** tag to the dashboard on the **DX Dashboards > Dashboard Settings > General** page.

> **NOTE**
> For more information, see the Service Analytics Details page.

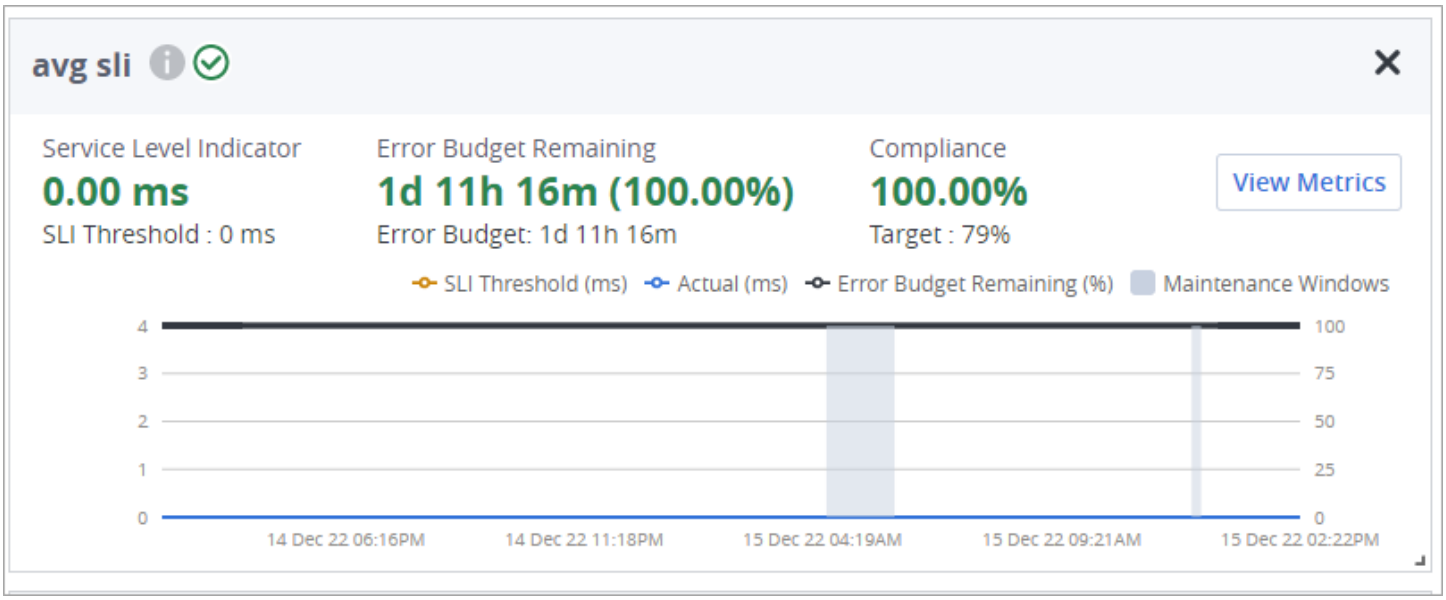
Exclude SLO Calculation During Maintenance Window

You can configure the maintenance window to exclude the SLO calculation using the **Remove from SLO Calculation** option. When this option is selected, the SLO calculation for the selected service is stopped during the maintenance window, and only SLI is calculated. The following image shows where you can configure this option.



NOTE
For more information, see the Create a Maintenance Window section.

You can also observe that the period during which the SLO calculation is stopped appears as grey within the **Service Level Indicators** widget as shown. Click the **Maintenance Windows** legend to clear the grey section on the chart.



General Availability of SLI/SLO

The SLI/SLO functionality is now enabled by default. Before this release, you had to contact **Broadcom Support** to get this feature enabled.

Message Templates Enhancements

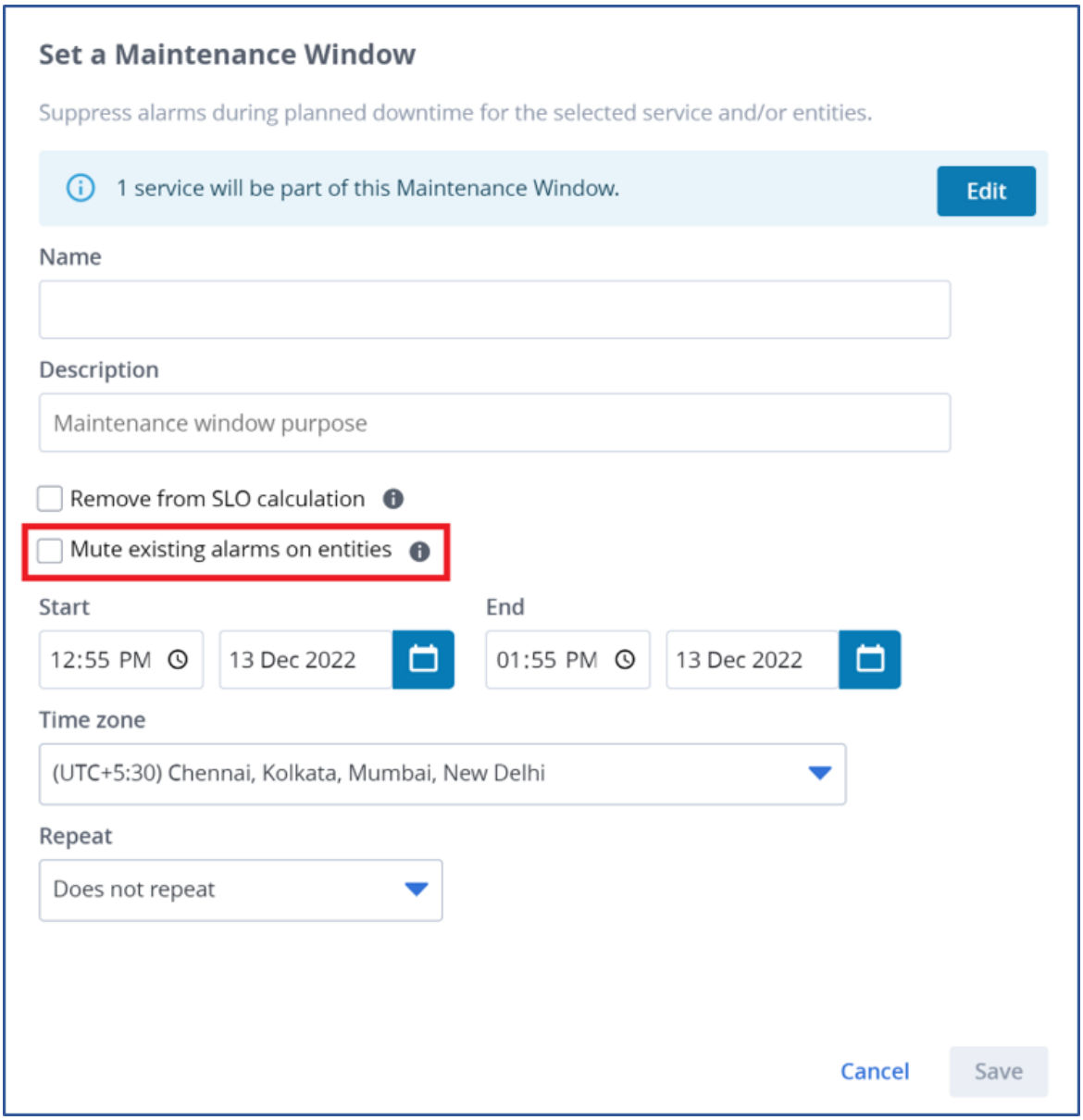
This release includes the following enhancements:

- New OOTB Message Templates: New default templates are available for DX APM, Spectrum, and UIM.
- The **Make as Default** option is no longer available.
- The default message templates cannot be edited or deleted. However, you make a copy of these templates.

> **NOTE**
> For more information, see the Message Templates section.

Mute Existing Open Alarms During Maintenance Window

You can now mute the existing open alarms that were raised before or during an active maintenance period using the **Mute existing alarms on Entities** option.



When you select this option, all updates to the existing alarms which are part of this maintenance window, are muted during the maintenance window. They are restored to their original state when the maintenance window ends. For example, the alarm is restored to the open state if the alarm is not closed during the maintenance period.

During the maintenance period, the muted alarms are greyed out, and a maintenance icon is also displayed for those alarms. If multiple maintenance windows are scheduled at the same time on the same entity, and if this option is selected in at least one of the schedules, then the alarms for all the overlapped schedules are muted.

If you end an active maintenance window or delete a window, the alarms are restored to their original state, that is, the alarms come out of the maintenance.

NOTE
For more information, see the Create a Maintenance Window section.

ServiceNow Version Support

DX Operational Intelligence now supports integration with the San Diego version of ServiceNow.

Timezone Support for SLO Calendar Window Calculation

You can now select the time zone for the SLO calendar window calculation. Earlier, the time window for the SLO Percentage and Error Budget Aggregation functions started based on the system time. Starting this release, you can select the time zone.

User Tokens Support for Maintenance APIs

The Maintenance APIs now support both Bearer and User tokens.

View SLI Pipeline Errors

You can view logs for the SLI pipeline errors (run time exception) using the **View Logs** option. For example, you can view logs for errors such as,

- Consecutive SLI is divided by 0
- The Aggregator fails to receive the required parameters from all the upstream functions due to incorrect skew value.

When you click the **View Logs** button, the error messages for the SLI are displayed in DX Dashboards.

> **NOTE**
> For more information, see the Create SLIs and SLOs section.

Known Issues

- **New SLI Alarm is Not Created**
  After you close the alert manually, a new SLI alarm is not created even when the condition is met again. The alarm is created only if the severity is changed.
- **Existing Alert is Not Closed**
  If the SLI name is changed or the alert or SLI is deleted, the existing alert is not closed.
- **SLI Alarm is Not Generated**
  When the SLI is created on a service, if you add another service to the SLI after adding the alert configuration, the new service is not added.
  Workaround: Close the alert.

22.08 Release Notes

This release includes the following new features enhancements, fixed defects, and known issues:

- Documentation Changes
- OI Universes
- Documentation Changes
- Role-Based Access Privileges Enhancements
- Ticket Enrichment Rules
- Service Analytics Enhancements
- Alarm Analytics Enhancements
- Performance Analytics Enhancements
- Maintenance Windows Enhancement
- Dashboard Enhancements
- Vulnerability Fixes
- Known Issues
- Fixed Defects

Documentation Changes

Starting with this release, the RESTMon section has been restructured for a better user experience. Any RESTMon-related information that was bookmarked earlier will not work. We recommend you to create bookmarks for the RESTMon-related information from the RESTMon documentation. Searching RESTMon-related information through google displays some of the old links that will not work. Google takes time to index the restructured RESTMon documentation. We recommend you to use the search functionality in the documentation site until Google indexes our content.

OI Universes

Data Level Access Permissions for Capabilities

The new OI Universe presents the RBAC data scope (RBAC Subject). Using an OI Universe, a Tenant Administrator or a Security Administrator can define the data access permissions to control what data users have access to. They can also limit access to a specific set of CIs and their related Alarms, Metrics, and Topology. For example, a Tenant Administrator can create a universe consisting of only a subset of DX Operational Intelligence services that users must have access to and must assign users or SAML groups to that universe. With data access being restricted, users can view data only for the universes that they have been assigned explicit access to. The OI Universe leverages the DX Operational Intelligence services to select the CIs a user should be provided access to.

A user applies a data filter context to their current Operational Intelligence view through the selection of a specific universe from the universe dropdown list. This universe dropdown list is available at the top of the following DX Operational Intelligence pages:

- **Service Analytics:**
  - Services Overview page
  - Service Details page
- **Alarm Analytics:**
  - All Alarms
  - Service Alarms
  - Situations
- **Monitored Inventory:**
  - Monitored Inventory page
- **Capacity Analytics:**
  - Top Capacity Consumers
  - Configured Services
  - Configured Groups
- **Predictive Insights:**

  - Predictions page
- **Performance Analytics**
  - Performance Analytics page.

For more information, see the OI Universe section.

Role-Based Access Privileges Enhancements

Universe Management Access Privileges

The Create and Edit Role pages include the following universe management -related changes:

- The **DX Platform > Settings** section includes the **Universe Management** access to define and manage data access permissions for users.
- By default, a user with the out-of-the-box role can access all the DX Operational Intelligence data in the tenant. However, for a user with a custom role, a Tenant Administrator or a Security Administrator must add this access specifically under **Universe Management**.

Alarm Analytics Access Privileges

Starting with this release, for Raw alarms, the role permissions granularity has been enhanced. A user with a custom role can now provide privileges to the following alarm actions:

- Clear
- Acknowledge/Un-Acknowledge
- Ticketing
- Assign/Unassign
- Hide/Unhide
- Channel

For more information, see the Alarm Analytics Access Privileges section.

Ticket Enrichment Rules

A Ticket Enrichment Rule enables you to map one or more alarm attributes to one or more incident fields so that you can assign incidents to the right queues, better prioritize, and enrich incidents with more details for a quicker resolution to problems. To map the attributes with the incident fields, create a mapping rule on the **Settings > Ticket Enrichment Rules** page and associate that rule with a channel or policy. You can create a mapping rule for all alarms, service alarms, or situations.

For more information, see the Ticket Enrichment Rules section.

Service Analytics Enhancements

SLO Calendar Support

Make the SLO performance indicators of your service easier to relate to with the SLO window calendar support. Service Level Objective compliance and error budget metrics can now be calculated on a defined calendar window (week, month, quarter, and year). The Calendar window provides a consistent boundary for the metric calculation and period comparison. The SLO metrics can be calculated based on week, month, quarter, and year.

For more information, see Service Level Objective (SLO) section.

Added Anomalous Information on Services Chart

As a user of DX Operational Intelligence, you can take advantage of anomaly metric calculations and anomaly-based alerting for service level indicators. Configure an anomaly threshold for the service level indicator to be alerted when the performance is beyond an expected range. You can now view the service level indicator performance along with the

calculated anomaly metric bands on the service level indicator charts to compare the service level performance to the historical baseline.

For more information, see the Services Details section.

Enhanced SLI and SLO Creation

Uplevel your service monitoring with the enhanced service level indicator user interface. Service owners and administrators can now leverage a completely updated user experience to define service level indicators based on multiple metric filters, intermediate derived metrics, and arithmetic expressions. Understand the performance of your service based on metrics that match your business and technical requirements.

For more information, see the Create SLI and SLO section.

Alarm Analytics Enhancements

Column Customization and Personalization Support

With this update, you can extensively customize and preserve customization that is made to the alarm analytics view to improve efficiency across your alarm operations teams. Solve problems faster with the layout customization that matches your personalized environment needs. Select up to 15 alarm attribute columns and preserve column settings on the Alarm Analytics pages (All Alarms and Situations). Customize the alarm attribute columns, such as Alarm type, Severity, Service, Source, and Cause Code. Save the page filter, column sort, column selection, and other page personalization options to a new alarm queue for rapid reuse. Administrators, use this feature to customize the alarm console to match your operator's needs.

For more information, see the All Alarms, Situations sections.

Performance Analytics Enhancements

Anomaly Detection Improvement

When viewing the actual value for an anomaly instance in the chart data, you can now also see the minimum, maximum, and count for the given frequency. You can use this to understand the anomalies that are triggered on the individual samples within the given frequency.

Maintenance Windows Enhancement

In this update, users of Operational Intelligence can define maintenance windows by the metric path of the metric-based alarm. An administrator can set up a maintenance window using a regular expression to select the appropriate alarms by the alarm metric and metric path. The combination of alarm metric path and regular expression provides users with a powerful but elegant solution to apply maintenance windows to a precise set of alarms that may have been difficult to filter by the entity or CI relationship. An example would be setting up a maintenance window on a specific MQ or Kafka queue.

For more information, see the Maintenance Window section.

Dashboard Enhancements

Starting this release, DX Dashboards includes the following new features and enhancements:

- Data Access Permissions for Dashboards Using Universe
- AIOps_Inventory Data Source Enhancements
- Time Zone Honored in Reports and PDF Generated
- New OOTB Dashboards
- UMA Dashboard Enhancements

> **NOTE**
> For more information, see the DX Dashboards documentation.

Vulnerability Fixes

- Upgraded Tomee 8.0.11 in all the components.
- Upgraded zookeeper server to v3.8.0J.
- Upgraded Kafka server and clients to v3.2.0.

Known Issues

- **DE544221:** The SLI metrics with an operator other than Expression on the intermediate forward metric do not calculate correct values.
- **DE544096:** The SLI UI does not allow you to use intermediate metrics without first having a configured initial SLI that is published.
- **DE543045:** The sample displays the child services even if the parent has no access.

Fixed Defects

The following issues have been fixed:

- **DE524420:** Accessing **Monitored Inventory** displays error **410: /oi/v3/api/inventory/_search**.
- **DE520467:** Alarm Analytics (Ticket Management) - Provide more details for an error pop-up failure.
- **DE550931:** Situation Alarms Action APIs Not Working

22.06 Release Notes

The section lists the new, and enhanced features in DX Operational Intelligence 22.06

Alarm Analytics Enhancements

The following enhancements are made on the Alarm Analytics page:

- **Customize Columns:** The All Alarms and Situations view enables you to customize the list of alarm columns, such as Alarm type, Severity, Service, Source, and so on. You can pin and save the customized column to view the required information for alarms. You can sort the customized columns, apply a filter and save the view as a queue.
- **Clickable Queues:** You can view the complete list of Queues by clicking icon. The alarm queues are now clickable. You can click a queue or alarm count to view the queue details with the filter applied.
- **All Alarm Details - Overview tab:** A new property Alarm Attributes is added in the Alarm Details section of the Overview tab. You can click **Show Raw JSON** to view the alarm attributes for a particular alarm. You can also have an option to copy the JSON.

or more information, see All Alarms and Situations Alarms.

22.05 Release Notes

This release includes the following new features enhancements, fixed defects, and known issues:

- Situation Custom Definitions Enhancements
- Performance Analytics Enhancements
- RESTMon 2.2 Download From Settings Page
- Dependent Features Auto Selected
- DX Dashboards Enhancements
- Known Issues

Situation Custom Definitions Enhancements

This release includes the following enhancements:

- Added Stabilization TimeLine Options
- Added Service Hierarchy Option in Alarm Clustering Criteria
- Support for Out-of-the-Box Custom Situation Definitions
- Customize Alarm Clustering Criteria Field
- SLI Validation for Complex Aggregations and Metric Config APIs for Handling Multiple SLI/SLO

Added Stabilization TimeLine Options

You can now use the following stabilization timeline options to specify the situation stabilization criteria for situation clustering:

- **Stabilization Time:** Specifies the time period for situations to be stabilized before starting new clusters.
- **Auto Extend:** Specifies the time period to extend the situation clustering alerts before starting new clusters. This option extends the total clustering time until the Max Stabilization Time is reached when the new alerts are generated. You can use this time period with the Max Stabilization Time to ensure that continues to cluster alerts together that are related to the same failure. You can apply this option for new related alerts and not to existing alerts that are updated with the new events.
- **Max Stabilization Time:** Specifies the maximum time period that clusters alert before starting a new cluster.

For more information, see the Define Situation Stabilization Criteria section.

Added Service Hierarchy Option in Alarm Clustering Criteria

Starting with this release, the Service Hierarchy option is added in the Alarm Clustering Criteria. This option clubs the child services with the parent service for alarm clustering. For the existing services, the include child service option is applicable only after stabilizing the existing clusters.

For more information, see the Define Alarm Clustering Criteria section.

Support for Out-of-the-Box Custom Situation Definitions

DX Operational Intelligence now supports the following out-of-the-box rule-based clustering that allows you to use the default templates for creating the custom situation definitions.

- ServiceOnly
- Spectrum BGP Critical
- UIM ROBOT Critical

You can customize these templates based on your alerting requirements.

For more information, see the Out-of-the-box Custom Situation Defintions section.

Customize Alarm Clustering Criteria Field

DX Operational Intelligence now supports clustering Alarms by Severity and Product fields. Use these two additional fields in the definition of the custom situation and the existing Entity, Message, and Service fields.

You can specify the string start for the cluster comparison using the ‘Start at’ position option when clustering by Entity. This clustering criteria field is useful for APM entities as the APM entities have embedded separators in the entity definition.

For more information, see the Define Alarm Clustering Criteria section.

SLI Validation for Complex Aggregations and Metric Config APIs for Handling Multiple SLI/SLO

Starting with this release, you can create Service Level Indicators based upon complex arithmetic expressions and conditional logic statements. In addition to improving the usability of the APIs for building powerful Service Indicators, this feature also allows the creation of multiple derived SLIs per metric definition. If You want to try out Service Level Indicators and Service Level Objectives with DX Operational Intelligence, contact Broadcom Support.

Performance Analytics Enhancements

- **Metric Browser shows Entities based on Groups:**DX Operational Intelligence now lists the metrics based on Groups in the Metric Browser. You can expand any group to view the greyed-out entities (without metrics).
- **Added *Layer* as Filter Attribute:** A new filter attribute *Layer* is added to narrow down the entities, metrics, and CIs based on the Application, Infrastructure, Network, and Service layers.

RESTMon 2.2 Download From Settings Page

You can now download the latest RESTMon version - 2.2 directly from the Settings page in DX SaaS. The package contains the artifacts for deployment on orchestration platforms, PCs, or VMs.

Dependent Features Auto Selected

Starting this release, when creating or editing a custom role, selecting a feature auto selects all the dependent features. You may review the granted access before you save the selection.

DX Dashboards Enhancements

This release of DX Dashboards includes the following changes:

- RESTMon Self-Monitoring Dashboards Enhancements
- UMA Dashboards Enhancements

RESTMon Self-Monitoring Dashboards Enhancements

The following RESTMon dashboards have been enhanced:

- **RESTMon: Monitoring Overview Dashboard**: The Health section includes the Liveness Heartbeat and Readiness Heartbeat visualizations.
- **RESTMon: Publisher Dashboard**: This Alarms section in the Publisher dashboard includes the following new visualizations:
  - Published New Alarms Per Interval
  - Published Updated Alarms Per Interval
  - Published Closed Alarms Per Interval
  - Failed Published Alarms Per Interval
  - Failed Published Alarms Per Interval As OI Unavailable
- **RESTMon Profile Handler Dashboard:** This dashboard includes a new section named Streaming Queues. This section displays the following visualizations:
  - Pending Messages in Streaming Queue
  - Pending Messages in Streaming Queue

UMA Dashboards Enhancements

The UMA dashboards now include the Events visualization.

- Added a new visualization named **Events** to all the dashboards. This visualization provides information about the events such as kind of event, name of the event, severity, reason, and so on.
- **UMA Cluster Information** Dashboard: Added the **Failed Pods by Namespace** visualization to the **Container Information** section.
- **UMA Namespace/Project Information** Dashboard: Added the **Health** visualization to the **Deployment** and **DaemonSet** sections of this dashboard. These visualization display the health of the deployments and daemonset respectively. Green (1) indicates healthy and Red (0) indicates not healthy.
- **UMA Multi-cluster Information** Dashboard: **Top 5 CPU Utilization Namespaces/Projects** and **Top 5 Memory Utilization Namespaces/Projects** visualizations renamed.

Known Issues

- **SLI/SLO:**
  - You see a sudden dip in error budget for a few data points, and min or max values appears incorrect for comparator metrics.
  - If an SLI of type*Availability* is associated with the Availability of any service and when the SLI type is changed to any other type (other than Availability), the service to SLI association is not removed.

    As a workaround, to remove the association, trigger the service update.
- When more than 50 concurrent users access DX Dashboards with 15-seconds auto-refresh interval, the requests are failing.
- The DX Dashboards session is timing out after 24 hours even though the auto-refresh mode is enabled.

22.04 Release Notes

This section describes the new features, enhancements, fixed defects, and known issues in this rollout:

- Service Creation Enhancement
- Support for Lifecycle Event for Maintenance Alarms
- Improved Capacity Monitoring and Capacity Forecasting Capabilities in Capacity Analytics
- RESTMon: New Version RESTMon 2.2 Released
- Associate Policy with Message Template on Policy Page
- Channel Page Changes
- Message Template Page Changes
- DX Dashboards

Service Creation Enhancement

Starting with this release, the Service Creation flow has been enhanced with the following changes:

- The **Manage all elements** button is replaced by **Add Service** and **Add Shared Service** buttons.
- Clicking the **Create Service** button redirects you to an empty layout screen with the following service creation options:
  - **Add Service:** Use this option to create a service. This option adds an empty group service to the layout. You can set the service properties from the **Service Details** panel. To add entities to a service, click the **Filter** button on the Service Details panel.
  - **Add Shared Service:** Use this option to create a service from the existing shared service. This option allows you to search and select an existing service to add. Clicking apply adds the service and any children it might have.
- You can now save a service without entities.
- Removed restriction on service weights. You can save a service even if the weight of child services is not equal to 100%.

For more information, see the Create a Service section.

Support for Lifecycle Event for Maintenance Alarms

Starting with this release, a life cycle event is added for the alarms that are suppressed due to the maintenance window. You can view lifecycle information for an alarm such as the reason for alarm suppression and details from the Maintenance Window definition. For more information, see the All Alarms section.

Improved Capacity Monitoring and Capacity Forecasting Capabilities in Capacity Analytics

Starting from this release, DX Operational Intelligence provides improved capacity monitoring and capacity forecasting capabilities in Capacity Analytics:

- A new algorithm is implemented in Capacity Analytics which identifies the seasonality patterns in the previous year's data for available metrics while generating the forecast projections. The new algorithm can identify the seasonality patterns in the metric data for projection calculations at device, service, subservice, group, and subgroups levels.
- Capacity Analytics now shows the actual data along with the forecast projections for a given metric that is plotted on the same graph using different colors and legends. Capacity Planners can seamlessly compare the actual vs. forecast data and make informed decisions.
- Capacity Analytics now supports the capacity forecast at hourly, daily, weekly, monthly, and half-yearly granularity based on the identified patterns. Capacity planners can make informed decisions with short, medium, and long-term forecasts.

RESTMon: New Version RESTMon 2.2 Released

RESTMon 2.2 is the latest version and includes the following enhancements and fixes:

- RESTMon Performance Improvement and Memory Optimization
- Simplify Third-Party Metric Paths and Display on Performance Analytics
- Polling Interval by Data Type for Polling Integrations
- Token Authentication Support for Streaming Integrations
- SCOM 2019 UR1 Support
- Improved Alarm & Topology State Management
- Alarms Auto-Retry
- Dynatrace Integration using Webhooks
- Deprecated Features in RESTMon 2.2 Release

RESTMon Performance Improvement and Memory Optimization

Starting with this release, RESTMon users will notice the following performance improvements:

- Significant performance improvement (up to 4X) in the Alarm Ingestion process
- Reduction (up to 4X) in Memory Consumption
- Reduction (up to 3.5X) in CPU Utilization

Simplify Third-Party Metric Paths and Display on Performance Analytics

Starting with this release, RESTMon enables the Schema developers to add meaningful and readable display names and define a standard default path for third-party metrics: `Host|Metric Type|CI:Metric` . As a result, the users can now access the metrics seamlessly in the Performance Analytics UI that uses metrics metadata for the navigational hierarchy, thus improving the user experience. Schema Developers can also use the field “Display_Name” to build custom paths.

This enhancement is applicable to both existing and new integrations. Earlier, the user had to click multiple Ids and CI types for the navigation hierarchy in the Performance Analytics UI.

Polling Interval by Data Type for Polling Integrations

Starting with this release, RESTMon supports Dynamic polling intervals for PULL schemas to reduce the load on monitoring tools with optimized GET calls and improve the overall stability and performance of the data ingestion process. Dynamic Polling Interval enables the polling of different data at configurable intervals.

For example, Administrators can define different polling intervals for alarm, metric, and topology data ingestion. The Polling Interval also ensures that the next polling cycle starts when the data ingestion in the previous polling cycle is complete.

Earlier, the polling happened at a fixed time interval irrespective of the last polling cycle status. This resulted in unnecessary load and performance issues during the data ingestion process. For more information, see the Profile section.

Token Authentication Support for Streaming Integrations

RESTMon supports Token authentication, authorization, and secured data ingestion for streaming integrations from this release onwards. The administrators can generate the bearer tokens in DX Operational Intelligence. With this security enhancement, the administrators need not maintain separate authentication credentials at the RESTMon layer. For more information, see Configure Application Properties.

SCOM 2019 UR1 Support

RESTMon is enhanced to support SCOM 2019 UR1 Token authentication to prevent Cross-Site Request Forgery attacks. Schema Designers can generate tokens and can post back as a cookie to the subsequent calls.

Improved Alarm & Topology State Management

RESTMon provides improved State Management capabilities for Alarm and Topology ingested from third-party tools: Ensures that the Topology ingested through streaming integration does not expire abruptly with TTL expiry. Automatically manages the amount of data that is loaded into the cache so that the data always fits within the available memory, Calculates the difference between ingested Alarms and Topology at runtime.

Alarms Auto-Retry

From this release onwards, RESTMon supports the Alarms Auto-Retry option when the alarm ingestion process encounters errors. RESTMon auto-retries multiple times (default: three times) with an increasing interval counter to complete the alarm ingestion process when the alarm ingestion process returns an error code. This feature ensures that the alarm data is not lost during the ingestion process and provides complete, up-to-date information to the users to take necessary actions.

Dynatrace Integration using Webhooks

Starting with this release RESTMon supports the integration of Dynatrace-Webhooks Streaming Schema for third-party Alarm data ingestion.

Deprecated Features in RESTMon 2.2 Release

Log Parsing and Text Matching Capabilities

From RESTMon 2.2 release onwards, the Log Parsing and Text Matching Capabilities are deprecated. We recommend you to evaluate DX NetOps Insights - A Log Analytics offering for log-based alarm generation and log parsing.

Associate Policy with Message Template on Policy Page

Starting this release, you can associate a policy with both a notification channel and a message template on the policy page. This message template is used in the notification when the policy is met and a notification is triggered automatically.

To associate a message template, select the notification channel, and then select the required message template. The selected message template is displayed under the Linked Message Templates section on the associated Channels page. In the earlier version, you could only select the notification channel.

NOTE
When you select Webhook or Slack as the channel for a policy, the option to select a message template is disabled.

Channel Page Changes

Now that you can associate a policy with a channel on the Policy page, the Channels page includes the following changes:

- The option to associate a channel with a policy is no longer available on the Channels page. You can associate only with a message template. In the earlier version, you could associate a channel with both the message template and the policy.

  NOTE
  To associate a channel with a policy, navigate to the Policy page.
- Linked Message Templates and Linked Policies sections added. These sections display the message templates and policies that are associated with the channel. Click the link to navigate to the corresponding message template or policy.

Message Template Page Changes

The message templates include the following changes:

- The **Linked Notification Channels** label on the Message Templates pages is renamed to **Linked Channels**. This section now lists all the associated channels as links. Click the channel link to navigate to that channel edit page.

  NOTE
  You cannot associate or disassociate a channel on the message templates page. In the earlier version, you could associate or disassociate a channel.
- Linked Policies section added. This section displays all the policies that are associated with this message template. Click the policy link to navigate to that policy page.

DX Dashboards

This release of DX Dashboards includes the following enhancements:

- Enhancements to the Capacity Analytics Dashboards
- Dashboard Sharing Enhancement

For more information, see the DX Dashboards documentation.

22.03 Release Notes

The section lists the new, changed, and deprecated features in DX Operational Intelligence 22.03

Performance Analytics

Support for Show Entities Without Metrics

The Metric Browser panel on the Performance Analytics page includes Show Entities Without Metrics toggle switch. You can enable this toggle switch to show the entities without metrics. By default, you can view all the entities with and without metrics For more information, see the Performance Analytics.

Added Unit on the y-Axis of Metric Chart

Starting with this release, you can view the Unit of the performance data on the y-Axis of the metrics chart. The performance data value is rounded and displayed as a Unit on the y-axis of the Metric Chart for better UI readability. For more information, see Performance Analytics.

Service Analytics

Show Expired Topology Vertices

Starting with this release, you can now view the expired vertices on the Topology page. You can select the days ranging from one day to one week to view the expired vertices and the expired vertices appear in gray color with the expired date. For more information, see View Topology Details.

Added SLI/SLO Data in the Service Overview Page

The Service Overview page now includes the widget components for SLI and SLO. These widgets summarize the state of the services and act as a filter on the services. This information provides the health of the services using metrics. The Services Overview page includes the following enhancements:

- **All Services View:** The All Services view is the default view. You can customize this default view and can save it as a new view.
- **Manage Views:** Use this view to perform the following:
  - Save the existing view.
  - Save as to save a new view.
  - Reset to revert the changes to the last saved state.
  - Delete to delete the view tab permanently.
- **KPIs Pie Chart:** The KPIs Charts section is configurable and can display up to six KPIs charts on the Services Overview page. The following KPIs charts have been added:
  - **SLI State**
  - **SLO Error Budget**
- **SLI States and SLO States Columns:** Provides cumulative state of SLIs and SLOs associated with the services.

For more information, see the Services Overview page.

Chart Preview for a Metric on SLI Metrics List

Starting with this release, you can view the metric chart preview for an SLI on the Service Level Indicator metric list page. For more information, see Create SLIs and SLOs. For more information, see Create SLIs and SLOs.

Support Automatic Anomaly Detection for SLIs

As a service owner and user of DX Operational Intelligence, you can view the anomaly information for the configured SLI using the Anomaly Detection option. For more information, see Create SLIs and SLOs.

Enhancement to the Layer Attribute on the Topology Details Page

Starting with this release, the layer sets to the first selected attribute after selecting Group by value on the Topology details page. For example, If you select a Network Layer and then select Group by value, the layer is retained to the Network layer as this is the first selected attribute. For more information, see Topology Details.

Added Topology Correlation Settings Access Privileges

As a tools administrator, you can configure the compaction or association rules of the topology through the Topology Correlation Setting Access Privileges page. For more information, see Topology Correlation Settings Access Privileges.

Role Management

Copy Custom Role

You can now make a copy of a custom role. When you copy a custom role, the copied role inherits all the privileges of the base custom role. You can add or remove the privileges as required.

Follow these steps:

1. Open the Roles page.
2. Click the Ellipsis icon for the custom role that you want to make a copy.

3. Select Make a Copy.

   NOTE

   Note: Alternatively, you can click Edit to open the custom role and then click Make a Copy.

   The New Role page is displayed.
4. Edit the role as required.
5. Click Create.

This new custom role is added to the roles list.

Alarm Analytics

Support Configuration for Deviation Anomalies

As a Service Owner, you can configure the percentage deviations for anomaly alarm. This feature reduces noise by configuring the percentage deviations or static threshold values. This configuration triggers an alarm for any minor deviation in the metric data from the band identified by the algorithm based on the historical usage. For more information, see Configure Alarms section in Configure Monitoring.

Maintenance Window

Support for Duplicate Maintenance Window

Starting with this release, you can duplicate the existing maintenance window by clicking the Duplicate button on the Set Maintenance Window page. This feature allows you to speed up maintenance windows creation by leveraging existing scheduled maintenance definitions, once a definition has been duplicated, you must update key settings like maintenance name, dates, and content. For more information, see Maintenance Window.

Manage Active Maintenance Windows

Starting with this release, you can update the existing active maintenance window by adding or removing entities and editing the end time of the maintenance window. For more information, see Maintenance Window.

Added Drop-down Filter Descriptions While Creating Maintenance Windows

Starting with this release, you can further filter by descriptions for the filtered attributes while creating a maintenance window. For more information, see Maintenance Window.

Situation Alarms

Support for Topology tab Within Situation Alarm Template

Triage a situation alarm faster with the added context of topology in the situation alarm. As an Operations Engineer, you can now view the topology for the entity associated with the Situation alarm template. Use the hops selector to see the connected topology graph based upon the distance (hops) from the situation entity. For more information, see Situation Alarms.

Added Impacted Entity(s) Attribute in Overview Tab of Situation Details Page

Starting with this release, a new attribute “Impacted Entity(s)” for Spectrum Alarms has been added in the Overview Tab of the Situation Details Page. For more information, see Situation Alarms.

Custom Situation Definitions for Situation Clustering

DX Operational Intelligence now supports defining custom rules for situation clustering using Custom Situation Definitions.

- The existing situation clustering based on algorithm-based global policies restricts the organizations from adding organizational-specific rules. With Custom Situations Definitions, the Administrators can define the predictable rules that meet the organization-specific business objectives.
- The turnaround time for ticket triaging has also improved as ticket creation takes less than five (5) minutes for custom rules-driven situations.

Support for Lifecycle Events tab for Situations

DX Operational Intelligence provides a Lifecycle Events tab to view the changes done on situations such as Time of Situation creation, Situation Acknowledgement, Situation Assignment, Ticket Creation, Annotation, Status Changes, and Situation Closure. The Operation Engineers can now view and understand the actions performed on situations that are mandatory tasks as part of Operational visibility.

New Bi-directional ITSM Integration with WolkenSoft

DX Operational Intelligence now supports bi-directional ITSM integration with Wolkensoft. The new integration enables users to:

- Create automated or manual incidents in Wolkensoft.
- Keep incidents and alarms in sync with bi-directional integrations.
- Enrich the WolkenSoft tickets with optional WolkenSoft CMDB lookup.

For more information, see Integration with WolkenSoft.

DX Dashboards

This release of DX Dashboards includes the following new dashboards and enhancements:

- **New OOTB AXA Dashboards:** The App Sessions Events folder includes the out-of-the-box dashboards for App Experience Analytics.
- **Shared Dashboard Link Enhanced:** Any dashboard that is shared through a link automatically redirects to the dashboard if the user is logged into the tenant. However, if the user is not logged in, the shared link now opens the login page for that tenant. After entering the username and password, the user is redirected to the dashboard.

In the earlier version, you were required to enter the Tenant ID as well.

Use Existing Metrics Values for Threshold

You can now use existing metric values for their threshold definitions instead of using static values. For example, if the maximum log ingestion limit for the day has changed from 10 GB to 15 GB, using the query variable, you can have the new value dynamically updated and displayed on the panel. To do that, you would create a query variable with the AIOps_Metrics data source and metric value and use that query variable as the threshold value.

For more information, see the DX Dashboards documentation.

Known Issues

- The error budget and compliance calculations are not correct. This issue occurs when there is a change in the threshold value and when the pod is restarted.
- The error budget decreases after the completion of the compliance window with 1 minute and 5-mins aggregation interval.
- Few SLI data points are missing for a child service though it is coming for a parent service without service content and raw metrics are also generated.
- The service which is under the wrong error budget legend and SLI chart with old data points fail to display when the SLI metric is stopped.

Defects Fixed

The following defect is fixed in this release:

- The filter “Visible=true” did not work on the Service Alarm page.
- The Update button is enabled when no metric is selected. This issue occurs when you associate an SLI availability to service and select ASM or SLI from the Availability dropdown while editing a service.
- Users can now directly click and navigate to APM Metric Viewer to view the alerting metric from the Incident or from the Email or Webhook notifications. This is enabled by a new attribute “Metric View Link” available in the Message Templates and Webhooks functions.

22.02 Release Notes

DX Dashboards Features and Enhancements

This release includes the following new out-of-the-box (OOTB) dashboards and changes:

- **Pivotal Cloud Foundry (PCF) Dashboards:** The PCF dashboards is a new category in the OOTB DX APM dashboards. This category includes the following dashboards and they are available in the General folder:
  - PCF Application Information
  - PCF VM (BOSH) Information
  - PCF Cell Information
  - PCF OrgSpace Information
  - PCF Overview
- **DX APM/AXA: Product Usage Dashboard:** The DX APM/AXA: Product Usage is another new OOTB dashboard that is included in the Health Monitoring folder.
- **UMA Dashboards Changes:** Some of the dashboards and sections within the dashboards have been enhanced.
- **Connector Health:** The **Connector Health** dashboard includes the following enhancements:
  - Renamed the **Overview** section as **Overview of All Products and Connectors**.
  - Added the **Connector Supportability Metrics** section.

For more information, see the DX Dashboards documentation.

Defects Fixed

The following defect is fixed in this release:

- **RBAC Related:** A Tenant Administrator is unable to edit or delete a user when the associated custom role is deleted or deactivated.
  This issue is fixed.

22.01 Release Notes

This section describes the new features enhancements, fixed defects, and known issues in this rollout:

- Role-Based Access Control
- Security Administrator Role Introduced
- Service Analytics
- Alarm Analytics
- Age-Based Alerting Using Policies
- Create Maintenance Window based on Dynamic Grouping
- Token Management
- Connector Parameters Page Changed
- DX Dashboards
- Fixed Defects
- Known Issues

Role-Based Access Control

DX Operational Intelligence uses Role-based Access Control (RBAC) to restrict access based on the roles of the individual users within an enterprise. You can assign users the out-of-the-box roles that have pre-defined access privileges. However, these roles cannot be edited or deleted.

To edit and customize the roles, you can create custom roles and define the access privileges to suit your organization's requirements. After creating the custom roles, you can manage the entire life cycle of the custom roles. For example, you can activate or deactivate a custom role, update or revoke the privileges when required, and also delete the role when you no longer need it.

For more information, see Role-Based Access Control.

Security Administrator Role Introduced

As part of RBAC, the out-of-the-box roles include a new role named Security Admin. This role enables a user to administer security authentication and authorization but does not give the user access to the product features. As a result, this role minimizes a privileged user's ability to access the product functionality and subsequently data that would be difficult to detect even with auditing.

For more information, see the Security Administrator section.

Service Analytics

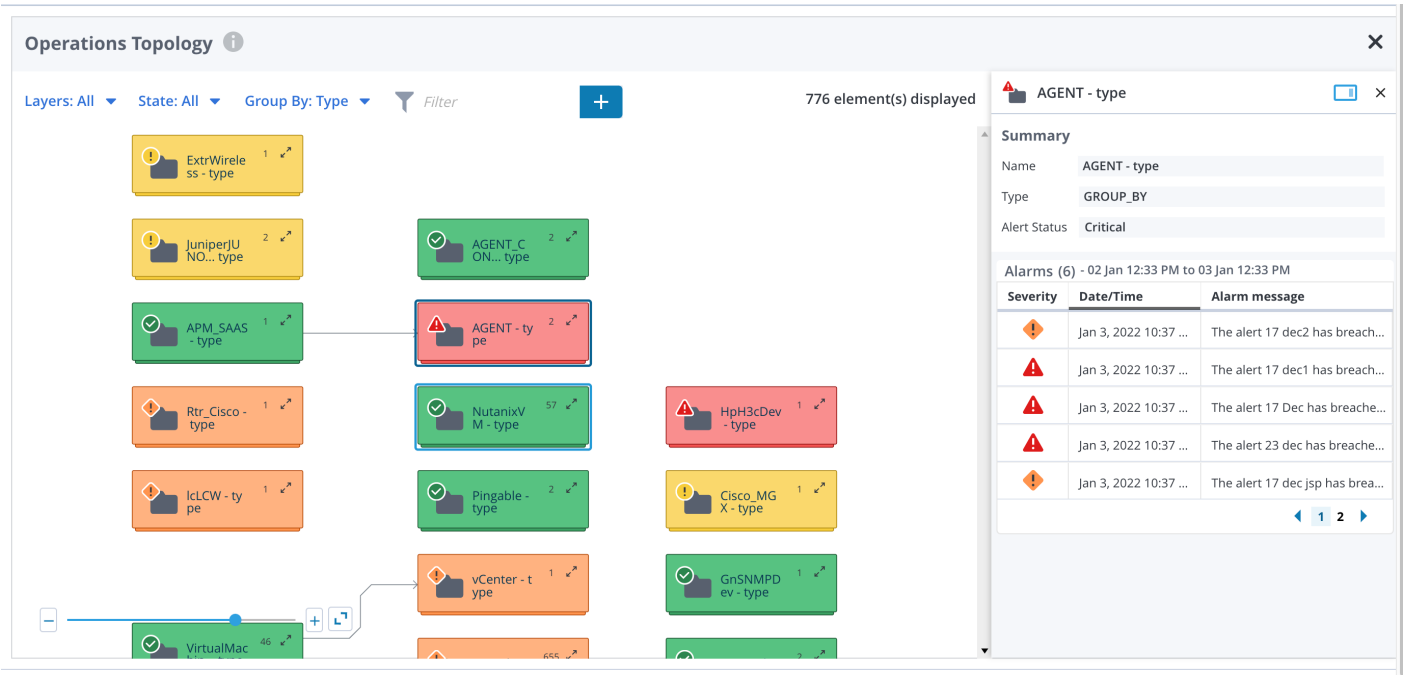
This section includes the following Service Analytics enhancements:

- Topology Groupby Support
- Security Administrator Role Introduced
- Get Services for an Inventory API Support

Topology Groupby Support

Starting with this release, you can now group the operational topology views by attributes. Use this to customize the topology to match with your own perspective of how to best organize infrastructure, application, or network element. Save your customizations and share with other users of Operation Intelligence in your tenant. In addition, the following options are added to the Operational Topology UI.

- **Layers:** Use this option to view the required layer on the topology view screen. If there is only one layer in the view, the Layers option is not available.
- **State:** Use this option to view all vertices or the vertices with alarms. The default is to view all vertices.
- **Group By:** Select the views to be displayed on the topology view screen. The following two groups are available:
  - **Off:** Displays all the elements that are based on the selected Layers.
  - **Default Groupings:** The default groupings are out-of-the-box groups, which cannot be edited or deleted. You can group the topology by the following grouping.
    - Applications
    - Kubernetes Cluster
    - Kubernetes Project
    - Type
  - **Custom Grouping:** Use this option to create and save your own groupings.



For more information, see the View Topology Details section.

Support for Service Level Indicator/Service Level Objective

NOTE
Note: By default, this feature is disabled. To enable this feature for your tenant, contact Broadcom Support.

DX Operational Intelligence supports service level indicators and service level objectives to increase the performance and reliability of the services through metric-driven service-level visibility. Then, you can add Service Level Indicators and apply the Service Level Objective to your existing services by using these options:

- **Derived Metric / Aggregation:**Define the right metric for the right scope. You can filter the metrics by service and aggregate them into metrics.
- **Thresholds with Availability:**Define the thresholds to get insights into the health of their services.
- **SLO Condition:**Provide the condition for SLO conformance and the window to measure it.
- **Error budgets:**The time remaining in the conformance window is the error budget. This metric is shown with the health indicator and is reported in time to know the criticality of an issue.
- **Attribution of conformance issues to root causes:** The SLO calculation provides a mechanism to track the impact of issues on a service in a normalized method (time).

Configure Service Level Indicator (SLI)

Input Metrics

Service

AllUIM, APM12 2 Total

Source

superdomain (Contains)

Output Metrics (SLI)

SLI Name

SLI Type

Select...

Description

Describe SLI's purpose

Aggregation

Select...

Aggregation Interval

Select...

Metric Type

Select...

Unit

Enable SLI Threshold and SLO

SLI Threshold

Undefined

≤

Value

Service Level Objective (SLO)

≥

%

Compliance Window

Rolling

Days

Cancel

Save

For more information, see the Service Level Indicator and Service Level Objective section.

Get Services for an Inventory API Support

Starting with this release, use this API to get the service details for an inventory, including the hierarchy (either parent or child services) along with service metrics.

For more information, see the Get Services for an Inventory section.

Alarm Analytics - Support for Creation of Alarm Queue Based on Status of Alarms and Age

As an Operational Engineer, you can create an alarm queue based on the states of alarms and based on the age of alarms and define the policy to trigger policy-driven notifications or ticketing.

The state of the alarm filter is as follows:

- **Acknowledged:** True, false
- **Assign State:** Assigned, Unassigned
- **Ticket State:** Ticketed, Unticketed

The age of the alarm filter is as follows:

- Greater than equal to
- less than equal to

  NOTE
  You can create a policy but notifications are not triggered when you provide the value as **one** for **less than equal to** filter.
- Created Date Range

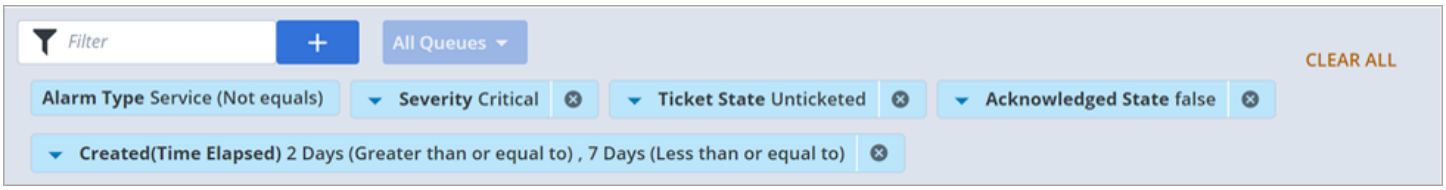
  NOTE
  The Create policy option is disabled for the **Created Date Range** filter.

For more information, see Create an Alarm Queue for State of the Alarms and Trigger Policy-Driven Notifications or Ticketing section in the All Alarms section.

Age-Based Alerting Using Policies

You can now get notified about alarms that were missed or not acted upon by the IT Operations teams for more than a certain period of time. You can use the new filter attribute **Create (Time Elapsed)** to create policies based on the created time of the alarms. You can also use this filter to create policies to automate the workflow of identifying any such old alarms existing in the system.

For example, you can give a time range as shown:



Or, you can give a relative time period:

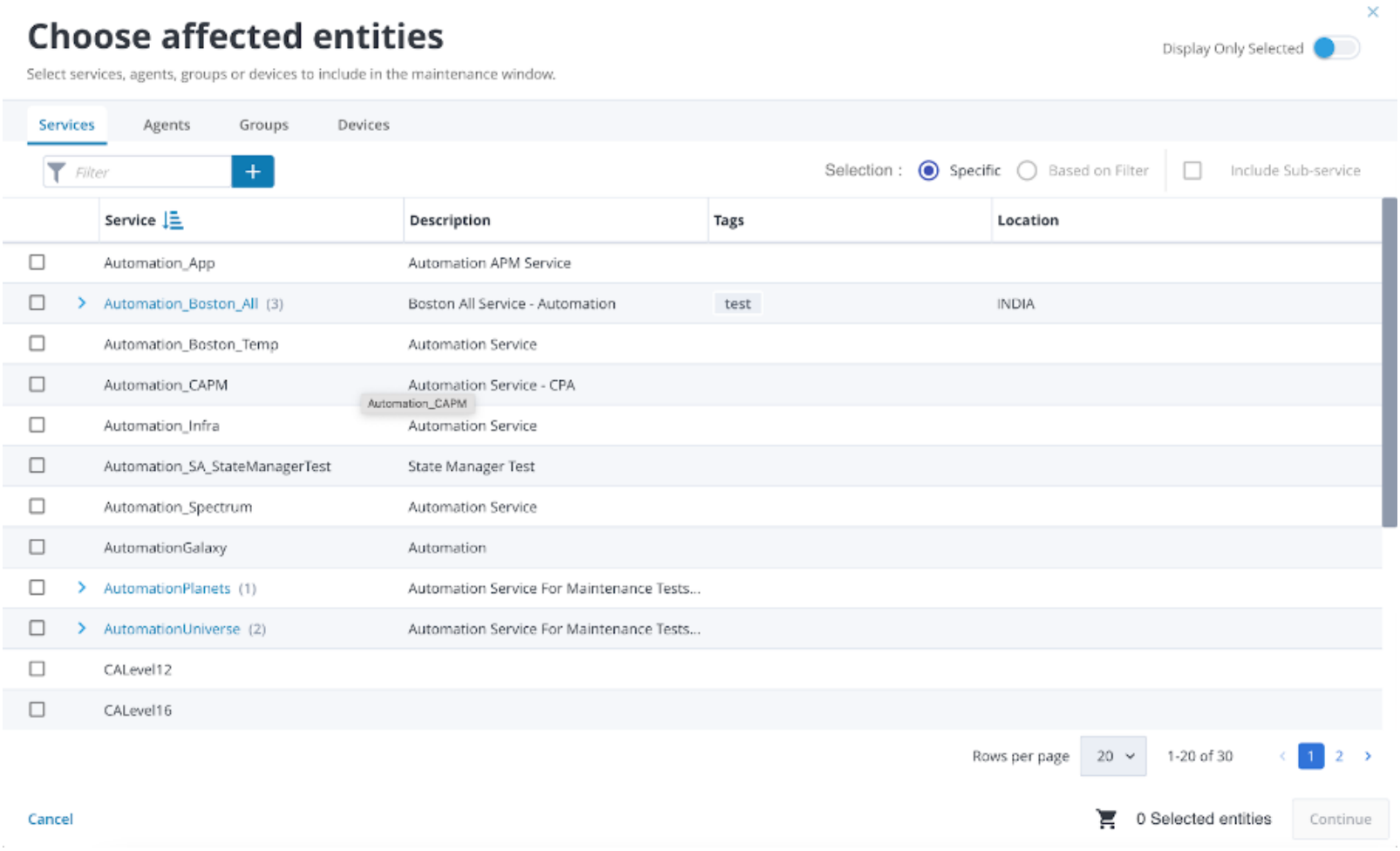
NOTE

- The highest supported value is 365 days.
- 2 Days (Greater than or equal to) signifies: Current time – 2 days
- 7 Days (Less than or equal to) signifies: Current time – 7 days
- Supported time units are: Days, Hours, and Minutes
- Creating age-based policies is not supported for Automic Integration.

Create Maintenance Window Based on Dynamic Grouping

As a System Administrator, you can define the dynamic criteria for selecting agents, applications, or servers on the maintenance windows definition. As a part of this feature, the following options are added on the Maintenance Window page:

- **Specific:** Use this option to select the list of entities with or without the filter applied.
- **Based on Filter, Include Sub-Service:** These options are enabled after the filter is applied. Selecting the Based on Filter option selects all the entities displayed on the page (with the filter criteria applied). Select the Include Sub-Service option to select all the child services.
- **Display Only Selected toggle switch:** Enable this switch to view the selected entities.



For more information, see the Maintenance Window section.

Token Management

The **Settings** page includes the **Tokens** tile to generate Agent tokens, Tenant tokens, and User tokens. A Tenant Administrator can generate an Agent Token, Tenant Token, and User Token. A Security Administrator can generate an Agent Token and a User Token. A Power User and a User can only generate a User token.

For more information, see the Token Management section.

Connector Parameters Page Changed

The option to generate a token is no longer available on the Connector Parameters page. Starting this rollout, you can use the **Tokens** tile to generate a token.

DX Dashboards

This rollout includes the following new out-of-the-box dashboards and changes:

- **APM-IBM ACE Dashboards:**The APM-IBM ACE dashboards is a new category of the out-of-the-box dashboards and includes the following dashboards:
  - IBM ACE Dashboard
  - IBM ACE CICS Resources
  - IBM ACE Independent Server
  - IBM ACE JMS Resources
  - IBM ACE JVM Resources
- **Anomalies Report:** The **Anomalies Report** is another new OOTB dashboard that is included in the Alarm Analytics folder.
- **APM-Mainframe Dashboards:** The APM-Mainframe folder includes the following new OOTB dashboards.
  - DB2 z/OS Buffer Pool Activity
  - DB2 z/OS CPU Activity
  - DB2 z/OS Data Sharing Groups
  - DB2 z/OS Data Sharing Groups Usage and Failures
  - DB2 z/OS EDM Pool Activity
  - DB2 Lock Activity
  - DB2 Log Activity
  - DB2 z/OS More Information
  - DB2 z/OS Subsystem Information
  - DB2 z/OS Workload
- **Alarm/Incident Monitoring Dashboard Changes:** The Alarm/Incident Monitoring dashboard is now available in the Alarm Analytics folder. Earlier this dashboard was available in the Health Monitoring folder.

For more information, see the DX Dashboards documentation.

Fixed Defects

The following defects are fixed in this release:

Monitored Inventory Access Issue (32914470)

Accessing Monitored Inventory causes the following error:

```
410: /oi/v3/api/inventory/_search
```

Known Issues

- The **Affected Metrics** tab is not appearing for UIM performance alarms in the Predictive Insights view.
- **RBAC Related:** A Tenant Administrator is unable to edit or delete a user when the associated custom role is deleted or deactivated.
- **SLI/SLO Issues**
  - The SLI threshold metric (Availability KPI) is visible on the Edit Service page even after you set the SLI key to false.
  - The Update button is enabled when no metric is selected while editing a service.

Release Notes 2021

This section contains the following release notes for 2021.

- 21.12
- 21.11
- 21.10
- 21.09
- 21.08
- 21.07
- 21.06
- 21.05
- 21.04
- 21.03
- 21.02
- 21.01

21.12

The 21.12 release includes the following features, enhancements, and fixed issues:

DX Operational Intelligence Features

Situations Onboarding for New Tenants

Starting with this release, the default alarm reduction experience for new tenants is through the Situation Alarms. For new tenants the Service Alarms is disabled. To re-enable the Service alarms for new tenants, contact Broadcom Support.

However, the existing tenants can see both Situations Alarms and Service Alarms in DX Operational Intelligence UI.

Policy Creation Changes

Starting with this release, the option to create a policy for a service alarm and rootcause alarm is no longer available for new tenants. The default policy is now changed to Situation.

ServiceAlarmNotificationTemplate No Longer Available

The ServiceAlarmNotificationTemplate default message template is not available for new tenants.

DX Dashboards - Features and Enhancements

DX Dashboards includes the following new dashboards and enhancements:

- **New Out-of-the-box Dashboards Added:**

  - **DX OI Summary:** This dashboard provides a summary of the service status, situations, and alarms in DX OI.
  - **RESTMon: Cache:** This dashboard provides information about the cache size, cache weight, cache hits and misses per interval, and so on.
  - **RESTMon: Profile Handler:** This dashboard provides information about the profiles, metrics, alarms, and topology.
- **Dashboards Enhancements:**
  - **RESTMon: Collector Dashboard:** Added the Data Collection Time Per Run (ms) visualization to this dashboard.
  - **RESTMon: Monitoring Overview Dashboard Changes:**
    - Moved the Published Metrics, Alarms, and Topology Data Per Interval visualization to the Traffic category from the Saturation category.
    - Added the following visualizations to the Saturation Category: GC Count, GC Time (ms), Cache Size, and Cache Weight.
  - **Service Details Dashboard:** Added the Total Situations and Latest Situation on Service visualizations to this dashboard.
  - **Service Overview Dashboard:** Added the Worst 5 Services by Situations Count visualization to this dashboard.

For more information, see the DX Dashboards SaaS documentation.

Documentation Changes

Starting this release,

- DX SaaS Documentation Site Decommissioned: The DX SaaS site will no longer be available. You can refer to the DX Operational Intelligence - SaaS documentation for information.
- New DX Dashboards - SaaS Documentation Site: The DX Dashboards SaaS documentation is now available independently as a standalone site.

RESTMon: New Version RESTMon 2.1.6 Released

RESTMon 2.1.6 is the latest version and includes the following enhancements and fixes:

- RESTMon includes the Log4j vulnerability fix.
- The RESTMon performance and memory optimization are improved with the following enhancements while processing the ingestions:
  - The duplicate entries are no longer stored in the HSQL database, thus resulting in the memory optimization and reduction in intermittent data process/published failures.

    NOTE
    Ensure that you delete the existing DB folder before upgrading to the latest RESTMon version(RESTMon 2.1.6).
  - Due to the alarm optimization fixes, the ingestion processing time is improved. The processing time is within the defined polling interval time.
  - The Garbage Collection and Memory leak issues are resolved and have resulted in the memory optimization.
- RESTMon includes the new Solarwinds schema.

Log4j Vulnerabilities Fix

The Log4j Vulnerabilities CVE-2021-44228, CVE-2021-45046 are fixed by upgrading to log4j-2.17.0 as suggested by Apache. In the Log Analytics and Elasticserach components, these vulnerabilities were addressed by deleting the JndiLookup.class file from the log4j package.

Capacity Analytics Enhancements

The Capacity Analytics is enhanced with the following features:

- Capacity Planners can now view the information about the breached configured item (sub-ci) for a given metric in the various Views. In the Device Details page, they can further drill down to analyze the sub-ci level forecasting.
- The metric utilization percentage has an upper limit of 500% for display in the Capacity Analytics overview pages.
- The Subgroups and Subservices tabs are displayed only when the group or service has underlying subgroups or services, respectively.

For more information, see Navigating Capacity Analytics

Predictive Insights Enhancements

The Support personnel can now view the following critical information for the capacity and prediction alarms in Predictive Insights while triaging the issues:

- The metrics information in the context of both prediction and capacity alarms in the Affected Metrics tab.
- The topology for the device in the context of both prediction and capacity alarms in the Topology tab.
- The affected device information in the Overview tab.

For more information, see Predictive Insights User Interface

Defects Fixed

The following defects are fixed in this release:

Issues with the screen refresh rate on Alarms View (32951639)

The alarm table refresh interval is not tenant-specific. By default, the refresh interval is configured for 5 mins for all tenants. Due to this issue, when multiple operators use DX Operational Intelligence for alarm views there is a synchronization issue when displaying alarms. As a solution, you can use the ALARM_REFRESH_INTERVAL environment variable to change the refresh interval per tenant.

Alarm Entity field value is not saved completely when exported to Excel (32928170)

The exported excel sheet does not provide complete cluster information for the Entity field when alerts on multiple unique nodes are in the same cluster.

Time filter disappears from Alarms View (32920193)

The Time filter disappears from the Alarms view when Auto-Update is enabled. This issue has been fixed.

Known Issues

Sorting is not working in the Top consumers section of Capacity Analytics (Firefox browser only)Users may notice the sorting issue in the Top Consumers section when they access Capacity Analytics using the Firefox Browser. The sorting problem is due to React-Table Version 6.0. After we upgrade to the next version of React-Table (Version 7.0), the issue will resolve automatically.

21.11

The 21.11 release includes the following features, enhancements, and fixed issues:

Performance Analytics - Enhancements in Metric Charts View

As an Operations Engineer, you can perform the following enhancements in the Metric Charts view:

- **Edit an Existing Metric Chart View:** You can edit an existing chart card by clicking the

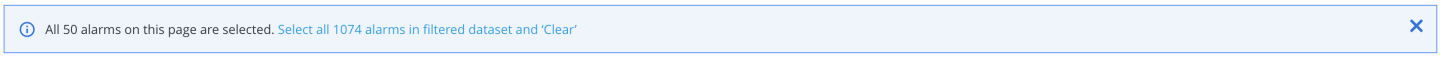
  icon, **Show Metrics.**The tree view appears with the devices and metrics filter applied. You can select or deselect the metrics and save the view.
- **Remove a Metric Chart Card:** You can remove an existing chart card by clicking the

  icon. The associated metric chart is unselected.

For more information, see Show Metricsand Remove Metric Chart Card.

Clear Bulk Alarms in Alarm Analytics

Starting with this release, you can bulk clear the alarms on the Alarm Analytics page. You see the following pop-up to clear all the alarms:



For more information, see*Bulk Clear Alarms* section in Alarm Management.

Process Enhancements for Situation Clustering

DX Operational Intelligence now provides a cleaner maintenance mode feature for situations clustering. When the alarms are in 'maintenance mode,' DX Operational Intelligence discards those alarms and does not create situations or ITSM notifications/tickets. IT Operations Engineers do not get alarms or tickets in maintenance mode.

DX Operational Intelligence is now enhanced to reduce the time taken from the problem occurred till the situation clustering. With this enhancement, the support personnel has enough response time to meet the defined SLOs.

Capacity Analytics User Interface Improvements to manage 20k Services

The Capacity Analytics User Interface has been enhanced to scale and manage 20k services. Capacity Planners can now seamlessly manage the 20k services with the new enhanced User Interface in Capacity Analytics.

- The Configuration pages, widgets and the various views have the following user interface elements:
  - Pagination: A Pagination option to help the users manage 20k services. The users can specify the number of records that they want to view in a single page. They can navigate between the pages using Previous and Next links to view and access the paginated data.
  - Filter: A filter option to define value-based filter criteria using filter attributes, operators and values.
- The Resource Capacity Status widget is renamed as 'Resource Health Chart'. The widget is displays the resource health as a horizontal stacked bar chart, categorized based on the severity.
- The Top Consumers, Monitored Services and Groups Widgets are displayed in separate tabs on the Capacity Analytics home page.

21.10

The 21.10 release includes the following features, enhancements, and fixed issues:

New Features

Support for Alert Queue Sharing Among All Users

Starting with this release, the alert queues can be shared among all the users of a tenant. For more information, see Alert Queues.

Added Impact Legends to Situation Alarms Overview Tab

Starting with this release, the following impact legends are shown beside impacted services and entities on the Overview tab:

- Most impacted
- Initial impact
- In Maintenance

For more information, see Situation Alarms.

Topology Processor APIs

As an API user of DX Operational Intelligence, you can leverage the topology processor API to modify the rules for compaction. You can use this API to understand the following:

- The number of impacted vertexes to a compaction rule.
- Current topology compaction rules.
- Validate the changes without impacting TAS

For more information, see Topology Processor APIs.

Defects Fixed

The following issues are fixed in this release:

User Interface Issues in Operational Topology Widget after Applying the Filter

In the Operational Topology widget, after you apply the filter, you see the following user interface issues:

- Filter criteria issues
- CI box is not aligned with the widget

21.09

New Features

The 21.09 release includes the following features, enhancements and fixed issues:

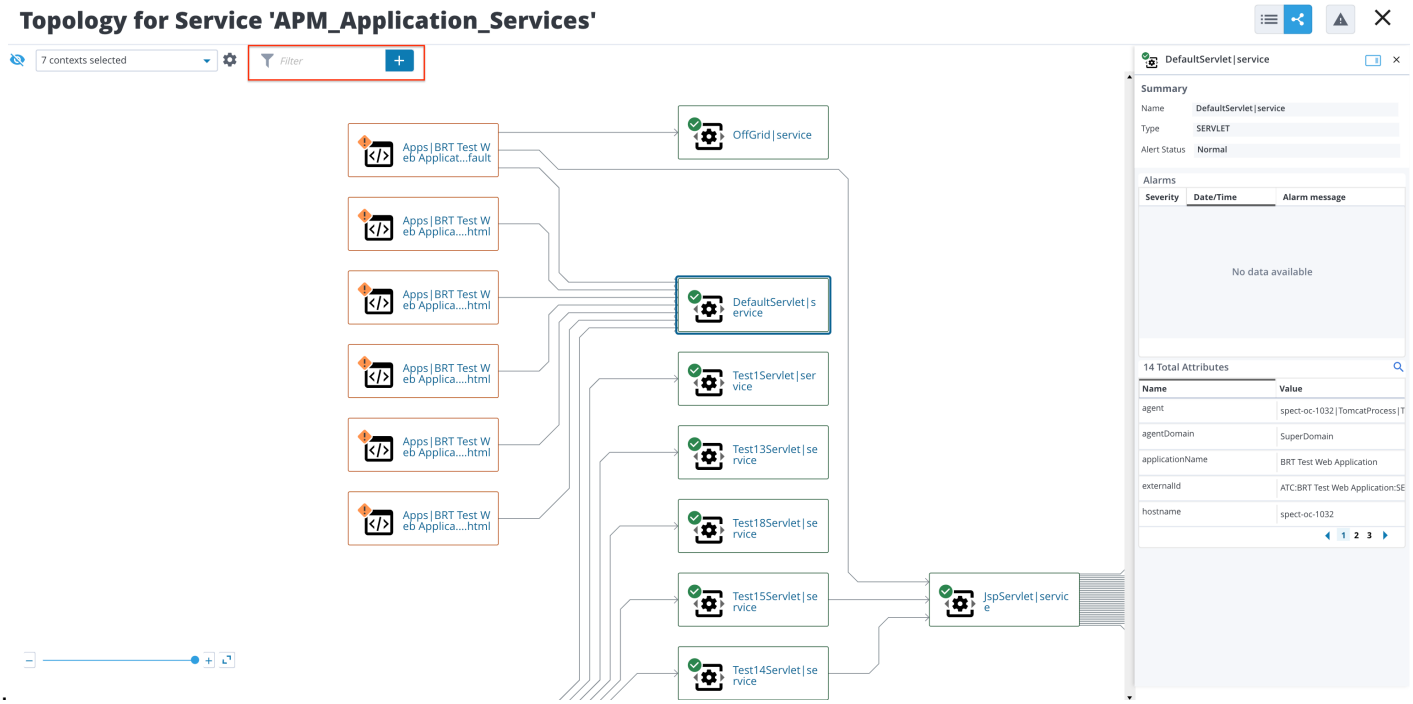
Support for Predictions Widget in Service Details Pag

As an Operational Engineer, you can view the predictions in the context of the service using the Predictions widget. Using this widget, you can understand when an issue might impact the service. For more information, see Predictions Widget

Filter Option on Operational Topology

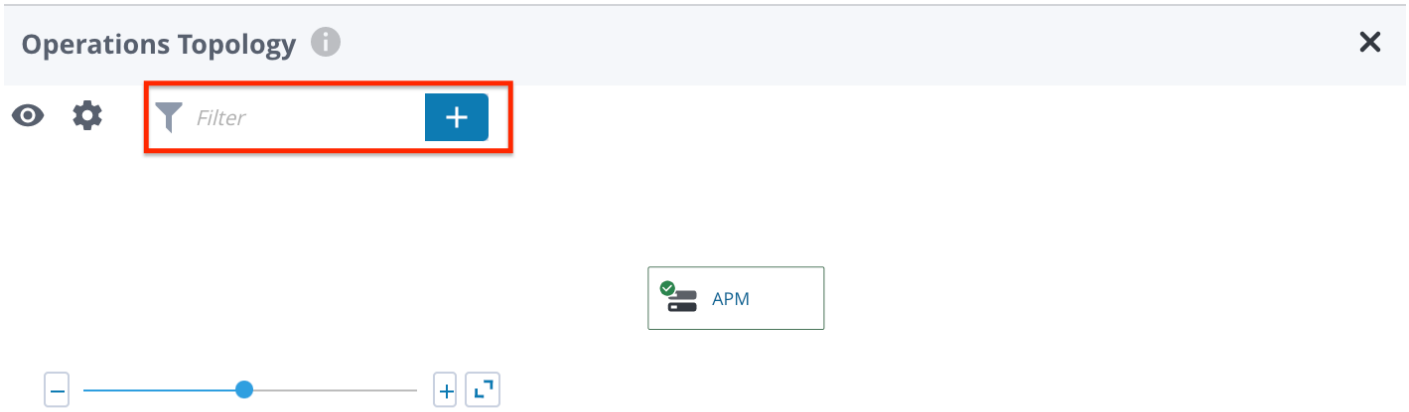
As an Operational Engineer, you can filter the operational topology by searching an entity or set of entities using the filter attributes such as servletclassname, type, resourceType, and so on. The filter option is added at these locations in the Service Analytics User Interface:

- Topology View on the Service Details



page.

- Operations Topology widget on the Service Details



page.

Improved Service Metric Calculation

Starting with this release, the service metrics are calculated for every one-minute frequency. For more information, see Key Performance Indicators for Services.

Updated Tenant Configurations for Situation Alarms

In Situation Clustering Dimension API, the following new tenant configuration parameters are added:

- Alarm Update Consideration Period: Tenant Administrators can configure this parameter for the tenant to define the age of the alarms. The alarms that are within the configured age are considered for situation clustering.
- Single Alarm Association Enablement: Tenant Administrators can set this parameter to ‘true’ to create a situation only when a new alarm is raised. After the situation is stable, DX Operational Intelligence does not create new situations, rather ignores any updates to this alarm.

For more information, see Situation Clustering Dimensions API

Defect Fixes

Elasticsearch limit per Tenant

The performance issue in the Predictive Insights projections deployment is now fixed by optimizing the maximum number of elasticsearch connections per tenant.

Duplicate Alarm ID Issue

The duplicate alarm issue for Prediction alarms is fixed. No duplicate alarm IDs are getting generated. The new alarm is generated and the alarms that are in the 'closed' state having the same external product ID, metric unique ID as the new alarm are not reopened.

The Intermittent Save Issue in Service and Metric Configuration

The issue in the Service and Metric Configuration pages which intermittently was showing error on click of the Save button is now fixed. Users will be able to make all the metric configuration changes by clicking the Save button without any unexpected error pop-ups.

21.08

New Features

The 21.08 release includes the following features and enhancements:

Capacity Analytics User Interface Enhancements to Support 20K Services

The Capacity Analytics User Interface has been enhanced to scale and manage 20k services. The users can access, configure, and view the list of 20K services in Capacity Analytics:

The enhancements made to Capacity Analytics user interface to improve the user experience are as follows:

- A separate configuration page for Services and Groups.
- A Pagination option in the Services Configuration page to help the users manage 20k services. The users can specify the number of service records that they want to view in a single page. They can navigate between the pages using Previous and Next links to view and access the paginated services data.
- A filter option on the Service Configuration page to define filter criteria on Services based on Location, Service Name and tags.

For more information, see Service Configuration.

Improved Service Creation Usability

The Service Creation has been updated to provide a more effective and simplified user experience. The following changes are made on the Service Creation UI:

- More space is dedicated to the CI selection panel.
- Results or cart region is now expandable to show CIs mapped to service.

For more information, see Create Service .

Support for Alarm LifeCycle Event

Operations Engineers can now be able to view the complete lifecycle event of an alarm that occurred from the time it is created. The lifecycle events such as created time, status updates, annotation updates, threshold change. This helps you to track the alarm changes when troubleshooting an alarm. For more information, see Lifecycle Event.

Added Situations Column and Situations Widget in Services page.

Starting with this release, you can leverage the advanced noise reduction and automated root cause capabilities of Situations in context to Services. At a glance within Service Analytics, you can view the open Situation Alerts for all Services and the relationship to the service hierarchy (which child is problematic). You can further drill into a Situation from Service Analytics, view related alarm details, see the probable root cause, and other impacted services. The updates to the Service Analytics User Interface are as follows:

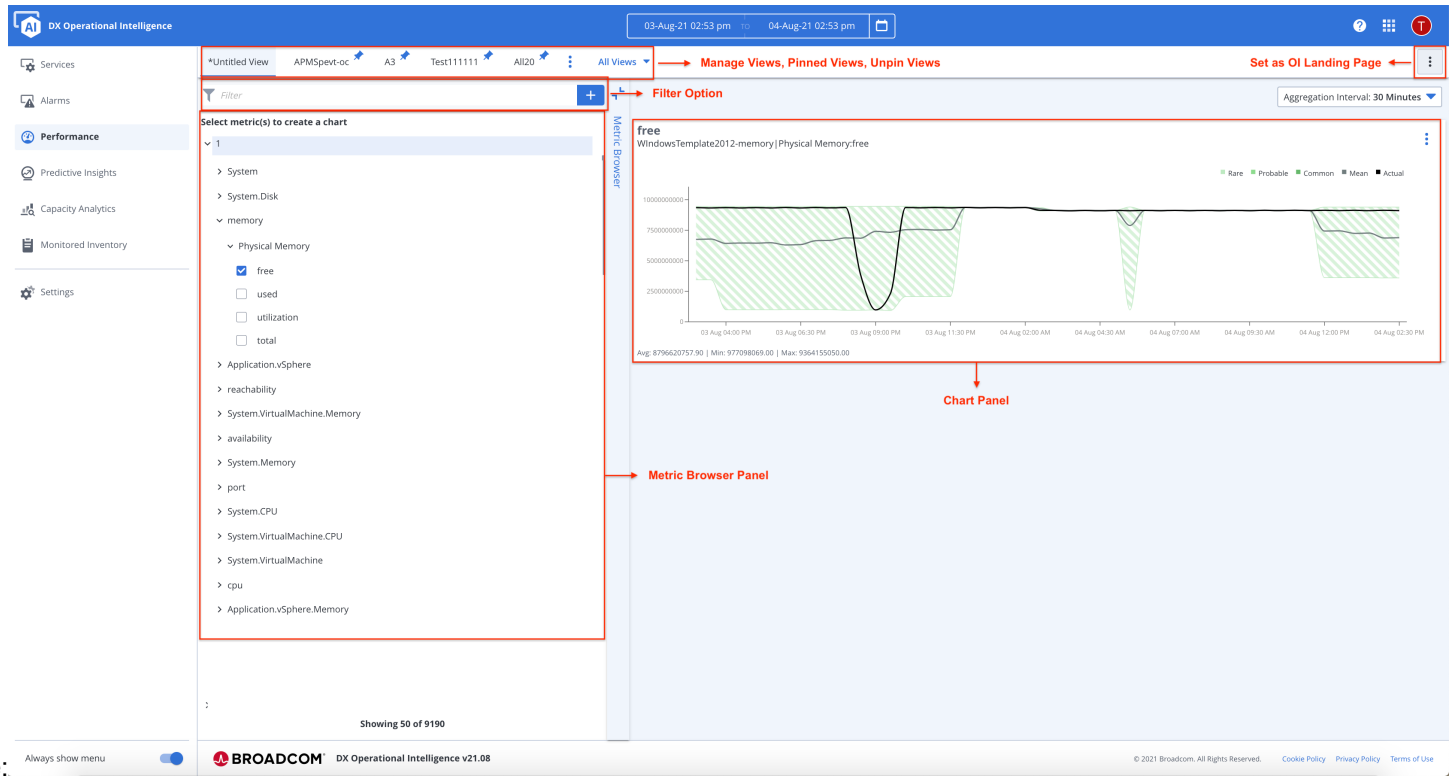
- **Situations (Subs) Column in Services Overview Page:** A new column **Situations (Subs)** is added to the Service Overview page. This column displays the count of the current situations. Clicking the value opens the Situations page that is filtered by a service. For more information, see Service Overview.
- **Situations Widget in Service Details Page:** Using the **Situations widget**, you can view the overview of the current situations alarms that are open for service including noise reduction. You can also launch the Situations page in the context of the service. For more information, see Service Details.

Monitored Inventory - Support for Maintenance Window

You can now schedule a maintenance period to perform maintenance activities on multiple entities in the Monitored Inventory. The maintenance schedule is a period of time that is designated to perform preventive maintenance activities that could cause the disruption of service. The maintenance period stops all the monitoring and metric calculations for the selected entities. You can create maintenance, add entities to the existing maintenance window, and remove entities from the maintenance window. For more information, see Maintenance Window.

Revamped Performance Analytics User Interface

The Performance Analytics user interface is revamped to provide better value and ease of use to end users. The following image illustrates the performance analytics page:



The new user interface provides the following:

- **Tree hierarchy of Entities and Metrics in Metric Browser Panel:** You can view all entities and their related metrics on a single page.
- **Filter:** Use this option to perform a quick search function to find entities and metrics by entering the name.
- **Manage Views:** Save the views, edit a view, and delete the views.
- **Pinned/Unpinned Views:** You can pin and unpin the views up to a maximum of five views.
- **Chart Panel:** The charts appear on the charts panel that is based on the metric selection.
- **Set as OI Landing Page:** Use this option to set the Performance Analytics page as a Landing page.

For more information, see Performance Analytics.

Known Issues

Services Count Mismatch in Capacity Analytics due to SA Cache Update

In the Service Configuration page of Capacity Analytics, the capacity planners may notice that the number of services does not match with the number of services in the Service Analytics page. This issue is due to a delay in updating the Service Analytics Cache (SA Cache). After the SA Cache is updated, the users will have access to all the services in the configuration pages of Capacity Analytics.

Service Hierarchy is not displayed in Monitored Services

In the Service Configuration process of Capacity Analytics, the user may notice a delay when displaying the Service Hierarchy with 28 levels in the Monitored Services section.

21.07

New Features

The DX Operational Intelligence 21.07 release includes the following features and enhancements:

Monitored Inventory - Added Entity Details Panel

Starting with this release, an **Entity Details Panel** is added to the Monitored Inventory page. This panel provides in-context view of selected entities to quickly understand entity details such as alarms, entity attributes, and maintenance. Click the non-hyperlinked area in any row to open the Entity Details panel for the selected entity. For more information, see Monitored Inventory.

Alarm Analytics - Added Created Timestamp Column

A new column Created timestamp is added to the Alarm Analytics page. This column shows the date and time when the alarm is created. For more information, see Alarm Analytics.

RESTMon: New Version RESTMon 2.1 Released

RESTMon 2.1 is the latest version and includes the following features and enhancements:

- **Supportability Metrics Enabled:** Using the supportability metrics, you can view the health of the RESTMon application. To enable the supportability metrics, configure the supportability metrics-related settings or environment variables (Instance Name, Agent Token, and API Endpoint) in the yaml files. Alternatively, you can pass them as arguments during deployment.

  NOTE
  Supportability Metrics is not supported if the proxy is configured between RESTMon and DX OI.
- **Introduced RESTMon Dashboards:** The following dashboards are available out-of-the-box for RESTMon. These dashboards display the supportability metrics. You can find these dashboards in DX Dashboards.
  - **RESTMon: Data Collector**
  - **RESTMon: Datastore**
  - **RESTMon: Monitoring Overview**
  - **RESTMon: Publisher**
- **Support for Liveness and Readiness Check:** You can check the liveness and readiness state of the RESTMon application using the APIs. The Liveness state of the application indicates the status of the application. The Readiness state indicates if the application is ready to accept the traffic or not.
- **New Schemas for Third-party Integrations:** This release of RESTMon supports integration with the following third-party products: AppNeta, DELL EMC ECS, MongoDB, and New Relic.
- **Single Point Ingestion:** You can now define a single endpoint for all data categories to ingest data into DX OI. You can define the endpoint settings in the **oisettings** section of the *restmon.json* file. In the earlier version, the endpoints had to be defined separately for each data category.

  If you have already defined the endpoints separately, you may continue to use the same endpoints.
- **YAML File Changes:**
  - *values.yaml:* The *values.yaml* file includes:
    - The Liveness and Readiness probes-specific configuration for the deployment.yaml file. Kubernetes uses this configuration to restart the pods when the liveness fails.
    - Settings for the supportability metrics and liveness and readiness check.
    - Configuration to enable node selector to deploy RESTMon on specific worker nodes in the Kubernetes cluster. You can configure the **restmon.nodeSelector** setting in the yaml file.
  - *docker-compose.yaml:* The *docker-compose.yaml* file includes new environment variables for supportability metrics, and for liveness and readiness check.

Situation Alarm Action APIs

Alarm analytics now supports a list of action APIs that the automated scripts can call to perform actions on situation alarms. Automation engineers can run these automated scripts manually, through UI, or through ITSM policies to perform these actions.

Using Situation Alarm action APIs in the automation scripts, the engineers can perform the following actions on situation alarms:

- Acknowledge or ‘un’acknowledge a situation
- Assign or ‘un’assign (dissociate) a user for a situation
- Annotate situation with useful notes and comments
- Close or Resolve a Situation
- Create Ticket in ServiceNow for unresolved issues
- Retrieve the situation alarms based on the severity and status that are associated with an alarm cluster

For more information, see Situations Alarm Action APIs.

Optimizing CPA Projection

The performance of CPA projections for the real-time configuration updates is improved. Capacity Analytics is enhanced to process the batch and real-time jobs separately so that the real-time projection calculations are faster.

Monitored Inventory - Sort Entity State by Alarm Severity

Starting with this release, you can now sort the Entity state by alarm severity. Click **Warning Icon**

to sort the entity state in the following order:

- Critical
- Major
- Minor
- Informational
- Normal

For more information, see Monitored Inventory.

Known Issues

Performance Analytics Page Appears Blank.

This issue occurs when you perform the following steps:

1. From the Performance Analytics, click the **Entity** button.
2. Select a device and click **Done**.
3. From the **Metrics** tab, select the same metrics of two separate CIs.
4. Deselect one of the selected metrics.

The Performance Analytics page appears blank.

21.06

New Features

The DX Operational Intelligence 21.06 release includes the following features and enhancements:

Service KPI Projection in Capacity Analytics

Capacity Analytics now supports the capability to provide 12-month projections for the Service level KPIs configured in Service Analytics. With this enhancement, you can have projections for the service level KPIs that are generated by the capacity projection algorithm by correlating the infrastructure metrics that are collected for the service. You can view these projections by creating a custom dashboard in DX Dashboard.

- Service Owners can use the forecast and projections to make performance improvements and plan for scaling up resources to meet SLAs.
- Capacity Planners can use the 12-month forecast and projections to predict the future cost or workload, to determine the required capacity of infrastructure resources. They can also plan for scaling up the resources to ensure operational continuity.

ITSM Configuration support for Active Situations

Alarm Analytics now supports the ITSM ticket handling capabilities to enable and drive alarm workflows around situation clusters both in Active and Stable state. Previously, this capability was enabled only for Stable Situations.

With this enhancement,

- Administrators can configure the ITSM Policies, and Notification Channels for active and stable situations to automate alarm workflows.
- Operations Engineers can perform manual actions such as raising a ticket, assigning the owner, and acknowledging the active and stable situations.

For more information, see Situation Alarms.

Launch Help Documentation from App Bar

Starting with this release, you can now access the DX Operational Intelligence documentation by using the Help icon from the app bar of DX Operational Intelligence User Interface.

Anomaly Alarm Configuration

The Anomaly Alarm Configuration is an intuitive UI that allows you to configure anomaly detection alarms across various metric types for each tenant. Based on the configuration, relevant anomaly alarms with proper context and details are generated on the metrics. Using this configuration, you can:

- Configure the number of anomaly occurrences or the time interval for which the anomaly has to persist to trigger an alarm.
- Set the anomaly alarm trigger conditions.
- Configure the alarm message and the notification channel.

For more information, see Configure Alarms

Support for Capacity Analytics Widget in Service Details Page

Starting with this release, you can view the capacity issues in context of service using the Capacity issues widget. This feature helps you to track the capacity issues ahead of time by viewing the relevant service information in one place. For more information, see Capacity Analytics Widget..

Connector Parameters Tile

A new tile **Connector Parameters** has been introduced on the **Settings** page of DX Operational Intelligence. This tile helps you find the unique parameters that are required to integrate DX Operational Intelligence with various on-premise products. For more information, see Connector Parameters.

Known Issues

- The DX Dashboards can process a maximum of 9999 records for aggregation. This limitation impacts the forecast and projection dashboard generation, if the metrics monitored in Capacity Analytics are more than the specified maximum limit.
- The Monitored Inventory fails to display the existing network inventory.
- The Anomalies and Raw alarms count mismatches on the Alarm Widget in the Service Details page and Alarm Analytics page.

Defects Fixed

The following issues have been fixed in this release:

- The service health heatmap dashlet rendering issue is fixed.
- The Service Analytics search is hanging when filtering by custom attributes. This issue has been fixed, as the Service filter now handles the larger list of services per tenant.
- The maintenance windows were not able to handle the daylight savings time to adjust the windows.
- The Predictive Insights page appears blank when you provide values in the filter box.
- The 504 gateway error while clearing the bulk alarms from the UI has been fixed.
- The service risk that is impacted by the number of critical alarms was not behaving as expected based on the service definition in the case of shared CIs between services.

21.05

Feature Updates

This section lists all the new and enhanced DX Operational Intelligence features:

Service Analytics CRUD APIs

The Service CRUD APIs allow you to leverage the capabilities of Service Analytics from a programmatic interaction. You can access the Service CRUD APIs through the following adminui or gateway endpoints using Tenant token or User token.

- <gateway-end-point>/oipublic/serviceanalytics/*
- <doi-adminui-end-point>/oi/v2/oipublic/serviceanalytics/*

Using these APIs, you can programmatically perform the following tasks:

- Create a service and Update service
- Get Service Details
- Get Service Details and Sub-Service Details
- Get Service Hierarchy
- Remove Service Association
- Get all Services
- Delete Services
- Manage Service and Attributes

For more information, see Service CRUD APIs.

Display of Integration Parameters in Connectors Parameters User Interface

The mandatory parameters that are required for integration are now displayed in the **Connector Parameters** page. To view the **Connector Parameters** page, log in to DX SaaS, and in the left navigation pane click **Settings, Connector Parameters**.

This feature allows you to easily configure the connectors by reducing the onboarding time. You can use these parameters to configure the following connectors:

- Spectrum Data Publisher
- OI Connector
- oi_connector probe and apm_bridge probe
- DX Operational Intelligence Plugin
- DX RESTmon

For more information, see Connector Parameters.

Role-Based Access Control Enhancement

Enhancements to Role-Based Access Control capabilities in DX Operational Intelligence now enable additional sophistication in a user's access to product functionality in accordance with their role's access privileges.

View Ticket ID for an Alarm Without Enabling Ticketing System in DX Operational Intelligence

This feature provides the consolidated view of all alarms and allows you to view the ticket id for an alarm that is generated from the source product without enabling the ticketing system in DX Operational Intelligence. For more information, see Ticket Management.

Monitored Inventory Capability

The Monitored Inventory capability provides a unified inventory view of all entities available in DX Operational Intelligence. The unified view of all entities across the environment allows you to:

- Quickly locate and fix the monitoring or sub-optimally monitored devices.
- Manage device redundancies from different monitoring tools and understand the potential impact of a planned change event, which helps to plan the monitoring coverage.
- View all relevant details about the devices associated with an incident, especially their cross-domain correlated impact, and allows you to dive deeper into any aspect of those devices as required.
- Navigate to Service Analytics, Performance Analytics, Alarm Analytics, and Capacity Analytics in context of an entity.
- View all entities in context of services in the Widget View and Monitored View on the Service Details page.
- Navigate from Situation Alarms to Monitored Inventory in context of entities that are part of situation alarms.

For more information, see Monitored Inventory.

Anomaly Detection Improvements

The Anomaly Detection algorithm is now improved to handle better scale and reduced noise. The tenant administrator can configure it by navigating to **Settings**, **Configure Monitoring** tile. For more information, see Configure Monitoring.

Deprecated Service Risk Predictions Functionality

Starting with this release, the **Service Risk Predictions** (24hr Prediction) functionality has deprecated from Service Analytics.

Known Issues

Closed-Loop Service Hirerachy Not Working

You cannot delete a service when the service is a part of a closed-loop service hierarchy.

ServiceList APIs Fails to Work

The service list API (**GET** /oi/v2/sa/servicesList) fails to work due to a null pointer exception on the readserver.

Cannot Access READ APIs with Public Endpoint and Tenant Token

The following READ APIs can be accessed only through adminui endpoint and user token:

- /oi/v2/sa/services
- /oi/v2/sa/services/<service name>
- /oi/v2/sa/services
- /oi/v2/sa/getcustomproperties
- /oi/v2/sa/gettags
- /oi/v2/sa/services/hierarchy

Health Chart Fails to Indicate when the Health is Zero

On the Service Details page, the health chart fails to show a red line when the health value is zero.

Alarm Count Mismatch

Alarm count is different for the services having the same service definition.

Existing Child Service Unable to Delete from the Service Hierarchy

Symptom: Follow these steps:

1. Log in to DX Operational Intelligence.
2. Select a service.
3. On the Service Details page, click , and **Edit Service.**
4. Click **Manage all elements**.
5. Delete any child service from the hierarchy.
6. Click **Update Elements**.

The child service fails to delete.

Solution: As a workaround, from the **Edit Service** page, select the child service and click on the Service Details field.

21.04

Feature Updates
This section lists all the new and enhanced DX Operational Intelligence related features:

Alarm Analytics

Auto Alarm Closure of Raw Alarms
Alarm Analytics is now enhanced to automatically close all open raw alarms that have no updates for more than 30 days. This enhancement is enabled by default and helps you focus on the alarms that need immediate attention. If there is any update to an auto-closed alarm, then the alarm gets reopened immediately.

For more information about the auto alarm closure of raw alarms, see All Alarms

Capacity Analytics

Filtering Non-Percentage metrics for the Capacity Analytics User Interface

You can now configure percentage and non-percentage metrics for projections in both the Capacity Analytics UI and the DX Dashboards respectively. While the Capacity Analytics UI continues to cater to the percentage or linked metrics, you need to use the DX Dashboards to view projections for the metrics that are non-linked or having a non-percentage (pct) unit and are enabled for Capacity Analytics.

For more information about the filtering of non-percentage metrics for the Capacity Analytics UI, refer to the following links:

- Services Configuration
- Groups Configuration
- Monitored Groups and Services
- DX Dashboards

Known Issues

This section lists all the known issues in this release:

Certain Service Metrics are not being displayed while using the Service Search Filter

Symptom: In the DX Operational Intelligence UI, perform the following steps:

1. Click on the **Capacity Analytics** capability on the left navigation pane.
2. Click icon on the top-right of the screen, and click **Configure** option.
3. Click on the **Services** tab.
4. Enter the name of the service in the **Filter** field.
5. Select the **Service**, and click **Next**.
   You can notice that certain service metrics are not being displayed.

Workaround: Clear the search text and click on **Next**.

21.03

Feature Updates

This section lists all the new and enhanced DX Operational Intelligence features:

Capacity Analytics

Support for Configuration Item Type

Capacity Analytics now supports the Configuration Item (CI) type metadata. This enhancement eases the process of viewing, understanding, and identifying the entity or the sub-entity type for which you are configuring the metrics. This is helpful in scenarios where you use RESTmon to collect such metrics.

When you monitor the storage or network (for example, entity/sub-entity) the same metrics are collected for the storage entity and also for the various disks that are part of the storage as sub-entities. The availability of the CI type details during the metric configuration for Capacity Analytics enables you to make an informed decision. For more information on Configuration Item type, see Services Configuration and Groups Configuration

Alarm Analytics

Annotation Support for Situations

You can now add annotations to Situations. This ability helps you add details about a specific situation; for example, cause, resolution, troubleshooting steps, or any other important information. This enhancement, therefore, eases the process of understanding the details associated with a situation, irrespective of whether a situation is in the closed or open state.

Additionally, you can also synchronize your annotations with the related **ServiceNow (SNOW)** tickets. This ensures that the SNOW tickets reflect the same annotation details. For more information on the Annotation Support for Situations, see Situation Alarms

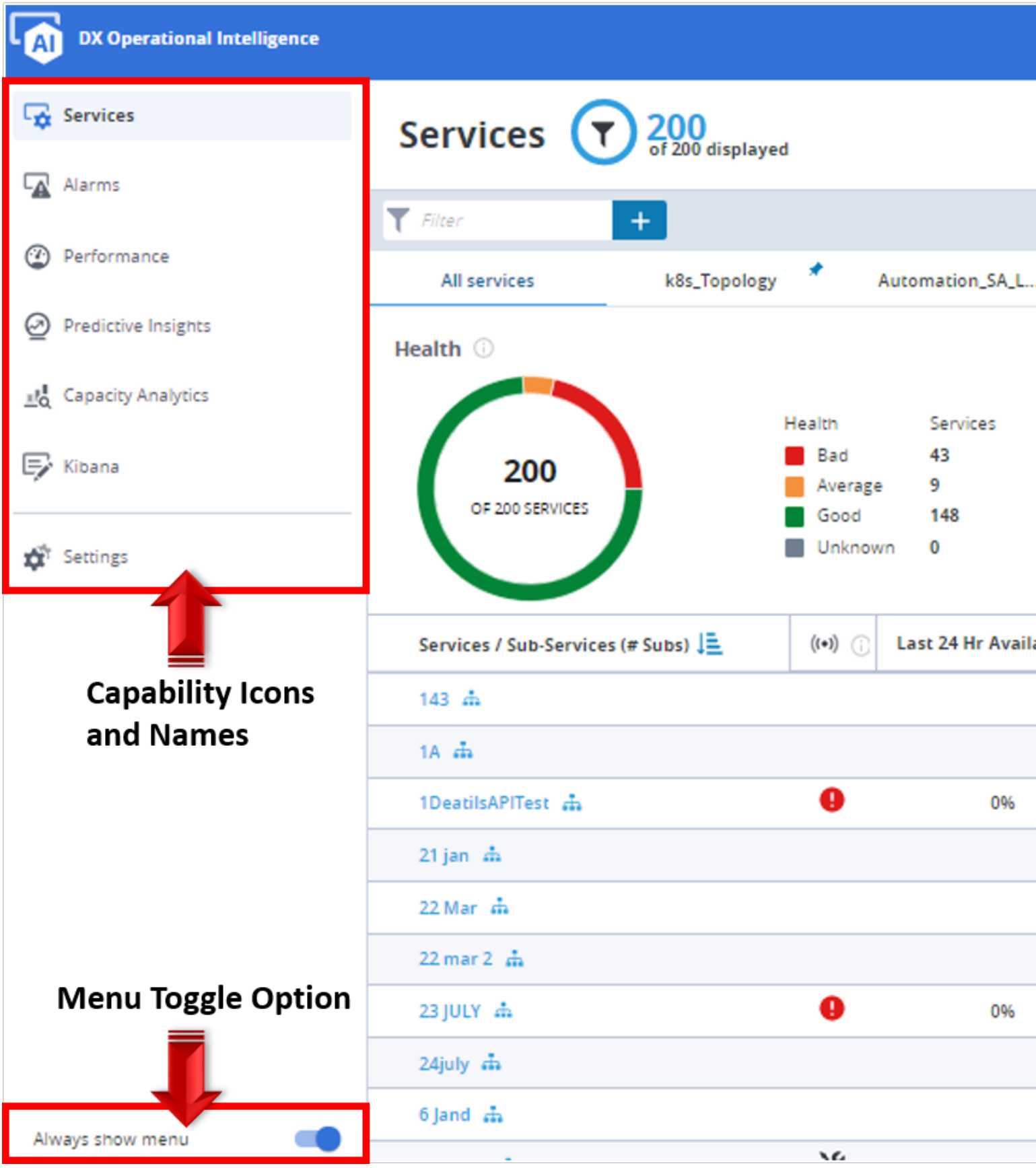
DX Operational Intelligence User Interface Enhancements

With the 21.03 release, we have an improved DX Operational Intelligence User Interface with new features and functionalities to make your work easier and more effective.

- The DX Operational Intelligence UI now enables you to **set your own default landing view** (or page) based on your business requirements.
- The **Settings** view is now enhanced to display all the configurations in the form of cards. The configurations that are mandatory for enabling DX Operational Intelligence are displayed in full-color cards format ( ), and the configurations that are optional but enhance your overall user experience in DX Operational Intelligence are displayed in semi-color cards ( ). If you are a first-time user, we recommend that you use the **Take a Tour** option on the **Settings** view to learn more about the configurations you need to perform to get started.
  These enhancements facilitate your onboarding process by helping you set up your system and effectively monitor your end-to-end IT Operations.
- The DX Operational Intelligence capabilities can now be accessed using the **App Launcher** ( ) icon at the top-right of the screen.
- The DX Operational Intelligence **version** is now displayed at the bottom of the screen.

The DX Operational Intelligence landing page now displays only the icons for each capability. Hovering over any capability on the left navigation pane expands the menu bar to display all the icons along with the capability names. Additionally, you can use the **Always show menu** toggle option to display all the capabilities.

The following illustration highlights the key UI enhancements in this release:


DX Operational Intelligence
Services
Alarms
Performance
Predictive Insights
Capacity Analytics
Kibana
Settings
Capability Icons and Names
Menu Toggle Option
Always show menu
Services
200
of 200 displayed
Filter
All services
k8s_Topology
Automation_SA_L...
Health
200
OF 200 SERVICES
Health
Services
Bad
43
Average
9
Good
148
Unknown
0
Services / Sub-Services (# Subs)
Last 24 Hr Availa
143
1A
1DeatilsAPITest
0%
21 jan
22 Mar
22 mar 2
23 JULY
0%
24july
6 Jand

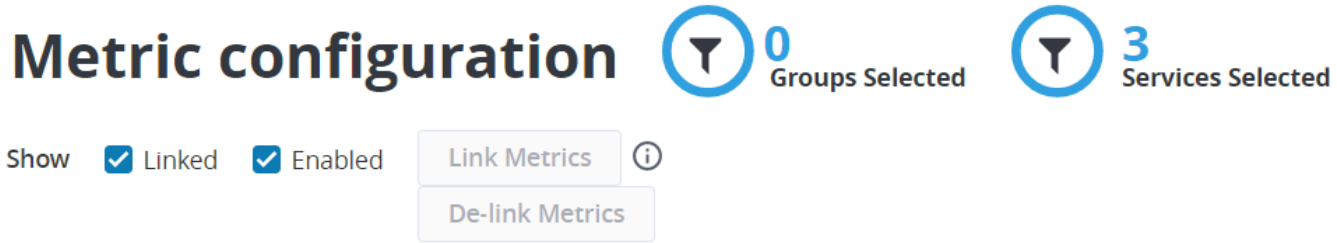
Known Issues

This section lists all the known issues in this release:

Button alignment issues in the Metric Configuration view when you enable the Always show menu toggle option

Symptom: In the DX Operational Intelligence UI, after you enable the **Always show menu** toggle option, perform the following steps:

1. Click on the **Capacity Analytics** capability on the left navigation pane.
2. Click **Configure** option on the top-right of the screen.
3. Select some **Services/Groups** and click **Next**.
4. Select the **Show Linked** and **Enabled** checkboxes respectively.
   You can notice the button alignment issues.



Workaround: This issue will be handled in the subsequent release.

21.02

Feature Updates

This section lists all the new and enhanced DX Operational Intelligence related features:

Capacity Analytics

What-If Analysis

This new feature enables you to understand the dynamics of a scenario and the impacts of levers and dependencies in your choices. What-If Analysis enables capacity planners to perform an analysis based on the service KPIs and also in the context of the service. What-if Analysis provides you with an option to compare the capacity for entities in different scenarios based on the service KPIs to help you plan better.

What-if Analysis helps you understand the future needs of resources by simulating growth in the monitored KPI metrics and enables you to predict bottlenecks based on the 'what-if' scenarios. What-if Analysis also helps you make sense of the business metric in relation to the infrastructure metric. For more information about What-if Analysis, see What-If Analysis

Performance Analytics

Metric Groups

This new feature offers anomaly detection support in the metric group configuration. You can use the Metric Groups to control and monitor anomaly detection on certain metric groups. You are provided with the ability to create specific metric groups and enable anomaly detection for the specific metrics that you want to focus on, instead of having anomaly detection enabled for all the metrics. For more information, see Metric Groups

Alarm Analytics

Alarm Management for Stable Situations

You can now perform standard workflows related to alarm handling in the context of situations that include alarm actions such as assign, clear, hide, unassign, assign, view remediations, which helps you make Situations your primary alarm management paradigm. You are now provided with the ability to view the columns and information that align with those controls in the Situations view, which enables you to understand the state and ownership details of a situation.

You can now control the scope of actions taken on a situation cluster by leveraging the situation root cause, which includes the entire situation cluster. You can also control the scope of actions taken on individual situation alarms within the cluster. For more information about Situation Alarms, see Situation Alarms

Manual Ticketing for Stable Situations

You can now manually create a ticket from a situation alarm that occurs, which enables you to triage and resolve the problem identified in the situation. For more information about Situation Alarms, see Situation Alarms

Known Issues

This section lists all the known issues in this rollout:

What-if Analysis does not display the projection data as per the selected Severity state

In Capacity Analytics, while performing a comparison of scenarios in What-If Analysis, if you select a particular Severity state, the What-if Analysis view does not display the projections based on the selected Severity state.

Mismatch in the total count of metrics between the Metric Monitoring Groups UI and the metrics processed by the OI Metric Publisher

In the Metric Monitoring Groups view, there is a mismatch between the Metric Monitoring Groups UI and the metrics processed by the OI Metric Publisher.

21.01

Feature Updates

This section lists all the new and enhanced DX Operational Intelligence related features:

Service Analytics

The metrics and service KPIs are now enhanced to get calculated and published every 5 mins instead of the previous 15 mins in the Services view. This helps you monitor your services even more proactively and provide a quick resolution. For more information, see Services Overview.

Capacity Analytics

Enhanced with User Interface and usability improvements in the Services/Groups Configuration view. For more information, see Services Configuration and Groups Configuration.

Alarm Analytics

- The Global Search filter in the All Alarms view is now enhanced to filter out the alarms using metric names. This helps you identify the alarms in a more efficient way and provide a quick resolution. For more information, see All Alarms.
- The raw alarms are now enhanced to display Service Alarm details. You can now view the Service Alarm IDs, and you are also provided with an option to drilldown to the service alarm details directly in the Service Alarms view. This

helps you identify the issues in a more efficient way and provide a quick resolution. For more information, see Service Alarms.

Known Issues

This section lists all the known issues in this rollout:

Issue while changing the severity value and viewing the respective entities based on metrics

In Capacity Analytics, while viewing a Services/Groups details, if you change the severity of a Services/Groups under the Projection by Entities tab and update the Metric filter value, then the severity option becomes unavailable and you would not be able to view the respective entities based on metrics.

Issue in viewing data when you enable a metric that does not have a metric unit

In Capacity Analytics, while configuring the metrics in the Metrics Configuration view, if you enable a metric that does not have a metric unit, you would see a blank page.

Workaround: Ensure that the metric you are going to enable has a metric unit assigned to it, else you need to provide a metric unit before enabling the metric.

Issue while linking the metrics when the metric unit is renamed

In Capacity Analytics, when you rename a metric unit in the metric configuration view, the Link metrics option becomes unavailable.

Compatibility Matrix

Before you begin with DX Operational Intelligence, review the following information:

Supported Web Browsers

- Google Chrome v61 and later
- Firefox v96 and later

Supported Integrated Product Requirements

To view the supported versions required for integration, see DX Operational Intelligence Interoperability.

Known Issues

DX Operational Intelligence includes the following known issues:

- **Universes**
  - **Sample Displays Child Services**
    When you click **View Sample** on the Create or Edit Universe pages, the sample displays the child services even if the parent has no access.
  - **Insights Does Not Honor Universe**
    The DX Operational Intelligence insights does not honor the universe and hence displays the same insights for all the universes.
- **Selected Transactions are Overriden**
  On the Topology page, when you apply the Groupby filter after selecting a few transactions, the selected transactions are overridden and all the transactions are selected.
- **SLIs/SLOs**
  - **New SLI Alarm is Not Created**

After you close the alert manually, a new SLI alarm is not created even when the condition is met again. The alarm is created only if the severity is changed.

- **Existing Alert is Not Closed**
  If the SLI name is changed or the alert or SLI is deleted, the existing alert is not closed.
- **SLI Alarm is Not Generated**
  When the SLI is created on a service, if you add another service to the SLI after adding the alert configuration, the new service is not added.
  Workaround: Close the alert.
- **Intermediate Forward Metric Does Not Calculate Correct Values**
  The SLI metrics with an operator other than Expression on the intermediate forward metric do not calculate correct values.
- **Publish SLIs to Use Intermediate Metrics**
  The SLI UI does not allow you to use intermediate metrics without first having a configured initial SLI that is published.
- You see a sudden dip in error budget for a few data points, and min or max values appears incorrect for comparator metrics.
- If an SLI of type *Availability* is associated with the Availability of any service, and when the SLI type is changed to any other type (other than Availability), the service to SLI association is not removed.
  As a workaround, trigger the service update to remove the association.
- The error budget and compliance calculations are not correct. This issue occurs when there is a change in the threshold value and when the pod is restarted.
- The error budget decreases after the completion of the compliance window with 1 minute and 5-mins aggregation interval.
- A few SLI data points are missing for a child service though it is coming for a parent service without service content and raw metrics are also generated.
- The service, which is under the wrong error budget legend and SLI chart with old data points, fails to display when the SLI metric is stopped.
- The SLI threshold metric (Availability KPI) is visible on the Edit Service page even after you set the SLI key to false.
- The Update button is enabled when no metric is selected while editing a service.

- **DX Dashboards**
  - When more than 50 concurrent users access DX Dashboards with 15-seconds auto-refresh interval, the requests are failing.
  - The DX Dashboards session is timing out after 24 hours even though the auto-refresh mode is enabled.
- The **Affected Metrics** tab in the Predictive Insights view is not appearing for UIM performance alarms.
- **RBAC Related**
  A Tenant Administrator is unable to edit or delete a user when the associated custom role is deleted or deactivated.
- **Message Templates**
  If the tenant ID variable in the message template payload has a space, the following error is displayed in the Read server logs:

```
2022-03-21 15:50:09,121 [JmsMQListener,pool-34-thread-8] DEBUG [] - TID[0|22314373|22314373|dxi-
adminui-75d6d7f97d-:2:1821] Request (ID:dxi-adminui-75d6d7f97d-wtc8c-39940-1647810638102-6:1:1:6:1179)
 served successfully by thread (pool-34-thread-8)
2022-03-21 15:50:31,496 [JmsMQListener,pool-34-thread-1] DEBUG [] - TID[0|22314373|
22314373|dxi-adminui-75d6d7f97d-:2:1822] Serving request (ID:dxi-adminui-75d6d7f97d-
wtc8c-39940-1647810638102-6:1:1:6:1180) by thread (pool-34-thread-1)
2022-03-21 15:50:31,496 [NotifyHandler,pool-34-thread-1] INFO  [] - TID[0|22314373|22314373|dxi-
adminui-75d6d7f97d-:2:1822] Request received to validate and send notification with SMTPServer details
......
```

  Workaround: Enter the Tenant ID variable as ${tenant_id}.

Third Party Software Requirements

To view the software license information for any of the listed components, download the Third-Party Acknowledgements.zip file.

Overview of DX Operational Intelligence

DX Operational Intelligence enables IT operations teams to make smarter, faster decisions for enhancing user experience and improving IT service quality and capacity through cross-domain contextual intelligence.

DX Operational Intelligence is built on an open, powerful engine, DX Operational Intelligence provides users with comprehensive insights by ingesting and analyzing a diverse data set including metric, topology, text, and log data. The machine learning–driven analytics, along with the out-of-the-box visualization and correlation, helps drive a superior user experience and deliver significant operational efficiencies. DX Operational Intelligence supports on-premise and SaaS deployment models.

This documentation focuses on the On-Premises version of DX Operational Intelligence.

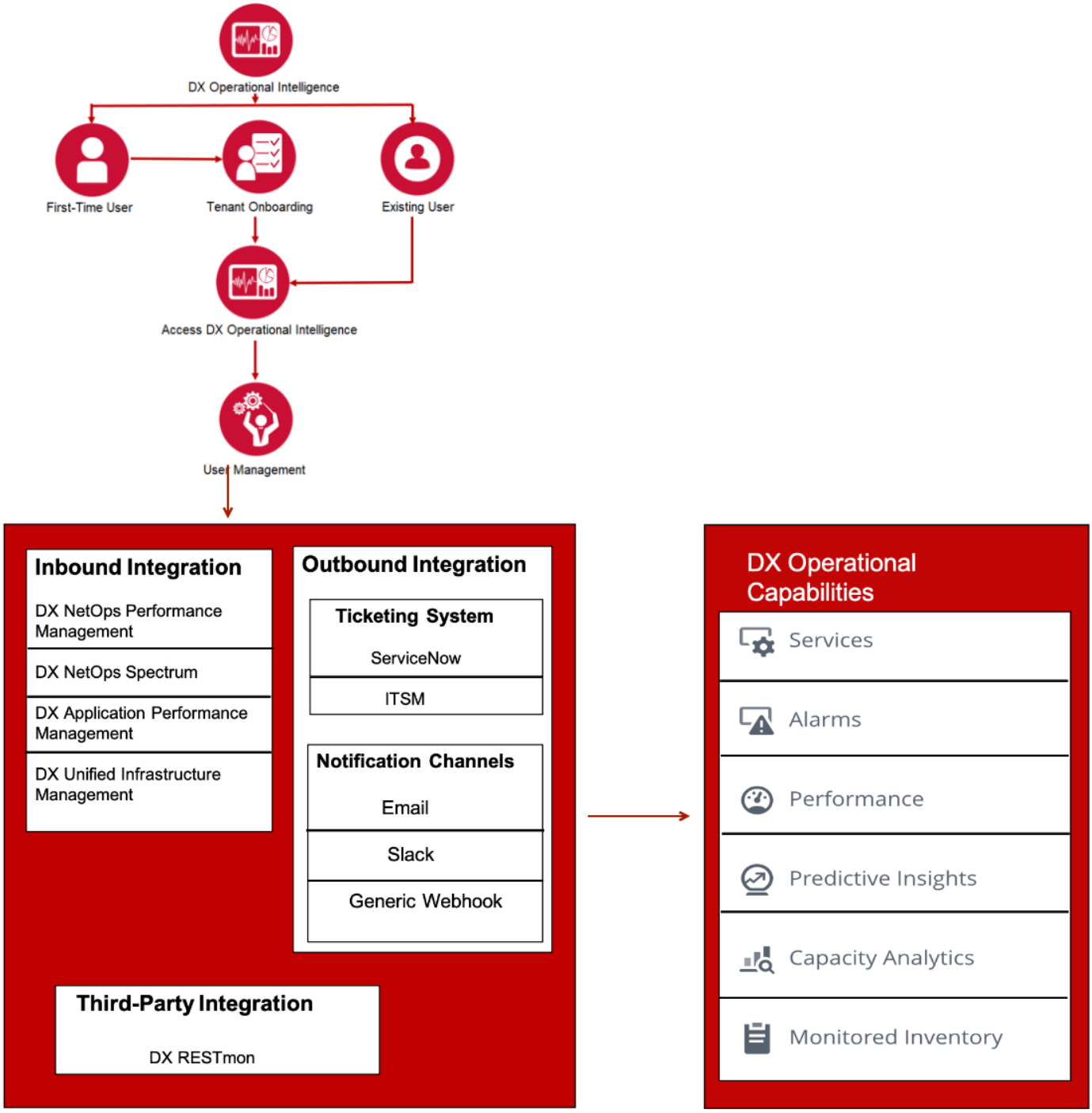
This documentation focuses on the SaaS version of DX Operational Intelligence. For more information, see Tenant Onboarding.

Use DX Operational Intelligence capabilities to achieve the goals and overcome some of the following key challenges that different types of users face.

Roles	Goals	Challenges
Vice Presidents Directors Managers of IT Operations	• Eliminate silos between the different teams for faster issue identification and resolution. • Enable teams to make smarter, faster decisions • Enable teams to focus on high-value activities • Enable teams to focus on core business-related innovation and automation.	• Different monitoring tools and data sources create siloed views • Teams are stuck in reactive triage and issue troubleshooting
IT, System, and Network Administrators	• Identify potential issues proactively • Spend less time chasing issues, and reporting • Focus on innovation and new delivery models	• Tool Sprawl for point monitoring resulting in disparate views and reactive alerting
Service Owners, App Owners, and Developers.	Want more control and visibility over the infrastructure and network running their apps.	• Difficult to manage the increasing volume of apps and channels • Difficult to get to the root cause of problems due to lack of the end-to-end view of customer transactions

Flow Diagram

The following process diagram depicts the step-by-step tasks that you must perform for working with DX Operational Intelligence:



Monitored Inventory Predictive Insights Manage Users Performance Analytics Capacity Analytics Configure Generic Webhook Channel Configure Slack Channel Configure Email Channel Configure CA Service Management as Channel Configure ServiceNow Channel Alarm Analytics Service Analytics Integrate DX Unified Infrastructure Management Integrate DX Application Performance Management Integrate DX NetOps Spectrum Integrate DX NetOps Performance Management Access DX Operational Intelligence Tenant Onboarding Process

DX Operational Intelligence Capabilities

The DX Operational Intelligence provides the following capabilities:

Service Analytics

Unify service health and availability across all management domains to provide a holistic view of key business or IT services. Operations managers and administrators can drill down to identify key components that are causing service disruption. Service Analytics provides the following information:

- Health of service (availability and risk)
- Infrastructure and applications that are mapped to a service
- Impacted Services

For more information about Service Analytics, see Service Analytics.

Alarm Analytics

Provides overview and insights into service and derived alarms from multiple data sources. By using Alarm Analytics, you gain the following benefits:

- Reduce alarm noise from multiple products
- Correlate alarms across products to identify the root cause
- View Probability bands to determine buildup to an alarm
- Fine-tuning alarm threshold by analyzing the historical pattern

For more information about Alarm Analytics, see Alarm Analytics.

Performance Analytics

Enables users to detect performance bottlenecks and anomalies to predict problems before they occur. Performance Analytics in DX Operational Intelligence allows you to:

- Compare multiple metrics for the same device
- Compare multiple metrics across multiple devices
- Compare uni-variate metrics across single and multiple devices
- Compare multivariate metrics across multiple devices

For more information about Performance Analytics, see Performance Analytics.

Capacity Analytics

Helps you manage your IT resources by ensuring that resources are sized correctly to meet current and future business needs. Capacity analytics allow companies to:

- Predict capacity for peak seasons
- Know when more resources are needed and plan accordingly
- Only buy more resources when necessary
- Efficiently manage virtual and physical infrastructure
- Eliminate waste by identifying areas that are underutilized

For more information about Capacity Analytics, see Capacity Analytics.

Predictive Insights

With the power of machine learning, discover patterns and trends. Based on these trends, the application predicts events that are likely to happen in the future. The events that could be predicted are:

- Performance
- Capacity

For more information about Predictive Insights, see Predictive Insights.

Monitored Inventory

The Monitored Inventory capability provides a unified inventory view of all entities available in DX Operational Intelligence. The unified view of all entities across the environment allows you to:

- Quickly locate and fix the monitoring or sub-optimally monitored devices.
- Manage device redundancies from different monitoring tools and understand the potential impact of a planned change event, which helps to plan the monitoring coverage.
- View all relevant details about the devices associated with an incident, especially their cross-domain correlated impact, and allows you to dive deeper into any aspect of those devices as required.

For more information about Monitored Inventory, see Monitored Inventory.

DX Dashboards

The DX Dashboards is a visualization platform that is designed to search, view, and interact with the data that is stored. The DX Dashboards help you visualize real-time analytics by creating comprehensive business reports. See Root Cause Analysis and Alarm Correlation to understand how some of these capabilities work together to help a company quickly identify the root cause of an issue that impacts the customer experience.

For more information about DX Dashboards, see DX Dashboards.

Integrations

Integrations enable you to connect various data sources with DX Operational Intelligence. You can use integrations to analyze and correlate data and monitor data (metrics, alerts, logs, inventory) on DX Operational Intelligence.

DX Operational Intelligence supports integration with the following monitoring data sources:

Broadcom Products Integration

DX NetOps Performance Management	DX Unified Infrastructure Management
DX NetOps Spectrum	DX Application Performance Management

Third-Party Integration DX Restmon

Quick Start Guide for Tenant Administrators

This section helps the tenant administrator get started with the DX Operational Intelligence.

Use the links in the following table to quickly find information that will help you navigate and work with DX Operational Intelligence:

First Time Tenant Administrators	As a first-time tenant administrator, you must obtain the tenant details from your accounts team to get started with DX Operational Intelligence. You can access the following information: • Getting Started • User Management • Integration Overview • DX Operational Intelligence Capabilities • APIs
Existing Tenant Administrators	As an existing tenant administrator, you can directly add users and start with the integration. You can access the following information: • User Management • Integration Overview • DX Operational Intelligence Capabilities • APIs

Getting Started

An overview of the architecture and tasks list for tenant administrators to on board and access DX Operational Intelligence.

The Getting Started section contains information that the first time tenant administrators to understand the architecture, quickly onboard, access, and use DX Operational Intelligence:

- Architecture Overview
- Ports and Integration
- Tenant Onboarding Process
- Access DX Operational Intelligence
- User Management

Architecture Overview

DX Operational Intelligence uses a collection of services to process and store data.

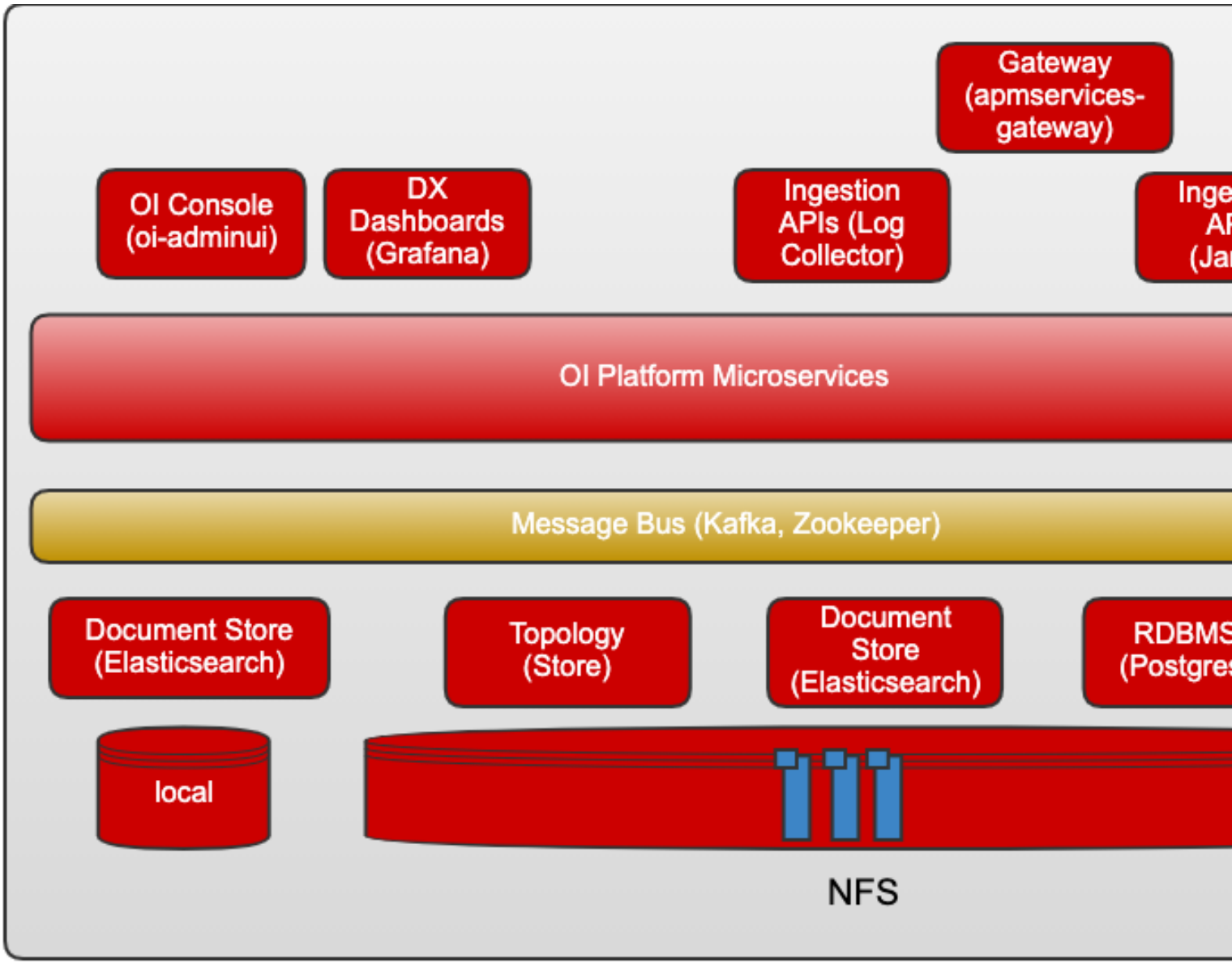
This section describes the data flow between different components in DX Operational Intelligence. The data includes monitored elements, relationships between monitored elements, events, alarms, metrics, and logs.

DX Operational Intelligence is built using microservices architecture and is delivered as a set of docker containers deployed through Kubernetes. DX Operational Intelligence receives data (alarms, metrics, logs, topology) from various connectors through the Ingestion APIs. This data is processed by various microservices to derive actionable insights as per specific use cases. It also provides outbound integration through webhook, email, and Incident Management (ServiceNow). You can view the received data and insights on the DX Operational Intelligence UI and Dashboards.

DX Operational Intelligence uses the following highly scalable technologies for processing and storing the data:

- **Elasticsearch:**Search engine and big data backend data store for textual data like logs, alarms, and events.
- **Kafka:** Kafka is a high throughput message bus that is used as the communications backbone for all DX Operational Intelligence services. The individual DX Operational Intelligencemicroservices communicate by publishing data to Kafka topics and consuming data from Kafka topics.
- **Metric Store, Topology Store & Blob Store (RocksDB based)**RocksDB is a fast and highly scalable persistent key-value store. DX Operational Intelligence uses these Data Platform services that are built using RocksDB as the backend for storing metrics, topology, and Blob data.
- **Hazelcast:**Hazelcast is a fast and scalable distributed cache that is used as the internal cache for various OI micro-services.

The following diagram depicts the DX Operational Intelligence architecture at a high level:



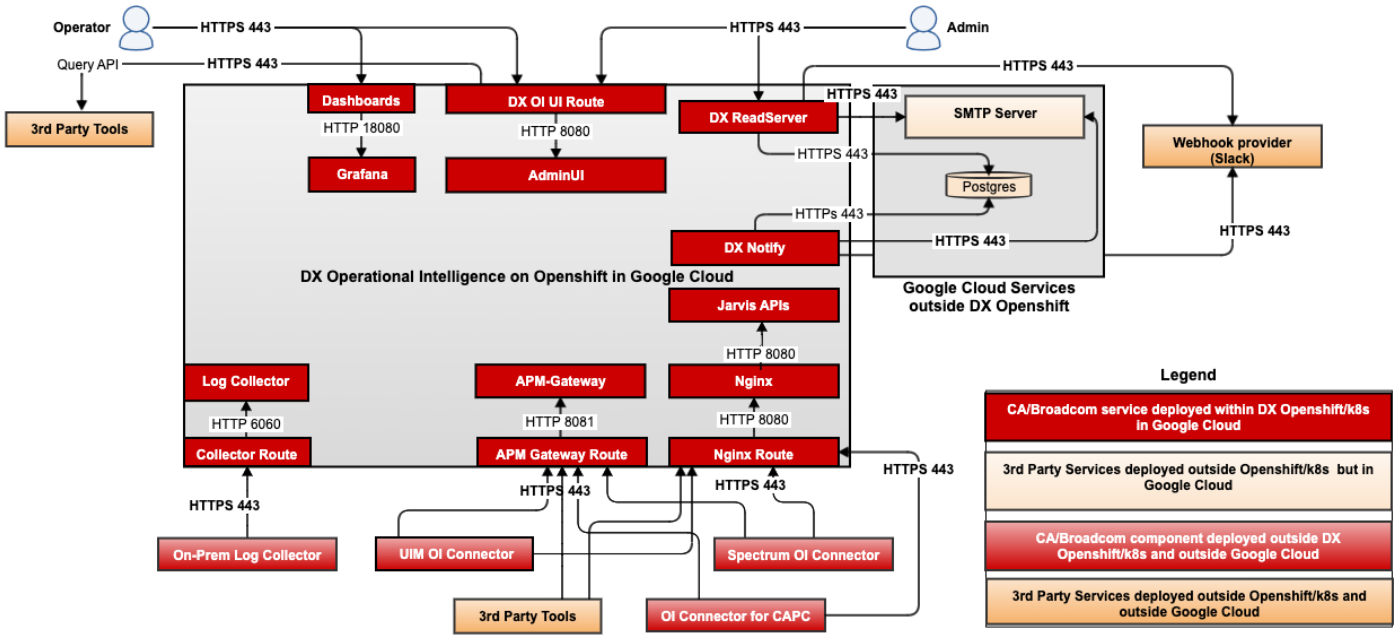
You can view the architecture video here:

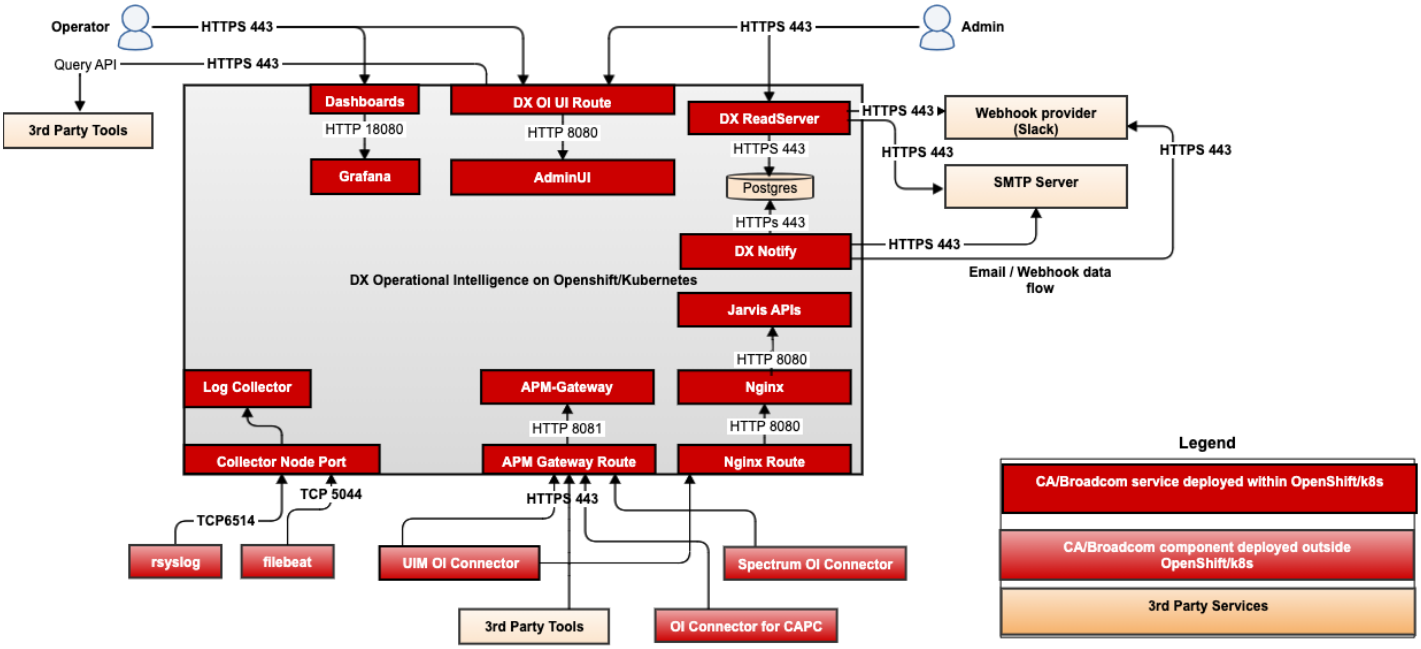
Ports and Integration

DX Operational Intelligence components require certain ports to be exposed outside of your datacenter firewall. Administrators must ensure that these ports are open before installation.

DX Operational Intelligence being an AIOps solution provides a robust integration framework to support the ingestion of various data from different sources. DX Operational Intelligence provides turn-key integrations with other domain monitoring tools of Broadcom like DX Netops, DX Application Performance Manager, and DX Infrastructure Manager. The third party monitoring tools can be integrated with DX Operational Intelligence through the RestMon utility. All the integrations that are related to ingestion and query ports are as follows:

Figure 1: DX Operational Intelligence - Ports and Integrations





Components	Port	Protocol	Needs to b
doi-adminui	80/443	http/https	Yes
dxi-adminui	80/443	http/https	Yes
• dxi-grafana • dxi-grafana-reporter • dxi-grafana-services	80/443	http/https	Yes

apmservices-gateway	80/443	http/https	No
doi-nginx	80/443	http/https	No
log-collector	80/443	http/https	No
cpa-ng	80/443	http/https	No
log-collector	6514	TCP	No

Tenant Onboarding Process

Before users can access DX Operational Intelligence, they must complete the tenant onboarding process to obtain their user credentials.

Tenant onboarding is a one-time process. You must contact your organization's account team to access the capabilities of the DX Operational Intelligence application.

The following flow diagram helps you understand the onboarding process:

Figure 2: Tenant Onboarding Process

CA Technologies - A Broadcom Company team creates a tenant administrator with the login credentials for your organization. An email notification with the tenant details is sent to your registered email address. You can sign in to the application using the tenant details, and create a password on the first-time login to access the DX Operational Intelligence application.

The following is a sample email that you receive from

Dear Customer

On behalf of the CA Technologies - A Broadcom Company team, we are glad to have you as a customer. We hope DX SaaS solution will help your business deliver world-class digital experiences to your customers.

Here's how to get started: As a new customer, you now have access to Application Console to interact with application.

Click here to sign in.

We have created Tenant Administrator with following credentials:

Client Name: Test Company
Username: test user
First Name: test
Last Name: user
Email: testuser@gmail.com
Tenant Name: test

Please refer to DX SaaS Documentation for additional product information.

Register for CA Support: Please make sure to register for CA Support. You will need your "client name" to create a CA Support account. Go to support.ca.com to register. Please also forward your "client name" & "sold-to ID" to your secondary contacts, so they can create their CA support accounts as well.

The contact information you provided as part of the on-boarding questionnaire will be used by DX SaaS to notify your business of maintenance and emergency announcements. You may update your key contacts at any time by submitting a case at support.ca.com

Notification Type: When requesting contacts to be added, please specify notification type for each contact submitted.

System Notification: Contact receives both Maintenance and Outage notifications. This is currently a requirement for receiving SLA reports.

Emergency Notification: Contact receives only Outage notifications and Maintenance notifications are not sent.

Access DX Operational Intelligence

Users can access DX Operational Intelligence from the DX SaaS/DX Platform Launch Pad. The DX SaaS/DX Platform Launch Pad is the landing page for all DX SaaS/ DX Platform products.

DX Operational Intelligence is available through the DX SaaS Launchpad.

To log into DX Operational Intelligence, follow these steps:

1. Open the DX Operational Intelligence URL that you have received from the account team of your organization.
2. Provide the login credentials.
   The Launchpad opens. The Launchpad enables you to access the registered capabilities.
3. Click **Open** on the DX Operational Intelligence tile.

DX Operational Intelligence User Interface

This section explains the following user interface:

- Launch Pad
- DX Operational Intelligence Home Page
- Settings Page

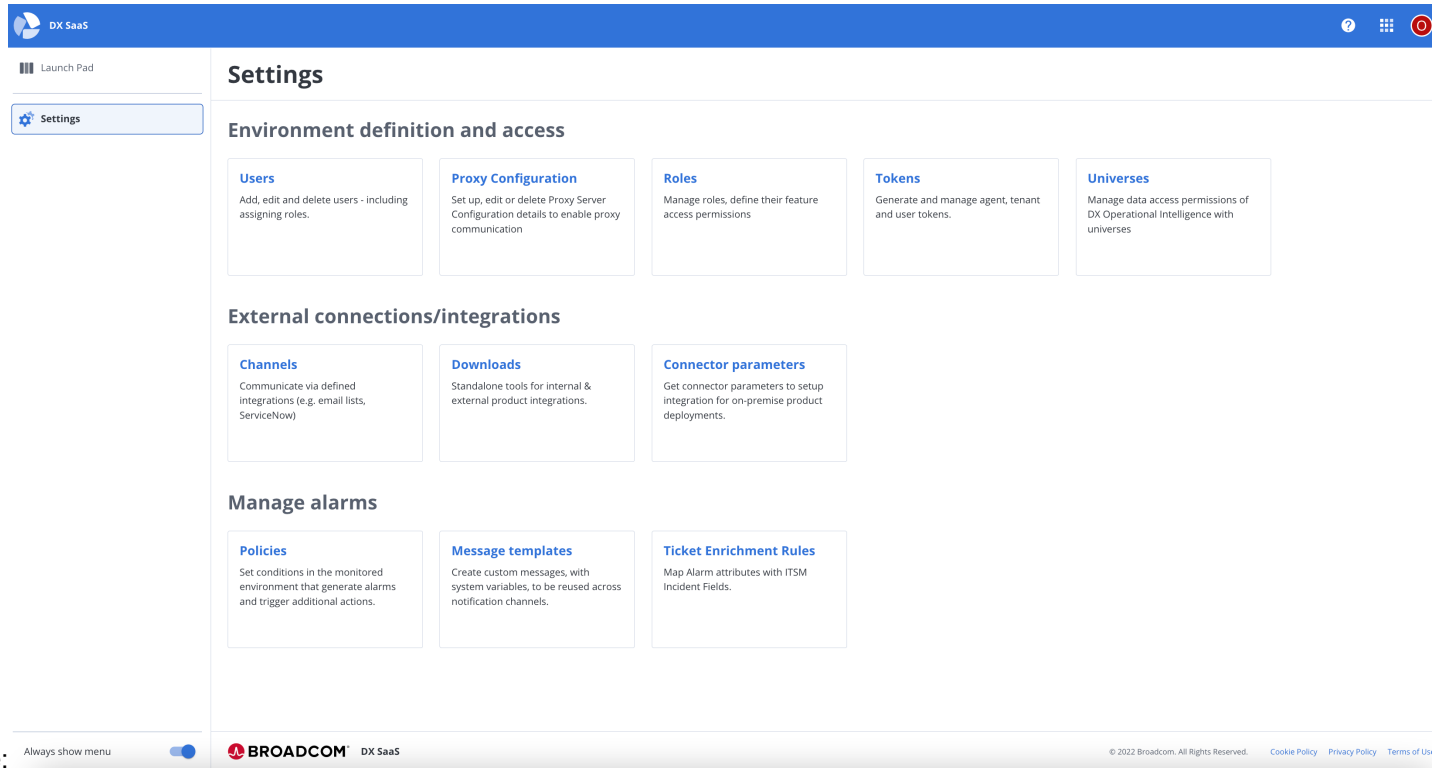
Launch Pad

The DX SaaS Launch Pad is the landing page for all DX SaaS products.

The Launch Pad is the landing page that appears as you log in to DX SaaS and this page displays the capabilities you have registered for. For example, DX Operational Intelligence, DX Application Performance Management, and so on.

The Launch Pad is the landing page that appears as you log in to DX Platform and this page displays the capabilities you have registered for. For example, DX Operational Intelligence, DX Application Performance Management, and so on.

The following image illustrates the launch pad and explains the various options available:

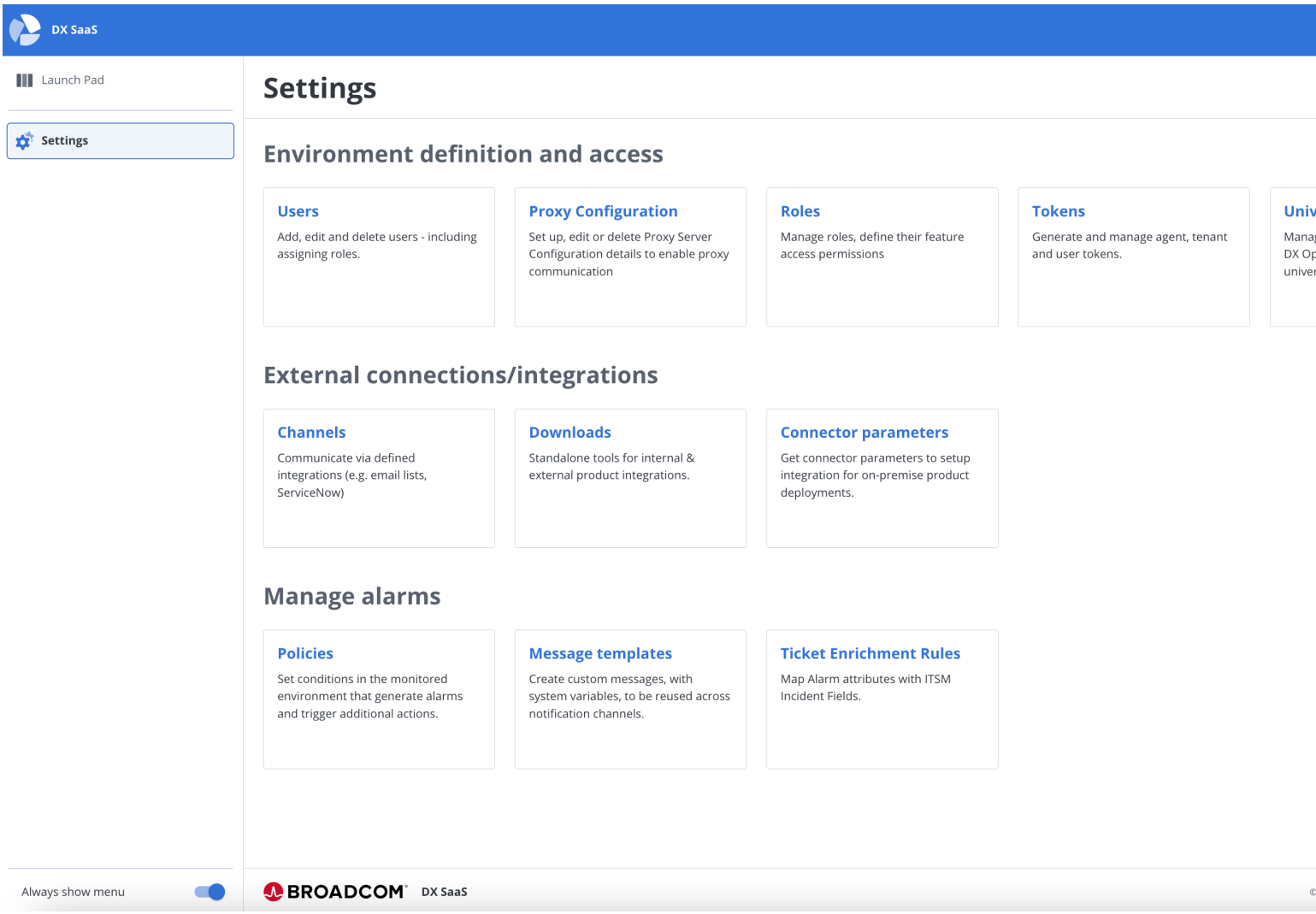


- **Launch Pad (1):** Displays the registered capabilities.
- **Settings (2):** Displays the Settings for the common services.
- **Always show menu (3):** Allows you to enable or disable the left side navigation pane.
- **App Launcher (4):** Allows you to navigate to the Capabilities. For example, DX Dashboards, DX Operational Intelligence.

{"URL":["https://cloudmanagement/#"],"description":"concept.dita_e3713f93-c7c4-45ab-b12f-ed568dfa6208","new":"","new_video":"","admin":"","troubleshooting":{"masterkb":"","text":"","URL":[]},"pendo":"","video":[]}

Common Services - Settings Page

The **Settings** page links to common services for all the capabilities.



The common services for all the capabilities are as follows:

- Users: Manage users and roles in DX Operational Intelligence.
- Proxy Configuration: Use proxy configuration to route all traffic through a host that has more permissive outbound policies.
- Roles: DX Operational Intelligence uses Role-based Access Control (RBAC) to restrict access based on the roles of the individual users within the enterprise.
- Tokens: DX Operational Intelligence uses the security tokens to authenticate requests and authorize access to automate user workflows using REST APIs.
- Universes: DX Operational Intelligence uses Universe to define the data access permissions for users.
- Channels: Configure the communication between DX Operational Intelligence and third-party integrations.
- Automation: Integrate the AIOps and Automic Automation integration. This integration is Broadcom's hosted, operated, and maintained solution to fully automate incident remediation.
- Downloads: Downloads the connectors to integrate with internal and external products. You can download the following connectors:

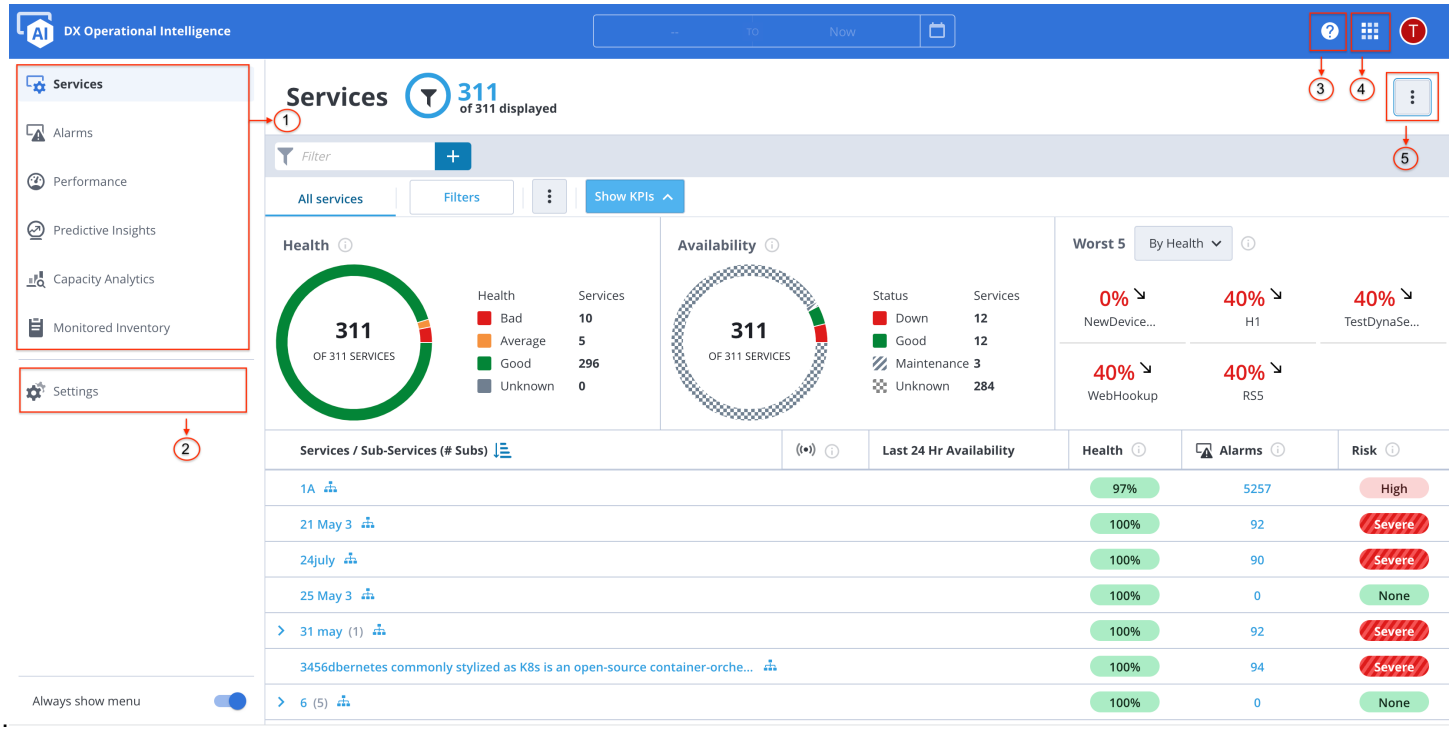
  - DX Gateway
  - Log Collector
  - NetOps Connector
  - Spectrum Data Publisher
- Connector Parameters: Provide all the mandatory parameters that are required to configure DX Operational Intelligence connectors on a single page.
- Policies: Set the conditions to determine when notifications are sent for alarms.
- Message templates: Define the content of the message to be sent when an alert occurs.

{"URL":["https://cloudmanagement/#/settings"],"description":"concept.dita_e1a1b718-19c6-4ca6-8cd9-f9bfc7d0611c","new":"","new_video":"","admin":"","troubleshooting":{"masterkb":"","text":"","URL":[]},"pendo":"","video":[]}

DX Operational Intelligence Home Page

The DX Operational Intelligence home page allows users to explore the capabilities of DX Operational Intelligence, view configuration requirements, and launch apps.

The DX Operational Intelligence home page contains the capabilities and Settings page. You can set any capability page or Setting page as a Landing page using the **Set As OI Landing Page** option.



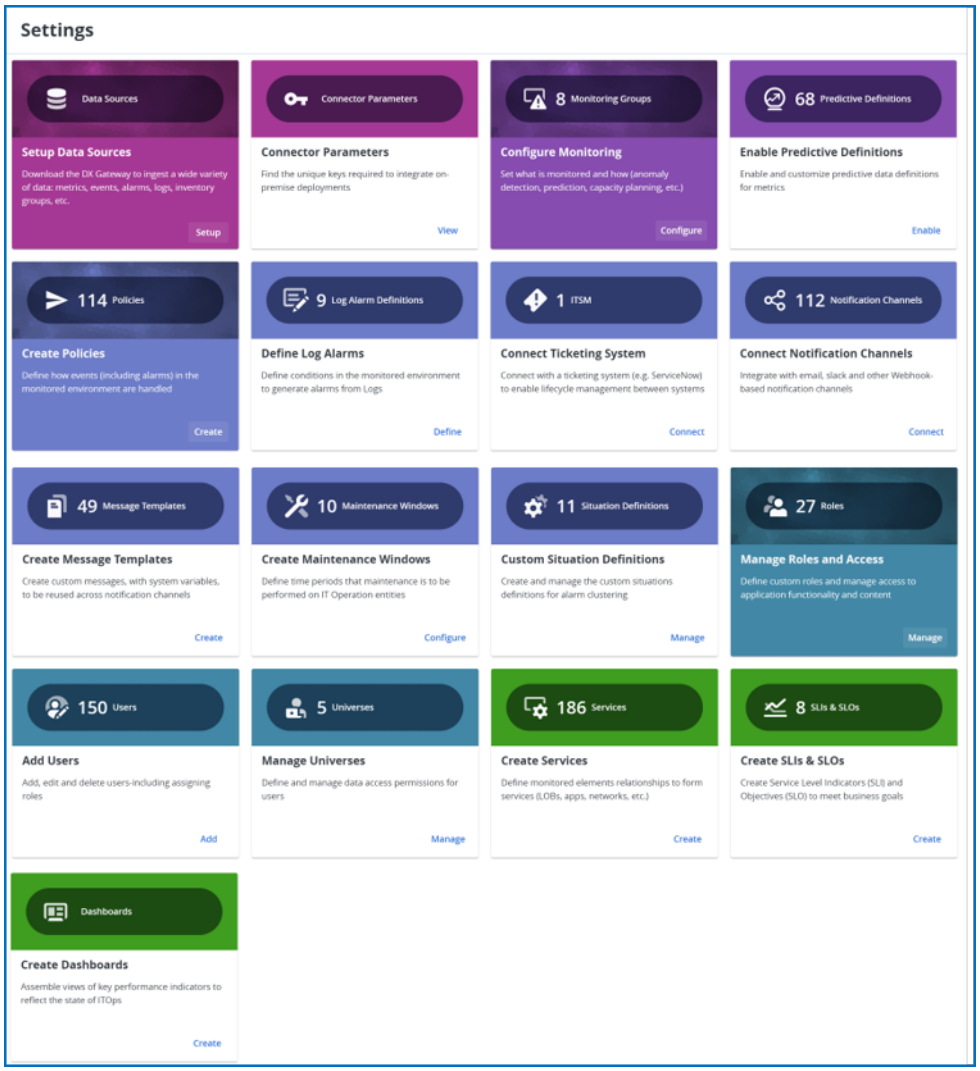
- **DX Operational Intelligence Capabilities (1):** Allows you to navigate to the capabilities of DX Operational Intelligence.
- **Settings (2):** Displays the list of configurations required for DX Operational Intelligence.
- **Help Link (3):** Launches the help documentation.
- **App Launcher (4):** Allows you to navigate to the different prodcuts.
- **Three Dot Menu (5):** Use this menu to set the capability page as a Landing page of DX Operational Intelligence.

Settings Page

The Settings Page in DX Operational Intelligence provides access to configuration options such as setting up data sources, providing connection parameters, and configuring monitoring.

The Settings view contains all the essential configurations to use DX Operational Intelligence seamlessly. It also includes other additional configurations to leverage the benefits of DX Operational Intelligence. The configurations that are mandatory for enabling DX Operational Intelligence are displayed in full-color cards format ( ), and the configurations that are optional but enhance your overall user experience in DX Operational Intelligence are displayed in semi-color cards ( ).

The following image illustrates the Settings page:



You can now use the following tiles to set up DX Operational Intelligence.

- **Setup Data Sources:** To ingest a wide variety of data such as metrics, events, alarms, logs, inventory groups, and so on, download the DX Gateway, Log Collector, NetOps Connector, Spectrum Data Publisher.
- **Setup Data Sources:** To ingest a wide variety of data such as metrics, events, alarms, logs, inventory groups, and so on, download RESTMon.
- **Connector Parameters**: Provide the mandatory parameters required for the integration to configure the connectors.
- **Configure Monitoring:** Use this tile to configure the monitoring capabilities such as anomaly detection, prediction, capacity planning, and so on.
- **Enable Predictive Definitions:** Enable and customize predictive data definitions for metrics.
- **Create Policies:** Define how events (including alarms) in the monitored environment are handled.
- **Define Log Alarms:** Define conditions in the monitored environment to generate alarms from logs.
- **Connect Ticketing System:** To enable lifecycle management between systems, connect with a ticketing system. For example, **ServiceNow.**
- **Connect Notification Channels:** Use this title to integrate with email, Slack, and other Webhook-based notification channels.
- **Create Message Templates:** Use this tile to create custom messages with system variables for all notification channels.
- **Create Maintenance Windows:** Define time periods for the maintenance to be performed on IT Operation entities.
- **Custom Situation Definitions:** Create and manage custom situations definitions for alarm clustering.
- **Manage Roles and Access:** Define custom roles and manage access to the application functionality and content.
- **Add Users:** Add, edit, and delete users and also assign roles.
- **Manage Universes:** Define and manage data access permissions for users.
- **Create Services:** Define monitored elements relationships to form services (LOBs, apps, networks, etc.)
- **Create SLIs & SLOs:** Create Service Level Indicators and Objectives to meet business goals.
- **Create Dashboards:** Assemble views of KPIs to reflect the state of your IT Operations

{"URL":["https://digital-oi/alarms-analytics/settings"],"customLabelGetStarted":"Settings","description":"concept.dita_729ed38b-3bc1-4927-8e7e-7a76a0a98a1e"}

Learning DX Operational Intelligence

Visit DX Operational Intelligence Academy to quickly level up your DX Operational Intelligence skills.

- Academy Courses
- Video Tutorials
- Blogs
- Communities

Academy Courses

Learn the following DX Operational Intelligence capabilities courses available in the Enterprise Software Academy.

For more information, see Enterprise Software Academy.

DX Operational Intelligence Overview

Course Name	Overview
DX Operational Intelligence: A Virtual Tour	In this virtual tour, first, you will identify the need for an AIOpssolution. You will learn the architecture of DX Operational Intelligence and its underlying benefits. Further, this tour will show you the built-in capabilities of DX Operational Intelligence as well as highlights the product integration with other Broadcom Products.

Integrations

Course Name	Overview
DX Operational Intelligence and DX UIM Integration	Integrate the DX Unified Infrastructure Management (DX UIM) with the DX Operational Intelligence (DX OI), to analyze UIM data and display it within the DX Operational Intelligence UI.
DX Operational Intelligence and DX NetOps Integration	This course provides you with the steps to integrate DX Operational Intelligence and DX NetOps.
DX Operational Intelligence and DX Spectrum Integration	This course provides you with on-premise integration steps for Spectrum and DX Operational Intelligence.
hird-Party Integrations with DX RESTmon	This course provides you with steps for third-party integration with DX RESTmon.

Dashboards

Course Name	Overview
DX Operational Intelligence : DX Dashboards 200	This course will help you learn set up, maintain, and perform simple troubleshooting on DX Dashboards.

Video Tutorials

Watch these video tutorials to learn how to perform specific actions in DX Operational Intelligence. You can subscribe to the DX Operational Intelligence Channel to learn more about the product.

- AIOps - DX Dashboards - What's New
- AIOps - DX Dashboards Troubleshooting
- Architecture Overview Refresher
- Anomaly Detection Improvements (Metric and Alarm Configuration)
- Anomaly Detection Troubleshooting
- Capacity Analytics - Whats New
- Capacity Analytics - Troubleshooting
- DX RESTMon Health Dashboards
- DX RESTMon - How it Works
- DX RESTMon - Liveness and Readiness Probes
- DX RESTMon - What's New
- ITSM Configuration support for Active Situations
- Monitored Inventory Capability
- Notifications - What's new
- Notifications - Troubleshooting
- Role-Based Access Control Enhancements
- Service Analytics - CRUD APIs
- Service Analytics - What's new

Blogs

You can view the latest blogs on DX Operational Intelligence capabilities here: DX Operational Intelligence Blogs.

Communities

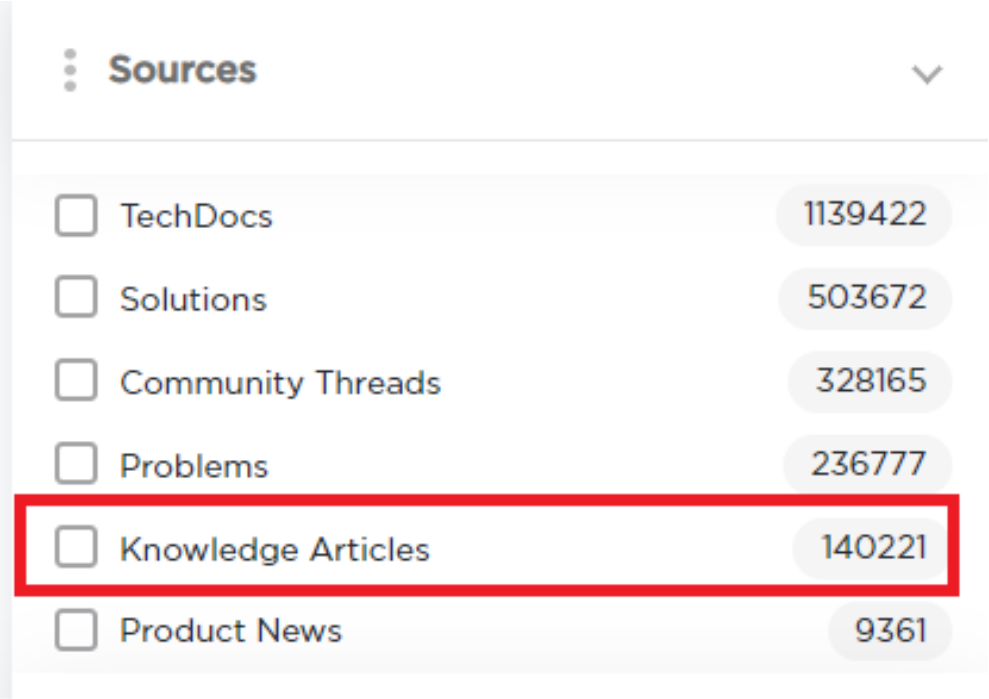
The DX Operational Intelligence communities provide the latest news, use cases, white papers, and analyst reports. For more information, see DX Operational Intelligence Communities.

Knowledge Base Articles

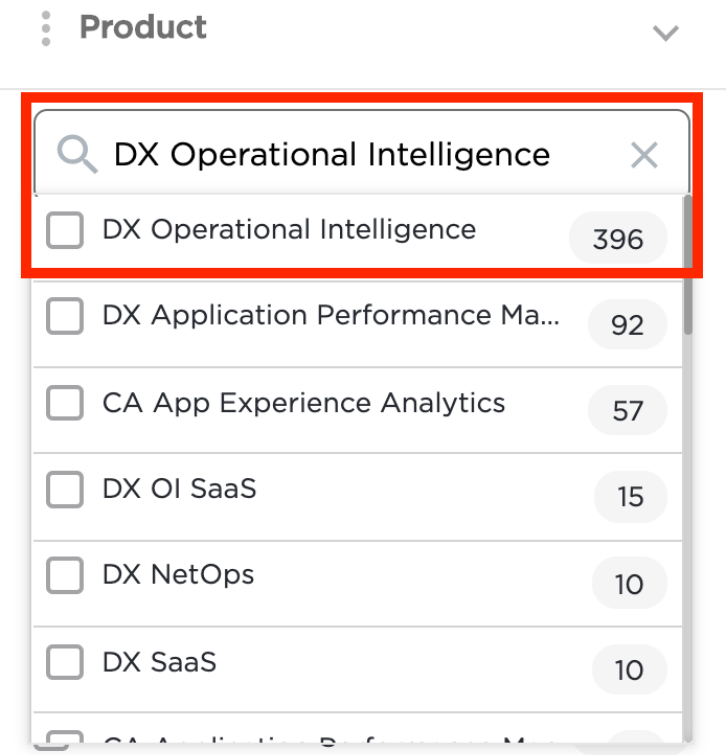
To view the full list of Knowledge Base articles for DX Operational Intelligence, click here.

Use the Advanced Search filters to narrow down your search criteria.

1. Select Knowledge Articles from the list of options available under **Sources.**

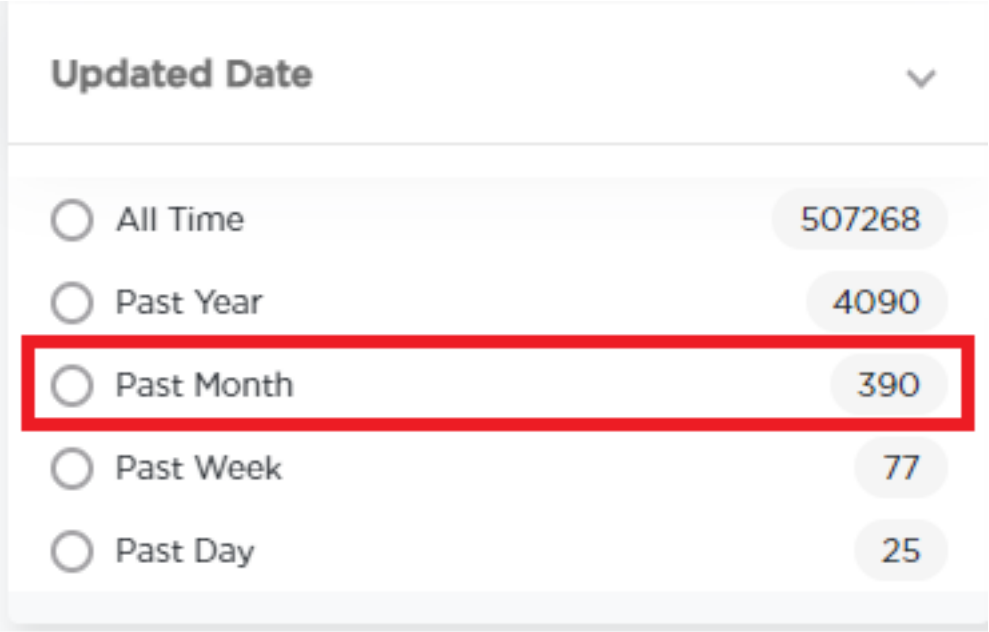


2. Based on your entitlement, search for your product and select your product from the **Product** options.



For example, search for **DX Operational Intelligence** and select the product from the list.

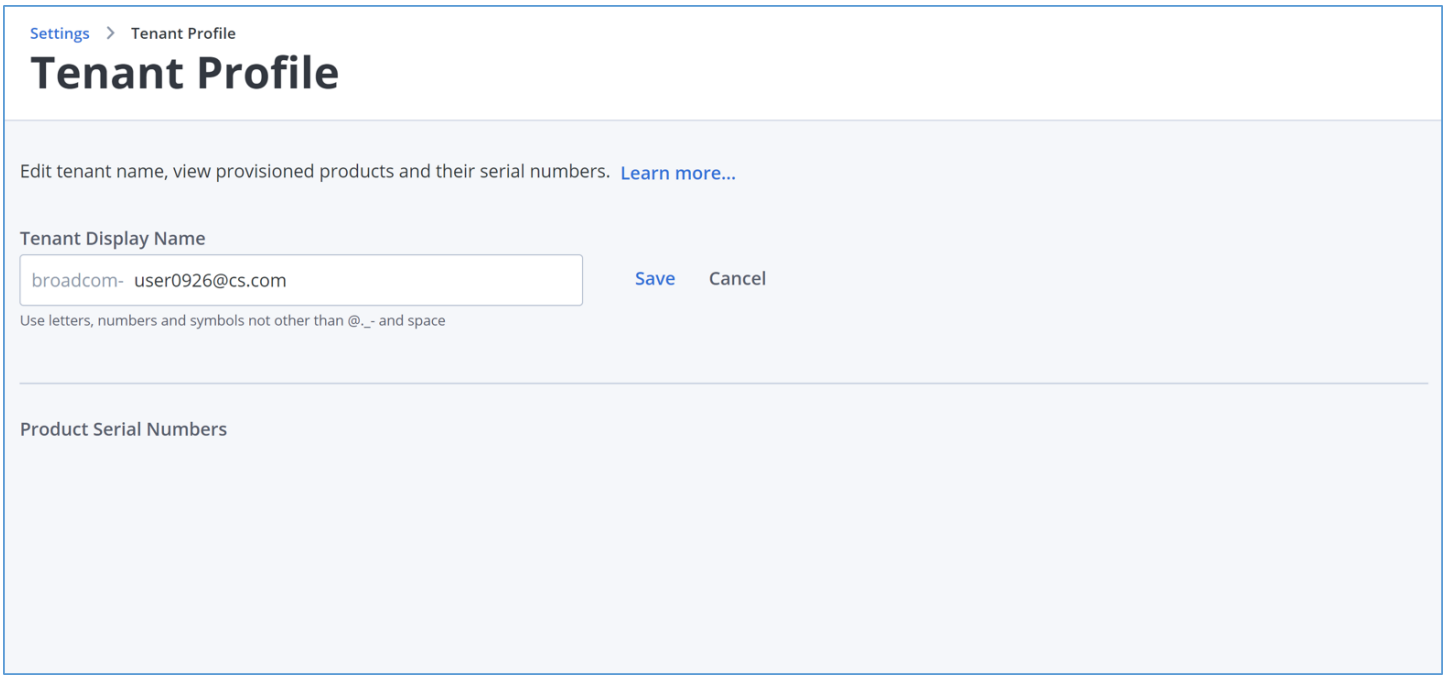
3. Select the required language.
4. Select the required duration from the **Updated Date** options.



The relevant knowledge articles for the specified filter criteria are displayed.

Tenant Profile Management

The **Tenant Profile** page displays the tenant display name, products that are provisioned for the tenant, and their serial numbers. On this page, you can edit the tenant display name, which has the following format: **<organizationnameprefix>- <name>**.

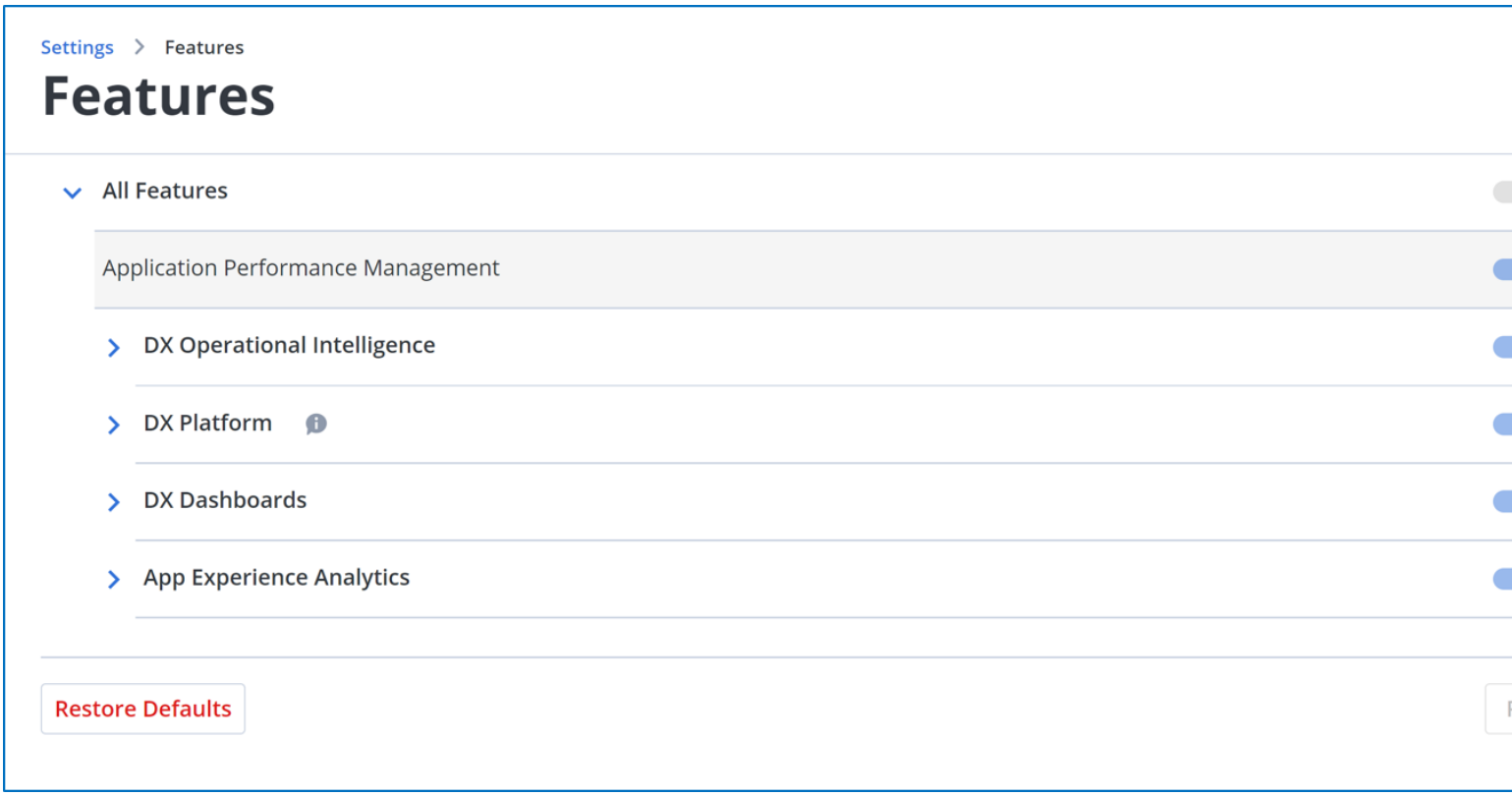


NOTE

- You can edit only **<name>** but not the **<organizationnameprefix>.**
- You can use alphanumerical and special characters.
- Only ampersand (@), dot (.), hyphen (-), space ( ), and underscore (_) are supported.

Features Management

The **Features** tile on the **Settings** page enables you to manage features at the tenant level. You can enable or disable the features for each of the capabilities as required.



NOTE
For DX APM, you can enable or disable all the features at once.

User and Role Management

DX Operational Intelligence supports custom roles to meet your requirement. The custom role is a collection of permissions that you add from the Roles page. You can create a custom role definition, and provide feature level and data level access to a role.

This section provides the following information:

- Role Management
- User Management

Role Management

{"URL":["https://cloudmanagement/#/settings/roles"],"description":"concept.dita_6101b304-6efb-418a-8a31-def3038894d6","new":"","new_video":"","admin":"","troubleshooting":{"masterkb":"","text":"","URL":[]},"pendo":"","video":[]}

A user role determines the access rights or privileges that a user has to the features of a product. DX Operational Intelligence uses Role-based Access Control (RBAC) to restrict access based on the roles of the individual users within the enterprise. RBAC lets users have access rights only to the tasks that they need to perform or have access to information that pertains to them.

DX Operational Intelligence provides out-of-the-box (OOTB) roles that you can assign to a user. These roles have a set of access privileges that are predefined, and that cannot be edited or deleted. However, to edit and customize the roles, you can create custom roles and define the access privileges to suit your organization's requirements. For example, you can create a custom role with the privileges of a Power User role which is the out-of-the-box role but with DX Dashboards access privileges set to Administrator.

> **NOTE**
> The OOTB roles have access to all the DX Operational Intelligence data in the tenant. This access is covered under Universe Management.

After you create the custom roles, you can manage the entire life cycle of the custom roles. For example, you can activate or deactivate a custom role, update or revoke the privileges when required, and also delete the role when you no longer need it.

> **NOTE**
> The RBAC implementation is independent of the authentication type that is enabled on the tenant. Currently, this implementation is at the per-tenant level and does not support sub-tenancy.

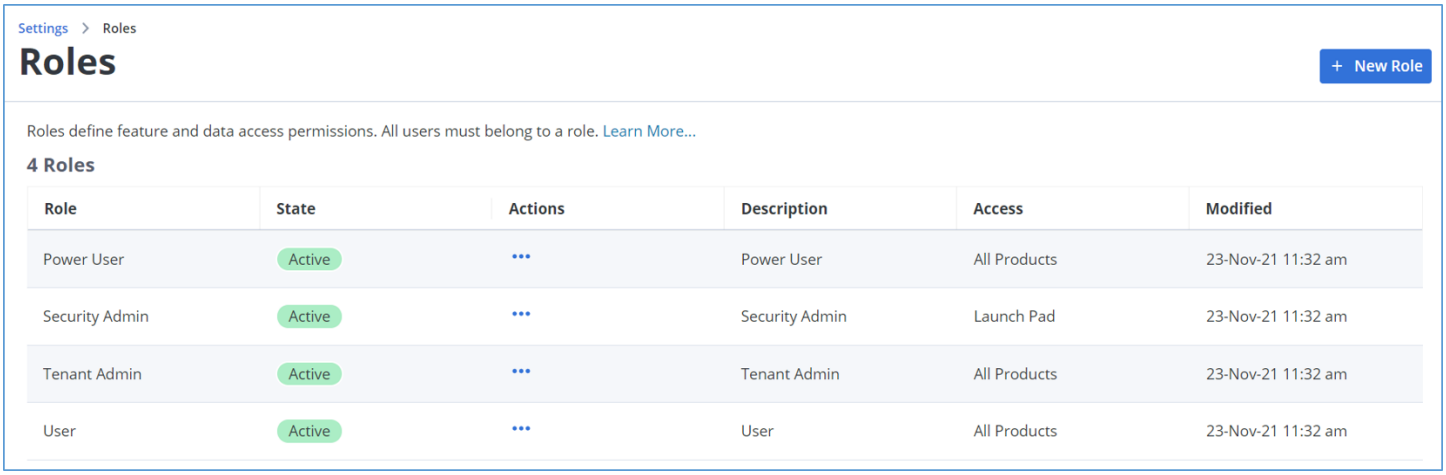
This section provides the following information:

- Supported User Roles and their Access Privileges
- Best Practices for RBAC Implementation
- Create and Manage Custom Roles
- Capabilities Access Privileges

Supported User Roles and their Access Privileges

DX Operational Intelligence provides the following roles out-of-the-box and each role has access privileges that are assigned by default. These roles are read-only. You can only view the access privileges, but you cannot edit them.

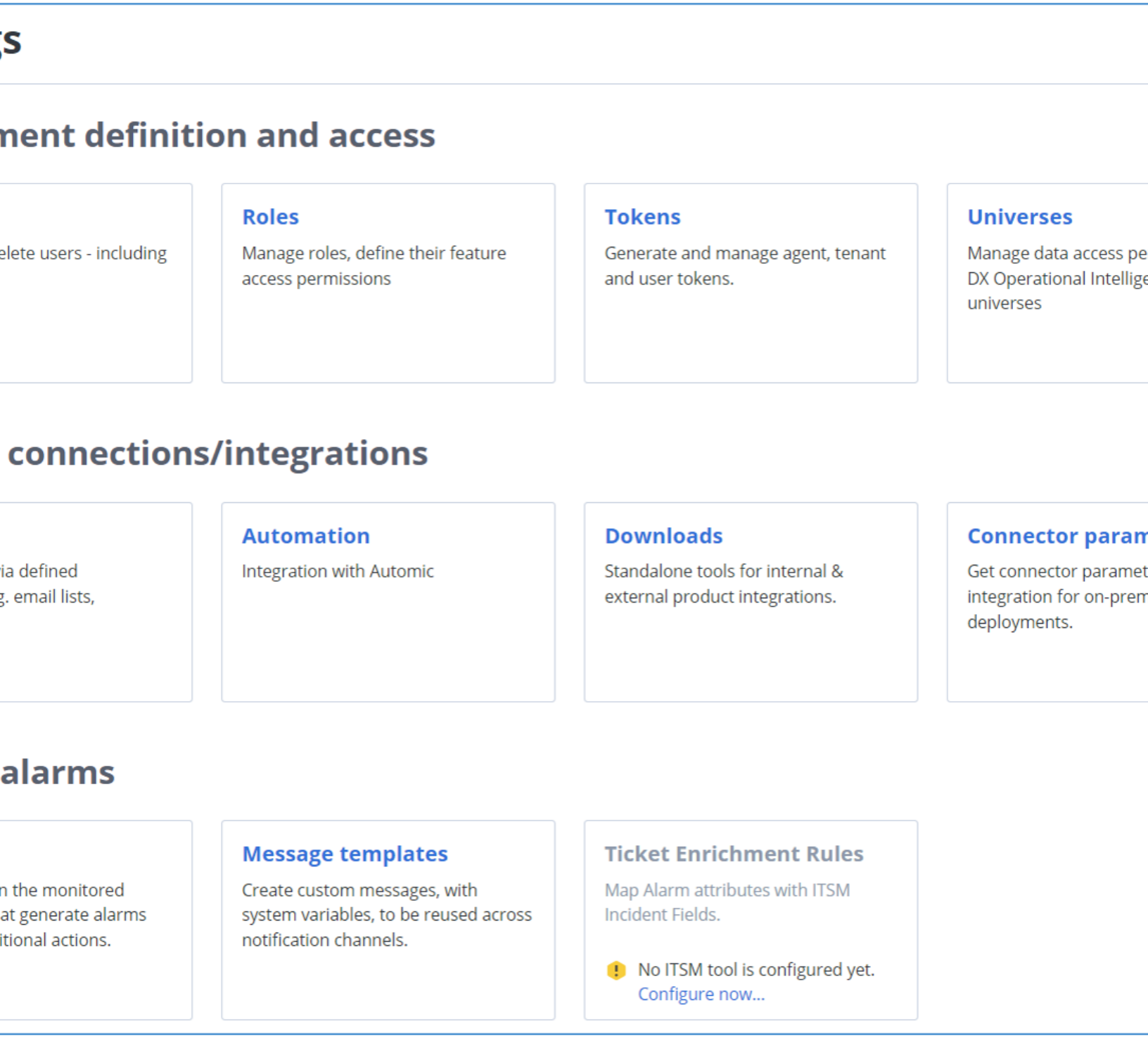
> **NOTE**
> Click the **Ellipsis** icon under **Actions** to view the access privileges.



- **Tenant Administrator:** A Tenant Administrator is at the highest level in the administrative hierarchy and can perform all administrative tasks for a tenant. A Tenant Administrator has access to all the products.
- **Power User:** A Power User is at the second level in the hierarchy and has access to all the products.
- **User:** Each employee is referred to as a user and has access to all the products.
- **Security Administrator:** A Security Administrator role enables you to administer the security authentication and authorization but does not have access to the product features. This role minimizes the ability of a privileged user from accessing the product functionality and subsequently data that would be difficult to detect even with auditing.

Tenant Administrator

The Tenant Administrator is at the highest level in the administrative hierarchy and is available out-of-the-box. A Tenant Administrator can perform all administrative tasks for a tenant.



The following section lists the feature access privileges that are assigned to this role by default:

NOTE
Alternatively, you can view these privileges on the Roles page.

Application Performance Management

A Tenant Administrator has access to all the features.

DX Operational Intelligence

The following table lists the access privileges for the features that are assigned by default to the Tenant Administrator role:

Feature	Access Privileges
Service Analytics	
Services To access alarms in the context of a service, provide access to the Alarms View privilege under Alarm Analytics.	• View Services • Create Or Update Service • Delete Service
Tenant Deprovision	• No Access
Alarm Analytics	
Alarm Views	• View Raw Alarms • View Service Alarms • View Situations Alarms • View Unassociated Alarms
Alarm Actions	• Clear • Acknowledge/Unacknowledge • Ticketing • Assign/Unassign • Hide/Unhide • Channel
Alarm Annotations	• View Alarm Annotation • Create Alarm Annotation
Alarm Filters	• View Alarm Filters • Create Or Update Alarm Filters • Delete Alarm Filters
Alarm Insights	• View Alarm Insights
Situations	• Situation Stable Window – View Situation Stable Window – Update Situation Stable Window • Situation Tenant Configuration – View Situation Tenant Configuration – Update Situation Tenant Configuration – Delete Situation Tenant Configuration • Situation Search Action – Situation Search Action
Performance Analytics	
Metric Browser Provide access to the monitored inventory to view Metric Browser.	• View Metric Browser
Performance Analytics Views	• View Views • Create or Update Views • Delete Views
Predictive Insights	• View Predictions
Capacity Analytics	

Feature	Access Privileges
View Capacity Analytics	• View Capacity Analytics
Configure Capacity Analytics	• Configure Services Provide View Services access. • Configure Groups
Monitored Inventory	• View Entities
Settings	
Log Alarm Definitions	• View Alarm Definitions • Create Alarm Definitions • Update Alarm Definitions • Delete Alarm Definitions
Gateways	• View Gateways • Create Gateway
Monitoring Groups	• View Monitoring Groups • Create Or Update Monitoring Groups • Delete Monitoring Groups
Service Level Indicator Provide access to View Services to access the Service Level indicator.	• View Service Level Indicator Provide access to View Monitoring Groups to view Service Level indicator. • Create or Update Service Level Indicator Provide access to Create or Update Monitoring Groups to Create or Update Service Level indicator. • Delete Service Level Indicator Provide access to Delete Monitoring Groups to Delete Service Level indicator.
Predictive Definitions	• View Predictive Definitions • Update Predictive Definitions
Maintenance	• View, Maintenance Schedules • Create Maintenance Schedules • Update Maintenance Schedules • Delete Maintenance Schedules
Policy Based Situations	• View Policy Based Situations • Create or Update Policy Based Situations • Delete Policy Based Situations
Config Attributes	• View config attributes • Create or Update config attibutes
Log Analytics	• Log Purge APIs • Log Retrieval APIs
Topology Correlation Settings	• View Correlation Rules • Modify Correlation Rules

DX Platform

The following table lists the access privileges that a Tenant Administrator has to the DX Platform features by default:

Feature	Access Privileges
Configuration	Global Configuration
Admin Management	View Administrators • Create Administrators • Update Administrators • Delete Administrators
Settings	Audit Trail • Connector Parameters • Downloads • Onprem Ticket Management APIs • Proxy Configuration
Automation	Integration with Automic • Feature Store Upload
Outbound Notification	View • Create, Edit, and Delete
Role Management	List Roles • Create Role
Token Management	Manage Tokens • Manage Tenant Tokens

Feature	Access Privileges
User Management	View Users • Create Users • Update Users • Delete Users • Reset Password • Update Authentication Configuration
Universal Management	Create Users • All DXOI Data in the Tenant

DX Dashboards

A Tenant Administrator has the access privileges of the Administrator role in DX Dashboards.

Feature	Access Privileges
Dashboards	• Create, Edit, and Delete Folders • Create, Edit, and Delete Reports • Create, Edit, and Delete Data Sources
Playlists	• View Playlists • Create, Edit, and Delete Playlists
Snapshots	• View Snapshots • Create, Delete Snapshots

App Experience Analytics

The following table lists the feature access privileges that are assigned by default to the Tenant Administrator role:

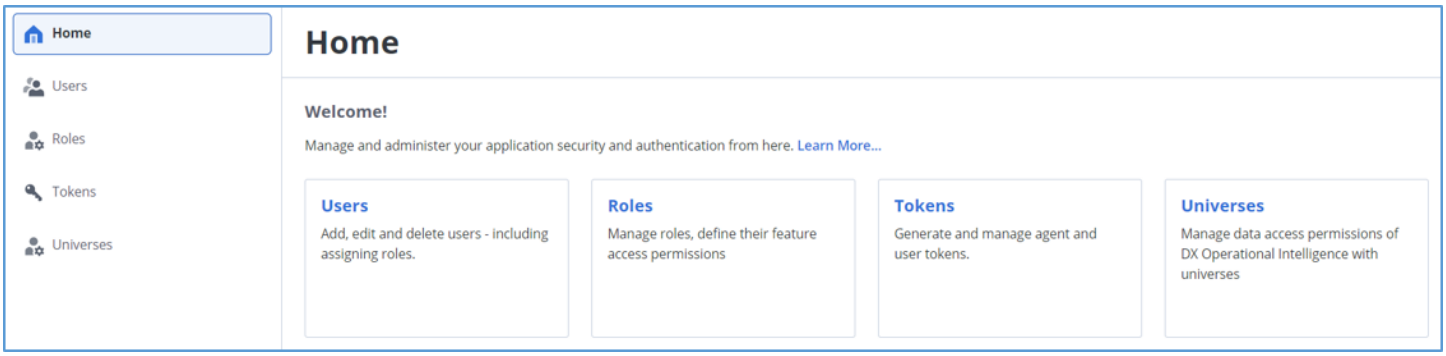
Feature	Access Privileges
Crash Handler APIs	• No access
Filters	• View Filters • Create Filters
Apps	• View Apps • Create Apps • Delete Apps

Feature	Access Privileges
Profiles	• View Profiles • Create Profiles • Update Profiles • Delete Profiles
Policies	• View Policies • Create Policies • Update Policies • Delete Policies
Protected MDO Admin APIs	• Has access

Security Administrator

A Security Administrator role enables a user to administer security authentication and authorization but does not give the user access to the product features. This role minimizes the ability of a privileged user to access the product functionality and, subsequently, data that would be difficult to detect even with auditing.

The Security Administrator role has access only to User Management, Roles Management, Tokens Management, and Universe Management. They do not have access to the other product features as shown:



A Security Administrator role has access only to the following DX Platform features by default:

Features	
Configuration	**Global - No Access**
Admin Management	• View Administrators • Create Administrators • Update Administrators • Delete Administrators
Settings	

Features	Privileges
Audit Trail, Connector Parameters, Downloads, Onprem Ticketing APIs, Proxy Configuration	**No Access**
Outbound Notifications	**View** - **No Access** • **Create/ Edit/ Delete - No Access**
Role Management	**List Roles - No Access** • Create Role
Token Management	Manage Tokens (Generate and Delete Agent Tokens and User Tokens) • **Manage Tenant Tokens - No Access**

Features	Privileges
User Management	View User Management • Create Users • Update Users • Delete Users • Reset Password • Update Authentication Configuration
Universal Management	Create Universal Management • All DX OI Data in the Tenant
Tenant Profile (View provisioned products and their serial numbers)	**Tenant Profile Access**

Power User

A Power User is at the next level in the hierarchy and has access to all the products. A Tenant Administrator or a Security Administrator can create a Power User.

Settings

Environment definition and access

Tokens

Generate and manage user permanent tokens.

Universes

Manage data access permissions of DX Operational Intelligence with universes

External connections/integrations

Channels

Communicate via defined integrations (e.g. email lists, ServiceNow)

Manage alarms

Policies

Set conditions in the monitored environment that generate alarms and trigger additional actions.

Message templates

Create custom messages, with system variables, to be reused across notification channels.

Ticket Enrichment Rules

Map Alarm attributes with ITSM Incident Fields.

No ITSM tool is configured yet.
Configure now...

The following section lists the feature access privileges that are assigned to this role by default:

NOTE
Alternatively, you can view these privileges on the Roles page.

Application Performance Management

A Power User has access to all the features in the product.

DX Operational Intelligence

The following table lists the features access privileges that are assigned by default to the Power User role:

Feature	Access Privileges
Service Analytics	
Services	• View Services To access alarms in the context of a service, provide access to the Alarms View privilege under Alarm Analytics. • Create Or Update Service - No Access • Delete Service - No Access
Tenant Deprovision	• No Access
Alarm Analytics	
Alarm Views	• View Raw Alarms • View Service Alarms • View Situations Alarms • View Unassociated Alarms - No Access
Alarm Actions	• Clear • Acknowledge/Unacknowledge • Ticketing • Assign/Unassign • Hide/Unhide • Channel
Alarm Annotations	• View Alarm Annotation • Create Alarm Annotation
Alarm Filters	• View Alarm Filters • Create or Update Alarm Filters • Delete Alarm Filters
Alarm Insights	• View Alarm Insights
Situations	• Situation Stable Window — View Situation Stable Window — Update Situation Stable Window - No Access • Situation Tenant Configuration — View Situation Tenant Configuration — Update Situation Tenant Configuration - No Access — Delete Situation Tenant Configuration - No Access • Situation Search Action — Situation Search Action
Performance Analytics	
Metric Browser Provide access to the monitored inventory to view Metric Browser.	• View Metric Browser
Performance Analytics Views	• View Views • Create or Update Views • Delete Views

Feature	Access Privileges
Predictive Insights	• View Predictions
Capacity Analytics	
View Capacity Analytics	
Configure Capacity Analytics	• Configure Services Provide View Services access. • Configure Groups
Monitored Inventory	• View Entities
Settings	
Log Alarm Definitions	• View Alarm Definitions • Create Alarm Definitions • Update Alarm Definitions • Delete Alarm Definitions - No Access
Gateways	• View Gateways • Create Gateways - No Access
Monitoring Groups	• View Monitoring Groups • Create or Update Monitoring Groups - No Access • Delete Monitoring Groups - No Access
Service Level Indicator Provide access to View Services to access the Service Level indicator.	• View Service Level Indicator Provide access to View Monitoring Groups to view Service Level indicator. • Create or Update Service Level Indicator - No Access • Delete Service Level Indicator - No Access
Predictive Definitions	• View Predictive Definitions • Update Predictive Definitions - No Access
Maintenance	• View Maintenance Schedules • Create Maintenance Schedules - No Access • Update Maintenance Schedules - No Access • Delete Maintenance Schedules - No Access
Policy Based Situations	• View Policy Based Situations - No Access • Create or Update Policy Based Situations - No Access • Delete Policy Based Situations - No Access
Config Attributes	• View config attributes - No Access • Create or Update config attributes - No Access
Log Analytics	• Log Purge APIs - No Access • Log Retrieval APIs - No Access
Topology Correlation Settings	• View Correlation Rules - No Access • Modify Correlation Rules - No Access

DX Platform

The following table lists the access privileges that a Power User has to the DX Platform features by default:

Feature	Access Privileges
Configuration	View Configuration - No Access
Admin Management	View Administrators - No Access • Create Administrators - No Access • Update Administrators - No Access • Delete Administrators - No Access
Settings	
Audit Trail, Connector Parameters, Downloads, Onprem Ticketing APIs, Proxy Configuration	No Access
Outbound Notifications	View • Create/ Edit/ Delete - No Access
Role Management	List Roles - No Access • Create Role - No Access

Access Feature Privileges	
Tokens Management	• Manage Tokens - No Access • Manage Tenant Tokens - No Access
User Management	• View Users - No Access • Create Users - No Access • Update Users - No Access • Delete Users - No Access • Reset Password - No Access • Update Authentication Configuration - No Access
Universe Management	• View OXOI Data in the Tenant • Create Universe - No Access

DX Dashboards

A Power User has the access privileges of the Editor role in DX Dashboards.

Feature	Access Privileges
Dashboards	• Create, Edit, Delete Dashboards • Import, Export Dashboards • View Dashboards
Folders	• Create, Edit, and Delete Folders
Reports	• Create, Edit, and Delete Reports
Playlists	• Create, Edit, and Delete Playlists • View Playlists
Snapshots	• Create, and Delete Snapshots • View Snapshots

App Experience Analytics

The following table lists the feature access privileges that are assigned by default to the Power User role:

Feature	Access Privileges
Crash Handler APIs	• No access
Filters	• View Filters • Create Filters
Apps	• View Apps • Create Apps • Delete Apps
Profiles	• View Profiles • Create Profiles • Update Profiles • Delete Profiles
Policies	• View Policies • Create Policies • Update Policies • Delete Policies
Protected MDO Admin APIs	• Has Access

User

Each employee of the enterprise is referred to as a user. A user record exists either in the enterprise LDAP repository or in the DX Operational Intelligence database. Using the Administration Console, a Tenant Administrator can create and manage users in the DX Operational Intelligence database. A Tenant Administrator can also manage the DX Operational Intelligence-specific data of users whose records are in the LDAP repository. This data is stored in the DX Operational Intelligence database. Power Users and Users cannot access the User Management page. Users can be created only in the DX Operational Intelligence database.

The following section lists the feature access privileges that are assigned to this role by default:

Application Performance Management

A user with the user role has access to all the features of the product.

DX Operational Intelligence

The following table lists the features access privileges that are assigned by default to the User role:

Feature	Access Privileges
Service Analytics	
Services To access alarms in the context of a service, provide access to the Alarms View privilege under Alarm Analytics.	• View Services • Create or Update Service - No Access • Delete Service - No Access
Tenant Deprovision	• No Access
Alarm Analytics	
Alarm Views	• View Raw Alarms • View Service Alarms • View Situations Alarms • View Unassociated Alarms - No Access
Alarm Actions	• Clear - No Access • Acknowledge/Unacknowledge - No Access • Ticketing - No Access • Assign/Unassign - No Access • Hide/Unhide - No Access • Channel - No Access
Alarm Annotations	• View Alarm Annotation • Create Alarm Annotation - No Access
Alarm Filters	• View Alarm Filters • Create or Update Alarm Filters • Delete Alarm Filters
Alarm Insights	• View Alarm Insights
Situations	• Situation Stable Window — View Situation Stable Window — Update Situation Stable Window - No Access • Situation Tenant Configuration — View Situation Tenant Configuration — Update Situation Tenant Configuration - No Access — Delete Situation Tenant Configuration - No Access • Situation Search Action — Situation Search Action
Performance Analytics	
Metric Browser	• View Metric Browser Provide access to the monitored inventory to view Metric Browser.
Performance Analytics Views	• View Views • Create or Update Views • Delete Views
Predictive Insights	• View Predictions
Capacity Analytics	

Feature	Access Privileges
View Capacity Analytics	• View Capacity Analytics
Configure Capacity Analytics	• Configure Services - No Access Provide View Services access. • Configure Groups - No Access
Monitored Inventory	• View Entities
Settings	
Log Alarm Definitions	• View Alarm Definitions - No Access • Create Alarm Definitions - No Access • Update Alarm Definitions - No Access • Delete Alarm Definitions - No Access
Gateways	• View Gateways - No Access • Create Gateways - No Access
Monitoring Groups	• View Monitoring Groups - No Access • Create or Update Monitoring Groups - No Access • Delete Monitoring Groups - No Access
Service Level Indicator	• View Service Level Indicator - No Access Provide access to View Monitoring Groups to view Service Level indicator. • Create or Update Service Level Indicator - No Access • Delete Service Level Indicator - No Access
Predictive Definitions	• View Predictive Definitions - No Access • Update Predictive Definitions - No Access
Maintenance	• View Maintenance Schedules - No Access • Create Maintenance Schedules - No Access • Update Maintenance Schedules - No Access • Delete Maintenance Schedules - No Access
Policy Based Situations	• View Policy Based Situations - No Access • Create or Update Policy Based Situations - No Access • Delete Policy Based Situations - No Access
Config Attributes	• View config attributes - No Access • Create or Update config attributes - No Access
Log Analytics	
Log Analytics	• Log Purge APIs - No Access • Log Retrieval APIs - No Access
Topology Correlation Settings	
Topology Correlation Settings	• View Correlation Rules - No Access • Modify Correlation Rules - No Access

DX Platform

A user role does not have access to any of the DX Platform features.

Feature	Access Privileges
Configuration	• View Administrators - No Access • Create Administrators - No Access • Update Administrators - No Access • Delete Administrators - No Access
Admin Management	• View Administrators - No Access • Create Administrators - No Access • Update Administrators - No Access • Delete Administrators - No Access

Access Feature Privileges

Settings

- Audit Trail - No Access
- Connector Parameters - No Access
- Downloads - No Access
- Onprem Ticket Management APIs - No Access
- Proxy Configuration - No Access
- Outbound Notifications - No Access
- Role Management - No Access
- Token Management - No Access
- User Management - No Access
- Universe Management
  - All DXOI data in the tenant
  - Create Universe - No Access

Feature	Access Privileges
Tenant Profile (View provisioned products and their serial numbers)	• Tenant Profile View • Access

DX Dashboards

A User has only read-only access to DX Dashboards.

Feature	Access Privileges
Dashboards	• Export Dashboards • View Dashboards
Playlists	• View Playlists
Snapshots	• View Snapshots • Create Snapshots

App Experience Analytics

The following table lists the feature access privileges that are assigned by default to the User role:

Feature	Access Privileges
Crash Handler APIs	• No Access
Filters	• View Filters • Create Filters
Apps	• View Apps • Create Apps - No Access • Delete Apps - No Access
Profiles	• View Profiles • Create Profiles - No Access • Update Profiles - No Access • Delete Profiles - No Access
Policies	• View Policies • Create Policies - No Access • Update Policies - No Access • Delete Policies - No Access
Protected MDO Admin APIs	• No Access

Best Practices for RBAC Implementation

Consider the following points while implementing RBAC:

- Administration

  - You may use the Security Administrator role to administer roles, tokens, and users.
  - DX Operational Intelligence supports more than 20 custom roles. However, for best performance, a maximum of 20 custom roles is recommended.
  - Minimize the role administration when the roles are likely being used by the logged-in users.
    - **Adding privileges to a role:** When you add privileges to a logged-in user, the user may have to log off and log back in to gain access to the new features that were added.

      NOTE
      To gain access to new actions that are added for **Alarm Actions**, you must provide access to the Alarm Actions for the custom role.
    - **Removing privileges from a role:** When you revoke privileges from a role that is likely being used by the logged-in users, attempting to access the unauthorized features results in error. The user may have to log off and log back in for the previously and no longer authorized features to be suppressed.
  - To control access to each product, select the product features explicitly.
- SAML-Enabled Tenants
  - The RBAC implementation design includes the SAML user groups. The best practice is to assign a user to a single SAML user group and map that user group to a single custom role.
  - A user can implicitly inherit more than one custom role by being part of more than one SAML user group. Review the role responsibilities and create the custom roles to possibly avoid sharing the responsibilities. Then map the SAML user groups to the custom roles.
  - When a user is part of multiple SAML groups, that user has all the privileges that are available to the groups.

Create and Manage Custom Roles

DX Operational Intelligence supports custom roles to meet your requirement. The custom role is a collection of permissions that you add from the predefined list.

You can create a custom role definition and assign feature level access to a role using the Roles page.

This section lists the following topics:

- Create Custom Role
- Edit Custom Role
- Copy Custom Role
- Deactivate Custom Role
- Delete Custom Role

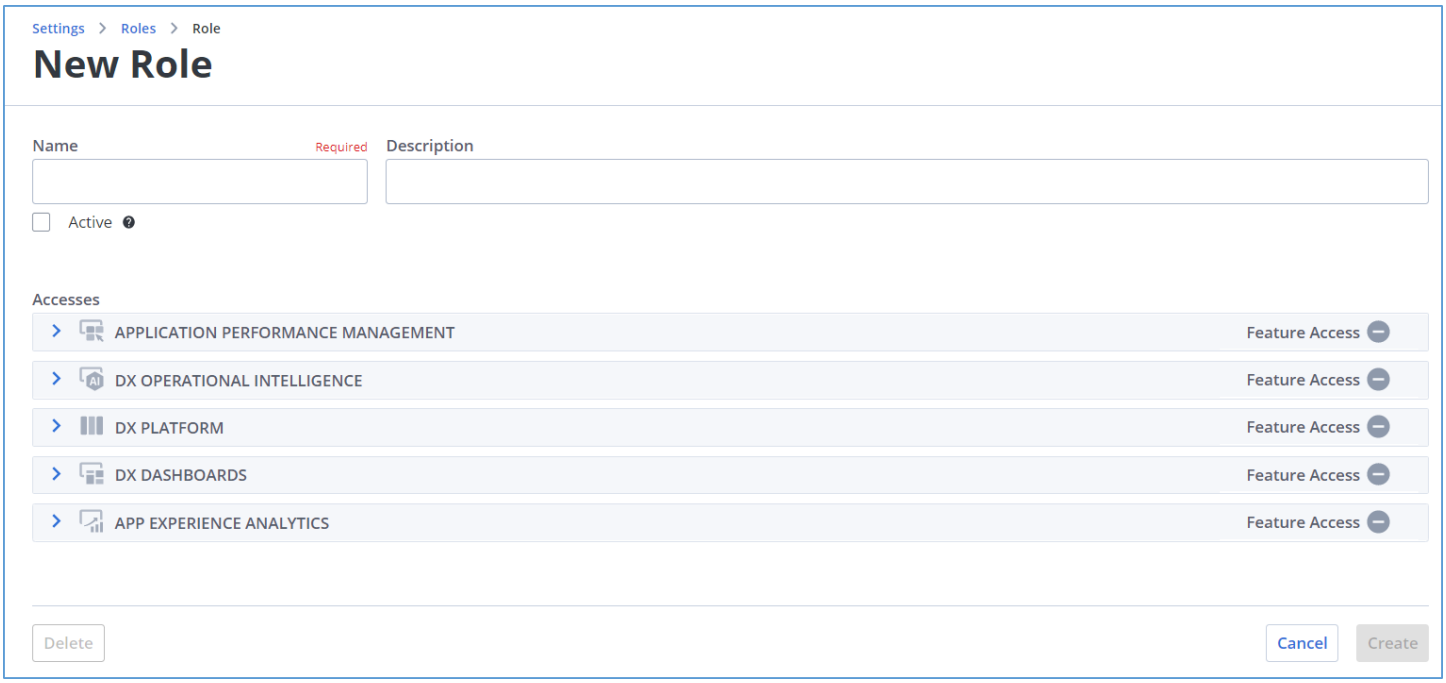
Create Custom Role

{"URL":["https://cloudmanagement/#/settings/roles/role/create"],"description":"task.dita_03029c57-5477-405f-92e6-0ca65a40fdaa","new":"","new_video":"","admin":"","troubleshooting":{"masterkb":"","text":"","URL":[]},"pendo":"","video":[]}

As a Tenant Administrator or a Security Administrator, you can create a custom role by selecting the features that the role must have access to. By default, a custom role does not have access to the DX Operational Intelligence data in the tenant. To grant access, click the **Give access to all data** link and select the **All DXOI data in the tenant** option under **Universe Management**.

When you select some of the features, the dependent features are automatically selected and disabled. For example, when you select **Create/Edit/Delete** under **Outbound Notifications,** the **All DX OI data in the tenant** option under **Universe Management** is selected automatically. If you do not want this role to have access to the DX Operational Intelligence data, then review your selection.

The following image illustrates the New Role page:



Follow these steps:

1. Login to DX Operational Intelligence as a Tenant Administrator.
2. Click **Settings** in the left navigation pane.
3. Click on the **Roles** tile.
   The Roles page appears.
4. Click **+ New Role**.
   The New Role page appears.
5. Enter the following information:
   a) Name: Enter a name for the custom role.
   b) Description: Enter a description for the custom role.
   c) Active: Select to activate this custom role on creation. You can assign only active custom roles to a user. A user can access the features only if the associated custom role is active.
   d) (Only for SAML-Based Tenants) SAML Groups: Specify the SAML groups that you have configured in the SAML IdP. Ensure that you enter the exact group name as configured in the IdP.
   e) Select the required access level for the features.
6. Click **+ Create**.
   The custom role is created. Once the custom role is created, you can edit, delete, and activate or deactivate that role.

Edit Custom Role

Perform the following steps to edit a role. You cannot edit the role name.

> **NOTE**
> If the custom role has access to all the DX Operational Intelligence data in the tenant for the selected features, a message is displayed. To remove access, click the **Remove access to all data** link and deselect the **All DXOI Data in the tenant** option under**Universe Management**.

Follow these steps:

1. Login to DX Operational Intelligence as a Tenant Administrator.
2. Click **Settings** in the left navigation pane.
3. Click on the **Roles** tile.
   The Roles page appears.
4. Click on the **Ellipsis** icon.
   Options to manage the custom role are displayed.
5. Click on **Edit**.
   The **Edit** page appears.
6. Update the required changes.
7. Click **Save**.
   The changes are updated.

Copy Custom Role

This section describes how to make a copy of a custom role. When you copy a custom role, the copied role inherits all the privileges of the base custom role. You can then add or remove the privileges as required.

Follow these steps:

1. Login to DX Operational Intelligence as a Tenant Administrator.
2. Click **Settings** in the left navigation pane.
3. Click on the **Roles** tile.
   The Roles page appears.
4. Click the **Ellipsis** icon for the custom role that you want to copy.
   The different options are displayed.
5. Select **Make a Copy**. Alternatively, you can click **Edit** to open the custom role and then click **Make a Copy**.
   The **New Role** page is displayed.
6. Edit the role as required.
7. Click **+ Create**.
   This new custom role is added to the roles list.

Deactivate Custom Role

As a Tenant Administrator or a Security Administrator, you can deactivate a role.

Follow these steps:

1. Login to DX Operational Intelligence as a Tenant Administrator.
2. Click **Settings** in the left navigation pane.
3. Click on the **Roles** tile.
   The Roles page appears.
4. Click on the **Ellipsis** icon.
   Options to manage the custom role are displayed.

5. Click on **Deactivate**.
   The role is deactivated and the status is changed to **Inactive**.

Delete Custom Role

As a Tenant Administrator or a Security Administrator, you can delete a role.

Follow these steps:

1. Login to DX Operational Intelligence as a Tenant Administrator.
2. Click **Settings** in the left navigation pane.
3. Click on the **Roles** tile.
   The Roles page appears.
4. Click on the **Ellipsis** icon.
   Options to manage the custom role are displayed.
5. Click on **Delete**.
   The role is deleted.

Capabilities Access Privileges

This section describes the access privileges for the following capabilities.

This section provides the access privileges for the following capabilities:

- DX Operational Intelligence
- DX Platform
- DX Dashboards

DX Operational Intelligence

This section provides information about the level of access each user has to the features by products:

- Service Analytics Access Privileges
- Alarm Analytics Access Privileges
- Performance Analytics Access Privileges
- Predictive Insights Access Privileges
- Capacity Analytics Access Privileges
- Monitored Inventory Access Privileges
- Topology Correlation Settings Access Privileges
- Settings Access Privileges

Service Analytics Access Privileges

Learn about various privileges and descriptions of Service Analytics.

The **Create Services** card on the **DX Operational Intelligence Settings** page gets enabled after you provide access to Service Analytics privileges on the **Roles** page.

Privileges	Description	Auto-Selected Privileges
View Services	Allows access to view the list of services.	Alarm Analytics > Alarm Views • View Raw Alarms • View Service Alarms • View Situation Alarms
Create Or Update Service	Allows access to create a service or update an existing service.	• Services > View Services • Alarm Analytics > Alarm Views — View Raw Alarms — View Service Alarms — View Situation Alarms
Delete Service	Allows access to delete a service.	• Services — View Services — Create Or Update Service • Alarm Analytics > Alarm Views — View Raw Alarms — View Service Alarms — View Situation Alarms

NOTE

- To view the widgets on the Service Details page, provide access to the following privileges:

Widgets on Service Details Page	Required Privileges
Capacity Issues	**View Capacity Analytics** under Capacity Analytics
Predictions	**View Predictions** under Predictive Insights
Monitored Inventory	**View Entities** under Monitored Inventory
SLI Metrics and SLOs	**View Service Level Indicator** under Service Level Indicator
Add Maintenance Window	Provide access to Maintenance privileges on the Settings page

Alarm Analytics Access Privileges

Learn about various privileges and descriptions of Alarm Analytics.

Privileges	Description	Auto-Selected Privileges
Alarm Views		
View Raw Alarms	Allows access to view all alarms.	
View Service Alarms	Allows access to view all the service alarms.	View Raw Alarms
View Situation Alarms	Allows access to view the situation alarms.	View Raw Alarms
View Unassociated Alarms	Allows access to view alarms that are not associated with any service.	View Raw Alarms

Privileges	Description	Auto-Selected Privileges
Alarm Actions		
• Clear • Acknowledge/Un-Acknowledge • Ticketing • Assign/Un-Assign • Hide/Un-Hide • Channel	Allows access to perform these actions on the alarms.	
Alarm Annotations		
View Alarm Annotation	Allows access to view the annotation details such as details of the person who added the annotation, the date, and the time.	
Create Alarm Annotation	Allows access to add any additional information or details for an alarm in the Annotation tab. When you select this privilege, View is automatically selected.	View Alarm Annotation
Alarm Filters		
View Alarm Filters	Allows access to filter the alarm data using the saved filters.	
Create or Update Alarm Filters	Allows access to create and update the filters on alarms. When you select this privilege, View is automatically selected.	View Alarm Filters
Delete Alarm Filters	Allows access to delete a saved filter. When you select this privilege, View is automatically selected.	View Alarm Filters
Alarm Insights		
View Alarm Insights	Allows access to view insights for all alarms based on a given time frame.	
Situations		
Situation Stable Window	**View Situation Stable Window:** Allows access to view the Situation Window settings for the Situations view.	• View Raw Alarms • View Situation Alarms
	Update Situation Stable Window: Allows access to update the Situation Window settings for the Situations view.	• View Situation Stable Window • View Raw Alarms • View Situation Alarms
Situation Tenant Configuration	**View Situation Tenant Configuration:** Allows access to view the algorithmic clustering.	• View Raw Alarms • View Situation Alarms
	Update Situation Tenant Configuration: Allows access to update the algorithmic clustering.	• View Situation Tenant Configuration • View Raw Alarms • View Situation Alarms
	Delete Situation Tenant Configuration: Allows access to delete the algorithmic clustering.	• View Situation Tenant Configuration • View Raw Alarms • View Situation Alarms

Privileges	Description	Auto-Selected Privileges
Situation Search Action	Allows access to search alarm actions.	• View Raw Alarms • View Situation Alarms

Performance Analytics Access Privileges

Learn about various privileges and descriptions of Performance Analytics.

Privileges	Description	Auto-Selected Privileges
Metric Browser		
View Metric Browser	Allows access to select entities, metrics, and configuration items.	Monitored Inventory > View Entities
Performance Analytics Views		
View Views	Allows access to select/view the saved views.	
Create or Update Views	Allows access to create a view by selecting the required metrics from the metric browser.	Performance Analytics Views > View Views
Delete Views	Allows access to delete an existing view.	Performance Analytics Views > View Views

Predictive Insights Access Privileges

Learn about various privileges and descriptions of Predictive Insights.

Privileges	Description
View Predictions	Allows access to the Predictive Insights landing page.

NOTE
The user can view the performance and capacity alarm predictions in the Predictive Insights view only when the following configurations are available:

- Metric threshold definitions for configured services and groups in Capacity Analytics.
- Metric threshold definitions to generate performance alarms.

Capacity Analytics Access Privileges

Learn about various privileges and descriptions of Capacity Analytics.

Privileges	Description	Auto-Selected Privileges
View Capacity Analytics	Allows view access to the Capacity Analytics landing page.	By default, the users can view only the configured groups chart. The user must have access to **View Services** privilege to view the other charts on the Capacity Analytics landing page.
Configure Capacity Analytics	Allows access to configure both services and groups.	• Service Analytics > View Services • Alarm Views – View Raw Alarms – View Service Alarms – View Situation Alarms
Configure Services	Allows access to configure services and the associated metrics for capacity analytics.	• Service Analytics > View Services • Alarm Views – View Raw Alarms – View Service Alarms – View Situation Alarms
Configure Groups	Allows access to configure groups and the associated metrics for capacity analytics.	

Monitored Inventory Access Privileges

Privileges	Description
Monitored Inventory	
View Entities	Allows access to view all the entities.

Topology Correlation Settings Access Privileges

Learn about various privileges and descriptions of topology correlation.

Privileges	Description	Auto-Selected Privileges
View Correlation Rules	Allows access to view the correlation rules.	
Modify Correlation Rules	Allows access to modify the existing correlation rules.	View Correlation Rules

Settings Access Privileges

This section provides the following information:

- Log Alarm Definitions
- Gateways
- Monitoring Groups
- Service Level Indicator Access Privileges
- Predictive Definitions
- Maintenance
- Policy Based Situations
- Config Attributes

Log Alarm Definitions

Learn about various privileges for log alarm definitions.

Privileges	Description	Auto-Selected Privileges
View Alarm Definitions	Allows access to view the alarm definitions.	
Create Alarm Definitions	Allows access to create the alarm definitions.	View Alarm Definitions
Update Alarm Definitions	Allows access to update the alarm definitions.	View Alarm Definitions
Delete Alarm Definitions	Allows access to delete the alarm definitions.	View Alarm Definitions

Gateways

Learn about various privileges of Gateways.

Privileges	Description
View Gateways	Allows access to view the configured gateways.
Create Gateway	Allows access to configure gateways to connect with the source products.

Monitoring Groups

Learn about various privileges of Monitoring Groups.

Privileges	Description
View Monitoring Groups	Allows access to view the monitored groups.
Create Or Update Monitoring Group	Allows access to create or update the monitoring group.
Delete Monitoring Group	Allows access to delete the required monitoring group.

NOTE

To create a new Notification policy for Anomaly alerts in the Monitoring Groups, you must have access to the following privileges:

- 'Create Or Update Monitoring Groups' under Monitoring Groups
- 'Create/Edit/Delete' under Outbound Notifications in DX Platform
- 'View Raw Alarms' under Alarm Analytics
- 'View Alarm Filters' under Alarm Analytics

Service Level Indicator Access Privileges

Learn about various privileges of Service Level Indicator.

NOTE

- You must provide access to 'View Services' under Service Analytics to access Service Level Indicator.
- For more information, see Service Level Indicator and Service Level Objectives.

Privileges	Description	Mandatory privileges to be provided
View Service Level Indicator	Allows access to view service level indicator	'View Monitoring Groups' under Monitoring Groups
Create Or Update Service Level Indicator	Allows access to create or update the service level indicator	'Create Or Update Monitoring Groups' under Monitoring Groups
Delete Service Level Indicator	Allows access to delete a service level indicator	'Delete Monitoring Groups' under Monitoring Groups

Predictive Definitions

Learn about various privileges provided for Predictive Definitions.

Privileges	Description
View Predictive Definitions	
Update Predictive Definitions	

Maintenance

Learn about various privileges provided for Maintenance.

Privileges	Description
View Maintenance Schedules	Allows access to view the maintenance schedules.
Create Maintenance Schedule	Allows access to create a maintenance schedule for service.
Update Maintenance Schedule	Allows access to update the existing maintenance schedule.
Delete Maintenance Schedule	Allows access to delete the maintenance schedules which are not required.

Policy Based Situations

Learn about various privileges for policy-based situations.

Privileges	Description	Auto-Selected Privileges
View Policy Based Situations	Allows access to view the policy-based situations.	
Create or Update Policy Based Situations	Allows access to create the policy-based situations.	View Policy Based Situations
Delete Policy Based Situations	Allows access to update the policy-based situations.	View Policy Based Situations

Config Attributes
Learn about various privileges for configuration attributes.

Privileges	Description	Auto-Selected Privileges
View config attributes	Allows access to view the configuration attributes.	
Create or Update config attributes	Allows access to create or update the configuration attributes.	View config attributes

DX Platform

The following table lists the access privileges to the DX Platform features that you can assign to the custom role:

Feature	Access Privileges
Settings	View • Audit Trail • Connector Parameters • Downloads • Onprem Ticket Management APIs • Proxy Configuration
Outbound Notification	View Create, Edit, and Delete
Universe Management	View DXOI data in the tenant

DX Dashboards

This section provides the list of privileges that are inherited when you select the role:

Tenant Administrator

A Tenant Administrator has the privileges that an Administrator role in DX Dashboards has. The Administrator role has the following access privileges to the features assigned by default:

Feature	Access Privileges
Dashboards	• Create, Edit, and Delete Folders • Create, Edit, and Delete Reports • Create, Edit, and Delete Data Sources
Playlists	• View, Create, Edit, and Delete Playlists
Snapshots	• View, Create, and Delete Snapshots

Security Admin

A Security Administrator does not have access to any DX Dashboards features.

Power User (Editor)

A Power User has the privileges that an Editor role in DX Dashboards has. The Editor role has the following access privileges to the features assigned by default:

Feature	Access Privileges
Dashboards	• Create, Edit, and Delete Folders • Create, Edit, and Delete Reports
Playlists	• View, Create, Edit, and Delete Playlists
Snapshots	• View, Create, and Delete Snapshots

User

A User role has only read-only access to the DX Dashboards features except for the following features that are assigned by default:

Feature	Access Privileges
Dashboards	• Export and View Dashboards
Playlists	• View Playlists
Snapshots	• View and Create Snapshots

OI Universes

Using a universe, a Tenant Administrator or a Security Administrator can define the data access permissions for users to control what data they have access to. For example, a Tenant Administrator can create a universe consisting of only the DX Operational Intelligence services that a user must have access to and assign that universe to the SAML group or the users that require restricted data access. With access being restricted, users will be able to view data only for the universe (or services in that universe) they are assigned to.

The **Select Universe** section lists the universes that a user is assigned to. A user can select the universe for which they want to view the data from this list.

By default, users with the out-of-the-box roles are provisioned with the **All Access** universe that enables them to view all the DX Operational Intelligence data in the tenant. For users with custom roles, the administrator must specifically add the

Universe Management privilege for the **All Access** universe to be provisioned. This privilege is available under the **DX Platform > Universe Management** section on the **Roles** page.

A Tenant Administrator or Security Administrator has the following privileges:

- Create a universe
- View the list of all the universes in the tenant. That is, even universes created by other Tenant Administrators and Security Administrators
- By default, edit and manage the universes in the following ways:
  - Edit the name and description of the universe
  - Add or remove users or SAML groups from the universe
  - Add or remove services from the universe
  - Delete, Deactivate, or Activate any universe in the tenant
- However, they cannot edit the Universe Management data access permissions for the out-of-the-box roles on the **Roles** page.

A Power User, User, and user with Custom Role have the following privileges:

- View the list of universes they are assigned to
- Edit and manage the universe if the **Can manage** option for users assigned to that universe is selected on the **Universes** page:
  - Edit the name and description of the universe
  - Add or remove users or SAML groups from the universe
  - Activate or Deactivate the universe
- However, they cannot:
  - Add or remove the services added by a Tenant Administrator or a Service Administrator
  - Delete the universe

Create and Manage an OI Universe

This section provides the following information:

Best Practices

The OI universes define the data level access for a SAML group or a user in DX Operational Intelligence. An OI Universe is based on the DX Operational Intelligence service names that are defined in a hierarchical format. Consider the following best practices while defining the OI Universes:

- When you select a service to define the data level access, by default, all the child services are also selected.
- When you grant access to the parent service, do not restrict access to the child services.
- When multiple accesses are defined for a universe, the accesses have an OR relationship with each other. That is, all the parent and child services belonging to each of these accesses are part of the universe. In such cases, use the **one of** option with **none of the** option carefully and avoid defining them with the same hierarchy.
- We recommend a maximum of 5000 services per universe. When the number of services in a universe goes beyond 5000, you will experience slower response times on the UI.

Create a Universe

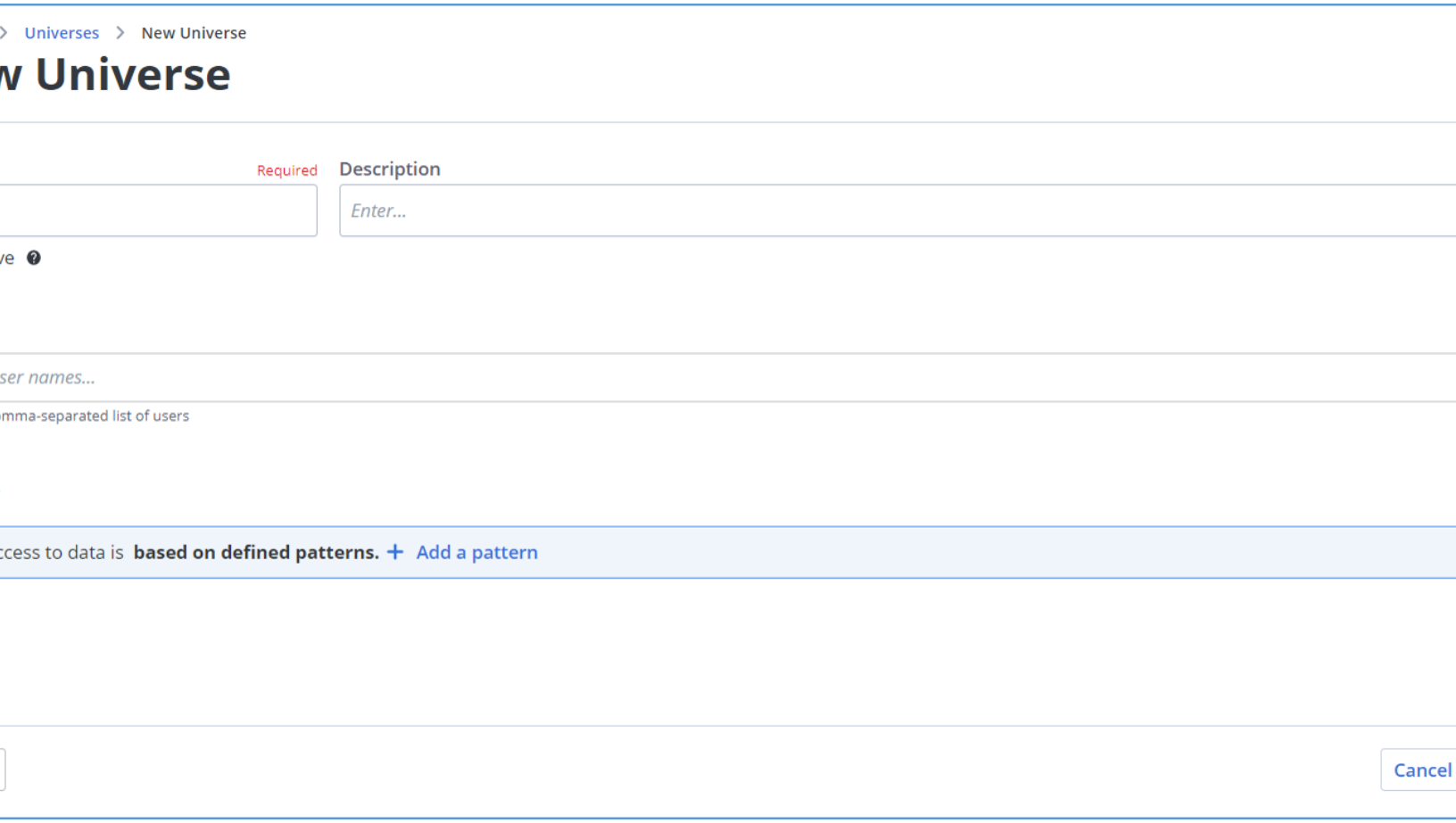
Only a user with the Tenant Administrator role or Security Administrator role can create a universe.

Follow these steps:

1. Log in to DX Operational Intelligence.
2. Click **Settings** in the left navigation pane.

3. Click **Manage** in the **Manage Universes** tile.
   The Universes page is displayed.

   NOTE
   Alternatively, you can navigate to the **Universes** page from the **Settings** page in DX SaaS.
4. Click **+ New Universe**.
   The **New Universe** page is displayed.



5. Provide the following information:
   - **Name:** Enter a name for the universe.
   - **Description:** Enter the description.
   - **Active:** Select this checkbox to activate this universe. Inactive universes are not displayed in the **Universe Selection** list.
   - **(Only for SAML-based tenants) SAML Groups:** Enter the comma-separated list of SAML groups that you have configured in the IdP. Ensure to enter the exact group name.
     After you enter the group name and press enter, the **Can manage** option is displayed.
     - Select this option to enable the users in this group to manage the universe. If this option is not selected, then the users can only view the universe, but they cannot edit or perform any other actions.
   - **Users:** Enter the comma-separated lists of user IDs that need access to this universe. Ensure to enter the exact user ID that was configured. After you enter the group and press enter, the **Can manage** option is displayed.

- Select this option to enable the users to manage the universe. If this option is not selected, then the users can only view the universe, but they cannot edit or perform any other actions.

– **Accesses:** Click **+ Add a pattern** to define the data access pattern for the services.

a. Select the filter:
   - **One of**
   - **None of**

     NOTE
     When multiple accesses are defined for a universe, the accesses have an OR relationship with each other. That is, all the parent and child services belonging to each of these accesses are part of the universe. In such cases, use the **one of** option with **none of the** option carefully and avoid defining them with the same hierarchy.

b. Select the service names from the list.

   NOTE

   - When you select a service to define the data level access, by default, all the child services are also selected.
   - When you grant access to the parent service, do not restrict access to the child services. If restricted, opening the Service Details or Topology pages of the parent service throws the following error:

     ```
     An error occurred retrieving the requested service. It may have been deleted.
     ```

c. Click **Save**.

d. (Optional) Click **View Sample**.
   The sample displays all the added services. If any service has child services, that service appears in bold. Click that service to view the child services.

e. Click **+ Add another** to add another pattern if required.

f. Click **Create**.
   The universe is created is displayed on the **Universes** page.

Edit a Universe

A Tenant Administrator or a Security Administrator can edit a universe in the following ways:

- Edit the name and description of the universe
- Add or remove users or SAML groups from the universe
- Add or remove services from the universe
- Delete, Deactivate, or Activate any universe in the tenant

A Power User, User, or a user with a Custom Role can edit or manage the universe in the following ways only if given access:

NOTE
A Tenant Administrator or a Security Administrator must select the **Can manage** option for the user on the **Universe** page to grant manage access.

- Edit the name and description of the universe
- Add or remove users or SAML groups from the universe
- Activate or Deactivate the universe

NOTE
They cannot,

- Add or remove the services added by a Tenant Administrator or a Service Administrator
- Delete the universe

Follow these steps:

1. Log in to DX Operational Intelligence.
2. Click **Settings** in the left navigation pane.
3. Click **Manage** in the **Manage Universes** tile.
   The Universes page is displayed.

   NOTE
   Alternatively, you can navigate to the **Universes** page from the **Settings** page in DX SaaS.
4. Click the **ellipses** icon under the **Actions** column for the universe that you want to edit.

   NOTE
   A Power User, User, or a user with a Custom Role can edit or deactivate a universe only if given access to manage the universe.
5. Select the required action.

Data Level Access Permission for Capabilities

Data access permissions enable you to control what data a user has access to. Using a universe, you can define the capabilities data access permissions for users.

By default, every user is provisioned with a universe named **All Access** that enables them to view all the data. You can select the Universe for which you want to view the data from the Universe dropdown list which displays all the universes that a user has access to in addition to the **All Access** universe.

Service Analytics Data Level Access Permission

You can select the universe for which you want to view the data from the Universe dropdown. By default, users with the out-of-the-box roles are provisioned with the **All Access** universe that enables them to view all the DX Operational Intelligence data in the tenant. For users with custom roles, the administrator must specifically add the **Universe Management** privilege for the **All Access** universe to be provisioned. This privilege is available under the **DX Platform > Universe Management** section on the **Roles** page. The Universe dropdown is added to the following pages:

- Services Overview page
- Service Details page

NOTE

- You must have access to **All Access** Universe to **Create, Edit,** and **Delete** a service.
- When you are viewing a service and select a different Universe and if that particular service is not part of the selected universe you see the following error:

  ```
  Cannot display the selected service. It may have been deleted or the service may not belong to the current universe
  ```
- If an entity has multiple services, then you can view the specific services that are part of the Universe.
- For more information about Universes, see Universes.

Alarm Analytics Data Level Access Permission

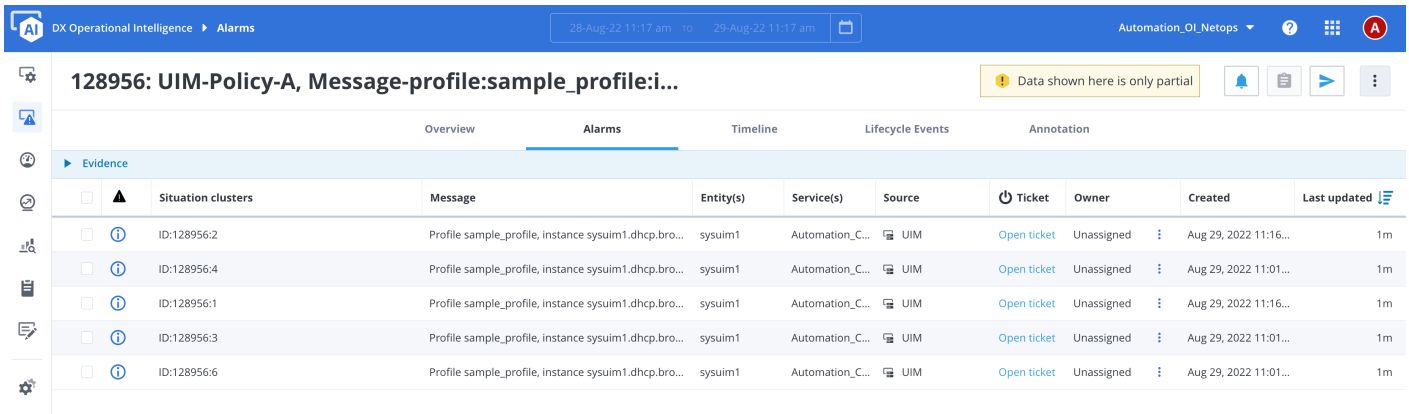
You can select the universe for which you want to view the data from the Universe dropdown. By default, a universe named All Access enables you to view all the data that are not associated with services.

Based on the data access pattern for services defined in the Universe, you can see the Alarms for the services on the Alarms Analytics page. The Universe dropdown list is added to the following pages:

- **All Alarms:** All alarms appear based on the Services defined in the Universe.
- **Service Alarms:** All service alarms appear based on the Services defined in the Universe.
- **Situations:** All situation alarms appear based on the Services defined in the Universe. If at least one service is part of the Universe you selected, only partial data is shown on the situations details page (Alarm tab, Overview tab, Timeline tab) and the top level situations also contains partial data with the following warning message:

  ```
  Data shown here is only partial
  ```

  For example, in the following image, partial data appears for the services that are part of the selected Universe **Automation_OI_Netops**.



NOTE

- For any alarm type, the life cycle events tab displays all the alarm events and is not context to a Universe.
- For more information about Universes, see Universes.

Monitored Inventory Data Level Access Permission

You can select the universe for which you want to view the data from the Universe dropdown. By default, users with the out-of-the-box roles are provisioned with the **All Access** universe that enables them to view all the DX Operational Intelligence data in the tenant. For users with custom roles, the administrator must specifically add the **Universe Management** privilege for the **All Access** universe to be provisioned. This privilege is available under the **DX Platform > Universe Management** section on the **Roles** page.

Based on the data access pattern for services defined in the Universe, you can see the entities on the Monitored Inventory pages.

NOTE

- Services of an entity are filtered based on the selected Universe.
- Users with **All Access** permission can perform **Manage Maintenance Window** operations such as **Create**, **Add to** and **Remove from**. However, Manage Maintenance Window operations get disabled when a logged in user does not have **All Access permission**.
- If an entity has multiple services, then you can view the specific services that are part of the Universe.
- **Maintenance Window:** Users with **All Access** permission can create a new maintenance window.
- For more information about Universes, see Universes.

Performance Analytics Data Level Access Permission

You can select the universe for which you want to view the data from the **Universe** dropdown. By default, users with the out-of-the-box roles are provisioned with the **All Access** universe that enables them to view all the DX Operational Intelligence data in the tenant. For users with custom roles, the administrator must specifically add the **Universe**

Management privilege for the **All Access** universe to be provisioned. This privilege is available under the **DX Platform > Universe Management** section on the **Roles** page. When you select a universe, only the metrics associated with that are available in the metric list.

Predictive Insights Data Level Access Permission

You can select the universe for which you want to view the data from the **Universe** dropdown list. By default, users with the out-of-the-box roles are provisioned with the **All Access** universe that enables them to view all the DX Operational Intelligence data in the tenant. For users with custom roles, the administrator must specifically add the **Universe Management** privilege for the **All Access** universe to be provisioned. This privilege is available under the **DX Platform > Universe Management** section on the **Roles** page.

When you are viewing insights for a universe, and if no insight information is available for the selected universe, you see the following error:

```
No Data found
```

Capacity Analytics Data Level Access Permission

You can select the universe for which you want to view the data from the **Universe** dropdown. By default, users with the out-of-the-box roles are provisioned with the **All Access** universe that enables them to view all the DX Operational Intelligence data in the tenant. For users with custom roles, the administrator must specifically add the **Universe Management** privilege for the **All Access** universe to be provisioned. This privilege is available under the **DX Platform > Universe Management** section on the **Roles** page. The **Universe** dropdown list is added to the following pages:

- Top Capacity Consumers
- Configured Services
- Configured Groups

When you are viewing details for Consumers Users, Configured Services, and Groups, and select a different universe and if that specific page is not part of the selected universe, you see the following error:

```
This page is not available for the selected universe. Switch to the 'All Access' universe
to view content.
```

NOTE

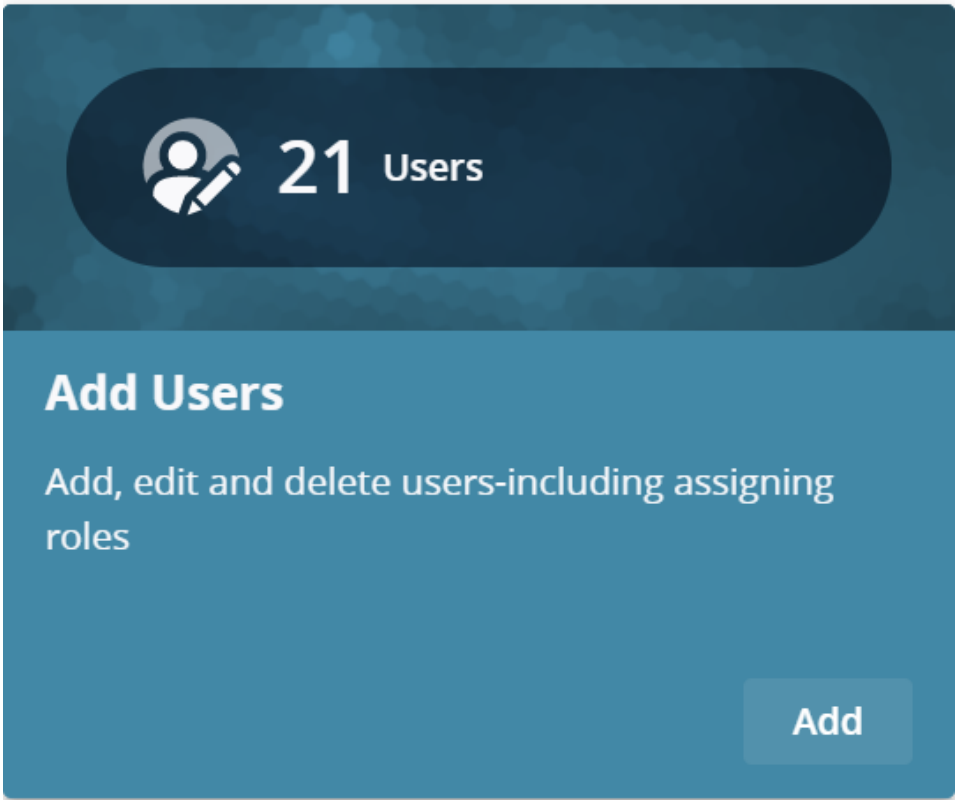
- You must have access to **All Access** universe to view the Groups data.
- For Top Consumers and Services, you can view data for the **All Access** universe and also for the selected universe.

User Management

User management allows the administrators to manage user access to various capabilities of DX Operational Intelligence.

The administrators can perform following tasks to manage DX Operational Intelligence users:

- Control user access to the DX Operational Intelligence application
- On-board and off-board users to and from DX Operational Intelligence
- Provide user access based on the combination of user logins, privileges, permissions, and roles (role-based access control)
- Manage Service Account user and SAML Authentication Users



- Supported User Roles and their Access Privileges
- Create and Manage Users
- Manage SAML Users

{"URL":["https://cloudmanagement/#/settings/users"],"description":"concept.dita_a690f1f8-51e6-4ee7-94a9-ef53507dd87a","new":"","new_video":"","admin":"","troubleshooting":{"masterkb":"","text":"","URL":[]},"pendo":"","video":[]}

Create and Manage Users

{"URL":["https://cloudmanagement/#/settings/users"],"description":"task.dita_b9aa0c3d-985e-4271-9841-b52a197f3431","new":"","new_video":"","admin":"","troubleshooting":{"masterkb":"","text":"","URL":[]},"pendo":"","video":[]}

As Tenant Administrator, you can perform the administrative tasks in the scope of the tenant account to which you belong:

- Create Tenant Administrators, Power Users, and Users.
- Update basic information of an administrator or a user.
- Update the administrative role and password.
- Enable or disable the user.
- Delete a user or power user account

This article explains the tasks that a Tenant Administrator can perform:

- Create User
  - Password Policy Rules
- Manage Users
- Manage Service Account Users
  - Activate Service Account User (New Tenant)
  - Create Service Account User (Existing Tenant)

Create User

{"URL":["https://cloudmanagement/#/settings/users/user/*/create"],"description":"task.dita_16739779-eaef-47b3-9862-4b79a7c868d8","new":"","new_video":"","admin":"","troubleshooting":{"masterkb":"","text":"","URL":[]},"pendo":"","video":[]}

As a Tenant Administrator, you can create Tenant Administrators, Power Users, and Users and assign roles to provide access to DX Operational Intelligence.

Follow these steps to create a user:

1. Log in to DX Operational Intelligence as a Tenant Administrator.
2. In the left navigation pane, click **Settings**.
3. Under Environment Definition and Access, click **Users**.
   The User Management page appears, which lists the created users.
4. Click **+ Add User**
5. Enter the following details:
   - First Name
   - Last name
   - Email
6. Assign one of the following roles:
   - Tenant Administrator
   - Power User
   - User
7. Enter the User Name.
8. Click **Save**.
   The user is created.

Password Policy Rules

When you provide your password, ensure that it adheres to the following guidelines:

- The length of the password must be between 6 to 25 characters.
- The password must have at least 1 number.
- The password must have at least 1 special character.
- The validity of the password is for 180 days from the time of its reset.
- The account will be locked after 5 unsuccessful login attempts.

Manage Users

You can manage the existing user accounts.

Follow these steps:

1. Login to DX Operational Intelligence as a Tenant Administrator.
2. In the left navigation pane, click **Settings**.
3. Click **Add** on the **Add Users** tile.
   The User Management page appears.

4. Perform the following actions on this page:
   - Filter by: In the Filter text box, specify the user name, user, role, status, or last updated.
   - Activate or Deactivate a user:
   - Click the User name to open the details of the user. Enable Active user to activate the user. Disable Active user to deactivate the user.
   - Changer the user details: Click the User name to open the details of the user. You can change the role, password, username, first name, and last name fields.
   - Reset the password: Click the User name to open the details of the user. Click RESET PASSWORD. An email to reset the password is sent to the email ID associated with the user.
   - Delete a user: Click the User name to open the details of the user. Click DELETE.
5. Click SAVE.

   The existing user role is configured.

Manage Service Account Users

The DX Gateway configuration includes the user credentials to generate a tenant token. This token is used to authenticate and authorize DX Gateway to call the Ingestion APIs to send data into DX Operational Intelligence. When a user is switched from Basic Authentication to SAML, DX Gateway does not support the SAML authentication and the token generation fails.

To enable communication between the On-Premise systems and DX Operational Intelligence through DX Gateway in the SAML enabled environments, create or activate a Service Account User and configure this user in DX Gateway.

As a Tenant Administrator, you can manage the service account user in the following ways:

- Enable (Activate) or disable the user using the Active User option
- Associate or update the the email ID of service account user
- Reset the password

> **NOTE**
> Note the following points:
>
> - Only one service account user can exist for each tenant.
> - A service account user login to DX Operational Intelligence using the UI is not allowed. However, the user can login using API.
> - A service account user will have no impact when the tenant authentication mechanism changes to SAML.
> - When the authentication mechanism changes to SAML, the User Management page displays the same list of users that was available before the change.

Activate Service Account User (New Tenant)

When a new tenant is provisioned, a service account user is created automatically with the first name, last name, email, role, and username configured and this user is inactive by default. While creating this user, email is configured with a dummy email address and the Role is selected as User and the User Name is set as SERVICE-ACCOUNT-USER.

The Role and User Name fields are not editable.

The service account user is displayed in the Settings > Users > User Management page.

To use this service account user, you must first activate the user in the User Management page and also change the email address.

Follow these steps:

1. Login to DX Operational Intelligence as a Tenant Administrator.
2. In the left navigation pane, click **Settings**.
3. Click **Add** on the **Add Users** tile.
   The User Management page appears.
4. Click the Service Account User.
5. In the User Management page, do the following:
   a) Enable Active User to activate the service account user.
   b) Edit the email address.
6. Click Reset Password.
7. Click Save.
   The Service Account User is activated.

Create Service Account User (Existing Tenant)

For basic authentication enabled tenants, follow the steps mentioned in the Create User section to create a service account user.

Ensure that you select Role as USER and enter the User Name as SERVICE-ACCOUNT-USER.

> **NOTE**
> For SAML enabled tenants, contact Broadcom Support to get the service account user created.

Manage SAML Users

DX Operational Intelligence supports SAML users. Configure any SAML Identity Provider to connect to DX Operational Intelligence, add users, and access DX Operational Intelligence from your SAML identity provider login dashboard.

Note the following points:

- SAML tenant administrators cannot view the list of users in DX Operational Intelligence.
- Do not assign the Tenant Administrator and Power User roles to the same DX User. When you create a SAML user with the same DX username and role as Tenant Administrator and Power User, the Power User is authorized as a SAML user instead of Tenant Administrator.
- If a user does not belong to any SAML group, that user cannot log in to DX Operational Intelligence.
- Consider a user is created using local authentication and is assigned a specific role (for example, Tenant Admin). Now, the authentication is switched to SAML, and the same user is assigned a different role (for example, Power User). In this case, DX Operational Intelligence honors the SAML configuration and considers the user as a Power User and not a Tenant Administrator. DX Operational Intelligence follows the same behavior when the switch is from SAML to local authentication. That is, it honors the local authentication and the associated role.
- SAML authentication requires that you first configure the SAML IdP before completing the SAML configuration in DX Operational Intelligence.
- DX SaaS supports a maximum of approximately 250 SAML groups depending on the length of the group names for a single user. We recommend that you configure the SAML assertion to filter and send only the groups that the DX SaaS applications require. The SAML groups used by the DX SaaS applications are available on the **Settings > Roles** page.
- Ensure that the added SAML users have the email address assigned and they are properly mapped in the SAML attributes mapping configuration.

Configure SAML Authentication

DX Platform supports encrypting SAML assertions using a public/private key pair. However, the ownership including the creation and management of the key pairs lies with a Tenant Administrator. To configure SAML authentication, upload the private key into DX Platform (Service Provider) and the public key into the IdP. For more information about the SAML encrypted payload configuration, see the IdP-specific documentation.

DX Operational Intelligence supports AES256 and RSA-OAEP_MGF1P (includes MGF1 and SHA1) Key Transport encryption algorithms.

> **NOTE**
> You can configure only one SAML per tenant.

Follow these steps:

1. Log in to DX Operational Intelligence as a Tenant Administrator.
2. In the left navigation pane, click **Settings**.
3. Click **Add** on the **Add Users** tile.
   The User Management page appears.
4. Click , and select SAML to switch to SAML authentication.
5. Enter the following details:
   - **SAML Authentication Details**

     a) Issuer: Specify the issuer name or IdP urn that you have created in your IdP account.
     b) Identity Provider (IDP) Login URL: Specify the IdP login URL to which DX Operational Intelligence must send its SAML authentication requests.
     c) Identity provider certificate: Enter identity certificate or signature verification certificate (.pem format).
   - **Map Attributes between DX Platform and SAML Account**

     a) Roles: Specify the value configured in the SAML identity provider.
     b) Email, First name, Last name: Specify the attributes for user details configured in the SAML identity provider. Roles and Email fields are mandatory.
   - **Add Group from SAML**

     a) Provide the group information from SAML. For group information, see Users.
   - **Populate your SAML Account (for bi-directional communication)**

     a) Callback IDP URL: The URL of DX Operational Intelligence to which the SAML assertions are sent, and the identity provider has authenticated the user.
     b) Audience: Specify DX Operational Intelligence tenant id.
     c) Logout URL: Specify DX Operational Intelligence URL.
     d) SAML Protocol settings: Specify the SAML protocol.

   SAML is configured.

6. Click Save.
7. (Optional) Click **Test SAML:** Allows you to test the connection between SAML and DX Operational Intelligence. You can perform the following actions on this page:
   a) **Close** the page when logging in to the SAML account is successful.
   b) **SAML Configuration:** Use this option to verify the SAML configuration.
   c) **Download Config and Revert:** Use this option to download the SAML configuration details if the login fails.
   d) **SAML Login Page:** Provide the IdP credentials to verify if the account is created successfully in SAML.

Log in as a SAML User

After you have configured the SAML authentication, you can log in to DX Operational Intelligence using SAML credentials. A SAML user cannot change the password using DX Operational Intelligence UI.

Follow these steps:

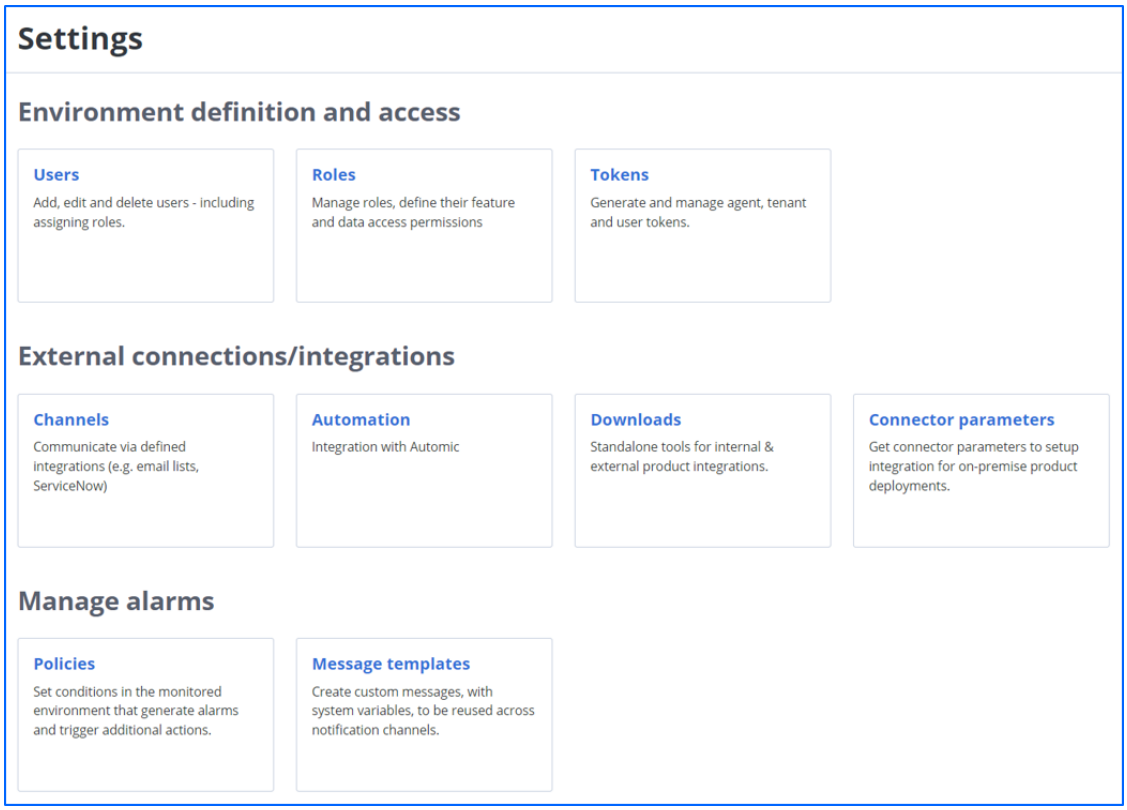
1. Log in to DX SaaS.
2. Enter the tenant ID.
   The SAML Identity Provider login page appears.
3. Provide your SAML credentials.
   The Launch Pad appears.
4. From the Settings page, select Users.
   The User Management page appears. You can review the SAML configured details and can manage SAML-defined groups that are mapped to the DX Platform role.

Manage SAML-Defined Groups Mapped to DX Platform Role

For a user to access DX Operational Intelligence using SAML authentication, map the SAML groups to a DX Platform role. The tenant administrator creates a SAML user and then maps that SAML user to the DX Platform role. When these users log in to DX Operational Intelligence, they can perform only those operations that the mapped role allows. They cannot view or access other UI options. For example, if a tenant administrator creates a Power User (PU) group while creating a user in SAML, then you can map that PU group to the DX Platform Power User role. This will allow the SAML user to log in to DX Operational Intelligence with the DX Platform Power User role privileges. That user can then view and perform only those actions that are allowed to the DX Platform Power User role. If a user does not belong to any SAML group, that user is not allowed to log in to DX Operational Intelligence.

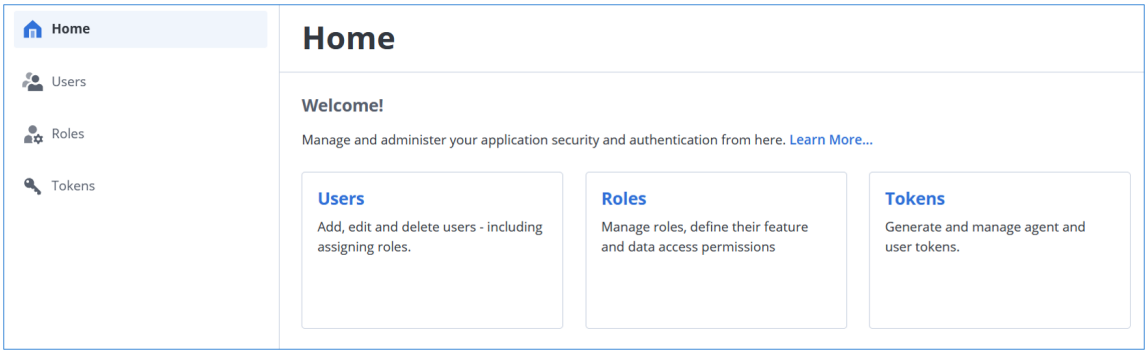
The following screenshot shows multiple SAML groups mapped to different roles:

Example - Tenant Administrator Role	The following screenshot shows that a SAML user is logged in to DX Operational Intelligence with the Tenant Administrator role privileges. The user can access only role-specific options. Note that various options are accessible to this role:



Example - Security Administrator Role

The following screenshot shows that a SAML user is logged in to DX Operational Intelligence with the Security Administrator role privileges. The user can access only Users, Role Management, and Token Management tiles:

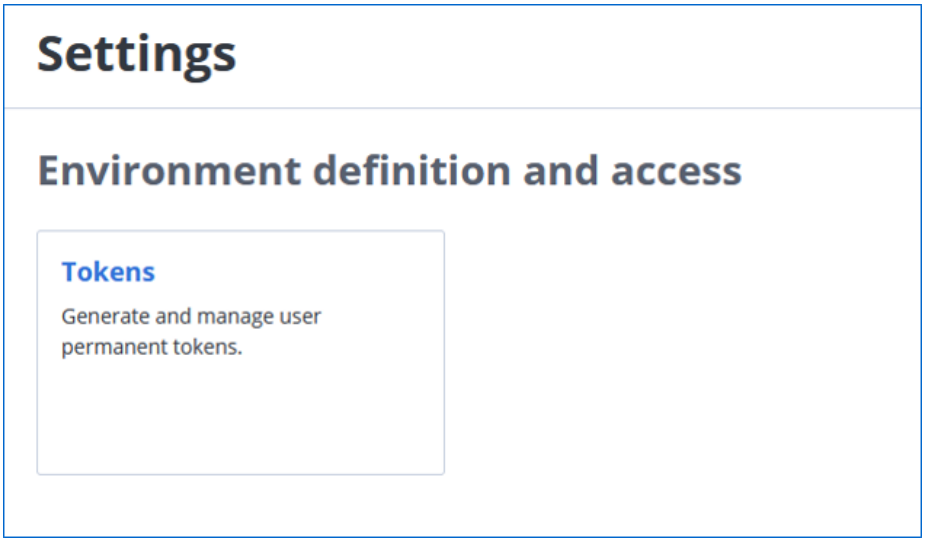


Example - Power User Role

The following screenshot shows that a SAML user is logged in to DX Operational Intelligence with the DX Platform Power User role privileges. The user can access only role-specific options. Note that various options that are accessible to the Tenant Administrator role are not visible to this user, because this role is not authorized to access that information:

Example - User Role

The following screenshot shows that a SAML user is logged in to DX Operational Intelligence with the DX Platform User role privileges. The user can access only role-specific options. Note that the Settings option in the left navigation bar is not visible to this user, because the role is not authorized to access that information:



Encrypt SAML Assertion

You can encrypt the SAML assertions that you receive from the IdP by enabling the **Encrypted SAML Assertion Support** option on the **User Management: SAML** page. You can enable the encryption only for an existing SAML configuration. Once the encryption is enabled, you can also update the **IdP Certificate** if it is valid and the **Private Key**.

DX Platform uses the AES256 encryption algorithm and RSA-OAEP_MGF1P (includes MGF1 and SHA1) Key Transport algorithm for encryption.

> **NOTE**
> The option to encrypt the assertions is displayed only while editing the SAML configuration.

Prerequisite:

Before you enable the encryption, ensure that the SAML connection is established successfully.

Follow these steps:

1. Log in to DX Operational Intelligence as a Tenant Administrator.
2. In the left navigation pane, click **Settings**.
3. Click **Add** on the **Add Users** tile.
   The User Management page appears.
4. Click **Edit/View Configuration**.
5. Enable the **Encrypted SAML Assertion Support** option. Click **Enable**.
6. Enter the private key.
7. Click Save.

Update the IdP Certificate

You can update or replace the IdP certificate on the User Management page.

> **NOTE**
> You must update the IdP certificate first in DX SaaS and then in the IdP. Updating the certificate first in the IdP results in the DX SaaS login failure. After you have saved the updates in DX SaaS, ensure to coordinate the IdP updates as soon as possible to avoid any subsequent login failures.

Follow these steps:

1. Login to DX Operational Intelligence as a Tenant Administrator.
2. In the left navigation pane, click **Settings**.
3. Click **Add** on the **Add Users** tile.
   The User Management page appears.
4. Click **Edit/View Configuration**.
5. Click **Replace Certificate** and paste the IdP certificate.
6. Click Save.

Configure SiteMinder as SAML Identity Provider

This topic describes how to set up SiteMinder as your identity provider by configuring SAML integration in both DX Operational Intelligence and SiteMinder. DX Operational Intelligence supports external SAML (Security Assertion Markup Language) identity provider to authenticate and authorize users.

DX Operational Intelligence SAML integration conforms to the Security Assertion Markup Language 2.0 (SAML 2.0) specification, so you can use any SAML 2.0- compliant identity provider. This document describes how to set up and administer SiteMinder SAML authentication. Set up SAML in DX Operational Intelligence.

Follow these steps:

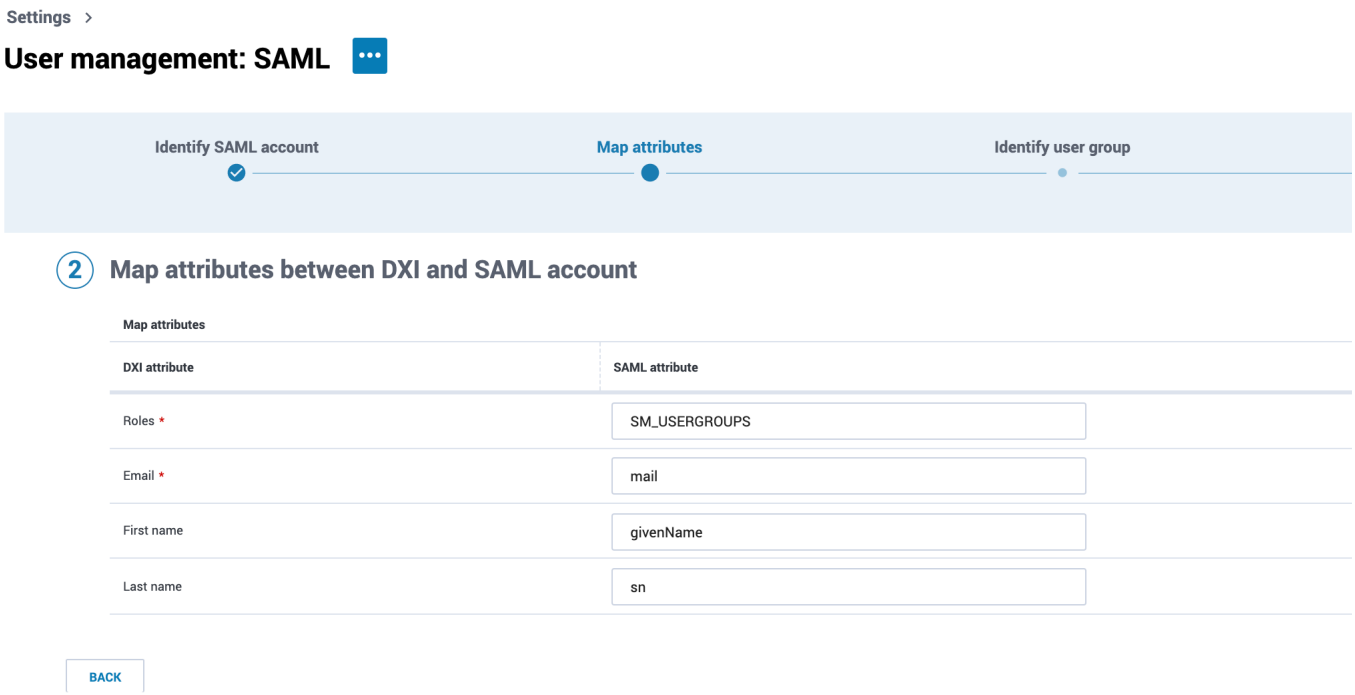
1. Login to DX Operational Intelligence as a Tenant Administrator.
2. In the left navigation pane, click **Settings**.
3. Click **Add** on the **Add Users** tile.

   The User Management page appears.
4. Click and select Switch user store/Authentication SAML.
5. Enter the following SAML Authentication Details and click Next:
   a) Issuer: - Specify the issuer name or IdP urn that you have created in your IdP account. For the SiteMinder integration, this is the Local IDP ID. The following snapshot highlights the Local IDP ID in SiteMinder.
   b) Identity Provider (IDP) Login URL: - Specify the IdP login URL to which DX Operational Intelligence must send its SAML authentication requests. For the SiteMinder integration, this is the SSO Service URL. The following snapshot highlights the SSO Service URL in SiteMinder.
   c) Identity provider certificate: - Enter the identity certificate or signature verification certificate (.pem format). The following snapshot illustrates the DX Operational Intelligence SAML Authentication details page.
6. Enter the following values to map attributes between DX Operational Intelligence and SiteMinder and click Next. For the SiteMinder integration, specify the LDAP values as the snapshot below illustrates in SiteMinder.
   a) Roles: - Specify the value that is configured in the SAML identity provider. For example, SM_USERGROUPS. DX Platform SAML Authentication works only with a SAML response containing the group name instead of the distinguished name. To receive only the group name from SiteMinder, create the following custom expression response:

      ENUMERATE(SM_USERGROUPS, STRING(RDN(STRING(%0), FALSE)))

      For more information, see Broadcom Communities, Broadcom Support KB Article, and SiteMinder Documentation (Review the sections - LocateAttributesinaUserDirectory in the Get Function, andTestSetElements in the Filter Function).
   b) Email, First name, Last name: - Specify the attributes for user details that are configured in the SAML identity provider. For example, Email - mail, First name - givenName, Last name - sn. Roles and Email fields are mandatory.

      The following snapshot illustrates the DX Operational Intelligence Map Attributes page.



c) Preview the SAML Account information which is used for bi-directional communication and click Save.

7. Enter the SAML group information which will be associated with the subtenant Admin and click Next. The following snapshot illustrates adding a group from SAML in DX Operational Intelligence.
8. Preview the SAML Account information which is used for bi-directional communication and click Save.
   a) Callback Idp URL: - Specifies the DX Operational Intelligence URL to which the SAML assertions are sent and the identity provider has authenticated the user.
   b) Audience: - Specifies the DX Operational Intelligence tenant Id.
   c) Logout URL: - Specifies the DX Operational Intelligence URL.
   d) SAML Protocol settings: - Specifies the SAML protocol. The following snapshot illustrates the DX Operational Intelligence Populate SAML Account page.
9. You can now test the SAML configuration between SiteMinder and DX Operational Intelligence. You can perform one of the following actions:
   - Close the page when logging in to SAML account is successful.
   - SAML Configuration - Use this option to verify the SAML configuration.
   - Download Config and Revert - Use this option to download the SAML configuration details if the login fails.
   - SAML Login Page - Provide the IdP credentials to verify if the account is created successfully in SAML.

Capability Configurations

Administrators can configure the following capabilities of DX Operational Intelligence for the users.

- Custom Situation Definitions

Custom Situation Definitions

Custom Situation Definitions in DX Operational Intelligence enables you to create custom rules for situation clustering per an organization's rules and objectives.

Situation clustering based on Global policies(Algorithmic-based) restricts the organizations to add any organizational-specific rules. With Custom Definition based situation clustering in place, you can define the predictable rules that meet your organization-specific business objectives.

This section provides the following information:

- How Situation Clustering Works
- Configure Custom Situation Definition
- Prioritize Custom Situation Definitions
- View Situation Clusters of Type Custom

{"URL":["https://digital-oi/settings/custom-situations"],"customLabelGetStarted":"Custom Situation Definitions","description":"concept.dita_5f57f3bc-93c4-4c11-926b-1ac8bd4e2a3a"}

How Situation Clustering Works

Before you begin configuring cluster situation definitions, you must understand how the situation clustering works:

- The situation clustering that is based on custom definitions takes precedence over the situation clustering using global policies.
- DX Operational Intelligence does not support moving of alarms from custom to global policy clusters.
- DX Operational Intelligence supports movement of alarms from one custom rule to another between iterations. The movement of alarms between custom rules depends on the alarm criteria and the custom rule prioritization. The application restricts movement of alarms that are assigned to a high priority custom rule when the criteria changes.

  For example, Custom Definition CD1 has a rule to filter alarms with severity 'Critical' and Custom Definition CD2 has a rule to filter the alarms with severity Major.

  - **Scenario 1**:CD2 is marked as High Priority custom rule:

    DX Operational Intelligence creates two clusters - Cluster C1 for Alarms with severity Critical and Cluster C2 for alarms with Severity Major.
    - DX Operational Intelligence adds the Alarm A1 with severity Critical to cluster C1.
    - If the alarm severity changes from Critical to Major, DX Operational Intelligence moves the alarm to Cluster C2.
    - If the alarm severity changes to unknown or all underlying alarms are closed, DX Operational Intelligence retains the alarm A1 in cluster C2.
  - **Scenario 2**:CD1 is marked as High Priority custom rule:

    DX Operational Intelligence creates two clusters - Cluster C1 for Alarms with Severity Critical and Cluster C2 for alarms with Severity Major.

    If the alarm severity changes from Critical to Major, DX Operational Intelligence restricts the alarm movement from cluster C1 to C2 as CD1 is marked as a high priority custom rule.
- DX Operational Intelligence restricts the movement of alarms with a most impacted host between custom definitions for 10 minutes. This restriction is to ensure that the sufficient time is given for automatic ticket creation for alarms with

a most impacted host. If the ticket is created within the 10-minutes time frame, DX Operational Intelligencecompletely stops the alarm movement.

- DX Operational Intelligence creates independent clusters when the clustering criteria does not match. For example, you defined the following clustering criteria: When the message match percentage is 50 percent or more, group the subclusters.
  - **Scenario 1:**
    The match percentage of messages is 66.6% (number of matches(words)/total number of words*100%) in Subclusters SC1 and SC2:
    - SC1 - message - host connection failed
    - SC2 - message - connection failed

    DX Operational Intelligence groups the SC1 and SC2 in one cluster.
  - **Scenario 2:**
    The match percentage of messages is 25% (<50%) in Sub Clusters SC1 and SC2:
    - SC1 - message - host connection failed
    - SC2 - message - connection failed pod down reason

    DX Operational Intelligence adds the subclusters SC1 and SC2 are added in their own clusters.

Configure Custom Situation Definition

As an administrator, you can configure a custom situation definition as per your organization's rules and objectives. A custom situation definition enables you to include the following information while defining the custom rules:

- Specify the filter criteria for alarms that the application must consider for clustering.
- Define a custom name
- Define the situation stabilization period for clustering.
- Define custom rules to determine the alarm similarity for clustering.
- Prioritize the custom situation definitions in order of importance when two or more custom definitions have similar criteria.

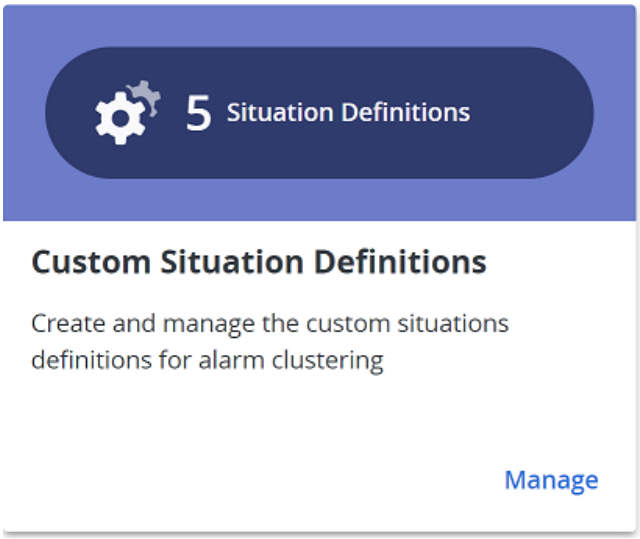
After you configure the custom definitions, DX Operational Intelligence enables the situation clustering as per the rules defined in the custom definitions. You can access the situations that are generated using custom definitions in the **Situations** window of **Alarm Analytics**.

Create Custom Situation Definition

You can create a custom situation definition by providing the basic details.

To create a custom situation definition, follow these steps:

1. Log in to DX Operational Intelligence and navigate to the Settings page.
2. Click **Manage** in the Situation Definition tile.
   The application displays the Custom Situation Definition page.



3. Click + Create Definition to define a custom situation definition.
   The application displays the Create Custom Situation Definition page.

Create Custom Situation Definition

Custom Situations enable users to cluster alarms based on various attributes like severity, service, entity, message, etc.

Status

Situation Definition Name Required

Enter Name

Alarm Filter

Define filter criteria to select the alarms for situation clustering.

Message: All × | Service: All × | Host Name: All × | Severity: All × | clear

Add Filter

Time Window 30 mins

Specify the situation stabilization period

Stabilization Time	0	Hours	30	Minutes
	Min: 5min, Max: 24hrs			
Auto Extend	0	Hours	0	Minutes
	Min: 0min, Max: 24hrs			
Max Stabilization Time	0	Hours	0	Minutes
	Min: 0min, Max: 24hrs			

Alarm Clustering Criteria

Define criteria to determine alarm similarity for clustering.

First by Required | Match percentage

Select... | 100 %

Cancel | Preview Last 24 hr | Save

Previe

Last 24 hr

4. Select **Status** to make the custom situation definition active.
5. Enter a unique name in the **Situation Definition Name** field.
6. Enter Custom Situation Name
7. Define Alarm Filters
8. Define Situation Stabilization Criteria
9. Define Alarm Clustering Criteria
10. Preview Results of Custom Situation Definition
11. Click Save.

Define Custom Situation Name

You can define a custom name for a situation definition using operations. Each operation has a set of attributes (also referenced as the supported fields) that are provided out-of-the-box. For example, you can add the operations **(Policy Name: {policyname()}** and **Count:{count(severity)})** to customize the definition name. This custom name is appended to the situation name and is displayed in the preview as well.

DX Operational Intelligence provides the following operations out-of-the-box.

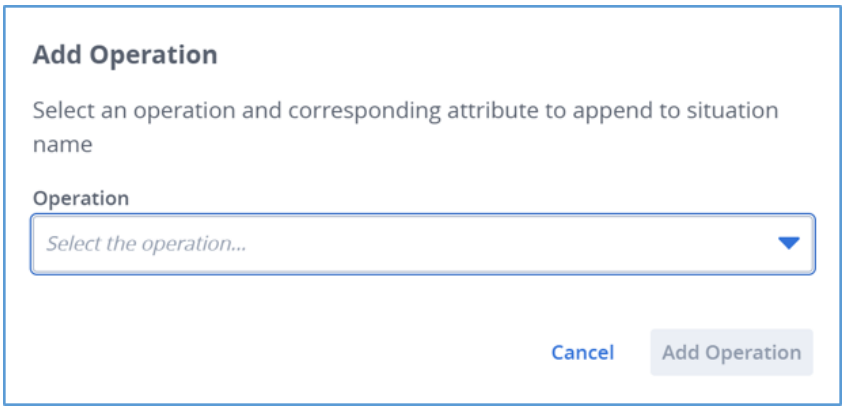
Operation	Description
Count	Provides the count of unique values of the specified field. **Supported Attributes/Fields:** Entity, Groups, Product, Message, Service, and Severity.
policyName	Provides the situation definition name.
nGram	Provides consecutive N words which occurred the most number of times. **Supported Attributes/Fields:** Message nGram is supported only for message. **Number of Items:** Select or enter the number of items.
listOf	Provides the N unique values for the specified field. **Supported Attributes/Fields:** Severity, Service, Entity, Groups, and Product **Number of Items:** Select or enter the number of items.
mostImpacted	Provides the N values of the given field from the mostImpacted template. **Supported Attributes/Fields:** Entity, Message, Severity, Service, Product, and Groups **Number of Items:** Select or enter the number of items.

NOTE
You can use the OOTB supported fields, or you can add fields for these operations using the API. The supported field that you want to add must be part of the **Operations definition, Clustering criteria fields,** and **Compressed alarm fields**. For more information about adding the custom attributes, see Add or Remove Supported Fields for Custom Situation Name Operations API in the **Situation Clustering Dimensions APIs** section.

The following procedure explains how to define the custom situation name.

Follow these steps:

1. Click **+ Add Operation.**
   The application displays the **Add Operation** dialog box.



2. Select the required operation and the attribute if available.

   You can select and add the operation. Alternatively, you can enter the free text and then select and add the operation. For example, enter **PolicyName:** and select **{policyname()}**. Similarly, enter **Count:** and select **{count(severity)}.**

3. Repeat the steps to add more operations.

Define Alarm Filter Criteria

The Alarm Filter section on the Create Situation Definition page enables you to define the filter criteria for alarms. You can define the filter criteria using one or more alarm attributes such as Message, Service, Hostname, Severity, and so on. DX Operational Intelligence identifies the subset of alarms for clustering based on the defined alarm filter criteria. After DX Operational Intelligence identifies the subset of alarms, you can define the cluster criteria and the stabilization period for the cluster.

NOTE

- The alarm filter criteria provide a set of alarms, which can be clustered based on the provided clustering criteria.
- While filtering, the **AND** operator is used between attributes and the **OR** operator is used between the attribute values. For example,

  ```
  (Severity: Critical OR Major) AND (Alarm Type:"application" OR "fault")  AND (Message contains
   "sshd" OR Message does not contain "ALARM: [SYSTEMS]")
  ```

- Only asterix (*) and dot (.) are supported in the filters.

You can use the following text-based filter operators while defining the filter criteria:

Match Criteria	Operator	Description
No Filter Criteria	All	Considers all alarms for clustering.
For Exact Matches	Equals	Filters the alarm for clustering when the filter attribute value matches with the specified value.
	Not equal	Filters the alarm for clustering when the filter attribute value does not match with the specified value.

Match Criteria	Operator	Description
	Predefined Value Checkboxes	Filters the alarm for clustering when the filter attribute value matches with the selected predefined value. If the selected filter attribute has any predefined set of values, the Select filter attribute drop-down displays them as options for selection. For example, the alarm severities such as Critical. Major, Minor, and so on, are displayed as options for selection.
For Partial Matches	Contain	Filters the alarm for clustering when the filter attribute value contains a match of the specified value.
	Does not contain	Filters the alarm for clustering when the filter attribute value does not contain a match of the specified value.
	Starts with	Filters the alarm for clustering when the first character or the string of the filter attribute value matches with the specified value.
	Does not start with	Filters the alarm for clustering when the first character or the string of the filter attribute value does not match with the specified value.
	Ends with	Filters the alarm for clustering when the last character or the string of the filter attribute value matches with the specified value.
	Does not end with	Filters the alarm for clustering when the last character or the string of the filter attribute value does not match with the specified value.

To define the alarm filter criteria for clustering, follow these steps:

1. Click

in the Alarm Filter section.
2. Select the filter attribute from the **Select filter attribute** list.
3. Select the appropriate filter operator and provide the value on which you want to filter the alarm data.
4. Click Add.
   The application defines the alarm filter criteria and associates the identified alarms with the custom situation definition.

Define Situation Stabilization Criteria

You can specify the situation stabilization criteria for situation clustering in a Custom Situation Definition.

- DX Operational Intelligence adds any new matching alarm to a situation cluster that is within the specified stabilization period.
- After the situation cluster is stabilized, DX Operational Intelligence creates a situation cluster for the matching alarms.

To define the stabilization criteria, select the following stabilization period in the Time Window section.

- **Stabilization Time:** Specifies the time period for situations to get stabilized before starting new clusters.
- **Auto Extend:** Specifies the time period to extend the situation clustering alerts before starting new clusters. When the new alerts are generated, the DX Operational Intelligence extends the total clustering time until the Max Stabilization Time is reached. You can use this time period with the Max Stabilization Time to ensure that DX Operational Intelligence continues to cluster alerts together that are related to the same failure. The Auto Extend option can be applied to new related alerts, not to existing alerts that are updated with new events.
- **Max Stabilization Time:** Specifies the Maximum time period that DX Operational Intelligence clusters alert before starting a new cluster.

Stabilization Time Line Example

The following example adds any new matching alarm to a situation cluster that is within the specified stabilization period and creates a situation cluster for the matching alarms.

Let us assume the stabilization criteria as follows:

stableWindow:1 hour, **extensionWindow:** 32 minutes, **maxStableWindow**: Two hours (max stable window)

Scenario 1: If the sum of the clusterAge and extension window is less than the stabilization time window, then the situation cluster is stabilized based on the **stableWindow** time.

For example: The sum of the clusterAge (26) and extension window (32) is 58, which is less than the stabilization time window (1 hour), then situation cluster stabilization is 1 hour.

Scenario 2: If the sum of the clusterAge and extension window is greater than the stabilization time window, then the situation cluster is stabilized based on the **sum of cluster age and extension window** time.

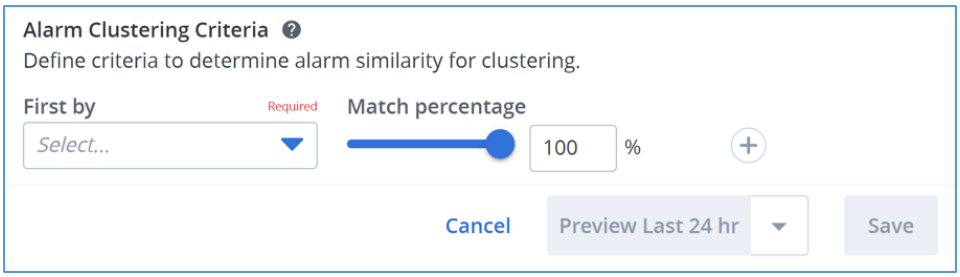
For example: The sum of the clusterAge (40) and extension window (32) is 72, which is greater than the stabilization time window (1 hour), then situation cluster stabilization is 1:17 which is less than max stable window.

Scenario 3: If the sum of the clusterAge and extension window is greater than the max stabilization time window, then the situation cluster is stabilized based on the **maxStableWindow** time.

For example: The sum of the clusterAge (89) and extension window (32) is 120, which is greater than the max stabilization time window (2 hour), then situation cluster stabilization is 2:01 which is greater than max stable window.

Define Alarm Clustering Criteria

After you identify the alarms for the custom definition, you must define the clustering criteria to determine the alarm similarity for clustering. You can customize the alarm clustering criteria using the Situation Clustering Dimensions APIs. For more information, see Customize Alarm Clustering Criteria section.



NOTE

- The alarm filter criteria provide a set of alarms, which can be clustered based on the provided clustering criteria.
- DX Operational Intelligence ignores any custom definitions for DX NetOps Spectrum alarms with a root cause. Instead, the application uses a predefined system definition rule while clustering the alarms with a root cause.

Follow these steps:

1. Select one of the following clustering criteria fields from the **First By** drop-down in the Alarm Clustering Criteria section.

   NOTE
   The following fields are out-of-the-box clustering criteria.

- **Entity:** DX Operational Intelligence considers the hostname and configuration item of an alarm as Entity. The supported characters for a hostname are the alphanumeric characters and the hyphen(-).

  Match Percentage Range: Minimum - 30% and Maximum - 100%.

  - The host string is pre-evaluated. The hostname with valid characters is considered for the alarm similarity. When the alarms are matched with the hostname, the match percentage is usually 100 percent.
  - In case the hostname contains any additional details, the application delimits the hostname using the unsupported characters. For example, the hostname of the alarm is '**eqx-lswa29-010520_ethernet53/1**'. During the host string evaluation, the application considers the string '**eqx-lswa29-010520**' as the hostname and ignores the string after the '_' as the underscore (_) is an unsupported character.

    NOTE
    DX Operational Intelligence considers the metric path as an Entity for DX Application Performance Management Alarms. The application calculates the match percentage that is based on the matching tokens from the left-hand side of the metric path.

    For example, the metric paths of two alarms are:

    - Alarm A: SuperDomain|tomcathost|Tomcat|Frontends|Apps|App1
    - Alarm B: SuperDomain|tomcathost|Tomcat|Frontends|Apps|App1|URLs|Default

    The '|' acts as a separator between tokens. After ignoring the restricted word 'SuperDomain', the application identifies the number of tokens in the alarms as five and seven, respectively. DX Operational Intelligence evaluates the metric path match from the Left-hand side(LHS). Alarms A and B have five token matches; hence the application calculates the similarity percentage as 71% (5/7 * 100).
- **Message:** DX Operational Intelligence calculates the message similarity by comparing of words between the alarm messages.

  NOTE
  The application ignores the restricted words, numbers, word case, and the word sequence while calculating message similarity for alarm clustering.

  For example,

  - Alarm A has the message: "CPU of host1 has breached Critical threshold of 80%"
  - Alarm B has the message: Cpu of host2 has breached Critical threshold of 82.5%
  - The restricted words are: of, has, 80%, 82.5%
  - DX Operational Intelligence considers the following words to calculate the match percentage after ignoring the restricted words and numbers, the application:
    - Alarm A has five words: CPU, host1, breached, Critical, threshold
    - Alarm B has five words: Cpu, host2, breached, Critical, threshold
  - Total number of words in Alarm A and Alarm B are 10.
  - Number of matched words are 8 (CPU, breached, Critical, Threshold
  - Number of unmatched words are 2(host1, host2)
  - Thus, the message similarity between the two messages is 80% (8/10*100).

  Match Percentage Range: Minimum - 30% and Maximum - 100%
- **Service:** DX Operational Intelligence considers the exact match of service name between alarms for alarm clustering. You can also select to **include child services** for alarm clustering. The child services are clubbed with

the parent service for alarm clustering. For the existing services, the **include child service** option is applicable only after the stabilization of the existing clusters.

Match Percentage Range: 100%

2. Specify the match percentage using **Match Percentage** slider field.
3. Click (+) to add additional cluster criteria rules.

   You can add maximum three rules for alarm clustering using the clustering fields.

   DX Operational Intelligence identifies the alarms that match the defined clustering criteria for clustering.
4. Preview the results.
5. Click **Save**.

   DX Operational Intelligence enables the clustering as per the defined custom definition.

   You can access and view the clusters in the Situation view of Alarm Analytics.

Customize Alarm Clustering Criteria

You can add custom Alarm Clustering Criteria to create custom situation definitions. You can customize these criteria based on your requirement

You can add the following custom alarm clustering criteria to determine the alarm similarity for clustering. Use Situation Clustering Dimensions APIs. to add the custom alarm criteria.

- Product
- Severity: The severity can be Critical, Major, Minor
- AlarmType

Out of the Box Custom Situation Definitions

The Out of the Box Custom Situation Definitions lets you use the following default rules to create the custom situation definitions. You can customize these rules based on your requirement.

- Spectrum Default policy
- ServiceOnly
- Spectrum BGP Critical
- UIM ROBOT Critical

Spectrum Default Policy

Spectrum alarms which are having root cause or symptom associated. You can enable or disable this policy based on your requirement. This policy is always ranked first.

> **NOTE**
> You cannot edit this policy.

- **Situation Definition Name:** SpectrumDefault
- **Alarm Filter:**
  - **Source:** Spectrum(equals)
  - **Symptoms:** All
  - **RootCause:** All

ServiceOnly Rule

The ServiceOnly Rule contains the following predefined attributes:

- **Situation Definition Name:** ServiceOnly
- **Alarm Filter:** Service: All
- **Stabilization Time:** 30 minutes
- **Alarm Clustering Criteria:** Service with 100% match percentage.

Spectrum BGP Critical

The ServiceOnly Rule contains the following predefined attributes:

- **Situation Definition Name:** Spectrum BGP Critical
- **Alarm Filter:** Service: Critical, Product: Spectrum, Message: BGP (Contains)
- **Stabilization Time:** 30 minutes
- **Alarm Clustering Criteria:** Message with 70% match percentage.

UIM ROBOT Critical

The ServiceOnly Rule contains the following predefined attributes:

- **Situation Definition Name:** UIM ROBOT Critical
- **Alarm Filter:** Service: Critical, Product: UIM, Message: robot (Contains)
- **Stabilization Time:** 30 minutes
- **Alarm Clustering Criteria:** Service with 100% match percentage.

To view how to edit these rules, see Configure Custom Situation Definition.

Preview Results of Custom Situation Definition

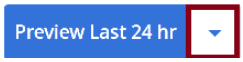
After you define the rules for custom situation definition, DX Operational Intelligence enables you to preview the results.

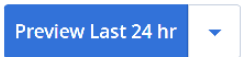
You can preview the situation clusters and make the necessary changes to the custom rules if required.

DX Operational Intelligence uses the existing alarms that match the clustering criteria for preview results. You can choose anyone of the following time intervals to view the results:

Time Intervals
Last 1 hr
Last 4 hr
Last 8 hr
Last 24 hr
Last 3days
Last 1 wk

To preview the results, follow these steps:

1. Select one of the time intervals from the 

dropdown.
2. Click 

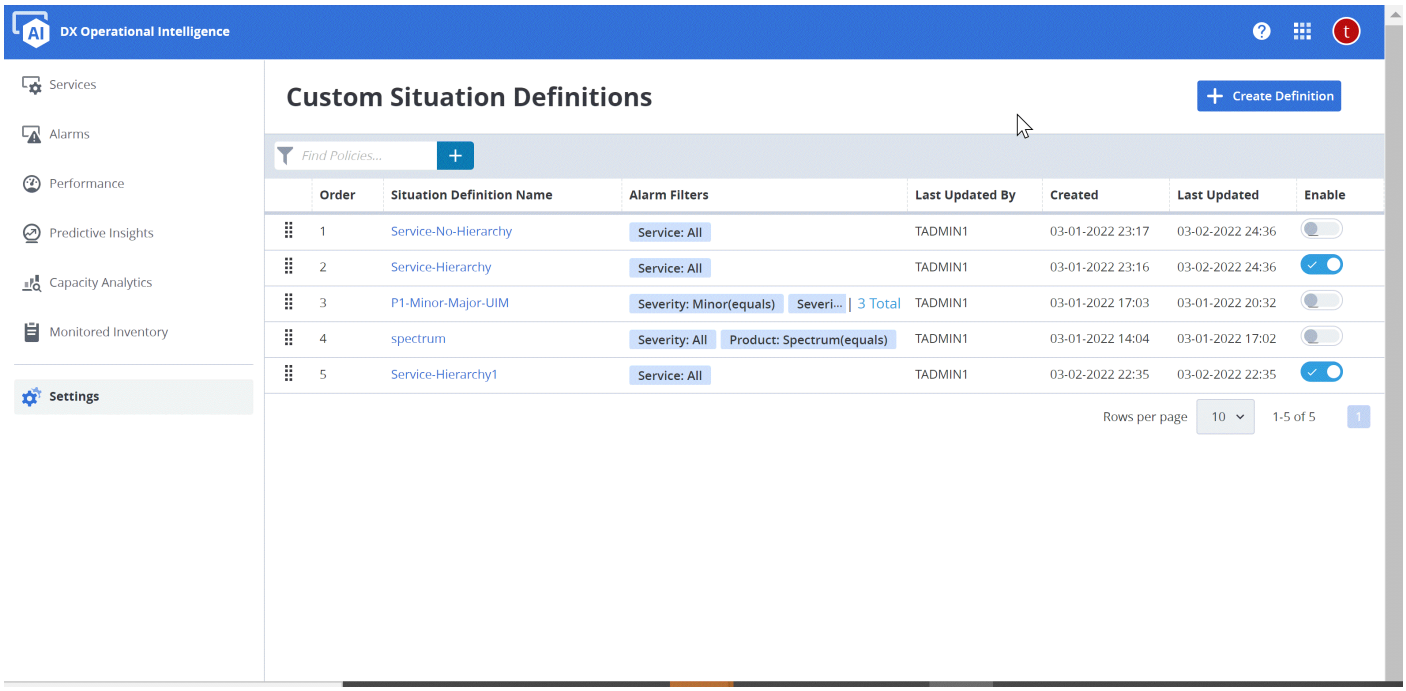
DX Operational Intelligence displays the results in the Preview Results section.

Prioritize Custom Situation Definitions

DX Operational Intelligence enables you to prioritize the custom situation definitions that the alarms must consider when the custom situation definitions have similar criteria.

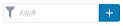
To prioritize the custom situation definitions, follow these steps:

1. Navigate to the Custom Situation Definitions page.
2. Use ⠿ to drag and drop the custom definitions per priority.



View Situation Clusters of Type Custom

You can view the situation clusters created based on the rules that are defined in the custom situation definition in Alarm Analytics.

1. Log in to DX Operational Intelligence, and click Alarm Analytics in the Left Navigation Menu.
2. Click ⋮ and select Situations.
3. Click  and define the following filter criteria:
   - Filter Attribute: 'Clustering Type'
   - Clustering Attribute: 'Custom'
4. Click Add.

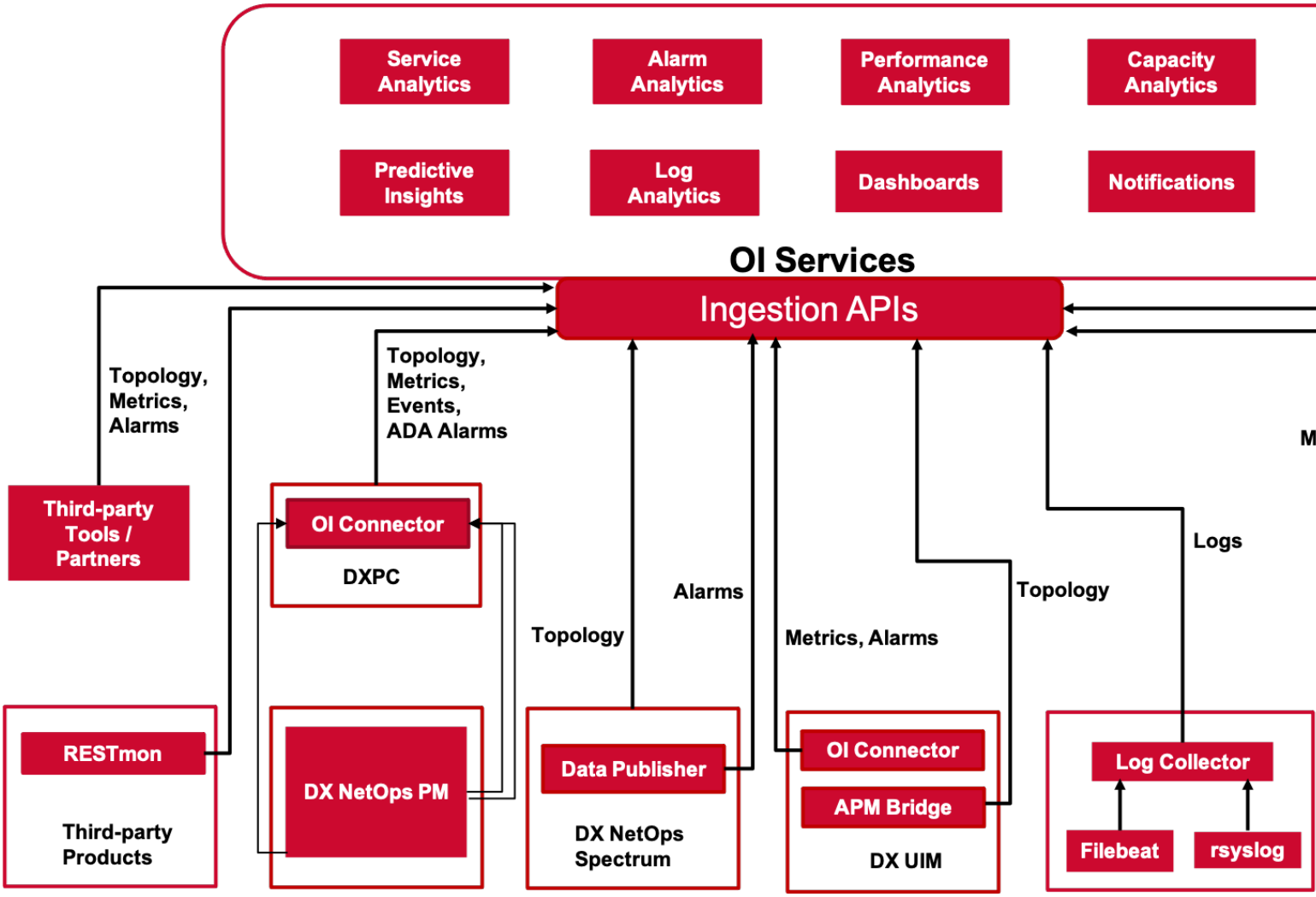
   The application filters the situation clusters generated based on the custom definitions.

Integration

Integrations enable you to connect various data sources with DX Operational Intelligence

Integration of CA Products or third-party tools with DX Operation Intelligence enables the data ingestion from the various data sources to the unified data lake. DX Operational Intelligence distributes this data to various DX Operational Intelligence capabilities.

The following diagram describes the flow of data from various integrated data sources to DX Operational Intelligence:



Integrations enable you to connect various data sources with DX Operational Intelligence. The integrated data is represented in the form of events, alarms, services, inventory, topology, and dashboards in DX Operational Intelligence.DX Operational Intelligence supports the following types of integration:

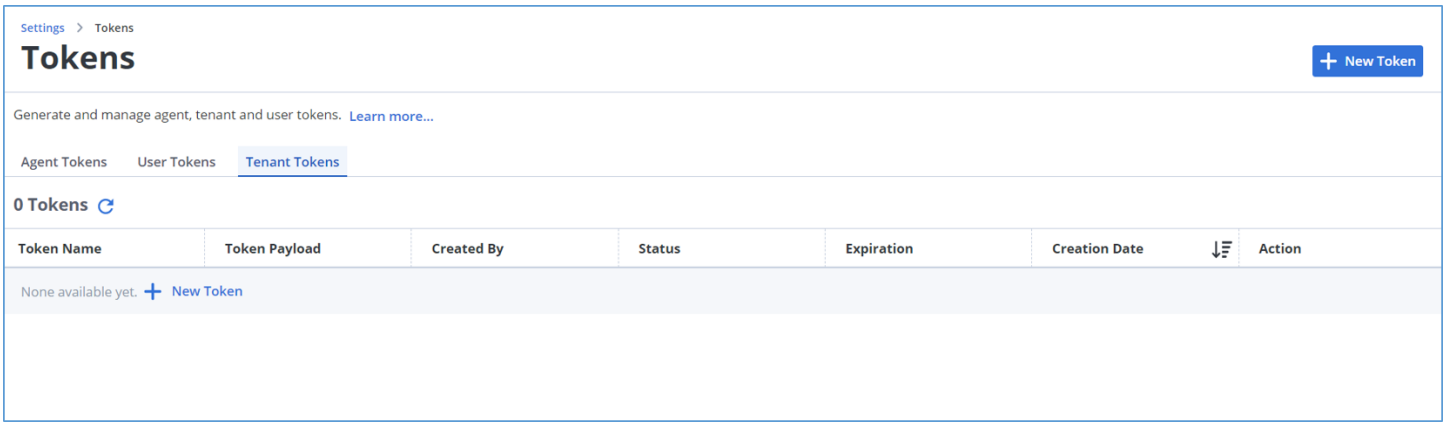
- Inbound Integration
  - Third-Party Integration
- Outbound Integration

NOTE
DX APM SaaS is integrated out-of-the-box. If you want to use DX APM On-premise and DX UIM On-premise, then verify the version of the connector and integrate accordingly. Integrating SaaS and On-premise simultaneously is not supported.

Product Integrations with DX Operational Intelligence	Integration Business Value
• DX Performance Management (DX PM)	Use this integration to apply network capabilities and service quality metrics to validate the application performance that is delivered over the network. You can validate the impact of changes and can resolve the issues faster. Your enterprise has an in-depth understanding of application response time and how the network impacts users. This knowledge helps you ensure the optimal user experience and improve the application performance management. More information: Integrate DX Performance Management
DX Unified Infrastructure Management	Use this integration to store UIM alarms, inventory, metrics (QoS), and UIM groups in DX Operational Intelligence (Elasticsearch). You can build Dashboards with the alarms, QoS, and inventory information. You can also store Spectrum alarms and inventory in DX Operational Intelligence (Elasticsearch). More information: Integrate DX Unified Infrastructure Management
DX NetOps Spectrum	Use this integration to analyze, correlate, and proactively resolve the network issues and monitor the data in DX Operational Intelligence. You can learn the trends of device availability by groups. The groups can be created for different criteria using global collections. You can view the alarm information that is presented using multiple filters and is available out of the box or with custom filters. You can use custom attributes to customize the out-of-the-box dashboards. You can also drill down through the dashboards to view the detailed information. More information: Integrate DX NetOps Spectrum
DX Application Performance Management	Use this integration to analyze, correlate, and monitor the data on DX Operational Intelligence. The integration provides an end-to-end unified view of applications and services. You can apply on the out-of-the-box dashboards that are based on the collected DX APM metrics. You can correlate anomalies and alerts of DX APM metrics and view logical clusters of APM alarms providing better insights for the ingested DX APM data. You can also determine the root cause for the correlated alarms that are ingested from DX APM. More information: Integrate DX Application Performance Management
Ingest Third-Party Data with DX RESTmon	Use this integration to monitor any technology or device data with the DX RESTmon. You can use one of the out-of-the-box schemas, or can use the templates that are provided by DX RESTmon to upload the schema. These schemas or the templates detail the QoS and the aggregation logic with the HTTP/HTTPS REST endpoints. You can define the metrics, alarms, inventory, and topology, to populate dashboards for the monitored devices. More information: Ingest Third-Party Data with DX RESTmon.

Token Management

DX Operational Intelligence uses the security tokens to authenticate requests and authorize access to automate user workflows using REST APIs. The following image illustrates the Tokens page for a Tenant Administrator:



Depending on your role, you can generate the following token types:

- **Agent Tokens:** You can use this token for connectors to ingest data into DX Operational Intelligence. These tokens have write-only access. Only a Tenant Administrator and a Security Administrator can generate an agent token.
- **User Tokens:** You can use this token as the authorization token to automate the user workflows. These tokens have both read and write access. They also can be audited. A Tenant Administrator, Security Administrator, Power User, User, or any user that is assigned a custom role can generate a user token.
- **Tenant Tokens:** You can use this token to provide access to a tenant. Only a Tenant Administrator can generate a tenant token.

{"URL":["https://cloudmanagement/#/settings/tokens"],"description":"concept.dita_6591e9e7-f7d7-440a-b550-4774d4106a46","new":"","new_video":"","admin":"","troubleshooting":{"masterkb":"","text":"","URL":[]},"pendo":"","video":[]}

Generate a Token as Tenant Administrator

As a Tenant Administrator, you can generate an agent token, user token, and tenant token. The generated token does not have an expiry date.

NOTE

- A Tenant Administrator can revoke and unrevoke only an Agent and Tenant token.
- Ensure to preserve the token carefully because the generated token will not be available once you close the New Token pane.
- You can filter the Agent Token and Tenant Token by Created By, Token Name, and Token Payload. You can filter the User Token by Token Name, Token Payload, and User ID.

Follow these steps:

1. Log in to DX Operational Intelligence as a Tenant Administrator.
2. Click **Settings** in the left navigation pane.
3. Under **Environment Definition and Access**, click **Tokens**.
   The Tokens page appears.

4. Click **+ New Token**.
5. Enter the following details:
   - **Token Name:** Enter a name for the token.
   - **Type:** Select the token type. Values: Agent, User, Tenant
6. Click **Generate**.

   The token is generated and is displayed.
7. Copy the generated token or click the **Copy** button to copy the token to the clipboard.
8. Click **Close**.

   The generated token is added to the respective tab with the following details: Token name, Token payload, Created by, Status, Expiration, and Creation Date.

Generate a Token as Security Administrator

As a Security Administrator, you can generate an agent token and user token. The generated token does not have an expiry date.

NOTE

- Ensure to preserve the token carefully because the generated token will not be available once you close the New Token pane.
- You can filter the Agent Token by Created By, Token Name, and Token Payload. You can filter the User Token by Token Name, Token Payload, and User ID.
- As a Security Administrator, you can revoke and unrevoke only an Agent token and User token.

Follow these steps:

1. Log in to DX Operational Intelligence as a Tenant Administrator.
2. Click **Settings** in the left navigation pane.
3. Under **Environment Definition and Access**, click **Tokens**.

   The Tokens page appears.
4. Click **+ New Token**.
5. Enter the following details:
   - **Token Name:** Enter a name for the token.
   - **Type:** Select the token type. Values: Agent, User
6. Click **Generate**.

   The token is generated and is displayed.
7. Copy the generated token or click the **Copy** button to copy the token to the clipboard. Ensure to preserve this token carefully because this token will not be available once you close the New Token pane.
8. Click **Close**.

   The generated token is added to the respective tab with the following details: Token name, Token payload, Created by, Status, Expiration, and Creation Date.

Generate a Token as Power User, User, or User With Custom Role

As a Power User, User, or a User with Custom Role, you can generate a token only for your account. The generated token does not have an expiry date. You can filter the user tokens by Token Name, Token Payload, and User ID. You can delete only the tokens that you have generated.

NOTE

- As a best practice, associate automating the user workflows to individual users as opposed to the generic service account.
- Ensure to preserve the generated token carefully because this token will not be available once you close the New Token pane.

Follow these steps:

1. Log in to DX Operational Intelligence as a Tenant Administrator.
2. Click **Settings** in the left navigation pane.
3. Under **Environment Definition and Access**, click **Tokens**.
   The Tokens page appears.
4. Click **+ New Token**.
5. Enter a name for the token.
6. Click **Generate**.
   The token is generated and is displayed.
7. Copy the generated token or click the **Copy** button to copy the token to the clipboard.
8. Click **Close**.
   The generated token is added to the respective tab with the following details: Token name, Token payload, Created by, Status, Expiration, and Creation Date.

Inbound Integration

Inbound integrations allow DX Operational Intelligence to ingest data from other products.

The inbound integration allows you to integrate DX Operational Intelligence with Broadcom products using connectors. The connectors collect alarms, inventory, topology, events, and metrics data from data sources and send to Key defintion for "pname" not found in the DITA map. For the supported products and connectors version, see Compatibility Matrix.

DX Operational Intelligence supports integration with the following Broadcom products:

- Integrate DX Application Performance Management
- Integrate DX NetOps Spectrum
- Integrate DX NetOps Performance Management
- Integrate DX Unified Infrastructure Management
- Integrate DX App Synthetic Monitor

Connector Parameters

{"URL":["https://cloudmanagement/#/settings/connector"],"description":"concept.dita_3284dd8b-2244-4fe6-bc02-c092fac70f2c","new":"","new_video":"","admin":"","troubleshooting":{"masterkb":"","text":"","URL":[]},"pendo":"","video":[]}

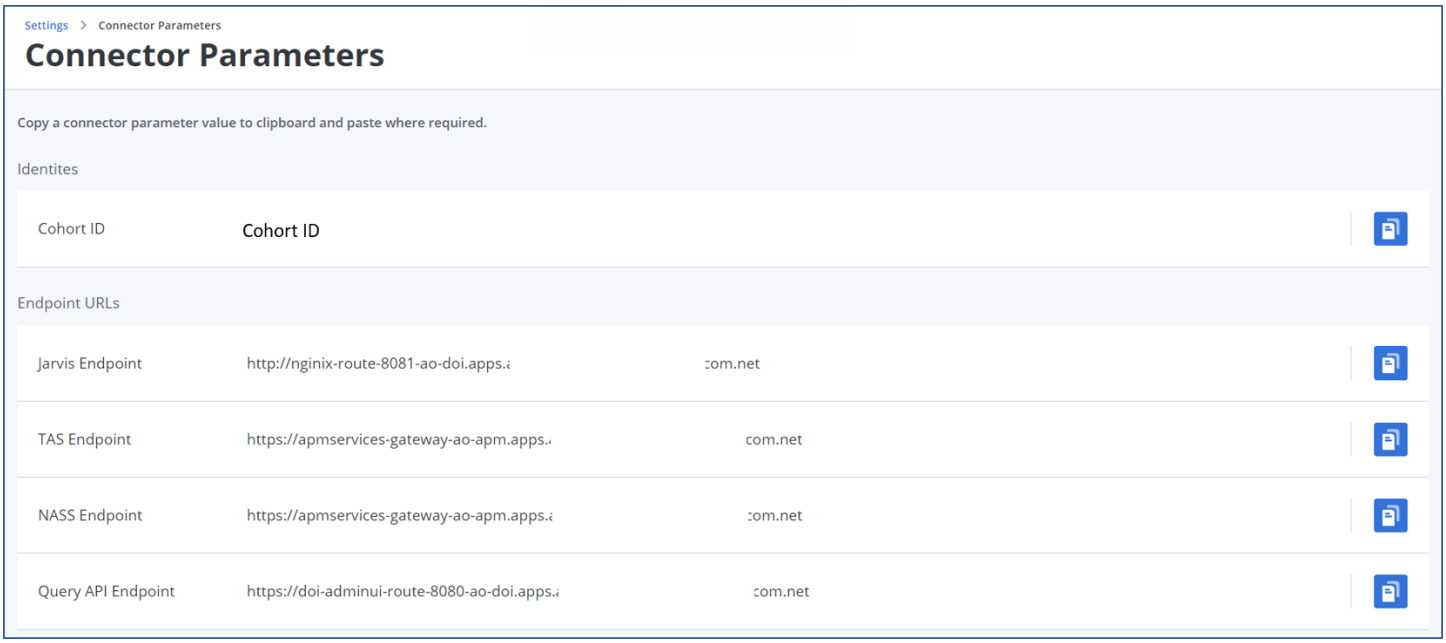
The **Connectors Parameters** page provides an overview of the mandatory parameters for some integrations so that you can easily configure connectors. Log in to **DX Operational Intelligence** and navigate to **Settings, Connector Parameters** tile to access these parameters. You can copy the values for these parameters using the **Clipboard** icon.

Use these parameters to configure the following connectors:

- Spectrum Data Publisher
- OI Connector
- oi_connector probe and apm_bridge probe
- DX Operational Intelligence Plugin
- RESTmon

The following image illustrates the Connector Parameters page:

Figure 3: Connector Parameters



These parameters are grouped into the following categories:

- **Identities:** Displays the Cohort ID value.
- **Endpoint URLs:** Displays endpoint URLs for Jarvis, TAS, NAS, and Query API.

Monitoring

This section provides the following information:

- Connector Health Monitoring
- RESTMon Self-Monitoring

Connector Health Monitoring

DX Operational Intelligence generates alarms that are related to the Connector's health based on pre-configured rules. For example, when the Connector host has consumed 95% of the memory usage, DX Operational Intelligence generates a Critical alarm which is displayed on the **All Alarms** page.

> **NOTE**
> The generated alarms are displayed on the **All Alarms** page with the **Alarm Type** as **ConnectorHealth**.

Currently, alarms are generated for the following metrics:

- **Uptime**

This metric indicates the time the Connector has been up and running since the Connector start time. This value either increases or is absent. If this value is absent, an alert is generated based on the following rules:

- **Major Alert:** A major alert is generated if the last five intervals do not have any values.
- **Critical Alert:** A critical alert is generated if the last 15 intervals did not have any values.

- **IntervalSinceLastSuccessfulPush**
  This metric indicates the number of intervals that have passed since the Connector has pushed the data successfully. The metric value is zero if the Connector pushes the data successfully. However, if the Connector is unable to push the data, the metric value is incremented by one for every interval the push is not successful. An alert is also generated based on the following rules:
  – **Major Alert:** A major alert is generated if the Connector is unable to push data for the last three intervals.
  – **Critical Alert:** A critical alert is generated if the Connector is unable to push data for the last five intervals.
- **Memory Usage**
  This metric indicates the percentage of the total memory that is used by the Connector host. When the memory usage exceeds the threshold value, an alert is generated based on the following rules:
  – **Major Alert:** A major alert is generated when the memory usage is greater than 80% of the total memory.
  – **Critical Alert:** A critical alert is generated when the memory usage is greater than 90% of the total memory.

RESTMon Self-Monitoring

This section provides the following information:

- Monitoring Dashboards
- Supportability Metrics
- Liveness and Readiness Check

Integration with CA Products

Integrate DX Application Performance Management

DX Operational Intelligence supports an integration with DX Application Performance Management. DX Application Performance Management provides large-scale monitoring for modern application stacks.

> **NOTE**
> Before you begin with the integration, collect the following information:
>
> - Mandatory parameters required for this integration. For more information, see Connector Parameters.
> - Supported version for this integration. For more information, see DX Operational Intelligence Interoperability.

Integrate DX Application Performance Management (DX APM) to analyze and correlate data and display the data on DX Operational Intelligence. The integration provides an end-to-end unified view of applications and services. DX APM integrates with DX Operational Intelligence using DX Operational Intelligence plugin. This plugin sends the following data to DX Operational Intelligence:

- Metrics
- Alerts
- Inventory

With this integration, your enterprise gains the following benefits:

- Out-of-the-box dashboards that are based on the collected DX APM metrics
- Correlates anomalies and alerts of DX APM metrics and view logical clusters of APM alarms providing better insights for the ingested DX APM data.
- Determine the root cause for the correlated alarms that are ingested from DX APM.

To integrate DX Operational Intelligence On-Prem/SaaS with DX Application Performance Management, 10.7 see Integrate DX APM On-Prem with DX Operational Intelligence

To integrate DX Operational Intelligence On-Prem with DX Application Performance Management 21.3.1, see Digital Operational Intelligence Plugin Guide

Integrate DX App Synthetic Monitor

Integrate DX Operational Intelligence and DX App Synthetic Monitor to use ASM monitor metric for service availability.

This integration sends the metric **Last Check Status** to DX Operational Intelligence.

The integration is complete when you:

- Create one or more monitoring rules in App Synthetic Monitor. See the Getting Started page for insight into creating and organizing rules.
- Download DX App Synthetic Monitor Agent. For more information, see the Download App Synthetic Monitor Agent section.
- Configure DX App Synthetic Monitor Agent
- Verify ASM Metrics in DX Operational Intelligence

Configure DX App Synthetic Monitor Agent

You can configure DX App Synthetic Monitor Agent settings in DX Application Performance Management to get the DX Operational Intelligence metrics.

Follow these steps:

1. Log into DX Application Performance Management.
2. Navigate to **Settings**.
3. Click **APM Command Center** under the **Integrations** tile.
4. Navigate to **Packages** in the **DX APM Command Center** and click **New**.
5. Enter *Agent Package Name* in **Agent Builder**, and then, in **Environment** select **OS Type** and **Process**: *CA APM Infrastructure Agent*.
6. Navigate to **Select Bundles** and select **ASM Monitor** from **Available Bundles** and click **Add** to add it to the **Selected Bundles**.
7. Click **Next**.
8. In the **Configure Bundles** page select the **ASM Monitor** bundle and configure the following properties for ASM integration:
   a) **asm.userEmail**: Enter the email address for the ASM account that enables access to the ASM API.
   b) **asm.APIPassword**: Enter the password for the ASM account that enables access to the ASM API.
   c) **introscope.agent.agentName:** Enter the agent name as **App Synthetic Monitor Agent, Agent,** or any other name. DX Operational Intelligence uses this setting as a filter to display the metrics.
9. Click **Done** for the configuration to apply to the integration. You can further download the agent from Bundles and view the metrics in DX Operational Intelligence.

Verify ASM Data in DX Operational Intelligence

After you have deployed the ASM agent, verify that the ASM metric appears in DX Operational Intelligence.

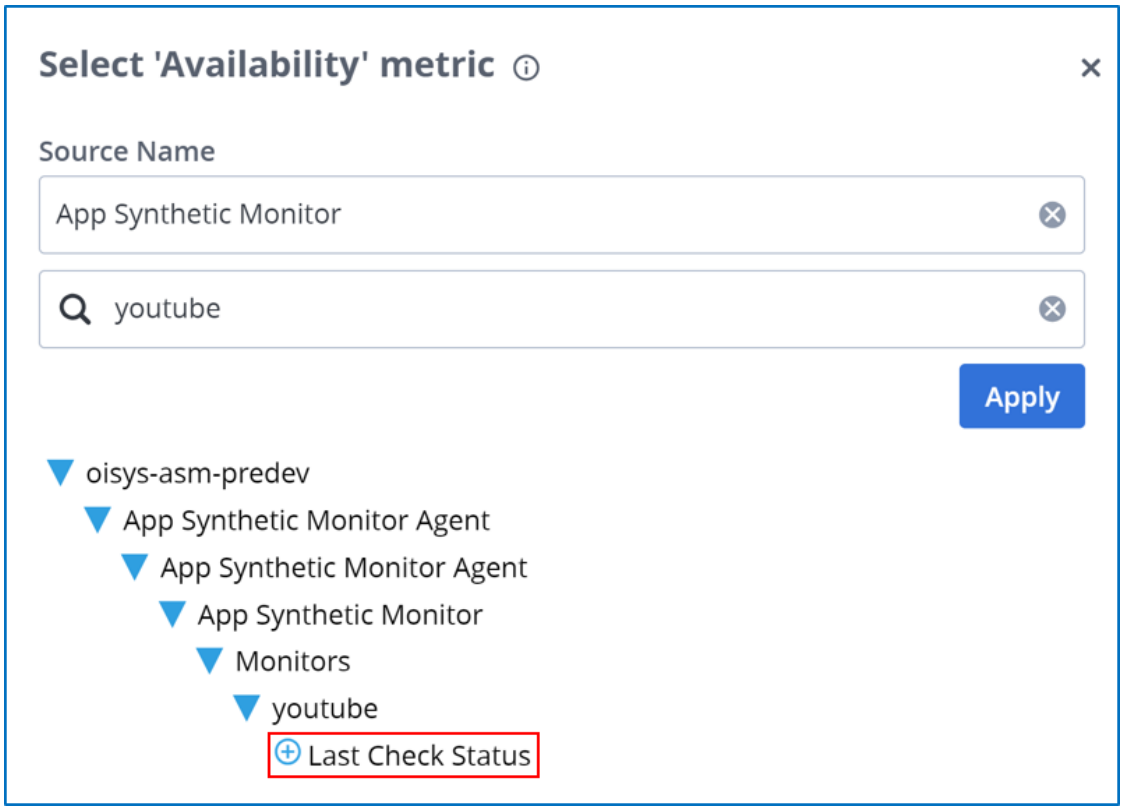
Follow these steps:

1. Log in to DX Operational Intelligence.
2. Navigate to the **Service Analytics** page.
3. Click the required service.
4. Click **More Options** in the top-right corner and click **Edit Service**.
   The **Service Details** panel is displayed.
5. Select the metric to participate in the **Availability**:
   NOTE
   Only the **Last Check Status** metric can be participate in the Availability.
   a. Navigate to the **Key Performance Indicators** section.
   b. Click the **Availability** drop-down and select **App Synthetic Monitor (ASM)**.
   c. Click ... next to **Select ASM Metric**.
   The **Select 'Availability' metric** pop-up window is displayed.
   d. Select the metric in one of the following ways:
   NOTE
   Only the **Last Check Status** metric can be participate in the Availability.

- Navigate to the required monitor.
- Filter the monitor:
  a. Enter the required information to filter the monitor:
     - **Source Name:** Enter the ASM source name. By default, the source name is displayed as **App Synthetic Monitor**.

       NOTE
       Ensure that the source name is the name you had configured for the **introscope.agent.agentName** property earlier.
     - **Filter Loaded Monitors:** Enter the name of the monitor if necessary.

  b. Click **Apply.**
     The metric is displayed as shown in the example:



NOTE
If no metric data is found, verify that the ASM source name matches the **introscope.agent.agentName** property configuration.

e. Click the ⊕ icon to select the metric.
   The metric is added to the **Availability** section.

Related Links

Create a Service on page 709
Create a service by defining multi-hierarchy services using service discovery and topologies across different monitoring domains like application, infrastructure, and network.

Edit a Service on page 734
The Edit service allows you to modify the service and sub-service properties and topologies. You can also manage elements, and modify service or sub-service details such as service name, description, tags associated with the service, availability metrics, location, and custom properties.

Integrate DX NetOps Performance Management

DX Operational Intelligence supports an integration with DX NetOps Performance Management. DX NetOps Performance Management monitors, stores, analyzes, and displays information for assuring service quality.

NOTE
Before you begin with the integration, collect the following information:

- Mandatory parameters required for this integration. For more information, see Connector Parameters.
- Supported version for this integration. For more information, see DX Operational Intelligence Interoperability.

Integrate DX NetOps Performance Management (PM) with DX Operational Intelligence using the OI Connector. The OI Connector collects the metrics, events, inventory, and topology data and displays it in DX Operational Intelligence.

With this integration, your enterprise gains the following benefits:

- Apply network capabilities and service quality metrics to validate application performance that is delivered over the network. You can validate the impact of changes and can solve problems faster.
- Your enterprise has an in-depth understanding of application response time and how the network impacts users. This knowledge helps you ensure optimal user experience and improve application performance management.

To integrate DX Operational Intelligence with DX NetOps PM, see Integrate with DX Operational Intelligence.

Integrate DX NetOps Spectrum

DX Operational Intelligence supports the integration with DX NetOps Spectrum. DX NetOps Spectrum provides sophisticated event and network fault management capabilities.

NOTE
Before you begin with the integration, collect the following information:

- Mandatory parameters required for this integration. For more information, see Connector Parameters
- Supported version for this integration. For more information, see DX Operational Intelligence Interoperability.

Integrate DX NetOps Spectrum to analyze, correlate, and proactively resolve the network issues and display the data in DX Operational Intelligence. DX NetOps Spectrum integrates with DX Operational Intelligence using Spectrum Data Publisher. The Spectrum Data Publisher is a utility/service in DX NetOps Spectrum that collects the following data and publishes this data to DX Operational Intelligence.

With this integration, IT operators can do the following:

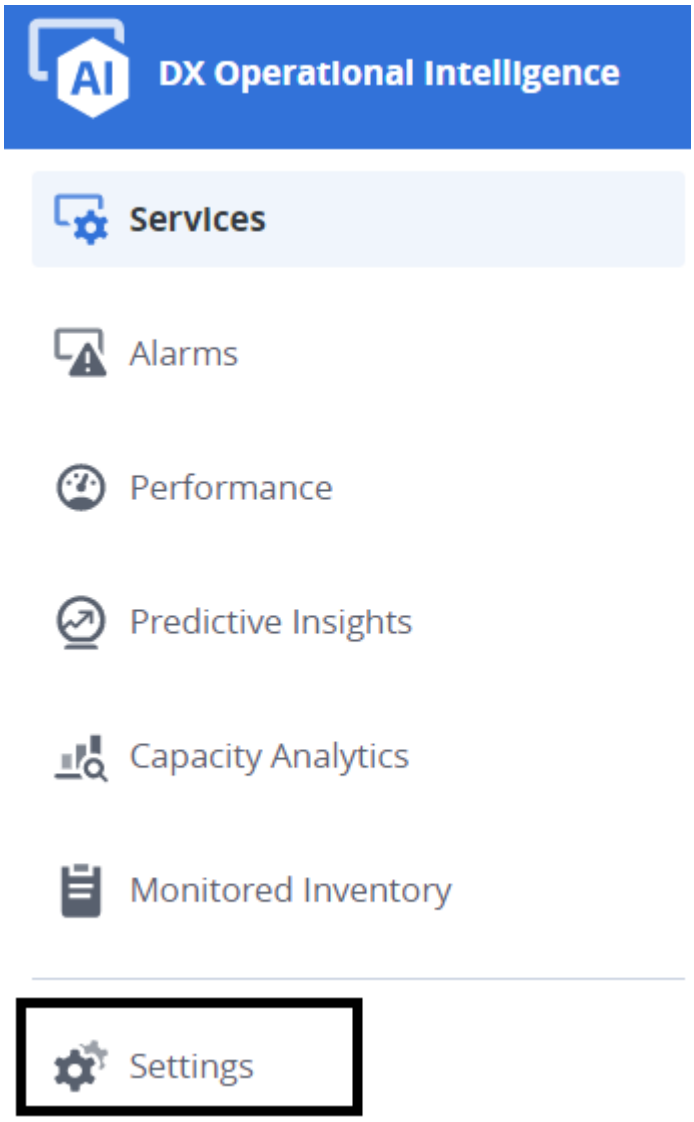
- Know the trends of device availability by groups. The groups can be created for different criteria using global collections.
- View the alarm information that is presented using multiple filters and is available out of the box or with custom filters.
- Use custom attributes to customize the out-of-the-box dashboards.
- Drill down through the dashboards to view detailed information.

Download Spectrum Data Publisher Installer

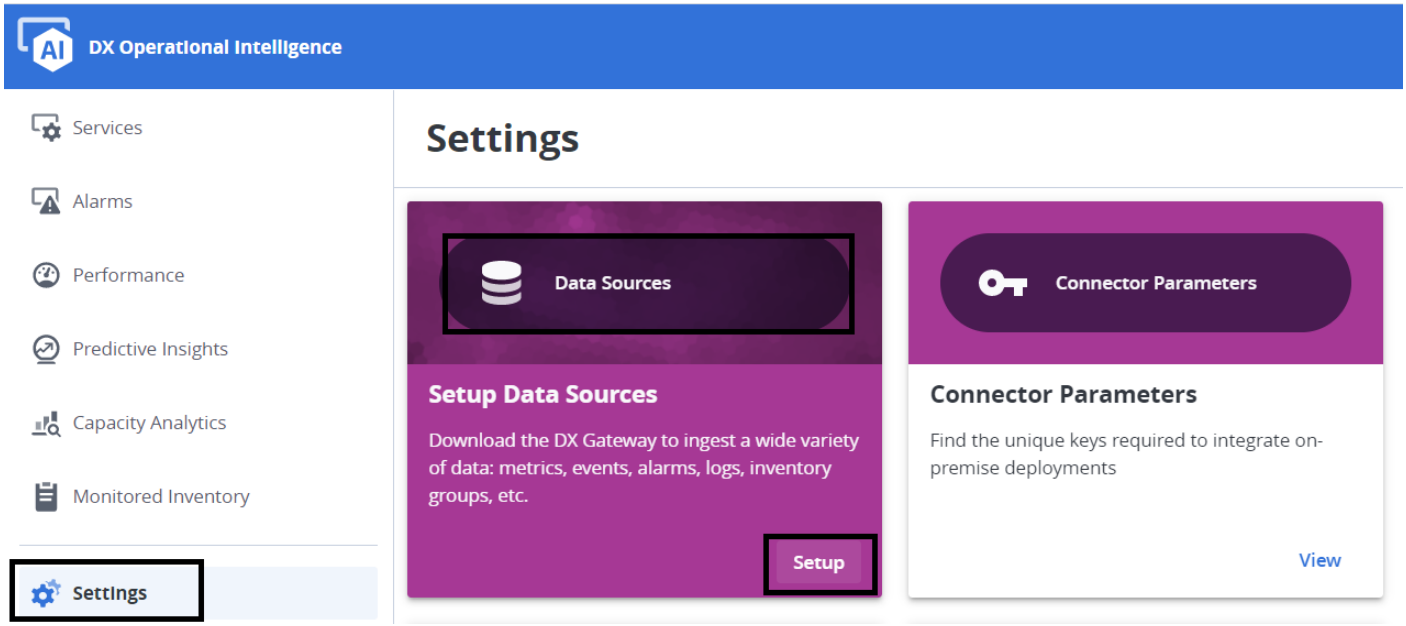
You can download the DX NetOps Spectrum Data Publisher installer from DX Operational Intelligence. The installer (`SpectrumDataPublisher-<version_number>.zip`) is pre-populated with the tenant ID, tenant token, and DX Operational Intelligence URL.

To download the installer, follow these steps:

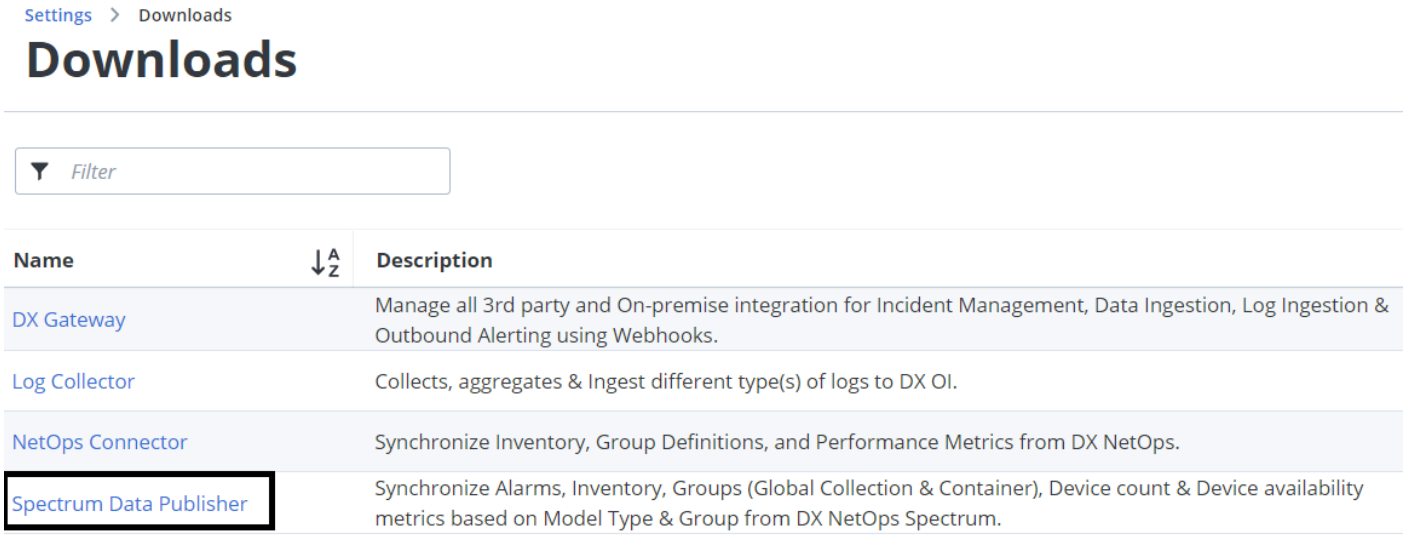
1. From the DX Operational Intelligence UI, navigate to the **Settings** page.



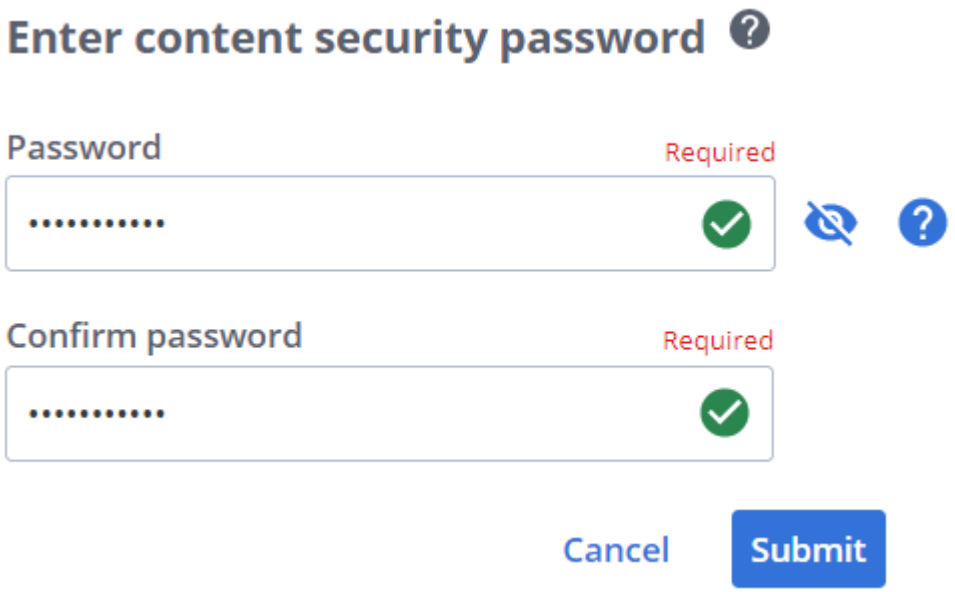
2. Click **Setup** on the **Data Sources** tile.



3. Click **Spectrum Data Publisher**.



4. On the Enter Content security password screen, enter the **Password** and **Confirm password**.



NOTE
This content security password is used to encrypt the tenant details. Note down this password. The installer prompts you to enter this password while installing the Spectrum Data Publisher.

5. Click **Submit**.
   The *SpectrumDataPublisher-<version_number>.zip* file gets downloaded to the default location. For example, if your default location is the **Downloads** folder, the *SpectrumDataPublisher-21.2.6.0.zip* file is downloaded to the **Downloads** folder.
6. Extract the *SpectrumDataPublisher-<version_number>.zip* file and extracted folder contain the following files:
   - **generic_config.json:** This file contains the cohort ID and token in an encrypted format. The installer decrypts the contents and uses them to pre-populate the tenant details like tenant ID, tenant token, and DX Operational Intelligence URL in the `connectorconfig.xml` file. In case of a fresh installation, you must manually provide the other parameters in the `connectorconfig.xml` file. However, during an upgrade, the parameter values in the previous version are retained in the `connectorconfig.xml` file.

     NOTE
     The generic_config.json file is deleted after the installation.
   - **SpectrumDataPublisher.jar:** Use this file to install SpectrumDataPublisher. The **SpectrumDataPublisher.jar** file extracts and launches the DX NetOps Spectrum - DX OI Connector. The property file (generic_config.json) and the installer must be in the same folder. The installation loads and uses the properties from the property file. The installer prompts you to enter the **Content security password** while installing SpectrumDataPublisher.

     NOTE
     During an upgrade, the previous values for tenant ID, tenant token, and DX Operational Intelligence URL are replaced with the values in the `generic_config.json` file.

The Spectrum Data Publisher is installed. After you install the Spectrum Data Publisher, configure the connector to synchronize data from DX NetOps Spectrum to DX Operational Intelligence.

For more information see, Integrate with DX Operational Intelligence

Southbound Gateway

To send alarm actions updates to DX NetOps Spectrum, configure the southbound gateways in DX Operational Intelligence.

You configure southbound gateways for sending alarm actions updates to DX NetOps Spectrum.

WARNING

- Southbound Gateway supports one instance of DX NetOps Spectrum.
- Sending alarm actions updates to custom products is not supported.

To manage southbound gateways, follow these steps:

1. From the DX Operational Intelligence navigation panel, click the (gear) icon.
   The **Settings** page appears.
2. Click **Southbound Gateway**.
   The Southbound Gateway page appears.

Configure a Southbound Gateway

You can configure the southbound gateway for a new source product with DX Operational Intelligence.

Follow these steps:

1. Navigate to Southbound Gateway page.
2. Based on your requirement, click Spectrum.
   The **Edit Southbound Gateway** page appears.
   a. **Spectrum Configuration:**
      a. **Name**, **Product Name**, and **Authentication Type** values are pre-populated. You cannot edit these fields.
      b. **URL:** Enter the Spectrum OneClick URL to send ticket actions from DX Operational Intelligence to Spectrum. For example, *http://<SpectrumOneClick_Hostname>/<port>/spectrum*
      c. Enter the username and password of Spectrum OneClick.
      d. **Map Attributes:** Specify the custom attribute value for ticket synchronization. By default, the following attributes are mapped:
         - Trouble Shooter
         - Ticket ID
         - Acknowledged
         - Clear
3. Click **Save**
4. Click **Done**.
   The southbound gateway is configured for Spectrum.

Integrate DX Unified Infrastructure Management

DX Operational Intelligence supports integration with DX Unified Infrastructure Management. DX Unified Infrastructure Management is a scalable IT monitoring solution.

NOTE
Before you begin with the integration, collect the following information:

- Mandatory parameters required for this integration. For more information, see Mandatory Parameters for Integration.
- Supported version for this integration. For more information, see DX Operational Intelligence Interoperability.

Integrate DX Unified Infrastructure Management (DX UIM) using the DX Operational Intelligence **oi_connector** probe and **apm_bridge** probe to analyze the data and display it in the DX Operational Intelligence UI.

- The **oi_connector** probe collects the following data and sends the data to DX Operational Intelligence

  - Metrics and metrics metadata
  - Alarm data
  - UIM groups
- The **apm_bridge** probe collects the Topology data and sends the data to DX Operational Intelligence.

To connect DX UIM with DX Operational Intelligence, and to monitor DX UIM data you must configure the oi_connector probe. To configure the oi_connector probe, see oi_connector Configuration.

To send topology data from the DX UIM probes to DX Operational Intelligence, you must configure *apm_bridge* probe. To configure the *apm_bridge* probe, see apm_bridge Configuration.

Southbound Gateway

To send alarm action updates to DX Unified Infrastructure Management, configure the southbound gateways in DX Operational Intelligence.

You configure southbound gateways for sending alarm actions updates to CA Unified Infrastructure Management.

WARNING

- Southbound Gateway supports one instance of CA UIM for a single tenant.
- Sending alarm actions updates to custom products is not supported.

To manage southbound gateways, follow these steps:

1. From the DX Operational Intelligence navigation panel, click the (gear) icon.
   The **Settings** page appears.
2. Click **Southbound Gateway**.
   The Southbound Gateway page appears.

Configure a Southbound Gateway

You can configure the southbound gateway for a new source product with DX Operational Intelligence.

Follow these steps:

1. Navigate to Southbound Gateway page.
2. Based on your requirement, click UIM.
   The **Edit Southbound Gateway** page appears.
   a. **UIM Configuration:**
      a. **Name**, **Product Name**, and **Authentication Type** values are pre-populated. You cannot edit these fields.
      b. **IM SaaS:** Use this toggle button to send alarm updates to IM SaaS or UIM.

         NOTE

         By default, this toggle button enabled which sends alarm updates to IM SaaS.
      c. **URL:** Enter the UIM URL to send ticket actions from DX Operational Intelligence to UIM.

         NOTE

         You must configure the Swagger probe for sending the tickets updates to UIM.
      d. Enter the username and password of UMP or UIM server.
         For example, http://<UIMUMP_Hostname>/<Port>/uimapi
      e. **Map Attributes:** Specify the custom attribute value for ticket synchronization. By default, custom_1 value is provided.
   b. **Spectrum Configuration:**
      a. **Name**, **Product Name**, and **Authentication Type** values are pre-populated. You cannot edit these fields.

b. **URL:** Enter the Spectrum OneClick URL to send ticket actions from DX Operational Intelligence to Spectrum. For example, http://<SpectrumOneClick_Hostname>/<Port>/spectrum
c. Enter the username and password of Spectrum OneClick.
d. **Map Attributes:** Specify the custom attribute value for ticket synchronization. By default, the following attributes are mapped:
   - Trouble Shooter
   - Ticket ID
   - Acknowledged
   - Clear

3. Click **Save**
4. Click **Done**.
   The southbound gateway is configured for UIM.

Integration with Third-Party Products

DX Operational Intelligence supports integration with third-party tools to monitor your IT environment using the DX RESTmon Utility.

Integration of DX Operational Intelligence with your infrastructure management applications has the following advantages:

- View infrastructure data of your enterprise in one convenient location.
- Build a rich data lake that you can visualize and search.
- Access broader source of data for analytics.

Ingestion APIs

You can integrate your third-party data into DX Operational Intelligence to view the infrastructure data of your enterprise in one convenient location. If you are using a third-party tool to monitor your IT environment, you can use the REST API to insert data into Jarvis. This inserted data is available in the data lake.

The ingestion APIs enable you to insert the following data: Alarms, Events, Metrics, and Topology. The APIs only support JSON-formatted payloads. The fields in the JSON payload for each data type are predefined. The product sending data must use the format that is described in this section. The API accepts an array of JSON documents to be ingested in bulk. However, a single document within an array can also be sent.

DX Operational Intelligence provides endpoints for each data that you can ingest. The list of ingestion APIs is:

- Alarm Ingestion API
- Change Events Ingestion API
- Metrics Ingestion
  - Metric Registration API
  - Metrics Ingestion API
  - Metric data Validation API
- Topology Ingestion API
  - Topology Validation API

Alarm Ingestion

Using the Alarm Ingestion API, you can ingest the alarm data from external sources into the DX Operational Intelligence data lake.

Resource URI

You can select the URI based on the deployment and region of your tenant:

- DX SaaS - USA:
  `https://apmgw.dxi-na1.saas.broadcom.com/jarvis/v2/ingestion`
- DX SaaS - EU:
  `https://apmgw.dxi-eu1.saas.broadcom.com/jarvis/v2/ingestion`

```
http://<APM gateway Route>/jarvis/v2/ingestion
```

Method

```
POST
```

Authorization

Authorization: Bearer {<APM Gateway Tenant token>}

For more information on generating the token, see Authentication and Authorization of APIs

HTTP Headers

Content-Type: application/json

index_name: ao_itoa_alarms_custom_1

Request Payload Syntax

```
{
    "documents" : [
        {
             "header" : {
             },
           "body" : [
                 {
                 },
                {
                },
               .
               .
             ]
         }
     ]
}
```

Response Syntax

```
{
    "code": 202,
}
```

Sample Request-Response

```
https://apmgw.dxi-na1.saas.broadcom.com/jarvis/v2/ingestion
http://apmservices-gateway.10.0.0.0.nip.io/jarvis/v2/ingestion
```

Sample Request Payload for Alarms


```
{
    "documents": [{
            "header": {
                "tenant_id": "ADD7BFA4-15B1-48EA-BD2D-F4E01243A384",
                "product_id": "ao",
                "doc_type_id": "itoa_alarms_custom",
                "doc_type_version": "1",
                "unique_id": "$['alarm_unique_id']"
            },
            "body": [{
                    "alarm_unique_id": "b8014f92c277196aa49001d23b1b94e7",
                    "alarmType": "Application",
                    "host": "HCLHVF2.Broadcom.net",
                    "product": "Dynatrace",
                    "product_version": "",
                    "summary": "Alarms",
                    "metric_name": "Disk Read Time",
                    "metric_type": "Disk",
                    "message": "SLOW_DISK: The Disk Read Time value is greater than the threshold
 value of 1",
                    "startTime": "2020-03-03T05:36:00+0000",
                    "severity": "major",
                    "timestamp": "2020-03-03T06:50:04+0000",
                    "ci_unique_id": "HOST-6DE9560308323DF3",
                    "configuration_item_type": "Host",
                    "configuration_item": "",
                    "tags": ["Dynatrace", "Alarms", "SLOW_DISK"],
                    "metric_unique_id": "roC-B-yI-1sw0w4",
                    "status": "NEW"

                }
            ]
        }
    ]
}
```


Table 1: Parameters for Alarms

Parameter	Mandatory/Optional	Type	Description
tenant_id	Mandatory	Header	Provides the tenant UUID (also called cohort_id) for DX Operational Intelligence to identify the tenant. **Format**: String
product_id	Mandatory	Header	Provides the fixed value "ao". **Format**: String
doc_type_id	Mandatory	Header	Provides the fixed value "itoa_alarms_custom". **Format**: String
doc_type_version	Mandatory	Header	Provides the fixed value "1". **Format**: String

Parameter	Mandatory/Optional	Type	Description
unique_id	Mandatory	Header	Provides the fixed value "$['alarm_unique_id']". **Format**: String
alarm_unique_id	Mandatory	Body	Provides the alarm unique ID that is used to manage the lifecycle of an alarm. **Format**: String Ensure that the same unique ID is sent when the status of the alarm changes (from NEW → UPDATED → CLOSED). A new unique ID must be used after an alarm is closed and a new alarm is generated for a similar condition. Example: Threshold crossing on a metric
product	Mandatory	Body	Provides the source product name from which the alarm is generated. **Example**: NewRelic **Format**: String
message	Mandatory	Body	Provides the alarm message.
status	Mandatory	Body	Provides the status of the alarm. Valid values: NEW, UPDATED and CLOSED **Format**: String
severity	Mandatory	Body	Provides the severity of the alarm. Valid values: Critical, Major, Minor, and Information **Format**: String
timestamp	Mandatory	Body	Provides the last updated time of the alarm in ISO 8601 date string format. Format: yyyy-MM-dd'T'HH:mm:ssZ **Format**: String
startTime	Mandatory	Body	Provides the alarm creation time in ISO 8601 date string format. Format: yyyy-MM-dd'T'HH:mm:ssZ **Format**: String
ci_unique_id	Mandatory	Body	Provides the unique ID for the entity on which alarm is raised.
summary	Optional	Body	Provides a short summary of the alarm. **Format**: String
host	Optional	Body	Provides the hostname of the device or entity on which alarm was generated. **Format**: String
alarmType	Optional	Body	Provides the type of the alarm. **Example**: Application, Infrastructure **Format**: String
product_version	Optional	Body	Provides the version of the source product. **Format**: String
metric_name	Optional	Body	Provides the metric name, if the alarm was generated based on a metric threshold crossing. **Format**: String
metric_type	Optional	Body	Provides the metric type, if the alarm was generated based on a metric threshold crossing. **Format**: String

Parameter	Mandatory/Optional	Type	Description
configuration_item_type	Optional	Body	Provides the entity type on which alarm is raised. **Format**: String
configuration_item	Optional	Body	Provides the entity name on which alarm is raised. **Format**: String
tags	Optional	Body	Provides the Ad hoc string values. **Format**: Array of Strings
metric_unique_id	Optional	Body	Provides the Metric Store (NASS) unique ID. This parameter is mandatory for an alarm to enable the metric drill-down in DX Operational Intelligence.
external_ids	Optional	Body	Provides the unique external IDs for the entity on which the alarm is raised from the Topology Store (TAS). If the external IDs are not provided in the API request, to map the alarm with an entity in Topology Store, DX Operational Intelligence creates the external ID in the following format: "CUSTOM:<tenant_id>$$<product>$ $<ci_unique_id>" **Format**: Array of Strings For more information, see the External IDs section.
alarmURL	Optional	Body	Provides the URL link to the source product in which the alarm is generated. **Format**: String
custom_1, custom_2 custom_n	Optional	Body	Provides the custom information from the alarm of type 'String'. **Maximum Custom Fields**: 10 **Format**: String
custom_num_1 custom_num_2	Optional	Body	Provides the custom information from the alarm of type 'Number'. **Format**: Double

External IDs

An External ID is a unique textual identifier of a topology entity in the DX Topology Store (TAS). This ID is generated by the source ingesting the topology or inventory data to TAS and should be globally unique and stable within a tenant context. A Vertex ID is the internal unique ID for a topology entity and is a numeric value that is generated inside TAS.

The external ID must have the following structure:

```
externalId:<TOPOLOGY LAYER>:<Unique Textual String>
```

For example, the APM ATC Application Layer uses the following format:

[layer(ATC)]:[ApplicationName]:[Type]:[Name]:[agentHost]:[agentProcess]:[agentName]

```
ATC:MathProxy:GENERICFRONTEND:Apps|MathProxy|URLs|/MathProxy/rest/:tas-cz-n89:Tomcat-MathApp-BA-PO:Tomcat-
MathApp-BA-PO
```

It is recommended to use a similar format for the APM Infrastructure vertices as well.

[layer(INFRASTRUCTURE)]:[Type]:[ExtensionId]:[identification(name, Id, ...)]:[agentHost]:[agentProcess]:[agentName]

```
VirtualMachine:EC2:My Virtual Machine:VMGUID::extensionHost:extension:EC2
```

For any third-party data ingested into DX Operational Intelligence, use the following format for the external ID:

CUSTOM:<Tenant Cohort ID>$$<product>$$<ci_unique_id>

Where

- **CUSTOM** is the Topology Layer for third-party sources.
  The following table lists the Topology Layer values for the different types of data that is ingested into DX Operational Intelligence:

Source	Layer	Description
Third-party	CUSTOM	For all third-party integrations.
CA APM	ATC	Application layer data
CA APM	APM_INFRASTRUCTURE	Infra/IA agents and extensions
CA APM	INFRASTRUCTURE	APM agents
CA PM	NETWORK_CAPC	CA NetOps Performance Management
CA UIM	INFRASTRUCTURE_UIM	DX Unified Infrastructure Management
Spectrum	NETWORK_SPECTRUM	DX NetOps Spectrum

- **<Tenant Cohort ID>** is the Tenant UUID (You can fetch this ID from the Connector Parameters page)
- **<product>** is the name of the source product from which the data is fetched for ingestion. For example, Dynatrace.
- **<ci_unique_id>** is the unique identifier of the entity. For example, FQDN for a HOST/Device.

Alarms and metrics are associated with the corresponding external IDs of the topology entities on which the alarms and metrics are generated using the external_ids field. For example,

- **Alarm Payload** - external_ids: ["NETWORK_SPECTRUM:saas-map-vm02_0x1000000"]
- **Metric Metadata** - external_ids: ["INFRASTRUCTURE:HOST:bl6781.mycomp.com"]

If an alarm is not associated with an external ID, the alarm is not associated with any service in DX Operational Intelligence. Similarly, if metrics are not associated with an external ID, those metrics cannot be used in Service Analytics (including SLI/SLO), Capacity Analytics, and Anomaly Detection. Also, any alerts or alarms that are generated from those metrics also cannot be used for Service Analytics as they do not have any external ID associated.

> **NOTE**
> Any APM metric that is required for DX Operational Intelligence is enriched with the external_ids attribute (and other DX Operational Intelligence required attributes) only when the metric is part of the special OOTB APM Metric Group called **APM OI**. If your metric or alert is missing the external_ids attribute, check if the metric path is included in the **APM OI** metric group. If not, add the metric to the APM OI metric group. For more information, see the **APM** documentation.

Change Events Ingestion

Using the Change Events Ingestion API, you can ingest the change events data from external sources into the DX Operational Intelligence data lake.

Resource URI

You can select the URI based on the deployment and region of your tenant:

- DX SaaS - USA:
  `https://apmgw.dxi-na1.saas.broadcom.com/jarvis/v2/ingestion`
- DX SaaS - EU:

```
https://apmgw.dxi-eu1.saas.broadcom.com/jarvis/v2/ingestion
```

```
http://<APM gateway Route>/jarvis/v2/ingestion
```

Method

```
POST
```

Authorization

Authorization: Bearer {<APM Gateway Tenant token>}

For more information on generating the token, see Authentication and Authorization of APIs

HTTP Headers

Content-Type: application/json

index_name: index_name: ao_itoa_events_change_custom_1

Request Payload Syntax

```
{
    "documents" : [
        {
             "header" : {
             },
           "body" : [
                 {
                 },
                {
                },
               .
               .
             ]
          }
     ]
}
```

Response Syntax

```
{
    "code": 202,
}
```

Sample Request-Response

Sample URI

```
https://apmgw.dxi-na1.saas.broadcom.com/jarvis/v2/ingestion
http://apmservices-gateway.10.0.0.0.nip.io/jarvis/v2/ingestion
```

Sample Request Payload for Change Events

```
{
    "documents": [
                  {
```

```
                    "header": {
                                "tenant_id": "D43CFF21-FA66-4341-88BD-D7CA9A98BF21",
                                "product_id": "ao",
                                "doc_type_id": "itoa_events_change_custom",
                                "doc_type_version": "1",
                                "unique_id": "$['event_unique_id']"
                    },
                    "body": [
                    {
                        "severity": "Warning",
                        "summary": "Pod down",
                        "product": "tixchange-v1",
                        "change_type": "k8s",
                        "message": "Liveness probe failed: Get http://10.1.104.97:3000/: net/http:
 request canceled (Client.Timeout exceeded while awaiting headers)",
                        "host": "HCLHVF2.Broadcom.net",
                        "event_unique_id": "K8s-b77675rf6-ba92-4235-
b08f-7897d1b6d3e7-1636487041096",
                        "ci_unique_id": "100893",
                        "timestamp": "2021-11-09T19:44:01+0000",
                        "status": "NEW"
                    }
                    ]
                }
               ]
}
```


Table 2: Parameters for Change Events

Parameter	Mandatory/Optional	Type	Description
tenant_id	Mandatory	Header	Provides the tenant UUID (also called cohort_id) for DX Operational Intelligence to identify the tenant. **Format**: String
product_id	Mandatory	Header	Provides the fixed value "ao". **Format**: String
doc_type_id	Mandatory	Header	Provides the fixed value "ao_itoa_events_change_custom". **Format**: String
doc_type_version	Mandatory	Header	Provides the fixed value "1". **Format**: String
unique_id	Mandatory	Header	Provides the fixed value "$['event_unique_id']". **Format**: String
event_unique_id	Mandatory	Body	Provides the event unique ID. **Format**: String
product	Mandatory	Body	Provides the source product name on which the event is generated. **Example**: NewRelic **Format**: String
message	Mandatory	Body	Provides the event message.

Parameter	Mandatory/Optional	Type	Description
status	Mandatory	Body	Provides the status of the event. Valid values: NEW, UPDATED and CLOSED **Format**: String
severity	Mandatory	Body	Provides the severity of the alarm. Valid values: Critical, Major, Minor, and Information **Format**: String
timestamp	Mandatory	Body	Provides the time of the event in ISO 8601 date string format. Format: yyyy-MM-dd'T'HH:mm:ssZ **Format**: String
ci_unique_id	Optional	Body	Provides the unique ID for the entity on which event is generated.
summary	Optional	Body	Provides a short summary of the event. **Format**: String
host	Optional	Body	Provides the hostname of the device or entity on which event is generated. **Format**: String
change_type	Optional	Body	Provides the type of the event. **Example**: Application, Infrastructure **Format**: String
product_version	Optional	Body	Provides the version of the source product. **Format**: String
configuration_item_type	Optional	Body	Provides the entity type on which event is generated. **Format**: String
configuration_item	Optional	Body	Provides the entity name on which event is generated. **Format**: String
tags	Optional	Body	Provides the Ad hoc string values. **Format**: Array of Strings
custom_1, custom_2 custom_n	Optional	Body	Provides the custom information from the event of type 'String'. **Maximum Custom Fields**: 10 **Format**: String
custom_num_1 custom_num_2	Optional	Body	Provides the custom information from the event of type 'Number'. **Format**: Double

Metrics Ingestion

Metric data ingestion into the NASS metadata store

You can ingest the metric data into NASS metadata store using the Metrics Ingestion API. The metric ingestion process involves the following steps:

- Register Metric into Metric store
- Validate Registered Metric Metadata
- Metric Data Ingestion

Metric Registration in Metric Store

Before you ingest metric data, you must complete the metric registration in the Metric Store using the Metrics Registration Ingestion API. When you register metrics, the Metric Store generates and returns a metric identifier. The Metrics Ingestion API requires the metric Identifier to ingest metric data into Metric store.

Resource URI

You can select the URI based on the deployment and region of your tenant:

- DX SaaS - USA:
  `https://apmgw.dxi-na1.saas.broadcom.com/metadata/registerMetric`
- DX SaaS - EU:
  `https://apmgw.dxi-eu1.saas.broadcom.com/metadata/registerMetric`

```
http://<APM gateway Route>/metadata/registerMetric
```

Method

```
POST
```

Authorization

Authorization: Bearer {<APM Gateway Tenant token>}

For more information on generating the token, seeAuthentication and Authorization of APIs

HTTP Headers

Content-Type: application/json

Request Payload Syntax

```
{
  "metrics": [
    {
        "sourceName": "<Source Name>",
        "type": 1,
        "attributeName": "<Attribute Name>",
          "attributes": {
              "metric_unit": "<Metric Units>",
              "product": "<Product Name>",
              "product_version": "<Product Version>",
              "metric_name": "<Metric Name>",
              "configuration_item": "<Configuration Item>",
              "metric_type": "<Metric Type>",
              "host": "<Host>",
              "ci_unique_id": "ci Unique ID",
              "external_ids": "<External IDs>"
         } //Attributes map is optional, useful for queries

    }
  ]}
```

Response Syntax

```
{

    "metrics":
        {
          "id": "<Metric ID used by Ingestion API to ingest Metric Data>",
          "sourceName": "custom|Solarwinds|lvn-dctor-3cr01c01sw01.nw-am.bro|General",
          "type": 1,
          "attributeName": "Solarwinds_lvn-dctor-3cr01c01sw01.nw-am.bro|Ethernet1/46|
Orion.NPM.Interfaces:Outbps",
          "attributes": {
              "metric_unit": "<Metric Units>",
              "product": "<Product Name>",
              "product_version": "<Product Version>",
              "metric_name": "<Metric Name>",
              "configuration_item": "<Configuration Item>",
              "metric_type": "<Metric Type>",
              "host": "<Host>",
              "ci_unique_id": "ci Unique ID",
              "external_ids": "<External IDs>"
        }
    ],
    "sources": [
        {
            "sourceId": "<Source ID>",
            "name": "<Source Name>"
        }
    ]
}
}
```

Sample Request-Response

Sample URI

```
https://apmgw.dxi-na1.saas.broadcom.com/metadata/registerMetric
http://apmservices-gateway.10.0.0.0.nip.io/metadata/registerMetric
```

Sample Request Payload for Metric Registration

```
{
  "metrics": [
    {
        "sourceName": "custom|Solarwinds|lvn-dctor-3cr01c01sw01.nw-am.bro|General",
        "type": 1,
        "attributeName": "Solarwinds_lvn-dctor-3cr01c01sw01.nw-am.bro|Ethernet1/46|
Orion.NPM.Interfaces:Outbps",
          "attributes": {
              "metric_unit": "bits",
              "product": "Solarwinds",
              "product_version": "1.0.0",
              "metric_name": "Outbps",
              "configuration_item": "Ethernet1/46",
              "metric_type": "Orion.NPM.Interfaces",
              "host": "lvn-dctor-3cr01c01sw01.nw-am.bro",
```

```
                "ci_unique_id": "SolarwindsTest22|lvn-dctor-3cr01c01sw01.nw-am.bro",
                "external_ids": "CUSTOM:F4DA97EF-7C62-408B-9A87-F86CF715A104$$SolarwindsTest22$
$SolarwindsTest22|lvn-dctor-3cr01c01sw01.nw-am.bro"

            }

    }
  ]}
```


Sample Response for Metric Registration


```
{

    "metrics": [
        {
            "id": "roC-B-yI-1sw0w4",
           "sourceName": "custom|Solarwinds|lvn-dctor-3cr01c01sw01.nw-am.bro|General",
          "type": 1,
          "attributeName": "Solarwinds_lvn-dctor-3cr01c01sw01.nw-am.bro|Ethernet1/46|
Orion.NPM.Interfaces:Outbps",
          "attributes": {
                "metric_unit": "bits",
                "product": "Solarwinds",
                "product_version": "1.0.0",
                "metric_name": "Outbps",
                "configuration_item": "Ethernet1/46",
                "metric_type": "Orion.NPM.Interfaces",
                "host": "lvn-dctor-3cr01c01sw01.nw-am.bro",
                "ci_unique_id": "SolarwindsTest22|lvn-dctor-3cr01c01sw01.nw-am.bro",
                "external_ids": "CUSTOM:F4DA97EF-7C62-408B-9A87-F86CF715A104$$SolarwindsTest22$
$SolarwindsTest22|lvn-dctor-3cr01c01sw01.nw-am.bro"
        }
    ],
    "sources": [
        {
            "sourceId": "yI-76dsg7",
            "name": "custom|Solarwinds|lvn-dctor-3cr01c01sw01.nw-am.bro|General"
        }
    ]
}
```


Registered Metric Metadata Validation

After you register the Metric metadata, you can validate the registered metric data using the Metric validation Ingestion API.

Resource URI

Y You can select the URI based on the deployment and region of your tenant:

- DX SaaS - USA:
  `https://apmgw.dxi-na1.saas.broadcom.com/metadata/queryMetric`
- DX SaaS - EU:
  `https://apmgw.dxi-eu1.saas.broadcom.com/metadata/queryMetric`

```
http://<APM gateway Route>/metadata/queryMetric
```

Method

```
POST
```

Authorization

Authorization: Bearer {<APM Gateway Tenant token>}

For more information on generating the token, see Authentication and Authorization of APIs

HTTP Headers

Content-Type: application/json

Request Payload Syntax

```
{
 "specifier": {
 "op": "SPEC",
 "sourceNameSpecifier": {
 "op": "REGEX",
 "pattern": "custom\\|.*"
 },
 "attributeNameSpecifier": {
 "op": "ALL"
 }
 }
}
```

Response Syntax

```
{
    "metrics": [
        {
            "id": "<Metric ID>",

            "sourceName": "Source Name",
            "type": 1,
            "attributeName": "<Attribute Name>"
        },
        {
            "id": "<Metric ID>",
            "sourceName": "Source Name",
            "type": 2,
            "attributeName": "<Attribute Name>"
        }
    ],
    "folders": []
}
```

Sample Request-Response

Sample URI

```
https://apmgw.dxi-na1.saas.broadcom.com/metadata/queryMetric
http://apmservices-gateway.10.0.0.0.nip.io/metadata/queryMetric
```

Sample Request Payload for Registered Metric Validation

```
 "specifier": {
 "op": "SPEC",
 "sourceNameSpecifier": {
 "op": "REGEX",
 "pattern": "custom\\|Dynatrace\\|.*"
 },
 "attributeNameSpecifier": {
 "op": "ALL"
 }
 }
}
```

(OR) Sample Input including Product Name:

```
{
"specifier": {
 "op": "SPEC",
 "sourceNameSpecifier": {
 "op": "REGEX",
 "pattern": "custom\\|Dynatrace\\|.*"
 },
 "attributeNameSpecifier": {
 "op": "ALL"
 }
 }
}
```

Sample Response for Registered Metric Validation

```
{
    "metrics": [
        {
            "id": "roC-B-yI-1sw0w4",

            "sourceName": "custom|Dynatrace|HOST-6DE9560308323DF3|HOST",
            "type": 1,
            "attributeName": "HOST-6DE9560308323DF3||Disk:Disk Read Time"
        },
        {
            "id": "vU-C-D-H1tMBr",
            "sourceName": "custom|Dynatrace|Host",
            "type": 2,
            "attributeName": "HOST-6DE9560308323DF3||Disk:Disk Used"
        }
    ],
    "folders": []
```

```
}
```

Metric Ingestion

After registering the metric data into the Metric store, you can ingest the metric data from the external sources into the DX Operational Intelligence Metric Store using Metric Ingestion API.

> **NOTE**
> Ensure that the metric registration is complete before you ingest metric data. The metric ID that the Metric store generates is required for the metric data ingestion.

Resource URI

You can select the URI based on the deployment and region of your tenant:

- DX SaaS - USA:
  `https://apmgw.dxi-na1.saas.broadcom.com/nass/metricValue/store`
- DX SaaS - EU:
  `https://apmgw.dxi-eu1.saas.broadcom.com/nass/metricValue/store`

```
http://<APM gateway Route>/nass/metricValue/store
```

Method

```
POST
```

Authorization

Authorization: Bearer {<APM Gateway Tenant token>}

For more information on generating the token, seeAuthentication and Authorization of APIs

HTTP Headers

Content-Type: application/json

Request Payload Syntax

```
{

"values": [["Metric ID", <time(unixTimestamp/seconds)>, <min>, <max>, <value>, <count>, <interval>]]
}
```

Table 3: Parameters for Metric Ingestion

Parameter	Description
Metric ID	Provides the Unique ID created by Metric Store. For more information, see Metric Registration **Format**: String
interval	Defines the width of interval in number of seconds. The Interval is normalized during the ingestion to the next largest value from the following list: 15, 30, 60, 300, 900, 1800, 3600, 7200. If the value is greater that 7200, it is rounded off to 7200. **Format**: int
endTime	Defines end time in number of seconds from UNIX epoch. **Format**: int

Parameter	Description
min	Specifies the minimum value that was identified in interval. If the Metric is a string, then the minimum value is 'Null' as there is no minimum value for string metrics. **Format**: long or double or null
max	Maximum value that was identified in interval. Null is used for string value metrics (string metrics has no max value). **Format**: long or double or null
value	Specifies the actual value of the metric. **Format**: long or double or string
count	Specifies the count of sample values. The count is useful for calculating the weighted average. **Format**: long
extensionId	Defines the Identifier of the extension for a given metric. If the metric is not an extension, then the extensionId is null. **Format**: string or null

Sample Request

Sample URI

```
https://apmgw.dxi-na1.saas.broadcom.com/nass/metricValue/store
 http://apmservices-gateway.10.0.0.0.nip.io/nass/metricValue/store
```

Sample Request Payload

```
{

"values": [["roC-B-yI-1sw0w4", 1583418300, 48.0, 48.0, 48.0, 1, 300]]
}
```

Topology Ingestion

Using Topology Ingestion API, you can ingest the topology data into Topology Store(TAS). DX Operational Intelligence maps and transforms the third-party custom data to the expected TAS format before ingesting the data into the Topology Store as a "CUSTOM" layer.

Resource URI

YYou can select the URI based on the deployment and region of your tenant:

- DX SaaS - USA:
  `https://apmgw.dxi-na1.saas.broadcom.com/tas/graph/store`
- DX SaaS - EU:
  `https://apmgw.dxi-eu1.saas.broadcom.com/tas/graph/store`

```
http://<APM gateway Route>/tas/graph/store
```

Method

```
POST
```

Authorization

Authorization: Bearer {<APM Gateway Tenant token>}

For more information on generating the token, see Authentication and Authorization of APIs

HTTP Headers

Content-Type: application/json

Request Payload Syntax

```
{
   "graph": {
       "vertices": [{
               "externalId": <External ID>,
               "attributes": {
                   "name": "Name",
                   "type": "HOST",
                   "ci_unique_id": "<ci_unique_id>",
                   "applicationname": "<Application Name>",
                   "hostNames": "<Host Name>",
                   "product": "<Product Name>",
                   "entity_id": "<Entity ID>",
                   "ipAddresses": ["10.0.0.0"],
                   "agentversion": "{major=1, minor=185, revision=137, timestamp=20200212-183600,
sourceRevision=}",
                   "hostGroup": "{meId=<Host Group>, name=Windows}"
               },
               "startTime": <Start Time>,
               "endTime": <End Time>
           }, {...
              }
       "edges":
               "targetExternalId": "<Target External ID>",
               "sourceExternalId": "<Source External ID>",
               "startTime": <Start Time>,
               "endTime": <End Time>
           }, {...
           }
       ]
   }
}
```

Table 4: Mandatory Attributes for Topology Ingestion

Parameter	Description
External ID	Defines the unique identifier of the entity. The format of the unique identifier is: "CUSTOM:<Tenant ID>$$<product>$$<ci_unique_id>"
Name	Specifies the entity display name.
type	Specifies the type of the entity. Example: HOST, AGENT
ci_unique_id	Specifies the unique identifier of the entity. Example: FQDN for a HOST/Device.
product	Specifies the name of the source product from which the data is fetched for ingestion.
startTime and endTime	Specifies the start and end time of the entity life in micro seconds of Unix Epoch.

Table 5: Other Attributes for Topology Ingestion

Parameter	Description
ipAddresses	Specifies the list of IP Addresses of the host or device. Ensure that the first IP Address in the list is the primary IP Address.
macAddresses	Specifies the list of MAC Addresses of the host or device. Ensure that the first MAC address in the list is the primary MAC Address.
hostNames	Specifies the list of host names including the FQDN (Fully Qualified Domain Name) and Non-FQDN. Ensure that the first hostname in the list is a primary FQDN host name.

Response Syntax

```
{
    "code": 200,
}
```

NOTE
If the service is not available, the API returns 503 error code.

Sample Request-Response

Sample URI

```
https://apmgw.dxi-na1.saas.broadcom.com/tas/graph/store
http://apmservices-gateway.10.0.0.0.nip.io/tas/graph/store
```

Sample Request Payload

```
 {
    "graph": {
        "vertices": [{
                "externalId": "CUSTOM:D4684A21-8DEE-49AF-98B9-0B12DACB1B27$$Dynatrace$
$HOST-6DE9560308323DF3",
                "attributes": {
                    "name": "sp031362-WVM01",
                    "type": "HOST",
                    "ci_unique_id": "HOST-6DE9560308323DF3",
                    "applicationname": "My web application",
                    "hostNames": "sp031362-WVM01",
                    "product": "Dynatrace",
                    "entity_id": "HCLHVF2.Broadcom.net",
                    "ipAddresses": ["10.52.37.88"],
                    "agentversion": "{major=1, minor=185, revision=137, timestamp=20200212-183600,
 sourceRevision=}",
                    "hostGroup": "{meId=HOST_GROUP-3DE4C38BF9FCDA9F, name=Windows}"
                },
                "startTime": 1582610988591,
                "endTime": 1582783788591
            }, {
                "externalId": "CUSTOM:D4684A21-8DEE-49AF-98B9-0B12DACB1B27$$Dynatrace$
$SERVICE-09D9FA5003BB8F14",
                "attributes": {
                    "name": "Welcome to Tomcat",
                    "type": "SERVICE",
```

```
                    "ci_unique_id": "SERVICE-09D9FA5003BB8F14",
                    "applicationname": "My web application",
                    "hostname": "HCLHVF2.Broadcom.net",
                    "product": "Dynatrace",
                    "entity_id": "SERVICE-09D9FA5003BB8F14"
                },
                "startTime": 1582610988591,
                "endTime": 1582783788591
            }, {
                "externalId": "CUSTOM:D4684A21-8DEE-49AF-98B9-0B12DACB1B27$$Dynatrace$
$PROCESS_GROUP_INSTANCE-8D94C22B811F9175",
                "attributes": {
                    "name": "apache-tomcat-*",
                    "type": "PROCESS",
                    "ci_unique_id": "PROCESS_GROUP_INSTANCE-8D94C22B811F9175",
                    "applicationname": "My web application",
                    "hostname": "HCLHVF2.Broadcom.net",
                    "product": "Dynatrace",
                    "entity_id": "PROCESS_GROUP_INSTANCE-8D94C22B811F9175"
                },
                "startTime": 1582610988591,
                "endTime": 1582783788591
            }, {
                "externalId": "CUSTOM:D4684A21-8DEE-49AF-98B9-0B12DACB1B27$$Dynatrace$$APPLICATION-
EA7C4B59F27D43EB",
                "attributes": {
                    "name": "My web application",
                    "type": "APPLICATION",
                    "ci_unique_id": "APPLICATION-EA7C4B59F27D43EB",
                    "applicationname": "My web application",
                    "hostname": "HCLHVF2.Broadcom.net",
                    "product": "Dynatrace",
                    "entity_id": "APPLICATION-EA7C4B59F27D43EB",
                    "entity_name": "My web application"
                },
                "startTime": 1582610988591,
                "endTime": 1582783788591
            }
        ],
        "edges": [{
                "targetExternalId": "CUSTOM:D4684A21-8DEE-49AF-98B9-0B12DACB1B27$$Dynatrace$
$HOST-01B7BC15E8D87E91",
                "sourceExternalId": "CUSTOM:D4684A21-8DEE-49AF-98B9-0B12DACB1B27$$Dynatrace$
$PROCESS_GROUP_INSTANCE-8D94C22B811F9175",
                "startTime": 1582610988591,
                "endTime": 1582783788591
            }, {
                "targetExternalId": "CUSTOM:D4684A21-8DEE-49AF-98B9-0B12DACB1B27$$Dynatrace$
$PROCESS_GROUP_INSTANCE-8D94C22B811F9175",
                "sourceExternalId": "CUSTOM:D4684A21-8DEE-49AF-98B9-0B12DACB1B27$$Dynatrace$
$SERVICE-09D9FA5003BB8F14",
                "startTime": 1582610988591,
                "endTime": 1582783788591
```

```
            }, {
                "targetExternalId": "CUSTOM:D4684A21-8DEE-49AF-98B9-0B12DACB1B27$$Dynatrace$
$SERVICE-09D9FA5003BB8F14",
                "sourceExternalId": "CUSTOM:D4684A21-8DEE-49AF-98B9-0B12DACB1B27$$Dynatrace$
$APPLICATION-EA7C4B59F27D43EB",
                "startTime": 1582610988591,
                "endTime": 1582783788591
            }
        ]
    }
}
```

Sample Response

```
{
    "code": 200,
}
```

Topology Validation

You can validate topology ingestion using the validation API.

Resource URI

You can select the URI based on the deployment and region of your tenant:

- DX SaaS - USA:
  `https://apmgw.dxi-na1.saas.broadcom.com/tas/graph/query`
- DX SaaS - EU:
  `https://apmgw.dxi-eu1.saas.broadcom.com/tas/graph/query`

```
http://<APM gateway Route>/jarvis/v2/tas/graph/query
```

Method

```
POST
```

Authorization

Authorization: Bearer {<APM Gateway Tenant token>}

For more information on generating the token, see Authentication and Authorization of APIs

HTTP Headers

Content-Type: application/json

Request Payload Syntax

```
{
    "filter": {
        "op": "JOIN",
        "input": {
            "op": "AND",
            "input": [{
                    "op": "ATTRIBUTE",
                    "expressions": [{
```

```
                            "name": "product",
                            "values": [
                                "Dynatrace"
                            ]
                        }
                    ]
                }
            ]
        }
    },
    "universe": null,
    "version": null,
    "time": 0,
    "stitchingEnabled": true,
    "includeStatus": false,
    "projection": null,
    "offset": 0,
    "limit": 0
}
```

Response Syntax

```
{
    "code": 200,
}
```

Sample Request-Response

Sample URI

```
https://apmgw.dxi-na1.saas.broadcom.com/tas/graph/query
http://apmservices-gateway.10.0.0.0.nip.io/tas/graph/query
```

Sample Request Payload

```
{
    "filter": {
        "op": "JOIN",
        "input": {
            "op": "AND",
            "input": [{
                    "op": "ATTRIBUTE",
                    "expressions": [{
                            "name": "product",
                            "values": [
                                "Dynatrace"
                            ]
                        }
                    ]
                }
            ]
        }
    },
    "universe": null,
```

```
        "version": null,
        "time": 0,
        "stitchingEnabled": true,
        "includeStatus": false,
        "projection": null,
        "offset": 0,
        "limit": 0
    }
```

Sample Response

```
    {
        "code": 200,
    }
```

RESTMon

Third-party data sources are integrated using RESTMon.

DX Operational Intelligence consumes both structured and unstructured data such as topology, metrics, traces, alarms, and groups from heterogeneous sources to distill the signal from noise intelligently. RESTMon enables you to connect to a third-party data source and ingest that data into DX Operational Intelligence using out-of-the-box schemas.

This section provides the following information:

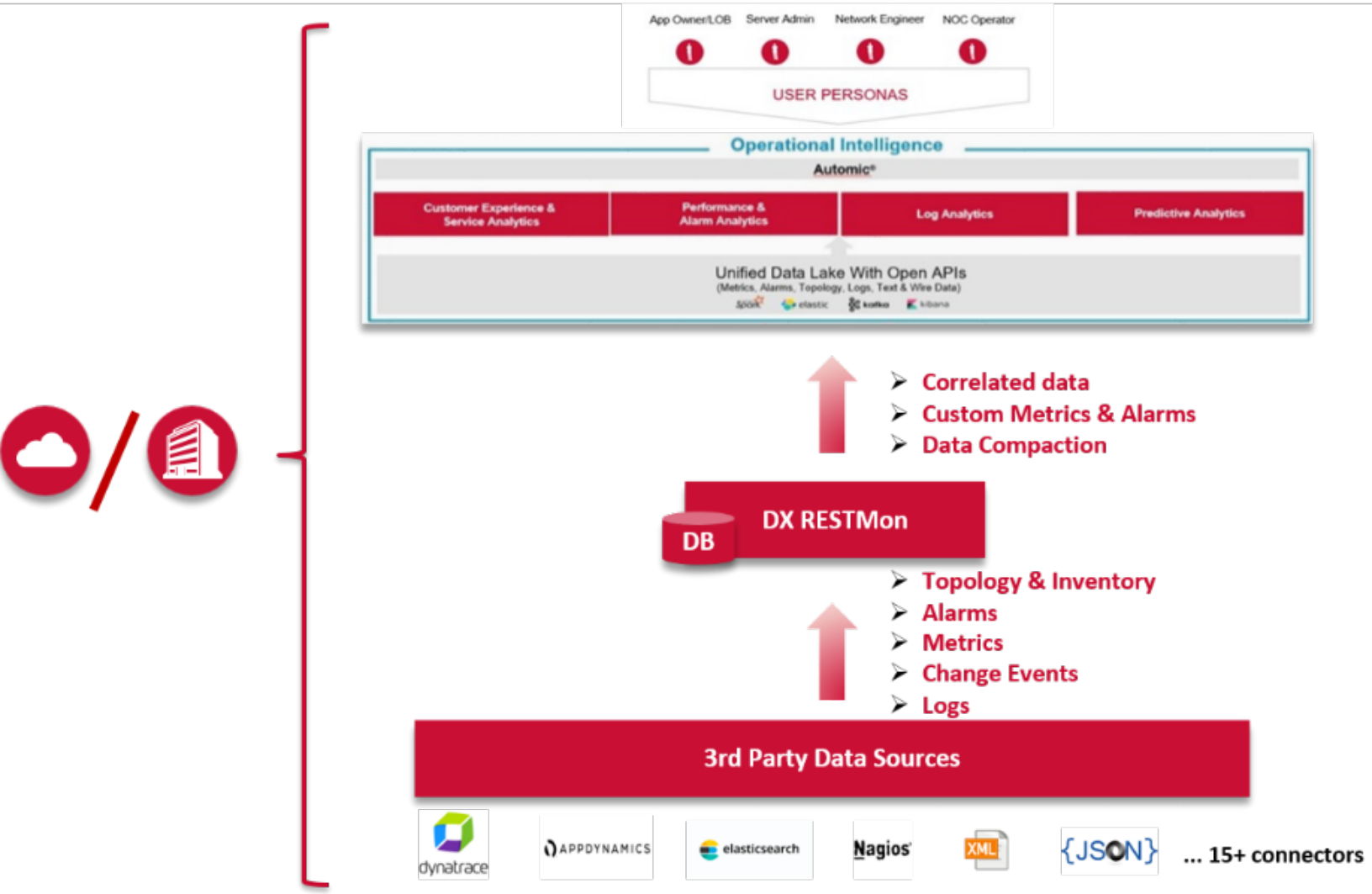
Key Capabilities

Using RESTMon, you can perform the following tasks:

- Ingest monitoring data (metrics, alarms, topology, and events) from the third-party systems:
- Maintain relationships between Alarms & Topology.
- Create relationships between Alarms & Metrics for custom integrations
- Secure
  - Data transfer over HTTPS
  - Proxy support
- Available as a container with built-in liveness and readiness
- Custom metrics and alarms
- Support for JSON and XML formats

Architecture Overview

The following image illustrates the architecture:



- **Ingest Data:** Ingest alarms, change events, metrics, and topology into DX Operational Intelligence.
- **Correlate Data:**
  - **Alarm and Metric Correlation**: The hostname, Configuration Item (CI), CI Unique ID, and the Metric Name give the metric a unique ID that is passed to the alarm.
  - **Alarm and Topology Correlation:** If the hostname is not available in the alarm, then get the hostname from the topology based on the CI unique ID or Entity ID.
- **Data Compaction:** Apply Compaction Rules or Association Rules to correlate the third-party data with the monitoring data such as UIM, Spectrum, and RESTMon. The schema rules are provided by RESTMon, but the entities are maintained by the correlation manager of DX Operational Intelligence. For example, the Host Name or IP address from Spectrum and the third-party data are correlated to have one entity in DX Operational Intelligence.

Video: Introduction to RESTMon

The following video walks you through RESTMon.

Configure and Deploy

RESTMon deployment is supported on a virtual machine or an orchestrated platform such as Kubernetes. Choose the correct flavor depending on where you want to deploy.

This section provides the following information:

Compatibility Matrix

DX Operational Intelligence	RESTMon
DX OI SaaS	RESTMon 2.2.1, 2.2, and 2.1.6
DX OI 23.1.0	RESTMon 2.2.1, 2.2, and 2.1.6
DX OI 22.1.0	RESTMon 2.2 and 2.1.6
DX OI 21.3.1	RESTMon 2.2 and 2.1.6 For RESTMon 2.2, additional components will need to be deployed on DX OI 21.3.1. These components are part of the RESTMon 2.2 package.
DX OI 20.2.1	RESTMon 2.1.6

Prerequisites

- Software Requirements
- Hardware Requirements
- Download RESTMon Package
- Download RESTMon Package
- Package Contents

Software Requirements

RESTMon	Supported Version
Docker Container	Linux (CentOS 7), Kubernetes v1.14
Java	Java SE 11

Hardware Requirements

	Minimum Requirement
CPU	6/8 cores
Memory	4 GB
Storage	50 GB for Database and 10 GB for Logs

NOTE

- Depending on the maximum memory configured, you may have to change the heap size.
- You may have to increase the storage if the logs are configured for longer retention.

Download RESTMon Package

You can download the RESTMon 2.2 package from the **Settings** page.

Follow these steps:

1. Login to DX Operational Intelligence as a Tenant Administrator.
2. Click **Settings** in the left navigation pane.
3. Click on the **Setup Data Sources** tile to navigate to the **Downloads** page.
4. Click **DX Restmon** to start the download. The size of the package is about 1.x GB.
   The package is downloaded. For more information, see the Package Contents section.

Download RESTMon Package

You can download the RESTMon package for the On-Premise deployments from Broadcom Support.

Follow these steps:

1. Login to Broadcom Support.
2. Select **Enterprise Software** from the list that is in the top-right corner.
3. Click **My Downloads** in the left navigation pane.
4. Enter the product name as **DIGITAL OPERATIONAL INTELLIGENCE** and search for the product.
   Two tabs appear under the product name: **Products** and **Solutions**. Some products may not have solutions.
5. Expand the list and click **23.1** under the **Release** column to display the files.
6. Download the **RESTMon** package using one of the following methods:
   – **HTTPS Download:** Click the **HTTPS Download** icon to start the package download from https://downloads.broadcom.com immediately. Check your browser for the download progress.
   – **Secure FTP Download:** Click the **Secure FTP Download** icon and then click **Details** to view the instructions. This option is the fastest and most efficient download method.

     NOTE
     If you are unable to download the image, contact the ***Broadcom Support***.

   The size of the package is about 1.x GB. The package is downloaded. For more information, see the Package Contents section.

Package Contents

The downloaded **RESTMon -*.zip** file includes the following folders:

- **docker:** The dx-restmon-2.2.tar file in this folder contains the containerized docker image for deploying RESTMon as a container.
- **vm:** The DX-RESTmon-2.2.zip file in this folder contains the artifacts to deploy and run RESTMon on Windows or Linux machines.
- **helm:** The dx-restmon-2.2.zip file in this folder contains the deployment templates and the **values.yaml** required to deploy DX RESTMon on the orchestration platforms such as Kubernetes, OpenShift, or any cloud platforms such as Google Kubernetes Engine (GKE).

Container Version

This section describes how to configure and deploy the Container version of RESTMon:

- RESTMon 2.2
- RESTMon 2.1.6

RESTMon 2.2 or 2.2.1

This section provides the following information to configure and deploy RESTMon 2.2 or 2.2.1 on the cluster. You can deploy RESTMon 2.2.1 on 2.2 and 2.1.6.

Prerequisites for the Deployment

Before you deploy the RESTMon 2.2, ensure that the prerequisites are met:

- The Docker Engine is installed on the system where you want to deploy RESTMon. For more information, see the Docker Engine Installation documentation.
- Helm 3 is installed.
- RESTMon package is downloaded from the **Settings** page for SaaS deployment. For more information, see the Download RESTMon Package section.
- RESTMon package is downloaded from **Broadcom Support** for On-Premise deployment. For more information, see the Download RESTMon Package section.
- The RESTMon docker image is loaded into the docker host system. Run the following command to load the image to the Docker host system from the tar archive:

  For RESTMon 2.2.1

  ```
  docker load < dx-restmon:2.2.1.tar.gz
  ```

  For RESTMon 2.2

  ```
  docker load < dx-restmon:2.2.tar.gz
  ```

  NOTE
  Ensure that the image is loaded on all the worker nodes. Alternatively, you can push the image to a private docker repository and can pass the image as an argument.
- The Tenant Cohort ID and Tenant Token are available.

  NOTE
  Cohort ID is available on the **Settings > Connector Parameters** page in DX Operational Intelligence. You can generate the tenant token on the **Tokens** page in DX SaaS.
- The Tenant Cohort ID and APM Agent Token are available.

  NOTE
  Cohort ID and Agent Token are available on the **Settings > Connector Parameters** page in DX Operational Intelligence.
- For an On-Premise deployment, the ODC Lifecycle Management service is deployed. For more information, see the Deploy ODC Lifecycle Management Service section on this page.

Configure values.yaml Properties

Before you start the RESTMon deployment, you must configure the properties in the **values.yaml** file. The *values.yaml* file is part of the helm chart archive **dx-restmon-*.tgz** in the helm folder. This file contains the properties that are required for the RESTMon deployment.

The following table lists the minimum mandatory properties that are required to start RESTMon. Some of these properties have pre-configured values. You can reconfigure if necessary.

NOTE

- For the complete list of the properties, see the Configuration Properties Reference section on this page.
- **Use Default Value or Reconfigure *:** Indicates that these properties have default values. You may reconfigure them if necessary.
- **Must Configure **:** Indicates that these properties are mandatory and must be configured with the right values.

Property	Description	Use Default Value or Reconfigure *	Must Configure **
restmon.instanceName	Defines the name of the instance that is displayed in the dashboard. **Default:** restmon-master	Yes	
restmon.restmonport	Defines the RESTMon pod application port. You can use the default values or can define them. If this port is not defined, • RESTMon uses 8080 if **restmon.secured.enabled=false** • RESTMon uses 8443 if **restmon.secured.enabled=true**	Yes	
restmon.serviceport	Defines the RESTMon service port. You can use the default values or can define them. If this port is not defined, • RESTMon uses 9090 if **restmon.secured.enabled=false** • RESTMon uses 8443 if **restmon.secured.enabled=true**	Yes	
restmon.secured.enabled	Enables HTTPS for the RESTMon deployment when set to **true**. **Default:** false RESTMon also creates a secret object using the following naming convention when the attribute is set to **true**: `restmon-keystore-<restmon instance name>` If **true**, then configure the following properties: • storeType • storeName • storePassword • storeAlias For more information, see the **Configuration Properties** section.	Yes	
restmon.properties.replaceOIAttributes	If the DX Operational Intelligence environment variables are updated during run time using the APIs, this property determines if the already configured OI settings in the **restmon.json** file should be replaced with the values from the **values.yaml** file on upgrade using Helm Charts. **Supported Values:** true, false **Default:** true	Yes	

Property	Description	Use Default Value or Reconfigure *	Must Configure **
restmon.properties.oiIngestionAPIHost	Specifies the APM Gateway ingestion endpoint that RESTMon uses for topology or NASS metrics data ingestion when **restmon.properties.replaceOIAttribute** is set to **true**.	Yes	
restmon.properties.oiIngestionAPIPort	Defines the port on which the APM Gateway ingestion endpoint is exposed. Default:**443**	Yes	
restmon.properties.oiIngestionAPIProtocol	Defines the protocol using which the APM Gateway ingestion endpoint is accessible. **Supported Values:** http or https **Default:** https	Yes	
restmon.properties.oiIngestionTenantToken	Defines the tenant token that the user must use for authenticating the APM Gateway ingestion endpoint.		Yes
restmon.properties.cohortID	Defines the Cohort ID of the tenant ingesting data into DX Operational Intelligence		Yes
restmon.properties.username	Defines the username for the RESTMon API access when basic auth is enabled.		Yes
restmon.properties.password	Defines the password for the RESTMon API access when basic auth is enabled.		Yes
restmon.properties.authType	Defines the authentication type. **Supported Values:** basic, bearer **Default:** basic	Yes	
image.repository	Specifies the folder path of the RESTMon package. For example: `<repository_path>/dx-restmon` **Default:** dx-restmon		Yes
image.pullPolicy	Defines the pull policy for the image. **Supported Values:** IfNotPresent, Always, or Never **Default:** IfNotPresent		Yes
imagePullSecrets	Defines the docker repository secret to pull the images from the private repository. **Default:** [ ] If Yes, configure the following properties: **nameOverride**, and **fullnameOverride**. For more information, see the **Configuration Properties** section.		Yes
volume.size	Defines the storage size for RESTMon metadata and logs.	Yes	
volume.className	Defines the class name. If the storage class is being used, update with the appropriate storage class name.	Yes	
volume.subPath (only for RESTMon 2.2.1)	Defines the subpath for the volume. **Value:** data	Yes	

Property	Description	Use Default Value or Reconfigure *	Must Configure **
volume.existingClaim (Only for RESTMon 2.2.1)	**Default:** Commented out. To use an existing PV/PVC (without creating one as part of the helm installation), you can uncomment this property and can map the existing PVC name to this property.	Yes	
serviceAccount	Service accounts are used to provide pods with an identity.	Yes	
ingress	Deploys ingress when the attribute is set to **true**.	Yes	
hosts	Defines a list of host rules that are used to configure the Ingress.		Yes
resources		Yes	

Start or Deploy RESTMon

After configuring the properties in the values.yaml file, you can deploy RESTMon.

Follow these steps:

1. Execute the corresponding command to start or create the RESTMon container:
   - **Image is available on all the cluster nodes:**
     ```
     helm install <name> restmon-2.2.1.tgz -f values.yaml --namespace <namespace>
     helm install <name> restmon-2.2.tgz -f values.yaml --namespace <namespace>
     ```
   - **Image is pushed to the private docker repository:**
     ```
     helm install <name> restmon-2.2.1.tgz -f values.yaml --namespace <namespace>
     helm install <name> restmon-2.2.tgz -f values.yaml --namespace <namespace>
     ```
   - **Image is pushed to private docker repository with authentication:**
     ```
     helm install <name> restmon-2.2.1.tgz --namespace <namespace>
     helm install <name> restmon-2.2.tgz --namespace <namespace>
     ```
2. Run the following command to check the deployment status:
   ```
   helm ls -n <namespace>
   ```
3. Add the profile to the **restmon.json** file using the following POST Profile REST API call:
   ```
   For HTTP:
   http://localhost:8080/restmon/api/v1/profiles
   For HTTPS:
   https://localhost:8443/restmon/api/v1/profiles
   ```
   Provide the profile as the input body. You can take reference of the sample profile from the **profile** directory. For more information, see the Add Profile illustration.
4. View the logs at the given NFS PV path or from the cloud console if deployed in the cloud cluster.
5. Perform all the other operations using the REST APIs accessible at:
   ```
   For HTTP: http://localhost:8080/restmon/api/swagger-ui/
   For HTTPS: http://localhost:8443/restmon/api/swagger-ui/
   ```

   NOTE
   If the DX Operational Intelligence routes are SSL enabled with a self-signed certificate and the RESTMon connection to the DX Operational Intelligence endpoint fails with the following exception,
   ```
   javax.net.ssl.SSLHandshakeException: PKIX path building failed:
    sun.security.provider.certpath.SunCertPathBuilderException
   ```

Download the SSL certificate of the DX Operational Intelligence and import the file into the Java Keystore file.

Configuration Properties Reference

The properties in the **values.yaml** file are grouped under various sections:

- restmon
- secured
- readiness
- liveness
- properties
- image
- imagePullSecrets
- volume
- serviceAccount
- podAnnotations
- podSecurityContext
- containerSecurityContext
- service
- ingress
- resources
- nodeSelector

restmon

You can specify the RESTMon connection details in the **restmon** section.

Attribute	Default	Description
instanceName		Defines the name of the instance that is displayed in the DX Operational Intelligence dashboard
servicePort (Optional)		Defines the RESTMon service port. If this port is not defined, • RESTMon uses 9090 if **restmon.secured.enabled=false** • RESTMon uses 8443 if **restmon.secured.enabled=true**
restmonPort (Optional)		Defines the RESTMon pod application port. If this port is not defined, • RESTMon uses 8080 if **restmon.secured.enabled=false** • RESTMon uses 8443 if **restmon.secured.enabled=true**
profileCheck	10 minutes	Defines the time in minutes for checking the `Profile Liveness`.

secured

You can enable or disable HTTPS in the **secured** section.

Attribute	Default	Description
enabled	false	Enables HTTPS for the RESTMon deployment when the attribute is set to **true**. RESTMon also creates a secret object using the following naming convention when the attribute is set to **true**: `restmon-keystore-<restmon instance name>`
storeType	PKCS12	Defines the keystore type. To create PKCS12 from the private key and certificate: `openssl pkcs12 -inkey key.pem -in certificate.pem -export -out restmon-certificate.p12 -name restmon-certificate-alias`

storeName		Defines the keystore. To get the keystore, run the following command:
		`cat <PKCS filestore> |base64| tr -d '\n'`
		
		`For example,`
		`cat restmon-certificate.p12 |base64| tr -d '\n'`

storePassword		Defines the keystore password.
storeAlias		Defines the alias for the keystore.

readiness

You can define the readiness properties in the **readiness** section. The kubelet uses the readiness probe to know when a container is ready to start accepting traffic.

Attribute	Default	Description
restmon.readiness.initialDelay	60 Seconds	Defines the time to wait before performing the first probe.
restmon.readiness.period	30 Seconds	Schedules the probes at the specified time intervals.
restmon.readiness.timeout	30 Seconds	Specifies the time-out period for the probe.
restmon.readiness.successThreshold	1	Defines the minimum number of consecutive successes for the probe to be considered as successful after a failure.
restmon.readiness.failureThreshold	5	Defines the minimum number of consecutive failures of the probe for RESTMon to consider the probe as a failure.

liveness

You can define the liveness properties in this section. The kubelet uses the liveness probe to know when to restart a container.

Attribute	Default	Description
restmon.liveness.initialDelay	60	Defines the time to wait before performing the first probe.
restmon.liveness.period	30 Seconds	Defines the frequency at which the liveness probe is invoked.
restmon.liveness.timeout	30 Seconds	Specifies the time-out period for the probe.

Attribute	Default	Description
restmon.liveness.successThreshold	1	Defines the minimum number of consecutive successes for the probe to be considered as successful after a failure.
restmon.liveness.failureThreshold	5	Defines the minimum number of consecutive failures of the probe for RESTMon to consider the probe as a failure.

properties

You can define the environment properties that are pushed into the config map in the **properties** section.

Attributes	Default	Description
restmon.properties.urlResponseLogger	off	Captures the request responses in the `restmon_schema_url_response.log` file. Supported values: off or debug
restmon.properties.logLevel	info	Defines the level of log information that must be captured in the RESTMon log file. **Supported Values:** info or debug
restmon.properties.replaceOIAttributes	true	If the DX Operational Intelligence environment variables are updated during run time using the APIs, this property determines if the already configured OI settings in the **restmon.json** file should be replaced with the values from the **values.yaml** file on upgrade using Helm Charts. **Supported Values:** true, false
restmon.properties.oiIngestionAPIHost	apmgw.dxi-na1.saas.broadcom.com	Specifies the APM Gateway ingestion endpoint that RESTMon uses for topology or NASS metrics data ingestion when **restmon.properties.replaceOIAttribute** is set to 'true'.
restmon.properties.oiIngestionAPIPort	443	Defines the port on which the APM Gateway ingestion endpoint is exposed.
restmon.properties.oiIngestionAPIProtocol	https	Defines the protocol using which the APM Gateway ingestion endpoint is accessible. **Supported Values:** http or https
Token Details:		
restmon.properties.oiIngestionTenantToke		Defines the tenant token that the user must use for authenticating the APM Gateway ingestion endpoint.
Cohort ID:		
restmon.properties.cohortID		Defines the Cohort ID of the tenant ingesting data into DX Operational Intelligence
restmon.properties.maxDocs	1000	Defines the maximum number of documents that RESTMon can ingest.
restmon.properties.minMemoryLimit	512M	Defines the minimum memory limit for the RESTMon process.

Attributes	Default	Description
restmon.properties.maxMemoryLimit	2G	Defines the maximum memory limit for the RESTMon process.
Supportability Agent Details:		
restmon.properties.supportabilityAgentName	\"SuperDomain| newoiservices | opendataconnector\"	Defines the agent name against which the metrics are ingested for the RESTMon connector. We recommend not to change the default agent name that is defined.
restmon.properties.username	admin	Defines the username for the RESTMon API access when basic auth is enabled.
restmon.properties.password	password	Defines the password for the RESTMon API access when basic auth is enabled.
restmon.properties.profileQueueSizeMax	20000	Defines the maximum queue size.
restmon.properties.profileQueueSizeMin	15000	Defines the minimum queue size.
restmon.properties.profilesMax	5	Defines the maximum profiles to execute.
restmon.properties.healthCheckInterval	15	Defines the health check interval for the local in-memory database. 15
restmon.properties.healthLogLevel	info	Defines the database Log level. Supported Values: info, debug, or off
restmon.properties.retryAttemptsInterval	10 Seconds	Defines the retry attempts interval.
restmon.properties.retryAttemptsMax	10	Defines the maximum number of retry attempts.
restmon.properties.logsFileSizeMax	50 MB	Defines the maximum size of the rolling file for the RESTMon log.
restmon.properties.logsFileHistoryMax	10	Defines the maximum file history for the log files to maintain. Recommended value: 10
restmon.properties.showSwagger	true	Enables the swagger UI for RESTMon ingestion when the property is set as **true**.
restmon.properties.reloadSchemas	false	Enables you to reload the out-of-the-box schemas when you set the flag to **true.**
restmon.properties.publish_max_retries	3	Defines the maximum number of retries that are allowed for the alarms ingestion to OI if ingestion failed (OI service failure or timeout).
restmon.properties.publish_backoff_initial_interval_sec	15	Defines the initial delay added for alarm ingestion retry.
restmon.properties.publish_backoff_max_interval_sec	120	Defines the maximum wait between retries.
restmon.properties.publish_backoff_multiplier	2	Defines the multiplier that is used to generate the next backoff interval from the last. With the delay, maxDelay, and multiplier defined, the backoff is exponentially grown up to the maximum value for each retry attempt until a max retry time is reached.
restmon.properties.authType	basic	Defines the authentication type. **Supported Values:** basic, bearer

image

You can define the path or the image name that must be downloaded in the **image** section.

Attribute	Default	Description
image.repository	dx-restmon	Specifies the folder path of the RESTMon package. For example: `<repository_path>/dx-restmon`
image.pullPolicy	IfNotPresent	Defines the pull policy for the image. **Supported Values:** IfNotPresent, Always or Never
image.tag		Specifies the RESTMon Image release tag. Overrides the image tag whose default is the chart appVersion. For example: `2.2`

imagePullSecrets

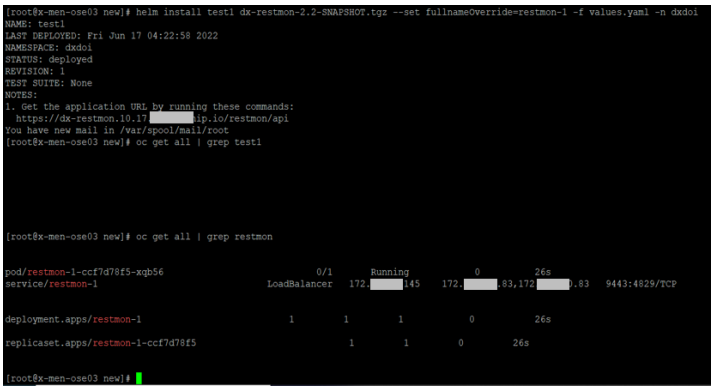
You can configure the docker repository secret in this section to pull the images from the private docker repository.

Attribute	Default	Description

imagePullSecrets	[ ]	Defines the docker repository secret to pull the images from the private repository.
		To pull the images from the private docker repository, you can configure the required secret in the following way:
		1. Remove the square brackets for imagePullSecrets.
		2. Uncomment the line defining the name for the docker secret as shown:
		`imagePullSecrets:`
		`# docker repository secret if pulling the images from private repository`
		`  - name: docker-secret`
		More Information! To create docker-secret for a private repository, pass the following parameters are arguments:
		`kubectl create secret docker-registry docker-secret --docker-server=<docker-repo> --docker-username=<username> --docker-password=<password> --docker-email=<email address> -n <namespace>`
		
		`For example, kubectl create secret docker-registry docker-secret --docker-server=esd-oi-docker-dev-local.artifactory-lvn.broadcom.net --docker-username=user --docker-password=password --docker-email=abcd@gmail.com -n mynamespace`

nameOverride	odc	Defines the suffix that is used for the names if there is a name override for the helm release name.

Attribute	Default	Description

fullnameOverride		Defines the default naming used for deployment and is derived from the helm install definition. For example, releasename (passed as the helm argument).
		If fullnameOverride is set, the deployment objects are named using the configured value.
		For example,
		`helm install test1 dx-restmon-2.2-SNAPSHOT.tgz --set fullnameOverride=restmon-1 -f values.yaml -n dxdoi`
		 
		If fullnameOverride is not set, the deployment objects are named using the helm release name that is passed in the install command concatenated with the value that is configured for nameOverride.
		If multiple instances of RESTMon are deployed into the same namespace, make sure that a unique value is configured for each chart along with a unique value for the instance name.
		`For example, helm install test2 dx-restmon-2.2-SNAPSHOT.tgz --set fullnameOverride=restmon-2 --set restmon.instanceName=restmon-master-2 --set nameOverride=odc-2 -f values.yaml -n dxdoi`
		For the fields without any mapped values (optional), the values are configured with either '[ ]' or { } depending on the type of object it refers to. You must remove these empty values [ ] or { } if configuring with any values for them.

volume

You can define the persistent volume requirements in the **volume** section.

Attribute	Default	Description
volume.size	1Gi	Defines the storage size for RESTMon metadata and logs.
volume.className	ssd	Defines the class name. If the storage class is being used, update with the appropriate storage class name. • ssd • nfs
volume.subpath (Only for RESTMon 2.2.1)	data	Defines the subpath.

Attribute	Default	Description
volume.existingClaim (Only for RESTMon 2.2.1)	Commented out	If you want to use an existing PV/PVC (without creating one as part of the helm installation), you can uncomment this existingClaim property and can map the existing PVC name to this property.

The following image illustrates how className is configured.

```
# Persistence details
volume:
  size: 1Gi
  className: ssd
  # If NFS is configured it will be used for the persi
  # nfs:
    # server: "10.1[redacted].97"
    # path: "/var/nfs/dxi/restmon"
```

To use NFS, change the className to nfs, remove the flower brackets, uncomment the nfs section, and update the NFS server and path details as shown.

```
# Persistence details
volume:
  size: 1Gi
  className:
  # If NFS is configured it will be used for the persistence
  nfs:
    server: "10.[redacted].78"
    path: "/var/nfs/helmcheck"
```

Attribute	Default	Description
volume.nfs.server		Defines the NFS server details.
volume.nfs.path		Defines the Persistent Volume path to use the NFS storage. Create a folder name **data** inside the path that you configured. For example, **/var/nfs/helmcheck/data**. Then provide the ownership 1010 to this folder. `chown -R 1010:1010 /var/nfs/helmcheck` The ownership of this folder must also honor the **podSecurityContext** section in the **values.yaml** file.

serviceAccount

Service accounts are used to provide pods with an identity. This identity is used to authenticate pods to the Kubernetes and OpenShift API servers and enable the processes or applications in the container to access the cluster.

> **NOTE**
> This configuration is optional, and if none is configured, the default service account in the same namespace is automatically assigned.

Attribute	Default	Description
serviceAccount.create	false	Defines if the service account has to be created.
serviceAccount.satoken		Defines the service account token.
serviceAccount.annotations		Defines the annotations to add to the service account.

You can configure the service account in the following ways:

- **Use an Existing Service Account:** To use an existing service account for the pod deployment, update the appropriate service account name. If no name is provided, the **default** service account is used.

```
serviceAccount:
  create: false
  #existing service account to be used
  name: dxi-doi
  satoken:
  # Annotations to add to the service account
  annotations: {}
    # kubernetes.io/service-account.name: "restmon-service-account"
```

- **Create a Service Account:** To create a service account using helm and to use that account for security:
  a. Update the **serviceAccount.create** flag to **true**.
  b. Update the **serviceAccount.name** field to refer to the new service account name to be created. If the name is empty, the naming that is used for deployment will be used (For more information, see the **fullnameOverride** and **nameOverride** attributes).
  c. Update **serviceAccount.satoken** to refer to the available service account token. If left blank, the Kubernetes engine automatically creates and references the token with the service account.
  d. Update the **serviceAccount.annotations** as required. It defaults to { }, without any annotations.

podAnnotations

You can configure the additional annotations to be added to the pods of this component and to your deployments that are used to create the pods as shown:

```
# Any pod specific annotation
podAnnotations:
  app-version: "0.1"
  annotation1: annotation-value-1
  annotation1: annotation-value-2
```

podSecurityContext

You can configure the security context information for the pods. SecurityContext holds the pod-level security attributes and common container settings.

Attribute	Default	Description
podSecurityContext.runAsUser	1010	Defines the UID to run the entry point of the container process.
podSecurityContext.runAsGroup	1010	Defines the GID to run the entry point of the container process.

Attribute	Default	Description
podSecurityContext.fsGroup	2000	Specifies the special supplemental group that applies to all containers in a pod. Some volume types allow the Kubelet to change the ownership of that volume to be owned by the pod: • The owning GID is the FSGroup. • The setgid bit is set (new files that are created in the volume are owned by FSGroup) • The permission bits are OR'd with rw-rw---- If unset, the Kubelet will not modify the ownership and permissions of any volume. Note: This field cannot be set when **spec.os.name** is **windows**.
podSecurityContext.fsGroupChangePolicy	Always	(Only for Kubernetes environments. Delete if OpenShift.) Defines the behavior of changing ownership and permissions of the volume before being exposed inside the pod.

containerSecurityContext

If you want to optimize the container security context further, you can configure the security settings for a container in the `containerSecurityContext` section. These settings override the pod level settings when there is overlap. The Container settings do not affect the pod's volumes.

Attribute	Default	Description
containerSecurityContext.allowPrivilegeEscalation	true	Controls whether a process can gain more privileges than its parent process. This bool directly controls whether the no_new_privs flag gets set on the container process. **Allowed Values:** true, false
containerSecurityContext.privileged	true	Container running as privileged or unprivileged. **Allowed Values:** true, false
containerSecurityContext.runAsGroup	1010	Defines which primary group ID the containers are run with.
containerSecurityContext.runAsUser	1010	Defines which user ID the containers are run with.

service

You can configure the Service resource providing a stable endpoint that can be used to address pods created by the Deployment controller.

Attribute	Default	Description
service.type	LoadBalancer	Defines the service type. Supported Values: ExternalName, ClusterIP, NodePort, LoadBalancer
service.annotations		Defines the unstructured key-value map that is stored in a resource that the external tools may set to store and retrieve the arbitrary metadata.

ingress

You can configure the ingress router to manage the external access to the services in a cluster in the ingress section.

Attribute	Default	Description
ingress.enabled	false	Deploys ingress when the attribute is set to 'true'. Using the Helm Charts, you cannot create the HTTPS ingress with the HTTPS backend (RESTMon on HTTPS) in OpenShift environments. If this deployment mode is required, disable the ingress creation from the helm and create the route manually for the RESTMon service created.
ingress.className		Specifies the IngressClass cluster resource. The associated IngressClass defines the controller that implements the resource.

ingress.annotations:		Defines the Annotations that RESTMon must apply on ingress. Example: haproxy.org/ingress.class: haproxy Annotations must be followed while creating the Ingresses. Refer to the ingress section in the yaml file for haproxy, nginx, and gke. For other cluster environments with different Ingress Controllers, update the annotations accordingly.
		Note: The following example illustrates how to create an Ingress when NGINX Controller is used in Kubernetes:
		`ingress:`
		`enabled: true`
		`className: ""`
		`# Add annotations for your ingress Implementation specific.`
		`annotations:`
		`kubernetes.io/ingress.class: nginx`
		`nginx.ingress.kubernetes.io/backend-protocol: HTTPS`
		`nginx.ingress.kubernetes.io/app-root: /restmon/api`
		`kubernetes.io/tls-acme: "true"`
		`kubernetes.io/ingress.allow-http: "true"`

hosts:		
ingress.hosts.host		Defines a list of host rules that are used to configure the Ingress. To configure Ingress, remove the square brackets ([]) and uncomment the lines for the host configuration. The **hosts.host** should be as per the routing rules that are defined in the environment. For example, dx-restmon-10.17.182.75.nip.io.
ingress.hosts.paths.path		Defines the list of paths.
ingress.hosts.paths.pathType		Defines the path type.
tls:		

Attribute	Default	Description

tls is supported from the client to the cluster load balancer. tls defaults to [ ] and is treated as unsecured. To configure the tls termination, uncomment the section and remove the default value [ ].
`tls:`
`- secretName: "restmon.helm"`
`hosts:`
`- "restmon.local"`		

ingress.tls.secretName		Defines the secretName parameter that is used for the TLS termination.
ingress.tls.hosts		Defines the hostname. Hosts in the tls section must match the host in the rules/hosts section explicitly.

NOTE
More Information: The secret is managed separately (outside helm) and referencing this secret in an Ingress tells the Ingress controller to secure the channel from the client to the load balancer using TLS. Ensure that the TLS secret that you created came from a certificate that contains a Common Name (CN), also known as a Fully Qualified Domain Name (FQDN) <hostname> (For example, dx-restmon.10.17.182.75.nip.io) configured.

You can generate a self-signed certificate and private key using the following command:

```
$ openssl req -x509 -nodes -days 365 -newkey rsa:4096 -keyout ${KEY_FILE} -out ${CERT_FILE} -subj "/
CN=${HOST}/O=${HOST}"

For example, Example: openssl req -x509 -nodes -days 365 -newkey rsa:4096 -keyout key-new.pem -out
 certificate-new.pem } -subj "/CN=dx-restmon.10.17.182.75.nip.io/O= dx-restmon.10.17.182.75.nip.io"
```

Then create the secret in the cluster:

```
kubectl create secret tls ${CERT_NAME} --key ${KEY_FILE} --cert ${CERT_FILE}

For example, Exmaple: kubectl create secret tls dx-restmon-https-new.helm --key=key-new.pem --
cert=certificate-new.pem
```

resources

You can configure the resource requirements or limits in this section.

Attribute	Default	Description
resources.limits.cpu	4000m	Defines the CPU limit.
resources.limits.memory	4Gi	Defines the memory limit.
requests:		
resources.requests.cpu	4000m	Defines the CPU requests limit,
resources.requests.memory	4Gi	Defines the memory requests limit.
autoscaling:		
resources.autoscaling.enabled	false	Defines if autoscaling should be enabled.
resources.autoscaling.minReplicas	1	Defines the minimum replicas for autoscaling.

Attribute	Default	Description
resources.autoscaling.maxReplicas	100	Defines the maximum replicas for autoscaling.
resources.autoscaling.targetCPUUtilizationPercentage	80	Defines the target CPU utilization for autoscaling.

nodeSelector

You can configure a pod so that it can run only on a particular set of nodes using this node selection option. The configured value is used for the pod specification. Kubernetes schedules the pods only onto nodes that have the labels you specify.

Attribute	Default	Description
nodeSelector.kubernetes.io/hostname		Defines the node selector.
tolerations		
affinity		

The default value is**{}** where the scheduler automatically selects the nodes. You can configure the label of interest. Ensure that the default value of **{}** is removed, and the required label is added.

```
# Any node specific requirement
nodeSelector:
  disktype: ssd
  # kubernetes.io/hostname: docker-desktop
```

Or

```
# Any node specific requirement
nodeSelector:
  kubernetes.io/hostname: x-men-ose03
```

Deploy ODC Lifecycle Management Service

The **odc-lifecyclemgmt** service is a microservice that is required as a supporting service for RESTMon 2.2. This microservice includes:

- **RESTMon APIs**: RESTMon APIs that are used by RESTMon to search active alarms in the Elasticsearch.
- **Topology Lifecycle**: The eligible topology entities to be kept alive are republished into TAS.
- **Alarm Reconciliation**: A reconciliation job that runs every 24 hours and clears off old alarms that have not had any updates.

You can deploy this service using the **odc-lifecyclemgmt** yaml file that is available in the **RESTMon 2.2/odc-lifecyclemgmt** folder.

This section provides the following information:

> **NOTE**
> The ODC Lifecycle Management service is required only for RESTMon 2.2 On-Premise deployments.

Deploy ODC Lifecycle Management Service

To deploy the microservice, create the **odc-lifecyclemgmt** deployment and service objects using the **odc-lifecyclemgmt.yaml** file.

Follow these steps:

1. Untar the archive and load the image. The tarball is available in the **RESTMon 2.2>odc-lifecyclemgmt** folder:
   a. Load the **odc-lifecyclemgmt** docker image from the given tarball.
   ```
   docker load < odc-lifecyclemgmt-21.3.1.tar.gz
   ```
   b. Validate the loaded image using the following command:
   ```
   docker images odc-lifecyclemgmt:21.3.1
   ```
   c. (In multi-node deployments) Ensure that the image is loaded on the required node.
2. Update the following environment variables in the **odc-lifecyclemgmt.yaml** file:

   > **NOTE**
   > This YAML file defaults to the namespace of dxdoi. You must change the value appropriately based on the OI deployment namespace.

Environment Variable	Default	Description
ODC_LOG_LEVEL	DEBUG	Defines the level of log information that must be captured in the ODC Lifecycle Management Service log file. **Supported Values:** info or debug
FULL_ODC_ROUTE	http://odc-lifecyclemgmt.dxdoi:8080	Defines the service endpoint for the lifecycle pod. It is the service hostname exposed on the configured port and protocol. Defines the ODC route. The general naming convention is **<service-name>.**<**namespace**> and dxdoi being the namespace used in this deployment.
APM_SERVICES_GATEWAY_URL	http://apmservices-gateway.dxdoi:8004	Defines the service endpoint for the **apmservices-gateway** deployment. It is the service hostname exposed on the configured port and protocol. The general naming convention is **<service-name>.<namespace**> and dxdoi is the namespace used in this deployment. In multi-node deployments, the APM services are deployed in a different namespace. Make sure that the appropriate service hostname is updated here.

3. Run the following command to deploy the service:
   ```
   oc create -f odc-lifecyclemgmt.yaml
   ```
4. Verify the deployment status using the following command:
   ```
   oc get all | grep lifecycle
   ```
   The following image is a sample deployment status:

```
[root@x-men-ose03 ~]# oc get all | grep lifecycle

pod/odc-lifecyclemgmt-66844f9fcd-d6hqd                    1/1       Running           0          1h
service/odc-lifecyclemgmt                      ClusterIP      172.3[redacted]08   <none>                          8080/TCP



deployment.apps/odc-lifecyclemgmt                  1         1         1            1           1h
replicaset.apps/odc-lifecyclemgmt-66844f9fcd                  1         1         1         1h


[root@x-men-ose03 ~]#
```

Additional Information

This section provides the following information:

- Enable Alarm Reconciliation Job
- Check Alarm Reconciliation Job Status
- Monitor Service Health
- Liveness Check
- Readiness Check

Enable Alarm Reconciliation Job

The lifecycle management service executes the alarm reconciliation to clear or delete the older alarms without any update for the last 24hrs (assuming the RESTMon instance ingesting the data is alive). This reconciliation that is enabled by default can be disabled by executing the following API if the auto closure is not required for that tenant.

Name	Description
Method	POST
URL	<OI ingestion/APM Gateway endpoint>**/restmon/v2/config/alarmJobConfig**
Authorization	Bearer <tenant> token
Body	true/false
Response	200 OK

Check Alarm Reconciliation Job Status

Use the following API information to verify the status of the job:

Name	Description
Method	GET
URL	<OI ingestion/APM Gateway endpoint>**/restmon/v2/config/alarmJobConfig**
Authorization	Bearer <tenant> token

Name	Description

Response	200 OK
	`{`
	`"entries":[`
	`{`
	`"key":"odc.lifecyclemanagement.alarmjob.status",`
	`"value":{`
	`"id":null,`
	`"predicates":null,`
	`"uri":null,`
	`"order":0,`
	`"metadata":null,`
	`"enabled":true`
	`},`
	`"tenantId":10`
	`}`
	`]`
	`}`

Monitor Service Health

You can monitor the health of the service using the following details:

Name	Description
Method	GET
URL	<OI ingestion/APM Gateway endpoint>**/restmon/v2/config/alarmJobConfig**
Authorization	Bearer <tenant> token

Response	200 OK
	`{`
	`"entries":[`
	`{`
	`"key":"odc.lifecyclemanagement.alarmjob.status",`
	`"value":{`
	`"id":null,`
	`"predicates":null,`
	`"uri":null,`
	`"order":0,`
	`"metadata":null,`
	`"enabled":true`
	`},`
	`"tenantId":10`
	`}`
	`]`
	`}`

Liveness Check

You can monitor the health of the service using the following details:

Name	Description
Method	GET
URL	<OI ingestion/APM Gateway endpoint>**/restmon/heartbeat/liveness**
Authorization	Bearer <tenant/internal/master> token

Response	200 OK
	`{`
	`"ODC-LifeCycle is running normally"`
	`}`

Readiness Check

You can monitor the readiness of the service using the following details:

Name	Description
Method	GET
URL	<OI ingestion/APM Gateway endpoint>**/restmon/heartbeat/readiness**
Authorization	Bearer <tenant/internal/master> token

Response	200 OK
	`{`
	`"ODC-LifeCycle is ready"`
	`}`

RESTMon 2.1.6

RESTmon supports Helm chart deployment. Helm Charts help you install and upgrade Kubernetes applications.

This section describes the steps to configure and deploy RESTMon 2.1.6 on the cluster using Helm Charts.

Prerequisites for the Deployment

Before you deploy RESTMon, ensure that the prerequisites are met:

- The Docker Engine is installed on the system where you want to deploy RESTMon. For more information, see the Docker Engine Installation documentation.
- Helm 3 is installed.
- RESTMon package is downloaded and the RESTMon docker image is loaded into the docker host system. Run the following command to load the image to the Docker host system from the tar archive:

  ```
  docker load < dx-restmon:2.1.6.tar.gz
  ```

  NOTE
  Ensure that the image is loaded on all the worker nodes. Alternatively, you can push the image to a private docker repository and can pass the image as an argument.
- The Tenant Cohort ID and Tenant Token are available.

  NOTE
  Cohort ID is available on the **Settings > Connector Parameters** page in DX Operational Intelligence. You can generate the tenant token on the **Tokens** page in DX SaaS.
- The Tenant Cohort ID and APM Agent Token are available.

NOTE
Cohort ID and Agent Token are available on the **Settings > Connector Parameters** page in DX Operational Intelligence.

Configure RESTMon

Before you deploy RESTMon, consider the following points:

- By default, the RESTMon container is started on HTTP with port 8080. To start the container on HTTPS with port 8443, pass the following argument during the deployment:

  NOTE
  Before you run the command, place the certificate (PKCS12 or JKS) in the extracted **helm-charts-*-restmon.tgz/Restmon** folder. Run the command from within the extracted folder:

  ```
  --set restmon.settings.restmon_protocol=https --set restmon.settings.restmon_port=8443 --set
   restmon.settings.ssl_key_store=<restmon-certificate> --set
   restmon.settings.ssl_key_store_password=<password> --set restmon.settings.ssl_key_store_alias=<alias>
  ```

- To start the container on any other port, configure the new port and pass the following argument during the deployment:

  ```
  --set restmon.settings.restmon_port=<portnumber>
  ```

- By default, 8080 is used as the service port for HTTP and 8443 for HTTPS. Pass the following argument during the deployment to change the port on which the service is exposed:

  ```
  --set restmon.settings.service_port=<service_port>
  ```

- By default, LoadBalancer is used as the service type. To change to NodePort, pass the following argument during the deployment:

  ```
  --set restmon.service.type=NodePort
  ```

- By default, storageclass (ssd) is used for PV. To change to another storage class, pass the following argument during the deployment:

  ```
  --set restmon.storage.className=ssd-new
  ```

- For the PV to use the NFS Storage, pass the following argument during the deployment:

  ```
  --set restmon.storage.nfs_path=<nfs-path> --set restmon.storage.nfs_server=<nfs-server>
  ```

- To enable supportability metrics, pass the following arguments during the deployment:

  ```
  --set restmon.settings.supportability_instanceName=<instanceName> --set
   restmon.settings.supportability_agentToken=<agentToken> --set
   restmon.settings.supportability_apiEndpoint=<apiEndpoint>
  ```

  NOTE
  For more information, see the Configurations section.

Start or Deploy RESTMon

To deploy RESTMon, run the command with the required settings as arguments.

Follow these steps:

1. Execute the corresponding command to start or create the RESTMon container:
   - **Image is available on all the cluster nodes:**

     ```
     helm install <name> restmon-2.1.x.tgz --set restmon.id=<id> --set restmon.settings.tenant_id=<Cohort-id>
      --set restmon.settings.oi_ingestion_tenant_token=<OI/APM Agent token> --namespace <namespace>
     ```

   - **Image is pushed to the private docker repository:**

```
helm install <name> restmon-2.1.x.tgz --set restmon.id=<id> --set restmon.settings.tenant_id=<Cohort-id>
 --set restmon.settings.oi_ingestion_tenant_token=<OI/APM Agent token> --set restmon.imageName=<private
 docker repository> --namespace <namespace>
```

– **Image is pushed to private docker repository with authentication:**

```
helm install <name> restmon-2.1.x.tgz --set restmon.id=<id> --set restmon.settings.tenant_id=<Cohort-id>
 --set restmon.settings.oi_ingestion_tenant_token=<OI/APM Agent token> --set restmon.imageName=<private
 docker repository> --set restmon.imagepullSecret=<dx-restmon-reg-secret> --namespace <namespace>
```

2. Run the following command to check the deployment status:

```
helm ls -n <namespace>
```

3. Add the profile to the **restmon.json** file using the following POST Profile REST API call:

```
For http:
http://localhost:8080/restmon/api/v1/profiles
For https:
https://localhost:8443/restmon/api/v1/profiles
```

   Provide the profile as the input body. You can take reference of the sample profile from the **profile** directory. For more information, see the Add the Profile illustration. The schema is uploaded when the profile is added.

4. View the logs at the given NFS PV path or from the Kubernetes cloud console if deployed in the cloud cluster.
5. Perform all the other operations using the REST APIs accessible at:

```
For http: http://localhost:8080/restmon/api/swagger-ui/
For https: http://localhost:8443/restmon/api/swagger-ui/
```

   NOTE
   If you make any changes to the schema, you can upload the schema using the following POST Upload Schema REST API call:

```
For http:
http://localhost:8080/restmon/api/v1/schema/{schemaName}
For https:
https://localhost:8443/restmon/api/v1/schema/{schemaName}
```

   Provide the content of the schema file as the input body. You can take reference of the sample schema from the **schema** directory. For more information, see the Upload the Schema illustration.

Configurations

Review this section to understand the different configurations and settings that are available in the **values.yaml** file. Some of these configurations and settings are preconfigured with the default values. You can change the values in the yaml file. Alternatively, you can pass them as arguments during the deployment.

Helm Chart Configurations

The following table lists the Helm Chart configurations that are set at the deployment level. Some of these configurations are pre-configured with the default values. To change these values, you can pass them as arguments during the deployment.

Configurations	Default	Description
restmon.id	1	Indicates the ID. • Must be lower case alphanumeric characters or **-**. • Must start with an alphabetic character and end with an alphanumeric character. For example, **my-name**, or **abc-123**. • Regex that is used for validation is '[a-z]([-a-z0-9]*[a-z0-9])?' `helm install <name> <chart> --set restmon.id=<id>`
restmon.storage.size	1Gi	Indicates the storage size for RESTMon. `helm install <name> <chart> --set restmon.storage.size=<size>`
restmon.storage.className		Indicates the class name. By default, *storageclass (ssd)* is used for Persistent Volumes (PV). To change the storage class, pass the following argument: `helm install <name> <chart> --set restmon.id=<id> --set restmon.storage.className=ssd-new`
restmon.storage.nfs_path	/var/nfs/kubedata/testfolder	Indicates to modify the Persistent Volume to use the NFS storage. `helm install <name> <chart> --set restmon.storage.nfs_path=<nfs_path>`
restmon.storage.nfs_server	test.broadcom.net	Indicates the NFS server name. `helm install <name> <chart> --set restmon.storage.nfs_server=<nfs_server>`

Configurations	Default	Description

restmon.imageName		Indicates the image name. To deploy RESTMon in the On-Premise Kubernetes cluster, ensure that the image is loaded on all the worker nodes. Alternatively, you can push the image to the private docker repository and can pass the image as a helm argument:
		`helm install <name> <chart> --set restmon.id=<id> --set restmon.storage.size=<size> --set restmon.imageName=<private docker repository>`
		When you push the image to a private docker repository that requires authentication, create a Kubernetes secret against the private registry as shown:
		`kubectl create secret docker-registry dx-restmon-reg-secret --docker-server=<docker-registry> --docker-username=<username> --docker-password <password> --docker-email=<docker-email> -n <namespace>`

restmon.service.type	loadbalancer, Nodeport	Indicates the service type. By default, *LoadBalancer* is used as the service type. To change the service type to *NodePort*, pass the following argument: `helm install <name> <chart> --set restmon.id=<id> --set restmon.service.type=NodePort`
restmon.resources.requests.memory	512Gi	Indicates the minimum memory value to be set. `helm install <name> <chart> --set restmon.resources.requests.memory=<memory>`
restmon.resources.requests.cpu	2	Indicates the minimum CPU value to be set. `helm install <name> <chart> --set restmon.resources.requests.cpu=<value>`
restmon.resources.limits.memory	4Gi	Indicates the maximum memory value to be set. `helm install <name> <chart> --set restmon.resources.limits.memory=<memory_lin`
restmon.resources.limits.cpu	4	Indicates the maximum CPU value to be set. `helm install <name> <chart> --set restmon.resources.limits.cpu=<limit>`

Configurations	Default	Description
restmon.nodeSelector	disktype: "ssd"	Enables node selectors to deploy RESTMon on the specific worker node in the Kubernetes cluster. • Ensure that the key value is available in the Kubernetes cluster. • Comment the entire key:value pair to disable the node selector.

Settings

The following table lists all the settings that are defined in the **values.yaml** file. Some of these settings are pre-configured with the default values. To change these values, you can pass them as arguments during the deployment:

Category	Settings	Supported Value/ Example	Description
Generic Settings	**restmon.settings.restmon_protocol**	**Default:** HTTPS **Supported Values/Example:** http, https	Indicates the protocol to start RESTMon. `helm install <name> <chart> --set restmon.id=<id> --set restmon.settings.restmon_protocol https>`
	restmon.settings.restmon_port	8080, 8443	Indicates the port on which the application listens. By default, 8080 is the port that is used for the pod deployment. `helm install <name> <chart> --set restmon.id=<id> --set restmon.settings.restmon_port=<po`
	restmon.settings.service_port	8080, 8443	Indicates the port on which the Kubernetes service is exposed. Default: 8080. `helm install <name> <chart> --set restmon.id=<id> --set restmon.settings.service_port=<se`
	restmon.settings.log_level	info, debug, trace	Indicates the required log level for the application logging. `helm install <name> <chart> --set restmon.id=<id> --set restmon.settings.log_level=<info/ debug/trace>`

Category	Settings	Supported Value/ Example	Description
	restmon.settings.replace_oi_attributes	true, false	Indicates if the DX OI environment variables must replace the existing values in the **restmon.json** file. If true, the DX OI-specific environment variables replace the existing values. `helm install <name> <chart> --set restmon.id=<id> --set restmon.settings.replace_oi_attri false>`
	restmon.settings.jarvis_ingestion_hostname	mi-hostname.8081.broadcom.com	Indicates the NGINX hostname of the Jarvis API ingestion endpoint that is used for the alarms or change events ingestion. Used if replace_oi_attributes=true. `helm install <name> <chart> --set restmon.id=<id> --set restmon.settings.jarvis_ingestion`
	restmon.settings.jarvis_ingestion_port	443	Indicates the port on which the NGINX Jarvis API ingestion endpoint is exposed. Used if replace_oi_attributes=true. `helm install <name> <chart> --set restmon.id=<id> --set restmon.settings.jarvis_ingestion`
	restmon.settings.jarvis_ingestion_protocol	http, https	Indicates the protocol using which the NGINX Jarvis API ingestion endpoint is accessible. Used if replace_oi_attributes=true. `helm install <name> <chart> --set restmon.id=<id> --set restmon.settings.jarvis_ingestion https>`
	restmon.settings.oi_ingestion_api_host	api-services-gateway.broadcom.com	Indicates the APM Gateway ingestion endpoint that is used for topology or NASS metrics data ingestion. Used if replace_oi_attributes=true. `helm install <name> <chart> --set restmon.id=<id> --set restmon.settings.oi_ingestion_api`

Category	Settings	Supported Value/ Example	Description
	restmon.settings.oi_ingestion_api_port	443	Indicates the port on which the APM Gateway ingestion endpoint is exposed. Used if replace_oi_attributes=true. `helm install <name> <chart> --set restmon.id=<id> --set restmon.settings.oi_ingestion_api`
	restmon.settings.oi_ingestion_api_protocol	http/https	Indicates the protocol using which the APM Gateway ingestion endpoint is accessible. Used if replace_oi_attributes=true. `helm install <name> <chart> --set restmon.id=<id> --set restmon.settings.oi_ingestion_api https>`
	restmon.settings.oi_ingestion_tenant_token	eyJ0eXAiOiJKV1QiLCJhbGciOiJFUzI1NiJ9.eyJpbnQiOnRydWUsInRpZCI6NTMsImp0aSI6ImJkZTZkOWVlLTEzZDMtNDVjMy04MDllLTgxNDkwNGE1YjFiMCJ9.3TFeZAgljW48s4uVc3u7fJ7JLc0lTxA_YSWiUo8U1-jMzcZXEVTad1F5sKw00HXD-n_WbB-guFGeqi9xy5EWU	Indicates the tenant or agent token to be used as the authentication for the APM Gateway ingestion endpoint. Used if replace_oi_attributes=true. `helm install <name> <chart> --set restmon.id=<id> --set restmon.settings.oi_ingestion_ten`
	restmon.settings.tenant_id	042F295A-2BE0-4A10-86BA-A5F9ED818BD0	Indicates the Cohort ID of the tenant ingesting data into DX OI. Used if replace_oi_attributes=true. `helm install <name> <chart> --set restmon.id=<id> --set restmon.settings.tenant_id=<tenan`
	restmon.settings.max_number_of_docs	Default: 1000	Indicates the number of JSON documents to be included in a single payload for ingestion to DX OI. `helm install <name> <chart> --set restmon.id=<id> --set restmon.settings.max_number_of_do`
	restmon.settings.restmon_min_memory_limit	512M	Used for -Xms while launching the application. `helm install <name> <chart> --set restmon.id=<id> --set restmon.settings.restmon_min_memo`

Category	Settings	Supported Value/ Example	Description
	restmon.settings.restmon_max_memory_limit	8G	Used for -Xmx while launching the application. `helm install <name> <chart> --set restmon.id=<id> --set restmon.settings.restmon_max_memo`
SSL Settings	**restmon.settings.ssl_key_store_type**	JKS, PKCS12	Indicates the Keystore type for the HTTPS communication. `helm install <name> <chart> --set restmon.id=<id> --set` **`restmon.settings.ssl_key_store`** `PKCS12>`
	restmon.settings.ssl_key_store	restmon.keystore	Indicates the Keystore for the HTTPS communication. `helm install <name> <chart> --set restmon.id=<id> --set restmon.settings.ssl_key_store=<k`
	restmon.settings.ssl_key_store_password		Indicates the Keystore password for the HTTPS communication. `helm install <name> <chart> --set restmon.id=<id> --set restmon.settings.ssl_key_store_pa`
	restmon.settings.ssl_key_store_alias		Indicates the Keystore alias for the HTTPS communication. `helm install <name> <chart> --set restmon.id=<id> --set restmon.settings.ssl_key_store_al`
	restmon.settings.user_name		Indicates the RESTMon username to be used for the basic authentication that is available by default. **Default: admin** `helm install <name> <chart> --set restmon.id=<id> --set restmon.settings.user_name=<usern`
	restmon.settings.password	Default: password	Indicates the password to be used for the basic authentication that is available by default. `helm install <name> <chart> --set restmon.id=<id> --set restmon.settings.password=<passwo`

Category	Settings	Supported Value/ Example	Description
Liveness and Readiness	**restmon.settings.readiness_check_interval_sec**	**Default:** 15 seconds **Interval in seconds (15, 60, so on)**	Indicates the frequency at which the scheduler that checks the readiness of the application runs. If the DX OI endpoints were not available, the health of the application is marked as **not ready**. `helm install <name> <chart> --set restmon.id=<id> --set restmon.settings.readiness_check_`
	restmon.settings.profile_queue_size_max_limit	**Default:** 20000 **Integer**	Indicates the threshold of the pending queue size. If the number of messages in the steaming queue for any of the profiles is more than the configured number, the readiness of the application is marked as **not ready**. `helm install <name> <chart> --set restmon.id=<id> --set restmon.settings.profile_queue_si`
	restmon.settings.profile_queue_size_min_limit	**Default:** 15000 **Integer**	Indicates the limit on the queue size against which the **not ready** application can come back to the **ready** state. If the application becomes **not ready** because of the above configuration against the queue threshold, the application can come back to the **ready** state once the number of messages comes below this configured number. `helm install <name> <chart> --set restmon.id=<id> --set restmon.settings.profile_queue_si`

Category	Settings	Supported Value/ Example	Description
	restmon.settings.max_allowed_running_profiles	**Default:** 5 profiles	Indicates the maximum number of polling profiles that are allowed to run at a time. The readiness of the application is disabled (**not ready**) if the number of profiles that are running at a time is more than this configured value. If the application is **not ready**, further polling and processing do not start until the application comes back to the **ready** state. `helm install <name> <chart> --set restmon.id=<id> --set restmon.settings.max_allowed_runn`
	restmon.settings.db_health_check_interval_sec	**Default:** 15 seconds **Supported Values:** Interval in seconds (15, 60, so on)	Indicates the frequency at which the scheduler which checks the database health is run. The health of the application is marked as **not live** if the database is down or any other issue in connecting to the database. `helm install <name> <chart> --set restmon.id=<id> --set restmon.settings.db_health_check_`
	restmon.settings.health_statistics_log_level	**Default:** ERROR **Supported Values:** error, info, debug, trace	Indicates the new log file restmon_statistics.log is available with the default log level as **ERROR**. `helm install <name> <chart> --set restmon.id=<id> --set restmon.settings.health_statistic info/debug/trace>`

Category	Settings	Supported Value/ Example	Description
	restmon.settings.liveness_check_interval	**Default:** 10_min **Supported Value:** interval in minutes (10, 15, so on)	Indicates the frequency at which the liveness check scheduler is run. This basically checks if the data ingestion for the streaming profile is active, but the processing/publishing could not be completed within the configured time. If the processing is not completed, the liveness of the application is disabled. The application is back to live when the processing is completed. `helm install <name> <chart> --set restmon.id=<id> --set restmon.settings.liveness_check_i`
	restmon.settings.unready_profile_allowed_retries	**Default:** 10 **Supported Value/Example:** Integer	Indicates the maximum number of times the polling profile can be retried for starting processing in case the processing could not be started because of the readiness being down (**not ready**). `helm install <name> <chart> --set restmon.id=<id> --set restmon.settings.unready_profile_`
	restmon.settings.unready_profile_retry_time_sec	**Default:** 10 **Supported Value/Example:** interval in seconds (10, 15, so on)	Indicates the wait time for the profile retry. `helm install <name> <chart> --set restmon.id=<id> --set restmon.settings.unready_profile_`
	restmon.settings.reload_ootb_schemas	**Default:** false	Indicates If the out-of-the-box schemas must be reloaded. `helm install <name> <chart> --set restmon.id=<id> --set restmon.settings.reload_ootb_sche false>`
Supportability Metrics	**restmon.settings.supportability_agentToken**		Indicates the token that is generated in APM. `helm install <name> <chart> --set restmon.id=<id> --set restmon.settings.supportability_a`

Category	Settings	Supported Value/ Example	Description
	restmon.settings.supportability_apiEndpoint		Indicates the API endpoint to send metrics through. `helm install <name> <chart> --set restmon.id=<id> --set restmon.settings.supportability_a`
	restmon.settings.supportability_instanceName	restmon1	Indicates the unique identifier for the RESTMon instance. This has to be unique in the tenant context. This identifier helps in identifying the supportability metrics of each of the instances that are deployed per tenant in the dashboard. `helm install <name> <chart> --set restmon.id=<id> --set restmon.settings.supportability_i`
	restmon.settings.supportability_agentName	restmon|newoiservices|opendataconnector	Indicates the agent name against which the metrics are ingested for the RESTMon connector. This is not required to be modified. `helm install <name> <chart> --set restmon.id=<id> --set restmon.settings.supportability_a`

Liveness Probe Configurations

The following table lists the Liveness probe configurations. All these settings are preconfigured with the default values. To change these values, you can pass them as arguments during the deployment:

Configuration	Supported Value/Example	Description
restmon.livenessProbe.initialDelaySeconds	**Default:** 60 **Supported Value/Example:** Integer in seconds	Indicates the time to wait before performing the first probe by Kubernetes.
restmon.livenessProbe.periodSeconds	**Default:** 30 **Supported Value/Example:** Integer in seconds	Indicates the frequency at which the liveness probe is invoked.
restmon.livenessProbe.timeoutSeconds	**Default:** 30 **Supported Value/Example:** Integer in seconds	Indicates the number of seconds after which the probe times out.
restmon.livenessProbe.successThreshold	**Default:** 1 **Supported Value/Example:** Integer	Indicates the minimum consecutive successes for the probe to be considered successful after having failed.
restmon.livenessProbe.failureThreshold	**Default:** 5 **Supported Value/Example:** Integer	Indicates the number of retries before restarting the pod.

Readiness Probe Configurations

The following table lists the Readiness probe configurations. All these settings are preconfigured with the default values. To change these values, you can pass them as arguments during the deployment:

Configuration	Supported Value/Example	Description
restmon.readinessProbe.enable	**Default:** true **Supported Value/Example:** true, false	Enables or disables the readiness probe for the pod deployment.
restmon.readinessProbe.initialDelaySeconds	**Default:** 60 **Supported Value/Example:** Integer in seconds	Indicates the time to wait before performing the first probe by Kubernetes.
restmon.readinessProbe.periodSeconds	**Default:** 30 **Supported Value/Example:** Integer in seconds	Indicates the frequency at which the readiness probe is invoked.
restmon.readinessProbe.timeoutSeconds	**Default:** 10 **Supported Value/Example:** Integer in seconds	Indicates the number of seconds after which the probe times out.
restmon.readinessProbe.successThreshold	**Supported Value/Example:** Integer	Indicates the minimum consecutive successes for the probe to be considered successful after having failed.
restmon.readinessProbe.failureThreshold	**Supported Value/Example:** Integer	Indicates the number of retries before marking the pod as **unready** and the data routing is stopped.

Upgrade

After the deployment, you can upgrade RESTMon using helm.

This section describes how to upgrade RESTMon to the latest version.

Pre-Upgrade

Before you start the upgrade process, perform the following tasks:

- Deactivate the polling profiles using the API. For more information, see the *RESTMon APIs for a Profile* section.
- Take a backup of the **config** and the **schema** folders under the **data** folder in the container.

Upgrade RESTMon

The following upgrade paths are supported:

- 2.2 to 2.2.1
- 2.1.6 to 2.2.1
- 2.1.6 to 2.2

Run the following command to upgrade RESTMon:

- Upgrade RESTMon:
  - 2.2 to 2.2.1
    ```
    helm upgrade <chart name> dx-restmon-2.2.1.tgz -n <name space> -f values.yaml
    ```
  - 2.1.6 to 2.2.1
    ```
    helm upgrade <chart name> dx-restmon-2.2.1.tgz -n <name space> -f values.yaml
    ```
  - 2.1.6 to 2.2
    ```
    helm upgrade <chart name> dx-restmon-2.2.tgz -n <name space> -f values.yaml
    ```

After the upgrade is successful, the old pod is terminated, and the latest pod is activated.

- Update the **values.yaml** to modify the nfs server details and execute the below command:

```
helm upgrade <deployname> <chart.zip> --f values.yaml --namespace <namespace>
```

Post Upgrade

After the upgrade is successful, perform the following tasks:

- Ensure that the swagger UI is accessible and all the other endpoints are accessible.

```
For http: http://<localhost>:8080/restmon/api/v1/oiconnection
For https: https://<localhost>:8443/restmon/api/v1/oiconnection
```

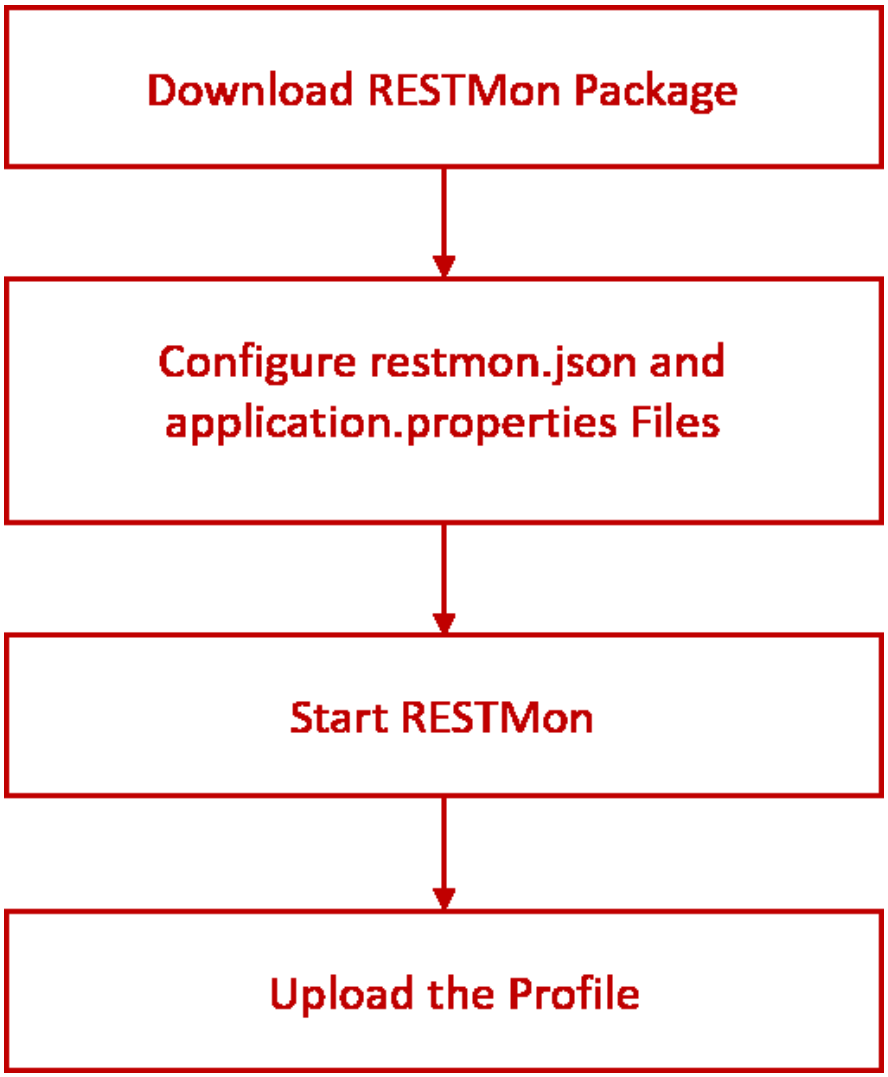
- Activate all the profiles that you deactivated during the pre-upgrade process using the API. For more information, see the *RESTMon APIs for a Profile* section.

Virtual Machine (VM) Version

The **DX-RESTmon-2.2.zip** file in the **vm** folder contains the artifacts for deploying RESTMon on virtual or standalone Windows or Linux machines. Configure the properties described in this section and then start RESTMon.

Deployment Overview

The following image illustrates the deployment flow for the VM version:



Configure the Properties

Configure the following properties before you start RESTMon:

- Define Endpoints for Data Ingestion
- Configure RESTMon Authentication
  - Modify the Default User Credentials
  - Enable Bearer Authentication
- Verify HTTPS Configuration
- Use Your Certificate

Define Endpoints for Data Ingestion

You can define the endpoints using the single endpoint method for all the data categories.

Follow these steps:

1. Open the *restmon.json* file that is available in the **<restmon>\vm\DX-RESTmon-2.2\DX-RESTmon-2.2\config** folder.
2. Update the oisettings section as shown in the following sample:

NOTE
You can get the values for the following properties on the following pages:

- Hostname and Cohort ID: **Settings > Connector Parameters** page
- Tenant Token: **Launch Pad > Settings > Tokens** page


```
{
  "restmon" : {
    "config" : {
      "oisettings" : {
        "type" : "https",
        "hostname" : "apmservices-gateway-ao-apm.apps.aiops.broadcom.net",
        "token" :
 "eyJ0eXAiOiJKV1QiLCJhbGciOiJFUzI1NiJ9.eyJpbnQiOnRydWUsInRpZCI6NzMzLCJqdGkiOiI2ZDlhOTNlMi01MzJkLTQ4OWMtYWY2Yi02NGNjN
GaHalHOsNHdffoMKMz-msmcC6WQD5Gs_sB38Fxd44X39cSl32ZzMQ",
        "port" : "443",
        "cohortid" : "5981C2EE-F852-4611-B705-C357942C85E5",
        "adminroute" : "ADMIN_ROUTE",
        "httptimeout" : "30000"
      },
      "settings" : {
        "publish_pool_size" : 20,
        "request_pool_size" : 100,
        "httpclientMaxConnections" : "5000"
      },
….
```


Configure RESTMon Authentication

RESTMon supports the following authentication types:

- **Basic**: Basic Authentication requires access credentials to authenticate the API requests. Provide valid credentials (username and password) to access and send the API Requests.
- **Bearer**: Bearer Authentication requires a bearer token to authenticate the API requests. Provide a valid bearer token to access and send the API requests using the following format:


```
Bearer
 eyJ0eXAiOiJKV1QiLCJhbGciOiJFUzI1NiJ9.eyJpbnQiOnRydWUsInRpZCI6MTMsImp0aSI6ImMxNDk0MWYyLTgwMjAtNGIxNS05YTM3LTBjNWY2Nm
dZV8HAOsqd6dAwsxbrLmZWP_mI4M0GsQyZyaqmgXK-BcFPo-5lZy2yW2RoOoyYQ
```


NOTE
Basic authentication is enabled by default. You can modify the default user credentials and also change the authentication to Bearer if required.

Modify the Default User Credentials

You can modify the default user credentials, if necessary, in the **application.properties** file.

Follow these steps:

1. Navigate to the **../config/application.properties** file.
2. Provide the following information in the **#custom username and password** section:
   - **security.user.name**: Enter the user name based on your requirements. **Default:** *admin*.
   - **security.user.password**: Enter the required password. **Default:** *password* (encrypted format)
   - **passwordEncrypted**: Enter **false** if you do not want password encryption.

Default is true. When you start the application, the encryption is enabled (changed to *true*) and is applied at the start of the application. The password that you provide in plain text is then encrypted.

3. Save your changes.

Enable Bearer Authentication

You can change the authentication to bearer if required.

Follow these steps:

1. Generate the tenant token on the **Launch Pad > Settings > Tokens** page.
2. Navigate to the **../config/application.properties** file.
3. Set the authentication type as basic:
   ```
   restmon.authType=bearer
   ```

Verify HTTPS Configuration

By default, RESTMon uses HTTPS for all inbound and outbound communication. This implies that all the incoming and outgoing data is sent over HTTPS, making the data transfer more secure.

Verify that the following parameters in the ***application.properties*** file have the correct values for the HTTPS support:

- **server.port**: Enter the server port for HTTPS. **Default:** 8443.
- **server.ssl.key-store-type**: Enter the KeyStore format (JKS or PKCS12). **Default:** PKCS12.
- **server.ssl.key-store**: Enter the path to the KeyStore, which contains the certificates. For example, **config/restmon.keystore**.
- **server.ssl.key-store-password**: Enter the password used to generate the certificate. For example, **restmon**.
- **server.ssl.key-alias**: Enter the alias that is mapped to the certificate. For example, **restmonhttps**.
- **isKeyStorePasswordEncrypted**: Enter whether the KeyStore password is encrypted or not.

> **NOTE**
> The default KeyStore file (**../config/restmon.keystore**) is packaged with RESTMon. However, you can generate a self-signed certificate. You can also use your SSL certificate and update the details accordingly in the *application.properties* file.

Use Your Certificate

If you are already using HTTP for RESTMon and you want to use HTTPS as the data transfer type, you can enable HTTPS by generating an SSL certificate and modifying the ***application.properties*** file.

Follow these steps:

1. Create a self-signed SSL certificate. We recommend the PKCS12 format.
   a. Open the command prompt on the system where RESTMon is installed and enter the following command.
      ```
      keytool -genkeypair -alias tomcat -keyalg RSA -keysize 2048 -storetype PKCS12 -keystore keystore.p12 -validity 3650
      ```
   b. Enter the password for your Keystore at the prompt.
      Once your password is accepted, you are asked to enter your first name, last name, city, and country. You may skip this step. After you have entered all the required information, a Keystore is created.
   c. Verify the Keystore using the following command:
      ```
      keytool -list -v -storetype pkcs12 -keystore keystore.p12
      ```
2. Edit the ***application.properties*** file.
   a. Open the *application.properties* file that is available in the **<*Restmon*/config>** directory.
   b. Comment the HTTP section by adding #.

c. Uncomment the following HTTPS options by removing # and updating your configuration:
   - **server.port**: Enter the server port for HTTPS.
   - **server.ssl.key-store-type**: Enter the KeyStore format (JKS or PKCS12).
   - **server.ssl.key-store**: Enter the path to the KeyStore which contains the certificates.
   - **server.ssl.key-store-password**: Enter the password used to generate the certificate.
   - **server.ssl.key-alias**: Enter the alias that is mapped to the certificate.
   - **security.require-ssl**: Enter if only HTTPS requests should be accepted.

d. Save the ***application.properties*** file.

Start RESTMon

After the configurations are done, execute the following commands to start RESTMon. The batch and script files are available in the **RESTMon-2.2\vm\DX-RESTmon-2.2\DX-RESTmon-2.2** folder.

- **Windows**: run.bat
- **Linux**: run.sh

Upload Profile

After you start RESTMon, upload the profile. To upload, provide the profile as the input body to the **restmon.json** file using one of the following methods:

- Add Profile in Swagger using the following POST profile REST API call:
  `http://localhost:8080/restmon/api/v1/profiles`
- Add the profile information directly to the **restmon.json** file. You can take the sample profile from the **profile** directory.

> **NOTE**
> For more information, see the Add the Profile section. You can perform the other operations using the REST APIs accessible at **http://localhost:8080/restmon/api/swagger-ui/**.

Upgrade RESTMon

Perform the following steps to upgrade RESTMon 2.1.6 to 2.2.

Follow these steps:

1. Back up the profiles and schemas.
2. Start RESTMon 2.2 following the steps described on this page.
3. Reupload the profiles.

Using

Using RESTMon, you can ingest data from third-party tools and products into DX OI.

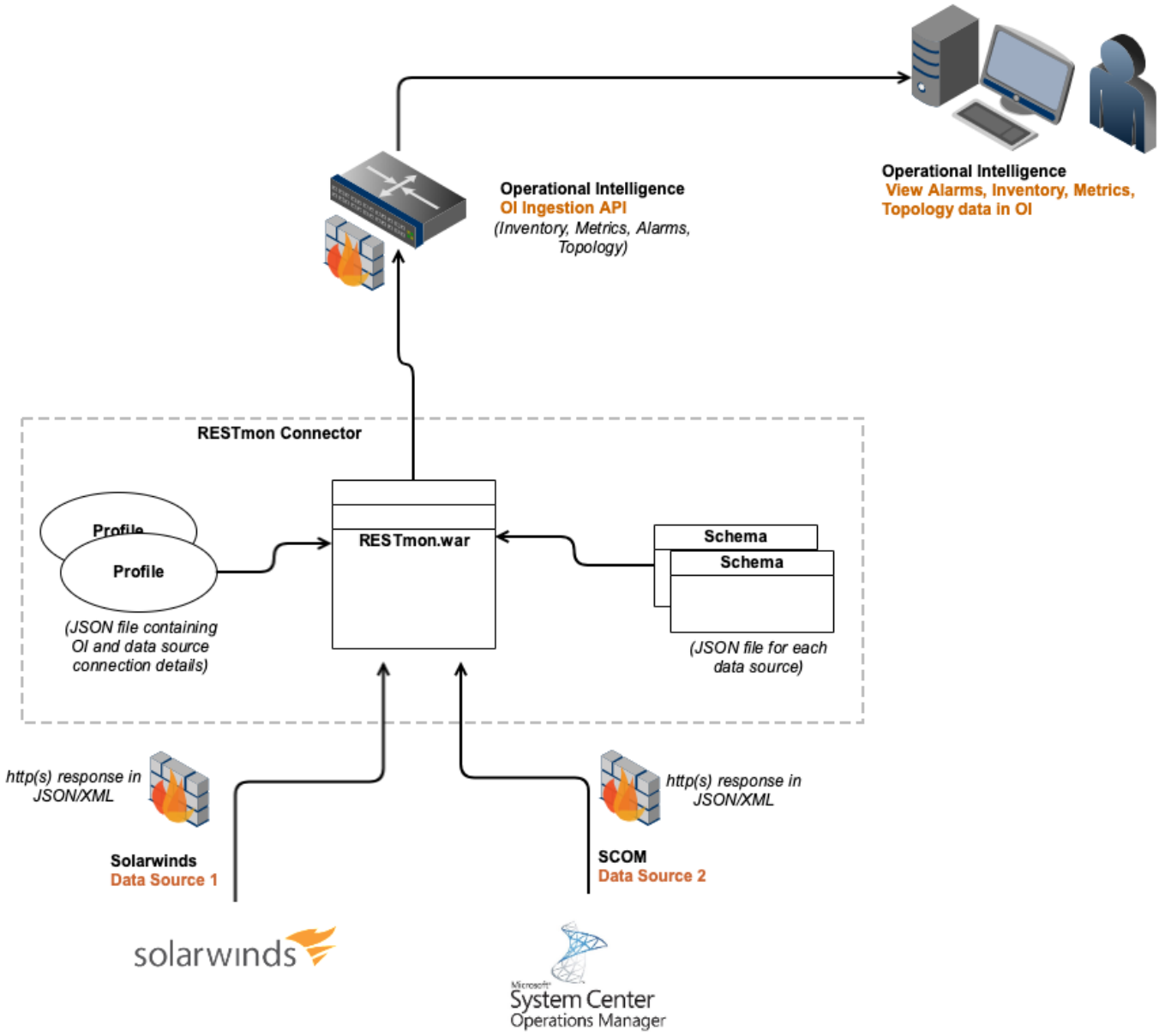
DX Operational Intelligence (DX OI) consumes both structured and unstructured data such as topology, metrics, traces, alarms, logs, and groups from heterogeneous sources. Using RESTMon, you can ingest data from third-party tools and products into DX OI.

How the Data Ingestion Works

To ingest data, the source tool must have exposed REST APIs that can be used to get the required monitoring data at regular time intervals. RESTMon queries the REST APIs of the source tool and in return, gets the data (metrics, alerts, and so on). This data is then formatted as per the requirements of the DX OI Ingestion API, and correlations are made between Alerts and Inventory, Alerts and Metrics, and so on so that you continue to see the relationships in DX OI as they exist in the source tool.

If the data sources do not have the exposed REST APIs or the security policies prohibit the incoming calls, you can either send limited data, such as Alarms only, or you can use the RESTMon REST endpoint. The tools can then securely connect to this endpoint and send in data using Webhooks or any other alternative mechanisms.

The following diagram illustrates how the RESTMon Connector works with the (Solarwinds and SCOM) data sources.



Schema Types

RESTMon uses the following schemas types for parsing data:

- **PULL Schema (Polling):** In the PULL schema, RESTMon pulls data from other sources. The URLs section of the PULL schema defines how and where the data must be pulled from. The type of request and response can also be mentioned here. Dynatrace, AppDynamics, Datadog, Solarwinds, and others are examples of PULL schemas.
- **PUSH Schema (Streaming):** In the PUSH schema, the outside data sources push data into RESTMon. The data is ingested to RESTMon through an API (*<restmon-hostname>:<port>/v1/logs*). The URLs section of the PUSH schema must be empty, as RESTMon is not expecting to pull any data from outside sources. Syslog, Google Cloud Monitoring, and so on are examples of PUSH schemas.

Supported Data Types for Ingestion

You can ingest the following types of data into DX OI:

- **Alarms:** You can ingest alarms from different data sources into DX OI using RESTMon. In DX OI, you can view all the alarms either as part of a Service or Alert Queues. The following relationships are maintained while importing alarms:
  - Alarms to Metric mapping
  - Alarms to Topology/Inventory mapping

  As a result, you can view metrics for which alarms were generated. You can also view the source, which may be a process, infrastructure, business transaction, and so on. RESTMon closes and updates alarms based on the data coming from the source. This ensures that you have a real-time view of the alarms.
- **Metrics:** You can view metrics that are ingested from different source systems along with their CI relationship in DX OI. You can program the schema to group metrics together. You can also define a hierarchy between them for easy retrieval and viewing in Performance Analytics in DX OI. All metrics are automatically eligible for anomaly detection.
- **Topology:** You can import the entire service or monitored topology into DX OI using RESTMon. You can also import the relationships between different CIs as edges to these vertices. All data is regularly refreshed and kept up to date. RESTMon also employs correlation rules, so you do not see redundant entries in the Topology in DX OI. You can:
  - View the entire monitored landscape, which is monitored using ‘n’ number of products.
  - View Alarms – Topology/Inventory/Service Mapping
  - View Alarms – Metrics Mapping
  - Any redundancies in topological elements are intelligently filtered

Configure Data Ingestion Process

The third-party product integration with DX Operational Intelligence involves the following steps:

- Configure the Third-party Environment
- Add the Profile
  - Configure Proxy
- View the Data in DX Operational Intelligence
- Upload the Schema

Configure the Third-Party Environment

Before you start the integration, configure the third-party environment if the integration requires it.

> **NOTE**
> For more information, see the Integrations section.

Add the Profile

Configure the profile to connect to your third-party environment. You can also configure the **profile.json** file to filter the entities and the related data by attributes such as application name or hostname, and so on so that only the required data is ingested and displayed on the DX Operational Intelligence UI.

You can add the profile information using one of the following methods. When the profile is added, the schema is also uploaded automatically:

- Add the Profile in Swagger
- Add the Profile Information Directly

NOTE
Prerequisites:

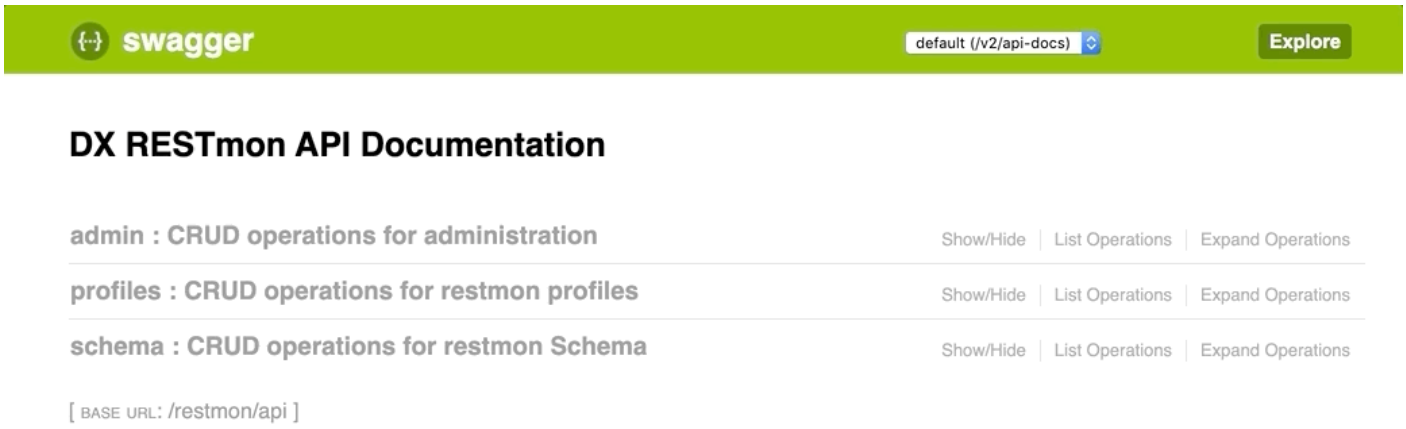
Before you add the profile, ensure that the following requirements are met:

- Have access to DX Operational Intelligence.
- RESTMon is installed and deployed successfully.
- The OI Connection details are updated.
- To filter the entities, the attribute filter is added to the profile.json file. For more information, see the Filter Entities and Related Data Before Ingestion section.
- Configure Proxy if required. For more information, see the Configure Proxy section on this page.
- (Required for some products) For Java 11, the **contains** method is replaced with **includes** in the schema.

Add the Profile in Swagger

Follow these steps:

1. Navigate to the **<restmon\profile>** folder.
2. Open the required profile.json file.
3. Copy the profile information to a text editor and update the required fields.
4. Open the RESTMon Swagger API UI:
   - **For http:**
     `http://localhost:8080/restmon/api/v1/profiles`
   - **For https:**
     `https://localhost:8443/restmon/api/v1/profiles`
5. Expand the **Profile** group and select the **POST** operation.
6. Paste the updated profile information in the **inputBody** as shown in the following illustration.



7. Click **Try it out!** to update the **restmon.json** file.
8. Perform all other operations using the RESTMon APIs accessible at:
   - **For HTTP:**
     `http://localhost:8080/restmon/api/swagger-ui/`
   - **For HTTPS:**
     `https://localhost:8443/restmon/api/swagger-ui/`

Add the Profile Information Directly

You can add the profile information directly to the **restmon.json** file. You can take the sample profile from the **profile** directory.

Follow these steps:

1. Navigate to the **<restmon\profile>** folder.
2. Open the required profile.json file.
3. Update the profile section as shown:

```
"profiles" : [ {
      "profile" : {
        "name" : "dynatracesimulator",
        "active" : "yes",
        "schema" : "dynatracesimulator",
```

```
      "polling_interval_secs" : "300",
      "inventory_topology_fullsync_interval_mins" : "30",
      "topology_ttl_mins" : "30",
      "keepalive" : "yes"
    },
    "servicedefinition" : {
      "name" : "",
      "status" : ""
    },
    "restapiconnectdetails" : {
      "type" : "http",
      "hostname" : "10.17.185.08",
      "port" : "9177",
      "authentication" : "urltoken",
      "username" : "",
      "password" : "",
      "realmdomain" : "",
      "token" : "",
      "httptimeout" : "30000",
      "checkcert" : "no"
    },
    "monitored_groups" : {
      "Hosts" : "yes",
      "Alarms" : "yes",
      "Processes" : "yes",
      "Services" : "yes",
      "Application" : "yes",
      "host-group" : "no",
      "Hosts_Inventory" : "no",
      "Processes_Inventory" : "no",
      "Services_Inventory" : "no"
    }
  }, {
    "profile" : {
      "name" : "googlecloudmonitoring",
      "active" : "yes",
      "schema" : "googlecloudmonitoring",
      "product_name" : "Google Cloud Monitoring",
      "product_version" : "17.8.1.1",
      "streaming" : "yes",
      "polling_interval_secs" : 300,
      "batch_size" : 1000,
      "batch_wait_time_milli" : 2000,
      "inventory_topology_fullsync_interval_mins" : "1440",
      "topology_ttl_mins" : "1440",
      "streaming_array_size" : 10,
      "is_array_input" : "true",
      "keepalive" : "no"
    },
    "restapiconnectdetails" : {
      "type" : "http",
      "hostname" : "",
      "port" : 9600,
```

```
            "authentication" : "",
            "username" : "",
            "password" : "",
            "realmdomain" : "",
            "token" : "",
            "httptimeout" : 300,
            "checkcert" : "no"
          },
          "monitored_groups" : {
            "Topology" : "yes",
            "Alerts" : "yes"
          }
        },
```

Configure Proxy

If the data source or the DX Operational Intelligence environment is configured with proxy, use the following samples to update the *restmon.json* file. RESTMon supports the following authentication types for a proxy configuration:

- Basic
- Digest
- OAuth
- NTLM

Data Source Proxy

If your data source is configured with proxy, then update the proxy details in the **profiles** section of the *restmon.json* file. The following snippet is a snippet. Replace the values with your environment details.

```
"restapiproxy" : {
"type" : "<http|https>",
"hostname" : "<proxy host name or ip address>",
"port" : "<proxy port >",
"authentication" : "<authentication type>",
"username" : "<proxy username>",
"password" : "<proxy password>",
"realmdomain" : "<realm>",
"token" : "<token>",
"httptimeout" : "<time out>",
"checkcert" : "<yes|no>"
},
```

Operational Intelligence Proxy

If your DX OI is configured with proxy, then update the proxy details in the **profiles** section of the *restmon.json* file. The following snippet is a sample. Replace the values with your environment details.

```
"oiproxydetails" : {
"type" : "<http|https>",
"hostname" : "<proxy host name or ip address>",
"port" : "<proxy port >",
"authentication" : "<authentication type>",
"username" : "<proxy username>",
"password" : "<proxy password>",
"realmdomain" : "<realm>",
"token" : "<token>",
```

```
"httptimeout" : "<time out>",
"checkcert" : "<yes|no>"
},
```

Data Source Reverse Proxy

If your data source is configured with reverse proxy, then update the proxy details in the **profiles** section of the *restmon.json* file. The following snippet is a sample. Replace the values with your environment details.

```
"restapiconnectdetails" : {
"type" : "<http|https>",
"hostname" : "<proxy host name or ip address>",
"port" : "<proxy port >",
"authentication" : "<authentication type>",
"username" : "<proxy username>",
"password" : "<proxy password>",
"realmdomain" : "<realm>",
"token" : "<token>",
"httptimeout" : "<time out>",
"checkcert" : "<yes|no>"
},
```

Operational Intelligence Reverse Proxy

If your DX OI is configured with reverse proxy, then update the proxy details in the **profiles** section of the *restmon.json* file. The following snippet is a sample. Replace the values with your environment details.

```
"oiconnectdetails" : {
"type" : "http",
"hostname" : "logcollector.oilab",
"port" : "80",
"category" : "metrics",
"maxnoofdocuments" : "1000",
"path" : "/mdo/v2/aoanalytics/ingestion/",
"oi_pool_size" : 4
},
```

View Data in DX OI

To view the ingested data, navigate to Alarm Analytics, Service Analytics, and Performance Analytics pages in DX OI which provide an overview and insights into service, raw, and anomaly alarms.

On the **Alarm Analytics** page, you can view information about the alarms, inventory, and topology. For alarms, you can view information such as Alarm ID, Alarm Type, Configuration Item, Affected Metric, and Alarm URL. Using this URL, you can navigate to the third-party product. In the **Affected Metric** tab, you can view the details of the affected metric. If the alarm is associated with any service, you can view the details in the **Impacted Service** tab.

On the **Services** page, you can monitor the topology by creating a service. Once the service is created, you can view the service health and the number of alarms that are associated with this service. Click the service to view the details and also the topology.

On the **Performance Analytics** page, you can view details such as the Entity Name, Service, Type, and so on. Click any entity and in the **Available Metrics** section select the metrics to view. Graphs are displayed for the selected metrics.

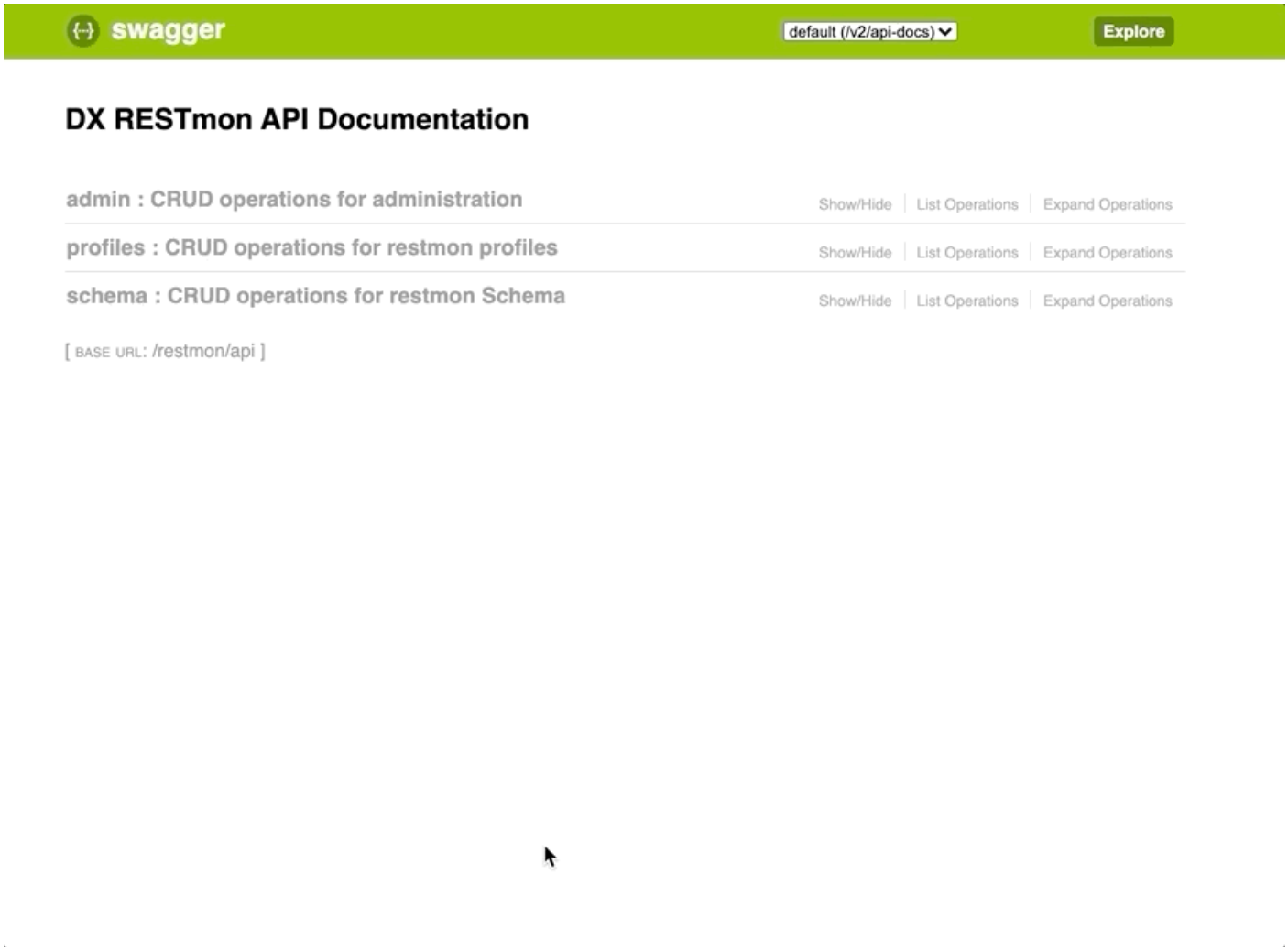
In DX Operational Intelligence, you can also generate notifications for the alarms using the Webhook channel.

Upload the Schema

The schema is automatically uploaded when you add the profile. Perform the following steps only if you want to upload the updated or edited schema.

Follow these steps:

1. Navigate to the **<restmon\schema>** folder.
2. Open the required schema.json file.
3. Copy this schema information to a text editor and update the required fields.
4. Open the RESTMon Swagger API UI:
   - **For http:**
     `http://localhost:8080/restmon/api/v1/schema/{schemaName}`
   - **For https:**
     `https://localhost:8443/restmon/api/v1/schema/{schemaName}`
5. Expand the **Schema** group and select the **POST** operation. For more information, see the POST Upload Schema REST API call section.
6. Paste the schema information in the **inputBody** as shown in the following illustration.



7. Click **Try it out!**
8. Perform all other operations using the RESTMon APIs.

Filter Entities and Related Data Before Ingestion

You can configure RESTMon to ingest and display only the required information on the DX Operational Intelligence UI using the attribute filter. For example, you can define the attributes such as application name or hostname in the attribute filter, and RESTMon filters the entities and the related data by these attributes.

To filter the data, add the following criteria in the profile.json file:

```
"attribute filter":[ {
      "name":"<attribute>",
      "value":[ "<value>" ]
   } ]
},
```

You can filter the data by applying a filter on single or multiple attributes. When the filter includes multiple attributes, the OR operation is used to fetch the results.

> **NOTE**
> The filter attributes are available in the Inventory and Topology sections of the schema. Alternatively, if the data is already ingested into DX Operational Intelligence, you can find the attributes in the **Entity Details** section on the **Monitored Inventory** page.

Sample Code: The following sample illustrates the code to ingest data only for agent **6422750-vm-5|TomcatProcess|Tomcat Agent**.

```
{
   "profile":{
      "name":"appdynamics",
      "active":"yes",
      "schema":"appdynamics",
      "polling_interval_secs":"60",
      "inventory_topology_fullsync_interval_mins":"30",
      "topology_ttl_mins":"240",
      "keepalive":"yes"
   },
   "servicedefinition":{
      "name":"",
      "status":""
   },
   "restapiconnectdetails":{
      "type":"https",
      "hostname":"20230101163400.saas.appdynamics.com",
      "port":"",
      "authentication":"Bearer",
      "username":"",
      "password":"",
      "realmdomain":"",
      "token":"b9coShsxBS1fDEuH4vSnlU8lZP4bbvXTmKmBbhgTi/zsPaSjVzPmUdawmfquZTJQkezlf7d6lPCqi+9qwVTsh/
Bxd8YSVG1So27VpS9L9sOXLn9obFQhxLzUdmfe8IueOwU7C3qZqoROPBpLphxXYvDKV7OgHBVHH031Z4Emh7hjjw34206hMAjdcMCUdL0K3EUWlpBA1OFpX
Zkw==",
      "httptimeout":"120000",
      "checkcert":"no"
   },
   "monitored_groups":{
      "JVM":"yes",
      "Agent":"yes",
      "Hardware Resources":"yes",
```

```
        "Applications":"yes",
        "Databases":"yes",
        "Application Infrastructure Performance":"yes",
        "Overall Application Performance":"yes",
        "CPU":"yes",
        "tier-tier-backend":"yes",
        "Business Transactions":"yes"
    },
    "attribute_filter":[
        {
            "name":"agent",
            "value":["6422750-vm-5|TomcatProcess|Tomcat Agent"]
        }]
}
```

The attribute filter also supports the REGEX pattern. For example, this filter ingests data only for all the hosts that start with **642750-vm**:

```
"attribute_filter":[
        {
            "name":"hostname",
            "value":["642750-vm-.*"]
        }]
}
```

Additional Usage Examples Reference

This section provides additional examples.

- Usage example to filter by the same attribute but with different values.

```
"attribute_filter":[{
        "name":"hostname",
        "value":[ "642750-vm-5","642750-vm-7" ]
          }]
```

- Usage example to filter by different attributes.

```
"attribute_filter":[{
        "name":"hostname",
        "value":[ "642750-vm-5" ]
    },
    {
        "name":" applicationName",
        "value":[ "Test Application Name" ]
    }]
```

- Example using REGEX:

```
"attribute_filter":[{
        "name":"name",
        "value":["agent-[^1-3]"]
    }]
```

Topology Correlation Support

Using topology correlation, you can define topology mapping for devices from monitored product data.

The entities such as host machines, critical servers, applications, processes, services, routers, switches, firewalls, AD Servers, Volumes, Storage devices that the monitoring tool monitors must be ingested to DX OI for the creation of service and to utilize the analytics capabilities of AIOps platform. The collection of these entities and the relationship between them is represented as a graph (TAS model) for the given monitored product.

Note: By default the payload received is assumed to have the complete topology. RESTMon calculates the delta topology by comparing the vertices or edges that are derived from the current payload with the already published topology. Only this delta data is published to DX OI/TAS for every polling interval.

- Assuming id1, id2 are the vertices that are ingested from the first run.
  - Assume that the payload for the second run contains "id1" and "id2", then no topology is ingested to DX OI.
  - Assume that the payload for the third run contains "id1", "id2" and "id3", then only "id3" is ingested to DX OI.
  - Assume that the payload for the fourth run contains "id2" and "id3", then the vertex "id1" is marked for deletion by mapping the "endtime": "-1" and is ingested to DX OI.
- Full sync of the topology happens based on the full sync frequency given in the profile, which is by default 24 hours.

You can define the topology mapping from the monitored product data to the DX OI mapping in the schema under the **topology** section as shown:

```
{
    "appdynamics": {
        "definition": {
            "resource_category": null,
            "uploadedBy": "RESTMON",
            "updatedBy": "",
            "version": "2.0",
            "defaults": {
                "port": 443,
                "interval": 60,
                "httptimeout": 30000
            },
            "auth": "basic",
            "xml_ns": "",
            "name": "appdynamics",
            "type": "https"
        },
        "urls": [],
        "topology": [{
                "xml_ns": "",
                "url": "applications",
                "group": "Applications",
                "layer": "CUSTOM",
                "attributes": {
                    "oi": {
                        "name": "$[*].name",
                        "entity_id": "$[*].id",
                        "entity_name": "$[*].name",
                        "type": "Application",
                        "product": "AppDynamics",
                        "ci_unique_id": "$[*].name",
                        "hostname": "$[*].name",
                        "applicationname": "$[*].name",
                        "hasentityidnamedetails": "true",
                        "globalContext": "true"
```

```
                    }
                }
            }
        ],
        "alarms": [],
        "metrics": [],
        "calculated_methods": [],
        "calculated_metrics": [],
        "groups": [],
        "change_events": []
    }
}
```

The following table lists the various attributes that you can define for this mapping:

Attribute Name	Mandatory	Attribute Type	Attribute Not Sent to DX OI /TAS	Description
ci_unique_id	Yes	String	No	Unique value identifying the configuration_item or topology element being monitored. This is used in creating the externalID (the unique identifier in TAS) for the monitored entity.
name	Yes	String	No	Name describing the topology entity.

Attribute Name	Mandatory	Attribute Type	Attribute Not Sent to DX OI /TAS	Description
type	Yes	String	No	Type of the topology entity. Reference types that are supported by TAS are: *EXTERNAL*, *BUSINESSTRANSACTION*, *EJB*, *EJBCLIENT*, *DATABASE*, *DATABASE_SOCKET*, *SOCKET*, *WEBSERVICE*, *WEBSERVICE_SERVER*, *JMSSERVER*, *SERVLET*, *TRANSACTION_PROCESSOR*, *GENERICFRONTEND*, *GENERICBACKEND*, *DEFAULT*, *AUTOMATICENTRYPOINT*, *APPLICATION_ENTRYPOINT*, *INFERRED_DATABASE*, *INFERRED_SOCKET*, *INFERRED_WEBSERVICE*, *INFERRED_GENERICBACKEND*, ENTERPRISE_MANAGER, *EM_MASTER*, *EM_PROVIDER*, *EM_MOM*, *EM_COLLECTOR*, *APM_SAAS*, *AGENT*, *EM_DATABASE*, *DATABASE_SERVER*, *AGENT_CONNECTION*, *EM_CONNECTION*, *EMDB_CONNECTION*, *STRUTS*, *HOST*, *EXPRESSJS*, *TIBCOPROCESS*, *DOCKER*, *OPENSHIFT*, *KUBERNETES*, *SPRINGCONTROLLER*, *SPRINGSERVICE*, *MESSAGE_QUEUE*,*SPRINGASYN* **Not Allowed** : SERVICE
product	Yes	String	No	The product name from which the data being ingested.
applicationname		String	No	Required, if the topology is specific to the application entity or the topology element is mapping to any application.

Attribute Name	Mandatory	Attribute Type	Attribute Not Sent to DX OI /TAS	Description
hasapplicationdetails	N/A	N/A	Yes	Used internally by RESTMon. Similar to hostname, if the url payload used does not have the application name for all the entities and can be derived from other url entities, the mapping with the availability of the application be marked as "hasapplicationdetails: true". Other parent/ child entities get the application name attached based on the mapping provided for parent_type/type and entity_id.
hostname	Yes	JSON array	No	Maps to the hostname on which the entity is being deployed. Ex: FQDN name, non-FQDN name, any other aliases that source products collect. Note: If the payload used for the entity does not have the hostname available with the given payload, instead has the parent ci_unique_id to which this entity is related. In this case, the hostname can be mapped to an empty string or removed from the schema mapping, and based on the configured parent_ci_unique_id the hostname is derived (assuming the vertex with the hostname has configured "hashostdetails":"true").
FQDNHostname	No (Yes, for the device inventory to help in the host correlation)	String	No	Fully qualified hostname of the device.

Attribute Name	Mandatory	Attribute Type	Attribute Not Sent to DX OI /TAS	Description
hashostdetails	N/A	N/A	Yes	Used internally by RESTMon. This has to be marked "true" if that particular element is the parent of any other element and has the hostname details.
ipAddresses	No (Yes, for device inventory to help in the host correlation)	JSON array	No	List of IpAddresses the hostname of the topology element maps to. Should be a JSON array with all ip addresses of that host or device.
macAddresses	No (Yes, for device inventory to help in host correlation)	JSON array	No	All mac addresses of that host/device.
parent_ci_unique_id	Yes, if a relationship to be created between two vertices.	N/A	Yes	Used internally by RESTMon. • Used to represent the parent entity against which the edge is created. • edge is created from parent_ci_unique_id to ci_unique_id. • If more than one parent_ci_unique_id to be defined in single mapping, can be done using separator "&&". Example: "parent_ci_unique_id" : "id1&&id2&&id3"
parent_type	N/A	N/A	Yes	Used internally by RESTMon. • Used to create the relationship/ edge between two elements that are defined by ci_unique_id and combination of parent_ci_unique_id/ parent_type

Attribute Name	Mandatory	Attribute Type	Attribute Not Sent to DX OI /TAS	Description
semantic	No	String	No	• The attribute that is used for the edge creation in attaching the proper relationship between two vertices, represented by ci_unique_id and parent_ci_unique_id. • Default is null, which marks the direct from/to relationship between two vertices. • If more than one parent_ci_unique_id is defined in a single mapping, semantic as well to be defined with matching number that is separated by &&. **Valid values:** null, contains Example: "semantic": "contains"
child_ci_unique_id	Yes, if a relationship to be created between 2 vertices	N/A	Yes	Used internally by RESTMon. • If there exists an entity that is a child of the current vertex definition(ci_unique_id), that can be defined against child_ci_unique_id. • This is used in creating the parent-child edges/ relationships, ci_unique_id being the "from" vertex and child_ci_unique_id being the "to" vertex. • If more than one child_ci_unique_id to be defined in single mapping, can be done using separator "&&".

Attribute Name	Mandatory	Attribute Type	Attribute Not Sent to DX OI /TAS	Description
semantic_child	No	N/A	Yes	Used internally by RESTMon. • The attribute that is used for the edge creation in attaching the proper relationship between the current vertex and its child vertex that is represented as ci_unique_id and child_ci_unique_id. • **Valid values:** null, contains • If more than one child_ci_unique_id is defined in a single mapping, semantic_child as well must be defined with matching number that is separated by &&. Example: "semantic_child": "contains"
create_edges_with_childs_parents_of_other_type	No	N/A (true/false)	Yes	Used internally by RESTMon. Creates edge with its child's parent vertex. Sometimes we are not able to get the relational data between the current vertex's child's parent and this flag can be used to create the edge between these vertices. Example: In the case of Dynatrace, we have relations ship available between Host → process, process → service and service → application from various topology APIs. And this flag can be used when we need a relationship between service and host.
child_parent_type	No	N/A	Yes	Used internally by RESTMon. Used in relation with the above attribute, "create_edges_with_childs_parents to represent the type of the child's parent.

Attribute Name	Mandatory	Attribute Type	Attribute Not Sent to DX OI /TAS	Description
condition	No	N/A	Yes	Used internally by RESTMon. Used to filter the topology data ingestion based on some condition. This value should derive to a Boolean value (true/false) and the data is ingested only if the value is "true".
entity_id	No	String	No	Required by RESTMon, if the parent-child relationship is maintained by this unique id and the hostname has to be derived using this. Refer to "Topology Mapping with both entity_id and entity_name details" for sample mapping.
entity_name	No	String		Required by RESTMon, if all the topology APIs do not have the entity name and instead have only entity_id and only one of the APIs has the entity_id and entity_name mapping. Refer to "Topology Mapping with both entity_id and entity_name details" for sample mapping.
hasentityidnamedetails	No	N/A	Yes	Used internally by RESTMon. Boolean indicating if the given mapping has both entity_id and entity_name details. If marked "yes", the entity_id and entity_name map is maintained locally. Refer to "Topology Mapping with both entity_id and entity_name details" for sample mapping.

Attribute Name	Mandatory	Attribute Type	Attribute Not Sent to DX OI /TAS	Description
deriveentityvaluesforfields	No	N/A	Yes	Used internally by RESTMon. If given "Yes", the selected topology mappings fields with entity_id are attached with related entity_name (from the map created based on hasentityidnamedetails) Refer to "Topology Mapping with deriveentityvaluesforfields, chil and other fields" for sample mapping.
delimiter	No	N/A	Yes	Used internally by RESTMon. • If the given field for deriving entity_name is having any internal separator to be considered. (Ex: name=Apps|<app_entity_id>|<agent_entity_id>, in this case delimiter="|" should be used). • Used in combining multiple entity ids for which entity name has to be derived. • Any value except "&&" can be used. Refer to "Topology Mapping with deriveentityvaluesforfields, chil and other fields" for sample mapping.
isDeltaTopology	No	N/A	Yes	Used internally by RESTMon. • To disable the default behavior of RESTMon in calculating the delta topology. • Used in case the incoming payload itself is having only delta data. Refer to "Topology Mapping with isDeltaTopology" for sample mapping.

Attribute Name	Mandatory	Attribute Type	Attribute Not Sent to DX OI /TAS	Description
startTime	No	Long(ms)	No	• Represents the start time of the topology entity (vertex/edge). • Defaults to current system time if not available in the schema mapping.

Topology Correlation Support

You can create a rule-based edge by defining the correlation metadata for the different DX OI monitored products. The data source that is integrated can create rules that will be executed on top of the topological data that is available in the DX OI platform from various related monitored inventories. The two types of correlation available are:

- Compaction
- Association

Compaction Rule Definition

The compaction rule is useful in compacting the vertex when two different vertices from two different sources depict the same object.

Sample compaction metadata definition:

```
{
    "xml_ns": "",
    "url": "hosts",
    "group": "Hosts",
    "layer": "CONNECTOR_METADATA",
    "attributes": {
        "oi": {
            "ci_unique_id": "DYNATRACE_HOST",
            "name": "Dynatrace Compaction Rule",
            "type": "SERVICE_UI_CONFIG",
            "compactionRules": [
                {
                    "sourceTasVertices": {
                        "attributeFilters": [
                            {
                                "attributeName": "product",
                                "attributeValues": [
                                    "Dynatrace"
                                ],
                                "operator": "IN"
                            },
                            {
                                "attributeName": "type",
                                "attributeValues": [
                                    "HOST"
                                ],
                                "operator": "IN"
                            }
                        ],
```

```
                        "layer": "CUSTOM"
                    },
                    "matchingAttributeNames": [
                        "ipAddresses",
                        "hostname"
                    ],
                    "matchingAttributeTypes": [
                        "IP_ADDRESS",
                        "ACN_HOST"
                    ]
                }
            ]
        }
    }
}
```

Note: The attributes that depict the compaction rules are defined under the **compactionRules** section. The following table lists the various attributes that you can define in the schema:

Attribute Name			Mandatory	Allowed Values	Description
layer			Yes	CONNECTOR_METADATA	The TAS layer to which the metadata to be ingested.
ci_unique_id			Yes	String	Unique identifier to identify the defined rule.
name			Yes	String	USed to describe the defined rule.
type			Yes	SERVICE_UI_CONFIG	The type of the TAS vertex.
sourceTasVertices - used to identify specific vertices to apply compaction.					
	attributeFilters		Yes. (Filter definition for product and the type of vertex to be considered for compaction are mandatory)		An array of possible attribute filters.
		attributeName	Yes	String	Name of the attribute Example: product, type
		attributeValues	Yes	Array of Strings	Value of the attribute Example: ["Dynatrace"]
		operator	Yes	String	The operator for the value to be compared to filter the vertices. Example: IN

Attribute Name		Mandatory	Allowed Values	Description
	layer	Yes	CUSTOM	The layer of the vertex where this compaction has to be applied, and it "CUSTOM" for RESTMon based products.
matchingAttributeNames		Yes	Array of Strings • The allowed values are any combinations from below. [ipAddresses, macAddresses, hostname/ FQDNHostname]	This defines the attributes whose value should be used for comparison with other similar vertices from other products.
matchingAttributeTypes		Yes	Array of Strings • The allowed values are any combinations from below. ["IP_ADDRESS", MAC_ADDRESS, "ACN_HOST"]	This defines the category/type of attributes to which the attribute mentioned in matchingAttributeName belongs to. This will help in identifying which attribute in other products can be used for comparing with this attribute in this product.

Association Rule Definition

The association rule is useful when two different vertices from one or more different sources depict different objects which relate in some manner to each other. In this case, an edge is created between those two vertices with the preconfigured semantic.

Sample metadata definition with Association Rules:

```
{
  "xml_ns": "",
  "url": "databases",
  "group": "Databases",
  "layer": "CONNECTOR_METADATA",
  "attributes": {
      "oi": {
          "ci_unique_id": "APMOSE_TO_APPD_AGENT1",
          "name": "APM Openshift to App Dynamics AGENT association",
          "type": "SERVICE_UI_CONFIG",
          "associationRules": [
              {
                  "sourceTasVertices": {
```

```
                    "attributeFilters": [
                        {
                            "attributeName": "product",
                            "attributeValues": [
                                "APM"
                            ],
                            "operator": "IN"
                        },
                        {
                            "attributeName": "type",
                            "attributeValues": [
                                "OPENSHIFT"
                            ],
                            "operator": "IN"
                        }
                    ],
                    "layer": "INFRASTRUCTURE",
                    "matchingAttributeNames": [
                        "ose_pod_name"
                    ]
                },
                "targetTasVertices": {
                    "attributeFilters": [
                        {
                            "attributeName": "product",
                            "attributeValues": [
                                "AppDynamics"
                            ],
                            "operator": "IN"
                        },
                        {
                            "attributeName": "type",
                            "attributeValues": [
                                "AGENT"
                            ],
                            "operator": "IN"
                        }
                    ],
                    "layer": "CUSTOM",
                    "matchingAttributeNames": [
                        "hostname"
                    ]
                },
                "semantic": "contains"
            }
        ]
    }
  }
}
```

Note: Define the attributes depicting the association rules under the **associationRules** section. The following table lists the various attributes that you can define in the schema:

Attribute Name			Mandatory	Allowed Values	Description
layer			Yes	CONNECTOR_METADATA	The TAS layer to which the metadata to be ingested.
ci_unique_id			Yes	String	Unique identifier to identify the defined rule.
name			Yes	String	USed to describe the defined rule.
type			Yes	SERVICE_UI_CONFIG	The type of the TAS vertex.
sourceTasVertices - used to identify specific vertices to apply association.					
	attributeFilters		Yes. (Filter definition for product and the type of vertex to be considered for compaction are mandatory)		An array of possible attribute filters
		attributeName	Yes	String	Name of the attribute Example: product, type
		attributeValues	Yes	Array of Strings	Value of the attribute Example: ["APM"]
		operator	Yes	String	The operator for the value to be compared to filter the vertices. Example: IN
	layer		Yes	String	The layer of the vertex where this association has to be applied. Can be CUSTOM/ INFRASTRUCTURE_UIM/ INFRASTRUCTURE etc based on the TAS layer to which the data is ingested.
matchingAttributeNames			Yes	Array of Strings	The name of the actual attribute which is used for comparing and associating across source and target.
targetTasVertices- used to identify specific vertices to apply association.					

Attribute Name			Mandatory	Allowed Values	Description
	attributeFilters		Yes. (Filter definition for product and the type of vertex to be considered for the association are mandatory)		An array of possible attribute filters
		attributeName	Yes	String	Name of the attribute Example: product, type
		attributeValues	Yes	Array of Strings	Value of the attribute Example: ["AppDynamics"]
		operator	Yes	String	The operator for the value to be compared to filter the vertices. Example: IN
	layer		Yes	String	The layer of the vertex where this association has to be applied. Can be CUSTOM/ INFRASTRUCTURE_UIM/ INFRASTRUCTURE etc based on the TAS layer to which the data is ingested.
matchingAttributeNames			Yes	Array of Strings	The name of the actual attribute which is used for comparing and associating across source and target.
semantic			No	String	The type of association to be created between the source and target vertices. Example: contains

Configure Correlation of the Third-Party Ingested Data

To correlate the ingested third-party data with the Broadcom products, you must set the normalized attributes and the correlation attributes at the tenant level in the metadata vertex for a tenant. Once you configure these attributes, the Topology Processor correlates the configured attributes in the tenant.

You can define the mapping rules in the attributeMappingRules section and the correlation attributes in the correlationAttributes section. The following snippet is a sample of the metadata vertex:

```
{
    "externalId: "CONNECTOR_METADATA:ABC",
    "attributes: {
      "name: "Connector for product ABC",
```

```
        "type: "SERVICE_UI_CONFIG",
        "attributeMappingRules: [{
          "products: ["ABC"],
          "rules: [{
            "toAttrName: "dx_ip_address",
            "toValueType: "scalar",
            "fromAttrNames: ["primaryIp"],
            "fromDataValidator: "IP",
            "excludedValues: ["127.0.0.1", ::1"]
          }, {
            "toAttrName: "dx_hostname",
            "toValueType: "scalar",
            "toValueFormat: "lowercase",
            "fromAttrNames: ["dnsName", "host"],
            "excludedValues: [""],
            "conditionalRules: [{
              "condition: {
               "attrName: "type",
               "attrValue: "vSwitch",
               "ignoreCase: false,
               "isRegex: false
            },
            "fromAttrNames: ["dnsName", "vmName"]
          }]
        }],
        "correlationAttributes: ["dx_ip_address","dx_hostname"]
      }
  }
```

APIs

Representational State Transfer (REST) APIs are service endpoints that support sets of HTTP operations (methods), which provide create, retrieve, update, or delete access to the RESTMon resources. This topic provides the following information:

Call RESTMon APIs Using Swagger

Once you have deployed RESTMon, you can call the RESTMon APIs using Swagger. By default, the REST APIs are accessible at:

```
https://localhost:8443/restmon/api/swagger-ui/
```

You can use HTTPS by configuring the SSL properties in the *application.properties* file.

Follow these steps:

1. Generate a self-signed certificate or use an existing SSL certificate. We recommend you to use the PKCS12 format for the certificate.
2. Configure the SSL properties in the *application.properties* file:
   a. Navigate to the *<RESTMon_installation_directory>/config* folder.
   b. Open the *application.properties* file.
   c. Uncomment the SSL section by removing the **#** character and provide the values for the SSL-related fields:

```
ssl
# The format used for the keystore. It could be set to JKS in case it is a JKS file
server.ssl.key-store-type=PKCS12
# The path to the keystore containing the certificate
```

```
server.ssl.key-store=config/restmon.keystore
# The password used to generate the certificate
server.ssl.key-store-password=ENC(otWZvrgx/RKz6TBQPRni7g==)
# The alias mapped to the certificate
server.ssl.key-alias=restmonhttpsaug2019
# Flag to notify whether the Password is encrypted or not
isKeyStorePasswordEncrypted=true
```

d. Save the file.

3. Start RESTMon.

```
Windows:
./run.bat

Linux:
./run.sh
```

4. Navigate to the Swagger UI.

```
https://<restmon-hostname>:<https-port>/restmon/api/swagger-ui/
```

Components of REST APIs

A REST API request/response pair can be separated into the following components:

- Request URI
- HTTP Method
- HTTP Response Message Header

Request URI

The request URI has the following format:

```
{URI-scheme}://{URI-host}/{resource-path}?{query-string}
```

- **URI scheme**: Indicates the protocol used to transmit the request. For example, *http* or *https*.
- **URI host**: Specifies the domain name or IP address of the server where the REST service endpoint is hosted. For example, *example.broadcom:8080*.
- **Resource path**: Specifies the resource or resource collection, which may include multiple segments used by the service in determining the selection of those resources. For example, *restmon/api/v1/uploadSchema//{schemaName}.*
- **Query string** (Optional): Provides additional simple parameters, such as the API version or resource selection criteria. For example, */v1/logs?profileName={profileName}&schemaName={schemaName}.*

HTTP Method

A required HTTP method (also known as an operation or verb), which tells the service what type of operation you are requesting. RESTMon APIs support GET, PUT, POST, and DELETE methods.

HTTP Response Message Header

An HTTP status code, ranging from 2xx success codes to 4xx or 5xx error codes.

APIs Reference

The RESTMon APIs provide CRUD operations for admin, profiles, and schema:

- APIs for Admin
- APIs for a Profile
- APIs for Schema

APIs for Admin

The following table lists the different RESTMon APIs that are available for administration:

Operation	API
POST /v1/jsonpaths	Get the JSON paths
POST /v1/logs	Stream Data to RESTMon
GET /v1/version	Get the RESTMon version details
POST/restmon/v2/config/alarmJobConfig	Manage Alarm Reconciliation Job
GET/restmon/v2/config/alarmJobConfig	Get Alarm Reconciliation Job Status

Get JsonPaths

This API returns the JSON path definitions related to the provided JSON data.

Name	Description
URL	http://<host-name>:8080/restmon/api/v1/jsonpaths https://<host-name>:8443/restmon/api/v1/jsonpaths
Method	POST
Response Code	200 OK
CURL Command	Create a file with the required JSON payload. For example, create testpayload.json with the given payload. Execute the following curl command to use this payload. `curl -k --user <username>:<password> -X POST --header 'Content-Type: application/json;charset=UTF-8' --data @testpayload.json https://<host-name>:8443/restmon/api/v1/jsonpaths`

Body:

```
{ "store": {
    "book": [
      { "category": "reference",
        "author": "Nigel Rees",
        "title": "Sayings of the Century",
        "price": 8.95
      },
      { "category": "fiction",
        "author": "J. R. R. Tolkien",
        "title": "The Lord of the Rings",
        "isbn": "0-395-19395-8",
        "price": 22.99
      }
    ],
    "bicycle": {
      "color": "red",
      "price": 19.95
    }
```

```
  }
}
```

Sample Response:


```
{
  "jsonpaths": [
    {
      "jsonpath": "$['store']"
    },
    {
      "jsonpath": "$['store']['book']"
    },
    {
      "jsonpath": "$['store']['bicycle']"
    },
    {
      "jsonpath": "$['store']['book'][0]"
    },
    {
      "jsonpath": "$['store']['book'][1]"
    },
    {
      "jsonpath": "$['store']['book'][0]['category']"
    },
    {
      "jsonpath": "$['store']['book'][0]['author']"
    },
    {
      "jsonpath": "$['store']['book'][0]['title']"
    },
    {
      "jsonpath": "$['store']['book'][0]['price']"
    },
    {
      "jsonpath": "$['store']['book'][1]['category']"
    },
    {
      "jsonpath": "$['store']['book'][1]['author']"
    },
    {
      "jsonpath": "$['store']['book'][1]['title']"
    },
    {
      "jsonpath": "$['store']['book'][1]['isbn']"
    },
    {
      "jsonpath": "$['store']['book'][1]['price']"
    },
    {
      "jsonpath": "$['store']['bicycle']['color']"
    },
    {
      "jsonpath": "$['store']['bicycle']['price']"
```

```
      }
    ]
  }
```

Stream Data to RESTMon

You can use this API to stream data to RESTMon. Before you stream the data, ensure that the required schema and profile are created.

Name	Description
URL	http://<host-name>:8080/restmon/api/v1/logs?profileName={profileName}&schemaName={schemaName} https://<host-name>:8443/restmon/api/v1/logs?profileName={profileName}&schemaName={schemaName}
Method	POST
Authentication	Basic
Response	200 OK
Curl Command	Create a file (For example, streampayload.json) with the required JSON payload and execute the following curl command to use this payload: `curl -k --user <user-name>:<password> -X POST --header 'Content-Type: application/json;charset=UTF-8' --data @streampayload.json https://<host-name>:8443/restmon/api/v1/logs?profileName=googlecloudmonitoring&schemaName=googlecloudmonitoring`

Sample Request:

```
http://localhost:8080/restmon/api/v1/logs?profileName=googlecloudmonitoring&schemaName=googlecloudmonitoring
```

Body:


```
{
"summaryFailedMessages": "7",
"summaryActiveDomains": "3",
"summaryDeliveryRate": "65387",
"summaryPendingDNSQueries": "0",
"summaryTempFailedMessages": "3617",
"summaryTotalMessageCount": "4",
"summaryDNSMXQueries": "9",
"summaryRejectedMessages": "6",
"summaryInboundConcurrency": "9",
"summaryDNSAAAAQueries": "1",
"summaryOutboundConcurrency": "2",
"summaryActiveQueueSize": "8",
"summaryDNSAQueries": "0",
"summaryReceivedMessages": "940",
"summarySuccessfulDeliveries": "2403",
"summaryUptime": "214337",
"summaryReceptionRate": "644724",
"summaryStatisticsLastReset": "29503430",
"summaryDNSQueryRate": "115966",
"summaryDelayedQueueSize": "0",
"host": "host9.com"
```

```
}
```

Get the RESTMon Version Details

You can use this API to get the RESTMon version details.

Name	Description
URL	http://<host-name>:8080/restmon/api/v1/version https://<host-name>:8443/restmon/api/v1/version
Method	GET
Curl Command	`curl -X GET --header 'Accept: text/plain' "https://<host-name>:8443/restmon/api/v1/version"`

Sample Request:

```
https://<host-name>:8443/restmon/api/v1/version
```

Sample Response:

```
2.0 2021-01-25 11:13
```

Manage Alarm Reconciliation Job

You can enable or disable the alarm reconciliation job for a tenant using the Alarm Reconciliation Job API.

Name	Description
URL	http://<host-name>:8080/restmon/v2/config/alarmJobConfig https://<host-name>:8443/restmon/v2/config/alarmJobConfig
Method	POST
Authentication	Bearer `<tenant> token`
Body	true/false

Get Alarm Reconciliation Job Status

You can get the alarm reconciliation job status for a tenant using this API.

Name	Description
URL	http://<host-name>:8080/restmon/v2/config/alarmJobConfig https://<host-name>:8443/restmon/v2/config/alarmJobConfig
Method	GET
Authentication	Bearer `<tenant> token`

Admin APIs

none

```
    {
"swagger": "2.0",
"info": {
```

```
    "title": "RESTMon API Documentation"
  },
  "host": "localhost:8080",
  "basePath": "/restmon/api",
  "tags": [
    {
      "name": "profiles",
      "description": "CRUD operations for restmon profiles"
    },
    {
      "name": "admin",
      "description": "CRUD operations for administration"
    },
    {
      "name": "schema",
      "description": "CRUD operations for restmon Schema"
    }
  ],
  "paths": {
    "/v1/jsonpaths": {
      "post": {
        "tags": [
          "admin"
        ],
        "summary": "Get JsonPaths",
        "operationId": "getJsonPathUsingPOST",
        "consumes": [
          "application/json"
        ],
        "produces": [
          "*/*"
        ],
        "parameters": [
          {
            "in": "body",
            "name": "jsonText",
            "description": "jsonText",
            "required": true,
            "schema": {
              "type": "string"
            },
            "example": "DX-RESTmon/profile/solarwinds_profile.json"
          }
        ],
        "responses": {
          "200": {
            "description": "OK",
            "schema": {
              "type": "object"
            }
          },
          "201": {
            "description": "Created"
```

```
      },
      "401": {
        "description": "Unauthorized"
      },
      "403": {
        "description": "Forbidden"
      },
      "404": {
        "description": "Not Found"
      }
    }
  }
},
"/v1/logs": {
  "post": {
    "tags": [
      "admin"
    ],
    "summary": "readLogs",
    "operationId": "readLogsUsingPOST",
    "consumes": [
      "application/json"
    ],
    "produces": [
      "*/*"
    ],
    "parameters": [
      {
        "name": "profileName",
        "in": "query",
        "description": "profileName",
        "required": false,
        "type": "string"
      },
      {
        "name": "schemaName",
        "in": "query",
        "description": "schemaName",
        "required": false,
        "type": "string"
      },
      {
        "in": "body",
        "name": "data",
        "description": "data",
        "required": true,
        "schema": {
          "$ref": "#/definitions/JsonNode"

        }
      }
    ],
    "responses": {
```

```
      "200": {
        "description": "OK",
        "schema": {
          "type": "object"
        }
      },
      "201": {
        "description": "Created"
      },
      "401": {
        "description": "Unauthorized"
      },
      "403": {
        "description": "Forbidden"
      },
      "404": {
        "description": "Not Found"
      }
    }
  }
},
"/v1/oiconnection": {
  "get": {
    "tags": [
      "profiles"
    ],
    "summary": "Get OI connection details",
    "operationId": "getOIDetailsUsingGET",
    "consumes": [
      "application/json"
    ],
    "produces": [
      "application/json;charset=UTF-8"
    ],
    "responses": {
      "200": {
        "description": "OK",
        "schema": {
          "type": "object"
        }
      },
      "401": {
        "description": "Unauthorized"
      },
      "403": {
        "description": "Forbidden"
      },
      "404": {
        "description": "Not Found"
      }
    }
  },
  "put": {
```

```
      "tags": [
        "profiles"
      ],
      "summary": "Update OI connection details",
      "operationId": "updateOIConnetionDetailsUsingPUT",
      "consumes": [
        "application/json;charset=UTF-8"
      ],
      "produces": [
        "application/json;charset=UTF-8"
      ],
      "parameters": [
        {
          "in": "body",
          "name": "inputBody",
          "description": "inputBody",
          "required": true,
          "schema": {
            "$ref": "#/definitions/ObjectNode"
          }
        }
      ],
      "responses": {
        "200": {
          "description": "OK",
          "schema": {
            "type": "object"
          }
        },
        "201": {
          "description": "Created"
        },
        "401": {
          "description": "Unauthorized"
        },
        "403": {
          "description": "Forbidden"
        },
        "404": {
          "description": "Not Found"
        }
      }
    }
  },
  "/v1/oiconnection/{oiconnectionName}": {
    "delete": {
      "tags": [
        "profiles"
      ],
      "summary": "Delete OI connection details",
      "operationId": "deleteOIDetailsUsingDELETE",
      "consumes": [
        "application/json"
```

```
      ],
      "produces": [
        "application/json;charset=UTF-8"
      ],
      "parameters": [
        {
          "name": "oiconnectionName",
          "in": "path",
          "description": "oiconnectionName",
          "required": true,
          "type": "string"
        }
      ],
      "responses": {
        "200": {
          "description": "OK",
          "schema": {
            "type": "object"
          }
        },
        "204": {
          "description": "No Content"
        },
        "401": {
          "description": "Unauthorized"
        },
        "403": {
          "description": "Forbidden"
        }
      }
    }
  },
  "/v1/profiles": {
    "get": {
      "tags": [
        "profiles"
      ],
      "summary": "Get all profiles",
      "operationId": "getProfileListUsingGET",
      "consumes": [
        "application/json"
      ],
      "produces": [
        "application/json;charset=UTF-8"
      ],
      "responses": {
        "200": {
          "description": "OK",
          "schema": {
            "type": "object"
          }
        },
        "401": {
```

```
        "description": "Unauthorized"
      },
      "403": {
        "description": "Forbidden"
      },
      "404": {
        "description": "Not Found"
      }
    }
  },
  "post": {
    "tags": [
      "profiles"
    ],
    "summary": "Add a profile",
    "operationId": "addProfileUsingPOST",
    "consumes": [
      "application/json"
    ],
    "produces": [
      "application/json"
    ],
    "parameters": [
      {
        "in": "body",
        "name": "inputBody",
        "description": "inputBody",
        "required": true,
        "schema": {
          "$ref": "#/definitions/ObjectNode"
        }
      }
    ],
    "responses": {
      "200": {
        "description": "OK",
        "schema": {
          "type": "object"
        }
      },
      "201": {
        "description": "Created"
      },
      "401": {
        "description": "Unauthorized"
      },
      "403": {
        "description": "Forbidden"
      },
      "404": {
        "description": "Not Found"
      }
    }
```

```
      }
    },
    "/v1/profiles/{schemaName}/{profileName}": {
      "get": {
        "tags": [
          "profiles"
        ],
        "summary": "Get a profile by profileName and schemaName",
        "operationId": "getProfileDetailsUsingGET",
        "consumes": [
          "application/json"
        ],
        "produces": [
          "application/json;charset=UTF-8"
        ],
        "parameters": [
          {
            "name": "schemaName",
            "in": "path",
            "description": "schemaName",
            "required": true,
            "type": "string"
          },
          {
            "name": "profileName",
            "in": "path",
            "description": "profileName",
            "required": true,
            "type": "string"
          }
        ],
        "responses": {
          "200": {
            "description": "OK",
            "schema": {
              "type": "object"
            }
          },
          "401": {
            "description": "Unauthorized"
          },
          "403": {
            "description": "Forbidden"
          },
          "404": {
            "description": "Not Found"
          }
        }
      },
      "put": {
        "tags": [
          "profiles"
        ],
```

```
"summary": "Update a profile",
"operationId": "updateProfileDetailsUsingPUT",
"consumes": [
  "application/json;charset=UTF-8"
],
"produces": [
  "application/json;charset=UTF-8"
],
"parameters": [
  {
    "in": "body",
    "name": "inputBody",
    "description": "inputBody",
    "required": true,
    "schema": {
      "$ref": "#/definitions/ObjectNode"
    }
  },
  {
    "name": "schemaName",
    "in": "path",
    "description": "schemaName",
    "required": true,
    "type": "string"
  },
  {
    "name": "profileName",
    "in": "path",
    "description": "profileName",
    "required": true,
    "type": "string"
  }
],
"responses": {
  "200": {
    "description": "OK",
    "schema": {
      "type": "object"
    }
  },
  "201": {
    "description": "Created"
  },
  "401": {
    "description": "Unauthorized"
  },
  "403": {
    "description": "Forbidden"
  },
  "404": {
    "description": "Not Found"
  }
}
```

```
    },
    "delete": {
      "tags": [
        "profiles"
      ],
      "summary": "Delete a profile",
      "operationId": "deleteProfileUsingDELETE",
      "consumes": [
        "application/json"
      ],
      "produces": [
        "application/json"
      ],
      "parameters": [
        {
          "name": "schemaName",
          "in": "path",
          "description": "schemaName",
          "required": true,
          "type": "string"
        },
        {
          "name": "profileName",
          "in": "path",
          "description": "profileName",
          "required": true,
          "type": "string"
        }
      ],
      "responses": {
        "200": {
          "description": "OK",
          "schema": {
            "type": "object"
          }
        },
        "204": {
          "description": "No Content"
        },
        "401": {
          "description": "Unauthorized"
        },
        "403": {
          "description": "Forbidden"
        }
      }
    }
  },
  "/v1/schema/getAllSchemaName": {
    "get": {
      "tags": [
        "schema"
      ],
```

```
      "summary": "Get All schema ",
      "operationId": "getAllSchemaUsingGET",
      "consumes": [
        "application/json"
      ],
      "produces": [
        "application/json"
      ],
      "responses": {
        "200": {
          "description": "OK",
          "schema": {
            "type": "object"
          }
        },
        "401": {
          "description": "Unauthorized"
        },
        "403": {
          "description": "Forbidden"
        },
        "404": {
          "description": "Not Found"
        }
      }
    }
  },
  "/v1/schema/getSchemaDetails/{schemaName}": {
    "get": {
      "tags": [
        "schema"
      ],
      "summary": "Get schema details",
      "operationId": "getSchemaDetailsUsingGET",
      "consumes": [
        "application/json"
      ],
      "produces": [
        "application/json"
      ],
      "parameters": [
        {
          "name": "schemaName",
          "in": "path",
          "description": "schemaName",
          "required": true,
          "type": "string"
        }
      ],
      "responses": {
        "200": {
          "description": "OK",
          "schema": {
```

```
            "type": "object"
          }
        },
        "401": {
          "description": "Unauthorized"
        },
        "403": {
          "description": "Forbidden"
        },
        "404": {
          "description": "Not Found"
        }
      }
    }
  },
  "/v1/schema/{schemaName}": {
    "post": {
      "tags": [
        "schema"
      ],
      "summary": "Upload schema",
      "operationId": "uploadSchemaUsingPOST",
      "consumes": [
        "application/json"
      ],
      "produces": [
        "application/json"
      ],
      "parameters": [
        {
          "in": "body",
          "name": "inp",
          "description": "inp",
          "required": true,
          "schema": {
            "$ref": "#/definitions/ObjectNode"
          }
        },
        {
          "name": "schemaName",
          "in": "path",
          "description": "schemaName",
          "required": true,
          "type": "string"
        }
      ],
      "responses": {
        "200": {
          "description": "OK",
          "schema": {
            "type": "object"
          }
        },
```

```
      "201": {
        "description": "Created"
      },
      "401": {
        "description": "Unauthorized"
      },
      "403": {
        "description": "Forbidden"
      },
      "404": {
        "description": "Not Found"
      }
    }
  },
  "delete": {
    "tags": [
      "schema"
    ],
    "summary": "Delete schema",
    "operationId": "deleteSchemaUsingDELETE",
    "consumes": [
      "application/json"
    ],
    "produces": [
      "application/json"
    ],
    "parameters": [
      {
        "name": "schemaName",
        "in": "path",
        "description": "schemaName",
        "required": true,
        "type": "string"
      }
    ],
    "responses": {
      "200": {
        "description": "OK",
        "schema": {
          "type": "object"
        }
      },
      "204": {
        "description": "No Content"
      },
      "401": {
        "description": "Unauthorized"
      },
      "403": {
        "description": "Forbidden"
      }
    }
  }
```

```
    },
    "/v1/schema/{schemaName}/{fieldName}": {
      "get": {
        "tags": [
          "schema"
        ],
        "summary": "Get list of all urls,inventory,topology,metric,alarm,calculated_methods or
calculated_metrics",
        "operationId": "getDetailsByFieldNameUsingGET",
        "consumes": [
          "application/json"
        ],
        "produces": [
          "application/json;charset=UTF-8"
        ],
        "parameters": [
          {
            "name": "schemaName",
            "in": "path",
            "description": "schemaName",
            "required": true,
            "type": "string"
          },
          {
            "name": "fieldName",
            "in": "path",
            "description": "fieldName",
            "required": true,
            "type": "string",
            "enum": [
              "urls",
              "metrics",
              "alarms",
              "inventory",
              "topology",
              "groups",
              "calculated_metrics",
              "calculated_methods"
            ]
          }
        ],
        "responses": {
          "200": {
            "description": "OK",
            "schema": {
              "type": "object"
            }
          },
          "401": {
            "description": "Unauthorized"
          },
          "403": {
            "description": "Forbidden"
```

```
      },
      "404": {
        "description": "Not Found"
      }
    }
  },
  "post": {
    "tags": [
      "schema"
    ],
    "summary": "Add urls,inventory,metric,alarm,calculated_methods or calculated_metrics",
    "operationId": "addDetailsUsingPOST",
    "consumes": [
      "application/json;charset=UTF-8"
    ],
    "produces": [
      "application/json;charset=UTF-8"
    ],
    "parameters": [
      {
        "in": "body",
        "name": "inpField",
        "description": "inpField",
        "required": true,
        "schema": {
          "$ref": "#/definitions/ObjectNode"
        }
      },
      {
        "name": "schemaName",
        "in": "path",
        "description": "schemaName",
        "required": true,
        "type": "string"
      },
      {
        "name": "fieldName",
        "in": "path",
        "description": "fieldName",
        "required": true,
        "type": "string",
        "enum": [
          "urls",
          "metrics",
          "alarms",
          "inventory",
          "topology",
          "groups",
          "calculated_metrics",
          "calculated_methods"
        ]
      }
    ],
```

```
        "responses": {
          "200": {
            "description": "OK",
            "schema": {
              "type": "object"
            }
          },
          "201": {
            "description": "Created"
          },
          "401": {
            "description": "Unauthorized"
          },
          "403": {
            "description": "Forbidden"
          },
          "404": {
            "description": "Not Found"
          }
        }
      }
    },
    "/v1/schema/{schemaName}/{fieldName}/{fieldId}": {
      "get": {
        "tags": [
          "schema"
        ],
        "summary": "Get details of  urls,inventory,topology,metric,alarm,calculated_methods or
calculated_metrics by fieldId",
        "operationId": "getDetailsByfieldIdUsingGET",
        "consumes": [
          "application/json"
        ],
        "produces": [
          "application/json;charset=UTF-8"
        ],
        "parameters": [
          {
            "name": "schemaName",
            "in": "path",
            "description": "schemaName",
            "required": true,
            "type": "string"
          },
          {
            "name": "fieldName",
            "in": "path",
            "description": "fieldName",
            "required": true,
            "type": "string",
            "enum": [
              "urls",
              "metrics",
```

```
          "alarms",
          "inventory",
          "topology",
          "groups",
          "calculated_metrics",
          "calculated_methods"
        ]
      },
      {
        "name": "fieldId",
        "in": "path",
        "description": "fieldId",
        "required": true,
        "type": "string"
      }
    ],
    "responses": {
      "200": {
        "description": "OK",
        "schema": {
          "type": "object"
        }
      },
      "401": {
        "description": "Unauthorized"
      },
      "403": {
        "description": "Forbidden"
      },
      "404": {
        "description": "Not Found"
      }
    }
  },
  "put": {
    "tags": [
      "schema"
    ],
    "summary": "Update  urls,inventory,metric,alarm,calculated_methods or calculated_metrics  by fieldId",
    "operationId": "updateDetailsByIdUsingPUT",
    "consumes": [
      "application/json;charset=UTF-8"
    ],
    "produces": [
      "application/json;charset=UTF-8"
    ],
    "parameters": [
      {
        "in": "body",
        "name": "inp",
        "description": "inp",
        "required": true,
        "schema": {
```

```
          "$ref": "#/definitions/ObjectNode"
        }
      },
      {
        "name": "schemaName",
        "in": "path",
        "description": "schemaName",
        "required": true,
        "type": "string"
      },
      {
        "name": "fieldName",
        "in": "path",
        "description": "fieldName",
        "required": true,
        "type": "string",
        "enum": [
          "urls",
          "metrics",
          "alarms",
          "inventory",
          "topology",
          "groups",
          "calculated_metrics",
          "calculated_methods"
        ]
      },
      {
        "name": "fieldId",
        "in": "path",
        "description": "fieldId",
        "required": true,
        "type": "string"
      }
    ],
    "responses": {
      "200": {
        "description": "OK",
        "schema": {
          "type": "object"
        }
      },
      "201": {
        "description": "Created"
      },
      "401": {
        "description": "Unauthorized"
      },
      "403": {
        "description": "Forbidden"
      },
      "404": {
        "description": "Not Found"
```

```
        }
      }
    },
    "delete": {
      "tags": [
        "schema"
      ],
      "summary": "Delete urls,inventory,metric,alarm,calculated_methods or calculated_metrics by fieldId",
      "operationId": "deleteDetailsByIdUsingDELETE",
      "consumes": [
        "application/json"
      ],
      "produces": [
        "application/json;charset=UTF-8"
      ],
      "parameters": [
        {
          "name": "schemaName",
          "in": "path",
          "description": "schemaName",
          "required": true,
          "type": "string"
        },
        {
          "name": "fieldName",
          "in": "path",
          "description": "fieldName",
          "required": true,
          "type": "string",
          "enum": [
            "urls",
            "metrics",
            "alarms",
            "inventory",
            "topology",
            "groups",
            "calculated_metrics",
            "calculated_methods"
          ]
        },
        {
          "name": "fieldId",
          "in": "path",
          "description": "fieldId",
          "required": true,
          "type": "string"
        }
      ],
      "responses": {
        "200": {
          "description": "OK",
          "schema": {
            "type": "object"
```

```
          }
        },
        "204": {
          "description": "No Content"
        },
        "401": {
          "description": "Unauthorized"
        },
        "403": {
          "description": "Forbidden"
        }
      }
    }
  },
  "/v1/validatedata/{schemaName}/{profileName}": {
    "get": {
      "tags": [
        "profiles"
      ],
      "summary": "Validate Data Collection",
      "operationId": "validateDataCollectionUsingGET",
      "consumes": [
        "application/json"
      ],
      "produces": [
        "application/json;charset=UTF-8"
      ],
      "parameters": [
        {
          "name": "schemaName",
          "in": "path",
          "description": "schemaName",
          "required": true,
          "type": "string"
        },
        {
          "name": "profileName",
          "in": "path",
          "description": "profileName",
          "required": true,
          "type": "string"
        }
      ],
      "responses": {
        "200": {
          "description": "OK",
          "schema": {
            "type": "object"
          }
        },
        "401": {
          "description": "Unauthorized"
        },
```

```
          "403": {
            "description": "Forbidden"
          },
          "404": {
            "description": "Not Found"
          }
        }
      }
    },
    "/v1/version": {
      "get": {
        "tags": [
          "admin"
        ],
        "summary": "Get Restmon version details",
        "operationId": "getVersionUsingGET",
        "consumes": [
          "application/json"
        ],
        "produces": [
          "*/*"
        ],
        "responses": {
          "200": {
            "description": "OK",
            "schema": {
              "type": "object"
            }
          },
          "401": {
            "description": "Unauthorized"
          },
          "403": {
            "description": "Forbidden"
          },
          "404": {
            "description": "Not Found"
          }
        }
      }
    }
  },
  "definitions": {
    "JsonNode": {
      "type": "object"
    },
    "ObjectNode": {
      "type": "object"
    }
  }
}
```

APIs for a Profile

This section provides information about the following APIs:

OI Connection Operations

The following table lists the different APIs that are available for OI operations:

API	Operation
GET /v1/oiconnection	Get the OI Connection Details
PUT /v1/oiconnection	Update the OI Connection Details
DELETE /v1/oiconnection/{oiconnectionName}	Delete the OI Connection Details
GET /v1/oisettings	Get the OI Setting Details
PUT /v1/oisettings	Update the OI Setting Details

GET the OI Connection Details

The following table provides the details that are required to get the OI connection details:

Name	Description
URL	http://<hostname>:8080/restmon/api/v1/oiconnection https://<hostname>:8443/restmon/api/v1/oiconnection
Method	GET
Authorization	Basic
Response Code	200 (Success)
CURL Command	`curl -k --user <username>:<password> -X GET "https://<hostname>:8443/restmon/api/v1/oiconnection"`

Sample Response:


```
{
    "oiconnectalarmsdetails": {
        "type": "https",
        "hostname": "<hostname>.broadcom.com",
        "port": 443,
        "maxnoofdocuments": "1000",
        "path": "/ingestion",
        "oi_pool_size": 4,
        "product_id": "ao",
        "doc_type_id": "itoa_alarms_custom",
        "doc_type_version": "1"
    },
    "oiconnecttopologydetails": {
        "type": "https",
        "hostname": "<hostname>.broadcom.com",
        "token": "Yy7cZDP/4dHgIw8sREyMXEF2L+7+OzLjvCENpR3996l3KAmKEO8K0xTvdj598O2DA4jP6ipabTLwpeo/
SQmx5cZmzLooBVihb0pdQsThEgrV2lqs93agD4S7PrqDxnr5SG7FUgaET+mvPh3bntuezbOCeuWGuO1s2n9Bgmdtz/
nnH95vqbHF0PTqSqkY3934aasPPx0LHLD/Odt/WNsZj0JasVUd3cRP+QI/Y/3SZaQPfT6bpqAJgl/i/Gfn08/
Nb07ocz7IMPzTvzniTvDid48BlUlH+U2uGZaPKFfLIbY=",
        "port": 443,
```

```
        "maxnoofdocuments": "1000",
        "path": "/tas/graph/store",
        "oi_pool_size": 4,
        "tastenantpath": "/tenants/tenants"
    },
    "oiconnectmetricsnassdetails": {
        "nass_type": "https",
        "nass_hostname": "<hostname>.broadcom.com",
        "nass_port": 443,
        "nass_token": "Yy7cZDP/4dHgIw8sREyMXEF2L+7+OzLjvCENpR3996l3KAmKEO8K0xTvdj598O2DA4jP6ipabTLwpeo/
SQmx5cZmzLooBVihb0pdQsThEgrV2lqs93agD4S7PrqDxnr5SG7FUgaET+mvPh3bntuezbOCeuWGuOls2n9Bgmdtz/
nnH95vqbHF0PTqSqkY3934aasPPx0LHLD/Odt/WNsZj0JasVUd3cRP+QI/Y/3SZaQPfT6bpqAJgl/i/Gfn08/
Nb07ocz7IMPzTvzniTvDid48BlUlH+U2uGZaPKFfLIbY=",
        "maxnoofdocuments": "5000",
        "storage_path": "/nass/metricValue/store",
        "registry_path": "/metadata/registerMetric",
        "oi_pool_size": 4
    },
    "oiconnectchangeeventdetails": {
        "type": "https",
        "hostname": "<hostname>.broadcom.com",
        "port": 443,
        "maxnoofdocuments": "1000",
        "path": "/ingestion",
        "oi_pool_size": 4,
        "product_id": "ao",
        "doc_type_id": "itoa_events_change_custom",
        "doc_type_version": "1"
    }
}
```


Update the OI Connection Details

The following table provides the details that are required to update the OI connection details:

Name	Description
URL	http://<hostname>:8080/restmon/api/v1/oiconnection https://<hostname>:8443/restmon/api/v1/oiconnection
Method	PUT
Authorization	Basic
Response Code	• 200 - Success • 404 - Field/parameter validation failure • 401 - Authentication failure

Name	Description
CURL Command	curl -k --user <username>:<password> -X PUT --header 'Content-Type: application/json;charset=UTF-8' --data '{"oiconnectmetricsnassdetails": {"nass_type": "https","nass_hostname": "<hostname>.broadcom.com","nass_port": 443,"nass_token": "Yy7cZDP/4dHgIw8sREyMXEF2L+7+OzLjvCENpR3996l3KAmKEO8K0xTvdj598O2DA4jP6ipabTLwpeo/SQmx5cZmzLooBVihb0pdQsThEgrV2lqs93agD4S7PrqDxnr5SG7FUgaET+mvPh3bntuezbOCeuWGuO1s2n9Bgmdtz/nnH95vqbHF0PTqSqkY3934aasPPx0LHLD/Odt/WNsZj0JasVUd3cRP+QI/Y/3SZaQPfT6bpqAJgl/i/Gfn08/Nb07ocz7IMPzTvzniTvDid48BlUlH+U2uGZaPKFfLIbY=","maxnoofdocuments": "2000","storage_path": "/nass/metricValue/store","registry_path": "/metadata/registerMetric","oi_pool_size": 4}}' "https://<hostname>:8443/restmon/api/v1/oiconnection"

Sample Body 1:


```
{
    "oiconnectmetricsnassdetails": {
        "nass_type": "https",
        "nass_hostname": "<hostname>.broadcom.com",
        "nass_port": 443,
        "nass_token": "Yy7cZDP/4dHgIw8sREyMXEF2L+7+OzLjvCENpR3996l3KAmKEO8K0xTvdj598O2DA4jP6ipabTLwpeo/
SQmx5cZmzLooBVihb0pdQsThEgrV2lqs93agD4S7PrqDxnr5SG7FUgaET+mvPh3bntuezbOCeuWGuO1s2n9Bgmdtz/
nnH95vqbHF0PTqSqkY3934aasPPx0LHLD/Odt/WNsZj0JasVUd3cRP+QI/Y/3SZaQPfT6bpqAJgl/i/Gfn08/
Nb07ocz7IMPzTvzniTvDid48BlUlH+U2uGZaPKFfLIbY=",
        "maxnoofdocuments": "2000",
        "storage_path": "/nass/metricValue/store",
        "registry_path": "/metadata/registerMetric",
        "oi_pool_size": 4
    }
}
```


Sample Body 2:


```
{
    "oiconnecttopologydetails": {
        "type": "https",
        "hostname": "<hostname>.broadcom.com",
        "token": "Yy7cZDP/4dHgIw8sREyMXEF2L+7+OzLjvCENpR3996l3KAmKEO8K0xTvdj598O2DA4jP6ipabTLwpeo/
SQmx5cZmzLooBVihb0pdQsThEgrV2lqs93agD4S7PrqDxnr5SG7FUgaET+mvPh3bntuezbOCeuWGuO1s2n9Bgmdtz/
nnH95vqbHF0PTqSqkY3934aasPPx0LHLD/Odt/WNsZj0JasVUd3cRP+QI/Y/3SZaQPfT6bpqAJgl/i/Gfn08/
Nb07ocz7IMPzTvzniTvDid48BlUlH+U2uGZaPKFfLIbY=",
        "port": 443,
        "maxnoofdocuments": "1000",
        "path": "/tas/graph/store",
        "oi_pool_size": 4,
        "tastenantpath": "/tenants/tenants"
    },
    "oiconnectalarmsdetails": {
        "type": "https",
        "hostname": "<hostname>.broadcom.com",
        "port": 443,
```

```
        "maxnoofdocuments": "2000",
        "path": "/ingestion",
        "oi_pool_size": 4,
        "product_id": "ao",
        "doc_type_id": "itoa_alarms_custom",
        "doc_type_version": "1"
    }
}
```

Sample Response

```
200 (Success)

{
"Status": "oiconnection detail is updated"
}

Other possibles responses:

404 - Field/parameter validation failure
401 - Authentication failure.
```

Delete the OI Connection Details

The following table provides the required information to delete the OI connection details:

Name	Description
URL	http://<hostname>:8080/restmon/api/v1/oiconnection/{oiconnectionname} https://<hostname>:8443/restmon/api/v1/oiconnection/{oiconnectionname} **Values:** oiconnectchangeeventdetails, oiconnectmetricsnassdetails, oiconnecttopologydetails, oiconnectalarmsdetails
Method	DELETE
Authorization	Basic
Response Code	200 - OK {"Status": "oiconnectchangeeventdetails is deleted"}
CURL Command	`curl -k --user <username>:<password> -X DELETE "https://<hostname>:8443/restmon/api/v1/oiconnection/oiconnectchangeeventdetails"`

Sample Request:

```
http://<hostname>:8080/restmon/api/v1/oiconnection/oiconnectchangeeventdetails
```

Sample Response:

```
200 OK
{ "Status": "oiconnectchangeeventdetails is deleted" }
```

Get the OI Setting Details

The following table lists the required information to get the OI setting details:

Name	Description
URL	http://<hostname>:8080/restmon/api/v1/oisettings https://<hostname>:8443/restmon/api/v1/oisettings
Method	GET
Authorization	Basic
Response Code	200 - OK
CURL Command	`curl -k --user <username>:<password> -X GET "https://<hostname>:8443/restmon/api/v1/oisettings"`

Sample Request:

```
http://<hostname>:8080/restmon/api/v1/oisettings
```

Sample Response:

```
{
  "oisettings": {
    "oitenantname": "<tenant-id>",
    "adminroute": "ADMIN_ROUTE",
    "oihttptimeout": "30000"
  }
}
```

Update the OI Setting Details

The following table lists the required information to update the DX OI settings details:

Name	Description
URL	http://<hostname>:8080/restmon/api/v1/oisettings https://<hostname>:8443/restmon/api/v1/oisettings
Method	PUT
Authorization	Basic
Response Code	200 - OK {"Status": "oiSetting details is updated"}
CURL Command	The file oisettings.json contains the payload to be used. `curl -k --user newuser:newpassword -X PUT --header 'Content-Type: application/json;charset=UTF-8' --data @oisettings.json "http://<hostname>:8080/restmon/api/v1/oisettings"`

Sample Request:

```
http://<hostname>:8080/restmon/api/v1/oisettings
```

Sample Response:

```
{
  "oisettings": {
    "oitenantname": "<new-tenant-id>",
    "oihttptimeout": "30000"
  }
}
```

Profile

The following table lists the different APIs that are available for RESTMon profiles:

API	Operation
GET /v1/profiles	Get All Profiles
GET /v1/profiles/{schemaName}/{profileName}	Get Profile by Profile Name and Schema Name
POST /v1/profiles	Add a Profile
PUT /v1/profiles/{schemaName}/{profileName}	Update a Profile
DELETE /v1/profiles/{schemaName}/{profileName}	Delete a Profile
PUT /v1/profiles/{schemaName}/{profileName}/activate	Enable a Profile
PUT /v1/profiles/{schemaName}/{profileName}/deactivate	Disable a Profile
PUT /v1/profiles/activate	Enable All Profiles
PUT /v1/profiles/deactivate	Disable All Profiles
GET /v1/validatedata/{schemaName}/{profileName}	Validate Data Collection

Get the Profiles

The following table lists the details that are required to display the list of profiles using the profileName and schemaName:

Name	Description
URL	http://<hostname>:8080/restmon/api/v1/profiles https://<hostname>:8443/restmon/api/v1/profiles
Method	GET
Authorization	Basic
Response Code	200 OK
CURL Command	`curl -k --user <username>:<password> -X GET "https://<hostname>:8443/restmon/api/v1/profiles"`

Sample Response:

```
{
    "profile": [
        {
            "name": "dynatrace",
            "active": "no",
            "schema": "dynatrace",
            "polling_interval_secs": "120",
            "inventory_topology_fullsync_interval_mins": "1440",
            "topology_ttl_mins": "2880"
        },
        {
            "name": "logstash",
            "active": "no",
            "schema": "googlecloudmonitoring",
            "streaming": "yes",
            "polling_interval_secs": 60
        },
        {
```

```
            "name": "purestorage",
            "active": "no",
            "schema": "purestorage",
            "polling_interval_secs": "300",
            "inventory_topology_fullsync_interval_mins": "1440",
            "topology_ttl_mins": "14400"
        }
    ]
}
```


Get the Profile Details

The following table lists the required details to get the profile details:

Name	Description
URL	http://<hostname>:8080/restmon/api/v1/profiles/{schemaName}/{profileName} https://<hostname>:8443/restmon/api/v1/profiles/{schemaName}/{profileName}
Method	GET
Authorization	Basic
Response Code	200 OK
CURL Command	`curl -k --user <username>:<password> -X GET "https://<hostname>:8443/restmon/api/v1/profiles/dynatrace/dynatrace"`

Sample Request:

```
http://<hostname>:8080/restmon/api/v1/profiles/dynatrace/dynatrace
```

Sample Response:


```
{
  "profile": {
    "name": "dynatrace",
    "active": "no",
    "schema": "dynatrace",
    "polling_interval_secs": "120",
    "inventory_topology_fullsync_interval_mins": "1440",
    "topology_ttl_mins": "2880"
  },
  "servicedefinition": {
    "name": "",
    "status": ""
  },
  "restapiconnectdetails": {
    "type": "https",
    "hostname": "hostname.live.dynatrace.com",
    "port": "",
    "authentication": "urltoken",
    "username": "",
    "password": "",
    "realmdomain": "",
    "token": "Ye5Idmpch4CqiKEd/ya+ERFEBTqwjGE3HahWNbtpgTg=",
    "httptimeout": "30000",
```

```
    "checkcert": "no"
  },
  "monitored_groups": {
    "Hosts": "yes",
    "Alarms": "yes",
    "Processes": "yes",
    "Services": "yes",
    "Application": "yes",
    "host-group": "no",
    "Hosts_Inventory": "yes",
    "Processes_Inventory": "yes",
    "Services_Inventory": "yes",
    "Dotnet-Process-Metrics": "yes"
  }
}
```

Add a Profile

The following table lists the details that are required to add a profile:

NOTE
A profile cannot be added if a profile with the same profile_name and same schema_name already exists.

Name	Description
URL	http://<hostname>:8080/restmon/api/v1/profiles https://<hostname>:8443/restmon/api/v1/profiles
Method	POST
Authorization	Basic
Response Code	200 OK
CURL Command	Create a file with the required json payload. For example, create *newprofile.json* with the given payload. Execute the following curl command to post this payload: `curl -k --user <username>:<password> -X POST --header 'Content-Type: application/json;charset=UTF-8' --data @newprofile.json https://<hostname>:8443/restmon/api/v1/profiles`

Sample Request:

```
http://<hostname>:8080/restmon/api/v1/profiles
```

Sample Body:

```
{
    "profile": {
        "name": "dynatrace",
        "active": "no",
        "schema": "dynatrace",
        "polling_interval_secs": "120",
        "inventory_topology_fullsync_interval_mins": "1440",
        "topology_ttl_mins": "2880"
    },
    "restapiconnectdetails": {
        "type": "https",
```

```
        "hostname": "<hostname>.dynatrace.com",
        "port": "",
        "authentication": "urltoken",
        "username": "",
        "password": "",
        "realmdomain": "",
        "token": "sampletoken",
        "httptimeout": "60000",
        "checkcert": "no"
    },
    "monitored_groups": {
        "Hosts": "yes",
        "Alarms": "yes",
        "Processes": "yes",
        "Services": "yes",
        "Application": "yes",
        "host-group": "yes",
        "Hosts_Inventory": "yes",
        "Processes_Inventory": "yes",
        "Services_Inventory": "yes",
        "Dotnet-Process-Metrics": "yes"
    }
}
```

Sample Response:

```
{ "Status": "profile is added with profileName dynatrace and schemaName dynatrace" }
```

Update the Profile Details

The following table lists the details that are required to update the profile by providing profileName and schemaName of that profile:

Name	Description
URL	http://<hostname>:8080/restmon/api/v1/profiles/{schemaName}/{profileName} https://<hostname>:8443/restmon/api/v1/profiles/{schemaName}/{profileName}
Method	PUT
Authorization	Basic
Response Code	200 OK
CURL Command	Create a file with the required json payload. For example, create *updateprofile.json* with the given payload. Execute the following curl command to use this payload: `curl -k --user <username>:<password> -X PUT --header 'Content-Type: application/json;charset=UTF-8' --data @updateprofile.json https://<hostname>:8443/restmon/api/v1/profiles/dynatrace/dynatrace`

Sample Request:

```
http://<hostname>:8080/restmon/api/v1/profiles/dynatrace/dynatrace
```

Sample Body:

```
{
    "profile": {
```

```
        "name": "dynatrace",
        "active": "no",
        "schema": "dynatrace",
        "polling_interval_secs": "120",
        "inventory_topology_fullsync_interval_mins": "1440",
        "topology_ttl_mins": "2880"
    },
    "restapiconnectdetails": {
        "type": "https",
        "hostname": '<hostname>.dynatrace.com",
        "port": "",
        "authentication": "urltoken",
        "username": "",
        "password": "",
        "realmdomain": "",
        "token": "sampletoken",
        "httptimeout": "60000",
        "checkcert": "no"
    },
    "monitored_groups": {
        "Hosts": "yes",
        "Alarms": "yes",
        "Processes": "yes",
        "Services": "yes",
        "Application": "yes",
        "host-group": "no",
        "Hosts_Inventory": "no",
        "Processes_Inventory": "no",
        "Services_Inventory": "no",
        "Dotnet-Process-Metrics": "yes"
    }
}
```

Sample Response:

```
{ "Status": "Profile is updated with profileName dynatrace and schemaName dynatrace" }
```

Delete a Profile

The following table lists the details that are required to delete a profile matching the given profile name and schema name from restmon.json:

Name	Description
URL	http://<hostname>:8080/restmon/api/v1/profiles/{schemaName}/{profileName} https://<hostname>:8443/restmon/api/v1/profiles/{schemaName}/{profileName}
Method	DELETE
Authorization	Basic
Response Code	200 OK
CURL Command	`curl -k --user <username>:<password> -X DELETE "https://<hostname>:8443/restmon/api/v1/profiles/dynatrace/dynatrace"`

Sample Request:

```
http://<hostname>:8080/restmon/api/v1/profiles/dynatrace/dynatrace
```

Sample Response:

```
{ "Status": "Profile is deleted with profileName dynatrace and schemaName dynatrace" }
```

Enable a Profile

The following table provides the details that are required to enable a profile:

Name	Description
URL	http://<hostname>:8080/restmon/api/v1/profiles/{schemaName}/{profileName}/activate https://<hostname>:8443/restmon/api/v1/profiles/{schemaName}/{profileName}/activate
Method	PUT
Authorization	Basic
Response Code	200 OK
CURL Command	`curl --user <username>:<password> -X PUT --header 'Content-Type: application/json;charset=UTF-8' http://<hostname>:8080/restmon/api/v1/profiles/dynatrace/dynatrace/activate`

Sample Request:

```
http://<hostname>:8080/restmon/api/v1/profiles/dynatrace/dynatrace/activate
```

Sample Response:

```
{ "Status": "Profile with profileName dynatrace and schemaName dynatrace is activated." }
```

Disable a Profile

The following table lists the details that are required to disable a profile:

Name	Description
URL	http://<hostname>:8080/restmon/api/v1/profiles/{schemaName}/{profileName}/deactivate https://<hostname>:8443/restmon/api/v1/profiles/{schemaName}/{profileName}/deactivate
Method	PUT
Authorization	Basic
Response Code	200 OK
CURL Command	`curl --user <username>:<password> -X PUT --header 'Content-Type: application/json;charset=UTF-8' http://<hostname>:8080/restmon/api/v1/profiles/dynatrace/dynatrace/deactivate`

Sample Request:

```
http://<hostname>:8080/restmon/api/v1/profiles/dynatrace/dynatrace/deactivate
```

Sample Response:

```
{ "Status": "Profile with profileName dynatrace and schemaName dynatrace is deactivated." }
```

Enable All Profiles

The following table lists the details that are required to enable all the profiles:

Name	Description
URL	http://<hostname>:8080/restmon/api/v1/profiles/activate https://<hostname>:8443/restmon/api/v1/profiles/activate
Method	PUT
Authorization	Basic
Response Code	200 OK
CURL Command	`curl --user <username>:<password> -X PUT --header 'Content-Type: application/json;charset=UTF-8' http://<hostname>:8080/restmon/api/v1/profiles/activate`

Sample Request:

```
http://<hostname>:8080/restmon/api/v1/profiles/activate
```

Sample Response:

```
{ "Status": "All the profiles are activated." }
```

Disable All Profiles

The following table lists the details that are required to disable all the profiles:

Name	Description
URL	http://<hostname>:8080/restmon/api/v1/profiles/deactivate https://<hostname>:8443/restmon/api/v1/profiles/deactivate
Method	PUT
Authorization	Basic
Response Code	200 OK
CURL Command	`curl --user <username>:<password> -X PUT --header 'Content-Type: application/json;charset=UTF-8' http://<hostname>:8080/restmon/api/v1/profiles/deactivate`

Sample Request:

```
http://<hostname>:8080/restmon/api/v1/profiles/deactivate
```

Sample Response:

```
{ "Status": "All the profiles are deactivated." }
```

Validate the Data Collection

This API validates the profile and schema configurations for correctness. If the configuration is valid and the data collection and transformation are happening properly, the profile to be created or updated with polling interval 0, and

the data that is collected is written into the **target/output/{profilename~~schemaname}** folder. If there are any issues, this folder is created. The following table lists the details that are required to validate the data.

Name	Description
URL	http://<hostname>:8080/restmon/api/v1/validatedata/{schemaName}/{profileName} https://<hostname>:8443/restmon/api/v1/validatedata/{schemaName}/{profileName}
Method	GET
Authorization	Basic
Response Code	200 OK
CURL Command	`curl --user <username>:<password> -X GET http://<hostname>:8080/restmon/api/v1/validate/dynatrace/dynatrace`

Sample Request:

```
http://<hostname>:8080/restmon/api/v1/validate/dynatrace/dynatrace
```

Sample Response:

```
If the data collection is enabled with polling interval 0:
{ "Status": "Data collection is happening." }If the data collection is not enabled with a polling interval 0
{ "Status": "Data collection is not happening." }
```

Profile APIs

none

```
{
    "swagger": "2.0",
    "info": {},
    "host": "localhost:8080",
    "basePath": "/restmon/api",
    "tags": [{
        "name": "profiles",
        "description": "These set of APIs enable you to perform CRUD operations for
 RESTmon profiles."
    }],
    "paths": {
        "/v1/oiconnection": {
            "get": {
                "tags": [
                    "profiles"
                ],
                "summary": "Gets OI connection details.",
                "operationId": "getOIDetailsUsingGET",
                "consumes": [
                    "application/json"
                ],
                "produces": [
```

```
            "application/json;charset=UTF-8"
        ],
        "responses": {
            "200": {
                "description": "OK",
                "schema": {
                    "type": "object"
                }
            },
            "401": {
                "description": "Unauthorized"
            },
            "403": {
                "description": "Forbidden"
            },
            "404": {
                "description": "Not Found"
            }
        }
    },
    "put": {
        "tags": [
            "profiles"
        ],
        "summary": "Updates OI connection details.",
        "operationId": "updateOIConnetionDetailsUsingPUT",
        "consumes": [
            "application/json;charset=UTF-8"
        ],
        "produces": [
            "application/json;charset=UTF-8"
        ],
        "parameters": [{
            "in": "body",
            "name": "inputBody",
            "description": "Update the OI Connection details.",
            "required": true,
            "schema": {
                "$ref": "#/definitions/ObjectNode"
            }
        }],
        "responses": {
            "200": {
                "description": "OK",
                "schema": {
                    "type": "object"
```

```
                }
            },
            "201": {
                "description": "Created"
            },
            "401": {
                "description": "Unauthorized"
            },
            "403": {
                "description": "Forbidden"
            },
            "404": {
                "description": "Not Found"
            }
        }
    }
},
"/v1/oiconnection/{oiconnectionName}": {
    "delete": {
        "tags": [
            "profiles"
        ],
        "summary": "Deletes the OI connection details.",
        "operationId": "deleteOIDetailsUsingDELETE",
        "consumes": [
            "application/json"
        ],
        "produces": [
            "application/json;charset=UTF-8"
        ],
        "parameters": [{
            "name": "oiconnectionName",
            "in": "path",
            "description": "Specify the OI connection Name.",
            "required": true,
            "type": "string"
        }],
        "responses": {
            "200": {
                "description": "OK",
                "schema": {
                    "type": "object"
                }
            },
            "204": {
                "description": "No Content"
```

```
            },
            "401": {
                "description": "Unauthorized"
            },
            "403": {
                "description": "Forbidden"
            }
        }
    }
},
"/v1/profiles": {
    "get": {
        "tags": [
            "profiles"
        ],
        "summary": "Lists all Profiles available for editing.",
        "operationId": "getProfileListUsingGET",
        "consumes": [
            "application/json"
        ],
        "produces": [
            "application/json;charset=UTF-8"
        ],
        "responses": {
            "200": {
                "description": "OK",
                "schema": {
                    "type": "object"
                }
            },
            "401": {
                "description": "Unauthorized"
            },
            "403": {
                "description": "Forbidden"
            },
            "404": {
                "description": "Not Found"
            }
        }
    },
    "post": {
        "tags": [
            "profiles"
        ],
```

```
                "summary": "Adds a profile. Note - A profile cannot be added if a
profile already exists with the same profile_name and schema_name.",
                "operationId": "addProfileUsingPOST",
                "consumes": [
                    "application/json"
                ],
                "produces": [
                    "application/json"
                ],
                "parameters": [{
                    "in": "body",
                    "name": "inputBody",
                    "description": "inputBody",
                    "required": true,
                    "schema": {
                        "$ref": "#/definitions/ObjectNode"
                    }
                }],
                "responses": {
                    "200": {
                        "description": "OK",
                        "schema": {
                            "type": "object"
                        }
                    },
                    "201": {
                        "description": "Created"
                    },
                    "401": {
                        "description": "Unauthorized"
                    },
                    "403": {
                        "description": "Forbidden"
                    },
                    "404": {
                        "description": "Not Found"
                    }
                }
            }
        },
        "/v1/profiles/{schemaName}/{profileName}": {
            "get": {
                "tags": [
                    "profiles"
                ],
                "summary": "Lists a profile by Profile Name and Schema Name.",
```

```
            "operationId": "getProfileDetailsUsingGET",
            "consumes": [
                "application/json"
            ],
            "produces": [
                "application/json;charset=UTF-8"
            ],
            "parameters": [{
                    "name": "schemaName",
                    "in": "path",
                    "description": "Specify the Schema Name.",
                    "required": true,
                    "type": "string"
                },
                {
                    "name": "profileName",
                    "in": "path",
                    "description": "Specify the Profile Name.",
                    "required": true,
                    "type": "string"
                }
            ],
            "responses": {
                "200": {
                    "description": "OK",
                    "schema": {
                        "type": "object"
                    }
                },
                "401": {
                    "description": "Unauthorized"
                },
                "403": {
                    "description": "Forbidden"
                },
                "404": {
                    "description": "Not Found"
                }
            }
        },
        "put": {
            "tags": [
                "profiles"
            ],
            "summary": "Updates a Profile.",
            "operationId": "updateProfileDetailsUsingPUT",
```

```
"consumes": [
    "application/json;charset=UTF-8"
],
"produces": [
    "application/json;charset=UTF-8"
],
"parameters": [{
        "in": "body",
        "name": "inputBody",
        "description": "inputBody",
        "required": true,
        "schema": {
            "$ref": "#/definitions/ObjectNode"
        }
    },
    {
        "name": "schemaName",
        "in": "path",
        "description": "Specify the Schema Name.",
        "required": true,
        "type": "string"
    },
    {
        "name": "profileName",
        "in": "path",
        "description": "Specify the Profile Name.",
        "required": true,
        "type": "string"
    }
],
"responses": {
    "200": {
        "description": "OK",
        "schema": {
            "type": "object"
        }
    },
    "201": {
        "description": "Created"
    },
    "401": {
        "description": "Unauthorized"
    },
    "403": {
        "description": "Forbidden"
    },
```

```
            "404": {
                "description": "Not Found"
            }
        }
    },
    "delete": {
        "tags": [
            "profiles"
        ],
        "summary": "Deletes a Profile.",
        "operationId": "deleteProfileUsingDELETE",
        "consumes": [
            "application/json"
        ],
        "produces": [
            "application/json"
        ],
        "parameters": [{
                "name": "schemaName",
                "in": "path",
                "description": "Specify the Schema Name.",
                "required": true,
                "type": "string"
            },
            {
                "name": "profileName",
                "in": "path",
                "description": "Specify the Profile Name.",
                "required": true,
                "type": "string"
            }
        ],
        "responses": {
            "200": {
                "description": "OK",
                "schema": {
                    "type": "object"
                }
            },
            "204": {
                "description": "No Content"
            },
            "401": {
                "description": "Unauthorized"
            },
            "403": {
```

```
                    "description": "Forbidden"
                }
            }
        }
    },
    "get": {
        "tags": [
            "schema"
        ],
        "summary": "Get All schema ",
        "operationId": "getAllSchemaUsingGET",
        "consumes": [
            "application/json"
        ],
        "produces": [
            "application/json"
        ],
        "responses": {
            "200": {
                "description": "OK",
                "schema": {
                    "type": "object"
                }
            },
            "401": {
                "description": "Unauthorized"
            },
            "403": {
                "description": "Forbidden"
            },
            "404": {
                "description": "Not Found"
            }
        }
    }
},
"/v1/validatedata/{schemaName}/{profileName}": {
    "get": {
        "tags": [
            "profiles"
        ],
        "summary": "Validate Data Collection",
        "operationId": "validateDataCollectionUsingGET",
        "consumes": [
            "application/json"
        ],
```

```
            "produces": [
                "application/json;charset=UTF-8"
            ],
            "parameters": [{
                    "name": "schemaName",
                    "in": "path",
                    "description": "schemaName",
                    "required": true,
                    "type": "string"
                },
                {
                    "name": "profileName",
                    "in": "path",
                    "description": "profileName",
                    "required": true,
                    "type": "string"
                }
            ],
            "responses": {
                "200": {
                    "description": "OK",
                    "schema": {
                        "type": "object"
                    }
                },
                "401": {
                    "description": "Unauthorized"
                },
                "403": {
                    "description": "Forbidden"
                },
                "404": {
                    "description": "Not Found"
                }
            }
        }

    },
    "definitions": {
        "JsonNode": {
            "type": "object"
        },
        "ObjectNode": {
            "type": "object"
        }
    }
}
```

APIs for Schema

The following table lists the different APIs that are available for schema at a high level:

API	Operation
GET /v1/schema/getAllSchemaName	Get All Schema Names
GET /v1/schema/getSchemaDetails/{schemaName}	Get Schema Details
GET /v1/schema/{schemaName}/{fieldName}	Get Schema Details by Field Name
GET /v1/schema/{schemaName}/{fieldName}/{fieldId}	Get Schema Details by Field Name and Field ID
POST /v1/schema/{schemaName}	Upload the Schema
POST /v1/schema/{schemaName}/{fieldName}	Add Field Information to Schema
PUT /v1/schema/{schemaName}/{fieldName}/{fieldId}	Update Field Information by Field ID
DELETE /v1/schema/{schemaName}	Delete Schema
DELETE /v1/schema/{schemaName}/{fieldName}/{fieldId}	Delete Field Details by Field ID

GET All Schema Names

You can use this API to list all the schemas available in the schema directory. The following table lists the details that are required to get all the schema names:

Name	Description
URL	http://<host-name>:8080/restmon/api/v1/schema/getAllSchemaName https://<host-name>:8443/restmon/api/v1/schema/getAllSchemaName
Method	GET
Authorization	Basic
Response Code	200 OK

Sample Response:

```
{
  "schema": [
    {
      "schema_name": "appdynamics",
      "schema_version": "1.6",
      "uploadedBy": "RESTMON"
    },
    {
      "schema_name": "appoptics",
      "schema_version": "1.6",
      "uploadedBy": "RESTMON"
    },
    {
      "schema_name": "dynatrace",
```

```
      "schema_version": "1.5",
      "uploadedBy": "TENANT"
    },
    {
      "schema_name": "netcoolomnibus",
      "schema_version": "1.6",
      "uploadedBy": "RESTmon"
    },
    {
      "schema_name": "netcoolomnibuswebhook",
      "schema_version": "1.6",
      "uploadedBy": "RESTMON"
    },
    {
      "schema_name": "purestorage",
      "schema_version": "1.6",
      "uploadedBy": "RESTMON"
    },
    {
      "schema_name": "scom",
      "schema_version": "1.6",
      "uploadedBy": "RESTMON"
    },
    {
      "schema_name": "snow",
      "schema_version": "1.6",
      "uploadedBy": "RESTMON"
    },
    {
      "schema_name": "testschema",
      "schema_version": "1.6.1",
      "uploadedBy": "TENANT"
    }
  ]
}
```


GET the Schema Details

You can use this API to list the details for the specified schema. The following table provides the details required to get the schema details:

Name	Description
URL	http://<hostname>:8080/restmon/api/v1/schema/getSchemaDetails/{schemaName} https://<hostname>:8443/restmon/api/v1/schema/getSchemaDetails/{schemaName}
Method	GET
Authorization	Basic
Response Code	200 OK
Curl Command	curl -k --user <username>:<password> -X GET 'https://localhost:8443/restmon/api/v1/schema/getSchemaDetails/testschema'

Sample Request:

```
http://<hostname>:8080/restmon/api/v1/schema/getSchemaDetails/testschema
```

Sample Response:


```
{
  "testschema": {
    "definition": {
      "resource_category": "",
      "auth": "",
      "xml_ns": "",
      "name": "testschema",
      "type": "",
      "uploadedBy": "TENANT"
    },
    "urls": [],
    "topology": [
      {
        "xml_ns": "",
        "url": "",
        "group": "Topology",
        "layer": "CUSTOM",
        "attributes": {
          "oi": {
            "ci_unique_id": "$['FQDN']",
            "type": "HOST",
            "product": "Netcool/Omnibus",
            "name": "$['CI Name']",
            "hostname": "$['FQDN']",
            "Display Name": "$['CI Name']",
            "ipAddresses": "#(function() {return root['IP Address'].split(' ').join('');})();",
            "macAddresses": "#(function() {return root['MAC
 Address'].split('.').join(':').split('-').join(':').split(' ').join('');})();",
            "FQDNHostname": "$['FQDN']",
            "it_service_instance": "$['IT Service Instance']",
            "environment": "$['Environment']]",
            "infrastructure_itsi": "$['Infrastructure ITSI']",
            "class_label": "$['Class Label']",
            "itsi_support_group": "$['ITSI Support Group']",
            "infra_itsi_support_group": "$['Infra ITSI Support Group']"
          }
        },
        "fieldId": "957866A8EC754E579B5FDF0974152234"
      }
    ],
    "inventory": [],
    "alarms": [],
    "metrics": [],
    "calculated_methods": [],
    "calculated_metrics": [],
    "groups": [],
    "change_events": []
  }
}
```

GET the Schema Details by Field Name

You can use this API to list the schema details by the specified field name. You can get details by URLs, metrics, alarms, inventory, topology, groups, calculated_metrics, calculated_methods, and change_events. The following table lists the details required to get the schema details by field name:

Name	Description
URL	http://<hostname>:8080/restmon/api/v1/schema/{schemaName}/{fieldName} https://<hostname>:8443/restmon/api/v1/schema/{schemaName}/{fieldName}
Method	GET
Authorization	Basic

NOTE
For more information about each field name, see the Get the Schema Details By Field Name section.

GET the Schema Details by Field Name and Field ID

You can use these APIs to get the schema details by the specified field ID against the given Field ID. You can get the details by URLs, metrics, alarms, inventory, topology, groups, calculated_metrics, calculated_methods, and change_events against the given field ID. The following table lists the details that are required to get the schema details by field name and field ID:

Name	Description
URL	http://<hostname>:8080/restmon/api/v1/schema/{schemaName}/{fieldName}/{fieldId} https://<hostname>:8443/restmon/api/v1/schema/{schemaName}/{fieldName}/{fieldId}
Method	GET
Authorization	Basic

NOTE
For more information, see the Get the Schema Details By Field Name and Field ID section.

Upload the Schema File (POST)

You can use this API to upload a schema file. The following table lists the details that are required to upload the schema file:

Name	Description
URL	http://<hostname>:8080/restmon/api/v1/schema/{schemaName} https://<hostname>:8443/restmon/api/v1/schema/{schemaName}
Method	POST
Authorization	Basic
Response	200 OK { "Status": "Schema File is uploaded" }

Name	Description
Curl Command	Create a file with the required JSON payload (for example, testschema.json) and then run the following command to use this payload: `curl -k --user <user-name>:<password> -X PUT --header 'Content-Type: application/json;charset=UTF-8' --data @testschema.json https://<hostname>:8443/restmon/api/v1/schema/testschema` While uploading the schema, if: • Schema exists with content **"uploadedBy" = "RESTMON",** a backup is taken and **"uploadedBy"** will have a new value **"TENANT"**. • Schema exists with content **"uploadedBy" = "RESTMON",** and **"updatedBy"="TENANT",** no backup is taken, and schema will be uploaded. And **"uploadedBy"** will have a new value **"TENANT"**. • Schema exists with content **"uploadedBy" = "TENANT",** no backup is taken, and the schema will be uploaded.

Sample Request:

```
http://<hostname>:8080/restmon/api/v1/schema/testschema
```

Body:

```
{
    "testschema": {
        "definition": {
            "resource_category": "",
            "auth": "",
            "xml_ns": "",
            "name": "testschema",
            "type": ""
        },
        "urls": [],
        "topology": [
            {
                "xml_ns": "",
                "url": "",
                "group": "Topology",
                "layer": "CUSTOM",
                "attributes": {
                    "oi": {
                        "ci_unique_id": "$['FQDN']",
                        "type": "HOST",
                        "product": "product",
                        "name": "$['CI Name']",
                        "hostname": "$['FQDN']",
                        "Display Name": "$['CI Name']",
                        "ipAddresses": "#(function() {return root['IP Address'].split(' ').join('');})();",
                        "macAddresses": "#(function() {return root['MAC
Address'].split('.').join(':').split('-').join(':').split(' ').join('');})();",
                        "FQDNHostname": "$['FQDN']",
                        "it_service_instance": "$['IT Service Instance']",
                        "environment": "$['Environment']]",
                        "infrastructure_itsi": "$['Infrastructure ITSI']",
                        "class_label": "$['Class Label']",
```

```
                        "itsi_support_group": "$['ITSI Support Group']",
                        "infra_itsi_support_group": "$['Infra ITSI Support Group']"
                    }
                }
            }
        ],
        "inventory": [],
        "alarms": []
    }
}
```

Add Field Information to Schema (POST)

You can use these APIs to add the specified field information to the specified schema. You can add details to the schema by urls, metrics, alarms, inventory, topology, groups, calculated_metrics, calculated_methods, and change_events. This method adds an object in the provided field name by an auto-generated ID:

The following table lists the details required to add the schema data using the field names:

Name	Description
URL	http://<hostname>:8080/restmon/api/v1/schema/{schemaName}/{fieldName} https://<hostname>:8443/restmon/api/v1/schema/{schemaName}/{fieldName}
Method	POST
Authorization	Basic

NOTE
For more information about each field, see the Add the Schema Field Data By Field Name section.

Update Schema Details by Field Name and Field ID (PUT)

You can use these APIs to modify the field information by the field names and specified field ID. You can update by urls, metrics, alarms, inventory, topology, groups, calculated_metrics, calculated_methods, and change_events. This method modifies an object in the provided field name and field ID.

NOTE

- Single JSON block per entity is allowed.
- The FieldID field is mandatory in the payload.

The following table lists the details required to update the schema details by field name and field ID:

Name	Description
URL	http://<hostname>:8080/restmon/api/v1/schema/{schemaName}/{fieldName}/{fieldId} https://<hostname>:8443/restmon/api/v1/schema/{schemaName}/{fieldName}/{fieldId}
Method	PUT
Authorization	Basic

NOTE
For more information, see the Update Schema Details by Field Name and Field ID section.

Delete Schema

You can use this API to delete the Schema file by providing the schema name. The following table lists the details required to delete the schema file:

NOTE
The Schema file is not deleted if the schema has "uploadedBy"="RESTMON" (the OOTB schema).

Name	Description
URL	http://<hostname>:8080/restmon/api/v1/schema/{schemaName} https://<hostname>:8443/restmon/api/v1/schema/{schemaName}
Method	DELETE
Authorization	Basic
Response	200 OK { "Status": "testschema Schema File is deleted" }
Curl Command	`curl -k --user <username>:<password> -X DELETE 'https://localhost:8443/restmon/api/v1/schema/testschema'`

Delete Schema Details by Field Name and Field ID

You can use this API to delete the specified field name by the specified field ID. You can delete by urls, metrics, alarms, inventory, topology, groups, calculated_metrics, calculated_methods, and change_events against the given field ID. This method will delete the object as per the provided field name and the field ID.

The following table lists the details required to delete the schema details by the field name and field ID:

Name	Description
URL	http://<hostname>:8080/restmon/api/v1/schema/{schemaName}/{fieldName}/{fieldId} https://<hostname>:8443/restmon/api/v1/schema/{schemaName}/{fieldName}/{fieldId}
Method	DELETE
Authorization	Basic

NOTE
For more information about each field, see the Delete the Schema Details by Field Name and Field ID section.

Add the Schema Field Data By Field Name

You can add field information to the specified schema for the following fields. This method adds an object in the provided field name by an auto-generated ID.

API	Operation
POST /v1/schema/{schemaName}/{urls}	Add the field information to the specified schema by URL
POST /v1/schema/{schemaName}/{metrics}	Add the field information to the specified schema by Metrics
POST /v1/schema/{schemaName}/{alarms}	Add the field information to the specified schema by Alarms
POST /v1/schema/{schemaName}/{inventory}	Add the field information to the specified schema by Inventory
POST /v1/schema/{schemaName}/{topology}	Add the field information to the specified schema by Topology
POST /v1/schema/{schemaName}/{groups}	Add the field information to the specified schema by Groups
POST /v1/schema/{schemaName}/{calculated_metrics}	Add the field information to the specified schema by calculated_metrics

API	Operation
POST /v1/schema/{schemaName}/{calculated_methods}	Add the field information to the specified schema by calculated_methods
POST /v1/schema/{schemaName}/{changed_events}	Add the field information to the specified schema by change_events

Add the Field Data by URLs

The following table provides the details required to add details to the schema by URLs:

Name	Description
URL	http://<host-name>:8080/restmon/api/v1/schema/{schemaName}/{urls} https://<host-name>:8443/restmon/api/v1/schema/{schemaName}/{urls}
Method	POST
Authorization	Basic
Response Code	200 OK
Curl Command	Copy the payload into a file, for example, **url.json** and execute the following command: `curl -k --user <user-name>:<password> -X POST --header 'Content-Type: application/json;charset=UTF-8' --data @url.json 'https://localhost:8443/restmon/api/v1/schema/testschema/urls'`

Sample Request:

```
https://localhost:8443/restmon/api/v1/schema/testschema/urls
```

Body:

```
{
"src": "",
"xml_ns": "",
"var": "",
"id": "hosts",
"url": "/api/v1/hosts"
}
```

Sample Response:

```
{ "Status:": "urls is added to schema testschema with fieldId E377E130716D4D2EB2D53D9F36A38804" }
```

Add the Field Data by Metrics

The following table provides the details required to add details to the schema by metrics:

Name	Description
URL	http://<host-name>:8080/restmon/api/v1/schema/{schemaName}/{metrics} https://<host-name>:8443/restmon/api/v1/schema/{schemaName}/{metrics}
Method	POST
Authorization	Basic
Response Code	200 OK

Name	Description
Curl Command	Copy the payload into a file, for example, metric.json and execute the following command: `curl -k --user <user-name>:<password> -X POST --header 'Content-Type: application/json;charset=UTF-8' --data @metric.json 'https://localhost:8443/restmon/api/v1/schema/testschema/metrics'`

Sample Request:

```
https://localhost:8443/restmon/api/v1/schema/testschema/metrics
```

Body:

```
{
                "attributes": {
                    "oi": {
                        "metric_name": "$['series'][*]['metric']",
                        "metric_type": "Memory Breakdown",
                        "metric_unique_id": "$['series'][*]['metric']%//%:%//%$['series'][*]['scope']",
                        "metric_unit": "MB",
                        "timestamp": "#(function(){var result=[];for(var i=0;i<root.series.length;i+
+){var timeLength=root.series[i].pointlist.length-1;var newTime=root.series[i].pointlist[timeLength]
[0];result.push(newTime.toString());}return result;})();",
                        "configuration_item": "",
                        "configuration_item_type": "Host",
                        "host": "#(function(){var result=[];for(var i=0;i<root.series.length;i++){var
 firstSplit=root.series[i].scope.split(':');result.push(firstSplit[1]);}return result;})();",
                        "product": "DataDog",
                        "type": "2",
                        "product_version": "1.0.0",
                        "ci_unique_id": "#(function(){var result=[];for(var i=0;i<root.series.length;i++){var
 firstSplit=root.series[i].scope.split(':');result.push(firstSplit[1]);}return result;})();",
                        "tags": ["DataDog", "System Metrics", "$['series'][*]['metric']"]
                    }
                },
                "value": "#(function(){var result=[];for(var i=0;i<root.series.length;i++){var
 pointsLength=root.series[i].length-1;var pointValue=root.series[i].pointlist[pointsLength][1];var totalSizeMB
 = pointValue / Math.pow(1024,2);result.push(totalSizeMB.toFixed(2));}return result;})();",
                "url": "memory-cached",
                "group": "Metrics"
            }
```

Sample Response:

```
{ "Status:": "metrics is added in schema testschema with fieldId 17022273AA2E45769F841206470BFCC1" }
```

Add the Field Data by Alarms

The following table lists the details that are required to add details to the schema by alarms:

Name	Description
URL	http://<host-name>:8080/restmon/api/v1/schema/{schemaName}/{alarms} https://<host-name>:8443/restmon/api/v1/schema/{schemaName}/{alarms}
Method	POST
Authorization	Basic
Response Code	200 OK
Curl Command	Copy the payload into a file, for example, alarm.json and execute the following command: `curl -k --user <user-name>:<password> -X POST --header 'Content-Type: application/json;charset=UTF-8' --data @alarm.json 'https://localhost:8443/restmon/api/v1/schema/testschema/alarms'`

Sample Request:

```
https://localhost:8443/restmon/api/v1/schema/testschema/alarms
```

Body:

```
{
    "xml_ns": "",
    "url": "alert-details",
    "group": "Alarms",
    "attributes": {
        "oi": {
            "timestamp": "%timestamp",
            "host": "$[*]['host']",
            "product": "DataDog",
            "product_version": "1.0.0",
            "startTime": "#(function(){var result=[];var payloadString=new
 Date(root.event.date_happened*1000).toISOString();result.push(payloadString); return result;})();",
            "summary": "Alarms",
            "severity": "$['event']['alert_type']",
            "severity_conversion": "error:Critical,warning:Minor,Default:",
            "metric_name": "#(function(){var result=[];if(root.event){var
 payLoadString = JSON.parse(root.event.payload);if(payLoadString.result)
{result.push( payLoadString.result.metadata.metric);}else{result.push(payLoadString.result_aggr.result_metadata.metric)
 result;})();",
            "metric_type": "",
            "configuration_item_type": "Host",
            "configuration_item": "",
            "message": "$[*]['alarm_message']",
            "alarm_unique_id": "$[*]['alarm_id']",
            "status": "",
            "tags": [
                "DataDog",
                "Alarms"
            ],
            "ci_unique_id": "$[*]['host']",
```

```
            "alarmURL": "#(function(){var test='';var result=[];var jsonString =
 JSON.parse(root.event.payload);var alertString='';if(jsonString.result_aggr){alertString =
 jsonString.result_aggr.result_metadata.alert_url;}else{alertString=jsonString.result.metadata.alert_url;}if(alertStrin
{result.push('https://app.datadoghq.com'+alertString);}else{result.push('https://
app.datadoghq.com'+alertString);}return result;})();",
            "alarmType": "Infrastructure",
            "condition": "#(function(){if(root.event.host){return true;}else{return false;}})();"
        }
    }
}
```

Sample Response:

```
{ "Status:": "alarms is added in schema testschema with fieldId 019943440D014F36A45ECC3B9384DC5D" }
```

Add the Field Data by Inventory

The following table lists the details that are required to add the schema data by inventory:

Name	Description
URL	http://<host-name>:8080/restmon/api/v1/schema/{schemaName}/{inventory} https://<host-name>:8443/restmon/api/v1/schema/{schemaName}/{inventory}
Method	POST
Authorization	Basic
Response Code	200 OK
Curl Command	Copy the payload into a file, for example, inventory.json and execute the following command: `curl -k --user <user-name>:<password> -X POST --header 'Content-Type: application/json;charset=UTF-8' --data @inventory.json 'https://localhost:8443/restmon/api/v1/schema/testschema/inventory'`

Sample Request:

```
https://localhost:8443/restmon/api/v1/schema/testschema/inventory
```

Body:

```
{
    "xml_ns": "",
    "url": "hosts",
    "group": "Hosts_Inventory",
    "attributes": {
        "oi": {
            "ci_unique_id": "$['host_list'][*]['name']",
            "host": "$['host_list'][*]['name']",
            "product": "DataDog",
            "configuration_item": "$['host_list'][*]['name']",
            "product_version": "1.0.0",
            "tags": [
                "DataDog",
                "Hosts",
                "$['host_list'][*]['name']",
                "ExcludeFromTAS"
```

```
            ],
            "display_name": "$['host_list'][*]['name']",
            "configuration_item_type": "HOST",
            "type": "Host"
        }
    }
}
```

Sample Response:

```
{ "Status:": "inventory is added in schema testschema with fieldId 197A32D66DDF4246AA960F0E153F26BC" }
```

Add the Field Data by Topology

The following table provides the details required to add the schema data by topology:

Name	Description
URL	http://<host-name>:8080/restmon/api/v1/schema/{schemaName}/{topology} https://<host-name>:8443/restmon/api/v1/schema/{schemaName}/{topology}
Method	POST
Authorization	Basic
Response Code	200 OK
Curl Command	Copy the payload into a file, for example, topology.json and execute the following command: `curl -k --user <user-name>:<password> -X POST --header 'Content-Type: application/json;charset=UTF-8' --data @topology.json 'https://localhost:8443/restmon/api/v1/schema/testschema/topology'`

Sample Request:

```
https://localhost:8443/restmon/api/v1/schema/testschema/topology
```

Body:

```
{
    "xml_ns": "",
    "url": "hosts",
    "group": "Hosts",
    "layer": "CUSTOM",
    "attributes": {
        "oi": {
            "name": "$['host_list'][*]['name']",
            "type": "HOST",
            "hostname": "$['host_list'][*]['name']",
            "ci_unique_id": "$['host_list'][*]['name']",
            "product": "DataDog",
            "ipAddresses": "$['host_list'][*]['ip']",
            "macAddresses": "$['host_list'][*]['mac']",
            "product_version": "1.0.0"
        }
    }
}
```

Sample Response:

```
{ "Status:": "topology is added in schema testschema with fieldId FD92035E695B42D6A73951A1A4984525" }
```

Add the Field Data by Groups

The following table lists the details required to add the schema data by groups:

Name	Description
URL	http://<host-name>:8080/restmon/api/v1/schema/{schemaName}/{groups} https://<host-name>:8443/restmon/api/v1/schema/{schemaName}/{groups}
Method	POST
Authorization	Basic
Response Code	200 OK
Curl Command	Copy the payload into a file, for example, groups.json and execute the following command: `curl -k --user <user-name>:<password> -X POST --header 'Content-Type: application/json;charset=UTF-8' --data @groups.json 'https://localhost:8443/restmon/api/v1/schema/testschema/groups'`

Sample Request:

```
https://localhost:8443/restmon/api/v1/schema/testschema/groups
```

Body:

```
{
    "xml_ns": "",
    "url": "hosts",
    "group": "host-group",
    "layer": "GROUPS_CUSTOM",
    "attributes": {
        "oi": {
            "group_id": "$[*]['hostGroup']['meId']",
            "name": "$[*]['hostGroup']['name']",
            "groupname": "$[*]['hostGroup']['name']",
            "product": "DataDog",
            "type": "HOST_GROUP"
        }
    }
}
```

Sample Response:

```
{ "Status:": "groups is added in schema testschema with fieldId 79F90DB5E48E480FABE38AACEC6FFD03" }
```

Add the Field Data by Calculated_Metrics

The following table lists the details required to add the schema data by calculated_metrics:

Name	Description
URL	http://<host-name>:8080/restmon/api/v1/schema/{schemaName}/{calculated_metrics} https://<host-name>:8443/restmon/api/v1/schema/{schemaName}/{calculated_metrics}
Method	POST

Name	Description
Authorization	Basic
Response Code	200 OK
Curl Command	Copy the payload into a file, for example, calculated_metrics.json and execute the following command: `curl -k --user <user-name>:<password> -X POST --header 'Content-Type: application/json;charset=UTF-8' --data @calculated_metrics.json 'https://localhost:8443/restmon/api/v1/schema/testschema/calculated_metrics'`

Sample Request:

```
https://localhost:8443/restmon/api/v1/schema/testschema/calculated_metrics
```

Body:

```
{
    "calculation": "($total_provisioned - $total_free) / 1048576",
    "xml_ns": "",
    "values": [
        {
            "name": "$total_free",
            "value": "$['totalFree_gb']"
        },
        {
            "name": "$total_provisioned",
            "value": "$['totalProvisioned_gb']"
        }
    ],
    "group": "CapacityMetrics",
    "attributes": {
        "oi": {
            "metric_name": "Total Used",
            "metric_type": "Cluster Capacity",
            "metric_unique_id": "%clusterName%//%:%//%TotalUsed",
            "metric_unit": "PB",
            "type": "1",
            "configuration_item_type": "ECS.Capacity",
            "configuration_item": "%clusterName",
            "ci_unique_id": "%clusterName",
            "host": "%clusterName",
            "ip": "",
            "product": "ECS",
            "product_version": "1.17",
            "tags": [
                "ECS",
                "Total Used",
                "Capacity: Total Used",
                "%clusterName"
            ]
        }
    },
    "url": "capacity"
```

```
}
```

Sample Response:

```
{ "Status:": "calculated_metrics is added in schema testschema with fieldId
 A9139757986149CAA72310892A8A09EE" }
```

Add the Field Data by Calculated_Methods

The following table lists the details that a re required to add the schema data by calculated_methods:

Name	Description
URL	http://<host-name>:8080/restmon/api/v1/schema/{schemaName}/{calculated_methods} https://<host-name>:8443/restmon/api/v1/schema/{schemaName}/{calculated_methods}
Method	POST
Authorization	Basic
Response Code	200 OK
Curl Command	Copy the payload into a file, for example, calculated_methods.json and execute the following command: `curl -k --user <user-name>:<password> -X POST --header 'Content-Type: application/json;charset=UTF-8' --data @calculated_methods.json 'https://localhost:8443/restmon/api/v1/schema/testschema/calculated_methods'`

Sample Request:

```
https://localhost:8443/restmon/api/v1/schema/testschema/calculated_methods
```

Body:

```
{
    "convertBytestoGB": "/ 1073741824;"
}
```

Sample Response:

```
{ "Status:": "calculated_methods is added in schema testschema with fieldId
 15C25200A89149A58CF94748C0821CC0" }
```

Add the Field Data by Change_Events

The following table lists the details required to all the schema data by change_events:

Name	Description
URL	http://<host-name>:8080/restmon/api/v1/schema/{schemaName}/{change_events} https://<host-name>:8443/restmon/api/v1/schema/{schemaName}/{change_events}
Method	POST
Authorization	Basic
Response Code	200 OK

Name	Description
Curl Command	Copy the payload into a file, for example, change_events.json and execute the following command: `curl -k --user <user-name>:<password> -X POST --header 'Content-Type: application/json;charset=UTF-8' --data @change_events.json 'https://localhost:8443/restmon/api/v1/schema/testschema/change_events'`

Sample Request:

```
https://localhost:8443/restmon/api/v1/schema/testschema/change_events
```

Body:

```
{
    "xml_ns": "",
    "url": "changestable",
    "group": "ChangeEvents",
    "attributes": {
        "oi": {
            "timestamp": "%timestamp",
            "startTime": "",
            "host": "$['result'][*]['cmdb_ci']['display_value']",
            "ip": "",
            "status": "$['result'][*]['number']['display_value']",
            "product": "ServiceNow",
            "product_version": "1.0.0",
            "group": [],
            "group_id": "",
            "summary": "$['result'][*]['number']['display_value']",
            "previous_value": "",
            "current_value": "",
            "severity": "$['result'][*]['priority']['value']",
            "severity_conversion": "1:Critical,2:Major,3:Minor,Default:Informational",
            "metric_name": "",
            "metric_type": "",
            "message": "$['result'][*]['number']['display_value']%//%: %//%$['result'][*]['short_description']
['display_value']",
            "change_event_unique_id": "['result'][*]['number']['display_value']",
            "ci_unique_id": "$['result'][*]['cmdb_ci']['display_value']",
            "configuration_item_type": "ServiceNow.Host",
            "configuration_item": "$['result'][*]['cmdb_ci']['display_value']",
            "tags": [
                "ServiceNow",
                "Events",
                "Changes"
            ]
        }
    }
}
```

Sample Response:

```
{ "Status:": "change_events is added in schema testschema with fieldId 9CB08D5A52DC4DEEBEF0A537D4C9CBB9" }
```

Delete the Schema Details by Field Name and Field ID

You can delete the schema details by the following field name and field ID. This method deletes the object as per the provided field name and the field ID.

API	Operation
DELETE /v1/schema/{schemaName}/{urls}/{fieldId}	Delete the schema details by URL and Field ID
DELETE /v1/schema/{schemaName}/{metrics}/{fieldId}	Delete the schema details by Metrics and Field ID
DELETE /v1/schema/{schemaName}/{alarms}/{fieldId}	Delete the schema details by Alarms and Field ID
DELETE /v1/schema/{schemaName}/{inventory}/{fieldId}	Delete the schema details by Inventory and Field ID
DELETE /v1/schema/{schemaName}/{topology}/{fieldId}	Delete the schema details by Topology and Field ID
DELETE /v1/schema/{schemaName}/{groups}/{fieldId}	Delete the schema details by Groups and Field ID
DELETE /v1/schema/{schemaName}/{calculated_metrics}/{fieldId}	Delete the schema details by calculated_metrics and Field ID
DELETE /v1/schema/{schemaName}/{calculated_methods}/{fieldId}	Delete the schema details by calculated_methods and Field ID
DELETE /v1/schema/{schemaName}/{changed_events}/{fieldId}	Delete the schema details by change_events and Field ID

Delete by URLs and Field ID

The following table provides the details required to delete the schema details by URLs and Field ID:

Name	Description
URL	http://<host-name>:8080/restmon/api/v1/schema/{schemaName}/{urls}/{fieldId} https://<host-name>:8443/restmon/api/v1/schema/{schemaName}/{urls}/{fieldId}
Method	DELETE
Authorization	Basic
Response Code	200 OK
Curl Command	`curl -k --user <user-name>:<password> -X DELETE 'https://localhost:8443/restmon/api/v1/schema/testschema/urls/E377E130716D4D2EB2D53D9F36A38804'`

Sample Request:

```
https://localhost:8443/restmon/api/v1/schema/testschema/urls/E377E130716D4D2EB2D53D9F36A38804
```

Sample Response:

```
{ "Status": "urls is deleted with fieldId E377E130716D4D2EB2D53D9F36A38804" }
```

Delete by Metrics and Field ID

The following table provides the details required to delete the schema details by metrics and field ID:

Name	Description
URL	http://<host-name>:8080/restmon/api/v1/schema/{schemaName}/{metrics}/{fieldId} https://<host-name>:8443/restmon/api/v1/schema/{schemaName}/{metrics}/{fieldId}
Method	DELETE

Name	Description
Authorization	Basic
Response Code	200 OK
Curl Command	`curl -k --user <user-name>:<password> -X DELETE 'https://localhost:8443/restmon/api/v1/schema/testschema/metrics/17022273AA2E45769F841206470BFCC1'`

Sample Request:

```
https://localhost:8443/restmon/api/v1/schema/testschema/metrics/17022273AA2E45769F841206470BFCC1
```

Sample Response:

```
{ "Status": "metrics is deleted with fieldId 17022273AA2E45769F841206470BFCC1" }
```

Delete by Alarms and Field ID

The following table provides the details required to delete the details by alarms and field ID:

Name	Description
URL	http://<host-name>:8080/restmon/api/v1/schema/{schemaName}/{alarms}/{fieldId} https://<host-name>:8443/restmon/api/v1/schema/{schemaName}/{alarms}/{fieldId}
Method	DELETE
Authorization	Basic
Response Code	200 OK
Curl Command	`curl -k --user <user-name>:<password> -X DELETE 'https://localhost:8443/restmon/api/v1/schema/testschema/alarms/019943440D014F36A45ECC3B9384DC5D'`

Sample Request:

```
https://localhost:8443/restmon/api/v1/schema/testschema/alarms/019943440D014F36A45ECC3B9384DC5D
```

Sample Response:

```
{ "Status": "alarms is deleted with fieldId 019943440D014F36A45ECC3B9384DC5D
```

Delete by Inventory and Field ID

The following table provides the details required to delete the schema details by inventory and field ID:

Name	Description
URL	http://<host-name>:8080/restmon/api/v1/schema/{schemaName}/{inventory}/{fieldId} https://<host-name>:8443/restmon/api/v1/schema/{schemaName}/{inventory}/{fieldId}
Method	DELETE
Authorization	Basic
Response Code	200 OK
Curl Command	`curl -k --user <user-name>:<password> -X DELETE 'https://localhost:8443/restmon/api/v1/schema/testschema/inventory/197A32D66DDF4246AA960F0E153F26BC'`

Sample Request:

```
https://localhost:8443/restmon/api/v1/schema/testschema/inventory/197A32D66DDF4246AA960F0E153F26BC
```

Sample Response:

```
{ "Status": "inventory is deleted with fieldId 197A32D66DDF4246AA960F0E153F26BC" }
```

Delete by Topology and Field ID

The following table provides the details required to delete the schema details by topology and field ID:

Name	Description
URL	http://<host-name>:8080/restmon/api/v1/schema/{schemaName}/{topology}/{fieldId} https://<host-name>:8443/restmon/api/v1/schema/{schemaName}/{topology}/{fieldId}
Method	DELETE
Authorization	Basic
Response Code	200 OK
Curl Command	`curl -k --user <user-name>:<password> -X DELETE 'https://localhost:8443/restmon/api/v1/schema/testschema/topology/FD92035E695B42D6A73951A1A4984525'`

Sample Request:

```
https://localhost:8443/restmon/api/v1/schema/testschema/topology/FD92035E695B42D6A73951A1A4984525
```

Sample Response:

```
{ "Status": "topology is deleted with fieldId FD92035E695B42D6A73951A1A4984525" }
```

Delete by Groups and Field ID

The following table provides the details required to delete the schema details by groups and field ID:

Name	Description
URL	http://<host-name>:8080/restmon/api/v1/schema/{schemaName}/{groups}/{fieldId} https://<host-name>:8443/restmon/api/v1/schema/{schemaName}/{groups}/{fieldId}
Method	DELETE
Authorization	Basic
Response Code	200 OK
Curl Command	`curl -k --user <user-name>:<password> -X DELETE 'https://localhost:8443/restmon/api/v1/schema/testschema/groups/79F90DB5E48E480FABE38AACEC6FFD03'`

Sample Request:

```
https://localhost:8443/restmon/api/v1/schema/testschema/groups/79F90DB5E48E480FABE38AACEC6FFD03
```

Sample Response:

```
{ "Status": "groups is deleted with fieldId 79F90DB5E48E480FABE38AACEC6FFD03" }
```

Delete by Calculated_Metrics and Field ID

The following table provides the details required to delete the schema data by calculated_metrics and field ID:

Name	Description
URL	http://<host-name>:8080/restmon/api/v1/schema/{schemaName}/{calculated_metrics}/{fieldId} https://<host-name>:8443/restmon/api/v1/schema/{schemaName}/{calculated_metrics}/{fieldId}
Method	DELETE
Authorization	Basic
Response Code	200 OK
Curl Command	`curl -k --user <user-name>:<password> -X DELETE 'https://localhost:8443/restmon/api/v1/schema/testschema/calculated_metrics/A9139757986149CAA72310892A8A09EE'`

Sample Request:

```
https://localhost:8443/restmon/api/v1/schema/testschema/calculated_metrics/A9139757986149CAA72310892A8A09EE
```

Sample Response:

```
{ "Status": "calculated_metrics is deleted with fieldId A9139757986149CAA72310892A8A09EE" }
```

Delete by Calculated_Methods and Field ID

The following table provides the details required to delete the schema details by calculated_methods and field ID:

Name	Description
URL	http://<host-name>:8080/restmon/api/v1/schema/{schemaName}/{calculated_methods}/{fieldId} https://<host-name>:8443/restmon/api/v1/schema/{schemaName}/{calculated_methods}/{fieldId}
Method	DELETE
Authorization	Basic
Response Code	200 OK
Curl Command	`curl -k --user <user-name>:<password> -X DELETE 'https://localhost:8443/restmon/api/v1/schema/testschema/calculated_methods/0CAC3D282C73443B88A8611868C571E5'`

Sample Request:

```
https://localhost:8443/restmon/api/v1/schema/testschema/calculated_methods/0CAC3D282C73443B88A8611868C571E5
```

Sample Response:

```
{ "Status": "calculated_methods is deleted with fieldId 0CAC3D282C73443B88A8611868C571E5" }
```

Delete by Change_Events and Field ID

The following table provides the details required to delete the schema details by change_events and field ID:

Name	Description
URL	http://<host-name>:8080/restmon/api/v1/schema/{schemaName}/{change_events}/{fieldId} https://<host-name>:8443/restmon/api/v1/schema/{schemaName}/{change_events}/{fieldId}
Method	DELETE
Authorization	Basic
Response Code	200 OK
Curl Command	`curl -k --user <user-name>:<password> -X GET 'https://localhost:8443/restmon/api/v1/schema/testschema/change_events/9CB08D5A52DC4DEEBEF0A537D4C9CBB9'`

Sample Request:

```
https://localhost:8443/restmon/api/v1/schema/testschema/change_events/9CB08D5A52DC4DEEBEF0A537D4C9CBB9
```

Sample Response:

```
{ "Status": "change_events is deleted with fieldId 9CB08D5A52DC4DEEBEF0A537D4C9CBB9" }
```

Get the Schema Details By Field Name

You can get the schema details by urls, metrics, alarms, inventory, topology, groups, calculated_metrics, calculated_methods, and change_events.

API	Operation
GET /v1/schema/{schemaName}/{urls}	List the schema details by URL
GET /v1/schema/{schemaName}/{metrics}	List the schema details by Metrics
GET /v1/schema/{schemaName}/{alarms}	List the schema details by Alarms
GET /v1/schema/{schemaName}/{inventory}	List the schema details by Inventory
GET /v1/schema/{schemaName}/{topology}	List the schema details by Topology
GET /v1/schema/{schemaName}/{groups}	List the schema details by Groups
GET /v1/schema/{schemaName}/{calculated_metrics}	List the schema details by calculated_metrics
GET /v1/schema/{schemaName}/{calculated_methods}	List the schema details by calculated_methods
GET /v1/schema/{schemaName}/{changed_events}	List the schema details by change_events

List the Schema Details by URLs

The following table lists the details required to get the schema details by URLs:

Name	Description
URL	http://<host-name>:8080/restmon/api/v1/schema/{schemaName}/{urls} https://<host-name>:8443/restmon/api/v1/schema/{schemaName}/{urls}
Method	GET
Authorization	Basic

Name	Description
Response Code	200 OK
Curl Command	`curl -k --user <user-name>:<password> -X GET 'https://localhost:8443/restmon/api/v1/schema/testschema/urls'`

Sample Request:

```
https://localhost:8443/restmon/api/v1/schema/dynatrace/urls
```

Sample Response:


```
{
  "urls": [
    {
      "xml_ns": "",
      "src": "",
      "var": "",
      "id": "applications",
      "url": "/api/v1/entity/applications",
      "fieldId": "E674153B73C54D04B91D8D94C093A14D"
    },
    {
      "xml_ns": "",
      "src": "",
      "var": "",
      "id": "services",
      "url": "/api/v1/entity/services",
      "fieldId": "59186BE88EEA47489F82020336099318"
    },
    {
      "xml_ns": "",
      "src": "",
      "var": "",
      "id": "processes",
      "url": "/api/v1/entity/infrastructure/processes",
      "fieldId": "CBA82D91FB8243A895BF00F1C8780BBD"
    },
    {
      "xml_ns": "",
      "src": "",
      "var": "",
      "id": "hosts",
      "url": "/api/v1/entity/infrastructure/hosts",
      "fieldId": "3D945E586DC64D32BBB8FD013DED9F83"
    }
  ]
}
```

List the Schema Details by Metrics

The following table lists the details required to get the schema details by metrics:

Name	Description
URL	http://<host-name>:8080/restmon/api/v1/schema/{schemaName}/{metrics} https://<host-name>:8443/restmon/api/v1/schema/{schemaName}/{metrics}
Method	GET
Authorization	Basic
Response Code	200 OK
Curl Command	`curl -k --user <user-name>:<password> -X GET 'https://localhost:8443/restmon/api/v1/schema/testschema/metrics'`

Sample Request:

```
https://localhost:8443/restmon/api/v1/schema/dynatrace/metrics
```

Sample Response:


```
{
    "metrics": [
        {
            "calculation": "$value $convertBytestoGB",
            "attributes": {
                "oi": {
                    "type": "2",
                    "timestamp": "#(function(){var result=[]; for(var key in root.dataResult.dataPoints)
 { if(null !== key) { var metricvalues=root.dataResult.dataPoints[key]; for(var j=metricvalues.length-1;j>=0;
 j--) { if( metricvalues[j][1] !== null) { var metrictimestamp = metricvalues[j][0]; result.push(new
 Date(metrictimestamp).toISOString()); break; } else { continue; } } } } return result;})();",
                    "metric_name": "Disk Available",
                    "metric_type": "Disk|Disk Space Usage",
                    "configuration_item_type": "Host",
                    "metric_unit": "GB",
                    "configuration_item": "#(function(){ var result=[]; var keyValueMap = {}; for(var entry in
 root.dataResult.entities) { var entryObj =root.dataResult.entities; keyValueMap[entry] = entryObj[entry]; }
 for(var key in root.dataResult.dataPoints) { if(null !== key) { var metricname=key.split(','); var hostId =
 metricname[1].trim(); result.push(keyValueMap[hostId]) } } return result;})();",
                    "host": "#(function(){ var result=[]; var keyValueMap = {}; for(var entry in
 root.dataResult.entities) { var entryObj =root.dataResult.entities; keyValueMap[entry] = entryObj[entry]; }
 for(var key in root.dataResult.dataPoints) { if(null !== key) { var metricname=key.split(','); var hostId =
 metricname[0].trim(); result.push(keyValueMap[hostId]) } } return result;})();",
                    "ci_unique_id": "#(function(){var result=[]; for(var key in root.dataResult.dataPoints)
 { if(null !== key) { var metricname=key.split(','); result.push(metricname[0].trim()); } } return result;})
();",
                    "product": "Dynatrace",
                    "product_version": "1.0.0"
                }
            },
```

```
            "value": "#(function(){var result=[]; for(var key in root.dataResult.dataPoints) { if(null !
== key) { var metricvalues=root.dataResult.dataPoints[key]; for(var j=metricvalues.length-1;j>=0; j--)
 { if( metricvalues[j][1] !== null) { var metricValue = metricvalues[j][1]; result.push(metricValue); break; }
 else { continue; } } } } return result;})();",
            "url": "disk-availablespace",
            "condition": "#(function(){var result = '';if(root){return true}else{return false;}})();",
            "group": "Hosts",
            "fieldId": "FFFB07AE746F4418B6052B6B76643179"
        }
    ]
}
```

List the Schema Details by Alarms

The following table lists the details required to get the details by alarms:

Name	Description
URL	http://<host-name>:8080/restmon/api/v1/schema/{schemaName}/{alarms} https://<host-name>:8443/restmon/api/v1/schema/{schemaName}/{alarms}
Method	GET
Authorization	Basic
Response Code	200 OK
Curl Command	`curl -k --user <user-name>:<password> -X GET 'https://localhost:8443/restmon/api/v1/schema/testschema/alarms'`

Sample Request:

```
https://localhost:8443/restmon/api/v1/schema/dynatrace/alarms
```

Sample Response:

```
{
    "alarms": [
        {
            "xml_ns": "",
            "url": "problem",
            "group": "Alarms",
            "attributes": {
                "oi": {
                    "timestamp": "%timestamp",
                    "alarmType": "Infrastructure",
                    "host": "$.result.problems[*].rankedEvents[?(@.eventType == 'SLOW_DISK' &&
@.status=='OPEN')].entityName",
                    "startTime": "$.result.problems[*].rankedEvents[?(@.eventType == 'SLOW_DISK' &&
@.status=='OPEN')].startTime",
                    "status": "",
                    "ip": "",
                    "product": "Dynatrace",
                    "product_version": "",
                    "summary": "Alarms",
                    "severity": "$.result.problems[*].rankedEvents[?(@.eventType == 'SLOW_DISK' &&
@.status=='OPEN')].severityLevel",
```

```
                    "severity_conversion":
 "ERROR:Critical,RESOURCE_CONTENTION:Major,PERFORMANCE:Major,AVAILABILITY:Information,Default:",
                    "metric_name": "Disk Read time",
                    "metric_type": "Disk|Disk Latency",
                    "message": "SLOW_DISK%//%:%//%The Disk Read Time value is greater than the threshold.",
                    "alarm_unique_id": "SLOW_DISK%//%:%//%$.result.problems[*].rankedEvents[?(@.eventType
 == 'SLOW_DISK' && @.status=='OPEN')].entityId%//%:%//%$.result.problems[*].rankedEvents[?(@.eventType ==
 'SLOW_DISK' && @.status=='OPEN')].resourceId",
                    "ci_unique_id": "$.result.problems[*].rankedEvents[?(@.eventType=='SLOW_DISK' &&
 @.status=='OPEN')].entityId",
                    "configuration_item_type": "Host",
                    "configuration_item": "$.result.problems[*].rankedEvents[?(@.eventType == 'SLOW_DISK' &&
 @.status=='OPEN')].resourceName",
                    "alarmURL": "#(function(){var urlArr=[]; var uniqueArray=[]; var sd = url.split('/api');
 var final=''; for (var i =0 ;i< root.result.problems.length;i++) { if( root.result.problems[i].rankedEvents)
 { var rankedEventsArr = root.result.problems[i].rankedEvents; for(var n =0; n<rankedEventsArr.length; n++)
 { if(rankedEventsArr[n].status == 'OPEN' && rankedEventsArr[n].eventType == 'SLOW_DISK') { final = sd[0]+'/
#problems/problemdetails;pid='+root.result.problems[i].id;urlArr.push(final); } } } }return urlArr; })();",
                    "tags": [
                        "Dynatrace",
                        "Alarms",
                        "SLOW_DISK"
                    ]
                }
            },
            "fieldId": "CEBB1EFDAB6943EC91AA8442DB378122"
        }
    ]
}
```

List the Schema Details by Inventory

The following table lists the details required to get the schema details by inventory:

Name	Description
URL	http://<host-name>:8080/restmon/api/v1/schema/{schemaName}/{inventory} https://<host-name>:8443/restmon/api/v1/schema/{schemaName}/{inventory}
Method	GET
Authorization	Basic
Response Code	200 OK
Curl Command	`curl -k --user <user-name>:<password> -X GET 'https://localhost:8443/restmon/api/v1/schema/testschema/inventory'`

Sample Request:

```
https://localhost:8443/restmon/api/v1/schema/dynatrace/inventory
```

Sample Response:

```
{"inventory":[]}
```

List the Schema Details by Topology

The following table lists the details required to get the schema details by topology:

Name	Description
URL	http://<host-name>:8080/restmon/api/v1/schema/{schemaName}/{topology} https://<host-name>:8443/restmon/api/v1/schema/{schemaName}/{topology}
Method	GET
Authorization	Basic
Response Code	200 OK
Curl Command	`curl -k --user <user-name>:<password> -X GET 'https://localhost:8443/restmon/api/v1/schema/testschema/topology'`

Sample Request:

```
https://localhost:8443/restmon/api/v1/schema/dynatrace/topology
```

Sample Response:


```
{
    "topology": [
        {
            "xml_ns": "",
            "url": "hosts",
            "group": "Hosts",
            "layer": "CUSTOM",
            "attributes": {
                "oi": {
                    "name": "$[*]['displayName']",
                    "type": "HOST",
                    "hostname": "$[*]['displayName']",
                    "ci_unique_id": "$[*]['entityId']",
                    "product": "Dynatrace",
                    "entity_id": "$[*]['entityId']",
                    "ipAddresses": "$[*]['ipAddresses']",
                    "child_ci_unique_id": "#(function(){  var parentIds = []; for(var i=0; i<
 root.length;i++) { var parentCisStr=''; if(root[i].toRelationships && root[i].toRelationships.isProcessOf)
 { var parentCis = root[i].toRelationships.isProcessOf; for(var m=0;m<parentCis.length;m++)
 { parentCisStr=parentCisStr + parentCis[m]+'&&'; } } else { parentCisStr=parentCisStr + 'null' + '&&'; }
 parentIds.push(parentCisStr.substring(0,parentCisStr.length-2)) } return parentIds;})();",
                    "semantic_child": "#(function(){ var parentIds = []; for(var i=0; i< root.length;i
++) { var parentCisStr=''; if(root[i].toRelationships && root[i].toRelationships.isProcessOf)
 { var parentCis = root[i].toRelationships.isProcessOf; for(var m=0;m<parentCis.length;m++)
 { parentCisStr=parentCisStr + 'contains'+'&&'; } } else { parentCisStr=parentCisStr + 'null' + '&&'; }
 parentIds.push(parentCisStr.substring(0,parentCisStr.length-2)) } return parentIds;})();",
                    "hashostdetails": "true"
                }
            },
            "fieldId": "E4054E0AD2BF49D08FC1512B045C69E1"
        },
        {
            "xml_ns": "",
```

```
            "url": "processes",
            "group": "Processes",
            "layer": "CUSTOM",
            "attributes": {
                "oi": {
                    "name": "$[*]['displayName']",
                    "type": "PROCESS",
                    "hostname": "",
                    "parent_type": "#(function(){var parentIds = []; for(var i=0; i< root.length;i+
+) { var parentCisStr=''; if(root[i].toRelationships && root[i].toRelationships.isNetworkClientOf)
 { var parentCis2 = root[i].toRelationships.isNetworkClientOf; for(var n=0;n<parentCis2.length;n+
+) { if(parentCis2[n].startsWith('PROCESS')) { parentCisStr=parentCisStr + 'PROCESS' +'&&'; } else
 if(parentCis2[n].startsWith('SERVICE')) { parentCisStr=parentCisStr + 'BUSINESSSERVICE' +'&&'; } } }
 parentIds.push(parentCisStr.substring(0,parentCisStr.length-2)) } return parentIds;})();",
                    "ci_unique_id": "$[*]['entityId']",
                    "parent_ci_unique_id": "#(function(){var parentIds = []; for(var
 i=0; i< root.length;i++) { var parentCisStr=''; if(root[i].toRelationships &&
 root[i].toRelationships.isNetworkClientOf) { var parentCis2 = root[i].toRelationships.isNetworkClientOf;
 for(var n=0;n<parentCis2.length;n++) { parentCisStr=parentCisStr + parentCis2[n]+'&&'; } }
 parentIds.push(parentCisStr.substring(0,parentCisStr.length-2)) } return parentIds;})();",
                    "product": "Dynatrace",
                    "entity_id": "$[*]['entityId']"
                }
            },
            "fieldId": "AD4B9A377658438A8329DCBD7581CB0C"
        }
    ]
}
```

List the Schema Details by Groups

The following table lists the details required to get the schema details by groups:

Name	Description
URL	http://<host-name>:8080/restmon/api/v1/schema/{schemaName}/{groups} https://<host-name>:8443/restmon/api/v1/schema/{schemaName}/{groups}
Method	GET
Authorization	Basic
Response Code	200 OK
Curl Command	`curl -k --user <user-name>:<password> -X GET 'https://localhost:8443/restmon/api/v1/schema/testschema/groups'`

Sample Request:

```
https://localhost:8443/restmon/api/v1/schema/dynatrace/groups
```

Sample Response:

```
{ "groups": [] }
```

List the Schema Details by Calculated_Metrics

The following table lists the details required to get the schema details by calculated_metrics:

Name	Description
URL	http://<host-name>:8080/restmon/api/v1/schema/{schemaName}/{calculated_metrics} https://<host-name>:8443/restmon/api/v1/schema/{schemaName}/{calculated_metrics}
Method	GET
Authorization	Basic
Response Code	200 OK
Curl Command	`curl -k --user <user-name>:<password> -X GET 'https://localhost:8443/restmon/api/v1/schema/testschema/calculated_metrics'`

Sample Request:

```
https://localhost:8443/restmon/api/v1/schema/dynatrace/calculated_metrics
```

Sample Response:

```
{ "calculated_metrics": [] }
```

List the Schema Details by Calculated_Methods

The following table lists the details required to get the schema details by calculated_methods:

Name	Description
URL	http://<host-name>:8080/restmon/api/v1/schema/{schemaName}/{calculated_methods} https://<host-name>:8443/restmon/api/v1/schema/{schemaName}/{calculated_methods}
Method	GET
Authorization	Basic
Response Code	200 OK
Curl Command	`curl -k --user <user-name>:<password> -X GET 'https://localhost:8443/restmon/api/v1/schema/testschema/calculated_methods'`

Sample Request:

```
https://localhost:8443/restmon/api/v1/schema/dynatrace/calculated_methods
```

Sample Response:

```
{
  "calculated_methods": [
    {
      "convertBytestoGB": "/ 1073741824;",
      "fieldId": "262D3E7ACF264B898368DFB34B69F45E"
    },
    {
      "convertBytestoBits": "/ 125;",
      "fieldId": "A318C7E6AA4941108DBD58B81D11FE2E"
    },
    {
      "convertBytestoMB": "/ 1048576;",
```

```
      "fieldId": "84EADA1541064D3AA667D3F1F41DAB49"
    },
    {
      "convertBitsToKiloBits": "/ 1000;",
      "fieldId": "1C13800D981848C4865498EB9BE693D2"
    }
  ]
}
```

List the Schema Details by Change_Events

The following table lists the details required to get the schema details by change_events:

Name	Description
URL	http://<host-name>:8080/restmon/api/v1/schema/{schemaName}/{change_events} https://<host-name>:8443/restmon/api/v1/schema/{schemaName}/{change_events}
Method	GET
Authorization	Basic
Response Code	200 OK
Curl Command	`curl -k --user <user-name>:<password> -X GET 'https://localhost:8443/restmon/api/v1/schema/testschema/change_events'`

Sample Request:

```
https://localhost:8443/restmon/api/v1/schema/dynatrace/change_events
```

Sample Response:

```
{ "change_events": [] }
```

Get the Schema Details By Field Name and Field ID

You can get the schema details by urls, metrics, alarms, inventory, topology, groups, calculated_metrics, calculated_methods, and change_events against the given field ID.

API	Operation
GET /v1/schema/{schemaName}/{urls}/{fieldId}	List the schema details by URL and Field ID
GET /v1/schema/{schemaName}/{metrics}/{fieldId}	List the schema details by Metrics and Field ID
GET /v1/schema/{schemaName}/{alarms}/{fieldId}	List the schema details by Alarms and Field ID
GET /v1/schema/{schemaName}/{inventory}/{fieldId}	List the schema details by Inventory and Field ID
GET /v1/schema/{schemaName}/{topology}/{fieldId}	List the schema details by Topology and Field ID
GET /v1/schema/{schemaName}/{groups}/{fieldId}	List the schema details by Groups and Field ID
GET /v1/schema/{schemaName}/{calculated_metrics}/{fieldId}	List the schema details by calculated_metrics and Field ID
GET /v1/schema/{schemaName}/{calculated_methods}/{fieldId}	List the schema details by calculated_methods and Field ID
GET /v1/schema/{schemaName}/{changed_events}/{fieldId}	List the schema details by change_events and Field ID

List the Schema Details by URLs and Field ID

The following table lists the details required to get the schema details by URLs and Field ID:

Name	Description
URL	http://<host-name>:8080/restmon/api/v1/schema/{schemaName}/{urls}/{fieldId} https://<host-name>:8443/restmon/api/v1/schema/{schemaName}/{urls}/{fieldId}
Method	GET
Authorization	Basic
Response Code	200 OK
Curl Command	`curl -k --user <user-name>:<password> -X GET 'https://localhost:8443/restmon/api/v1/schema/testschema/urls/9E66EBD9625141A7B3C53E7A1A784B47'`

Sample Request:

```
https://localhost:8443/restmon/api/v1/schema/testschema/urls/9E66EBD9625141A7B3C53E7A1A784B47
```

Sample Response:

```
{
  "src": "hosts",
  "xml_ns": "",
  "var": "#(function(){var date = new Date();var fromDate=parseInt(date.getTime()-(%%interval
%%*1000)).toString();fromDate = Math.round(fromDate / 1000);var date2=new Date();var
 toDate=parseInt(date.getTime());toDate=Math.round(toDate/1000);var finalString='&from='+fromDate
+'&to='+toDate;return finalString;})();",
  "id": "memory-cached",
  "url": "/api/v1/query?&query=avg:system.mem.cached{*}by{host}%var",
  "fieldId": "9E66EBD9625141A7B3C53E7A1A784B47"
}
```

List the Schema Details by Metrics and Field ID

The following table lists the details required to get the schema details by metrics and field ID:

Name	Description
URL	http://<host-name>:8080/restmon/api/v1/schema/{schemaName}/{metrics}/{fieldId} https://<host-name>:8443/restmon/api/v1/schema/{schemaName}/{metrics}/{fieldId}
Method	GET
Authorization	Basic
Response Code	200 OK
Curl Command	`curl -k --user <user-name>:<password> -X GET 'https://localhost:8443/restmon/api/v1/schema/testschema/metrics/D7BFBDA179B84083A590EE9B26CE8100'`

Sample Request:

```
https://localhost:8443/restmon/api/v1/schema/testschema/metrics/D7BFBDA179B84083A590EE9B26CE8100
```

Sample Response:

```
{
```

```
  "attributes": {
    "oi": {
      "metric_name": "$['series'][*]['metric']",
      "metric_type": "Network Traffic",
      "metric_unique_id": "$['series'][*]['metric']%//%:%//%$['series'][*]['scope']",
      "metric_unit": "pkts/sec",
      "timestamp": "#(function(){var result=[];for(var i=0;i<root.series.length;i++){var
 timeLength=root.series[i].pointlist.length-1;var newTime=root.series[i].pointlist[timeLength]
[0];result.push(newTime.toString());}return result;})();",
      "configuration_item": "$['series'][*]['tag_set'][0]",
      "configuration_item_type": "Host",
      "host": "#(function(){var result=[];for(var i=0;i<root.series.length;i++){var
 firstSplit=root.series[i].scope.split(',');var
 secondSplit=firstSplit[1].split(':');result.push(secondSplit[1]);}return result;})();",
      "product": "DataDog",
      "type": "2",
      "product_version": "1.0.0",
      "ci_unique_id": "#(function(){var result=[];for(var i=0;i<root.series.length;i++){var
 firstSplit=root.series[i].scope.split(',');var
 secondSplit=firstSplit[1].split(':');result.push(secondSplit[1]);}return result;})();",
      "tags": [
        "DataDog",
        "System Metrics",
        "$['series'][*]['metric']"
      ]
    }
  },
  "value": "#(function(){var result=[];for(var i=0;i<root.series.length;i++){var
 pointListLength=root.series[i].length-1;result.push(root.series[i].pointlist[pointListLength]
[1].toFixed(2));}return result;})();",
  "url": "system-pkts-out-error",
  "group": "Metrics",
  "fieldId": "D7BFBDA179B84083A590EE9B26CE8100"
}
```


List the Schema Details by Alarms and Field ID

The following table lists the details required to get the details by alarms and field ID:

Name	Description
URL	http://<host-name>:8080/restmon/api/v1/schema/{schemaName}/{alarms}/{fieldId} https://<host-name>:8443/restmon/api/v1/schema/{schemaName}/{alarms}/{fieldId}
Method	GET
Authorization	Basic
Response Code	200 OK
Curl Command	`curl -k --user <user-name>:<password> -X GET 'https://localhost:8443/restmon/api/v1/schema/testschema/alarms/5FB532F524B94819B5D2AECCAF90F456'`

Sample Request:

```
https://localhost:8443/restmon/api/v1/schema/testschema/alarms/5FB532F524B94819B5D2AECCAF90F456
```

Sample Response:


```
{
    "xml_ns": "",
    "url": "alert-details",
    "group": "Alarms",
    "attributes": {
        "oi": {
            "timestamp": "%timestamp",
            "host": "$[*]['host']",
            "product": "DataDog",
            "product_version": "1.0.0",
            "startTime": "#(function(){var result=[];var payloadString=new
 Date(root.event.date_happened*1000).toISOString();result.push(payloadString); return result;})();",
            "summary": "Alarms",
            "severity": "$['event']['alert_type']",
            "severity_conversion": "error:Critical,warning:Minor,Default:",
            "metric_name": "CPU Percentage",
            "metric_type": "",
            "configuration_item_type": "Host",
            "configuration_item": "",
            "message": "$[*]['alarm_message']",
            "alarm_unique_id": "$[*]['alarm_id']",
            "status": "",
            "tags": [
                "DataDog",
                "Alarms"
            ],
            "ci_unique_id": "$[*]['host']",
            "alarmURL": "#(function(){var test='';var result=[];var jsonString =
 JSON.parse(root.event.payload);var alertString='';if(jsonString.result_aggr){alertString =
 jsonString.result_aggr.result_metadata.alert_url;}else{alertString=jsonString.result.metadata.alert_url;}if(alertStrin
{result.push('https://app.datadoghq.com'+alertString);}else{result.push('https://
app.datadoghq.com'+alertString);}return result;})();",
            "alarmType": "Infrastructure",
            "condition": "#(function(){if(root.event.host){return true;}else{return false;}})();"
        }
    },
    "fieldId": "5FB532F524B94819B5D2AECCAF90F456"
}
```


List the Schema Details by Inventory and Field ID

The following table lists the details required to get the schema details by inventory and field ID:

Name	Description
URL	http://<host-name>:8080/restmon/api/v1/schema/{schemaName}/{inventory}/{fieldId} https://<host-name>:8443/restmon/api/v1/schema/{schemaName}/{inventory}/{fieldId}
Method	GET
Authorization	Basic

Name	Description
Response Code	200 OK
Curl Command	`curl -k --user <user-name>:<password> -X GET 'https://localhost:8443/restmon/api/v1/schema/testschema/inventory/7524330BDD60403EB00E9AEDF9B6BD78'`

Sample Request:

```
https://localhost:8443/restmon/api/v1/schema/testschema/inventory/7524330BDD60403EB00E9AEDF9B6BD78
```

Sample Response:


```
{
  "xml_ns": "",
  "url": "hosts",
  "group": "Hosts_Inventory",
  "attributes": {
    "oi": {
      "ci_unique_id": "$['host_list'][*]['name']",
      "host": "$['host_list'][*]['name']",
      "product": "DataDog",
      "configuration_item": "$['host_list'][*]['name']",
      "product_version": "1.0.0",
      "tags": [
        "DataDog",
        "Hosts",
        "$['host_list'][*]['name']",
        "ExcludeFromTAS"
      ],
      "display_name": "$['host_list'][*]['name']",
      "configuration_item_type": "HOST",
      "type": "Host"
    }
  },
  "fieldId": "7524330BDD60403EB00E9AEDF9B6BD78"
}
```


List the Schema Details by Topology and Field ID

The following table lists the details required to get the schema details by topology and field ID:

Name	Description
URL	http://<host-name>:8080/restmon/api/v1/schema/{schemaName}/{topology}/{fieldId} https://<host-name>:8443/restmon/api/v1/schema/{schemaName}/{topology}/{fieldId}
Method	GET
Authorization	Basic
Response Code	200 OK
Curl Command	`curl -k --user <user-name>:<password> -X GET 'https://localhost:8443/restmon/api/v1/schema/testschema/topology/957866A8EC754E579B5FDF0974152234'`

Sample Request:

```
https://localhost:8443/restmon/api/v1/schema/testschema/topology/957866A8EC754E579B5FDF0974152234
```

Sample Response:


```
{
  "xml_ns": "",
  "url": "",
  "group": "Topology",
  "layer": "CUSTOM",
  "attributes": {
    "oi": {
      "ci_unique_id": "$['FQDN']",
      "type": "HOST",
      "product": "Netcool/Omnibus",
      "name": "$['CI Name']",
      "hostname": "$['FQDN']",
      "Display Name": "$['CI Name']",
      "ipAddresses": "#(function() {return root['IP Address'].split(' ').join('');})();",
      "macAddresses": "#(function() {return root['MAC
 Address'].split('.').join(':').split('-').join(':').split(' ').join('');})();",
      "FQDNHostname": "$['FQDN']",
      "it_service_instance": "$['IT Service Instance']",
      "environment": "$['Environment']]",
      "infrastructure_itsi": "$['Infrastructure ITSI']",
      "class_label": "$['Class Label']",
      "itsi_support_group": "$['ITSI Support Group']",
      "infra_itsi_support_group": "$['Infra ITSI Support Group']"
    }
  },
  "fieldId": "957866A8EC754E579B5FDF0974152234"
}
```


List the Schema Details by Groups and Field ID

The following table lists the details required to get the schema details by groups and field ID:

Name	Description
URL	http://<host-name>:8080/restmon/api/v1/schema/{schemaName}/{groups}/{fieldId} https://<host-name>:8443/restmon/api/v1/schema/{schemaName}/{groups}/{fieldId}
Method	GET
Authorization	Basic
Response Code	200 OK
Curl Command	`curl -k --user <user-name>:<password> -X GET 'https://localhost:8443/restmon/api/v1/schema/testschema/groups/5D3341456C4A487782E5E08C6DFFDD15'`

Sample Request:

```
https://localhost:8443/restmon/api/v1/schema/testschema/groups/5D3341456C4A487782E5E08C6DFFDD15
```

Sample Response:

```
{
  "xml_ns": "",
  "url": "hosts",
  "group": "host-group",
  "layer": "GROUPS_CUSTOM",
  "attributes": {
    "oi": {
      "group_id": "$[*]['hostGroup']['meId']",
      "name": "$[*]['hostGroup']['name']",
      "groupname": "$[*]['hostGroup']['name']",
      "product": "Dynatrace",
      "type": "HOST_GROUP"
    }
  },
  "fieldId": "5D3341456C4A487782E5E08C6DFFDD15"
}
```

List the Schema Details by Calculated_Metrics and Field ID

The following table lists the details required to get the schema data by calculated_metrics and field ID:

Name	Description
URL	http://<host-name>:8080/restmon/api/v1/schema/{schemaName}/{calculated_metrics}/{fieldId} https://<host-name>:8443/restmon/api/v1/schema/{schemaName}/{calculated_metrics}/{fieldId}
Method	GET
Authorization	Basic
Response Code	200 OK
Curl Command	`curl -k --user <user-name>:<password> -X GET 'https://localhost:8443/restmon/api/v1/schema/testschema/calculated_metrics/3E6EE5A7FEBF4DCF8DBA8D2187A967E8'`

Sample Request:

```
https://localhost:8443/restmon/api/v1/schema/testschema/calculated_metrics/3E6EE5A7FEBF4DCF8DBA8D2187A967E8
```

Sample Response:

```
{
  "calculation": "($total_provisioned - $total_free) / 1048576",
  "xml_ns": "",
  "values": [
    {
      "name": "$total_free",
      "value": "$['totalFree_gb']"
    },
    {
      "name": "$total_provisioned",
      "value": "$['totalProvisioned_gb']"
    }
  ],
```

```
  "group": "CapacityMetrics",
  "attributes": {
    "oi": {
      "metric_name": "Total Used",
      "metric_type": "Cluster Capacity",
      "metric_unique_id": "%clusterName%//%:%//%TotalUsed",
      "metric_unit": "PB",
      "type": "1",
      "configuration_item_type": "ECS.Capacity",
      "configuration_item": "%clusterName",
      "ci_unique_id": "%clusterName",
      "host": "%clusterName",
      "ip": "",
      "product": "ECS",
      "product_version": "1.17",
      "tags": [
        "ECS",
        "Total Used",
        "Capacity: Total Used",
        "%clusterName"
      ]
    }
  },
  "url": "capacity",
  "fieldId": "3E6EE5A7FEBF4DCF8DBA8D2187A967E8"
}
```


List the Schema Details by Calculated_Methods and Field ID

The following table lists the details required to get the schema details by calculated_methods and field ID:

Name	Description
URL	http://<host-name>:8080/restmon/api/v1/schema/{schemaName}/{calculated_methods}/{fieldId} https://<host-name>:8443/restmon/api/v1/schema/{schemaName}/{calculated_methods}/{fieldId}
Method	GET
Authorization	Basic
Response Code	200 OK
Curl Command	`curl -k --user <user-name>:<password> -X GET 'https://localhost:8443/restmon/api/v1/schema/testschema/calculated_methods/15C25200A89149A58CF94748C0821CC0'`

Sample Request:

```
https://localhost:8443/restmon/api/v1/schema/testschema/calculated_methods/15C25200A89149A58CF94748C0821CC0
```

Sample Response:


```
{ "convertBytestoGB": "/ 1073741824;", "fieldId": "15C25200A89149A58CF94748C0821CC0" }
```

List the Schema Details by Change_Events and Field ID

The following table lists the details required to get the schema details by change_events and field ID:

Name	Description
URL	http://<host-name>:8080/restmon/api/v1/schema/{schemaName}/{change_events}/{fieldId} https://<host-name>:8443/restmon/api/v1/schema/{schemaName}/{change_events}/{fieldId}
Method	GET
Authorization	Basic
Response Code	200 OK
Curl Command	`curl -k --user <user-name>:<password> -X GET 'https://localhost:8443/restmon/api/v1/schema/testschema/change_events/318A22B95C7A4C92A9A286E4F1AAF9B4'`

Sample Request:

```
https://localhost:8443/restmon/api/v1/schema/testschema/change_events/318A22B95C7A4C92A9A286E4F1AAF9B4
```

Sample Response:


```
{
  "xml_ns": "",
  "url": "changestable",
  "group": "ChangeEvents",
  "attributes": {
    "oi": {
      "timestamp": "#(function() { var result=[];var tz='PDT'; for(var i in root.result) { var temp=new
 Date(root.result[i].work_end.display_value + ' ' + tz); result.push(temp.toJSON()) } return result; })();",
      "startTime": "#(function() { var result=[];var tz='PDT'; for(var i in root.result) { var temp=new
 Date(root.result[i].work_start.display_value + ' ' + tz); result.push(temp.toJSON()) } return result; })();",
      "host": "$['result'][*]['cmdb_ci']['display_value']",
      "ip": "",
      "status": "$['result'][*]['number']['display_value']",
      "product": "ServiceNow",
      "product_version": "1.0.0",
      "group": [],
      "group_id": "",
      "summary": "$['result'][*]['number']['display_value']",
      "previous_value": "",
      "current_value": "",
      "severity": "$['result'][*]['priority']['value']",
      "severity_conversion": "1:Critical,2:Major,3:Minor,Default:Informational",
      "metric_name": "",
      "metric_type": "",
      "message": "$['result'][*]['number']['display_value']%//%: %//%$['result'][*]['short_description']
['display_value']",
      "change_event_unique_id": "['result'][*]['number']['display_value']",
      "ci_unique_id": "$['result'][*]['cmdb_ci']['display_value']",
      "configuration_item_type": "ServiceNow.Host",
      "configuration_item": "$['result'][*]['cmdb_ci']['display_value']",
      "tags": [
```

```
        "ServiceNow",
        "Events",
        "Changes"
      ]
    }
  },
  "fieldId": "318A22B95C7A4C92A9A286E4F1AAF9B4"
}
```

Update Schema Details by Field Name and Field ID

You can modify the schema details by urls, metrics, alarms, inventory, topology, groups, calculated_metrics, calculated_methods, and change_events and field IDs. This method modifies the object in the provided field name and field ID.

NOTE

- Single json block per entity is allowed.
- FieldID is a mandatory field in the payload.

API	Operation
PUT /v1/schema/{schemaName}/{urls}/{fieldId}	Update the schema details by URL and Field ID
PUT /v1/schema/{schemaName}/{metrics}/{fieldId}	Update the schema details by Metrics and Field ID
PUT /v1/schema/{schemaName}/{alarms}/{fieldId}	Update the schema details by Alarms and Field ID
PUT /v1/schema/{schemaName}/{inventory}/{fieldId}	Update the schema details by Inventory and Field ID
PUT /v1/schema/{schemaName}/{topology}/{fieldId}	Update the schema details by Topology and Field ID
PUT /v1/schema/{schemaName}/{groups}/{fieldId}	Update the schema details by Groups and Field ID
PUT /v1/schema/{schemaName}/{calculated_metrics}/{fieldId}	Update the schema details by calculated_metrics and Field ID
PUT /v1/schema/{schemaName}/{calculated_methods}/{fieldId}	Update the schema details by calculated_methods and Field ID
PUT /v1/schema/{schemaName}/{changed_events}/{fieldId}	Update the schema details by change_events and Field ID

Update the Schema Details by URLs and Field ID

The following table lists the details required to update the schema details by URL and Field ID:

Name	Description
URL	http://<host-name>:8080/restmon/api/v1/schema/{schemaName}/{urls}/{fieldId} https://<host-name>:8443/restmon/api/v1/schema/{schemaName}/{urls}/{fieldId}
Method	PUT
Authorization	Basic
Response Code	200 OK
Curl Command	Copy the sample payload into a file, for example, **url.json** and execute the following command: `curl -k --user <user-name>:<password> -X PUT --header 'Content-Type: application/json;charset=UTF-8' --data @url.json 'https://localhost:8443/restmon/api/v1/schema/testschema/urls/9E66EBD9625141A7B3C53E7A1A784B47'`

Sample Request:

```
https://localhost:8443/restmon/api/v1/schema/testschema/urls/9E66EBD9625141A7B3C53E7A1A784B47
```

Body:

```
{
    "src": "hosts",
    "xml_ns": "",
    "var": "#(function(){var date = new Date();var fromDate=parseInt(date.PUTTime()-(%
%interval%%*1000)).toString();fromDate = Math.round(fromDate / 1000);var date2=new Date();var
 toDate=parseInt(date.PUTTime());toDate=Math.round(toDate/1000);var finalString='&from='+fromDate
+'&to='+toDate;return finalString;})();",
    "id": "memory-cached",
    "url": "/api/v1/query?&query=avg:system.mem.cached{*}by{host}%var",
    "fieldId": "9E66EBD9625141A7B3C53E7A1A784B47"
}
```

Sample Response:

```
{ "Status": "urls is updated with fieldId 9E66EBD9625141A7B3C53E7A1A784B47" }
```

Update the Schema Details by Metrics and Field ID

The following table lists the details required to update the schema details by metrics and field ID:

Name	Description
URL	http://<host-name>:8080/restmon/api/v1/schema/{schemaName}/{metrics}/{fieldId} https://<host-name>:8443/restmon/api/v1/schema/{schemaName}/{metrics}/{fieldId}
Method	PUT
Authorization	Basic
Response Code	200 OK
Curl Command	Copy the sample payload into a file, for example, **metrics.json** and execute the following command: `curl -k --user <user-name>:<password> -X PUT --header 'Content-Type: application/json;charset=UTF-8' --data @metric.json 'https://localhost:8443/restmon/api/v1/schema/testschema/metrics/D7BFBDA179B84083A590EE9B26CE8100'`

Sample Request:

```
https://localhost:8443/restmon/api/v1/schema/testschema/metrics/D7BFBDA179B84083A590EE9B26CE8100
```

Body:

```
{
    "attributes": {
        "oi": {
            "metric_name": "$['series'][*]['metric']",
            "metric_type": "Network Traffic",
            "metric_unique_id": "$['series'][*]['metric']%//%:%//%$['series'][*]['scope']",
            "metric_unit": "pkts/sec",
```

```
            "timestamp": "#(function(){var result=[];for(var i=0;i<root.series.length;i++){var
 timeLength=root.series[i].pointlist.length-1;var newTime=root.series[i].pointlist[timeLength]
[0];result.push(newTime.toString());}return result;})();",
            "configuration_item": "$['series'][*]['tag_set'][0]",
            "configuration_item_type": "Host",
            "host": "#(function(){var result=[];for(var i=0;i<root.series.length;i++){var
 firstSplit=root.series[i].scope.split(',');var
 secondSplit=firstSplit[1].split(':');result.push(secondSplit[1]);}return result;})();",
            "product": "DataDog",
            "type": "2",
            "product_version": "1.0.0",
            "ci_unique_id": "#(function(){var result=[];for(var i=0;i<root.series.length;i++){var
 firstSplit=root.series[i].scope.split(',');var
 secondSplit=firstSplit[1].split(':');result.push(secondSplit[1]);}return result;})();",
            "tags": [
                "DataDog",
                "System Metrics",
                "$['series'][*]['metric']"
            ]
        }
    },
    "value": "#(function(){var result=[];for(var i=0;i<root.series.length;i++){var
 pointListLength=root.series[i].length-1;result.push(root.series[i].pointlist[pointListLength]
[1].toFixed(2));}return result;})();",
    "url": "system-pkts-out-error",
    "group": "Metrics",
    "fieldId": "D7BFBDA179B84083A590EE9B26CE8100"
}
```

Sample Response:

```
{ "Status": "metrics is updated with fieldId D7BFBDA179B84083A590EE9B26CE8100" }
```

Update the Schema Details by Alarms and Field ID

The following table lists the details required to update the details by alarms and field ID:

Name	Description
URL	http://<host-name>:8080/restmon/api/v1/schema/{schemaName}/{alarms}/{fieldId} https://<host-name>:8443/restmon/api/v1/schema/{schemaName}/{alarms}/{fieldId}
Method	PUT
Authorization	Basic
Response Code	200 OK
Curl Command	Copy the sample payload into a file, for example, **alarm.json** and execute the following command: `curl -k --user <user-name>:<password> -X PUT --header 'Content-Type: application/json;charset=UTF-8' --data @alarm.json 'https://localhost:8443/restmon/api/v1/schema/testschema/alarms/5FB532F524B94819B5D2AECCAF90F456'`

Sample Request:

```
https://localhost:8443/restmon/api/v1/schema/testschema/alarms/5FB532F524B94819B5D2AECCAF90F456
```

Body:


```
{
    "xml_ns": "",
    "url": "alert-details",
    "group": "Alarms",
    "attributes": {
        "oi": {
            "timestamp": "%timestamp",
            "host": "$[*]['host']",
            "product": "DataDog",
            "product_version": "1.0.0",
            "startTime": "#(function(){var result=[];var payloadString=new
 Date(root.event.date_happened*1000).toISOString();result.push(payloadString); return result;})();",
            "summary": "Alarms",
            "severity": "$['event']['alert_type']",
            "severity_conversion": "error:Critical,warning:Minor,Default:",
            "metric_name": "I/O stat",
            "metric_type": "Input/Output Statistics",
            "configuration_item_type": "Host",
            "configuration_item": "",
            "message": "$[*]['alarm_message']",
            "alarm_unique_id": "$[*]['alarm_id']",
            "status": "",
            "tags": [
                "DataDog",
                "Alarms"
            ],
            "ci_unique_id": "$[*]['host']",
            "alarmURL": "#(function(){var test='';var result=[];var jsonString =
 JSON.parse(root.event.payload);var alertString='';if(jsonString.result_aggr){alertString =
 jsonString.result_aggr.result_metadata.alert_url;}else{alertString=jsonString.result.metadata.alert_url;}if(alertStrin
{result.push('https://app.datadoghq.com'+alertString);}else{result.push('https://
app.datadoghq.com'+alertString);}return result;})();",
            "alarmType": "Infrastructure",
            "condition": "#(function(){if(root.event.host){return true;}else{return false;}})();"
        }
    },
    "fieldId": "5FB532F524B94819B5D2AECCAF90F456"
}
```


Sample Response:


```
{ "Status": "alarms is updated with fieldId 5FB532F524B94819B5D2AECCAF90F456" }
```

Update the Schema Details by Inventory and Field ID

The following table lists the details required to update the schema details by inventory and field ID:

Name	Description
URL	http://<host-name>:8080/restmon/api/v1/schema/{schemaName}/{inventory}/{fieldId} https://<host-name>:8443/restmon/api/v1/schema/{schemaName}/{inventory}/{fieldId}
Method	PUT
Authorization	Basic
Response Code	200 OK
Curl Command	Copy the sample payload into a file, for example, **inventory.json** and execute the following command: `curl -k --user <user-name>:<password> -X PUT --header 'Content-Type: application/json;charset=UTF-8' --data @inventory.json 'https://localhost:8443/restmon/api/v1/schema/testschema/inventory/7524330BDD60403EB00E9AEDF9B6BD78'`

Sample Request:

```
https://localhost:8443/restmon/api/v1/schema/testschema/inventory/7524330BDD60403EB00E9AEDF9B6BD78
```

Body:

```
{
    "xml_ns": "",
    "url": "hosts",
    "group": "Hosts_Inventory",
    "attributes": {
        "oi": {
            "ci_unique_id": "$['host_list'][*]['name']",
            "host": "$['host_list'][*]['name']",
            "product": "DataDog",
            "configuration_item": "$['host_list'][*]['name']",
            "product_version": "1.0.0",
            "tags": [
                "DataDog",
                "Hosts",
                "$['host_list'][*]['name']",
                "ExcludeFromTAS"
            ],
            "display_name": "$['host_list'][*]['name']",
            "configuration_item_type": "HOST",
            "type": "Host"
        }
    },
    "fieldId": "7524330BDD60403EB00E9AEDF9B6BD78"
}
```

Sample Response:

```
{ "Status": "inventory is updated with fieldId 7524330BDD60403EB00E9AEDF9B6BD78" }
```

Update the Schema Details by Topology and Field ID

The following table lists the details required to update the schema details by topology and field ID:

Name	Description
URL	http://<host-name>:8080/restmon/api/v1/schema/{schemaName}/{topology}/{fieldId} https://<host-name>:8443/restmon/api/v1/schema/{schemaName}/{topology}/{fieldId}
Method	PUT
Authorization	Basic
Response Code	200 OK
Curl Command	Copy the sample payload into a file, for example, **topology.json** and execute the following command: `curl -k --user <user-name>:<password> -X PUT --header 'Content-Type: application/json;charset=UTF-8' --data @topology.json 'https://localhost:8443/restmon/api/v1/schema/testschema/topology/957866A8EC754E579B5FDF0974152234'`

Sample Request:

```
https://localhost:8443/restmon/api/v1/schema/testschema/topology/957866A8EC754E579B5FDF0974152234
```

Body:

```
{
    "xml_ns": "",
    "url": "",
    "group": "Topology",
    "layer": "CUSTOM",
    "attributes": {
        "oi": {
            "ci_unique_id": "$['FQDN']",
            "type": "HOST",
            "product": "Netcool/Omnibus",
            "name": "$['CI Name']",
            "hostname": "$['FQDN']",
            "Display Name": "$['CI Name']",
            "ipAddresses": "#(function() {return root['IP Address'].split(' ').join('');})();",
            "macAddresses": "#(function() {return root['MAC
 Address'].split('.').join(':').split('-').join(':').split(' ').join('');})();",
            "FQDNHostname": "$['FQDN']",
            "it_service_instance": "$['IT Service Instance']",
            "environment": "$['Environment']]",
            "infrastructure_itsi": "$['Infrastructure ITSI']",
            "class_label": "$['Class Label']",
            "itsi_support_group": "$['ITSI Support Group']",
            "infra_itsi_support_group": "$['Infra ITSI Support Group']"
        }
    },
    "fieldId": "957866A8EC754E579B5FDF0974152234"
}
```

Sample Response:

```
{ "Status:": "topology is added in schema testschema with fieldId FD92035E695B42D6A73951A1A4984525" }
```

Update the Schema Details by Groups and Field ID

The following table lists the details required to update the schema details by groups and field ID:

Name	Description
URL	http://<host-name>:8080/restmon/api/v1/schema/{schemaName}/{groups}/{fieldId} https://<host-name>:8443/restmon/api/v1/schema/{schemaName}/{groups}/{fieldId}
Method	PUT
Authorization	Basic
Response Code	200 OK
Curl Command	Copy the sample payload into a file, for example, **groups.json** and execute the following command: `curCopy the above payload into a file, for example, groups.json and execute the below command:l -k --user <user-name>:<password> -X PUT --header 'Content-Type: application/json;charset=UTF-8' --data @groups.json 'https://localhost:8443/restmon/api/v1/schema/testschema/groups/5D3341456C4A487782E5E08C6DFFDD15'`

Sample Request:

```
https://localhost:8443/restmon/api/v1/schema/testschema/groups/5D3341456C4A487782E5E08C6DFFDD15
```

Body:

```
{
    "xml_ns": "",
    "url": "hosts",
    "group": "host-group",
    "layer": "GROUPS_CUSTOM",
    "attributes": {
        "oi": {
            "group_id": "$[*]['hostGroup']['meId']",
            "name": "$[*]['hostGroup']['name']",
            "groupname": "$[*]['hostGroup']['name']",
            "product": "Dynatrace",
            "type": "HOST_GROUP"
        }
    },
    "fieldId": "5D3341456C4A487782E5E08C6DFFDD15"
}
```

Sample Response:

```
{ "Status": "groups is updated with fieldId 5D3341456C4A487782E5E08C6DFFDD15" }
```

Update the Schema Details by Calculated_Metrics and Field ID

The following table lists the details required to update the schema data by calculated_metrics and field ID:

Name	Description
URL	http://<host-name>:8080/restmon/api/v1/schema/{schemaName}/{calculated_metrics}/{fieldId} https://<host-name>:8443/restmon/api/v1/schema/{schemaName}/{calculated_metrics}/{fieldId}
Method	PUT
Authorization	Basic
Response Code	200 OK
Curl Command	Copy the sample payload into a file, for example, **calculated_metrics.json** and execute the following command: `curl -k --user <user-name>:<password> -X PUT --header 'Content-Type: application/json;charset=UTF-8' --data @calculated_metrics.json 'https://localhost:8443/restmon/api/v1/schema/testschema/calculated_metrics/3E6EE5A7FEBF4DCF8DBA8D2187A967E8'`

Sample Request:

```
https://localhost:8443/restmon/api/v1/schema/testschema/calculated_metrics/3E6EE5A7FEBF4DCF8DBA8D2187A967E8
```

Body:

```
{
    "calculation": "($total_provisioned - $total_free) / 1048576",
    "xml_ns": "",
    "values": [
        {
            "name": "$total_free",
            "value": "$['totalFree_gb']"
        },
        {
            "name": "$total_provisioned",
            "value": "$['totalProvisioned_gb']"
        }
    ],
    "group": "CapacityMetrics",
    "attributes": {
        "oi": {
            "metric_name": "Total Used",
            "metric_type": "Cluster Capacity",
            "metric_unique_id": "%clusterName%//%:%//%TotalUsed",
            "metric_unit": "PB",
            "type": "1",
            "configuration_item_type": "ECS.Capacity",
            "configuration_item": "%clusterName",
            "ci_unique_id": "%clusterName",
            "host": "%clusterName",
            "ip": "",
            "product": "ECS",
            "product_version": "1.17",
```

```
            "tags": [
                "ECS",
                "Total Used",
                "Capacity: Total Used",
                "%clusterName"
            ]
        }
    },
    "url": "capacity",
    "fieldId": "3E6EE5A7FEBF4DCF8DBA8D2187A967E8"
}
```

Sample Response:

```
{ "Status": "calculated_metrics is updated with fieldId 3E6EE5A7FEBF4DCF8DBA8D2187A967E8" }
```

Update the Schema Details by Calculated_Methods and Field ID

The following table lists the details required to update the schema details by calculated_methods and field ID:

Name	Description
URL	http://<host-name>:8080/restmon/api/v1/schema/{schemaName}/{calculated_methods}/{fieldId} https://<host-name>:8443/restmon/api/v1/schema/{schemaName}/{calculated_methods}/{fieldId}
Method	PUT
Authorization	Basic
Response Code	200 OK
Curl Command	Copy the sample payload into a file, for example, **calculated_methods.json** and execute the following command: `curl -k --user <user-name>:<password> -X PUT --header 'Content-Type: application/json;charset=UTF-8' --data @calculated_methods.json 'https://localhost:8443/restmon/api/v1/schema/testschema/calculated_methods/15C25200A89149A58CF94748C0821CC0'`

Sample Request:

```
https://localhost:8443/restmon/api/v1/schema/testschema/calculated_methods/15C25200A89149A58CF94748C0821CC0
```

Body:

```
{
    "convertBytestoGB": "/ 1073741824;",
    "fieldId": "15C25200A89149A58CF94748C0821CC0"
}
```

Sample Response:

```
{ "Status": "calculated_methods is updated with fieldId 15C25200A89149A58CF94748C0821CC0" }
```

Update the Schema Details by Change_Events and Field ID

The following table lists the details required to PUT the schema details by change_events and field ID:

Name	Description
URL	http://<host-name>:8080/restmon/api/v1/schema/{schemaName}/{change_events}/{fieldId} https://<host-name>:8443/restmon/api/v1/schema/{schemaName}/{change_events}/{fieldId}
Method	PUT
Authorization	Basic
Response Code	200 OK
Curl Command	Copy the sample payload into a file, for example, **change_events.json** and execute the following command: `curl -k --user <user-name>:<password> -X PUT --header 'Content-Type: application/json;charset=UTF-8' --data @change_events.json 'https://localhost:8443/restmon/api/v1/schema/testschema/change_events/318A22B95C7A4C92A9A286E4F1AAF9B4'`

Sample Request:

```
https://localhost:8443/restmon/api/v1/schema/testschema/change_events/318A22B95C7A4C92A9A286E4F1AAF9B4
```

Body:

```
{
    "xml_ns": "",
    "url": "changestable",
    "group": "ChangeEvents",
    "attributes": {
        "oi": {
            "timestamp": "#(function() { var result=[];var tz='PDT'; for(var i in root.result) { var temp=new
 Date(root.result[i].work_end.display_value + ' ' + tz); result.push(temp.toJSON()) } return result; })();",
            "startTime": "#(function() { var result=[];var tz='PDT'; for(var i in root.result) { var temp=new
 Date(root.result[i].work_start.display_value + ' ' + tz); result.push(temp.toJSON()) } return result; })();",
            "host": "$['result'][*]['cmdb_ci']['display_value']",
            "ip": "",
            "status": "$['result'][*]['number']['display_value']",
            "product": "ServiceNow",
            "product_version": "1.0.0",
            "group": [],
            "group_id": "",
            "summary": "$['result'][*]['number']['display_value']",
            "previous_value": "",
            "current_value": "",
            "severity": "$['result'][*]['priority']['value']",
            "severity_conversion": "1:Critical,2:Major,3:Minor,Default:Informational",
            "metric_name": "",
            "metric_type": "",
            "message": "$['result'][*]['number']['display_value']%//%: %//%$['result'][*]['short_description']
['display_value']",
            "change_event_unique_id": "['result'][*]['number']['display_value']",
            "ci_unique_id": "$['result'][*]['cmdb_ci']['display_value']",
```

```
            "configuration_item_type": "ServiceNow.Host",
            "configuration_item": "$['result'][*]['cmdb_ci']['display_value']",
            "tags": [
                "ServiceNow",
                "Events",
                "Changes"
            ]
        }
    },
    "fieldId": "318A22B95C7A4C92A9A286E4F1AAF9B4"
}
```

Sample Response:

```
{ "Status": "change_events is updated with fieldId 318A22B95C7A4C92A9A286E4F1AAF9B4" }
```

Schema APIs

none

```
{
    "swagger": "2.0",
    "info": {},
    "host": "localhost:8080",
    "basePath": "/restmon/api",
    "tags": [{
        "name": "schema",
        "description": "These set of APIs enable you to perform CRUD operations for
 RESTmon Schema."
    }],
    "paths": {
        "/v1/schema/getAllSchemaName": {
            "get": {
                "tags": [
                    "schema"
                ],
                "summary": "Lists all Schemas available in the schema directory. ",
                "operationId": "getAllSchemaUsingGET",
                "consumes": [
                    "application/json"
                ],
                "produces": [
                    "application/json"
                ],
                "responses": {
                    "200": {
                        "description": "OK",
                        "schema": {
                            "type": "object"
```

```
                }
            },
            "401": {
                "description": "Unauthorized"
            },
            "403": {
                "description": "Forbidden"
            },
            "404": {
                "description": "Not Found"
            }
        }
    }
},
"/v1/schema/getSchemaDetails/{schemaName}": {
    "get": {
        "tags": [
            "schema"
        ],
        "summary": "Lists the details for the specified Schema.",
        "operationId": "getSchemaDetailsUsingGET",
        "consumes": [
            "application/json"
        ],
        "produces": [
            "application/json"
        ],
        "parameters": [{
            "name": "schemaName",
            "in": "path",
            "description": "Specify the Schema Name.",
            "required": true,
            "type": "string"
        }],
        "responses": {
            "200": {
                "description": "OK",
                "schema": {
                    "type": "object"
                }
            },
            "401": {
                "description": "Unauthorized"
            },
            "403": {
                "description": "Forbidden"
```

```
            },
            "404": {
                "description": "Not Found"
            }
        }
    }
},
"/v1/schema/{schemaName}": {
    "post": {
        "tags": [
            "schema"
        ],
        "summary": "Uploads a schema.",
        "operationId": "uploadSchemaUsingPOST",
        "consumes": [
            "application/json"
        ],
        "produces": [
            "application/json"
        ],
        "parameters": [{
                "in": "body",
                "name": "inp",
                "description": "inp",
                "required": true,
                "schema": {
                    "$ref": "#/definitions/ObjectNode"
                }
            },
            {
                "name": "schemaName",
                "in": "path",
                "description": "schemaName",
                "required": true,
                "type": "string"
            }
        ],
        "responses": {
            "200": {
                "description": "OK",
                "schema": {
                    "type": "object"
                }
            },
            "201": {
                "description": "Created"
```

```
            },
            "401": {
                "description": "Unauthorized"
            },
            "403": {
                "description": "Forbidden"
            },
            "404": {
                "description": "Not Found"
            }
        }
    },
    "delete": {
        "tags": [
            "schema"
        ],
        "summary": "Deletes a schema.",
        "operationId": "deleteSchemaUsingDELETE",
        "consumes": [
            "application/json"
        ],
        "produces": [
            "application/json"
        ],
        "parameters": [{
            "name": "schemaName",
            "in": "path",
            "description": "Specify the Schema Name.",
            "required": true,
            "type": "string"
        }],
        "responses": {
            "200": {
                "description": "OK",
                "schema": {
                    "type": "object"
                }
            },
            "204": {
                "description": "No Content"
            },
            "401": {
                "description": "Unauthorized"
            },
            "403": {
                "description": "Forbidden"
```

```
                    }
                }
            }
        },
        "/v1/schema/{schemaName}/{fieldName}": {
            "get": {
                "tags": [
                    "schema"
                ],
                "summary": "Lists the schema details (URLs, inventory, topology,
metrics, alarms, calculated methods, or calculated metrics) by the specified field
name.",
                "operationId": "getDetailsByFieldNameUsingGET",
                "consumes": [
                    "application/json"
                ],
                "produces": [
                    "application/json;charset=UTF-8"
                ],
                "parameters": [{
                        "name": "schemaName",
                        "in": "path",
                        "description": "schemaName",
                        "required": true,
                        "type": "string"
                    },
                    {
                        "name": "fieldName",
                        "in": "path",
                        "description": "fieldName",
                        "required": true,
                        "type": "string",
                        "enum": [
                            "urls",
                            "metrics",
                            "alarms",
                            "inventory",
                            "topology",
                            "groups",
                            "calculated_metrics",
                            "calculated_methods"
                        ]
                    }
                ],
                "responses": {
                    "200": {
```

```
                        "description": "OK",
                        "schema": {
                            "type": "object"
                        }
                    },
                    "401": {
                        "description": "Unauthorized"
                    },
                    "403": {
                        "description": "Forbidden"
                    },
                    "404": {
                        "description": "Not Found"
                    }
                }
            },
            "post": {
                "tags": [
                    "schema"
                ],
                "summary": "Adds the specified field (URLs, inventory, topology,
metrics, alarms, calculated methods, or calculated metrics) information to the
specified schema.",
                "operationId": "addDetailsUsingPOST",
                "consumes": [
                    "application/json;charset=UTF-8"
                ],
                "produces": [
                    "application/json;charset=UTF-8"
                ],
                "parameters": [{
                        "in": "body",
                        "name": "inpField",
                        "description": "inpField",
                        "required": true,
                        "schema": {
                            "$ref": "#/definitions/ObjectNode"
                        }
                    },
                    {
                        "name": "schemaName",
                        "in": "path",
                        "description": "schemaName",
                        "required": true,
                        "type": "string"
                    },
```

```
                {
                    "name": "fieldName",
                    "in": "path",
                    "description": "fieldName",
                    "required": true,
                    "type": "string",
                    "enum": [
                        "urls",
                        "metrics",
                        "alarms",
                        "inventory",
                        "topology",
                        "groups",
                        "calculated_metrics",
                        "calculated_methods"
                    ]
                }
            ],
            "responses": {
                "200": {
                    "description": "OK",
                    "schema": {
                        "type": "object"
                    }
                },
                "201": {
                    "description": "Created"
                },
                "401": {
                    "description": "Unauthorized"
                },
                "403": {
                    "description": "Forbidden"
                },
                "404": {
                    "description": "Not Found"
                }
            }
        }
    },
    "/v1/schema/{schemaName}/{fieldName}/{fieldId}": {
        "get": {
            "tags": [
                "schema"
            ],
```

```
                "summary": "Lists the schema details (URLs, inventory, topology,
metrics, alarms, calculated methods, or calculated metrics) by the specified field
ID.",
                "operationId": "getDetailsByfieldIdUsingGET",
                "consumes": [
                    "application/json"
                ],
                "produces": [
                    "application/json;charset=UTF-8"
                ],
                "parameters": [{
                        "name": "schemaName",
                        "in": "path",
                        "description": "schemaName",
                        "required": true,
                        "type": "string"
                    },
                    {
                        "name": "fieldName",
                        "in": "path",
                        "description": "fieldName",
                        "required": true,
                        "type": "string",
                        "enum": [
                            "urls",
                            "metrics",
                            "alarms",
                            "inventory",
                            "topology",
                            "groups",
                            "calculated_metrics",
                            "calculated_methods"
                        ]
                    },
                    {
                        "name": "fieldId",
                        "in": "path",
                        "description": "fieldId",
                        "required": true,
                        "type": "string"
                    }
                ],
                "responses": {
                    "200": {
                        "description": "OK",
                        "schema": {
```

```
                        "type": "object"
                    }
                },
                "401": {
                    "description": "Unauthorized"
                },
                "403": {
                    "description": "Forbidden"
                },
                "404": {
                    "description": "Not Found"
                }
            }
        },
        "put": {
            "tags": [
                "schema"
            ],
            "summary": "Updates the URLs, inventory, topology, metrics, alarms,
calculated methods, or calculated metrics by the specified field ID.",
            "operationId": "updateDetailsByIdUsingPUT",
            "consumes": [
                "application/json;charset=UTF-8"
            ],
            "produces": [
                "application/json;charset=UTF-8"
            ],
            "parameters": [{
                    "in": "body",
                    "name": "inp",
                    "description": "inp",
                    "required": true,
                    "schema": {
                        "$ref": "#/definitions/ObjectNode"
                    }
                },
                {
                    "name": "schemaName",
                    "in": "path",
                    "description": "schemaName",
                    "required": true,
                    "type": "string"
                },
                {
                    "name": "fieldName",
                    "in": "path",
```

```
                "description": "fieldName",
                "required": true,
                "type": "string",
                "enum": [
                    "urls",
                    "metrics",
                    "alarms",
                    "inventory",
                    "topology",
                    "groups",
                    "calculated_metrics",
                    "calculated_methods"
                ]
            },
            {
                "name": "fieldId",
                "in": "path",
                "description": "fieldId",
                "required": true,
                "type": "string"
            }
        ],
        "responses": {
            "200": {
                "description": "OK",
                "schema": {
                    "type": "object"
                }
            },
            "201": {
                "description": "Created"
            },
            "401": {
                "description": "Unauthorized"
            },
            "403": {
                "description": "Forbidden"
            },
            "404": {
                "description": "Not Found"
            }
        }
    },
    "delete": {
        "tags": [
            "schema"
```

```
                ],
                "summary": "Deletes the specified field-name (URLs, inventory, topology,
metrics, alarms, calculated methods, or calculated metrics) by the specified field
ID.",
                "operationId": "deleteDetailsByIdUsingDELETE",
                "consumes": [
                    "application/json"
                ],
                "produces": [
                    "application/json;charset=UTF-8"
                ],
                "parameters": [{
                        "name": "schemaName",
                        "in": "path",
                        "description": "schemaName",
                        "required": true,
                        "type": "string"
                    },
                    {
                        "name": "fieldName",
                        "in": "path",
                        "description": "fieldName",
                        "required": true,
                        "type": "string",
                        "enum": [
                            "urls",
                            "metrics",
                            "alarms",
                            "inventory",
                            "topology",
                            "groups",
                            "calculated_metrics",
                            "calculated_methods"
                        ]
                    },
                    {
                        "name": "fieldId",
                        "in": "path",
                        "description": "fieldId",
                        "required": true,
                        "type": "string"
                    }
                ],
                "responses": {
                    "200": {
                        "description": "OK",
```

```
                        "schema": {
                            "type": "object"
                        }
                    },
                    "204": {
                        "description": "No Content"
                    },
                    "401": {
                        "description": "Unauthorized"
                    },
                    "403": {
                        "description": "Forbidden"
                    }
                }
            }
        }
    },
    "definitions": {
        "JsonNode": {
            "type": "object"
        },
        "ObjectNode": {
            "type": "object"
        }
    }
}
```

Integrations

You can use RESTMon schemas to send and receive data from third-party products.

Integrating RESTMon with a third-party product enables you to retrieve and send data to DX Operational Intelligence (DX OI). RESTMon includes out-of-the-box schemas to enable you to extract data from different third-party tools and ingest that data into DX OI. These schemas are available in the **<restmon>/schema** folder.

The following table lists the schemas that are available out-of-the-box to ingest data into DX OI.

Technology	Metrics	Alarms	Topology	Groups
AppDynamics	Yes	Yes	Yes	No
AppNeta	No	Yes	No	No
Datadog	Yes	Yes	Yes	No
DELL EMC ECS	Yes	Yes	Yes	No
Dynatrace	Yes	Yes	Yes	Yes

Technology	Metrics	Alarms	Topology	Groups
Elasticsearch	Yes	Yes	Yes	No
Elasticsearch Log Data	No	Yes	Yes	No
Google Cloud Monitoring	No	Yes	Yes	No
MongoDB	No	Yes	Yes	No
Nagios	Yes	Yes	Yes	Yes
New Relic	Yes	Yes	Yes	No
PureStorage	Yes	Yes	Yes	No
ScienceLogic	No	Yes	Yes	No
SCOM	Yes	Yes	Yes	No
ServiceNow	No	Yes	No	No
SolarWinds	Yes	Yes	Yes	No
SolarWinds AppOptics	Yes	Yes	Yes	No
Splunk	Yes	Yes	Yes	No
ThousandEyes	Yes	Yes	Yes	Yes
Tivoli Netcool/OMNIbus	No	Yes	No	No
Zabbix	Yes	Yes	Yes	No

AppDynamics

DX Operational Intelligence supports integration with AppDynamics.

The AppDynamics schema is a Polling schema that queries the AppDynamics REST APIs to get the data. This schema helps ingest the following monitoring data from AppDynamics SaaS into DX Operational Intelligence:

- Metrics
- Inventory and Topology
- Alarms
- Groups

This section provides the following information:

Supported Versions

The AppDynamics - DX Operational Intelligence integration is supported for the following version:

Product	Supported Version
AppDynamics	SaaS

Configure the Integration

The AppDynamics - DX Operational Intelligence integration involves the following steps:

- Configure the AppDynamics Environment
- Configure RESTMon

Configure the AppDynamics Environment

Configure the Alert & Respond system in your environment to send the event data to DX Operational Intelligence. Before you configure, ensure that the following requirements are met:

- You have an active AppDynamics account.
- You have the required permissions to configure the Alert & Respond Templates, Actions, and Policies in AppDynamics.
- Your AppDynamics environment can make requests to external endpoints over port 443.

For more information, see the documentation.

Follow these steps:

1. Log in to AppDynamics.
2. Create an **HTTPS Request Template** in the Alert and Respond UI.
3. Create an **Action** for each AppDynamics business application that should report events to DX Operational Intelligence.
4. Create a **Policy** that applies the **Send to AIOps** action to health rules for each AppDynamics business application that should report events to DX Operational Intelligence.

Configure RESTMon

To configure RESTMon, update the OI connection details if not done earlier, and add the profile information to the **restmon.json** file. Before you configure RESTMon, ensure that the following requirements are met:

- You have access to DX Operational Intelligence.
- RESTMon is configured and deployed successfully. For more information, see the Configure and Deploy RESTMon section.
- RESTMon is configured and deployed successfully.
- For Java 11, replace the **contains** method with **includes** in the schema.

Add the Profile

To add the profile, configure the profile to connect to your AppDynamics environment and add the profile to the **restmon.json** file using the APIs for a Profile in Swagger. The **appdynamics_profile.json** file is available in the **<restmon\profile>** folder. When you add the profile, the schema for AppDynamics is automatically uploaded, and the data ingestion starts.

NOTE

- If you are using the Bearer authentication type to connect to your AppDynamics environment, use the **appdynamics_token_profile.json** file.
- To filter the entities, add the attribute filter to the profile.json file. For more information, see the **attribute_filter** section on this page.

You can also add this information directly in the **restmon.json** file.

The AppDynamics profile includes the following sections:

Profile

The **profile** section defines the profile-related information, and the following snippet is a sample of the profile section:

```
"{
  "profile": {
    "name": "appdynamics",
    "active": "yes",
    "schema": "appdynamics",
    "polling_interval_secs": "300",
```

```
"topology_ttl_mins": "2880"
},
```

Name	Description	Type	Example
name	Indicates the name of the profile.	String	appdynamics
active	Indicates if the data-processing is enabled. Enter *yes* to enable the profile.	Boolean	yes
schema	Indicates the schema name. The name that you specify for the schema should be the same as the schema attribute specified in the **restmon.json** file.	String	appdynamics
polling_interval_secs	Indicates the polling interval in seconds. **Supported Values:** 60, 300, 900, 1800, 3600, and 7200 **Supported Values for Metric Ingestion:** 15, 30, 60, 300, 900, 1800, 3600, and 7200 We recommend that you set the polling interval for a minimum of 60s or more to avoid performance issues. RESTMon also supports Dynamic Polling Intervals. Using Dynamic Polling Interval, you can schedule the data parsing intervals for each data category. For more information, see the Polling Interval for Schema Parsing section.	Integer	60
topology_ttl_min	Indicates the time-to-live (TTL) a record is cached in minutes.	Integer	2880
tenantname	Indicates the DX Operational Intelligence tenant name.	String	A00B111C-22DB-33C3-444C-000F0000B000

servicedefinition

In the **servicedefinition** section, define the service that appears when data is ingested to DX Operational Intelligence. The following snippet is a sample of the **servicedefinition** section:

```
"servicedefinition":{   "name":"appdynamics",
   "status":"Active"
},
```

Name	Description	Type	Example
name	Indicates the name of the service.	String	appdynamics
status	Indicates the status of the defined service.	String	Active

restapiconnectdetails

In the **restapiconnectdetails** section, enter the REST Endpoint details of your AppDynamics environment, and the following snippet is a sample of the **restapiconnectdetails** section:

```
{
    "type": "https",
    "hostname": "sampleHostname",
    "port": "",
    "authentication": "basic",
    "username": "sampleUser",
    "password": "samplePassword",
    "realmdomain": "",
    "token": "",
    "httptimeout": "30000",
    "checkcert": "no"
  }
```

Name	Description	Type	Example
type	Indicates the data transfer type with DX Operational Intelligence. Values: HTTP or HTTPS.	String	https
hostname	Indicates the hostname or IP address of the REST Endpoint.	String	sampleHostname
port	Indicates the port number of the REST Endpoint.	Integer	17778
authentication	Indicates the authentication type. For the AppDynamics integration, you can set **basic** or **bearer**. Additionally, the following authentication types are available: • **none:** No authorization is required. • **basic:** Enter the username and password. • **NTLM:** Enter the username and password. • **digest:** Enter the username, password, and real domain. • **OAuth2:** Enter the access token in the token parameter. • **bearer:** Enter the bearer token in the token parameter. • **urltoken:** Enter the token in the token parameter.	String	Bearer
username	Indicates the username. Applies only when the authentication type is basic or NTLM.	String	

Name	Description	Type	Example
password	Indicates the password. Applies only when the authentication type is basic or NTLM.	String	
realmdomain	Indicates the DNS realm or domain to encode in the token. Applies only when the authentication type is digest.	String	
token	Indicates the access token or bearer token when the authentication type is OAuth2 or bearer, respectively.	String	sampleToken
httptimeout	Indicates the value of the timeout that is expressed in milliseconds.	Integer	30000
checkcert	Indicates to verify if the certificate is valid and trusted.	Boolean	no

monitored_groups

In the **monitored_groups** section, specify the groups that you want to monitor, and the following snippet is a sample of the **monitored_groups** section:

```
 "monitored_groups" : {
         "JVM" : "yes",
         "JMX":"yes",
         "Agent" : "yes",
         "Hardware Resources" : "yes",
         "Applications" : "yes",
         "Applications_Inventory" : "no",
         "Databases_Inventory" : "no",
         "Databases" : "yes",
         "Application Infrastructure Performance" : "yes",
         "Overall Application Performance" : "yes",
         "CPU" : "yes",
         "tier-tier-backend" : "yes",
         "Business Transactions" : "yes"
   }
```

attribute_filter

You can configure RESTMon to ingest and display only the required information on the DX Operational Intelligence UI using the attribute filter. To filter the ingested data, add the attribute filter to the profile.json file. For more information, see the Filter Entities and Related Data Before Ingestion section.

> **NOTE**
> For the detailed steps, see the Add the Profile section.

Upload the AppDynamics Schema

The schema is automatically uploaded when you add the profile. Perform the steps only if you want to upload the updated or edited schema.

> **NOTE**
> For the detailed steps, see the Upload the Schema section.

View Data in DX OI

You can view the ingested data in the Alarm Analytics, Service Analytics, and Performance Analytics pages in DX OI for insights into service, raw, and anomaly alarms.

> **NOTE**
> For more information, see the View Data in DX OI section.

AppNeta

DX Operational Intelligence supports integration with AppNeta.

AppNeta monitors the network paths between locations. The AppNeta schema is a Streaming schema that enables you to send events to RESTMon. RESTMon converts the events to alarms and ingests these alarms into DX OI.

This section provides the following information:

Supported Versions

The AppNeta - DX Operational Intelligence integration is supported for the following version:

Product	Supported Version
AppNeta	SaaS

Configure the Integration

The AppNeta - DX Operational Intelligence involves the following steps:

- Configure the AppNeta Environment
- Configure RESTMon

Configure the AppNeta Environment

Consider the following points for this integration:

- You can use either HTTPS or HTTP to communicate between APM and your server.

  > **NOTE**
  > AppNeta requires an SSL Certificate from a Certificate Authority (CA). Self-signed certificates are not supported.

- Set up RESTMon as an Observer Webhook in AppNeta. Create JSON and set sqaEvents to true for the AppNeta to forward SQA Events to RESTMon as shown in the sample:

```
[
   {
      "url":"http://admin:password@<hostname>/restmon/api/v1/logs?profileName=appneta&schemaName=appneta",
      "testEvents":false,
      "seqEvents":false,
      "sqaEvents":true,
      "webAlertEvents":false,
      "networkChangeEvents":false,
      "blacklisted":false
   }
]
```

> **NOTE**
> For more information, see the AppNeta documentation.

Configure RESTMon

To configure RESTMon, update the OI connection details if not done earlier, and add the profile information to the **restmon.json** file. Before you configure RESTMon, ensure that the following requirements are met:

- You have access to DX Operational Intelligence.
- RESTMon is configured and deployed successfully. For more information, see the Configure and Deploy RESTMon section.
- RESTMon is configured and deployed successfully.

Add the Profile

To add the profile, configure the profile to connect to your AppNeta environment and add the profile to the **restmon.json** file using the POST Profile REST API call in Swagger. The **appneta_profile.json** file is available in the **<restmon \profile>** folder. When you add the profile, the schema for AppNeta is automatically uploaded, and the data ingestion starts.

> **NOTE**
> You can also add the attribute filter to the profile to filter the entities and the related data. For more information, see the **attribute_filter** section on this page.

You can also add this information directly in the **restmon.json** file.

The AppNeta profile includes the following sections:

Profile

The **profile** section defines the profile-related information, and the following snippet is a sample of the profile section:

```
"profile":{
      "name":"appneta",
      "active":"yes",
      "schema":"appneta",
      "streaming":"yes",
      "polling_interval_secs":1,
      "batch_size":1000,
      "batch_wait_time_milli":2000,
      "topology_ttl_mins":"2880",
      "tenantname":"sampleTenantName"
   },
```

Name	Description	Type	Example
name	Indicates the name of the profile.	String	appneta
active	Indicates if the data-processing is enabled. Enter *yes* to enable the profile.	Boolean	yes
schema	Indicates the schema name. The name that you specify for the schema should be the same as the schema attribute specified in the restmon.json.	String	appneta
streaming	Indicates the schema type.	String	yes

Name	Description	Type	Example
polling_interval_secs	Defines the polling interval in seconds. **Supported Values for Metric Ingestion:** 15, 30, 60, 300, 900, 1800, 3600, and 7200 We recommend that you set the polling interval for a minimum of 60s or more to avoid performance issues. RESTMon also supports Dynamic Polling Intervals. Using Dynamic Polling Interval, you can schedule the data parsing intervals for each data category. For more information, see the Polling Interval for Schema Parsing section.	Integer	1
batch_size	Indicates the size of batches with which processing happens. Applicable for only the PUSH schemas.	Integer	1000
batch_wait_time_milli	Indicates the wait time that is given to fill up the batch size. If the wait time is exceeded, then the existing batch is executed. Applicable for only the PUSH schemas.	Integer	2000
topology_ttl_min	Indicates the time-to-live (TTL) a record is cached in minutes.	Integer	2880
tenantname	Indicates the DX Operational Intelligence tenant name.	String	A00B111C-22DB-33C3-444C-000F0000B000

servicedefinition

In the **servicedefinition** section, define the service that appears when data is ingested to DX Operational Intelligence. The following snippet is a sample of the **servicedefinition** section:

```
"servicedefinition":{   "name":"",
   "status":""
},
```

Name	Description	Type	Example
name	Indicates the name of the service.	String	appneta
status	Indicates the status of the defined service.	String	Active

restapiconnectdetails

In the **restapiconnectdetails** section, enter the REST Endpoint details of your AppNeta environment and the following snippet is a sample of the **restapiconnectdetails** section:

```
{
```

```
        "type":"http",
        "hostname":"notneeded",
        "port":9600,
        "authentication":"",
        "username":"",
        "password":"",
        "realmdomain":"",
        "token":"",
        "httptimeout":300,
        "checkcert":"no"
    },
```

Name	Description	Type	Example
type	Indicates the data transfer type with DX Operational Intelligence. Values: HTTP or HTTPS.	String	https
hostname	Indicates the hostname or IP address of the REST Endpoint. The hostname is not required.	String	
port	Indicates the port number of the REST Endpoint.	Integer	9600
authentication	Indicates the authentication type. The following authentication types are available: • **none:** No authorization is required. • **basic:** Enter the username and password. • **NTLM:** Enter the username and password. • **digest:** Enter the username, password, and the realmdomain. • **OAuth2:** Enter the access token in the token parameter. • **bearer:** Enter the bearer token in the token parameter. • **urltoken:** Enter the token in the token parameter.	String	none
username	Indicates the username. Applies only when the authentication type is basic or NTLM.	String	
password	Indicates the password. Applies only when the authentication type is basic or NTLM.	String	
realmdomain	Indicates the DNS realm or domain to encode in the token. Applies only when the authentication type is digest.	String	

Name	Description	Type	Example
token	Indicates the access token or bearer token when the authentication type is OAuth2 or bearer, respectively.	String	sampleToken
httptimeout	Indicates the value of the timeout that is expressed in milliseconds.	Integer	120000
checkcert	Indicates to verify the certificate to ensure it is valid and trusted.	Boolean	no

monitored_groups

In the **monitored_groups** section, specify the groups that you want to monitor and the following snippet is a sample of the **monitored_groups** section:

```
"monitored_groups": {
    "Topology": "yes",
    "Alerts": "yes",
    }
```

attribute_filter

You can configure RESTMon to ingest and display only the required information on the DX Operational Intelligence UI using the attribute filter. To filter the ingested data, add the attribute filter to the profile.json file. For more information, see the Filter Entities and Related Data Before Ingestion section.

> **NOTE**
> For the detailed steps, see the Add the Profile section.

Upload the AppNeta Schema

The schema is automatically uploaded when you add the profile. Perform the steps only if you want to upload the updated or edited schema.

> **NOTE**
> For the detailed steps, see the Upload the Schema section.

View Data in DX OI

You can view the ingested data in the Alarm Analytics, Service Analytics, and Performance Analytics pages in DX OI for insights into service, raw, and anomaly alarms.

> **NOTE**
> For more information, see the View Data in DX OI section.

Datadog

DX Operational Intelligence supports integration with Datadog.

The Datadog schema is a Polling schema that queries the Datadog REST APIs to get the data. This schema enables you to ingest the following monitoring data from the Datadog SaaS into DX Operational Intelligence:

- **Inventory**
- **Metrics** (Memory, Logical Disk, Health Service, Network Adapter, Health Management Group, System, and Processor Information)
- **Alarms**
- **Topology**

This section provides the following information:

Supported Versions

The Datadog - DX Operational Intelligence integration is supported for the following version:

Product	Supported Version
Datadog	SaaS

Configure the Integration

To Datadog - DX Operational Intelligence integration involves the following steps:

- Configure the Datadog Environment
- Configure RESTMon

Configure the Datadog Environment

No integration-specific steps are required to be performed in your Datadog environment. However, ensure that the following requirements are met:

- The Datadog server port is open and is accessible.
- The Datadog system can accept the HTTPS requests.
- The Datadog server URL, Datadog API key, and Application key are available.

> **NOTE**
> For more information, see the Datadog documentation.

Configure RESTMon

To configure RESTMon, update the OI connection details if not done earlier and add the profile information to the **restmon.json** file. Before you configure RESTMon, ensure that the following requirements are met:

- You have access to DX Operational Intelligence.
- RESTMon is configured and deployed successfully. For more information, see the Configure and Deploy RESTMon section.
- RESTMon is configured and deployed successfully.
- For Java 11, replace the **contains** method with **includes** in the schema.

Add the Profile

To add the profile, configure the profile to connect to your third-party environment and add the profile to the **restmon.json** file using the POST Profile REST API call in Swagger. The **datadog_profile.json** file is available in the **<restmon \profile>** folder. When you add the profile, the schema for Datadog is automatically uploaded and the data ingestion starts.

> **NOTE**
> You can also add the attribute filter to the profile to filter the entities and the related data. For more information, see the **attribute_filter** section on this page.

You can also add the profile information directly in the **restmon.json** file.

The Datadog profile includes the following sections:

Profile

In the **profile** section, you define the profile-related information. The following snippet is a sample of the profile section.

```
"{
  "profile": {
    "name": "datadog",
    "active": "yes",
    "schema": "datadog",
    "polling_interval_secs": "480",
    "topology_ttl_mins": "2880"
  },
```

Name	Description	Type	Example
name	Indicates the name of the profile.	string	datadog
active	Indicates if the data-processing is enabled. Enter *yes* to enable the profile.	Boolean	yes
schema	Indicates the schema name. The name that you specify should be the same as the schema attribute specified in the restmon.json file.	string	datadog
polling_interval_secs	Defines the polling interval in seconds. **Supported Values:** 60, 300, 900, 1800, 3600, and 7200 **Supported Values for Metric Ingestion:** 15, 30, 60, 300, 900, 1800, 3600, and 7200 We recommend that you set the polling interval for a minimum of 60s or more to avoid performance issues. RESTMon also supports Dynamic Polling Intervals. Using Dynamic Polling Interval, you can schedule the data parsing intervals for each data category. For more information, see the Polling Interval for Schema Parsing section.	Integer	60
topology_ttl_min	Indicates the time-to-live (TTL) a record is cached in minutes.	Integer	2880
tenantname	Indicates the DX OI tenant name.	String	A00B111C-22DB-33C3-444C-000F0000B000

servicedefinition

In the **servicedefinition** section, define the service that appears when data is ingested to DX OI. The following snippet is a sample of the **servicedefinition** section:

```
"servicedefinition":{   "name":"Datadog",
   "status":"Active"
```

```
},
```

name	Description	Type	Example
name	Indicates the name of the service.	String	datadog
status	Indicates the status of the defined service.	String	active

restapiconnectdetails

In the **restapiconnectdetails** section, Indicates the REST Endpoint details of your Datadog environment. The following snippet is a sample of the **restapiconnectdetails** section:

```
"restapiconnectdetails": {
    "type": "https",
    "hostname": "sampleHostname",
    "port": "",
    "authentication": "basic",
    "username": "sampleUser",
    "password": "samplePassword",
    "realmdomain": "",
    "token": "",
    "httptimeout": "30000",
    "checkcert": "no"
  },
```

Name	Description	Type	Example
type	Indicates the data transfer type with DX OI. **Values:** http or https.	String	https
hostname	Indicates the hostname or the IP address of the REST Endpoint.	String	sampleHostname
port	Indicates the port number of the REST Endpoint.	Integer	9600
authentication	Indicates the authentication type. Indicates the authentication type as **urltoken**.	String	urltoken
username	Indicates the username. Applies only when the authentication type is **basic** or **NTLM**.	String	
password	Indicates the password. Applies only when the authentication type is **basic** or **NTLM**.	String	
realmdomain	Indicates the DNS realm or domain to encode in the token. Applies only when the authentication type is **digest.**	String	

Name	Description	Type	Example
token	Indicates whether it is the access token or bearer token when the authentication type is **OAuth2** or **bearer** respectively.	String	sampleToken
httptimeout	Indicates the value of the timeout expressed in milliseconds.	Integer	120000
checkcert	Indicates to verify if the certificate is valid and trusted.	Boolean	no

monitored_groups

In the **monitored_groups** section, specify the groups that you want to monitor. The following snippet is a sample of the **monitored_groups** section:

```
"monitored_groups" : {
          "Hosts" : "yes",
          "Hosts_Inventory" : "no",
        "Metrics" : "yes",
        "Alarms" : "yes"
      }
```

attribute_filter

You can configure RESTMon to ingest and display only the required information on the DX Operational Intelligence UI using the attribute filter. To filter the ingested data, add the attribute filter to the profile.json file. For more information, see the Filter Entities and Related Data Before Ingestion section.

> **NOTE**
> For the detailed steps, see the Add the Profile section.

Upload the Datadog Schema

The schema is automatically uploaded when you add the profile. Perform the steps only if you want to upload the updated or edited schema.

> **NOTE**
> For the detailed steps, see the Upload the Schema section.

View Data in DX OI

You can view the ingested data in the Alarm Analytics, Service Analytics, and Performance Analytics pages in DX OI for insights into service, raw, and anomaly alarms.

> **NOTE**
> For more information, see the View Data in DX OI section.

Dell EMC ECS

DX Operational Intelligence supports integration with Dell EMC Elastic Cloud Storage.

Dell EMC Elastic Cloud Storage (ECS) is an object storage platform that provides monitoring capabilities at both the cluster level (storage capacity) and the node level (node disks). The ECS schema is a Polling schema that connects to the REST API, and retrieves the cluster and node topology. The schema also collects metrics to help monitor the cluster level metrics for the current managed capacity through the Capacity API. The Dashboard API retrieves the local Virtual Data Center (VDC) details related to the cluster and nodes.

- The Capacity API (/object/capacity) retrieves the current managed capacity for the cluster.
  Metrics include: Total Free, Total Provisioned, Total Used
- The Dashboard API (/dashboard/zones/localzone) retrieves the local VDC details at the cluster and node levels.
  - **Cluster Level Metrics:** Avg CPU Percent, Avg Memory Usage, Relative Nic Percentage, Relative Memory Percentage, Read Latency, Write Latency, Read Bandwidth, Write Bandwidth, Read Transactions Per Second, and Write Transactions Per Second
  - **Node Level Metrics:** CPU Usage Percentage, Memory Usage, Relative NIC Percentage, Relative Memory Percentage, Read Latency, Write Latency, Read Bandwidth, Write Bandwidth, Read Transactions, and Write Transactions
- The Alerts API (/vdc/alerts) retrieves the audit alerts.

The ECS schema polls for the following data:

- Alarms
- Metrics
- Topology

This section provides the following information:

Supported Versions

The ECS - DX Operational Intelligence integration is supported for the following version:

Product	Supported Version
ECS	3.3

Configure the Integration

The ECS - DX Operational Intelligence integration involves the following steps:

- Configure the ECS Environment
- Configure RESTMon

Configure the ECS Environment

No integration-specific steps are required to be performed. However, authenticate with the ECS Management REST API to generate the authentication token. For more information, see the Authenticate with the ECS Management REST API documentation.

Configure RESTMon

To configure RESTMon, update the OI connection details if not done earlier, and add the profile information to the **restmon.json** file. Before you configure RESTMon, ensure that the following requirements are met:

- You have access to DX Operational Intelligence.
- RESTMon is configured and deployed successfully. For more information, see the Configure and Deploy RESTMon section.
- RESTMon is configured and deployed successfully.

Add the Profile

To add the profile, configure the profile to connect to your EMC ECS environment and add the profile to the **restmon.json** file using the POST Profile REST API call in Swagger. The **ecs_profile.json** file is available in the **<restmon\profile>** folder. When you add the profile, the schema for EMC ECS is automatically uploaded, and the data ingestion starts.

NOTE
You can also add the attribute filter to the profile to filter the entities and the related data. For more information, see the **attribute_filter** section on this page.

You can also add this information directly in the **restmon.json** file.

The ECS profile includes the following sections:

Profile

The **profile** section defines the profile-related information, and the following snippet is a sample of the profile section:

```
{
      "profile" : {
        "name" : "ecs",
        "active" : "yes",
        "schema" : "ecs",
        "polling_interval_secs" : "180",
        "topology_ttl_mins" : "10"
      },
```

Name	Description	Type	Example
name	Indicates the name of the profile.	String	ecs
active	Indicates if the data-processing is enabled. Enter *yes* to enable the profile.	Boolean	yes
schema	Indicates the schema name. The name that you specify for the schema should be the same as the schema attribute specified in the restmon.json.	String	ecs
polling_interval_sec	Indicates the polling interval in seconds. **Supported Values:** 60, 300, 900, 1800, 3600, and 7200 **Supported Values for Metric Ingestion:** 15, 30, 60, 300, 900, 1800, 3600, and 7200 We recommend that you set the polling interval for a minimum of 60s or more to avoid performance issues. RESTMon also supports Dynamic Polling Intervals. Using Dynamic Polling Interval, you can schedule the data parsing intervals for each data category. For more information, see the Polling Interval for Schema Parsing section.	integer	60
topology_ttl_min	Indicates the time-to-live (TTL) a record is cached in minutes.	Integer	10

restapiconnectdetails

In the **restapiconnectdetails** section, enter the REST Endpoint details of your EMC ECS environment, and the following snippet is a sample of the **restapiconnectdetails** section:

```
{
    "type": "https",
    "hostname": "<hostname>",
    "port": "443",
    "authentication": "basic",
    "username": "myusername",
    "password": "mypassword",
    "realmdomain": "",
    "token": "",
    "httptimeout": "30000",
    "checkcert": "no",
}
```

Name	Description	Type	Example
type	Indicates the data transfer type with DX OI. Values: HTTP or HTTPS.	String	https
hostname	Indicates the hostname or IP address of the REST Endpoint.	String	sampleHostname
port	Indicates the port number of the REST Endpoint.	Integer	443
authentication	Indicates the authentication type. The following authentication types are available: • **none:** No authorization is required. • **basic:** Enter the username and password. • **NTLM:** Enter the username and password. • **digest:** Enter the username, password, and the realmdomain. • **OAuth2:** Enter the access token in the token parameter. • **bearer:** Enter the bearer token in the token parameter. • **urltoken:** Enter the token in the token parameter.	String	basic
username	Indicates the username. Applies only when the authentication type is basic or NTLM.	String	
password	Indicates the password. Applies only when the authentication type is basic or NTLM.	String	

Name	Description	Type	Example
realmdomain	Indicates the DNS realm or domain to encode in the token. Applies only when the authentication type is digest.	String	
token	Indicates the authentication token.	String	
httptimeout	Indicates the value of the timeout that is expressed in milliseconds.	Integer	30000
checkcert	Indicates verify the certificate to ensure it is valid and trusted.	Boolean	no

monitored_groups

In the **monitored_groups** section, specify the groups that you want to monitor, and the following snippet is a sample of the **monitored_groups** section:

```
"monitored_groups" : {
        "Topology": "yes",
        "Inventory": "yes",
        "Alarms": "yes",
        "CapacityMetrics": "yes",
        "ClusterPerformance" : "yes",
        "NodePerformance" : "yes"
      }
```

attribute_filter

You can configure RESTMon to ingest and display only the required information on the DX Operational Intelligence UI using the attribute filter. To filter the ingested data, add the attribute filter to the profile.json file. For more information, see the Filter Entities and Related Data Before Ingestion section.

> **NOTE**
> For detailed steps, see the Add the Profile section.

Upload the ECS Schema

The schema is automatically uploaded when you add the profile. Perform the steps only if you want to upload the updated or edited schema.

> **NOTE**
> For the detailed steps, see the Upload the Schema section.

View Data in DX OI

You can view the ingested data in the Alarm Analytics, Service Analytics, and Performance Analytics pages in DX OI for insights into service, raw, and anomaly alarms.

> **NOTE**
> For more information, see the View Data in DX OI section.

Dynatrace

DX Operational Intelligence supports integration with Dynatrace.

The Dynatrace schema is a Polling schema that queries the Dynatrace REST APIs to get the data. This schema enables you to retrieve events from Dynatrace and ingest them into DX Operational Intelligence.

> **NOTE**
> For alarms with **eventType="MEMORY_SATURATED**", the affected metric is not working. The alarm is ingested into DX OI, but the affected metric tab is not displayed.

This section provides the following information:

Supported Versions

The Dynatrace - DX OI integration is supported for the following version:

Product	Supported Version
Dynatrace	SaaS

Configure the Integration

To Dynatrace - DX OI integration involves the following steps:

- Configure the Dynatrace Environment
- Configure RESTMon

Configure the Dynatrace Environment

For the integration, create a Webhook in Dynatrace to post data to DX OI when an event occurs. Before you configure the Webhook, ensure that the following requirements are met:

- You have an active Dynatrace account.
- You have the necessary permissions to create a Webhook integration in Dynatrace.
- Your Dynatrace environment can make requests to external endpoints over port 443.

Follow these steps:

1. Create a Webhook in Dynatrace. For more information, see the Webhook Integration documentation.
2. Enter your Dynatrace Notification integration URL.
3. Enable **Custom Payload** and apply the following template:

```
{
"State":"{State}"
"ProblemID":"{ProblemID}"
"ProblemTitle":"{ProblemTitle}"
}
```

> **NOTE**
> For more information, see the Dynatrace documentation.

Configure RESTMon

To configure RESTMon, update the OI connection details if not done earlier, and add the profile information to the **restmon.json** file. Before you configure RESTMon, ensure that the following requirements are met:

- You have access to DX OI.
- RESTMon is configured and deployed successfully. For more information, see the Configure and Deploy RESTMon section.
- RESTMon is configured and deployed successfully.
- For Java 11, replace the **contains** method with **includes** in the schema.

Add the Profile

To add the profile, configure the profile to connect to your Dynatrace environment and add the profile to the **restmon.json** file using the POST Profile REST API call in Swagger. The **dynatrace_profile.json** file is available in the **<restmon \profile>** folder. When you add the profile, the schema for Dynatrace is automatically uploaded, and the data ingestion starts.

> **NOTE**
> You can also add the attribute filter to the profile to filter the entities and the related data. For more information, see the **attribute_filter** section on this page.

You can also add this information directly in the **restmon.json** file.

The Dynatrace profile includes the following sections:

Profile
The **profile** section defines the profile-related information, and the following snippet is a sample of the profile section:

```
"profile":{
   "name":"dynatrace",
   "active":"yes",
   "schema":"dynatrace",
   "polling_interval_secs":"10",
   "inventory_topology_fullsync_interval_mins":"1440",
   "topology_ttl_mins":"2880",
   "tenantname":"B1A6A2FE-F4AE-05F2-1573-53491B8706D0"
},
```

Name	Description	Type	Example
name	Indicates the name of the profile.	String	dynatrace
active	Indicates if the data-processing is enabled. Enter *yes* to enable the profile.	Boolean	yes
schema	Indicates the schema name. The name that you provide for the schema should be the same as the schema attribute specified in the restmon.json file.	String	dynatrace

Name	Description	Type	Example
polling_interval_secs	Indicates the polling interval in seconds. **Supported Values:** 60, 300, 900, 1800, 3600, and 7200 **Supported Values for Metric Ingestion:** 15, 30, 60, 300, 900, 1800, 3600, and 7200 We recommend that you set the polling interval for a minimum of 60s or more to avoid performance issues. RESTMon also supports Dynamic Polling Intervals. Using Dynamic Polling Interval, you can schedule the data parsing intervals for each data category. For more information, see the Polling Interval for Schema Parsing section.	Integer	10
topology_ttl_min	Indicates the time-to-live (TTL) a record is cached in minutes.	Integer	2880
tenantname	Indicates the DX Operational Intelligence tenant name.	String	B1A6A2FE-F4AE-05F2-1573-53491B8706D0

servicedefinition

In the **servicedefinition** section, define the service that appears when data is ingested to DX OI. The following snippet is a sample of the **servicedefinition** section:

```
"servicedefinition":{   "name":"dynatrace",
   "status":"Active"
},
```

Name	Description	Type	Example
name	Indicates the name of the service.	String	dynatrace
status	Indicates the status of the defined service.	String	Active

restapiconnectdetails

In the **restapiconnectdetails**, specify the REST Endpoint details of your Dynatrace environment. The following snippet is a sample of the **restapiconnectdetails** section:

```
"restapiconnectdetails":{
   "type":"https",
   "hostname":"http://exampledynatraceapm1",
   "port":"8021",
   "authentication":"urltoken",
   "username":"",
   "password":"",
   "realmdomain":"",
   "token":"apiTokenValue",
   "httptimeout":"30000",
   "checkcert":"no"
```

```
},
```

Name	Description	Type	Example
type	Indicates the data transfer type with OI. Value: HTTP or HTTPS.	String	https
hostname	Indicates the hostname or the IP address of the REST Endpoint.	String	http://exampledynatraceapm1
port	Indicates the port number of the REST Endpoint.	Integer	8021
authentication	Indicates the authentication type. For the Dynatrace integration, set **urltoken**. Additionally, the following authentication types are available: • **none:** No authorization is required. • **basic:** Enter the username and password. • **NTLM:** Enter the username and password. • **digest:** Enter the username, password, and realmdomain. • **OAuth2:** Enter the access token in the token parameter. • **bearer:** Enter the bearer token in the token parameter. • **urltoken:** Enter the token in the token parameter.	String	urltoken
username	Indicates the username. Applies only when the authentication type is **basic** or **NTLM**.	String	
password	Indicates the password. Applies only when the authentication type is **basic** or **NTLM**.	String	
realmdomain	Indicates the DNS realm or domain to encode in the token. Applies only when the authentication type is **digest**.	String	
token	Indicates the access token or bearer token when the authentication type is OAuth2 or bearer, respectively.	String	apiTokenValue
httptimeout	Indicates the value of the timeout that is expressed in milliseconds.	Integer	30000
checkcert	Indicates to verify the certificate to ensure it is valid and trusted.	Boolean	no

monitored_groups
In the **monitored_groups** section, specify the groups that you want to monitor. The following snippet is a sample of the **monitored_groups** section:

```
"monitored_groups":{
   "Hosts":"yes",
   "Alarms":"no",
   "Processes":"yes",
   "Services":"yes",
   "Application":"yes",
   "host-group":"no"
}
```

attribute_filter

You can configure RESTMon to ingest and display only the required information on the DX Operational Intelligence UI using the attribute filter. To filter the ingested data, add the attribute filter to the profile.json file. For more information, see the Filter Entities and Related Data Before Ingestion section.

> **NOTE**
> For the detailed steps, see the Add the Profile section.

Upload the Dynatrace Schema

The schema is automatically uploaded when you add the profile. Perform the steps only if you want to upload the updated or edited schema.

> **NOTE**
> For the detailed steps, see the Upload the Schema section.

View Data in DX OI

You can view the ingested data in the Alarm Analytics, Service Analytics, and Performance Analytics pages in DX OI for insights into service, raw, and anomaly alarms.

> **NOTE**
> For more information, see the View Data in DX OI section.

Elasticsearch

DX Operational Intelligence supports integration with Elasticsearch.

The Elasticsearch schema is a Polling schema where RESTMon polls your Elasticsearch servers at regular intervals to collect data. This schema enables you to ingest the following data into DX Operational Intelligence (DX OI):

- Inventory
- Metrics
- Alarms
- Topology

This section provides the following information:

Supported Versions

The Elasticsearch - DX OI integration is supported for the following version:

Product	Supported Version
Elasticsearch	SaaS

Configure the Integration

To Elasticsearch - DX OI integration involves the following steps:

- Configure the Elasticsearch Environment
- Configure RESTMon

Configure the Elasticsearch Environment

No integration-specific steps are required to be performed in your Elasticsearch environment. However, ensure that the port for your Elasticsearch server is open and accessible. For more information about Elasticsearch, see the Elasticsearch documentation.

Configure RESTMon

To configure RESTMon, update the OI connection details if not done earlier and add the profile information to the **restmon.json** file. Before you configure RESTMon, ensure that the following requirements are met:

- You have access to DX OI.
- RESTMon is configured and deployed successfully. For more information, see the Configure and Deploy RESTMon section.
- RESTMon is configured and deployed successfully.

Add the Profile

To add the profile, configure the profile to connect to your Elasticsearch environment and add the profile to the **restmon.json** file using the POST Profile REST API call in Swagger. The **elasticsearch_profile.json** file is available in the **<restmon\profile>** folder. When you add the profile, the schema for Elasticsearch is automatically uploaded and the data ingestion starts.

> **NOTE**
> You can also add the attribute filter to the profile to filter the entities and the related data. For more information, see the **attribute_filter** section on this page.

You can also add this information directly in the **restmon.json** file.

The Elasticsearch profile includes the following sections:

Profile

The **profile** section defines the profile-related information. The following snippet is a sample of the profile section.

```
{
      "profile" : {
        "name" : "elasticsearch",
        "active" : "yes",
        "schema" : "elasticsearch",
        "polling_interval_secs" : "300",
        "inventory_topology_fullsync_interval_mins" : "1440",
        "topology_ttl_mins" : "2880"
```

```
        },
```

Name	Description	Type	Example
name	Indicates the name of the profile.	String	Elasticsearch
active	Indicates if the data-processing is enabled. Enter *yes* to enable the profile.	Boolean	yes
schema	Indicates the schema name. The name that you specify for the schema should be the same as the schema attribute specified in the restmon.json file.	String	Elasticsearch
polling_interval_secs	Indicates the polling interval in seconds. **Supported Values:** 15, 30, 60, 300, 900, 1800, 3600, and 7200 Any other value is rounded off to the nearest intervals.	Integer	300
inventory_topology_fullsync_interval_mins	Indicates the full synchronization interval in minutes.	Integer	1440
topology_ttl_mins	Indicates the time-to-live (TTL) a record is cached in minutes.	Integer	2880
tenantname	Indicates the DX OI tenant name.	String	sampleTenantName

restapiconnectdetails

In the **restapiconnectdetails** section, enter the REST Endpoint details of your Elasticsearch environment. The following snippet is a sample of the **restapiconnectdetails** section:

```
{
        "type" : "https",
        "hostname" : "sampleHostname",
        "port" : "samplePort",
        "authentication" : "basic",
        "username" : "elastic",
        "password" : "samplePassword",
        "realmdomain" : "",
        "token" : "",
        "httptimeout" : "100000",
        "checkcert" : "no"
```

```
        },
```

Name	Description	Type	Example
type	Indicates the data transfer type with DX OI. Values: HTTP or HTTPS.	String	http
hostname	Indicates the hostname or IP address of the REST Endpoint.	String	test.example.net
port	Indicates the port number of the REST Endpoint.	Integer	9600
authentication	For Elasticsearch, enter the authentication type as **basic**.	String	basic
username	Enter the username. Applies only when the authentication type is basic or NTLM.	String	
password	Enter the password. Applies only when the authentication type is basic or NTLM.	String	
realmdomain	Indicates the DNS realm or domain to encode in the token. Applies only when the authentication type is digest.	String	
httptimeout	Indicates the value of the timeout that is expressed in milliseconds.	Integer	100000
token	Indicates the token.		
checkcert	Indicates to verify the certificate to ensure it is valid and trusted.	Boolean	no

monitored_groups

In the **monitored_groups** section, specify the groups that you want to monitor and the following snippet is a sample of the **monitored_groups** section:

```
 {
         "Nodes OS" : "yes",
         "Nodes_OS_Inventory" : "no",
         "Cluster Indices Size" : "yes",
         "Cluster Nodes File System" : "yes",
         "Cluster Nodes JVM" : "yes",
         "Cluster Nodes Process" : "yes",
         "Cluster Nodes Network Types" : "yes",
         "Cluster Nodes" : "yes",
         "Cluster Nodes OS" : "yes",
         "Cluster Indices" : "yes",
         "Nodes Ingest" : "yes",
         "Nodes Discovery Cluster State Queue" : "yes",
         "Nodes Breakers" : "yes",
         "Nodes Script" : "yes",
         "Nodes HTTP" : "yes",
         "Nodes Transport" : "yes",
```

```
        "Nodes File Store" : "yes",
        "Nodes Thread Pool" : "yes",
        "Nodes JVM Classes" : "yes",
        "Nodes JVM Buffer Pools" : "yes",
        "Nodes JVM GC Collectors" : "yes",
        "Nodes JVM Threads" : "yes",
        "Nodes JVM Memory Pools" : "yes",
        "Nodes JVM Memory" : "yes",
        "Nodes Process" : "yes",
        "Nodes JVM" : "yes",
        "Nodes Indices" : "yes",
        "Cluster Metrics" : "yes",
        "Node Stats" : "yes",
        "Index" : "yes",
        "Indices Thorottle" : "yes",
        "Indices Index" : "yes",
        "Indices Disk" : "yes",
        "Indices Segment Count" : "yes"
      }
    }
```

attribute_filter

You can configure RESTMon to ingest and display only the required information on the DX Operational Intelligence UI using the attribute filter. To filter the ingested data, add the attribute filter to the profile.json file. For more information, see the Filter Entities and Related Data Before Ingestion section.

> **NOTE**
> For the detailed steps, see the Add the Profile section.

Upload the Elasticsearch Schema

The schema is automatically uploaded when you add the profile. Perform the steps only if you want to upload the updated or edited schema.

> **NOTE**
> For the detailed steps, see the Upload the Schema section.

View Data in DX OI

You can view the ingested data in the Alarm Analytics, Service Analytics, and Performance Analytics pages in DX OI for insights into service, raw, and anomaly alarms.

> **NOTE**
> For more information, see the View Data in DX OI section.

Elasticsearch Log Data

The Elasticsearch schema is a Polling schema that polls your Elasticsearch servers at regular intervals to collect data.

This schema enables you to collect the log data of the various indexes of the Elastic environment using the REST API and ingest the following data into DX Operational Intelligence (DX OI):

- Topology
- Alarms

This section provides the following information:

Supported Versions

The Elasticsearch - DX OI integration is supported for the following version:

Product	Supported Version
Elasticsearch	SaaS

Configure the Integration

To Elasticsearch - DX OI integration involves the following steps:

- Configure the Elasticsearch Environment
- Configure RESTMon

Configure the Elasticsearch Environment

No integration-specific steps are required to be performed in your Elasticsearch environment. However, ensure that the port for your Elasticsearch server is open and accessible. For more information, see the *Elasticsearch documentation*.

Configure RESTMon

To configure RESTMon, update the OI connection details if not done earlier and add the profile information to the **restmon.json** file. Before you configure RESTMon, ensure that the following requirements are met:

- You have access to DX OI.
- RESTMon is configured and deployed successfully. For more information, see the Configure and Deploy RESTMon section.
- RESTMon is configured and deployed successfully.

Add the Profile

To add the profile, configure the profile to connect to your Elasticsearch environment and add the profile to the **restmon.json** file using the POST Profile REST API call in Swagger. The **elasticsearh_profile.json** file is available in the **<restmon\profile>** folder. When you add the profile, the schema for Elasticsearch is automatically uploaded, and the data ingestion starts.

> **NOTE**
> You can also add the attribute filter to the profile to filter the entities and the related data. For more information, see the **attribute_filter** section on this page.

You can also add this information directly in the **restmon.json** file.

The Elasticsearch profile includes the following sections:

Profile
The **profile** section defines the profile-related information, and the following snippet is a sample of the profile section:

```
{
   "profile":{
      "name":"elasticsearch",
      "active":"yes",
      "schema":"elasticsearch",
      "polling_interval_secs":"300",
      "inventory_topology_fullsync_interval_mins":"1440",
      "topology_ttl_mins":"2880",
      "tenantname":"sampleTenantName"
```

```
}
```

Name	Description	Type	Example
name	Indicates the name of the profile.	String	Elasticsearch
schema	Indicates the schema name. The name that you specify for the schema should be the same as the schema attribute specified in the restmon.json file.	String	Elasticsearch
polling_interval_secs	Indicates the polling interval in seconds. **Supported Values:** 15, 30, 60, 300, 900, 1800, 3600, and 7200 Any other value is rounded off to the nearest intervals.	Integer	300
inventory_topology_fullsync_interval_mins	Indicates the full synchronization interval in minutes.	Integer	1440
topology_ttl_mins	Indicates the time-to-live (TTL) a record is cached in minutes.	Integer	2880
tenantname	Indicates the DX OI tenant name.	String	sampleTenantName
active	Indicates if the data-processing is enabled. Enter *yes* to enable the profile.	Boolean	yes

restapiconnectdetails

In the **restapiconnectdetails** section, enter the REST Endpoint details of your Elasticsearch environment, and the following snippet is a sample of the **restapiconnectdetails** section:

```
{
    "type": "https",
    "hostname": "sampleHostname",
    "port": "",
    "authentication": "basic",
    "username": "sampleUser",
    "password": "samplePassword",
    "realmdomain": "",
    "token": "",
    "httptimeout": "30000",
    "checkcert": "no"
  }
```

Name	Description	Type	Example
type	Indicates the data transfer type with DX OI. Values: HTTP or HTTPS.	String	https
hostname	Indicates the hostname or IP address of the REST Endpoint.	String	sampleHostname

Name	Description	Type	Example
port	Indicates the port number of the REST Endpoint.	Integer	17778
authentication	Indicates the authentication type. Additionally, the following authentication types are available: • **none:** No authorization is required. • **basic:** Enter the username and password. • **NTLM:** Enter the username and password. • **digest:** Enter the username, password, and the realmdomain. • **OAuth2:** Enter the access token in the token parameter. • **bearer:** Enter the bearer token in the token parameter. • **urltoken:** Enter the token in the token parameter.	String	Bearer
username	Indicates the username. Applies only when the authentication type is basic or NTLM.	String	
password	Indicates the password. Applies only when the authentication type is basic or NTLM.	String	
realmdomain	Indicates the DNS realm or domain to encode in the token. Applies only when the authentication type is digest.	String	
token	Indicates the access token or bearer token when the authentication type is OAuth2 or bearer, respectively.	String	sampleToken
httptimeout	Indicates the value of the timeout that is expressed in milliseconds.	Integer	30000
checkcert	Indicates to verify the certificate to ensure it is valid and trusted.	Boolean	no

monitored_groups

In the **monitored_groups** section, specify the groups that you want to monitor, and the following snippet is a sample of the **monitored_groups** section:

```
{
        "Nodes OS" : "yes",
        "Nodes_OS_Inventory" : "no",
        "Cluster Indices Size" : "yes",
        "Cluster Nodes File System" : "yes",
        "Cluster Nodes JVM" : "yes",
```

```
        "Cluster Nodes Process" : "yes",
        "Cluster Nodes Network Types" : "yes",
        "Cluster Nodes" : "yes",
        "Cluster Nodes OS" : "yes",
        "Cluster Indices" : "yes",
        "Nodes Ingest" : "yes",
        "Nodes Discovery Cluster State Queue" : "yes",
        "Nodes Breakers" : "yes",
        "Nodes Script" : "yes",
        "Nodes HTTP" : "yes",
        "Nodes Transport" : "yes",
        "Nodes File Store" : "yes",
        "Nodes Thread Pool" : "yes",
        "Nodes JVM Classes" : "yes",
        "Nodes JVM Buffer Pools" : "yes",
        "Nodes JVM GC Collectors" : "yes",
        "Nodes JVM Threads" : "yes",
        "Nodes JVM Memory Pools" : "yes",
        "Nodes JVM Memory" : "yes",
        "Nodes Process" : "yes",
        "Nodes JVM" : "yes",
        "Nodes Indices" : "yes",
        "Cluster Metrics" : "yes",
        "Node Stats" : "yes",
        "Index" : "yes",
        "Indices Thorottle" : "yes",
        "Indices Index" : "yes",
        "Indices Disk" : "yes",
        "Indices Segemnt Count" : "yes"
      }
    }
```

attribute_filter

You can configure RESTMon to ingest and display only the required information on the DX Operational Intelligence UI using the attribute filter. To filter the ingested data, add the attribute filter to the profile.json file. For more information, see the Filter Entities and Related Data Before Ingestion section.

> **NOTE**
> For the detailed steps, see the Add the Profile section.

Upload the Elasticsearch Schema

The schema is automatically uploaded when you add the profile. Perform the steps only if you want to upload the updated or edited schema.

> **NOTE**
> For the detailed steps, see the Upload the Schema section.

View Data in DX OI

You can view the ingested data in the Alarm Analytics, Service Analytics, and Performance Analytics pages in DX OI for insights into service, raw, and anomaly alarms.

> **NOTE**
> For more information, see the View Data in DX OI section.

Google Cloud Monitoring

DX Operational Intelligence supports integration with Google Cloud Monitoring.

The Google Cloud Monitoring schema is a streaming schema that enables you to receive alarms and device data from the Google Cloud Monitoring service using RESTMon streaming (Webhook). The incoming data is then processed to create a topology, inventory, and alarms in DX Operational Intelligence. This schema enables you to ingest the following data:

- Inventory
- Topology
- Alarm

This section provides the following information:

Supported Versions

The Google Cloud Monitoring - DX Operational Intelligence integration is supported for the following version:

Product	Supported Version
Google Cloud Monitoring	SaaS

Configure the Integration

To Google Cloud Monitoring - DX Operational Intelligence integration involves the following steps:

- Configure the Google Cloud Monitoring Environment
- Configure RESTMon

Configure the Google Cloud Monitoring Environment

To configure the Google Cloud Monitoring environment, perform the following configurations:

- Set up the Notification Channel for Webhook
- Define an Alerting Policy

Set Up the Notification Channel

Use the Google Cloud Monitoring Admin Console to set up a notification channel for Webhook. Before you set up the channel, review the following points:

- Use the appropriate RESTMon URL while defining the notification channel. For example, *https://<fqdn>:8443/restmon/api/v1/logs?profileName=googlecloudmonitoring&schemaName=googlecloudmonitoring.*
- Generate a valid SSL certificate for the system that is hosting RESTMon. Google does not support a self-signed certificate.
- Ensure that you know the username and password for the RESTMon installation. This information is required while setting up the notification channel.
- When you create the notification channel, provide a name. For example, *DX-RESTmon*.

For more information, see the Managing Notification Channels documentation.

Define the Alert Policy

After you set up the notification channel, define the alerting policy in Google Cloud Monitoring. When you define the alerting policy, select the notification channel (for example, *DX-RESTmon*) that was set up in the preceding section. For more information, see the Managing Alerting Policies documentation.

After the notification channel and alerting policy is defined, Google Cloud Monitoring sends JSON to RESTMon. A sample JSON is as follows:

```
{
    "incident": {
      "incident_id": "g2f19d444ed75dc10e86fbbc466404ca",
      "resource_id": "i-5b377b3e",
      "resource_name": "webser-96",
      "state": "open",
      "started_at": 1385085727,
      "ended_at": null,
      "policy_name": "Webserver Health",
      "condition_name": "CPU usage",
      "url": "https://console.cloud.google.com/monitoring/alerting/incidents?project=PROJECT_ID",
      "summary": "CPU for webser-96 is above the threshold of 1% with a value of 28.5%"
    },
    "version": 1.1
  }
```

Configure RESTMon

To configure RESTMon, update the OI connection details if not done earlier, and add the profile information to the **restmon.json** file. Before you configure RESTMon, ensure that the following requirements are met:

- You have access to DX Operational Intelligence.
- RESTMon is configured and deployed successfully. For more information, see the Configure and Deploy RESTMon section.
- RESTMon is configured and deployed successfully.

Add the Profile

To add the profile, configure the profile to connect to your Google Cloud Monitoring environment and add the profile to the **restmon.json** file using the POST Profile REST API call in Swagger. The **googlecloudmonitoring_profile.json** file is available in the **<restmon\profile>** folder. When you add the profile, the schema for Google Cloud Monitoring is automatically uploaded, and the data ingestion starts.

> **NOTE**
> You can also add the attribute filter to the profile to filter the entities and the related data. For more information, see the **attribute_filter** section on this page.

You can also add this information directly in the **restmon.json** file.

The Google Cloud Monitoring profile includes the following sections:

Profile

The **profile** section defines the profile-related information. The following snippet is a sample of the profile section.

```
"{
        "name":"googlecloudmonitoring",
        "active":"yes",
        "schema":"googlecloudmonitoring",
        "streaming":"yes",
        "polling_interval_secs":1,
        "inventory_topology_fullsync_interval_mins":"1",
        "topology_ttl_mins":"15",
        "streaming_array_size" : 10,
```

```
        "is_array_input" : "true",
        "batch_size":10,
        "batch_wait_time_milli":2000,
        "tenantname":"TENANTNAME"
    }
```

Name	Description	Type	Example
name	Indicates the name of the profile.	String	googlecloudmonitoring
active	Indicates whether the data processing is enabled. Enter *yes* to enable the profile.	Boolean	yes
schema	Indicates the schema name. The name that you specify for the schema should be the same as the schema attribute specified in the restmon.json file.	String	googlecloudmonitoring
streaming	Indicates whether streaming is enabled. If enabled, the integrating product posts data to RESTMon as JSON (Webhook). If not, RESTMon gets (polling) data from the integrating product.	Boolean	yes
polling_interval_secs	Indicates the polling interval in seconds. **Supported Values:** 15, 30, 60, 300, 900, 1800, 3600, and 7200 Any other value is rounded off to the nearest intervals.	Integer	1
inventory_topology_fullsync_interval_mins	Indicates the full synchronization interval in minutes.	Integer	1
topology_ttl_mins	Indicates the time-to-live (TTL) record is cached in minutes.	Integer	15
batch_size	Indicates the size of the batch for the incoming data.	Integer	10
batch_wait_time_milli	Indicates the wait time for the batch.	Integer	2000
tenantname	Indicates the DX OI tenant name.	String	TENANTNAME

servicedefinition

In the **servicedefinition** section, define the service that appears when data is ingested into DX OI.

```
{
        "name":"Google Cloud Monitoring",
        "status":"Active"
```

```
    },
```

Name	Description	Type	Example
name	Indicates the name of the service.	String	Google Cloud Monitoring
status	Indicates the status of the defined service.	String	Active

restapiconnectdetails

In the **restapiconnectdetails** section, specify the REST Endpoint details of your Google Cloud Monitoring environment. The following is a sample snippet of the **restapiconnectdetails** section:

```
 {
        "type":"http",
        "hostname":"dummy.hostname.com",
        "port":9600,
        "authentication":"",
        "username":"",
        "password":"",
        "realmdomain":"",
        "token":"",
        "httptimeout":300,
        "checkcert":"no"
     },
```

Name	Description	Type	Example
type	Indicates the data transfer type with OI. Values: HTTP or HTTPS.	String	http
hostname	Indicates the hostname or IP address of the REST endpoint.	String	dummy.hostname.com
port	Indicates the port number of the REST endpoint.	Integer	9600

Name	Description	Type	Example
authentication	Indicates the authentication type. For the Google Cloud Monitoring integration, you can set **none**. Additionally, the following authentication types are available: • **none:** No authorization is required. • **basic:** Enter the username and password. • **NTLM:** Enter the username and password. • **digest:** Enter the username, password, and the realm domain. • **OAuth2:** Enter the access token in the token parameter. • **bearer:** Enter the bearer token in the token parameter. • **urltoken:** Enter the token in the token parameter.	String	
username	Indicates the username. Applies only when the authentication type is basic or NTLM.	String	
password	Indicates the password. Applies only when the authentication type is basic or NTLM.	String	
realmdomain	Indicates the DNS realm or domain to encode in the token. Applies only when the authentication type is digest.	String	
token	Indicates the access token or bearer token when the authentication type is OAuth2 or bearer, respectively.	String	
httptimeout	Indicates the value of the timeout that is expressed in milliseconds.	Integer	300
checkcert	Indicates to verify the certificate to ensure it is valid and trusted.	Boolean	no

monitored_groups

In the **monitored_groups** section, specify the groups that you want to monitor. The following snippet is a sample of the **monitored_groups** section:

```
{
      "Topology":"yes",
      "Events":"yes
```

```
}
```

attribute_filter

You can configure RESTMon to ingest and display only the required information on the DX Operational Intelligence UI using the attribute filter. To filter the ingested data, add the attribute filter to the profile.json file. For more information, see the Filter Entities and Related Data Before Ingestion section.

> **NOTE**
> For the detailed steps, see the Add the Profile section.

Upload the Google Cloud Monitoring Schema

The schema is automatically uploaded when you add the profile. Perform the steps only if you want to upload the updated or edited schema.

> **NOTE**
> For the detailed steps, see the Upload the Schema section.

View Data in DX OI

You can view the ingested data in the Alarm Analytics, Service Analytics, and Performance Analytics pages in DX OI for insights into service, raw, and anomaly alarms.

> **NOTE**
> For more information, see the View Data in DX OI section.

Microsoft SCOM

The Microsoft SCOM schema enables you to monitor the services, devices, and operations for computers on your network by ingesting the metrics and alerts to DX Operational Intelligence (DX OI).

This schema ingests the following monitoring data:

- **Inventory**
- **Metrics** (Metrics include Memory, Logical Disk, Health Service, Network Adapter, Health Management Group, System, and Processor Information)
- **Alarms**
- **Topology**

This section provides the following information:

Configure the Integration

To Microsoft SCOM - DX OI integration involves the following steps:

- Configure the SCOM Environment
- Configure RESTMon

Configure the Microsoft SCOM Environment

No integration-specific steps are required to be performed in the Microsoft SCOM environment. However, ensure that the following requirements are met:

- You have enabled Internet Information Services 6.0 or later on your SCOM server.
- You have Administrator privileges to the SCOM server.
- If communications between the SCOM Server and RESTMon must pass through a proxy, ensure you have the proxy details, including IP address, port, and the required user credentials.

NOTE
For more information, see the *SCOM documentation*.

Configure RESTMon

To configure RESTMon, update the OI connection details if not done earlier, and add the profile information to the **restmon.json** file. Before you configure RESTMon, ensure that the following requirements are met:

- You have access to DX OI.
- RESTMon is configured and deployed successfully. For more information, see the Configure and Deploy RESTMon section.
- RESTMon is configured and deployed successfully.
- For Java 11, ensure that you replace the *contains* method with *includes* in the schema.

Add the Profile

To add the profile, configure the profile to connect to your SCOM environment and add the profile to the **restmon.json** file using the POST Profile REST API call in Swagger. The **scom_profile.json** file is available in the **<restmon\profile>** folder. When you add the profile, the schema for SCOM is automatically uploaded, and the data ingestion starts.

NOTE
You can also add the attribute filter to the profile to filter the entities and the related data. For more information, see the **attribute_filter** section on this page.

You can also add this information directly in the **restmon.json** file.

The Microsoft SCOM profile includes the following sections:

Profile

The **profile** section defines the profile-related information. The following snippet is a sample of the profile section.

```
{
        "name" : "scom",
        "active" : "yes",
        "schema" : "scom",
        "polling_interval_secs" : "300",
        "inventory_topology_fullsync_interval_mins" : "1440",
        "topology_ttl_mins" : "2880"
      }
```

Name	Description	Type	Example
name	Indicates the name of the profile.	String	scom
active	Indicates if the data-processing is enabled. Specify *yes* to enable the profile.	Boolean	yes
schema	Indicates the schema name. The name that you specify for the schema should be the same as the schema attribute specified in the restmon.json file.	String	scom

Name	Description	Type	Example
polling_interval_secs	Indicates the polling interval in seconds. **Supported Values:** 15, 30, 60, 300, 900, 1800, 3600, and 7200 Any other value is rounded off to the nearest intervals.	Integer	10
inventory_topology_fullsync_interval_mins	Indicates the full synchronization interval in minutes.	Integer	1440
topology_ttl_mins	Indicates the time-to-live (TTL) a record is cached in minutes.	Integer	2880
tenantname	Indicates the DX Operational Intelligence tenant name.	String	sampleTenantName

restapiconnectdetails

In the **restapiconnectdetails** section, enter the REST Endpoint details of your SCOM environment. The following snippet is a sample of the **restapiconnectdetails** section:

```
{
        "type" : "http",
        "hostname" : "sampleHostname",
        "port" : "",
        "authentication" : "ntlm",
        "username" : "sampleUsername",
        "password" : "samplePassword",
        "realmdomain" : "",
        "token" : "",
        "httptimeout" : "120000",
        "checkcert" : "no"
      }
```

Name	Description	Type	Example
type	Indicates the data transfer type with OI. Values: HTTP or HTTPS.	String	http
hostname	Indicates the hostname or IP address of the REST Endpoint.	String	test.example.net
port	Indicates the port number of the REST Endpoint.	Integer	9600
authentication	Indicates the authentication type as NTLM.	String	ntlm
username	Indicates the username. Applies only when the authentication type is basic or NTLM.	String	
password	Indicates the password. Applies only when the authentication type is basic or NTLM.	String	
httptimeout	Indicates the value of the timeout that is expressed in milliseconds.	Integer	120000

Name	Description	Type	Example
checkcert	Indicates to verify the certificate to ensure it is valid and trusted.	Boolean	no

monitored_groups

In the **monitored_groups** section, specify the groups that you want to monitor. The following snippet is a sample of the **monitored_groups** section:

```
{
        "Connections" : "yes",
        "state_Inventory" : "no",
        "Performance" : "yes",
        "Alerts" : "yes",
        "state" : "yes",
        "Monitor" : "yes",
        "Events" : "no"
      }
```

attribute_filter

You can configure RESTMon to ingest and display only the required information on the DX Operational Intelligence UI using the attribute filter. To filter the ingested data, add the attribute filter to the profile.json file. For more information, see the Filter Entities and Related Data Before Ingestion section.

> **NOTE**
> For the detailed steps, see the Add the Profile section.

Upload the SCOM Schema

The schema is automatically uploaded when you add the profile. Perform the steps only if you want to upload the updated or edited schema.

> **NOTE**
> For the detailed steps, see the Upload the Schema section.

View Data in DX OI

You can view the ingested data in the Alarm Analytics, Service Analytics, and Performance Analytics pages in DX OI for insights into service, raw, and anomaly alarms.

> **NOTE**
> For more information, see the View Data in DX OI section.

MongoDB

DX Operational Intelligence supports integration with MongoDB.

The MongoDB schema is a Polling schema that enables you to pull alerts, hosts, and clusters from the MongoDB Cloud Manager and ingest this data into DX Operational Intelligence (DX OI). This schema polls for the following information:

- Alarms
- Topology

> **NOTE**
> Severity mapping is not supported as MongoDB alerts do not have any severity.

This section provides the following information:

Supported Versions

The MongoDB - DX OI integration is supported for the following version:

Product	Supported Version
MongoDB	SaaS

Configure the Integration

The MongoDB - DX OI integration involves the following steps:

- Configure the MongoDB Environment
- Configure RESTMon

Configure the MongoDB Environment

No integration-specific steps are required to be performed in your MongoDB environment. However, create the API Key within the MongoDB Cloud Manager and note the Private Key and the Public Key.

Configure RESTMon

To configure RESTMon, update the OI connection details if not done earlier, and add the profile information to the **restmon.json** file. Before you configure RESTMon, ensure that the following requirements are met:

- You have access to DX OI.
- RESTMon is configured and deployed successfully. For more information, see the Configure and Deploy RESTMon section.
- RESTMon is configured and deployed successfully.

Add the Profile

To add the profile, configure the profile to connect to your MongoDB environment and add the profile to the **restmon.json** file using the POST Profile REST API call in Swagger. The **mongodb_profile.json** file is available in the **<restmon \profile>** folder. When you add the profile, the schema for MongoDB is automatically uploaded, and the data ingestion starts.

> **NOTE**
> You can also add the attribute filter to the profile to filter the entities and the related data. For more information, see the **attribute_filter** section on this page.

You can also add this information directly in the **restmon.json** file.

The MongoDB profile includes the following sections:

Profile

The **profile** section defines the profile-related information, and the following snippet is a sample of the profile section:

```
{
"profile" : {
"name" : "mongodb",
"active" : "yes",
"schema" : "mongodb",
"polling_interval_secs" : "60",
"inventory_topology_fullsync_interval_mins" : "1440",
"topology_ttl_mins" : "2880"
```

```
},
```

Name	Description	Type	Example
name	Indicates the name of the profile.	String	mongodb
active	Indicates if the data-processing is enabled. Enter *yes* to enable the profile.	Boolean	yes
schema	Indicates the schema name. The name that you specify for the schema should be the same as the schema attribute specified in the restmon.json file.	String	mongodb
polling_interval_secs	Indicates the polling interval in seconds. **Supported Values:** 15, 30, 60, 300, 900, 1800, 3600, and 7200 Any other value is rounded off to the nearest intervals.	Integer	60
inventory_topology_fullsync_interval_mins	Indicates the full synchronization interval in minutes.	Integer	1440
topology_ttl_min	Indicates the time-to-live (TTL) a record is cached in minutes.	Integer	2880

restapiconnectdetails

In the **restapiconnectdetails** section, enter the REST Endpoint details of your MongoDB environment, and the following snippet is a sample of the **restapiconnectdetails** section:

```
"restapiconnectdetails" : {
"type" : "https",
"hostname" : "cloud.mongodb.com",
"port" : "443",
"authentication" : "digest",
"username" : "samplePublicKey",
"password" : "samplePrivateKey",
"realmdomain" : "MMS Public API",
"token" : "",
"httptimeout" : "60000",
"checkcert" : "no"
},
```

Name	Description	Type	Example
type	Indicates the data transfer type with DX OI. Values: HTTP or HTTPS.	String	https
hostname	Indicates the hostname or IP address of the REST Endpoint.	String	sampleHostname
port	Indicates the port number of the REST Endpoint.	Integer	443

Name	Description	Type	Example
authentication	Indicates the authentication type. The following authentication types are available: • **none:** No authorization is required. • **basic:** Enter the username and password. • **NTLM:** Enter the username and password. • **digest:** Enter the username, password, and the realmdomain. • **OAuth2:** Enter the access token in the token parameter. • **bearer:** Enter the bearer token in the token parameter. • **urltoken:** Enter the token in the token parameter.	String	none
username	Indicates the username. Enter the Public Key.	String	samplePublicKey
password	Indicates the password. Enter the Private Key.	String	samplePrivateKey
realmdomain	Indicates the DNS realm or domain to encode in the token. Applies only when the authentication type is digest.	String	
token	Indicates the access token or bearer token when the authentication type is OAuth2 or bearer, respectively.	String	sampleToken
httptimeout	Indicates the value of the timeout that is expressed in milliseconds.	Integer	60000
checkcert	Indicates to verify the certificate to ensure it is valid and trusted.	Boolean	no

monitored_groups

In the **monitored_groups** section, specify the groups that you want to monitor, and the following snippet is a sample of the **monitored_groups** section:

```
"monitored_groups": {
    "Topology": "yes",
    "Alerts": "yes",
    }
```

attribute_filter

You can configure RESTMon to ingest and display only the required information on the DX Operational Intelligence UI using the attribute filter. To filter the ingested data, add the attribute filter to the profile.json file. For more information, see the Filter Entities and Related Data Before Ingestion section.

NOTE
For the detailed steps, see the Add the Profile section.

Upload the MongoDB Schema

The schema is automatically uploaded when you add the profile. Perform the steps only if you want to upload the updated or edited schema.

NOTE
For the detailed steps, see the Upload the Schema section.

View Data in DX OI

You can view the ingested data in the Alarm Analytics, Service Analytics, and Performance Analytics pages in DX OI for insights into service, raw, and anomaly alarms.

NOTE
For more information, see the View Data in DX OI section.

Nagios

DX Operational Intelligence supports integration with Nagios.

The Nagios schema is a Polling schema that queries the Nagios REST APIs to get the data. This schema enables you to retrieve events from Nagios and send them to DX Operational Intelligence (OI) using RESTMon.

This section provides the following information:

Supported Versions

The Nagios - DX OI integration is supported for the following version:

Product	Supported Version
Nagios	Nagios XI

Configure the Integration

To Nagios - DX OI integration involves the following steps:

- Configure the Nagios Environment
- Configure RESTMon

Configure the Nagios Environment

Configure the Alert and Respond system to send event data to DX OI. Before you set up the environment, ensure that the following requirements are met:

- You have an active Nagios installation.
- You have full permissions to the Nagios installation directory and files.
- You have installed CURL on the Nagios server.
- You have administrator access to the Nagios UI.
- Your Nagios environment can make requests to external endpoints over port 443.

Follow these steps:

1. Create an HTTPS request template in the Nagios Alert and Respond UI.

2. Create an action for each Nagios business application that you want to report events to DX AIOps.
3. Create a policy that applies the **Send to AIOps** action to health rules for each Nagios business application that should report events to DX OI.

> **NOTE**
> For more information, see the Nagios documentation.

Configure RESTMon

To configure RESTMon, update the OI connection details if not done earlier, and add the profile information to the **restmon.json** file. Before you configure RESTMon, ensure that the following requirements are met:

- You have access to DX OI.
- RESTMon is configured and deployed successfully. For more information, see the Configure and Deploy RESTMon section.
- RESTMon is configured and deployed successfully.
- For Java 11, replace the **contains** method with **includes** in the schema.

Add the Profile

To add the profile, configure the profile to connect to your Nagios environment and add the profile to the **restmon.json** file using the POST Profile REST API call in Swagger. The **nagios_profile.json** file is available in the **<restmon\profile>** folder. When you add the profile, the schema for Nagios is automatically uploaded, and the data ingestion starts.

> **NOTE**
> You can also add the attribute filter to the profile to filter the entities and the related data. For more information, see the **attribute_filter** section on this page.

You can also add this information directly in the **restmon.json** file.

The Nagios profile includes the following sections:

Profile

The **profile** section defines the profile-related information. The following snippet is a sample of the profile section.

```
{
        "name": "nagios",
        "schema": "nagios",
        "streaming": "no",
        "polling_interval_secs": "300",
        "inventory_topology_fullsync_interval_mins": "1440",
        "topology_ttl_mins": "2880"
        "active": "yes"
    }
```

In the **profile** section, specify the following details:

Name	Description	Type	Example
name	Indicates the name of the profile.	String	Nagios
active	Indicates if the data-processing is enabled. Enter *yes* to enable the profile.	Boolean	yes

Name	Description	Type	Example
schema	Indicates the schema name. The name that you specify for the schema should be the same as the schema attribute specified in the restmon.json file.	String	Nagios
polling_interval_secs	Indicates the polling interval in seconds. **Supported Values:** 15, 30, 60, 300, 900, 1800, 3600, and 7200 Any other value is rounded off to the nearest intervals.	Integer	60
inventory_topology_fullsync_interval_mins	Indicates the full synchronization interval in minutes.	Integer	1440
topology_ttl_min	Indicates the time-to-live (TTL) a record is cached in minutes.	Integer	2880
tenantname	Indicates the DX OI tenant name.	String	A00B111C-22DB-33C3-444C-000F0000B000

restapiconnectdetails

In the **restapiconnectdetails** section, enter the REST Endpoint details of your Nagios environment. The following snippet is a sample of the **restapiconnectdetails** section:

```
{
        "type": "https",
        "hostname": "sampleHostName",
        "port": "",
        "authentication": "urltoken",
        "username": "",
        "password": "",
        "realmdomain": "",
        "token": "qwkuxnciprnttl2hy5n6ons9brny4",
        "httptimeout": "30000",
        "checkcert": "no"
    },
```

Name	Description	Type	Example
type	Indicates the data transfer type with DX OI. Values: HTTP or HTTPS.	String	https
hostname	Indicates the hostname or IP address of the REST Endpoint.	String	sampleHostname
port	Indicates the port number of the REST Endpoint.	Integer	17778

Name	Description	Type	Example
authentication	Indicates the authentication type. For the Nagios integration, you can set **basic** or **bearer**. Additionally, the following authentication types are available: • **none:** No authorization is required. • **basic:** Enter the username and password. • **NTLM:** Enter the username and password. • **digest:** Enter the username, password, and the realmdomain. • **OAuth2:** Enter the access token in the token parameter. • **bearer:** Enter the bearer token in the token parameter. • **urltoken:** Enter the token in the token parameter.	String	Bearer
username	Indicates the username. Applies only when the authentication type is basic or NTLM.	String	
password	Indicates the password. Applies only when the authentication type is basic or NTLM.	String	
realmdomain	Indicates the DNS realm or domain to encode in the token. Applies only when the authentication type is digest.	String	
token	Indicates the access token or bearer token when the authentication type is OAuth2 or bearer, respectively.	String	sampleToken
httptimeout	Indicates the value of the timeout that is expressed in milliseconds.	Integer	30000
checkcert	Indicates to verify the certificate to ensure it is valid and trusted.	Boolean	no

monitored_groups

In the **monitored_groups** section, specify the groups that you want to monitor. The following snippet is a sample of the **monitored_groups** section:

```
{
        "inventory": "no",
        "host_topology": "yes",
        "service_topology": "yes",
        "host_groups": "yes",
```

```
        "service_groups": "yes",
        "disk_metric": "yes",
        "cpu_metric": "yes",
        "memory_metric": "yes",
        "load_metric": "yes",
        "host_metric": "yes",
        "open_files_metric": "yes",
        "users_metric": "yes",
        "total_process_metric": "yes",
        "protocol_metric": "yes",
        "host_alarm": "yes",
        "service_alarm": "yes"
    }
```

attribute_filter

You can configure RESTMon to ingest and display only the required information on the DX Operational Intelligence UI using the attribute filter. To filter the ingested data, add the attribute filter to the profile.json file. For more information, see the Filter Entities and Related Data Before Ingestion section.

> **NOTE**
> For the detailed steps, see the Add the Profile section.

Upload the Nagios Schema

The schema is automatically uploaded when you add the profile. Perform the steps only if you want to upload the updated or edited schema.

> **NOTE**
> For the detailed steps, see the Upload the Schema section.

View Data in DX OI

You can view the ingested data in the Alarm Analytics, Service Analytics, and Performance Analytics pages in DX OI for insights into service, raw, and anomaly alarms.

> **NOTE**
> For more information, see the View Data in DX OI section.

New Relic

DX Operational Intelligence supports integration with New Relic.

The New Relic schema is a Polling schema that enables you to collect application and infrastructure metrics and alarms and ingest that data into DX Operational Intelligence (DX OI).

This section provides the following information:

Supported Versions

The New Relic - DX OI integration is supported for the following version:

Product	Supported Version
New Relic	SaaS

Configure the Integration

The New Relic - DX OI integration involves the following steps:

- Configure the New Relic Environment
- Configure RESTMon

Configure the New Relic Environment

No integration-specific steps are required to be performed in your New Relic environment. However, ensure that the following requirements are met:

- You have the Account ID for the SaaS version of New Relic
- Token is created. For more information, see the New Relic documentation.

Configure RESTMon

To configure RESTMon, update the OI connection details if not done earlier and add the profile information to the **restmon.json** file. Before you configure RESTMon, ensure that the following requirements are met:

- You have access to DX OI.
- RESTMon is configured and deployed successfully. For more information, see the Configure and Deploy RESTMon section.
- RESTMon is configured and deployed successfully.

Add the Profile

To add the profile, configure the profile to connect to your New Relic environment and add the profile to the **restmon.json** file using the POST Profile REST API call in Swagger. The **newrelic_profile.json** file is available in the **<restmon \profile>** folder. When you add the profile, the schema for New Relic is automatically uploaded and the data ingestion starts.

> **NOTE**
> You can also add the attribute filter to the profile to filter the entities and the related data. For more information, see the **attribute_filter** section on this page.

You can also add this information directly in the **restmon.json** file.

The New Relic profile includes the following sections:

Profile

The **profile** section defines the profile-related information and the following snippet is a sample of the profile section:

```
{
  "profile": {
    "name": "newrelic",
    "active": "yes",
    "schema": "newrelic",
    "batch_size": 1,
    "batch_wait_time_milli": 5000,
    "streaming_array_size": 10,
    "is_array_input": "true",
    "polling_interval_secs": "300",
    "inventory_topology_fullsync_interval_mins": "1440",
    "topology_ttl_mins": "2880"
```

```
},
```

Name	Description	Type	Example
name	Indicates the name of the profile.	String	newrelic
active	Indicates if the data-processing is enabled. Enter *yes* to enable the profile.	Boolean	yes
schema	Indicates the schema name. The name that you specify for the schema should be the same as the schema attribute specified in the restmon.json file.	String	newrelic
batch_size	Indicates the size of batches with which processing happens.	Integer	1
batch_wait_time_milli	Indicates the wait time that is given to fill up the batch size. If the wait time is exceeded, then existing batch is executed.	Integer	5000
streaming_array_size	Indicates the array size to be considered while processing.	Integer	10
is_array_input	Indicates if the payload for processing is in an array format.	Boolean	true
polling_interval_secs	Indicates the polling interval in seconds. **Supported Values:** 15, 30, 60, 300, 900, 1800, 3600, and 7200 Any other value is rounded off to the nearest intervals.	Integer	300
inventory_topology_fullsync_interval_mins	Indicates the full synchronization interval in minutes.	Integer	1440
topology_ttl_min	Indicates the time-to-live (TTL) a record is cached in minutes.	Integer	2880
tenantname	Indicates the DX Operational Intelligence tenant name.	String	A00B111C-22DB-33C3-444C-000F0000B000

servicedefinition

In the **servicedefinition** section, define the service that appears when data is ingested to DX OI. The following snippet is a sample of the **servicedefinition** section:

```
"servicedefinition":{   "name":"",
   "status":""
},
```

Name	Description	Type	Example
name	Indicates the name of the service.	String	newrelic
status	Indicates the status of the defined service.	String	Active

restapiconnectdetails

In the **restapiconnectdetails** section, enter the REST Endpoint details of your New Relic environment and the following snippet is a sample of the **restapiconnectdetails** section:

```
{
    "type": "https",
    "hostname": "insights-api.newrelic.com",
    "port": "443",
    "authentication": "none",
    "username": "",
    "password": "",
    "realmdomain": "",
    "token": "sampleToken",
    "httptimeout": "120000",
    "checkcert": "no",
    "accountId": "sampleAccountID"
  },
```

Name	Description	Type	Example
type	Indicates the data transfer type with DX OI. Values: HTTP or HTTPS.	String	https
hostname	Indicates the hostname or IP address of the REST Endpoint.	String	sampleHostname
port	Indicates the port number of the REST Endpoint.	Integer	443
authentication	Indicates the authentication type. The following authentication types are available: • **none:** No authorization is required. • **basic:** Enter the username and password. • **NTLM:** Enter the username and password. • **digest:** Enter the username, password, and the realmdomain. • **OAuth2:** Enter the access token in the token parameter. • **bearer:** Enter the bearer token in the token parameter. • **urltoken:** Enter the token in the token parameter.	String	none
username	Indicates the username. Applies only when the authentication type is basic or NTLM.	String	

Name	Description	Type	Example
password	Indicates the password. Applies only when the authentication type is basic or NTLM.	String	
realmdomain	Indicates the DNS realm or domain to encode in the token. Applies only when the authentication type is digest.	String	
token	Indicates the access token that is generated in New Relic.	String	sampleToken
httptimeout	Indicates the value of the timeout that is expressed in milliseconds.	Integer	120000
checkcert	Indicates to verify the certificate to ensure it is valid and trusted.	Boolean	no
accountId	Indicates the New Relic account ID.		sampleAccountID

monitored_groups

In the **monitored_groups** section, specify the groups that you want to monitor, and the following snippet is a sample of the **monitored_groups** section:

```
"monitored_groups": {
    "Topology": "yes",
    "Alerts": "yes",
    "ApplicationAgentCheck": "no",
    "ApplicationAjax": "no",
    "ApplicationApex": "no",
    "ApplicationApmSummary": "yes",
    "ApplicationBjax": "no",
    "ApplicationBrowser": "no",
    "ApplicationBrowserSummary": "yes",
    "ApplicationCPU": "no",
    "ApplicationDatastore": "no",
    "ApplicationEndUser": "no",
    "ApplicationError": "no",
    "ApplicationErrors": "no",
    "ApplicationExternal": "no",
    "ApplicationGC": "no",
    "ApplicationInstance": "no",
    "ApplicationJava": "no",
    "ApplicationJMXBuiltIn": "no",
    "ApplicationMemory": "no",
    "ApplicationMemoryPool": "no",
    "ApplicationNetwork": "no",
    "ApplicationOtherTransaction": "no",
    "ApplicationSupportability": "no",
    "ApplicationThreads": "no",
    "ApplicationTransaction": "no",
    "ApplicationWebFrontendErrors": "no",
    "InfraCPU": "no",
    "InfraDisk": "no",
```

```
    "InfraHostSummary": "yes",
    "InfraMemory": "no",
    "InfraNetwork": "no",
    "InfraProcess": "no",
    "InfraSwap": "no"
  }
```

attribute_filter

You can configure RESTMon to ingest and display only the required information on the DX Operational Intelligence UI using the attribute filter. To filter the ingested data, add the attribute filter to the profile.json file. For more information, see the Filter Entities and Related Data Before Ingestion section.

> **NOTE**
> For the detailed steps, see the Add the Profile section.

Upload the New Relic Schema

The schema is automatically uploaded when you add the profile. Perform the steps only if you want to upload the updated or edited schema.

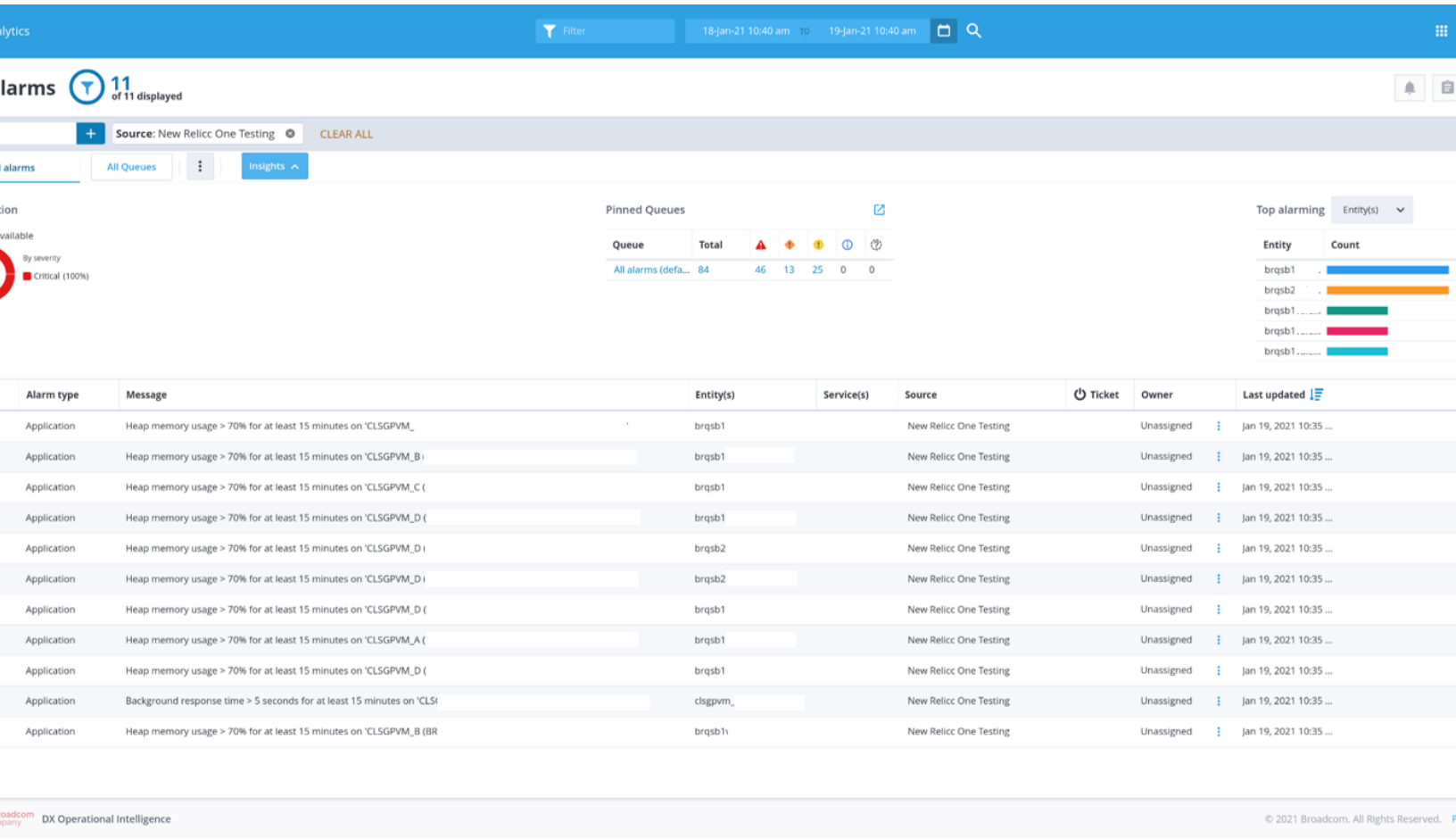
> **NOTE**
> For the detailed steps, see the Upload the Schema section.

View Data in DX OI

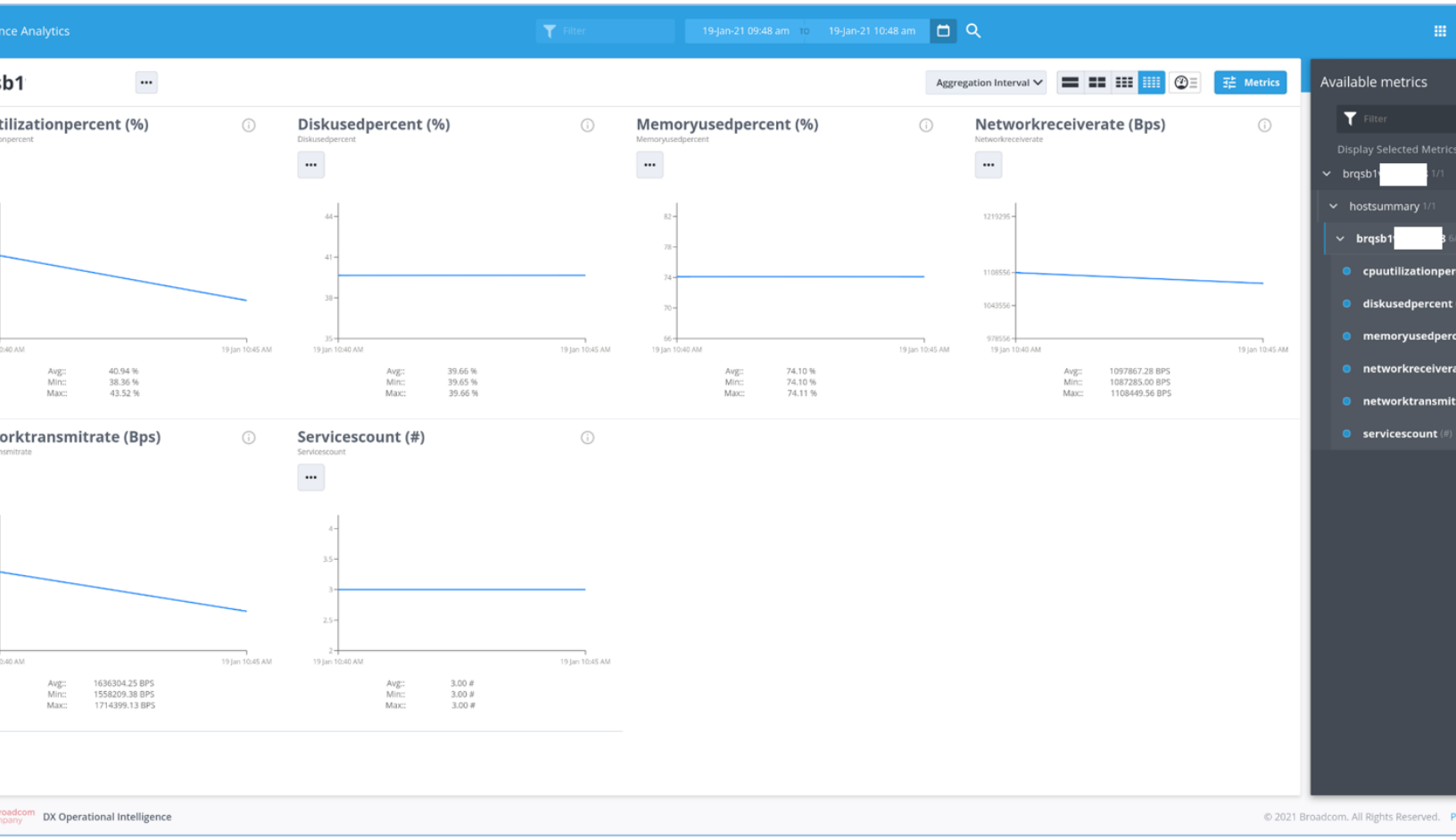
You can view the ingested data in the Alarm Analytics, Service Analytics, and Performance Analytics pages in DX OI for insights into service, raw, and anomaly alarms.

> **NOTE**
> For more information, see the View Data in DX OI section.

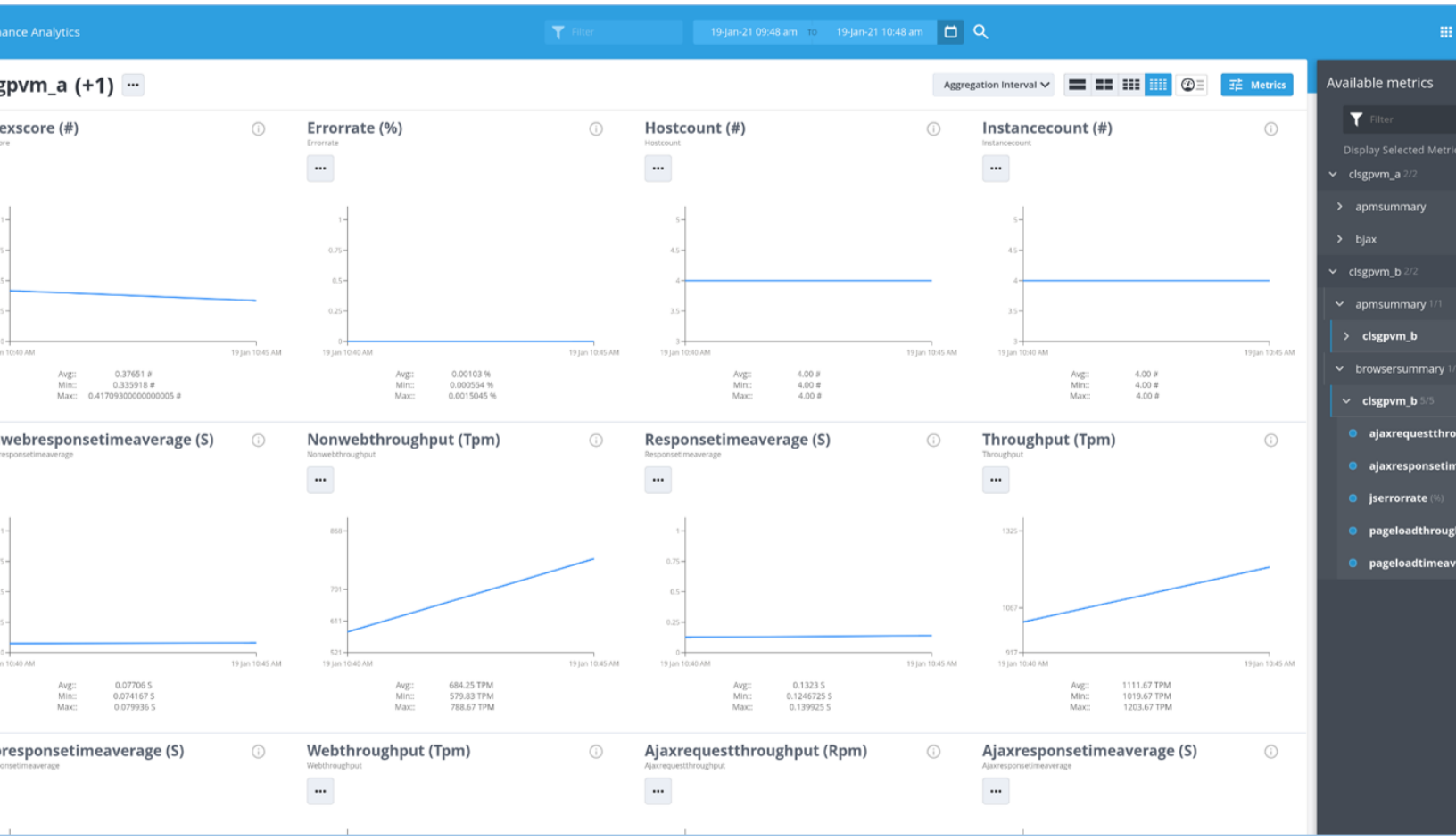
The following image illustrates the All Alarms page in DX OI:



The following image illustrates the summary of the metrics for the host:



The following image illustrates the summary of the metrics for an application:



Pure Storage

DX Operational Intelligence supports integration with Pure Storage.

The Pure Storage schema enables you to monitor the performance and usage of the Flash Array systems. This schema enables you to ingest metrics and messages to DX Operational Intelligence (DX OI) using RESTMon.

Pure Storage integration monitors the following information:

- **Inventory**: A single Flash Array for inventory that specifies the Pure Storage Flash Array name.
- **Metrics**: Metrics include Volume Space, Volume Performance, Array Space, and Array Performance.
- **Alarms**: The Pure Storage Category is mapped to the OI AlarmType, and Pure Storage Opened is mapped to OI Start Time. The Alarm Unique ID is created using the Pure Storage Array Name, Component Name, and Component Type.
- **Topology**: A single Flash Array for topology which specifies the Pure Storage Flash Array name.

This section provides the following information:

Supported Versions

The Pure Storage - DX OI integration is supported for the following version:

Product	Supported Version
Pure Storage	REST API version 1.17

Configure the Integration

To Pure Storage - DX OI integration involves the following steps:

- Configure the Pure Storage Environment
- Configure RESTMon

Configure the Pure Storage Environment

No specific integration steps are required to be performed in the Pure Storage environment. However, ensure that the following requirements are met:

- Set up your Pure Storage environment to retrieve Pure Storage metrics and messages. For more information, see the *Pure Storage Documentation*.
- API token to authenticate the connection between RESTMon and the Pure Storage system. You can retrieve the API token from the Pure Storage GUI.

Configure RESTMon

To configure RESTMon, update the OI connection details if not done earlier and add the profile information to the **restmon.json** file. Before you configure RESTMon, ensure that the following requirements are met:

- You have access to DX OI.
- RESTMon is installed and deployed successfully. For more information, see the Configure and Deploy RESTMon section.
- RESTMon is deployed successfully.

Add the Profile

To add the profile, configure the profile to connect to your Pure Storage environment and add the profile to the **restmon.json** file using the POST Profile REST API call in Swagger. The **purestorage_profile.json** file is available in the **<restmon\profile>** folder. When you add the profile, the schema for Pure Storage is automatically uploaded and the data ingestion starts.

> **NOTE**
> You can also add the attribute filter to the profile to filter the entities and the related data. For more information, see the **attribute_filter** section on this page.

You can also add this information directly in the **restmon.json** file.

The Pure Storage profile includes the following sections:

Profile

The **profile** section defines the profile-related information. The following snippet is a sample of the profile section.

```
":{
        "name":"purestorage",
        "schema":"purestorage",
        "streaming":"no",
        "polling_interval_secs":"300",
        "inventory_topology_fullsync_interval_mins":"1440",
        "topology_ttl_mins":"2880",
        "active":"yes"
```

```
    },
```

Name	Description	Type	Example
name	Indicates the name of the profile.	String	purestorage
active	Indicates if the data-processing is enabled. Enter *yes* to enable the profile.	Boolean	yes
schema	Indicates the schema name. The name that you specify for the schema should be the same as the schema attribute specified in the restmon.json file.	String	purestorage
streaming	Indicates if the data streaming is enabled.	Boolean	no
polling_interval_secs	Indicates the polling interval in seconds. **Supported Values:** 15, 30, 60, 300, 900, 1800, 3600, and 7200 Any other value is rounded off to the nearest intervals.	Integer	10
inventory_topology_fullsync_interval_mins	Indicates the full synchronization interval in minutes.	Integer	1440
topology_ttl_mins	Indicates the time-to-live (TTL) a record is cached in minutes.	Integer	2880
tenantname	Indicates the DX OI tenant name.	String	sampleTenantName

restapiconnectdetails

In the **restapiconnectdetails** section, enter the REST Endpoint details of your Pure Storage environment. The following snippet is a sample of the **restapiconnectdetails** section:

```
 ":{
        "type":"https",
        "hostname":"sampleHostname",
        "port":"",
        "authentication":"urltoken",
        "token":"sampleToken",
        "httptimeout":"30000",
        "checkcert":"no"
    }
```

Name	Description	Type	Example
type	Indicates the data transfer type with OI. Values: HTTP or HTTPS.	String	http
hostname	Indicates the hostname or IP address of the REST Endpoint.	String	test.example.net

Name	Description	Type	Example
port	Indicates the port number of the REST Endpoint.	Integer	9600
authentication	Enter the authentication type as urltoken.	String	urltoken
token	Enter the access token or the bearer token when the authentication type is OAuth2 or bearer respectively.	String	
httptimeout	Indicates the value of the timeout that is expressed in milliseconds.	Integer	300
checkcert	Indicates to verify the certificate to ensure it is valid and trusted.	Boolean	no

monitored_groups

In the **monitored_groups** section, specify the groups that you want to monitor. The following snippet is a sample of the **monitored_groups** section:

```
{
        "Alarms":"yes",
        "VolumeMetrics":"yes",
        "ArrayMetrics":"yes",
        "Topology":"yes"
    }
```

attribute_filter

You can configure RESTMon to ingest and display only the required information on the DX Operational Intelligence UI using the attribute filter. To filter the ingested data, add the attribute filter to the profile.json file. For more information, see the Filter Entities and Related Data Before Ingestion section.

> **NOTE**
> For the detailed steps, see the Add the Profile section.

Upload the Pure Storage Schema

The schema is automatically uploaded when you add the profile. Perform the steps only if you want to upload the updated or edited schema.

> **NOTE**
> For the detailed steps, see the Upload the Schema section.

View Data in DX OI

You can view the ingested data in the Alarm Analytics, Service Analytics, and Performance Analytics pages in DX OI for insights into service, raw, and anomaly alarms.

> **NOTE**
> For more information, see the View Data in DX OI section.

ScienceLogic

DX Operational Intelligence supports integration with ScienceLogic.

The ScienceLogic schema is a Polling schema that enables you to retrieve the events data from ScienceLogic and ingest into DX Operational Intelligence (DX OI) as alarms.

You can ingest the following monitoring data:

- Topology
- Alarms

This section provides the following information:

Supported Versions

The ScienceLogic - DX OI integration is supported for the following version:

Product	Supported Version
ScienceLogic	2019

Configure the Integration

The ScienceLogic - DX OI integration involves the following steps:

- Configure the ScienceLogic Environment
- Configure RESTMon

Configure the ScienceLogic Environment

No specific integration steps are required to be performed in the ScienceLogic environment. However, ensure that the following requirements are met:

- You have access to the ScienceLogic instance.
- This integration can communicate over the port 443. ScienceLogic uses HTTPS on port 443 (by default) with the Basic authentication.

> **NOTE**
> For more information, see the ScienceLogic documentation.

Configure RESTMon

To configure RESTMon, update the OI connection details if not done earlier, and add the profile information to the **restmon.json** file. Before you configure RESTMon, ensure that the following requirements are met:

- You have access to DX OI.
- RESTMon is configured and deployed successfully. For more information, see the Configure and Deploy RESTMon section.
- RESTMon is configured and deployed successfully.

Add the Profile

To add the profile, configure the profile to connect to your ScienceLogic environment and add the profile to the **restmon.json** file using the POST Profile REST API call in Swagger. The **sciencelogic_profile.json** file is available in the **<restmon\profile>** folder. When you add the profile, the schema for ScienceLogic is automatically uploaded, and the data ingestion starts.

> **NOTE**
> You can also add the attribute filter to the profile to filter the entities and the related data. For more information, see the **attribute_filter** section on this page.

You can also add this information directly in the **restmon.json** file.

The ScienceLogic profile includes the following sections:

Profile

The **profile** section defines the profile-related information. The following snippet is a sample of the profile section.

```
{
      "name" : "sciencelogic",
      "active" : "yes",
      "schema" : "sciencelogic",
      "streaming" : "no",
      "polling_interval_secs" : "300",
      "inventory_topology_fullsync_interval_mins" : "1440",
      "topology_ttl_mins" : "2880"
    }
```

Name	Description	Type	Example
name	Indicates the name of the profile.	String	sciencelogic
schema	Indicates the name of the schema. The name that you specify for the schema must be the same as the schema attribute specified in the restmon.json file.	String	sciencelogic
streaming	Indicates whether streaming is enabled. If enabled, the integrating product posts data to RESTMon as JSON (Webhook). If not, RESTMon gets (polling) data from the integrating product.	Boolean	no
polling_interval_secs	Indicates the polling interval in seconds. **Supported Values:** 15, 30, 60, 300, 900, 1800, 3600, and 7200 Any other value is rounded off to the nearest intervals.	Integer	300
inventory_topology_fullsync_interval_mins	Indicates the full synchronization interval in minutes.	Integer	1440
topology_ttl_mins	Indicates that the time-to-live (TTL) record is cached in minutes.	Integer	2880
active	Indicates whether the data processing is enabled. Specify *yes* to enable the profile.	Boolean	yes

restapiconnectdetails

In the **restapiconnectdetails** section, enter the REST Endpoint details of your ScienceLogic environment. The following snippet is a sample of the **restapiconnectdetails** section:

```
{
```

```
    "type": "https",
    "hostname": "sampleHostname",
    "port": "",
    "authentication": "basic",
    "username": "sampleUser",
    "password": "samplePassword",
    "realmdomain": "",
    "token": "",
    "httptimeout": "30000",
    "checkcert": "no"
  }
```

Name	Description	Type	Example
type	Indicates the data transfer type with DX OI. Values: HTTP or HTTPS.	String	https
hostname	Indicates the host name or IP address of the REST endpoint.	String	sampleHostName
port	Indicates the port number of the REST endpoint.	Integer	443
authentication	Indicates the authentication type. For ScienceLogic, specify the authentication type as basic. Additionally, the following authentication types are available: • **none:** No authorization required. • **basic:** Enter the username and password. • **NTLM:** Enter the username and password. • **digest:** Enter the username, password, and the realmdomain. • **OAuth2:** Enter the access token in the token parameter. • **bearer:** Enter the bearer token in the token parameter. • **urltoken:** Enter the token in the token parameter.	String	basic
username	Indicates the username. Applies only when the authentication type is basic or NTLM.	String	sampleUserName
password	Indicates the password. Applies only when the authentication type is basic or NTLM.	String	samplePassword
realmdomain	Indicates the DNS realm or domain to encode in the token. Applies only when the authentication type is digest.	String	

Name	Description	Type	Example
token	Indicates the access token or bearer token when the authentication type is OAuth2 or bearer respectively.	String	sampleToken
httptimeout	Indicates the value of the timeout that is expressed in milliseconds.	Integer	60000
checkcert	Indicates to verify the certificate to ensure it is valid and trusted.	Boolean	no

monitored_groups

In the **monitored_groups** section, specify the groups that you want to monitor. The following snippet is a sample of the **monitored_groups** section:

```
{
      "Events" : "yes",
      "Topology" : "yes"
    }
```

attribute_filter

You can configure RESTMon to ingest and display only the required information on the DX Operational Intelligence UI using the attribute filter. To filter the ingested data, add the attribute filter to the profile.json file. For more information, see the Filter Entities and Related Data Before Ingestion section.

> **NOTE**
> For the detailed steps, see the Add the Profile section.

Upload the ScienceLogic Schema

The schema is automatically uploaded when you add the profile. Perform the steps only if you want to upload the updated or edited schema.

> **NOTE**
> For the detailed steps, see the Upload the Schema section.

View Data in DX OI

You can view the ingested data in the Alarm Analytics, Service Analytics, and Performance Analytics pages in DX OI for insights into service, raw, and anomaly alarms.

> **NOTE**
> For more information, see the View Data in DX OI section.

ServiceNow

DX Operational Intelligence supports integration with ServiceNow.

The ServiceNow schema enables you to import Change Requests, Incidents, and Inventory into DX Operational Intelligence. The schema enables you to ingest the following monitoring data:

- **Inventory:** The CMDB tables in ServiceNow are ingested into DX Operational Intelligence as inventory.
- **Topology:** The CMDB tables in ServiceNow are ingested into DX Operational Intelligence as topology (without relationship).
- **Alarms:** The ServiceNow incidents and change requests are ingested into DX Operational Intelligence as alarms. The ServiceNow changes are ingested into DX OI as alarms or change events (based on your configurations in the

schema). Incidents and change requests that are ingested as alarms have a URL that links back to the ServiceNow incident or change request in the ServiceNow instance.

This section provides the following information:

Configure the Integration

The ServiceNow - DX OI integration involves the following steps:

- Configure the ServiceNow Environment
- Configure RESTMon

Configure the ServiceNow Environment

No integration-specific steps are required to be performed in the ServiceNow environment. However, ensure that the following requirements are met:

- Ensure that you have the ServiceNow instance URL.
- Ensure that the integration can communicate over port 443.

> **NOTE**
> For more information, see the ServiceNow documentation.

Configure RESTMon

To configure RESTMon, update the OI connection details if not done earlier, and add the profile information to the **restmon.json** file. Before you configure RESTMon, ensure that the following requirements are met:

- You have access to DX OI.
- RESTMon is configured and deployed successfully. For more information, see the Configure and Deploy RESTMon section.
- RESTMon is configured and deployed successfully.
- For Java 11, ensure that you replace the *contains* method with *includes* in the schema.

Add the Profile

To add the profile, configure the profile to connect to your ServiceNow environment and add the profile to the **restmon.json** file using the POST Profile REST API call in Swagger. The **snow_profile.json** file is available in the **<restmon\profile>** folder. When you add the profile, the schema for ServiceNow is automatically uploaded and the data ingestion starts.

> **NOTE**
> You can also add the attribute filter to the profile to filter the entities and the related data. For more information, see the **attribute_filter** section on this page.

You can also add this information directly in the **restmon.json** file.

The ServiceNow profile includes the following sections:

Profile

The **profile** section defines the profile-related information. The following snippet is a sample of the profile section.

```
"{
        "name":"snow",
        "schema":"snow",
        "streaming":"no",
        "polling_interval_secs":"300",
        "inventory_topology_fullsync_interval_mins":"1440",
```

```
        "topology_ttl_mins":"2880",
         "active":"yes"
    }
```

Name	Description	Type	Example
name	Indicates the of the profile.	String	servicenow
schema	Indicates name of the schema. The name that you specify for the schema should be the same as the schema attribute specified in the restmon.json file.	String	servicenow
polling_interval_secs	Indicates the polling interval in seconds. **Supported Values:** 15, 30, 60, 300, 900, 1800, 3600, and 7200 Any other value is rounded off to the nearest intervals.	Integer	300
inventory_topology_fullsync_interval_mins	Indicates the full synchronization interval in minutes.	Integer	1440
topology_ttl_mins	Indicates the time-to-live (TTL) record is cached in minutes.	Integer	2880
tenantname	Indicates the DX Operational Intelligence tenant name.	String	sampleTenantName
active	Indicates whether the data processing is enabled. Enter *yes* to enable the profile.	Boolean	yes

restapiconnectdetails

In the **restapiconnectdetails** section, enter the REST Endpoint details of your ServiceNow environment. The following snippet is a sample of the **restapiconnectdetails** section:

```
{
        "type":"https",
        "hostname":"sampleHostName",
        "port":"",
        "authentication":"basic",
        "username":"sampleUser",
        "password":"samplePassword",
        "realmdomain":"",
        "token":"",
        "httptimeout":"30000",
        "checkcert":"no"
    }
```

Name	Description	Type	Example
type	Indicates the data transfer type with OI. Values: HTTP or HTTPS.	String	https
hostname	Indicates the host name or IP address of the REST endpoint.	String	test.example.net

Name	Description	Type	Example
port	Indicates the port number of the REST endpoint.	Integer	443
authentication	Indicates the authentication type. For ServiceNow, enter the authentication type as basic. Additionally, the following authentication types are available: • **none:** No authorization required. • **basic:** Enter the username and password. • **NTLM:** Enter the username and password. • **digest:** Enter the username, password, and the realmdomain. • **OAuth2:** Enter the access token in the token parameter. • **bearer:** Enter the bearer token in the token parameter. • **urltoken:** Enter the token in the token parameter.	String	basic
username	Indicates the username. Applies only when the authentication type is basic or NTLM.	String	sampleUserName
password	Indicates the password. Applies only when the authentication type is basic or NTLM.	String	samplePassword
realmdomain	Indicates the DNS realm or domain to encode in the token. Applies only when the authentication type is digest.	String	
token	Indicates the access token or bearer token when the authentication type is OAuth2 or bearer respectively.	String	sampleToken
httptimeout	Indicates the value of the timeout that is expressed in milliseconds.	Integer	60000
checkcert	Indicates to verify the certificate to ensure it is valid and trusted.	Boolean	no

monitored_groups

In the **monitored_groups** section, specify the groups that you want to monitor. The following snippet is a sample of the **monitored_groups** section:

```
{
        "Alarms":"yes",
        "Topology":"yes",
```

```
        "ChangeEvents":"yes"
    }
```

attribute_filter

You can configure RESTMon to ingest and display only the required information on the DX Operational Intelligence UI using the attribute filter. To filter the ingested data, add the attribute filter to the profile.json file. For more information, see the Filter Entities and Related Data Before Ingestion section.

> **NOTE**
> For the detailed steps, see the Add the Profile section.

Upload the ServiceNow Schema

The schema is automatically uploaded when you add the profile. Perform the steps only if you want to upload the updated or edited schema.

> **NOTE**
> For the detailed steps, see the Upload the Schema section.

View Data in DX OI

You can view the ingested data in the Alarm Analytics, Service Analytics, and Performance Analytics pages in DX OI for insights into service, raw, and anomaly alarms.

> **NOTE**
> For more information, see the View Data in DX OI section.

SolarWinds

DX Operational Intelligence supports integration with SolarWinds.

The SolarWinds schema is a Polling schema that enables you to periodically poll the SolarWinds Orion systems for alerts and send them to DX Operational Intelligence. This schema polls SolarWinds at regular intervals to collect the event data (every 300 seconds by default).

This section provides the following information:

Video: Getting Started with RESTMon

The following video explains how to work with RESTMon by ingesting SolarWinds data.

Configure the Integration

The SolarWinds - DX OI integration involves the following steps:

- Configure the SolarWinds Environment
- Configure RESTMon

Configure the SolarWinds Environment

No additional integration-specific steps are required to be performed in the SolarWinds environment. Before you configure the environment, ensure that the following requirements are met:

- You have a local SolarWinds Orion user account with Administrator access.
- The Orion SDK is installed on your SolarWinds installation.
- You have the connection details for your SolarWinds Orion platform (hostname or IP address, username, and password).
- You have opened a port for SolarWinds to receive connections from DX SaaS. The default port is 17778.

> **NOTE**
> For more information, see the SolarWinds Orion documentation.

Configure RESTMon

To configure RESTMon, update the OI connection details if not done earlier, and add the profile information to the **restmon.json** file. Before you configure RESTMon, ensure that the following requirements are met:

- You have access to DX OI.
- RESTMon is configured and deployed successfully. For more information, see the Configure and Deploy RESTMon section.
- RESTMon is configured and deployed successfully.
- For Java 11, ensure that you replace the *contains* method with *includes* in the schema.

Add the Profile

To add the profile, configure the profile to connect to your SolarWinds environment and add the profile to the **restmon.json** file using the POST Profile REST API call in Swagger. The **solarwinds_profile.json** file is available in the **<restmon\profile>** folder. When you add the profile, the schema for SolarWinds is automatically uploaded, and the data ingestion starts.

> **NOTE**
> You can also add the attribute filter to the profile to filter the entities and the related data. For more information, see the **attribute_filter** section on this page.

You can also add this information directly in the **restmon.json** file.

The SolarWinds profile includes the following sections:

Profile

The **profile** section defines the profile-related information. The following snippet is a sample of the profile section.

```
{
        "name": "solarwinds",
        "active": "yes",
        "schema": "solarwinds",
        "polling_interval_secs": "300",
        "inventory_topology_fullsync_interval_mins": "1440",
        "topology_ttl_mins": "2880"
    }
```

Name	Description	Type	Example
name	Indicates the name of the profile.	String	solarwinds
active	Indicates if the data-processing is enabled. Specify *yes* to enable the profile.	Boolean	yes

Name	Description	Type	Example
schema	Name of the schema. The name that you specify for the schema should be the same as the schema attribute specified in the restmon.json file.	String	solarwinds
polling_interval_secs	Indicates the polling interval in seconds. **Supported Values:** 15, 30, 60, 300, 900, 1800, 3600, and 7200 Any other value is rounded off to the nearest intervals.	Integer	300
inventory_topology_fullsync_interval_mins	Indicates the full synchronization interval in minutes.	Integer	1440
topology_ttl_min	Indicates the time-to-live (TTL) a record is cached in minutes.	Integer	2880
tenantname	Indicates the DX Operational Intelligence tenant name.	String	SampleTenantName

restapiconnectdetails

In the **restapiconnectdetails** section, enter the REST Endpoint details of your SolarWinds environment. The following snippet is a sample of the **restapiconnectdetails** section:

```
{
        "type": "https",
        "hostname": "sampleHostName",
        "port": "SamplePort",
        "authentication": "basic",
        "username": "SampleUserName",
        "password": "SamplePassword",
        "realmdomain": "",
        "token": "",
        "httptimeout": "60",
        "checkcert": "no"
    }
```

Name	Description	Type	Example
type	Indicates the data transfer type with DX OI. Values: HTTP or HTTPS.	String	https
hostname	Indicates the hostname or IP address of the REST Endpoint.	String	https://solarwinds.example.com
port	Indicates the port number of the REST Endpoint.	Integer	17778

Name	Description	Type	Example
authentication	Indicates the authentication type. For the SolarWinds integration, you can set basic. You can also set any of the following authentication types: • **none:** No authorization is required. • **basic:** Enter the username and password. • **NTLM:** Enter the username and password. • **digest:** Enter the username, password, and the realmdomain. • **OAuth2:** Enter the access token in the token parameter. • **bearer:** Enter the bearer token in the token parameter. • **urltoken:** Enter the token in the token parameter.	String	basic
username	Indicates the username. Applies only when the authentication type is basic or NTLM.	String	
password	Indicates the password. Applies only when the authentication type is basic or NTLM.	String	
realmdomain	Indicates the DNS realm or domain to encode in the token. Applies only when the authentication type is digest.	String	
token	Indicates the access token or bearer token when the authentication type is OAuth2 or bearer, respectively.	String	
httptimeout	Indicates the value of the timeout that is expressed in milliseconds.	Integer	60
checkcert	Indicates to verify the certificate to ensure it is valid and trusted.	Boolean	no

monitored_groups

In the **monitored_groups** section, specify the groups that you want to monitor. The following snippet is a sample of the **monitored_groups** section:

```
{
    "topology": "yes",
    "alarms": "yes",
    "metrics": "yes"
  }
```

attribute_filter

You can configure RESTMon to ingest and display only the required information on the DX Operational Intelligence UI using the attribute filter. To filter the ingested data, add the attribute filter to the profile.json file. For more information, see the Filter Entities and Related Data Before Ingestion section.

> **NOTE**
> For the detailed steps, see the Add the Profile section.

Upload the SolarWinds Schema

The schema is automatically uploaded when you add the profile. Perform the steps only if you want to upload the updated or edited schema.

> **NOTE**
> For the detailed steps, see the Upload the Schema section.

View Data in DX OI

You can view the ingested data in the Alarm Analytics, Service Analytics, and Performance Analytics pages in DX OI for insights into service, raw, and anomaly alarms.

> **NOTE**
> For more information, see the View Data in DX OI section.

SolarWinds AppOptics

DX Operational Intelligence supports integration with SolarWinds AppOptics.

The SolarWinds AppOptics schema is a Polling schema that enables you to retrieve infrastructure and application monitoring data from SolarWinds AppOptics and send it to DX Operational Intelligence (DX OI). This schema polls your SolarWinds AppOptics server at regular intervals to collect data.

The SolarWinds AppOptics schema enables you to ingest the following monitoring data:

- **Inventory**
- **Topology**
- **Alarms**
- **Metrics**

This section provides the following information:

Configure the Integration

The SolarWinds AppOptics - DX OI integration involves the following steps:

- Configure the SolarWinds AppOptics Environment
- Configure RESTMon

Configure the SolarWinds AppOptics Environment

No integration-specific steps are required to be performed in your SolarWinds AppOptics environment. However, ensure that the following requirements are met:

- You can connect to the SolarWinds AppOptics instance.
- You have downloaded the appropriate application monitoring and host/infrastructure monitoring agents.

> **NOTE**
> For more information, see the SolarWinds AppOptics documentation.

Configure RESTMon

To configure RESTMon, update the OI connection details if not done earlier, and add the profile information to the **restmon.json** file. Before you configure RESTMon, ensure that the following requirements are met:

- You have access to DX OI.
- RESTMon is configured and deployed successfully. For more information, see the Configure and Deploy RESTMon section.
- RESTMon is configured and deployed successfully.
- For Java 11, replace the *contains* method with *includes* in the schema.

Add the Profile

To add the profile, configure the profile to connect to your SolarWinds AppOptics environment and add the profile to the **restmon.json** file using the POST Profile REST API call in Swagger. The **appoptics_profile.json** file is available in the **<RESTmon\profile>** folder. When you add the profile, the schema for SolarWinds AppOptics is automatically uploaded, and the data ingestion starts.

> **NOTE**
> You can also add the attribute filter to the profile to filter the entities and the related data. For more information, see the **attribute_filter** section on this page.

You can also add this information directly in the **restmon.json** file.

The SolarWinds AppOptics profile includes the following sections:

Profile

The **profile** section defines the profile-related information. The following snippet is a sample of the profile section.

```
{
        "name" : "appoptics",
        "active" : "yes",
        "schema" : "appoptics",
        "polling_interval_secs" : "300",
        "inventory_topology_fullsync_interval_mins" : "1440",
        "topology_ttl_mins" : "2880"
      }
```

Name	Description	Type	Example
name	Indicates the name of the profile.	String	appoptics
schema	Indicates the name of the schema. The name that you specify for the schema should be the same as the schema attribute specified in the restmon.json file.	String	appoptics
polling_interval_secs	Indicates the polling interval in seconds. **Supported Values:** 15, 30, 60, 300, 900, 1800, 3600, and 7200 Any other value is rounded off to the nearest intervals.	Integer	300
inventory_topology_fullsync_interval_mins	Indicates the full synchronization interval in minutes.	Integer	1440

Name	Description	Type	Example
topology_ttl_mins	Indicates the time-to-live (TTL) record is cached in minutes.	Integer	2880
active	Indicates whether the data processing is enabled. Specify *yes* to enable the profile.	Boolean	yes

servicedefinition

In the **servicedefinition** section, define the service that appears when data is ingested to DX OI. The following snippet is a sample of the **servicedefinition** section:

```
{
        "name" : "",
        "status" : ""
      },
```

Name	Description	Type	Example
name	Indicates the name of the service.	String	AppOptics
status	Indicates the status of the defined service.	String	Active

restapiconnectdetails

In the **restapiconnectdetails** section, enter the REST Endpoint details of your AppOptics environment. The following snippet is a sample of the **restapiconnectdetails** section:

```
{
        "type" : "https",
        "hostname" : "my.appoptics.com",
        "port" : "",
        "authentication" : "basic",
        "username" : "q6ySRuMpw_YcrMk_aYOBAl832JK7AduKyKkSHPUUZFvB3eScEyQdwoadQDvQ3g85K9UthgM",
        "password" : "",
        "realmdomain" : "",
        "token" : "",
        "httptimeout" : "30000",
        "checkcert" : "no"
      },
```

Name	Description	Type	Example
type	Indicates the data transfer type with OI. Values: HTTP or HTTPS.	String	https
hostname	Indicates the hostname or IP address of the REST endpoint.	String	my.appoptics.com
port	Indicates the port number of the REST endpoint.	Integer	

Name	Description	Type	Example
authentication	Indicates the authentication type. For the SolarWinds AppOptics integration, you can set **basic**. Additionally, the following authentication types are available: • **none:** No authorization is required. • **basic:** Enter the username and password. • **NTLM:** Enter the username and password. • **digest:** Enter the username, password, and the realmdomain. • **OAuth2:** Enter the access token in the token parameter. • **bearer:** Enter the bearer token in the token parameter. • **urltoken:** Enter the token in the token parameter.	String	basic
username	Indicates the username. Applies only when the authentication type is basic or NTLM.	String	sampleUserName
password	Indicates the password. Applies only when the authentication type is basic or NTLM.	String	samplePassword
realmdomain	Indicates the DNS realm or domain to encode in the token. Applies only when the authentication type is digest.	String	
token	Indicates the access token or bearer token when the authentication type is OAuth2 or bearer, respectively.	String	sampleToken
httptimeout	Indicates the value of the timeout that is expressed in milliseconds.	Integer	30000
checkcert	Indicates to verify the certificate to ensure it is valid and trusted.	Boolean	no

monitored_groups

In the **monitored_groups** section, specify the groups that you want to monitor. The following snippet is a sample of the **monitored_groups** section:

```
{
        "hosts" : "yes",
        "disks" : "yes",
        "process" : "no",
        "services" : "yes",
```

```
        "transaction" : "yes",
        "CpuLoad" : "yes",
        "DiskLoad" : "yes",
        "MemLoad" : "yes",
        "TransactionLoad" : "yes",
        "ApmLoad" : "yes",
        "IoLoad" : "yes",
        "SystemLoad" : "yes",
        "NicLoad" : "yes"
    }
```

attribute_filter

You can configure RESTMon to ingest and display only the required information on the DX Operational Intelligence UI using the attribute filter. To filter the ingested data, add the attribute filter to the profile.json file. For more information, see the Filter Entities and Related Data Before Ingestion section.

> **NOTE**
> For the detailed steps, see the Add the Profile section.

Upload the AppOptics Schema

The schema is automatically uploaded when you add the profile. Perform the steps only if you want to upload the updated or edited schema.

> **NOTE**
> For the detailed steps, see the Upload the Schema section.

View Data in DX OI

You can view the ingested data in the Alarm Analytics, Service Analytics, and Performance Analytics pages in DX OI for insights into service, raw, and anomaly alarms.

> **NOTE**
> For more information, see the View Data in DX OI section.

Splunk

DX Operational Intelligence supports integration with Splunk.

The Splunk schema enables you to send results from the Splunk search pipeline as alarms and metrics to DX Operational Intelligence (DX OI). This schema polls Splunk at regular intervals to collect event data (every 300 seconds by default).

The Splunk schema enables you to ingest the following monitoring data:

- Inventory
- Metrics
- Alarms
- Topology

This section provides the following information:

Supported Versions

The Splunk - DX OI integration is supported for the following version:

Product	Supported Version
Splunk	7.2

Configure the Integration

The Splunk - DX OI integration involves the following steps:

- Configure the Splunk Environment
- Configure RESTMon

Configure the Splunk Environment

No integration-specific steps are required to be performed in your Splunk environment. However, ensure that the following requirements are met:

- You have an active Splunk account.
- Splunk can make requests to external endpoints over port 443.

> **NOTE**
> For more information, see the Splunk documentation.

Configure RESTMon

To configure RESTMon, update the OI connection details if not done earlier, and add the profile information to the **restmon.json** file. Before you configure RESTMon, ensure that the following requirements are met:

- You have access to DX OI.
- RESTMon is configured and deployed successfully. For more information, see the Configure and Deploy RESTMon section.
- RESTMon is configured and deployed successfully.

Add the Profile

To add the profile, configure the profile to connect to your Splunk environment and add the profile to the **restmon.json** file using the POST Profile REST API call in Swagger. The **splunk_profile.json** file is available in the **<restmon\profile>** folder. When you add the profile, the schema for Splunk is automatically uploaded, and the data ingestion starts.

> **NOTE**
> You can also add the attribute filter to the profile to filter the entities and the related data. For more information, see the **attribute_filter** section on this page.

You can also add this information directly in the **restmon.json** file.

The Splunk profile includes the following sections:

Profile

The **profile** section defines the profile-related information. The following snippet is a sample of the profile section.

```
{
        "name" : "splunk",
        "active" : "yes",
        "schema" : "splunk",
        "polling_interval_secs" : "300",
```

```
        "inventory_topology_fullsync_interval_mins" : "1440",
        "topology_ttl_mins" : "2880",
        "httpReqRetryCount":"5",
        "httpReqRetryInterval" : "1"
    }
```

Name	Description	Type	Example
name	Indicates the name of the profile.	String	Splunk
active	Indicates if the data-processing is enabled. Enter *yes* to enable the profile.	Boolean	yes
schema	Indicates the name of the schema. The name that you specify for the schema should be the same as the schema attribute specified in the restmon.json file.	String	Splunk
polling_interval_secs	Indicates the polling interval in seconds.	Integer	300
inventory_topology_fullsync_interval_mins	Indicates the full synchronization interval in minutes.	Integer	1440
topology_ttl_min	Indicates the time-to-live (TTL) a record is cached in minutes.	Integer	2880
tenantname	Indicates the DX Operational Intelligence tenant name.	String	TenantName
httpReqRetryCount	Indicates the number of retries for the HTTP connection.	Integer	5
httpReqRetryInterval	Indicates the interval between HTTP connection retries. If set as 0, no retry is done between the HTTP connection retries.	Integer	1

restapiconnectdetails

In the **restapiconnectdetails** section, enter the REST Endpoint details of your Splunk environment. The following snippet is a sample of the **restapiconnectdetails** section:

```
{
        "type" : "https",
        "hostname" : "",
        "port" : "8089",
        "authentication" : "Basic",
        "username" : "",
        "password" : "",
        "realmdomain" : "",
        "token" : "",
        "httptimeout" : "30000",
        "checkcert" : "no"
```

```
}
```

Name	Description	Type	Example
type	Indicates the data transfer type with DX OI. Values: HTTP or HTTPS.	String	https
hostname	Indicates the hostname or IP address of the REST Endpoint.	String	https://splunk.example.com
port	Indicates the port number of the REST Endpoint.	Integer	8089
authentication	Indicates the authentication type. For the Splunk integration, you can set **basic**. You can also set any of the following authentication types: • **none:** No authorization is required. • **basic:** Enter the username and password. • **NTLM:** Enter the username and password. • **digest:** Enter the username, password, and the realmdomain. • **OAuth2:** Enter the access token in the token parameter. • **bearer:** Enter the bearer token in the token parameter. • **urltoken:** Enter the token in the token parameter.	String	basic
username	Indicates the username. Applies only when the authentication type is basic or NTLM.	String	
password	Indicates the password. Applies only when the authentication type is basic or NTLM.	String	
realmdomain	Indicates the DNS realm or domain to encode in the token. Applies only when the authentication type is digest.	String	
token	Indicates the access token or bearer token when the authentication type is OAuth2 or bearer, respectively.	String	
httptimeout	Indicates the value of the timeout that is expressed in milliseconds.	Integer	60
checkcert	Indicates to verify the certificate to ensure it is valid and trusted.	Boolean	no

monitored_groups

In the **monitored_groups** section, specify the groups that you want to monitor. The following snippet is a sample of the **monitored_groups** section:

```
{
        "Search" : "yes",
        "Search_Inventory" : "no"
      }
```

attribute_filter

You can configure RESTMon to ingest and display only the required information on the DX Operational Intelligence UI using the attribute filter. To filter the ingested data, add the attribute filter to the profile.json file. For more information, see the Filter Entities and Related Data Before Ingestion section.

> **NOTE**
> For the detailed steps, see the Add the Profile section.

Upload the Splunk Schema

The schema is automatically uploaded when you add the profile. Perform the steps only if you want to upload the updated or edited schema.

> **NOTE**
> For the detailed steps, see the Upload the Schema section.

View Data in DX OI

You can view the ingested data in the Alarm Analytics, Service Analytics, and Performance Analytics pages in DX OI for insights into service, raw, and anomaly alarms.

> **NOTE**
> For more information, see the View Data in DX OI section.

Splunk Example: Create Counter Metric and Custom Alarms

Using the Splunk schema, you can create counter metrics for a given pattern and can generate custom alarms that are ingested in DX Operational Intelligence.

This section provides the following information:

Create Counter Metric

The Splunk - DX OI integration uses the search/jobs endpoint to create a search job in a Splunk deployment. Once the log events are captured through the search result, they are iterated and a counter metric is generated against a given pattern.

For each search, a JSON document with an array of log events is captured as the following snippet illustrates.

Click to expand the log events snippet...

```
{
   "preview":false,
   "init_offset":0,
   "messages":[

   ],
   "fields":[
      {
```

```
        "name":"_bkt"
    },
    {
        "name":"_cd"
    },
    {
        "name":"_indextime"
    },
    {
        "name":"_raw"
    },
    {
        "name":"_serial"
    },
    {
        "name":"_si"
    },
    {
        "name":"_sourcetype"
    },
    {
        "name":"_time"
    },
    {
        "name":"host"
    },
    {
        "name":"index"
    },
    {
        "name":"linecount"
    },
    {
        "name":"logevents"
    },
    {
        "name":"message"
    },
    {
        "name":"source"
    },
    {
        "name":"sourcetype"
    },
    {
        "name":"splunk_server"
    }
],
"results":[
    {
        "_bkt":"main~3111~B3CA3274-73FC-47B5-B603-1E27BEC8DE3A",
        "_cd":"3111:53203393",
        "_indextime":"1586419929",
```

"_raw":"{'logData': {'messageType': 'DATA_MESSAGE', 'owner': '524080397461', 'logGroup': '/aws/elasticbeanstalk/SYMC-WSSE-1AI88EJ8S5QR3/var/log/nginx/error.log', 'logStream': 'i-008c47bc8727f41a4', 'subscriptionFilters': ['CWToSplunkLambda'], 'logEvents': [{'id': '35378346461070915104548027929990601245200404145890000896', 'timestamp': 1586419922120, 'message': '2020/04/09 08:12:01 [error] 24368#24368: *12602332 \"/etc/nginx/html/index.html\" is not found (2: No such file or directory), client: 172.168.2.170, server: pfms-stage.wss.symantec.com, request: \"GET / HTTP/1.0\"', 'extractedFields': {'11': 'No', '12': 'such', '13': 'file', '14': 'or', '15': 'directory),', '16': 'client:', '17': '172.168.2.170,', '18': 'server:', '19': 'pfms-stage.wss.symantec.com,', '1': '2020/04/09', '2': '08:12:01', '3': 'error', '4': '24368#24368:', '5': '*12602332', '6': '/etc/nginx/html/index.html', '7': 'is', '8': 'not', '9': 'found', '20': 'request:', '10': '(2:', '21': 'GET / HTTP/1.0'}},
{'id': 35378346483393961048277181694667855236121414013376397313', 'timestamp': 1586419923121, 'message': '2020/04/09 08:12:02 [error] 24368#24368: *12602333 \"/etc/nginx/html/index.html\" is not found (2: No such file or directory), client: 172.168.0.47, server: pfms-stage.wss.symantec.com, request: \"GET / HTTP/1.0\"', 'extractedFields': {'11': 'No', '12': 'such', '13': 'file', '14': 'or', '15': 'directory),', '16': 'client:', '17': '172.168.0.47,', '18': 'server:', '19': 'pfms-stage.wss.symantec.com,', '1': '2020/04/09', '2': '08:12:02', '3': 'error', '4': '24368#24368:', '5': '*12602333', '6': '/etc/nginx/html/index.html', '7': 'is', '8': 'not', '9': 'found', '20': 'request:', '10': '(2:', '21': 'GET / HTTP/1.0'}},
{'id': 35378346505694706246807804836203573508769775519356813314', 'timestamp': 1586419924121, 'message': '2020/04/09 08:12:03 [error] 24367#24367: *12602335 \"/etc/nginx/html/index.html\" is not found (2: No such file or directory), client: 172.168.0.47, server: pfms-stage.wss.symantec.com, request: \"GET / HTTP/1.0\"', 'extractedFields': {'11': 'No', '12': 'such', '13': 'file', '14': 'or', '15': 'directory),', '16': 'client:', '17': '172.168.0.47,', '18': 'server:', '19': 'pfms-stage.wss.symantec.com,', '1': '2020/04/09', '2': '08:12:03', '3': 'error', '4': '24367#24367:', '5': '*12602335', '6': '/etc/nginx/html/index.html', '7': 'is', '8': 'not', '9': 'found', '20': 'request:', '10': '(2:', '21': 'GET / HTTP/1.0'}},
{'id': 35378346550318497389067581742416545772339146892823625731', 'timestamp': 1586419926122, 'message': '2020/04/09 08:12:05 [error] 24368#24368: *12602347 \"/etc/nginx/html/index.html\" is not found (2: No such file or directory), client: 172.168.2.170, server: pfms-stage.wss.symantec.com, request: \"GET / HTTP/1.0\"', 'extractedFields': {'11': 'No', '12': 'such', '13': 'file', '14': 'or', '15': 'directory),', '16': 'client:', '17': '172.168.2.170,', '18': 'server:', '19': 'pfms-stage.wss.symantec.com,', '1': '2020/04/09', '2': '08:12:05', '3': 'error', '4': '24368#24368:', '5': '*12602347', '6': '/etc/nginx/html/index.html', '7': 'is', '8': 'not', '9': 'found', '20': 'request:', '10': '(2:', '21': 'GET / HTTP/1.0'}}]}, 'logGroupName': '/aws/elasticbeanstalk/SYMC-WSSE-1AI88EJ8S5QR3/var/log/nginx/error.log', 'logStreamName': 'i-008c47bc8727f41a4', 'logEvents': [{'id': 35378346461070915104548027929990601245200404145890000896', 'timestamp': 1586419922120, 'message': '2020/04/09 08:12:01 [error] 24368#24368: *12602332 \"/etc/nginx/html/index.html\" is not found (2: No such file or directory), client: 172.168.2.170, server: pfms-stage.wss.symantec.com, request: \"GET / HTTP/1.0\"', 'extractedFields': {'11': 'No', '12': 'such', '13': 'file', '14': 'or', '15': 'directory),', '16': 'client:', '17': '172.168.2.170,', '18': 'server:', '19': 'pfms-stage.wss.symantec.com,', '1': '2020/04/09', '2': '08:12:01', '3': 'error', '4': '24368#24368:', '5': '*12602332', '6': '/etc/nginx/html/index.html', '7': 'is', '8': 'not', '9': 'found', '20': 'request:', '10': '(2:', '21': 'GET / HTTP/1.0'}},
{'id': 35378346483393961048277181694667855236121414013376397313', 'timestamp': 1586419923121, 'message': '2020/04/09 08:12:02 [error] 24368#24368: *12602333 \"/etc/nginx/html/index.html\" is not found (2: No such file or directory), client: 172.168.0.47, server: pfms-stage.wss.symantec.com, request: \"GET / HTTP/1.0\"', 'extractedFields': {'11': 'No', '12': 'such', '13': 'file', '14': 'or', '15': 'directory),', '16': 'client:', '17': '172.168.0.47,', '18': 'server:', '19': 'pfms-stage.wss.symantec.com,', '1': '2020/04/09', '2': '08:12:02', '3': 'error', '4': '24368#24368:', '5': '*12602333', '6': '/etc/nginx/html/index.html', '7': 'is', '8': 'not', '9': 'found', '20': 'request:', '10': '(2:', '21': 'GET / HTTP/1.0'}},
{'id': 35378346505694706246807804836203573508769775519356813314', 'timestamp': 1586419924121, 'message': '2020/04/09 08:12:03 [error] 24367#24367: *12602335 \"/etc/nginx/html/index.html\" is not found (2: No such file or directory), client: 172.168.0.47, server: pfms-stage.wss.symantec.com, request: \"GET / HTTP/1.0\"', 'extractedFields': {'11': 'No', '12': 'such', '13': 'file', '14': 'or', '15': 'directory),', '16': 'client:', '17': '172.168.0.47,', '18': 'server:', '19': 'pfms-stage.wss.symantec.com,', '1': '2020/04/09', '2': '08:12:03', '3': 'error', '4': '24367#24367:', '5': '*12602335', '6': '/etc/nginx/html/index.html', '7': 'is', '8': 'not', '9': 'found', '20': 'request:', '10': '(2:', '21': 'GET / HTTP/1.0'}},

{'id': '35378346550318497389067581742416545772339146892823625731', 'timestamp': 1586419926122, 'message': '2020/04/09 08:12:05 [error] 24368#24368: *12602347 \"/etc/nginx/html/index.html\" is not found (2: No such file or directory), client: 172.168.2.170, server: pfms-stage.wss.symantec.com, request: \"GET / HTTP/1.0\"', 'extractedFields': {'11': 'No', '12': 'such', '13': 'file', '14': 'or', '15': 'directory),', '16': 'client:', '17': '172.168.2.170,', '18': 'server:', '19': 'pfms-stage.wss.symantec.com,', '1': '2020/04/09', '2': '08:12:05', '3': 'error', '4': '24368#24368:', '5': '*12602347', '6': '/etc/nginx/html/index.html', '7': 'is', '8': 'not', '9': 'found', '20': 'request:', '10': '(2:', '21': 'GET / HTTP/1.0'}}]}",
        "_serial":"0",
        "_si":[
           "idx-i-0f6ea5f2b68035f09.symcwss.splunkcloud.com",
           "main"
        ],
        "_sourcetype":"aws:lambda",
        "_time":"2020-04-09T08:12:09.000+00:00",
        "host":"http-inputs-symcwss.splunkcloud.com",
        "index":"main",
        "linecount":"1",
        "logevents":" [{'id': '35378346461070915104548027929990601245200404145890000896', 'timestamp': 1586419922120, 'message': '2020/04/09 08:12:01 [error] 24368#24368: *12602332 \"/etc/nginx/html/index.html\" is not found (2: No such file or directory), client: 172.168.2.170, server: pfms-stage.wss.symantec.com, request: \"GET / HTTP/1.0\"', 'extractedFields': {'11': 'No', '12': 'such', '13': 'file', '14': 'or', '15': 'directory),', '16': 'client:', '17': '172.168.2.170,', '18': 'server:', '19': 'pfms-stage.wss.symantec.com,', '1': '2020/04/09', '2': '08:12:01', '3': error', '4': '24368#24368:', '5': '*12602332', '6': '/etc/nginx/html/index.html', '7': 'is', '8': 'not', '9': 'found', '20': 'request:', '10': '(2:', '21': 'GET / HTTP/1.0'}}, {'id': '35378346483393961048277181694667855236121414013376397313', 'timestamp': 1586419923121, 'message': '2020/04/09 08:12:02 [error] 24368#24368: *12602333 \"/etc/nginx/html/index.html\" is not found (2: No such file or directory), client: 172.168.0.47, server: pfms-stage.wss.symantec.com, request: \"GET / HTTP/1.0\"', 'extractedFields': {'11': 'No', '12': 'such', '13': 'file', '14': 'or', '15': 'directory),', '16': 'client:', '17': '172.168.0.47,', '18': 'server:', '19': 'pfms-stage.wss.symantec.com,', '1': '2020/04/09', '2': '08:12:02', '3': 'error', '4': '24368#24368:', '5': '*12602333', '6': '/etc/nginx/html/index.html', '7': 'is', '8': 'not', '9': 'found', '20': 'request:', '10': '(2:', '21': 'GET / HTTP/1.0'}}, {'id': '35378346505694706246807804836203573508769775519356813314', 'timestamp': 1586419924121, 'message': '2020/04/09 08:12:03 [error] 24367#24367: *12602335 \"/etc/nginx/html/index.html\" is not found (2: No such file or directory), client: 172.168.0.47, server: pfms-stage.wss.symantec.com, request: \"GET / HTTP/1.0\"', 'extractedFields': {'11': 'No', '12': 'such', '13': 'file', '14': 'or', '15': 'directory),', '16': 'client:', '17': '172.168.0.47,', '18': 'server:', '19': 'pfms-stage.wss.symantec.com,', '1': '2020/04/09', '2': '08:12:03', '3': 'error', '4': '24367#24367:', '5': '*12602335', '6': '/etc/nginx/html/index.html', '7': 'is', '8': 'not', '9': 'found', '20': 'request:', '10': '(2:', '21': 'GET / HTTP/1.0'}}, {'id': '35378346550318497389067581742416545772339146892823625731', 'timestamp': 1586419926122, 'message': '2020/04/09 08:12:05 [error] 24368#24368: *12602347 \"/etc/nginx/html/index.html\" is not found (2: No such file or directory), client: 172.168.2.170, server: pfms-stage.wss.symantec.com, request: \"GET / HTTP/1.0\"', 'extractedFields': {'11': 'No', '12': 'such', '13': 'file', '14': 'or', '15': 'directory),', '16': 'client:', '17': '172.168.2.170,', '18': 'server:', '19': 'pfms-stage.wss.symantec.com,', '1': '2020/04/09', '2': '08:12:05', '3': 'error', '4': '24368#24368:', '5': '*12602347', '6': '/etc/nginx/html/index.html', '7': 'is', '8': 'not', '9': 'found', '20': 'request:', '10': '(2:', '21': 'GET / HTTP/1.0'}}]}, ",
        "message":[
           " '2020/04/09 08:12:01 [error] 24368#24368: *12602332 \"/etc/nginx/html/index.html\" is not found (2: No such file or directory), client: 172.168.2.170, server: pfms-stage.wss.symantec.com, request: \"GET / HTTP/1.0\"', ",
           " '2020/04/09 08:12:02 [error] 24368#24368: *12602333 \"/etc/nginx/html/index.html\" is not found (2: No such file or directory), client: 172.168.0.47, server: pfms-stage.wss.symantec.com, request: \"GET / HTTP/1.0\"', ",

```
            " '2020/04/09 08:12:03 [error] 24367#24367: *12602335 \"/etc/nginx/html/index.html\" is not found
 (2: No such file or directory), client: 172.168.0.47, server: pfms-stage.wss.symantec.com, request: \"GET /
 HTTP/1.0\"', ",
            " '2020/04/09 08:12:05 [error] 24368#24368: *12602347 \"/etc/nginx/html/index.html\" is not found
 (2: No such file or directory), client: 172.168.2.170, server: pfms-stage.wss.symantec.com, request: \"GET /
 HTTP/1.0\"', "
         ],
         "source":"http:LogsOverHttp",
         "sourcetype":"aws:lambda",
         "splunk_server":"idx-i-0f6ea5f2b68035f09.symcwss.splunkcloud.com"
      },
      {
         "_bkt":"main~3111~B3CA3274-73FC-47B5-B603-1E27BEC8DE3A",
         "_cd":"3111:53198835",
         "_indextime":"1586419903",
         "_raw":"{'logData': {'messageType': 'DATA_MESSAGE', 'owner': '524080397461', 'logGroup':
 '/aws/elasticbeanstalk/SYMC-WSSE-1AI88EJ8S5QR3/var/log/nginx/error.log', 'logStream':
 'i-008c47bc8727f41a4', 'subscriptionFilters': ['CWToSplunkLambda'], 'logEvents': [{'id':
 '35378345881050833235965050611284475263208966662210846720', 'timestamp': 1586419896111, 'message':
 '2020/04/09 08:11:35 [error] 24367#24367: *12602277 \"/etc/nginx/html/index.html\" is not found (2: No
 such file or directory), client: 172.168.2.170, server: pfms-stage.wss.symantec.com, request: \"GET /
 HTTP/1.0\"', 'extractedFields': {'11': 'No', '12': 'such', '13': 'file', '14': 'or', '15': 'directory),',
 '16': 'client:', '17': '172.168.2.170,', '18': 'server:', '19': 'pfms-stage.wss.symantec.com,', '1':
 '2020/04/09', '2': '08:11:35', '3': 'error', '4': '24367#24367:', '5': '*12602277', '6': '/etc/nginx/html/
index.html', '7': 'is', '8': 'not', '9': 'found', '20': 'request:', '10': '(2:', '21': 'GET / HTTP/1.0'}},
 {'id': '35378345903373879179694204375961729254129976529697243137', 'timestamp': 1586419897112, 'message':
 '2020/04/09 08:11:36 [error] 24368#24368: *12602278 \"/etc/nginx/html/index.html\" is not found (2: No
 such file or directory), client: 172.168.0.47, server: pfms-stage.wss.symantec.com, request: \"GET /
 HTTP/1.0\"', 'extractedFields': {'11': 'No', '12': 'such', '13': 'file', '14': 'or', '15': 'directory),',
 '16': 'client:', '17': '172.168.0.47,', '18': 'server:', '19': 'pfms-stage.wss.symantec.com,', '1':
 '2020/04/09', '2': '08:11:36', '3': 'error', '4': '24368#24368:', '5': '*12602278', '6': '/etc/nginx/html/
index.html', '7': 'is', '8': 'not', '9': 'found', '20': 'request:', '10': '(2:', '21': 'GET / HTTP/1.0'}},
 {'id': '35378345925674624378224827517497447526778338035677659138', 'timestamp': 1586419898112, 'message':
 '2020/04/09 08:11:37 [error] 24367#24367: *12602280 \"/etc/nginx/html/index.html\" is not found (2: No
 such file or directory), client: 172.168.0.47, server: pfms-stage.wss.symantec.com, request: \"GET /
 HTTP/1.0\"', 'extractedFields': {'11': 'No', '12': 'such', '13': 'file', '14': 'or', '15': 'directory),',
 '16': 'client:', '17': '172.168.0.47,', '18': 'server:', '19': 'pfms-stage.wss.symantec.com,', '1':
 '2020/04/09', '2': '08:11:37', '3': 'error', '4': '24367#24367:', '5': '*12602280', '6': '/etc/nginx/html/
index.html', '7': 'is', '8': 'not', '9': 'found', '20': 'request:', '10': '(2:', '21': 'GET / HTTP/1.0'}},
 {'id': '35378345947975369576755450659033165799426699541658075139', 'timestamp': 1586419899112, 'message':
 '2020/04/09 08:11:39 [error] 24367#24367: *12602283 \"/etc/nginx/html/index.html\" is not found (2: No such
 file or directory), client: 172.168.2.170, server: pfms-stage.wss.symantec.com, request: \"GET / HTTP/1.0\"',
 'extractedFields': {'11': 'No', '12': 'such', '13': 'file', '14': 'or', '15': 'directory),', '16': 'client:',
 '17': '172.168.2.170,', '18': 'server:', '19': 'pfms-stage.wss.symantec.com,', '1': '2020/04/09', '2':
 '08:11:39', '3': 'error', '4': '24367#24367:', '5': '*12602283', '6': '/etc/nginx/html/index.html', '7':
 'is', '8': 'not', '9': 'found', '20': 'request:', '10': '(2:', '21': 'GET / HTTP/1.0'}}]}, 'logGroupName': '/
aws/elasticbeanstalk/SYMC-WSSE-1AI88EJ8S5QR3/var/log/nginx/error.log', 'logStreamName': 'i-008c47bc8727f41a4',
 'logEvents': [{'id': '35378345881050833235965050611284475263208966662210846720', 'timestamp': 1586419896111,
 'message': '2020/04/09 08:11:35 [error] 24367#24367: *12602277 \"/etc/nginx/html/index.html\" is not found
 (2: No such file or directory), client: 172.168.2.170, server: pfms-stage.wss.symantec.com, request: \"GET /
 HTTP/1.0\"', 'extractedFields': {'11': 'No', '12': 'such', '13': 'file', '14': 'or', '15': 'directory),',
 '16': 'client:', '17': '172.168.2.170,', '18': 'server:', '19': 'pfms-stage.wss.symantec.com,', '1':
 '2020/04/09', '2': '08:11:35', '3': 'error', '4': '24367#24367:', '5': '*12602277', '6': '/etc/nginx/html/
```

index.html', '7': 'is', '8': 'not', '9': 'found', '20': 'request:', '10': '(2:', '21': 'GET / HTTP/1.0'}},
 {'id': '35378345903373879179694204375961729254129976529697243137', 'timestamp': 1586419897112, 'message':
 '2020/04/09 08:11:36 [error] 24368#24368: *12602278 \"/etc/nginx/html/index.html\" is not found (2: No
 such file or directory), client: 172.168.0.47, server: pfms-stage.wss.symantec.com, request: \"GET /
 HTTP/1.0\"', 'extractedFields': {'11': 'No', '12': 'such', '13': 'file', '14': 'or', '15': 'directory),',
 '16': 'client:', '17': '172.168.0.47,', '18': 'server:', '19': 'pfms-stage.wss.symantec.com,', '1':
 '2020/04/09', '2': '08:11:36', '3': 'error', '4': '24368#24368:', '5': '*12602278', '6': '/etc/nginx/html/
index.html', '7': 'is', '8': 'not', '9': 'found', '20': 'request:', '10': '(2:', '21': 'GET / HTTP/1.0'}},
 {'id': 35378345925674624378224827517497447526778338035677659138', 'timestamp': 1586419898112, 'message':
 '2020/04/09 08:11:37 [error] 24367#24367: *12602280 \"/etc/nginx/html/index.html\" is not found (2: No
 such file or directory), client: 172.168.0.47, server: pfms-stage.wss.symantec.com, request: \"GET /
 HTTP/1.0\"', 'extractedFields': {'11': 'No', '12': 'such', '13': 'file', '14': 'or', '15': 'directory),',
 '16': 'client:', '17': '172.168.0.47,', '18': 'server:', '19': 'pfms-stage.wss.symantec.com,', '1':
 '2020/04/09', '2': '08:11:37', '3': 'error', '4': '24367#24367:', '5': '*12602280', '6': '/etc/nginx/html/
index.html', '7': 'is', '8': 'not', '9': 'found', '20': 'request:', '10': '(2:', '21': 'GET / HTTP/1.0'}},
 {'id': 35378345947975369576755450659033165799426699541658075139', 'timestamp': 1586419899112, 'message':
 '2020/04/09 08:11:39 [error] 24367#24367: *12602283 \"/etc/nginx/html/index.html\" is not found (2: No such
 file or directory), client: 172.168.2.170, server: pfms-stage.wss.symantec.com, request: \"GET / HTTP/1.0\"',
 'extractedFields': {'11': 'No', '12': 'such', '13': 'file', '14': 'or', '15': 'directory),', '16': 'client:',
 '17': '172.168.2.170,', '18': 'server:', '19': 'pfms-stage.wss.symantec.com,', '1': '2020/04/09', '2':
 '08:11:39', '3': 'error', '4': '24367#24367:', '5': '*12602283', '6': '/etc/nginx/html/index.html', '7':
 'is', '8': 'not', '9': 'found', '20': 'request:', '10': '(2:', '21': 'GET / HTTP/1.0'}}]}",
        "_serial":"1",
        "_si":[
           "idx-i-0f6ea5f2b68035f09.symcwss.splunkcloud.com",
           "main"
        ],
        "_sourcetype":"aws:lambda",
        "_time":"2020-04-09T08:11:43.000+00:00",
        "host":"http-inputs-symcwss.splunkcloud.com",
        "index":"main",
        "linecount":"1",
        "logevents":" [{'id': '35378345881050833235965050611284475263208966662210846720', 'timestamp':
 1586419896111, 'message': '2020/04/09 08:11:35 [error] 24367#24367: *12602277 \"/etc/nginx/html/index.html
\" is not found (2: No such file or directory), client: 172.168.2.170, server: pfms-stage.wss.symantec.com,
 request: \"GET / HTTP/1.0\"', 'extractedFields': {'11': 'No', '12': 'such', '13': 'file', '14':
 'or', '15': 'directory),', '16': 'client:', '17': '172.168.2.170,', '18': 'server:', '19': 'pfms-
stage.wss.symantec.com,', '1': '2020/04/09', '2': '08:11:35', '3': 'error', '4': '24367#24367:', '5':
 '*12602277', '6': '/etc/nginx/html/index.html', '7': 'is', '8': 'not', '9': 'found', '20': 'request:',
 '10': '(2:', '21': 'GET / HTTP/1.0'}}, {'id': 35378345903373879179694204375961729254129976529697243137',
 'timestamp': 1586419897112, 'message': '2020/04/09 08:11:36 [error] 24368#24368: *12602278 \"/etc/
nginx/html/index.html\" is not found (2: No such file or directory), client: 172.168.0.47, server: pfms-
stage.wss.symantec.com, request: \"GET / HTTP/1.0\"', 'extractedFields': {'11': 'No', '12': 'such', '13':
 'file', '14': 'or', '15': 'directory),', '16': 'client:', '17': '172.168.0.47,', '18': 'server:', '19':
 'pfms-stage.wss.symantec.com,', '1': '2020/04/09', '2': '08:11:36', '3': 'error', '4': '24368#24368:',
 '5': '*12602278', '6': '/etc/nginx/html/index.html', '7': 'is', '8': 'not', '9': 'found', '20': 'request:',
 '10': '(2:', '21': 'GET / HTTP/1.0'}}, {'id': 35378345925674624378224827517497447526778338035677659138',
 'timestamp': 1586419898112, 'message': '2020/04/09 08:11:37 [error] 24367#24367: *12602280 \"/etc/
nginx/html/index.html\" is not found (2: No such file or directory), client: 172.168.0.47, server: pfms-
stage.wss.symantec.com, request: \"GET / HTTP/1.0\"', 'extractedFields': {'11': 'No', '12': 'such', '13':
 'file', '14': 'or', '15': 'directory),', '16': 'client:', '17': '172.168.0.47,', '18': 'server:', '19':
 'pfms-stage.wss.symantec.com,', '1': '2020/04/09', '2': '08:11:37', '3': 'error', '4': '24367#24367:',
 '5': '*12602280', '6': '/etc/nginx/html/index.html', '7': 'is', '8': 'not', '9': 'found', '20': 'request:',

'10': '(2:', '21': 'GET / HTTP/1.0'}}, {'id': '35378345947975369576755450659033165799426699541658075139', 'timestamp': 1586419899112, 'message': '2020/04/09 08:11:39 [error] 24367#24367: *12602283 \"/etc/nginx/html/index.html\" is not found (2: No such file or directory), client: 172.168.2.170, server: pfms-stage.wss.symantec.com, request: \"GET / HTTP/1.0\"', 'extractedFields': {'11': 'No', '12': 'such', '13': 'file', '14': 'or', '15': 'directory),', '16': 'client:', '17': '172.168.2.170,', '18': 'server:', '19': 'pfms-stage.wss.symantec.com,', '1': '2020/04/09', '2': '08:11:39', '3': 'error', '4': '24367#24367:', '5': '*12602283', '6': '/etc/nginx/html/index.html', '7': 'is', '8': 'not', '9': 'found', '20': 'request:', '10': '(2:', '21': 'GET / HTTP/1.0'}}]}, ",
        "message":[
           " '2020/04/09 08:11:35 [error] 24367#24367: *12602277 \"/etc/nginx/html/index.html\" is not found (2: No such file or directory), client: 172.168.2.170, server: pfms-stage.wss.symantec.com, request: \"GET / HTTP/1.0\"', ",
           " '2020/04/09 08:11:36 [error] 24368#24368: *12602278 \"/etc/nginx/html/index.html\" is not found (2: No such file or directory), client: 172.168.0.47, server: pfms-stage.wss.symantec.com, request: \"GET / HTTP/1.0\"', ",
           " '2020/04/09 08:11:37 [error] 24367#24367: *12602280 \"/etc/nginx/html/index.html\" is not found (2: No such file or directory), client: 172.168.0.47, server: pfms-stage.wss.symantec.com, request: \"GET / HTTP/1.0\"', ",
           " '2020/04/09 08:11:39 [error] 24367#24367: *12602283 \"/etc/nginx/html/index.html\" is not found (2: No such file or directory), client: 172.168.2.170, server: pfms-stage.wss.symantec.com, request: \"GET / HTTP/1.0\"', "
        ],
        "source":"http:LogsOverHttp",
        "sourcetype":"aws:lambda",
        "splunk_server":"idx-i-0f6ea5f2b68035f09.symcwss.splunkcloud.com"
     },
     {
        "_bkt":"main~3167~EFE57FE7-8487-4BE1-A8AC-6C7297F7ED80",
        "_cd":"3167:55447962",
        "_indextime":"1586419851",
        "_raw":"{'logData': {'messageType': 'DATA_MESSAGE', 'owner': '524080397461', 'logGroup': '/aws/elasticbeanstalk/SYMC-WSSE-1AI88EJ8S5QR3/var/log/nginx/error.log', 'logStream': 'i-008c47bc8727f41a4', 'subscriptionFilters': ['CWToSplunkLambda'], 'logEvents': [{'id': '35378344743311414697329719114310413435019895395550298112', 'timestamp': 1586419845093, 'message': '2020/04/09 08:10:44 [error] 24368#24368: *12602159 \"/etc/nginx/html/index.html\" is not found (2: No such file or directory), client: 172.168.0.47, server: pfms-stage.wss.symantec.com, request: \"GET / HTTP/1.0\"', 'extractedFields': {'11': 'No', '12': 'such', '13': 'file', '14': 'or', '15': 'directory),', '16': 'client:', '17': '172.168.0.47,', '18': 'server:', '19': 'pfms-stage.wss.symantec.com,', '1': '2020/04/09', '2': '08:10:44', '3': 'error', '4': '24368#24368:', '5': '*12602159', '6': '/etc/nginx/html/index.html', '7': 'is', '8': 'not', '9': 'found', '20': 'request:', '10': '(2:', '21': 'GET / HTTP/1.0'}}, {'id': '35378344765634460641058872878987667425940905263036694529', 'timestamp': 1586419846094, 'message': '2020/04/09 08:10:45 [error] 24368#24368: *12602161 \"/etc/nginx/html/index.html\" is not found (2: No such file or directory), client: 172.168.0.47, server: pfms-stage.wss.symantec.com, request: \"GET / HTTP/1.0\"', 'extractedFields': {'11': 'No', '12': 'such', '13': 'file', '14': 'or', '15': 'directory),', '16': 'client:', '17': '172.168.0.47,', '18': 'server:', '19': 'pfms-stage.wss.symantec.com,', '1': '2020/04/09', '2': '08:10:45', '3': 'error', '4': '24368#24368:', '5': '*12602161', '6': '/etc/nginx/html/index.html', '7': 'is', '8': 'not', '9': 'found', '20': 'request:', '10': '(2:', '21': 'GET / HTTP/1.0'}}, {'id': '35378344787935205839589496020523385698589266769017110530', 'timestamp': 1586419847094, 'message': '2020/04/09 08:10:46 [error] 24367#24367: *12602163 \"/etc/nginx/html/index.html\" is not found (2: No such file or directory), client: 172.168.2.170, server: pfms-stage.wss.symantec.com, request: \"GET / HTTP/1.0\"', 'extractedFields': {'11': 'No', '12': 'such', '13': 'file', '14': 'or', '15': 'directory),', '16': 'client:', '17': '172.168.2.170,', '18': 'server:', '19': 'pfms-stage.wss.symantec.com,', '1': '2020/04/09', '2': '08:10:46', '3': 'error', '4': '24367#24367:', '5': '*12602163', '6': '/etc/nginx/html/

index.html', '7': 'is', '8': 'not', '9': 'found', '20': 'request:', '10': '(2:', '21': 'GET / HTTP/1.0'}},
 {'id': '35378344832558996981849272926736357962158638142483922947', 'timestamp': 1586419849095, 'message':
 '2020/04/09 08:10:48 [error] 24368#24368: *12602167 \"/etc/nginx/html/index.html\" is not found (2: No such
 file or directory), client: 172.168.2.170, server: pfms-stage.wss.symantec.com, request: \"GET / HTTP/1.0\"',
 'extractedFields': {'11': 'No', '12': 'such', '13': 'file', '14': 'or', '15': 'directory),', '16': 'client:',
 '17': '172.168.2.170,', '18': 'server:', '19': 'pfms-stage.wss.symantec.com,', '1': '2020/04/09', '2':
 '08:10:48', '3': 'error', '4': '24368#24368:', '5': '*12602167', '6': '/etc/nginx/html/index.html', '7':
 'is', '8': 'not', '9': 'found', '20': 'request:', '10': '(2:', '21': 'GET / HTTP/1.0'}}]}, 'logGroupName': '/
aws/elasticbeanstalk/SYMC-WSSE-1AI88EJ8S5QR3/var/log/nginx/error.log', 'logStreamName': 'i-008c47bc8727f41a4',
 'logEvents': [{'id': '35378344743311414697329719114310413435019895395550298112', 'timestamp': 1586419845093,
 'message': '2020/04/09 08:10:44 [error] 24368#24368: *12602159 \"/etc/nginx/html/index.html\" is not found
 (2: No such file or directory), client: 172.168.0.47, server: pfms-stage.wss.symantec.com, request: \"GET /
 HTTP/1.0\"', 'extractedFields': {'11': 'No', '12': 'such', '13': 'file', '14': 'or', '15': 'directory),',
 '16': 'client:', '17': '172.168.0.47,', '18': 'server:', '19': 'pfms-stage.wss.symantec.com,', '1':
 '2020/04/09', '2': '08:10:44', '3': 'error', '4': '24368#24368:', '5': '*12602159', '6': '/etc/nginx/html/
index.html', '7': 'is', '8': 'not', '9': 'found', '20': 'request:', '10': '(2:', '21': 'GET / HTTP/1.0'}},
 {'id': '35378344765634460641058872878987667425940905263036694529', 'timestamp': 1586419846094, 'message':
 '2020/04/09 08:10:45 [error] 24368#24368: *12602161 \"/etc/nginx/html/index.html\" is not found (2: No
 such file or directory), client: 172.168.0.47, server: pfms-stage.wss.symantec.com, request: \"GET /
 HTTP/1.0\"', 'extractedFields': {'11': 'No', '12': 'such', '13': 'file', '14': 'or', '15': 'directory),',
 '16': 'client:', '17': '172.168.0.47,', '18': 'server:', '19': 'pfms-stage.wss.symantec.com,', '1':
 '2020/04/09', '2': '08:10:45', '3': 'error', '4': '24368#24368:', '5': '*12602161', '6': '/etc/nginx/html/
index.html', '7': 'is', '8': 'not', '9': 'found', '20': 'request:', '10': '(2:', '21': 'GET / HTTP/1.0'}},
 {'id': '35378344787935205839589496020523385698589266769017110530', 'timestamp': 1586419847094, 'message':
 '2020/04/09 08:10:46 [error] 24367#24367: *12602163 \"/etc/nginx/html/index.html\" is not found (2: No
 such file or directory), client: 172.168.2.170, server: pfms-stage.wss.symantec.com, request: \"GET /
 HTTP/1.0\"', 'extractedFields': {'11': 'No', '12': 'such', '13': 'file', '14': 'or', '15': 'directory),',
 '16': 'client:', '17': '172.168.2.170,', '18': 'server:', '19': 'pfms-stage.wss.symantec.com,', '1':
 '2020/04/09', '2': '08:10:46', '3': 'error', '4': '24367#24367:', '5': '*12602163', '6': '/etc/nginx/html/
index.html', '7': 'is', '8': 'not', '9': 'found', '20': 'request:', '10': '(2:', '21': 'GET / HTTP/1.0'}},
 {'id': '35378344832558996981849272926736357962158638142483922947', 'timestamp': 1586419849095, 'message':
 '2020/04/09 08:10:48 [error] 24368#24368: *12602167 \"/etc/nginx/html/index.html\" is not found (2: No such
 file or directory), client: 172.168.2.170, server: pfms-stage.wss.symantec.com, request: \"GET / HTTP/1.0\"',
 'extractedFields': {'11': 'No', '12': 'such', '13': 'file', '14': 'or', '15': 'directory),', '16': 'client:',
 '17': '172.168.2.170,', '18': 'server:', '19': 'pfms-stage.wss.symantec.com,', '1': '2020/04/09', '2':
 '08:10:48', '3': 'error', '4': '24368#24368:', '5': '*12602167', '6': '/etc/nginx/html/index.html', '7':
 'is', '8': 'not', '9': 'found', '20': 'request:', '10': '(2:', '21': 'GET / HTTP/1.0'}}]}",
        "_serial":"5",
        "_si":[
           "idx-i-04a7725142721e62b.symcwss.splunkcloud.com",
           "main"
        ],
        "_sourcetype":"aws:lambda",
        "_time":"2020-04-09T08:10:51.000+00:00",
        "host":"http-inputs-symcwss.splunkcloud.com",
        "index":"main",
        "linecount":"1",
        "logevents":" [{'id': '35378344743311414697329719114310413435019895395550298112', 'timestamp':
 1586419845093, 'message': '2020/04/09 08:10:44 [error] 24368#24368: *12602159 \"/etc/nginx/html/index.html
\" is not found (2: No such file or directory), client: 172.168.0.47, server: pfms-stage.wss.symantec.com,
 request: \"GET / HTTP/1.0\"', 'extractedFields': {'11': 'No', '12': 'such', '13': 'file', '14': 'or', '15':
 'directory),', '16': 'client:', '17': '172.168.0.47,', '18': 'server:', '19': 'pfms-stage.wss.symantec.com,',
 '1': '2020/04/09', '2': '08:10:44', '3': 'error', '4': '24368#24368:', '5': '*12602159', '6': '/etc/

```
nginx/html/index.html', '7': 'is', '8': 'not', '9': 'found', '20': 'request:', '10': '(2:', '21': 'GET /
 HTTP/1.0'}}, {'id': '35378344765634460641058872878987667425940905263036694529', 'timestamp': 1586419846094,
 'message': '2020/04/09 08:10:45 [error] 24368#24368: *12602161 \"/etc/nginx/html/index.html\" is not found
 (2: No such file or directory), client: 172.168.0.47, server: pfms-stage.wss.symantec.com, request: \"GET /
 HTTP/1.0\"', 'extractedFields': {'11': 'No', '12': 'such', '13': 'file', '14': 'or', '15': 'directory),',
 '16': 'client:', '17': '172.168.0.47,', '18': 'server:', '19': 'pfms-stage.wss.symantec.com,', '1':
 '2020/04/09', '2': '08:10:45', '3': 'error', '4': '24368#24368:', '5': '*12602161', '6': '/etc/nginx/html/
index.html', '7': 'is', '8': 'not', '9': 'found', '20': 'request:', '10': '(2:', '21': 'GET / HTTP/1.0'}},
 {'id': '35378344787935205839589496020523385698589266769017110530', 'timestamp': 1586419847094, 'message':
 '2020/04/09 08:10:46 [error] 24367#24367: *12602163 \"/etc/nginx/html/index.html\" is not found (2: No
 such file or directory), client: 172.168.2.170, server: pfms-stage.wss.symantec.com, request: \"GET /
 HTTP/1.0\"', 'extractedFields': {'11': 'No', '12': 'such', '13': 'file', '14': 'or', '15': 'directory),',
 '16': 'client:', '17': '172.168.2.170,', '18': 'server:', '19': 'pfms-stage.wss.symantec.com,', '1':
 '2020/04/09', '2': '08:10:46', '3': 'error', '4': '24367#24367:', '5': '*12602163', '6': '/etc/nginx/html/
index.html', '7': 'is', '8': 'not', '9': 'found', '20': 'request:', '10': '(2:', '21': 'GET / HTTP/1.0'}},
 {'id': '35378344832558996981849272926736357962158638142483922947', 'timestamp': 1586419849095, 'message':
 '2020/04/09 08:10:48 [error] 24368#24368: *12602167 \"/etc/nginx/html/index.html\" is not found (2: No such
 file or directory), client: 172.168.2.170, server: pfms-stage.wss.symantec.com, request: \"GET / HTTP/1.0\"',
 'extractedFields': {'11': 'No', '12': 'such', '13': 'file', '14': 'or', '15': 'directory),', '16': 'client:',
 '17': '172.168.2.170,', '18': 'server:', '19': 'pfms-stage.wss.symantec.com,', '1': '2020/04/09', '2':
 '08:10:48', '3': 'error', '4': '24368#24368:', '5': '*12602167', '6': '/etc/nginx/html/index.html', '7':
 'is', '8': 'not', '9': 'found', '20': 'request:', '10': '(2:', '21': 'GET / HTTP/1.0'}}]}, ",
        "message":[
           " '2020/04/09 08:10:44 [error] 24368#24368: *12602159 \"/etc/nginx/html/index.html\" is not found
 (2: No such file or directory), client: 172.168.0.47, server: pfms-stage.wss.symantec.com, request: \"GET /
 HTTP/1.0\"', ",
           " '2020/04/09 08:10:45 [error] 24368#24368: *12602161 \"/etc/nginx/html/index.html\" is not found
 (2: No such file or directory), client: 172.168.0.47, server: pfms-stage.wss.symantec.com, request: \"GET /
 HTTP/1.0\"', ",
           " '2020/04/09 08:10:46 [error] 24367#24367: *12602163 \"/etc/nginx/html/index.html\" is not found
 (2: No such file or directory), client: 172.168.2.170, server: pfms-stage.wss.symantec.com, request: \"GET /
 HTTP/1.0\"', ",
           " '2020/04/09 08:10:48 [error] 24368#24368: *12602167 \"/etc/nginx/html/index.html\" is not found
 (2: No such file or directory), client: 172.168.2.170, server: pfms-stage.wss.symantec.com, request: \"GET /
 HTTP/1.0\"', "
        ],
        "source":"http:LogsOverHttp",
        "sourcetype":"aws:lambda",
        "splunk_server":"idx-i-04a7725142721e62b.symcwss.splunkcloud.com"
      }
   ],
   "highlighted":{

   }
}
```


The following Javascript is used for calculating the match counter for each of the messages against the log event document.

```
"var counts="[
```

```
]; for (var i = 0; i<root.results.length;i++){
   "var message = root.results"[
      "i"
   ]   ".message; var count = (message.toString().match(/No such file or directory/g) ||"[

   ]   ").length; counts.push(count);"
}"return counts;"
```

For aggregating the counter values, the metric mapping parameter "performAggregation" is set to "true". When this parameter is available and is "true", then the sum of the metric values is derived based on the *metric_content_id (ci_unique_id+":"+metric_name+":"+host+ ":"+configuration_item)*. The following snippet illustrates the metric mapping:

```
"metrics":[
   {
      "xml_ns":"",
      "url":"searchresults",
      "group":"Search",
      "performAggregation":"true",
      "attributes":{
         "oi":{
            "type":"2",
            "metric_name":"No Such File or Directory Counter",
            "metric_type":"Counter Metrics",
            "configuration_item_type":"$['results'][*]['sourcetype']",
            "metric_unique_id":"$.legends.rows[*].id%//%:%//%$.legends.rows[*].performanceobject%//%:%//%
$.legends.rows[*].counter",
            "metric_unit":"",
            "configuration_item":"$['results'][*]['source']",
            "host":"$['results'][*]['host']",
            "ip":"",
            "product":"Splunk",
            "ci_unique_id":"$['results'][*]['host']",
            "product_version":"1.2",
            "message":"The No Such File or Directory value <metric_value> is greater than the threshold
 value",
            "severity_conversion":">100:Critical,20-80:Major,10-19:Minor,Default:Information",
            "alarm_unique_id":"$['results'][*]['host']%//%:%//%this is my alarm for No Such File or
 Directory",
            "generateAlarm":"true",
            "alarmType":"Log"
         }
      },
      "value":"#(function(){var counts=[]; for (var i = 0; i<root.results.length;i++) { var message =
 root.results[i].message; var count = (message.toString().match(/No such file or directory/g) || []).length;
 counts.push(count); } return counts;})();"
   }
]
```

Create Custom Alarms

Once you derive the metric counter, you can generate an alarm based on the following definition which is defined in the metrics section.

```
"message": "The No Such File or Directory value <metric_value> is greater than the threshold
 value", "severity_conversion":
 ">100:Critical,20-80:Major,10-19:Minor,Default:Information", "alarm_unique_id": "$['results'][*]['host']%//%:
%//%this is my alarm for No Such File or Directory", "generateAlarm": "true", "alarmType": "Log"
```

The metric and the alarm generated are ingested into OI as the following snapshot illustrates.

ThousandEyes

DX Operational Intelligence supports integration with ThousandEyes.

The ThousandEyes schema enables you to monitor the performance and usage of your network by ingesting the network metrics and alerts to DX Operational Intelligence (DX OI). This schema enables you to ingest the following monitoring data:

- Inventory
- Metrics
- Alarms
- Topology
- Groups

This section provides the following information:

Configure the Integration

The ThousandEyes - DX OI integration involves the following steps:

- Configure the ThousandEyes Environment
- Configure RESTMon

Configure the ThousandEyes Environment

No integration-specific steps are required to be performed in your ThousandEyes environment. However, ensure that the following requirements are met:

- The ThousandEyes environment is set up to retrieve metrics and alerts.
- The ThousandEyes API token is available. This token can be found in the User API Token section of the Profile tab of the Account Settings page.

For more information, see the ThousandEyes Documentation.

Configure RESTMon

To configure RESTMon, update the OI connection details if not done earlier, and add the profile information to the **restmon.json** file. Before you configure RESTMon, ensure that the following requirements are met:

- You have access to DX OI.
- RESTMon is configured and deployed successfully. For more information, see the Configure and Deploy RESTMon section.
- RESTMon is configured and deployed successfully.
- For Java 11, replace the *contains* method with *includes* in the schema.

Add the Profile

To add the profile, configure the profile to connect to your ThousandEyes environment and add the profile to the **restmon.json** file using the POST Profile REST API call in Swagger. The **thousandeyes_profile.json** file is available in the **<restmon\profile>** folder. When you add the profile, the schema for ThousandEyes is automatically uploaded, and the data ingestion starts.

> **NOTE**
> You can also add the attribute filter to the profile to filter the entities and the related data. For more information, see the **attribute_filter** section on this page.

You can also add this information directly in the **restmon.json** file.

The ThousandEyes profile includes the following sections:

Profile

The **profile** section defines the profile-related information. The following snippet is a sample of the profile section.

```
{
        "name": "thousandeyes",
        "active": "yes",
        "schema": "thousandeyes",
        "polling_interval_secs": "300",
        "inventory_topology_fullsync_interval_mins": "1440",
        "topology_ttl_mins": "2880"
    }
```

Name	Description	Type	Example
name	Indicates the of the profile.	String	thousandeyes
schema	Indicates the name of the schema. The name that you specify for the schema should be the same as the schema attribute specified in the restmon.json file.	String	thousandeyes
polling_interval_secs	Indicates the polling interval in seconds. **Supported Values:** 15, 30, 60, 300, 900, 1800, 3600, and 7200 Any other value is rounded off to the nearest intervals.	Integer	60
inventory_topology_fullsync_interval_mins	Indicates the full synchronization interval in minutes.	Integer	1440

Name	Description	Type	Example
topology_ttl_mins	Indicates the time-to-live (TTL) a record is cached in minutes.	Integer	2880
tenantname	Indicates the DX Operational Intelligence tenant name.	String	sampleTenantName
active	Indicates if the data-processing is enabled. Enter *yes* to enable the profile.	Boolean	yes

restapiconnectdetails

In the **restapiconnectdetails** section, enter the REST Endpoint details of your ThousandEyes environment. The following snippet is a sample of the **restapiconnectdetails** section:

```
{
        "type": "https",
        "hostname": "SampleHostName",
        "port": "",
        "authentication": "urltoken",
        "username": "",
        "password": "",
        "realmdomain": "",
        "token": "SampleToken",
        "httptimeout": "30000",
        "checkcert": "no"
    }
```

Name	Description	Type	Example
type	Indicates the data transfer type with OI in HTTP or HTTPS.	String	http
hostname	Indicates the hostname or IP address of the REST Endpoint.	String	test.example.net
port	Indicates the port number of the REST Endpoint.	Integer	9600
authentication	Indicates the authentication type. Enter urltoken.	String	none
realmdomain	Indicates the DNS realm or domain to encode in the token. Applies only when the authentication type is digest.	String	
token	Indicates the access token or bearer token when the authentication type is OAuth2 or bearer, respectively.	String	
httptimeout	Indicates the value of the timeout that is expressed in milliseconds.	Integer	300
checkcert	Indicates to verify the certificate to ensure it is valid and trusted.	Boolean	no

monitored_groups

In the **monitored_groups** section, specify the groups that you want to monitor. The following snippet is a sample of the **monitored_groups** section:

```
{
        "inventory_group":"no",
        "topology_group": "yes",
        "topology_group_cloud": "yes",
        "Alarms": "yes",
        "Alarms_pathTraces": "yes",
        "host_group": "yes",
        "cloud_metrics": "yes",
        "path_metrics": "yes"
    }
```

attribute_filter

You can configure RESTMon to ingest and display only the required information on the DX Operational Intelligence UI using the attribute filter. To filter the ingested data, add the attribute filter to the profile.json file. For more information, see the Filter Entities and Related Data Before Ingestion section.

> **NOTE**
> For the detailed steps, see the Add the Profile section.

Upload the ThousandEyes Schema

The schema is automatically uploaded when you add the profile. Perform the steps only if you want to upload the updated or edited schema.

> **NOTE**
> For the detailed steps, see the Upload the Schema section.

View Data in DX OI

You can view the ingested data in the Alarm Analytics, Service Analytics, and Performance Analytics pages in DX OI for insights into service, raw, and anomaly alarms.

> **NOTE**
> For more information, see the View Data in DX OI section.

Tivoli Netcool/OMNIbus

DX Operational Intelligence supports an integration with Tivoli Netcool/OMNIbus.

The Tivoli Netcool/OMNIbus schema enables you to ingest the Tivoli Netcool/OMNIbus alerts as alarms into DX Operational Intelligence .

You can perform this integration using one of the following methods:

- **Polling:** This method enables RESTMon to get the alert data from Tivoli Netcool/OMNIbus using the Tivoli Netcool/OMNIbus schema (**netcoolomnibus_schema.json**) and profile (**netcoolomnibus_profile.json**).
- **Webhook (RESTMon Streaming):** This method enables Tivoli Netcool/OMNibus to post the alert data to RESTMon as JSON using the Webhook schema (**netcoolomnibuswebhook_schema.json**) and profile (**netcoolomnibuswebhook_profile.json)**.

This section provides the following information:

Supported Versions

The Tivoli Netcool/OMNIbus - DX Operational Intelligence integration is supported for the following version:

Product	Supported Version
Tivoli Netcool/OMNIbus	8.1.0

Configure the Integration

The Tivoli Netcool/OMNIbus - DX Operational Intelligence integration involves the following steps:

- Configure the Tivoli Netcool/OMNIbus Environment
- Configure RESTMon

Configure the Tivoli Netcool/OMNIbus Environment

To set up the integration, ensure that the following requirements are met:

- For the Polling method, ensure that you have installed ObjectServer HTTP interface.
- For the Webhook method, ensure that you have installed Message Bus Gateway. Set up the Message Bus Gateway to send the alarm payload to the Webhook URL using the Basic authentication (user name and password). The example URL is *https://fqdn:8443/restmon/api/v1/logs?profileName=netcoolomnibus_webhook&schemaName=netcoolomnibus_webhook*.

  NOTE
  You can change the mapping of what data goes in the payload. If you change the format of the payload, you may have to change the schema.
- When using the Webhook schema, you must define the Webhook URL to which Tivoli Netcool/OMNIbus sends the data. An example of the URL is *https://fqdn:8443/restmon/api/v1/logs?profileName=netcoolomnibus_webhook&schemaName=netcoolomnibus_webhook.*
- You have the username and password for the Basic authentication connection to RESTMon.

  NOTE
  For more information about Tivoli Netcool/OMNIbus, see the Tivoli Netcool/OMNIbus documentation.

Configure RESTMon

To configure RESTMon, update the OI connection details if not done earlier and add the profile information to the **restmon.json** file. Before you configure RESTMon, ensure that the following requirements are met:

- You have access to DX Operational Intelligence.
- RESTMon is configured and deployed successfully. For more information, see the Configure and Deploy RESTMon section.
- RESTMon is configured and deployed successfully.

Add the Profile

To add the profile, configure the profile to connect to your Tivoli Netcool/OMNIbus environment and add the profile to the **restmon.json** file using the POST Profile REST API call in Swagger. The **netcoolomnibus_profile.json** file is available in the **<restmon\profile>** folder. When you add the profile, the schema for Tivoli Netcool/OMNIbus is automatically uploaded and the data ingestion starts.

You can also add this information directly in the **restmon.json** file.

The Tivoli Netcool/OMNIbus profile includes the following sections:

Profile

The **profile** section defines the profile-related information.

Profile - Polling Method

The following snippet is a sample of the profile section.

```
{
        "name":"netcoolomnibus",
        "schema":"netcoolomnibus",
        "streaming":"no",
        "polling_interval_secs":"300",
        "inventory_topology_fullsync_interval_mins":"1440",
        "topology_ttl_mins":"2880",
        "active":"yes"
               }
```

Name	Description	Type	Example
name	Indicates the name of the profile.	String	netcoolomnibus
schema	Indicates the name of the schema. The name that you provide for the schema should be the same as the schema attribute specified in the restmon.json file.	String	netcoolomnibus
streaming	Indicates whether streaming is enabled. If enabled, the integrating product posts data to RESTMon as JSON (Webhook). If not, RESTMon gets (polling) data from the integrating product.	Boolean	no
polling_interval_secs	Indicates the polling interval in seconds. **Supported Values:** 15, 30, 60, 300, 900, 1800, 3600, and 7200 Any other value is rounded off to the nearest intervals.	Integer	300
inventory_topology_fullsync_interval_mins	Indicates the full synchronization interval in minutes.	Integer	1440
topology_ttl_mins	Indicates the time-to-live (TTL) record is cached in minutes.	Integer	2880
tenantname	Indicates the DX Operational Intelligence tenant name.	String	sampleTenantName
active	Indicates whether the data processing is enabled. Enter *yes* to enable the profile.	Boolean	yes

Profile - Webhook Method

The following snippet is a sample of the profile section.

```
{
        "name":"netcoolomnibuswebhook",
        "active":"yes",
        "schema":"netcoolomnibuswebhook",
```

```
        "streaming":"yes",
        "polling_interval_secs":"1",
        "inventory_topology_fullsync_interval_mins":"1440",
        "topology_ttl_mins":"2880",
        "tenantname":"sampleTenantName",
        "batch_size":1000,
        "batch_wait_time_milli":2000
    }
```

Name	Description	Type	Example
name	Indicates the name of the profile.	String	netcoolomnibuswebhook
schema	Indicates the schema name. The name that you provide for the schema should be the same as the schema attribute specified in the restmon.json file.	String	netcoolomnibuswebhook
streaming	Indicates whether streaming is enabled. If enabled, the integrating product posts data to RESTMon as JSON (Webhook). If not, RESTMon gets (polling) data from the integrating product.	Boolean	yes
polling_interval_secs	Indicates the polling interval in seconds. **Supported Values:** 15, 30, 60, 300, 900, 1800, 3600, and 7200 Any other value is rounded off to the nearest intervals.	Integer	1
inventory_topology_fullsync_interval_mins	Indicates the full synchronization interval in minutes.	Integer	1440
topology_ttl_mins	Indicates the time-to-live (TTL) record is cached in minutes.	Integer	2880
tenantname	Indicates the DX Operational Intelligence tenant name.	String	sampleTenantName
active	Indicates whether the data processing is enabled. Enter *yes* to enable the profile.	Boolean	yes
batch_size	Indicates the size of the batch for the incoming data.	Integer	1000
batch_wait_time_milli	Indicates the wait time for the batch.	Integer	2000

restapiconnectdetails

In the **restapiconnectdetails** section, enter the REST Endpoint details of your environment.

restapiconnectdetails - Polling Method

The following snippet is a sample of the **restapiconnectdetails** section:

```
{
        "type":"https",
        "hostname":"sampleHostName",
```

```
    "port":"",
    "authentication":"basic",
    "username":"sampleUser",
    "password":"samplePassword",
    "realmdomain":"",
    "token":"",
    "httptimeout":"30000",
    "checkcert":"no"
}
```

Name	Description	Type	Example
type	Indicates the data transfer type with DX Operational Intelligence. Values: HTTP or HTTPS.	String	https
hostname	Indicates the hostname or IP address of the REST endpoint.	String	sampleHostName
port	Indicates the port number of the REST endpoint.	Integer	
authentication	Indicates the authentication type. The following authentication types are available: • **none:** No authorization required. • **basic:** Enter the username and password. • **NTLM:** Enter the username and password. • **digest:** Enter the username, password, and the realmdomain. • **OAuth2:** Enter the access token in the token parameter. • **bearer:** Enter the bearer token in the token parameter. • **urltoken:** Enter the token in the token parameter.	String	basic
username	Indicates the username. Applies only when the authentication type is basic or NTLM.	String	sampleUserName
password	Indicates the password. Applies only when the authentication type is basic or NTLM.	String	samplePassword
realmdomain	Indicates the DNS realm or domain to encode in the token. Applies only when the authentication type is digest.	String	

Name	Description	Type	Example
token	Indicates the access token or bearer token when the authentication type is OAuth2 or bearer respectively.	String	
httptimeout	Indicates the value of the timeout that is expressed in milliseconds.	Integer	30000
checkcert	Indicates to verify the certificate to ensure it is valid and trusted.	Boolean	no

restapiconnectdetails - Webhook Method

The following snippet is a sample of the **restapiconnectdetails** section:

```
{
        "type":"https",
        "hostname":"dummy.value.com",
        "port":"7083",
        "authentication":"none",
        "username":"",
        "password":"",
        "realmdomain":"",
        "token":"",
        "httptimeout":"60000",
        "checkcert":"no"
    }
```

Name	Description	Type	Example
type	Indicates the data transfer type with DX Operational Intelligence. Values: HTTP or HTTPS.	String	https
hostname	Indicates the hostname or IP address of the REST endpoint.	String	sampleHostName
port	Indicates the port number of the REST endpoint.	Integer	7083

Name	Description	Type	Example
authentication	Indicates the authentication type. The following authentication types are available: • **none:** No authorization required. • **basic:** Enter the username and password. • **NTLM:** Enter the username and password. • **digest:** Enter the username, password, and the realmdomain. • **OAuth2:** Enter the access token in the token parameter. • **bearer:** Enter the bearer token in the token parameter. • **urltoken:** Enter the token in the token parameter.	String	none
username	Indicates the username. Applies only when the authentication type is basic or NTLM.	String	
password	Indicates the password. Applies only when the authentication type is basic or NTLM.	String	
realmdomain	Indicates the DNS realm or domain to encode in the token. Applies only when the authentication type is digest.	String	
token	Indicates the access token or bearer token when the authentication type is OAuth2 or bearer respectively.	String	
httptimeout	Indicates the value of the timeout that is expressed in milliseconds.	Integer	60000
checkcert	Indicates to verify the certificate to ensure it is valid and trusted.	Boolean	no

monitored_groups

In the **monitored_groups** section, enter the groups that you want to monitor. The following snippet is a sample of the **monitored_groups** section:

```
{
        "Alarms":"yes"
    }
```

NOTE
For the detailed steps, see the Add the Profile section.

Upload the Tivoli Netcool/OMNIbus Schema

The schema is automatically uploaded when you add the profile. Perform the steps only if you want to upload the updated or edited schema.

> **NOTE**
> For the detailed steps, see the Upload the Schema section.

View Data in DX OI

You can view the ingested data in the Alarm Analytics, Service Analytics, and Performance Analytics pages in DX OI for insights into service, raw, and anomaly alarms.

> **NOTE**
> For more information, see the View Data in DX OI section.

Zabbix

DX Operational Intelligence supports an integration with Zabbix.

The Zabbix schema enables you to retrieve events from the Zabbix systems and ingest them into DX Operational Intelligence. This schema also enables you to post the alert data from Zabbix into DX Operational Intelligence. You can retrieve or post the following data:

- **Inventory:** Hosts are ingested as inventory.
- **Topology:** Hosts are ingested as topology.
- **Alarms:** Problems are ingested as alarms.

  > **NOTE**
  > Alarms do not have metrics attached to them. In Zabbix, events define thresholds; thresholds in Zabbix do not have to be based on an item (metric). Therefore, Zabbix problems have no link to the item (metric).
- **Metrics:** Items are ingested as metrics.

You can retrieve and post data using the following methods:

- **Polling Method:** In the polling method, you can enable RESTMon to poll your Zabbix server at regular intervals to retrieve data using the Zabbix schema (**zabbix_schema.json**) and profile (**zabbix_profile.json)**.
- **Webhook (RESTMon Streaming) Method:** In the streaming method, you can enable Zabbix to post the alert data to RESTMon as JSON using the Webhook schema (**zabbixwebhook_schema.json**) and profile (**zabbixwebhook_profile.json**).

Configure the Integration

The Zabbix-DX Operational Intelligence integration involves the following steps:

- Configure the Zabbix Environment
- Configure RESTMon

Configure the Zabbix Environment

No additional integration-specific steps are required to be performed in your Zabbix environment. However, ensure that the following requirements are met:

- You can connect to the Zabbix server.
- You have the Zabbix server API URL.
- The Zabbix server port is open and is accessible from DX Operational Intelligence.

For more information, see the Zabbix documentation.

Configure RESTMon

To configure RESTMon, update the OI connection details if not done earlier and add the profile information to the **restmon.json** file. Before you configure RESTMon, ensure that the following requirements are met:

- You have access toDX Operational Intelligence.
- RESTMon is configured and deployed successfully. For more information, see the Configure and Deploy RESTMon section.
- RESTMon is configured and deployed successfully.

Add the Profile

To add the profile, configure the profile to connect to your Zabbix environment and add the profile to the **restmon.json** file using the POST Profile REST API call in Swagger. The polling *(zabbix_profile.json)* and streaming *(zabbixwebhook_profile.json)* files are available in the **<restmon\profile>** folder. When you add the profile, the schema for Zabbix is automatically uploaded, and the data ingestion starts.

> **NOTE**
> You can also add the attribute filter to the profile to filter the entities and the related data. For more information, see the **attribute_filter** section on this page.

You can also add this information directly in the **restmon.json** file.

The Zabbix profile includes the following sections:

Profile

The **profile** section defines the profile-related information.

Profile - Polling Method

The following snippet is a sample of the profile section.

```
{
      "name":"zabbix",
      "active":"yes",
      "schema":"zabbix",
      "polling_interval_secs":"300",
      "inventory_topology_fullsync_interval_mins":"1440",
      "topology_ttl_mins":"2880"
   }
```

Name	Description	Type	Example
name	Indicates the name of the profile.	String	zabbix
schema	Indicates the name of the schema. The name that you provide for the schema should be the same as the schema attribute specified in the restmon.json file.	String	zabbix
polling_interval_secs	Indicates the polling interval in seconds. **Supported Values:** 15, 30, 60, 300, 900, 1800, 3600, and 7200 Any other value is rounded off to the nearest intervals.	Integer	60

Name	Description	Type	Example
inventory_topology_fullsync_interval_mins	Indicates the full synchronization interval in minutes.	Integer	1440
topology_ttl_mins	Indicates the time-to-live (TTL) record is cached in minutes.	Integer	2880
tenantname	Indicates the DX Operational Intelligence tenant name.	String	sampleTenantName
active	Indicates whether the data processing is enabled. Enter *yes* to enable the profile.	Boolean	yes

Profile - Webhook Method

The following snippet is a sample of the profile section.

```
{
      "name":"zabbixwebhook",
      "active":"yes",
      "schema":"zabbixwebhook",
      "streaming":"yes",
      "polling_interval_secs":1,
      "batch_size":1000,
      "batch_wait_time_milli":2000,
      "tenantname":"sampleTenantName"
   }
```

Name	Description	Type	Example
name	Indicates the name of the profile.	String	zabbix
schema	Indicates the name of the schema. The name that you provide for the schema should be the same as the schema attribute specified in the restmon.json file.	String	zabbix
polling_interval_secs	Indicates the polling interval in seconds. **Supported Values:** 15, 30, 60, 300, 900, 1800, 3600, and 7200 Any other value is rounded off to the nearest intervals.	Integer	60
inventory_topology_fullsync_interval_mins	Indicates the full synchronization interval in minutes.	Integer	1440
topology_ttl_mins	Indicates the time-to-live (TTL) record is cached in minutes.	Integer	2880
tenantname	Indicates the DX Operational Intelligence tenant name.	String	sampleTenantName
active	Indicates whether the data processing is enabled. Enter *yes* to enable the profile.	Boolean	yes

restapiconnectdetails

In the **restapiconnectdetails** section, enter the REST Endpoint details of your Zabbix environment.

restapiconnectdetails - Polling Method

The following snippet is a sample of the *restapiconnectdetails* section:

```
":{
      "type":"http",
      "hostname":"sampleHostName",
      "port":"samplePort",
      "authentication":"none",
      "username":"sampleUserName",
      "password":"samplePassword",
      "realmdomain":"",
      "token":"",
      "httptimeout":"30000",
      "checkcert":"no"
   },
```

Name	Description	Type	Example
type	Indicates the data transfer type with Operational Intelligence. Values: HTTP or HTTPS.	String	http
hostname	Indicates the hostname or IP address of the REST endpoint.	String	test.example.net
port	Indicates the port number of the REST endpoint.	Integer	80
authentication	Indicates the authentication type. For the Zabbix integration, you can set none. Additionally, the following authentication types are available: • **none:** No authorization is required. • **basic:** Enter the username and password. • **NTLM:** Enter the username and password. • **digest:** Enter the username, password, and the realmdomain. • **OAuth2:** Enter the access token in the token parameter. • **bearer:** Enter the bearer token in the token parameter. • **urltoken:** Enter the token in the token parameter.	String	none
username	Indicates the username. Applies only when the authentication type is basic or NTLM.	String	sampleUserName

Name	Description	Type	Example
password	Indicates the password. Applies only when the authentication type is basic or NTLM.	String	samplePassword
realmdomain	Indicates the DNS realm or domain to encode in the token. Applies only when the authentication type is digest.	String	
token	Indicates the access token or bearer token when the authentication type is OAuth2 or bearer, respectively.	String	sampleToken
httptimeout	Indicates the value of the timeout that is expressed in milliseconds.	Integer	60
checkcert	Indicates to verify the certificate to ensure it is valid and trusted.	Boolean	no

restapiconnectdetails - Webhook Method

The following snippet is a sample of the **restapiconnectdetails** section:

```
{
      "type":"http",
      "hostname":"notneeded",
      "port":9600,
      "authentication":"none",
      "username":"",
      "password":"",
      "realmdomain":"",
      "token":"",
      "httptimeout":300,
      "checkcert":"no"
   }
```

Name	Description	Type	Example
type	Indicates the data transfer type with DX Operational Intelligence. Values: HTTP or HTTPS.	String	http
hostname	Indicates the hostname or IP address of the REST endpoint.	String	test.example.net
port	Indicates the port number of the REST endpoint.	Integer	80

Name	Description	Type	Example
authentication	Indicates the authentication type. For the Zabbix integration, you can set none. Additionally, the following authentication types are available: • **none:** No authorization is required. • **basic:** Enter the username and password. • **NTLM:** Enter the username and password. • **digest:** Enter the username, password, and the realmdomain. • **OAuth2:** Enter the access token in the token parameter. • **bearer:** Enter the bearer token in the token parameter. • **urltoken:** Enter the token in the token parameter.	String	none
username	Indicates the username. Applies only when the authentication type is basic or NTLM.	String	sampleUserName
password	Indicates the password. Applies only when the authentication type is basic or NTLM.	String	samplePassword
realmdomain	Indicates the DNS realm or domain to encode in the token. Applies only when the authentication type is digest.	String	
token	Indicates the access token or bearer token when the authentication type is OAuth2 or bearer, respectively.	String	sampleToken
httptimeout	Indicates the value of the timeout that is expressed in milliseconds.	Integer	60
checkcert	Indicates to verify the certificate to ensure it is valid and trusted.	Boolean	no

monitored_groups

In the **monitored_groups** section, specify the groups that you want to monitor. The following snippet is a sample of the **monitored_groups** section:

```
{
      "Events":"yes"
   }
```

attribute_filter

You can configure RESTMon to ingest and display only the required information on the DX Operational Intelligence UI using the attribute filter. To filter the ingested data, add the attribute filter to the profile.json file. For more information, see the Filter Entities and Related Data Before Ingestion section.

> **NOTE**
> For the detailed steps, see the Add the Profile section.

Upload the Zabbix Schema

The schema is automatically uploaded when you add the profile. Perform the steps only if you want to upload the updated or edited schema.

> **NOTE**
> For the detailed steps, see the Upload the Schema section.

View Data in DX OI

You can view the ingested data in the Alarm Analytics, Service Analytics, and Performance Analytics pages in DX OI for insights into service, raw, and anomaly alarms.

> **NOTE**
> For more information, see the View Data in DX OI section.

RESTMon Self-Monitoring

This section provides the following information:

- Monitoring Dashboards
- Supportability Metrics
- Liveness and Readiness Check

Monitoring Dashboards

The following dashboards are available out-of-the-box for RESTMon. These dashboards display the supportability metrics. You can find these dashboards in DX Dashboards.

- RESTMon: Data Collector
- RESTMon: Datastore
- RESTMon: Monitoring Overview
- RESTMon: Publisher

> **NOTE**
> For more information, see the RESTMon Dashboards (SaaS) and RESTMon Dashboards (On-Premise) sections.

Supportability Metrics

Supportability metrics enable you to monitor the health of RESTMon. To enable the supportability metrics, configure the supportability metrics settings in the *values.yaml* file, or environment variables in the *docker-compose.yaml* file. Alternatively, you can pass them as arguments during deployment using Helm.

> **NOTE**
> Supportability Metrics is not supported if the proxy is configured between RESTMon and DX OI.

Configure the following settings:

- **supportability.instanceName:** Indicates the instance name. The instance name identifies the metrics of each of the instances that are deployed per tenant in the dashboard.
- **supportability.agentToken**: Indicates the agent token. This token is available in the Connector Parameters page.
- **supportability.agentName:** Indicates the agent name against which the metrics are ingested during the installation or anytime after the installation.
- **supportability.apiEndpoint:** Indicates the API endpoint.

For example, run the following command to pass these settings as arguments during the deployment:

```
helm install <name> <chart> --set restmon.id=<id> --set
 restmon.settings.supportability_agentToken=<agentToken> --set
 restmon.settings.supportability_instanceName=<instance> --set
 restmon.settings.supportability_apiEndpont=<apiEndpoint>

For example, helm install restmon1 helm-charts-1.0.5-restmon.tgz --set restmon.id=1
 --set restmon.settings.supportability_agentToken=eyJ0eXAi0iJKV1QiLCJhb.eyJhz2VudCI6
 --set restmon.settings.supportability_instanceName=myinstance1 --set
 restmon.settings.supportability_apiEndpont=https://axa.dxi-na1.saas.broadcom.com:443
```

Out-of-the-box Metrics

The following table lists the metrics that are ingested to NASS every 15 seconds. You can view these metrics in the RESTMon dashboards:

Metric Group	Metric Name	Operation of Metric
Resources	CPU:CPU Used (%)	Indicates the sum of the total CPU usage in percentage for individual cores. The usage can be more than 100%. For example, 230% means it consumed 2.3 CPU cores.
	CPU:CPU Used (ms)	Indicates the sum of the total CPU usage for individual cores in milliseconds per interval.
	CPU:Kernel CPU (%)	Indicates the sum of the CPU usage in percentage that is spent in Kernel (System time).
	CPU:Kernel CPU (ms)	Indicates the sum of the CPU usage that is spent in kernel (System time) in milliseconds per interval.
	CPU:User CPU (%)	Indicates the sum of the CPU usage in percentage that is spent in the user space.
	CPU:User CPU (ms)	Indicates the sum of the CPU usage that is spent in the user space in milliseconds per interval.
	Memory:GC Count	Indicates the number of the GC cycles per interval.
	Memory:GC Time (ms)	Indicates the GC collection time.
	Memory:Memory Heap Committed	Indicates the amount of memory that is committed in kernel.

Metric Group	Metric Name	Operation of Metric
	Memory:Memory Heap Max	Indicates the maximum heap memory (Xmx).
	Memory:Memory Heap Used	Amount of heap used.
	Memory:Memory No Heap Committed	Indicates that no heap memory is committed (direct buffers, and so on).
	Memory:Memory No Heap Max	Indicates the maximum no heap memory.
	Memory:Memory No Heap Used	Indicates the amount of no heap memory used.
	Memory:Memory Resident (byte)	Indicates the total memory that is used by process (RSS). For example, this memory metric is used by the OOM killer for memory constrained PODs.
	Memory:Memory Virtual (byte)	Indicates the total virtual memory size (VSS).
	Buffer Pool:Buffer Count	Indicates the count of buffers in the Java buffer pools (direct buffers, …).
	Buffer Pool:Buffer Memory Used	Indicates the total usage of all buffers.
	Buffer Pool:Buffer Total Capacity	Indicates the total capacity of all buffers.
	Storage:Disk Read (byte)	Indicates the total read bytes of the process.
	Storage:Disk Write (byte)	Indicates the total write bytes of the process.
Health	heartbeat:pulse	Indicates a 15-second pulse from RESTMon showing the active state of the current instance.
Scheduler	Scheduler|ProfilesProcessed:Profiles Processed	Indicates the number of incoming profiles that are seen in the RESTMon config file.
	Scheduler|ProfilesValidated:Profiles Validated	Indicates the number of incoming profiles that were seen and validated (schema validation).

DX OI Platform Telemetry Metrics

The following table lists different metrics that you can view in the RESTMon dashboards:

Path: *SuperDomain|Custom Metric Host (Virtual)|Custom Metric Process (Virtual)|Custom Metric Agent (Virtual)*

Metric Group	Metric Real Name	Description
Enterprise Manager	Metric Based Topology Metrics	Indicates the incoming topology metrics per interval.
	OI Alerts Supportability Metrics	Indicates the ingested OI alerts per interval.
	OI Inventory Supportability Metrics	Indicates the ingested inventory documents per interval.

Metric Group	Metric Real Name	Description
	OI MetaData Supportability Metrics	Indicates the incoming OI metadata per interval.

Custom Metrics

The following table lists different metrics that you can view in the RESTMon dashboards:

Metric Group	Metric Real Name	Metric Operation
DataCollector	URL Task List Size	Indicates the queued polling REST API calls to the third-party data source.
	Requests Completed Time (ms)	Indicates the average polling REST API call response time.
	Failed URL authentication request	Indicates the failed REST API authentication requests.
	Rest API Response Retries	Indicates the REST API polling call retries attempted.
	Rest API Call Response Exceptions	Indicates the REST API polling call exceptions.
JavaScriptEval	Javascript Expression Eval Time (ms)	Indicates the Javascript expression engine evaluation time.
Scheduler	ProfilesProcessed:Profiles Processed	Indicates the number of profiles seen.
	ProfilesValidated:Profiles Validated	Indicates the number of active profiles scheduled.
Database	Connection Pool Size	Indicates the database connection pool size.
DataStore	Data Table Map Size (Bytes)	Indicates the size of the URL index cache.
	Data Table Index Size (Bytes)	Indicates the schema URL index.
	DB Connection count	Indicates the active database connections.
Publisher	Published Alarms Per Interval	Indicates the published alarms per interval.
	Published Metrics Per Interval	Indicates the published metrics per interval.
	Published Topology Per Interval	Indicates the published topology per interval.
	Published Inventory Per Interval	Indicates the published inventory per interval.
	Published Alarms Data Per Interval	Indicates the published alarms data per interval.
	Published Metrics Data Per Interval	Indicates the published metrics data per interval.
	Published Topology Data Per Interval	Indicates the published topology data per interval.
	Published Inventory Data Per Interval	Indicates the published inventory data per interval.
	Published Alarms Processing Time (ms)	Indicates the processing time for the published alarms in milliseconds.
	Published Metrics Processing Time (ms)	Indicates the processing time for the published metrics in milliseconds.

Metric Group	Metric Real Name	Metric Operation
	Published Topology Processing Time (ms)	Indicates the processing time for the published topology in milliseconds.
	Published Inventory Processing Time (ms)	Indicates the processing time for the published inventory in milliseconds.
QueueManager	Active Publish Tasks	Indicates the active publish tasks.

Liveness and Readiness Check

Using the Liveness probe, you can monitor and manage applications that run as docker containers.

The Liveness state of the application indicates the active status of the application and the Readiness state indicates if the application is ready to accept the traffic or not. Using the Liveness and Readiness probes, you can monitor and manage applications that run as docker containers.

This section provides the following information:

Liveness Check

The Liveness probe checks the processing status versus the ingestion flow at regular intervals and updates this status. The probe checks if the data ingestion for the streaming profile is active and if the processing or publishing is completed within the configured time. If the processing is not completed within that time, the liveness of the application is disabled for the processing to complete. The application is back live when the processing is completed.

In the latest version of DX RESTMon, the attributes for the liveness are configured with the default values. You can edit the values by passing the attributes as arguments to suit your requirements. For more information, see the Liveness and Readiness Attributes section in this page.

The API endpoint for Liveness is `/restmon/api/actuator/health/liveness` .

- **UP**: This status indicates that the data is being processed without any issues.
  Response Code: 200
  Sample Response:
  ```
  {
     "status":"UP",
     "components":{
        "livenessProbe":{
           "status":"UP"
        }
     }
  }
  ```
- **OUT_OF_SERVICE.** This status indicates that the data processing has stopped.
  Response Code: 503
  Sample Response:
  ```
  {
     "status":"OUT_OF_SERVICE",
     "components":{
        "livenessProbe":{
           "status":"OUT_OF_SERVICE"
        }
     }
  }
  ```

When the status is **OUT_OF_SERVICE**, the pod is killed and is restarted. Upon restarting, the pod processes the data. The application status is marked as **OUT_OF_SERVICE** in the following cases:

- Database is not accessible
- Tomcat server is not up and running
- Application is ready, but the data processing of the streaming profiles is not happening even though the data ingestion continues
- Out of memory
- Maximum number of threads are created

Readiness Check

Readiness indicates if the application is ready to accept the traffic or not. In the DX RESTMon 2.1 version, the attributes for the readiness are configured with the default values. You can edit the values by passing the attributes as arguments to suit your requirements. For more information, see the Liveness and Readiness Attributes section in this page.

The API endpoint for Readiness is `/restmon/api/actuator/health/readiness` .

- **The Application Status is UP**. The **UP** status indicates that the application is processing without any issues and is ready to accept the traffic.
  Response Code: 200 (Success)
  Sample Response:
  ```
  {
     "status":"UP",
     "components":{
        "readinessProbe":{
           "status":"UP"
        }
     }
  }
  ```
- **The Application Status is OUT_OF_SERVICE**. The **OUT_OF_SERVICE** status indicates that the application is busy.
  Response Code: 503 (not ready)
  Sample Response:
  ```
  {
     "status":"OUT_OF_SERVICE",
     "components":{
        "readinessProbe":{
           "status":"OUT_OF_SERVICE"
        }
     }
  }
  ```
  If the application is busy with processing, the traffic that is being routed to the application is stopped for a while and the response code is displayed as **not ready**. The application is marked as **OUT_OF_SERVICE** in the following cases:
  - The DX OI endpoint is not ready or available.
    NOTE
    If the token is incorrect or the DX OI endpoint is typed wrong, the readiness is false. You cannot modify the endpoint against the running instance as Swagger is accessible. Restart the pod by updating the environment variables.
  - **Number of active profiles is high:** If the number of parallel running profiles count is more (as per the config file), then the readiness state would not be ready. The remaining profiles run is delayed and the run is initiated only after the run is completed.
  - **Queue size has reached the given threshold:** The number of messages in the steaming queue for any of the profiles is more than the configured number

When the application is in the **not ready** state:
- The Swagger UI is not accessible externally.
- The Health API is not accessible externally.
- Pod is running and processing the pending data.
- Further polling does not happen for the polling profiles.
- No data is routed to the pod until the pod is ready. There is no streamed in data against the streaming profiles.

When the traffic that is being routed to the application is stopped, further polling and processing start only after the application is back to the **ready** state. The pod status changes to **ready** when:
- The DX OI endpoints become accessible
- The number of the running profiles is within the limit
- The pending queue size becomes less than the minimum configured threshold for all the profiles

Liveness and Readiness Attributes

The liveness and readiness attributes are configured with the default values. To change these values, you can pass them as arguments during the deployment. For more information, see the Configurations section.

Schema Development Guide

In today's complex and hybrid landscapes, the IT teams are being tasked with the responsibility to maintain the business continuity with a minimal disruption. More often than not, these teams rely on a disparate set of specialized products to fulfill their objectives which can range from purpose-built market leading tools to homegrown custom applications – ultimately leading to operational silos, more noise, and increased MTTD and MTTR.

DX Operational Intelligence (DX OI) can consume both structured and un-structured data such as topology, metrics, traces, alarms, logs, groups from the heterogeneous sources to intelligently distill the signal from noise. The RESTMon connector enables DX OI to ingest data from third-party tools or services through REST APIs.

The RESTMon includes the following building blocks:

- **Profile:** A profile is a JSON document that contains the connection and polling-related details on which the schema works. Every product that RESTMon supports has its own profile.
- **Schema:** A schema is a JSON document that contains the API details and parsing logic that is written in JavaScript. You define the details or logic that is required parse the metrics, alarms, groups, and topology contents in the schema file. You also define or configure the urls from which data must be pulled and for further processing.

Schema File

The schema file is where you define the details or logic that is required to parse the metrics, alarms, groups, and topology contents.

Schema files also allow you to define or configure the urls from which data must be pulled for further processing.

This section provides the following information:

Supported Parsing Methods

The following parsing methods are supported:

Parsing Method	Format
JavaScript	`#(function() { var result = root.username; return result;})();`
JsonPath	`$['username']`

Parsing Method	Format
XPath	`/username`

Schema Types

You can use the following schemas types to get the data to the schema for parsing:

- **PULL Schema:** Using the PULL schema, you can pull data from other sources. The URLs section of the PULL schema defines how and where the data must be pulled from. You can also mention the type of request and response in the schema. Dynatrace, AppDynamics, Datadog, Solarwinds, and so on, are some examples of the PULL schemas.
  Sample Pull Schema Definition:
  ```
  {

  "definition": {
              "resource_category": null,
              "uploadedBy": "RESTMON",
              "updatedBy": "",
              "version": "2.0",
              "defaults": {
              "checkcert": "no",
              "port": "",
              "interval": 60,
              "httptimeout": 30000,
              "headers": {
                         "Authorization": "Api-Token %token"
              }
  },
  "addedProfileFields": [{
              "name": "token",
              "value": "",
              "type": "password",
              "label": "Token"
  }]
  }

  }
  ```
- **PUSH Schema**: In the PUSH schema, the outside data sources push data into RESTMon. The data is ingested to RESTMon through an API (***<RESTMon-hostname>:<port>/v1/logs***). The URLs section of the PUSH schema must be empty as DX RESTMon is not expecting to pull any data from the outside sources. Syslog, Google Cloud Monitoring, and so on, are some examples of the PUSH schemas.
  Sample Push Schema Definition:
  ```
  {

  "definition": {
                "resource_category": "",
                "auth": "",
                "xml_ns": "",
                "name": "",
                "type": ""
  }

  }
  ```

Basic Structure of Schema

A schema includes the definition, urls, and data category sections. The following snippet illustrates the basic structure of the schema file:

```
{
"schema_name":{

"definition"{},
"urls":[],
"topology":[],
"alarms":[],
"metrics":[],
"calculated_methods":[],
"calculated_metrics":[],
}
}
```

Section	Description
schema_name	Define the name of the data source or the monitoring tool.
definition	Define Schema related information such as description, and version.
urls	Define the list of REST endpoints that are used to gather metrics and other information.
Alarms, **Metrics**, **Topology**	Define the data to be collected in these sections.
calculated_methods	Use the Calculated methods in the conversion of metrics values from one unit to another.
calculated_metrics	Use Calculated Metrics that are user-defined metrics computed from other available metrics.

Best Practices

Consider the following points:

- To optimize the streaming performance, choose the *batch_size, batch_wait_time_milli* and *streaming_array_size* such that best results are obtained.
- You can expect better performance results when the payload for processing is in an array format, that is, is_array_input is "true".

definition

In the definitions section, you can define information such as schema description, version, uploadedBy, and updatedBy.

The following code block is a sample definition for the pull schema:

```
{
  "definition": {
    "resource_category": null,
    "uploadedBy": "RESTMON",
    "updatedBy": "",
    "version": "1.4",
    "defaults": {
      "checkcert": "no",
      "port": "",
      "interval": 60,
      "httptimeout": 30000,
      "headers": {
```

```
        "Authorization": "Api-Token %token"
      }
    },
    "addedProfileFields": [
      {
        "name": "token",
        "value": "",
        "type": "password",
        "label": "Token"
      }
    ]
  }
}
```

The following code block is a sample definition for push schema:

```
{
  "definition": {
    "resource_category": "",
    "name": ""
  }
}
```

The following table lists the attributes in the definition section:

Attribute	Attribute Type	Description
resource_category	String	Indicates the schema description.
uploadedBy	String	Indicates the name to be given for the initial upload. Default: RESTMON
updatedBy	String	Indicates the name to be given for all the version updates. Default: TENANT
version (Mandatory)	String	Indicates the version of the schema.
defaults.checkcert	Boolean	Creates an ssl socket factory for the update of OAuth access token. The values that are defined in the defaults of the schema are considered for processing if no value is provided in the profile section.
addedProfileFields	Json Array	Fields which can be passed to be used under the profile definition for processing.
name	String	Defines the schema name that can be specified in the definition.

urls

In the urls section, you can define the list of REST endpoints that are used to gather metrics and other information. For each URL that you want to monitor, create an entry and define the value in the url parameter.

Sample Snippets

The following snippets are samples of how to configure the urls section in the schema:

url Snippet for GET

```
{
  "xml_ns":  "",
  "src":  "",
  "var":  "",
  "id":  "<url-id>",
  "url":  "<url>"
}
```

url Snippet for POST

```
{
  "xml_ns":  "",
  "src":  "",
  "var":  "",
  "id":  "<url-id>",
  "url":  "<rest-endpoint>",
  "method":  "post",
  "body": "value"
}
```

Or

```
{
  "xml_ns":  "",
  "src":  "",
  "var":  "",
  "id":  "<url-id>",
  "url":  "<rest-endpoint>",
  "method":  "post",
  "body": {
    "key": "value"
  }
}
```

url Snippet for PUT

```
{
  "xml_ns": "",
  "src": "",
  "var": "",
  "id": "<url-id>",
  "url": "<rest-endpoint>",
  "method": "put",
  "body": {
    "key": "value"
  }
}
```

Or

```
{
  "xml_ns": "",
  "src": "",
  "var": "",
  "id": "<url-id>",
```

```
  "url": "<rest-endpoint>",
  "method": "put",
  "body": "value"
}
```

NOTE
POST and PUT support both JSON and String data body.

Example for JSON as body

```
{
  "xml_ns": "",
  "src": "domains",
  "var": "",
  "id": "currentdomain",
  "httpHeaders": {
    "Content-Type": "application/json"
  },
  "tokens": {
    "token": "$['token']"
  },
  "method": "put",
  "body": {
    "tenantId": "%tenantid",
    "domainId": "%domainid"
  },
  "url": "/ServicesAPI/API/V1/Session/CurrentDomain?token=%token"
}
```

Example for String as body

```
{
  "xml_ns": "",
  "src": "",
  "var": "",
  "id": "graphql",
  "method": "post",
  "url": "https://api.newrelic.com/graphql",
  "body": "{ actor { entitySearch(query: \"type = 'HOST'\") { results { entities { name ... on
 InfrastructureHostEntityOutline { name hostSummary { cpuUtilizationPercent } } } } } } }}"
}
```

The following table lists the attributes in the urls section:

Attribute	Attribute Type	Description
xml_ns (Mandatory If an xml document is used)	String	(Optional) Indicates the XML namespace to use when parsing the node information retrieved from the referenced url. This attribute is mandatory if the xml document is used.
src	String	(Optional) Indicates ID for a sibling url that contains the instance information that is needed for this url. Used with the **var** field.

Attribute	Attribute Type	Description

var	String	(Optional) Define the JPath, JavaScript, or XPath directive that is used with src value to parse the information that is returned by src endpoint and substitute it for the $var tokens in the url field. For example, in instances where a src url returns a list of hosts / or nodes / or volumes, where each instance value is targeted separately to get more detailed information. If you provide var, the corresponding src should be provided.
		Example:
		Javascript:
		`"var": "#(function(){ var result= []; for(var i=0;i<root.conditions.length; i++) { result.push(root.conditions[i].metric_name); } return result;})();"`
		Jpath:
		`"var":"$['conditions'][*]['metric_name']"`
		Xpath:
		`"var":"/conditions/*/metric_name"`

id (Mandatory)	String	Indicates the unique name for the url information.
url (Mandatory)	String	Indicates the REST endpoint that provides metric and other information.
formParameters(Mandatory for Http POST/PUT requests. Optional for GET.)	String	You can add additional form parameters based on the required data. The form-data can be sent as URL variables as the HTTP post transaction (with method="post"). • Appends the form-data inside the body of the HTTP request (data is not shown in URL) • Has no size limitations **Example:** • **id** : "searchID" (Optional. If not provided, Splunk assigns the id.) • **search:** The search criteria to be used to fetch the results. For example, `"search index=main source=\"http:LogsOverHttp \" sourcetype=\"aws:lambda\" logGroup \"/aws/elasticbeanstalk/SYMC-WSSE-1AI88EJ8S5QR3/var/log/nginx/error.log\" | rex field=_raw \"'logEvents':(?<logevents>.*?)(?='logGroupName')\" | rex field=logevents max_match=999999 \"'message':(?<message>.*?)(?='extractedFields')\"')` • **earliest_time:** The data to be fetched since x minutes. • **latest_time:** now
method	String	Defines the type of request to be made. Supported methods: PUT, POST, GET Default: GET Example: "method":"put"

Attribute	Attribute Type	Description

tokens	JSON	Values which can be used in different blocks of the schema. Tokens such as timestamp, url, interval are already defined in the schema. Also custom tokens can be generated from the output of any url requests. These tokens can be used as: **%<token-name>**
		Example for custom token:
		`"tokens":{`
		`"domainid":"$['domains'][0]['domainId']"`
		`}`
		
		`Example for how to use this token:`
		`"domain":"%domainid",`
		`or`
		`"timestamp":"%timestamp",`

body	String or JSON	Defines the body of the request and can be either JSON or String. Used for PUT or POST requests.
		For Example:
		`"body":{`
		`"tenantId":"%tenantid",`
		`"domainId":"%domainid"`
		`}`

httpHeaders	JSON	Specific headers that are required for the request can be mentioned here.
		For Example:
		`"httpHeaders":{`
		`"Content-Type":"application/json"`
		`}`

Multi-Variables Support

You can use **var** from the non-parent url block to build the current url. You can pass the url block id to *multiVar[<id>]* .

For Example: In this example, **%multiVar[metrics]** and **%multiVar[metrics-path]** are the multiVar used and they will have the var values generated for the url block with iD "metrics" and "metrics-path" respectively.

```
{
  "urls": [
    {
      "src": "",
      "xml_ns": "",
      "var": "",
      "id": "applications",
      "url": "/controller/rest/applications?output=json"
    },
    {
      "src": "applications",
      "xml_ns": "",
      "var": "$[*].name",
      "id": "metrics",
      "url": "/controller/rest/applications/%var/metrics?output=json"
    },
    {
```

```
      "src": "metrics",
      "xml_ns": "",
      "var": "$[*].name",
      "id": "metrics-path",
      "url": "/controller/rest/applications/%multiVar[metrics]/metrics?metric-path=%var&time-range-
type=BEFORE_NOW&duration-in-mins=60&output=json"
    },
    {
      "src": "metrics-path",
      "xml_ns": "",
      "var": "$[*].name",
      "id": "metrics-path-agents",
      "url": "/controller/rest/applications/%multiVar[metrics]/metrics?metric-path=%multiVar[metrics-path]|
%var&time-range-type=BEFORE_NOW&duration-in-mins=60&output=json"
    }
  ]
}
```

alarms

In the alarms section of the schema file, you can define attributes for the alarms that are received by DX Operational Intelligence

The following snippet is the sample schema for AppDynamics with the alarms section configured:

The following table lists the attributes that can be defined for this mapping:

```
{
    "appdynamics": {
        "definition": {
            "resource_category": null,
            "uploadedBy": "RESTMON",
            "updatedBy": "",
            "version": "2.0",
            "defaults": {
                "port": 443,
                "interval": 60,
                "httptimeout": 30000
            },
            "auth": "basic",
            "xml_ns": "",
            "name": "appdynamics",
            "type": "https"
        },
        "urls": [],
        "topology": [],
        "alarms": [{
            "xml_ns": "",
            "url": "applicationalarms",
            "group": "Applications",
            "attributes": {
                "oi": {
                    "timestamp": "%timestamp",
                    "startTime": "$[?(@.affectedEntityDefinition.entityType ==
 'BUSINESS_TRANSACTION')].startTimeInMillis",
```

```
                    "host": "#(function() {return url.split('/')[6]})();",
                    "product": "AppDynamics",
                    "product_version": "20.7.0-2824",
                    "summary": "Alarms",
                    "severity": "$[?(@.affectedEntityDefinition.entityType ==
 'BUSINESS_TRANSACTION')].severity",
                    "severity_conversion": "WARNING:Minor,CRITICAL:Critical,INFO:Information,Default:",
                    "metric_name": "",
                    "metric_type": "",
                    "configuration_item_type": "Applications.Tiers.Nodes.BusinessTransactions",
                    "configuration_item": "#(function() {var result = []; for(var
 i=0;i<root.length;i++){ if(root[i].affectedEntityDefinition.entityType == 'BUSINESS_TRANSACTION')
 { result.push(root[i].affectedEntityDefinition.name); } } return result; })();",
                    "ci_unique_id": "#(function() {var result = [];var values='%%BTTokenAlarm%%';var
 valuesArrya=values.split(';');for(var i=0;i<root.length;i++){ if(root[i].affectedEntityDefinition.entityType
 == 'BUSINESS_TRANSACTION') {for(var k=0;k<valuesArrya.length;k++){if(valuesArrya[k].split('|')
[1].includes(root[i].affectedEntityDefinition.entityId+'')){result.push('Apps|'+url.split('/')
[6]+'|tier|'+valuesArrya[k].split('|')[0]);}}  } } return result; })();%//%#(function()
 {var result = []; var searchString='violating<br>For Node <b>'; for(var i=0;i<root.length;i
++) { if(root[i].affectedEntityDefinition.entityType == 'BUSINESS_TRANSACTION') { var
 alarmMessage=root[i].description; if(alarmMessage.includes(searchString)){var tempStr =
 alarmMessage.substring(alarmMessage.indexOf(searchString)).substring(searchString.length);
 tempStr=tempStr.substring(0,tempStr.indexOf('</b>')); result.push('|node|'+tempStr+'|
businessTransaction|'+root[i].affectedEntityDefinition.name);} else{result.push('');} } }return result;})();",
                    "message": "#(function() {var result = []; for(var i=0;i<root.length;i+
+) { if(root[i].affectedEntityDefinition.entityType == 'BUSINESS_TRANSACTION') { var temp =
 root[i].description.replace(/(<([^>]+)>)/ig,' ').replace(/%/g,'Percent'); result.push(temp); } }return
 result;})();",
                    "alarm_unique_id": "$[?(@.affectedEntityDefinition.entityType ==
 'BUSINESS_TRANSACTION')].id",
                    "status": "#(function(){var result = [];for(var i=0;i<root.length;i++)
{ if(root[i].affectedEntityDefinition.entityType == 'BUSINESS_TRANSACTION')if(root[i].incidentStatus ==
 'RESOLVED' || root[i].incidentStatus == 'CANCELLED'){result.push('Closed');}else{result.push('');}}return
 result;})();",
                    "tags": ["AppDynamics", "Events"],
                    "alarmURL": "$[?(@.affectedEntityDefinition.entityType ==
 'BUSINESS_TRANSACTION')].deepLinkUrl",
                    "alarmType": "APPLICATION"
                }
            },
            "fieldId": "FCED59A3523F482C82ED9309C59A88F4"
        }],
        "metrics": [],
        "calculated_methods": [],
        "calculated_metrics": [],
        "groups": []
    }
```

}

Attributes	Mandatory	Attribute Type	Description
summary	No	String	Indicates the information about the alarm.
severity	Yes	String	Indicates the severity level which defines how serious the state or event is. This attribute is used for **severity_conversion** mentioned in this section.
severity_conversion	Yes	String	The **severity_conversion** is mapped based on the severity level. The mapping is given as *<Severity>:<Conversion-Value>*. For Example: "severity_conversion": "Low: Minor, Notice: Major, Error: Critical, Default: Information" In this example, based on the severity value("Low","Notice","Error"), the alarm severity is converted as ("Minor","Major","Critical") respectively. If no severity value is given, the value that is mapped to Default is taken as the alarm severity.
alarm_unique_id	Yes	String	Indicates a unique value identifying the alarm.
create_alarm_condition	No	Boolean(String)	Creates an alarm based on the condition. The condition is executed if the condition value is "**true**". If no mapping is available, then it is defaulted as true. For Example: "create_alarm_condition" : "#(function() { var results =[]; for (var i=0;i<root.length;i++){var result = false; if (root[i].syslog_message.toLowerCase().includes('removed session')) {result = true;results.push(result);}}return results;})();"
close_alarm_condition	No	Boolean(String)	Closes an alarm based on the condition. The condition is executed if the condition value is "true". If no mapping is available, then it is defaulted as false. For Example: "close_alarm_condition" : "#(function() { var results =[]; for (var i=0;i<root.length;i++){var result = false; if (root[i].syslog_message.toLowerCase().includes('new session')) {result = true;results.push(result);}}return results;})();"
timestamp	No	String	Indicates the timestamp of the alarm occurrence.
startTime	No	Long(ms)	Represents the start time of the alarm. Takes the default system time if not present.
condition	No	Boolean(String)	Used internally by RESTMon to filter the alarm data ingestion based on some condition. This value should derive to a boolean value (true/false) and the data is ingested only if the value is "**true**".
message	Yes	String	Indicates the message content for the alarm.
configuration_item_type	No	String	Indicates the CI type of the entity against which the alarm is raised.

Attributes	Mandatory	Attribute Type	Description
configuration_item	No	String	Indicates the CI of the entity against which alarm is raised.
metric_name	No	String	Used to map an alarm with an existing metric section. **This attribute is mandatory if the alarm-metric mapping is required.**
metric_type	No	String	Used to map an alarm with an existing metric section. **This attribute is mandatory if the alarm-metric mapping is required.**
ci_unique_id	Yes	String	Unique value identifying the **configuration_item** being monitored.
tags	No	JSON Array	Indicates the label that is attached for identification.
status	No	String	Indicates the alarm status: new, updated, or closed. No other status is accepted. Default status: if not mentioned already, the status is new. Else, the status is updated if any update is required.
host	Yes	String	Indicates the hostname of the device where the alarm occurs.
ip	No	String	Indicates the IP of the device where the alarm occurs.
alarmType	Yes	String	Indicates the type of alarm. For Example: "alarmType": "Security"
alarmUrl	No	String	Indicates the Url for the specific alarm if any.
fuzzy_match	No	String	Indicates the fuzzy pattern matching applied on the combined message.
message_merge_window_id	No	String	Indicates the fuzzy string, message merging is done based on the defined "message_merge_window_id".

NOTE
For alarm-metric mapping, metric_name, metric_type, configuration_item and configuration_item_type should be same as defined in metrics.

ChangeEvents as Alarms

The sample alarm structure to create changeevents is given here:

```
{
  "xml_ns": "",
  "url": "changestable",
  "group": "Alarms",
  "attributes": {
    "oi": {
      "timestamp": "#(function() { var result=[];var tz='PDT'; for(var i in root.result) { var temp=new
Date(root.result[i].work_end.display_value + ' ' + tz); result.push(temp.toJSON()) } return result; })();",
      "startTime": "#(function() { var result=[];var tz='PDT'; for(var i in root.result) { var temp=new
Date(root.result[i].work_start.display_value + ' ' + tz); result.push(temp.toJSON()) } return result; })();",
      "alarmType": "Change",
```

```
      "host": "#(function() {var regexHostname = new RegExp(/^[a-zA-Z0-9][a-zA-Z0-9-]{1,61}[a-zA-
Z0-9](?:\\.[a-zA-Z]{2,})+$/);var data=[]; for(var i=0;i<root.result.length;i++) {var isHostname =
 regexHostname.exec(root.result[i].cmdb_ci.display_value);if (isHostname === null) {data.push('')} else
 {data.push(root.result[i].cmdb_ci.display_value)}} return data;})();",
      "ip": "",
      "product": "Snow",
      "product_version": "1.0.0",
      "summary": "$['result'][*]['number']['display_value']%//%: %//%$['result'][*]['short_description']
['display_value']",
      "severity": "#(function() {var data=[]; for(var i=0;i<root.result.length;i++) { if
 (root.result[i].priority.value == '') {data.push(5)} else {data.push(root.result[i].priority.value)}} return
 data;})();",
      "severity_conversion": "1:Critical,2:Major,3:Minor,Default:Information",
      "metric_name": "",
      "metric_type": "",
      "configuration_item": "#(function() {var regexHostname = new RegExp(/^[a-zA-Z0-9][a-zA-Z0-9-]{1,61}
[a-zA-Z0-9](?:\\.[a-zA-Z]{2,})+$/);var data=[]; for(var i=0;i<root.result.length;i++) {var isHostname =
 regexHostname.exec(root.result[i].cmdb_ci.display_value);if (isHostname === null) {data.push('')} else
 {data.push(root.result[i].cmdb_ci.display_value)}} return data;})();",
      "configuration_item_type": "Snow.Host",
      "ci_unique_id": "#(function() {var regexHostname = new RegExp(/^[a-zA-Z0-9][a-zA-Z0-9-]{1,61}[a-
zA-Z0-9](?:\\.[a-zA-Z]{2,})+$/);var data=[]; for(var i=0;i<root.result.length;i++) {var isHostname =
 regexHostname.exec(root.result[i].cmdb_ci.display_value);if (isHostname === null) {data.push('')} else
 {data.push(root.result[i].cmdb_ci.display_value)}} return data;})();",
      "message": "$['result'][*]['number']['display_value']%//%: %//%$['result'][*]['short_description']
['display_value']",
      "alarmURL": "%type%//%://%//%%hostname%//%:%//%%port%//%/nav_to.do?uri=change_request.do?sys_id=%//%
$['result'][*]['sys_id']['display_value']",
      "alarm_unique_id": "$['result'][*]['number']['display_value']",
      "tags": [
        "Snow",
        "Alarms",
        "Changes"
      ]
    }
  }
}
```

metrics

In the metrics section of the schema file, you can configure the metric collection attributes.

The following snippet is the AppDynamics schema sample to collect metrics:

```
{
    "appdynamics": {
        "definition": {
            "resource_category": null,
            "uploadedBy": "RESTMON",
            "updatedBy": "",
            "version": "2.0",
            "defaults": {
                "port": 443,
                "interval": 60,
                "httptimeout": 30000
```

```
            },
            "auth": "basic",
            "xml_ns": "",
            "name": "appdynamics",
            "type": "https"
        },
        "urls": [],
        "topology": [],
        "alarms": [],
        "metrics": [{
            "attributes": {
                "oi": {
                    "metric_name": "#(function() {if(root[0]){var length=root[0].metricName.split('%%pipe%
%').length; return root[0].metricName.split('%%pipe%%')[length-1]}})();",
                    "metric_type": "JMX",
                    "metric_unique_id": "$[*].metricId",
                    "metric_unit": "",
                    "timestamp": "$[*]['metricValues'][*]['startTimeInMillis']",
                    "configuration_item": "#(function() {if(root[0]){var res=root[0].metricPath.split('%%pipe
%%');var result='';  for(var i=0;i<res.length-1;i++){result=result+res[i]+'%%pipe%%';} return
 result.substr(0,result.length-1)}})();",
                    "configuration_item_type": "#(function() {if(root[0]){var res=root[0].metricPath.split('%
%pipe%%');var result='';  for(var i=0;i<res.length-2;i++){result=result+res[i]+'%%pipe%%';} return
 result.substr(0,result.length-1)}})();",
                    "host": "#(function() {var url='%%currentURL%%';return url.split('/')[4]})();",
                    "product": "AppDynamics",
                    "type": "2",
                    "product_version": "20.7.0-2824",
                    "ci_unique_id": "#(function(){var finalResult='Apps|';var
 result=url.split('/');finalResult+=result[6]+'|tier|';var tierName=result[7].split('|');finalResult
+=tierName[1];return finalResult;})();",
                    "tags": ["AppDynamics", "JMX", "Application Infrastructure Performance", "JMX Metrics"],
                    "datapointfrequencyinsec": "60"
                }
            },
            "value": "$[*]['metricValues'][*]['value']",
            "url": "jmxmetrics",
            "group": "JMX",
            "fieldId": "DC5054F8B5E2477FB254909CDFD0FD6D"
        }],
        "calculated_methods": [],
        "calculated_metrics": [],
        "groups": []
    }
}
```

The following table lists the attributes that can be defined for this mapping:

Attribute	Mandatory	Attribute Type	Description
type	No	Integer (String)	Defines the type of metric. Some of the type values are: 1: Integer, 2 : Long, 5 : String
timestamp	No	String	Indicates the time of occurrence of a particular metric event. If Current timestamp will be taken if non provided.
metric_name	Yes	String	Indicates the name describing the metric element. RESTMon displays the metrics in the following hierarchy: `host|metric type|ci:metric name` . These four attributes are mandatory when defining the metric mappings in any schema. RESTMon uses a placeholder text if you do not map any of these four attributes per the given hierarchy.
metric_type	No	String	Indicates the type of the metric element. RESTMon displays the metrics in the following hierarchy: `host|metric type|ci:metric name` . These four attributes are mandatory when defining the metric mappings in any schema. RESTMon uses a placeholder text if you do not map any of these four attributes per the given hierarchy.
configuration_item_type	Yes (If entity being montiored is other than the host)	String	Indicates the arrangement of the metrics that are displayed based on type.
metric_unit	Yes	String	Indicates the measuring unit of the metric.
configuration_item	Yes (If entity being monitored is anything other than the host.)	String	Indicates the arrangement of the metrics display.
ci_unique_id	Yes	String	Indicates the unique value identifying the configuration item or the metric element. RESTMon displays the metrics in the following hierarchy: `host|metric type|ci:metric name` . These four attributes are mandatory when defining the metric mappings in any schema. RESTMon uses a placeholder text if you do not map any of these four attributes per the given hierarchy.
product	Yes	String	Indicates the product name for which data is being ingested.
product_version	No	String	Indicates the version of the product.
tags	No	JSON array	Indicates the label that is attached for identification.
value	Yes	String	Indicates the value of the metric.
url	Yes (For the push schema, this attribute can be empty)	String	To map the url-id with the url section id tagged with the metric url.
group	Yes	String	Indicates the name of the monitored group.

Attribute	Mandatory	Attribute Type	Description
entity_id	No	String	Indicates the entity ID which helps to map with the device entity stored before.
derivehostnamefromentityid	No	Boolean (String)	Helps in deriving the hostname from entity_id, as the same is stored during device (topology) publish. Default is false, if not given.
datapointfrequencyinsec	No	Integer (String)	Indicates the interval/frequency of the metric captured at the data source that is used for OI ingestion. Example : "datapointfrequencyinsec": "60"
includetimestampidentifier	No	Boolean (String)	Publishes all the samples, if this attribute is set to true. By Default, RESTMon publishes only the last value if if there are multiple samples for the same metric in the same polling cycle.
host	Yes(If derivehostnamefromentityid is false)	String	Indicates the host name of the device. RESTMon displays the metrics in the following hierarchy: `host|metric type|ci:metric name` . These four attributes are mandatory when defining the metric mappings in any schema. RESTMon uses a placeholder text if you do not map any of these four attributes per the given hierarchy.
calculation	No	String	Used to convert the value generated with the help of the calculated methods block. Examples on how to use this attribute are given later in this page.
performAggregation	No	Boolean (String)	When this flag is available and the flag is "**true**", sum of the metric values is derived based on the **metric_content_id (ci_unique_id+":"+metric_name+":"+host+ ":"+configuration_item)**. The Schema example is given under the **Metric Aggregation** section.
message	No	String	Indicates the message for the alarm to be generated.
severity_conversion	No	String	Indicates the severity conversion that is used for generating the alarm. This conversion is based on the aggregated metric value. Severity range can be mentioned using the aggregated value. **Format:** <The range>:<Severity level> **Example:** "severity_conversion": ">100:Critical,20-80:Major,10-19:Minor,Default:Information"
alarm_unique_id	No	String	Indicates the unique identifier for the alarm.
generateAlarm	No	Boolean (String)	Indicates the Boolean value which is required if the alarm generation must happen. The alarm is generated only for **"generateAlarm": "true"**.
close_alarm_condition	No	Integer (String)	Closes the created alarm if the severity value is less than or equal to the value given in close_alarm_condition.
alarmType	No	String	Specifies the type of alarm generated. For Example: "alarmType": "Log"

You can change the metric value as per the requirement. You can use **Calculated Methods** or **Calculated Metrics** to change the metric value.

Calculated Methods

Calculated Methods help in the conversion of metrics values from one unit to another. You can mention the methods under the **calculation** field in the **metrics** section. For example,

```
{
  "calculated_methods": [
  {
    "convertBytestoGB": "/ 1073741824;"
  },
  {
    "convertBytestoBits": "/ 125;"
  },
  {
    "convertBytestoMB": "/ 1048576;"
  },
  {
    "convertBitsToKiloBits": "/ 1000;"
  }
  ]
}
```

You can use these values in the **metrics** section as shown. **calculation** has the **value** field and the **calculated method** field. In the following example, the value is converted from Bytes to GB.

```
{
  "metrics": [
  {
  "calculation": "$value $convertBytestoGB",
  "attributes": {
  "oi": {
    "type": "2",
    "timestamp": "#(function(){var result=[];for(var i=0;i<root.result.length;i++)
{ if(root.result[i].metricId=='builtin:host.mem.recl'){ var check=root.result[i]; for(var
 k=0;k<check.data.length;k++){ result.push(new Date(check.data[k].timestamps[0]).toISOString()); }}} return
 result;})();",
    "metric_name": "Memory Reclaimable",
    "metric_type": "Memory",
    "configuration_item_type": "Host",
    "metric_unit": "GB",
    "configuration_item": "",
    "entity_id": "#(function(){var result=[];for(var i=0;i<root.result.length;i++)
{ if(root.result[i].metricId=='builtin:host.mem.recl'){ var check=root.result[i]; for(var
 k=0;k<check.data.length;k++){ result.push(check.data[k].dimensions[0]); }}} return result;})();",
    "host": "",
    "ip": "",
    "ci_unique_id": "#(function(){var result=[];for(var i=0;i<root.result.length;i++)
{ if(root.result[i].metricId=='builtin:host.mem.recl'){ var check=root.result[i]; for(var
 k=0;k<check.data.length;k++){ result.push(check.data[k].dimensions[0]); }}} return result;})();",
    "product": "Dynatrace",
    "product_version": "1.0.0",
    "derivehostnamefromentityid": "true",
```

```
    "tags": [
      "Dynatrace",
      "Page Faults"
    ]
    }
  },
  "value": "#(function(){var result=[];for(var i=0;i<root.result.length;i++)
{ if(root.result[i].metricId=='builtin:host.mem.recl'){ var check=root.result[i]; for(var
 k=0;k<check.data.length;k++){ var s=(check.data[k].values[0]).toFixed(1);result.push(s); }}} return result;})
();",
  "url": "host-disk-packets-pfps",
  "condition": "#(function(){var result = '';if(root){return true}else{return false;}})();",
  "group": "Hosts"
  }
  ]
}
```

Calculated Metrics

Calculated Metrics are user-defined metrics that are computed from multiple other metrics available. For example,

```
{
  "calculated_metrics": [
  {
    "calculation": "($total_provisioned - $total_free) / 1048576",
    "xml_ns": "",
    "values": [
      {
        "name": "$total_free",
        "value": "$['totalFree_gb']"
      },
     {
       "name": "$total_provisioned",
       "value": "$['totalProvisioned_gb']"
     }
    ],
  "group": "CapacityMetrics",
  "attributes": {
  "oi": {
    "metric_name": "Total Used",
    "metric_type": "Cluster Capacity",
    "metric_unique_id": "%clusterName%//%:%//%TotalUsed",
    "metric_unit": "PB",
    "type": "1",
    "configuration_item_type": "ECS.Capacity",
    "configuration_item": "%clusterName",
    "ci_unique_id": "%clusterName",
    "host": "%clusterName",
    "ip": "",
    "product": "ECS",
    "product_version": "1.17",
    "tags": [
        "ECS",
```

```
        "Total Used",
        "Capacity: Total Used",
        "%clusterName"
    ]
   }
  },
  "url": "capacity"
  }
  ],
}
```

Metric Aggregation

To aggregate the counters for the values, the metric mapping has the field **"performAggregation" : "true"** as shown. When this flag is available and is **"true"**, sum of the metric values is derived based on the **metric_content_id (ci_unique_id+":"+metric_name+":"+host+ ":"+configuration_item)**

```
"metrics": [{
               "xml_ns": "",
               "url": "searchresults",
               "group": "Search",
               "performAggregation" : "true",
               "attributes": {
                   "oi": {
                       "type": "2",
                       "metric_name": "No Such File or Directory Counter",
                       "metric_type": "Counter Metrics",
                       "configuration_item_type": "$['results'][*]['sourcetype']",
                       "metric_unique_id": "$.legends.rows[*].id%//%:%//%$.legends.rows[*].performanceobject
%//%:%//%$.legends.rows[*].counter",
                       "metric_unit": "",
                       "configuration_item": "$['results'][*]['source']",
                       "host": "$['results'][*]['host']",
                       "ip": "",
                       "product": "Splunk",
                       "ci_unique_id": "$['results'][*]['host']",
                       "product_version": "1.2",
                       "message": "The No Such File or Directory value <metric_value> is greater than the
 threshold value",
                       "severity_conversion": ">100:Critical,20-80:Major,10-19:Minor,Default:Information",
                       "alarm_unique_id": "$['results'][*]['host']%//%:%//%this is my alarm for No Such File
 or Directory",
                       "generateAlarm": "true",
                       "alarmType": "Log"
                   }
               },
               "value": "#(function(){var counts=[]; for (var i = 0; i<root.results.length;i++) { var message
 = root.results[i].message; var count = (message.toString().match(/No such file or directory/g) || []).length;
 counts.push(count); } return counts;})();"
           }
       ]
```

.

Custom Alarms

Based on the metric counter derived in the above example, an alarm can be generated with the help of following fields:

- *"message": "The No Such File or Directory value <metric_value> is greater than the threshold value",*
- *"severity_conversion": ">100:Critical,20-80:Major,10-19:Minor,Default:Information",*
- *"alarm_unique_id": "$['results'][*]['host']%//%:%//%this is my alarm for No Such File or Directory",*
- *"generateAlarm": "true",*
- *"alarmType": "Log"*

In the above example, if the aggregated value is greater than 100, then the severity is *Critical.* If the aggregated value is in between 20-80, then Major. The default severity can be mentioned under Default.

topology

In the topology section of the schema file, you can configure topology elements for the TAS model in DX Operational Intelligence.

To create a service and to utilize the analytics capabilities of the AIOps platform, the entities that are being monitored by the monitoring tool like host machines, critical servers, applications, processes, services, routers, switches, firewalls, AD Servers, Volumes, Storage devices must be ingested to DX Operational Intelligence. The collection of these entities and the relationship between them is represented as a graph (TAS model) for the given monitored product.

By default, the payload that is received is assumed to have the complete topology. RESTMon calculates the delta topology by comparing the vertices or edges that are derived from the current payload with the already published topology. Only this delta data is published to DX OI/TAS for every polling interval. For example,

- Assume id1, id2 are the vertices that are ingested from the first run.
  - If the payload from the second run contains "id1" and "id2", then no topology is ingested to DX OI.
  - If the payload from the third run contains "id1", "id2" and "id3", then only "id3" is ingested to DX OI.
  - If the payload from the fourth run contains "id2" and "id3", then the vertex "id1" is marked for deletion by mapping the "endtime": "-1" and is ingested to DX OI.
- Full synchronization of the topology happens based on the full sync frequency that is given in the profile, which is 24 hours by default.

In the topology section, specify the following details to define the topology mapping for any monitoring application. The group information is ingested into the TAS store in the CUSTOM layer. The following sample code is a snapshot of the topology mapping.

```
{
    "appdynamics": {
        "definition": {
            "resource_category": null,
            "uploadedBy": "RESTMON",
            "updatedBy": "",
            "version": "2.0",
            "defaults": {
                "port": 443,
                "interval": 60,
                "httptimeout": 30000
            },
            "auth": "basic",
            "xml_ns": "",
            "name": "appdynamics",
            "type": "https"
        },
        "urls": [],
```

```
        "topology": [{
                "xml_ns": "",
                "url": "applications",
                "group": "Applications",
                "layer": "CUSTOM",
                "attributes": {
                    "oi": {
                        "name": "$[*].name",
                        "entity_id": "$[*].id",
                        "entity_name": "$[*].name",
                        "type": "Application",
                        "product": "AppDynamics",
                        "ci_unique_id": "$[*].name",
                        "hostname": "$[*].name",
                        "applicationname": "$[*].name",
                        "hasentityidnamedetails": "true",
                        "globalContext": "true"
                    }
                }
            }
        ],
        "alarms": [],
        "metrics": [],
        "calculated_methods": [],
        "calculated_metrics": [],
        "groups": []
    }
}
```

The following table lists the attributes that you can define for this mapping:

Attribute Name	Mandatory	Attribute Type	Description
ci_unique_id	Yes	String	Indicates the unique value identifying the configuration_item or topology element being monitored. Used in creating the externalID (the unique identifier in TAS) for the monitored entity.
name	Yes	String	Indicates the name of the topology entity.

Attribute Name	Mandatory	Attribute Type	Description
type	Yes	String	Indicates the type of the topology entity. Reference types that are supported by TAS are: *EXTERNAL*, *BUSINESSTRANSACTION*, *EJB*, *EJBCLIENT*, *DATABASE*, *DATABASE_SOCKET*, *SOCKET*, *WEBSERVICE*, *WEBSERVICE_SERVER*, *JMSSERVER*, *SERVLET*, *TRANSACTION_PROCESSOR*, *GENERICFRONTEND*, *GENERICBACKEND*, *DEFAULT*, *AUTOMATICENTRYPOINT*, *APPLICATION_ENTRYPOINT*, *INFERRED_DATABASE*, *INFERRED_SOCKET*, *INFERRED_WEBSERVICE*, *INFERRED_GENERICBACKEND*, *ENTERPRISE_MANAGER*, *EM_MASTER*, *EM_PROVIDER*, *EM_MOM*, *EM_COLLECTOR*, *APM_SAAS*, *AGENT*, *EM_DATABASE*, *DATABASE_SERVER*, *AGENT_CONNECTION*, *EM_CONNECTION*, *EMDB_CONNECTION*, *STRUTS*, *HOST*, *EXPRESSJS*, *TIBCOPROCESS*, *DOCKER*, *OPENSHIFT*, *KUBERNETES*, *SPRINGCONTROLLER*, *SPRINGSERVICE*, *MESSAGE_QUEUE*,*SPRINGASYNC*,*SPR* **Not Allowed**: SERVICE
product	Yes	String	Indicates the product name from which the data is being ingested.
applicationname	No	String	Indicates the application name. Use this attribute if the topology is specific to the application entity or the topology element is mapping to any application.

Attribute Name	Mandatory	Attribute Type	Description
hasapplicationdetails	No	String (Boolean)	Used internally by RESTMon. Similar to hostname, if the URL payload used does not contain the application name for all the entities, then you can identify the details from the other URL entities. The mapping with the availability of the application be marked as "hasapplicationdetails: true". Other parent/child entities get the application name attached based on the mapping provided for parent_type/type and entity_id.
hostname	Yes	JSON array	Maps to the hostname on which the entity is being deployed. For example, FQDN name, non-FQDN name, and any other aliases that source products collect. **Note:** If the payload used for the entity does not have the hostname available with the given payload, instead has the parent ci_unique_id to which this entity is related. In this case, the hostname can be mapped to an empty string or removed from the schema mapping, and based on the configured parent_ci_unique_id the hostname will be derived (assuming the vertex with the hostname has configured "hashostdetails":"true").
FQDNHostname	No (Yes, for the device inventory to help in the host correlation.)	String	Indicates the fully qualified hostname of the device.
hashostdetails	N/A	N/A	Used internally by RESTMon. This attribute has to be marked as "**true**" if that particular element is the parent of any other element and has the hostname details.
ipAddresses	No (Yes, for the device inventory to help in the host correlation.)	JSON array	List of the IP addresses that the hostname of the topology element maps to. Should be a JSON array with all the IP addresses of that host or device.
macAddresses	No (Yes, for the device inventory to help in the host correlation.)	JSON array	Indicates all the mac addresses of that host or device.

Attribute Name	Mandatory	Attribute Type	Description
parent_ci_unique_id	Yes, if a relationship is to be created between the two vertices.	N/A	Used internally by RESTMon. • Used to represent the parent entity against which the edge is created. • Edge is created from parent_ci_unique_id to ci_unique_id. • To define more than one parent_ci_unique_id in a single mapping, use separator "&&". For example: "parent_ci_unique_id" : "id1&&id2&&id3"
parent_type	N/A	N/A	Used internally by RESTMon. • Used to create the relationship edge between two elements that are defined by ci_unique_id and a combination of parent_ci_unique_id/ parent_type.
semantic	No	String	• The attribute that is used for the edge creation in attaching the proper relationship between two vertices represented by ci_unique_id and parent_ci_unique_id. • Default is null, which marks the direct from/to relationship between two vertices. • If more than one parent_ci_unique_id is defined in a single mapping, define semantic as well with a matching number that is separated by &&. **Valid values:** null, contains For Example: "semantic": "contains"

Attribute Name	Mandatory	Attribute Type	Description
child_ci_unique_id	Yes, if a relationship to be created between 2 vertices	N/A	Used internally by RESTMon. • If an entity exists that is a child of the current vertex definition (ci_unique_id), that can be defined against child_ci_unique_id. • Used in creating the parent-child edges/relationships, ci_unique_id being the "from" vertex and child_ci_unique_id being the "to" vertex. • To define more than one child_ci_unique_id in a single mapping, use the separator "&&".
semantic_child	No	N/A	Used internally by RESTMon. • The attribute that is used for the edge creation in attaching the proper relationship between the current vertex and its child vertex that is represented as ci_unique_id and child_ci_unique_id. • **Valid values:** null, contains • If more than one child_ci_unique_id is defined in a single mapping, define semantic_child as well with the matching number that is separated by &&. For Example: "semantic_child": "contains"
create_edges_with_childs_parents_of_other_type	No	N/A (true/false)	Used internally by RESTMon. Creates edge with its child's parent vertex. In case you are unable to get the relational data between the current vertex's child's parent, use this flag to create an edge between these vertices. For Example: In the case of Dynatrace, we have the relationship available between Host → process, process → service, and service → application from various topology APIs. You can use this flag when you need a relationship between service and host.

Attribute Name	Mandatory	Attribute Type	Description
child_parent_type	No	N/A	Used internally by RESTMon. Used in relation with the above attribute, "create_edges_with_childs_parents_of_oth to represent the type of the child's parent.
condition	No	N/A	Used internally by RESTMon to filter the topology data ingestion based on some conditions. This value should derive to a Boolean value (true/false), and the data is ingested only if the value is "true".
entity_id	No	String	Required by RESTMon, if the parent-child relationship is maintained by this unique id and the hostname has to be derived using this. Refer to the **Topology Mapping with both entity_id and entity_name details** section for sample mapping.
entity_name	No	String	Required by RESTMon, if all the topology APIs do not have the entity name and instead have only entity_id and only one of the APIs has the entity_id and entity_name mapping. Refer to **Topology Mapping with both entity_id and entity_name details** section for sample mapping.
hasentityidnamedetails	No	N/A	Used internally by RESTMon. Boolean indicating if the given mapping has both the entity_id and entity_name details. If marked "yes", the entity_id and entity_name map is maintained locally. Refer to the **Topology Mapping with both entity_id and entity_name details** section for sample mapping.
deriveentityvaluesforfields	No	N/A	Used internally by RESTMon. If given "**Yes**", the selected topology mappings fields with entity_id are attached with related entity_name (from the map created based on hasentityidnamedetails). Refer to the **Topology Mapping with deriveentityvaluesforfields, child_c and other fields** section for sample mapping.

Attribute Name	Mandatory	Attribute Type	Description
delimiter	No	N/A	Used internally by RESTMon. • If the given field for deriving entity_name has any internal separator to be considered. (For example, name=Apps|<app_entity_id>|<agent_entity_id>, in this case delimiter="|" should be used). • Used in combining multiple entity ids for which entity name has to be derived. • Any value except "&&" can be used. Refer to the **Topology Mapping with deriveentityvaluesforfields, child_c and other fields** section for sample mapping.
isDeltaTopology	No	N/A	Used internally by RESTMon. • Disables the default behavior of RESTMon in calculating the delta topology. • Used in case the incoming payload has only the delta data. Refer to the **Topology Mapping with isDeltaTopology** section for sample mapping.
startTime	No	Long(ms)	• Represents the start time of the topology entity (vertex/edge). • Defaults to the current system time if not available in the schema mapping.

Topology Mapping with both entity_id and entity_name Details


```
{
    "xml_ns": "",
    "url": "backends",
    "group": "Applications",
    "layer": "CUSTOM",
    "attributes": {
        "oi": {
            "name": "$[*].name",
            "entity_id": "$[*].id",
            "entity_name": "$[*].name",
            "type": "Backend",
            "hostname": "#(function() {return url.split('/')[6]})();",
            "applicationname": "#(function() {return url.split('/')[6]})();",
            "ci_unique_id": "$[*].name",
            "product": "Appdynamics",
            "condition": "false",
            "hasentityidnamedetails": "true"
        }
    }
},
```


Topology Mapping with deriveentityvaluesforfields, child_ci_unique_id, and Other Fields


```
ctions-snapshot",
",

tion() {var tocallChainNames=[];var hasExitCallsData=false;for(var i=0;i<root.length;i++){var firstInChainBool=root[i].firstInChain;var
ction() {var tocallChainNames=[];var hasExitCallsData=false;for(var i=0;i<root.length;i++){var firstInChainBool=root[i].firstInChain;var
tion() {var backendToTypeMap={'JMS':'MESSAGE_QUEUE', 'MongoDB':'DATABASE','Mongo DB':'DATABASE', 'JDBC':'DATABASE', 'WEB_SERVICE':'WEBSE
actionId": "#(function() { var tocallChainNames=[];var hasExitCallsData=false;for(var i=0;i<root.length;i++){var firstInChainBool=root[i]
function() {return url.split('/')[6]})();",
me": "#(function() {return url.split('/')[6]})();",
"#(function() { var tocallChainNames=[];var hasExitCallsData=false;for(var i=0;i<root.length;i++){var firstInChainBool=root[i].firstInC
que_id": "#(function() { var tocallChainNames=[]; var hasExitCallsData=false; var nodemap ={}; for(var i=0;i<root.length;i++) { var first
"TIER&&AGENT&&BUSINESSTRANSACTION%//%&&%//%#(function() {var tocallChainNames=[];var x={};var hasExitCallsData=false;for(var i=0;i<root.
dynamics",
ue_id": "#(function(){var tocallChainNames=[]; var hasExitCallsData=false; var sequenceNodeMap={}; var parentChildRelations={}; var backe
#(function(){var tocallChainNames=[];var hasExitCallsData=false;for(var i=0;i<root.length;i++){var firstInChainBool=root[i].firstInChain;
ull&&null&&null%//%&&%//%#(function() {var tocallChainNames=[];var x={};var hasExitCallsData=false;for(var i=0;i<root.length;i++){var fir
",
luesforfields": "name,agent,ci_unique_id,parent_ci_unique_id,child_ci_unique_id"
```

Topology Mapping with isDeltaTopology

```
{
  "xml_ns": "",
  "url": "",
  "group": "Topology",
  "layer": "CUSTOM",
  "isDeltaTopology" : "true",
  "attributes": {
    "oi": {
      "ci_unique_id": "12321",
      "type": "HOST",
      "product": "TEST",
      "name": "12321",
      "hostname": "12321",
      "Display Name": "12321",
      "Class Label" : "Unix Servers"
    }
  },
  "fieldId": "7"
}
```

Topology-Specific Profile Configuration

The following topology-specific attributes must be defined in the profile configuration:

Attribute Name	Mandatory	Default Value	Attribute Type	Description
inventory_topology_fullsync_interval_mins	No (Default value is used, if Not defined)	24 hours	String	• The full sync of complete topology data is executed based on this value. • Default: 24 hours.
topology_ttl_mins	No (Default value is used, if Not defined)	48 hours It is advisable to use TTL of more than 24 hrs, though the configuration supports even lesser values.	String	• Defines the lifetime of the vertex. • "endTime" of the vertex is calculated based on the ttl that is given, which is startTime(ms)+ttl(ms)

Topology Correlation Support

You can attain a rule-based edge creation by defining the correlation metadata about the different DX OI monitored products. The data source that is integrated can create rules. These rules are executed on top of the topological data that is available in the DX OI platform from the various related monitored inventories. The following two types of correlation available are:

- Compaction
- Association

Compaction Rule Definition

The compaction rule is useful in compacting the vertex when two different vertices from two different sources depict the same object.

Sample Compaction Metadata Definition:

```
{
    "xml_ns": "",
    "url": "hosts",
    "group": "Hosts",
    "layer": "CONNECTOR_METADATA",
    "attributes": {
        "oi": {
            "ci_unique_id": "DYNATRACE_HOST",
            "name": "Dynatrace Compaction Rule",
            "type": "SERVICE_UI_CONFIG",
            "compactionRules": [
                {
                    "sourceTasVertices": {
                        "attributeFilters": [
                            {
                                "attributeName": "product",
                                "attributeValues": [
                                    "Dynatrace"
                                ],
                                "operator": "IN"
                            },
                            {
                                "attributeName": "type",
                                "attributeValues": [
                                    "HOST"
                                ],
                                "operator": "IN"
                            }
                        ],
                        "layer": "CUSTOM"
                    },
                    "matchingAttributeNames": [
                        "ipAddresses",
                        "hostname"
                    ],
                    "matchingAttributeTypes": [
                        "IP_ADDRESS",
                        "ACN_HOST"
                    ]
                }
            ]
        }
    }
}
```

Note: The attributes that depict the compaction rules are defined under the **compactionRules** section. The following table lists the various attributes to be defined in the schema.

Attribute Name			Mandatory	Allowed Values	Description
layer			Yes	CONNECTOR_METADATA	Indicates the TAS layer to which the metadata is to be ingested.
ci_unique_id			Yes	String	Indicates the unique identifier to identify the defined rule.
name			Yes	String	Describes the defined rule.
type			Yes	SERVICE_UI_CONFIG	Indicates the type of the TAS vertex.
sourceTasVertices: Identifies the specific vertices to apply the compaction.					
	attributeFilters		Yes (Filter definition for product and the type of vertex to be considered for compaction are mandatory.)		An array of possible attribute filters.
		attributeName	Yes	String	Name of the attribute For example: product, type
		attributeValues	Yes	Array of Strings	Value of the attribute For example: ["Dynatrace"]
		operator	Yes	String	The operator for the value to be compared to filter the vertices. For example: IN
	layer		Yes	CUSTOM	Indicates the layer of the vertex where this compaction has to be applied, and is "**CUSTOM**" for RESTMon based products.

Attribute Name			Mandatory	Allowed Values	Description
matchingAttributeNames			Yes	Array of Strings • Allowed values are any of the following combinations: [ipAddresses, macAddresses, hostname/ FQDNHostname]	Defines the attributes whose value should be used for comparison with other similar vertices from the other products.
matchingAttributeTypes			Yes	Array of Strings • Allowed values are any of the following combinations: ["IP_ADDRESS", MAC_ADDRESS, "ACN_HOST"]	Defines the category or type of attributes to which the attribute mentioned in **matchingAttributeName** belongs to. This helps in identifying which attribute in the other products can be used for comparing with this attribute in this product.

Association Rule Definition

The Association Rule is useful when two different vertices from one or more different sources depict different objects which relate in some manner to each other. In this case, an edge is created between those two vertices with the preconfigured semantic.

Sample Metadata Definition with Association Rules:

```
{
    "xml_ns": "",
    "url": "databases",
    "group": "Databases",
    "layer": "CONNECTOR_METADATA",
    "attributes": {
        "oi": {
            "ci_unique_id": "APMOSE_TO_APPD_AGENT1",
            "name": "APM Openshift to App Dynamics AGENT association",
            "type": "SERVICE_UI_CONFIG",
            "associationRules": [
                {
                    "sourceTasVertices": {
                        "attributeFilters": [
                            {
                                "attributeName": "product",
                                "attributeValues": [
                                    "APM"
                                ],
                                "operator": "IN"
                            },
                            {
```

```
                                "attributeName": "type",
                                "attributeValues": [
                                    "OPENSHIFT"
                                ],
                                "operator": "IN"
                            }
                        ],
                        "layer": "INFRASTRUCTURE",
                        "matchingAttributeNames": [
                            "ose_pod_name"
                        ]
                    },
                    "targetTasVertices": {
                        "attributeFilters": [
                            {
                                "attributeName": "product",
                                "attributeValues": [
                                    "AppDynamics"
                                ],
                                "operator": "IN"
                            },
                            {
                                "attributeName": "type",
                                "attributeValues": [
                                    "AGENT"
                                ],
                                "operator": "IN"
                            }
                        ],
                        "layer": "CUSTOM",
                        "matchingAttributeNames": [
                            "hostname"
                        ]
                    },
                    "semantic": "contains"
                }
            ]
        }
    }
}
```

Note: The attributes depicting the association rules are defined under the **associationRules** section. The following table lists the various attributes to be defined in the schema.

Attribute Name	Mandatory	Allowed Values	Description
layer	Yes	CONNECTOR_METADATA	Indicates the TAS layer to which the metadata is to be ingested.
ci_unique_id	Yes	String	Indicates the unique identifier to identify the defined rule.

Attribute Name			Mandatory	Allowed Values	Description
name			Yes	String	Describes the defined rule.
type			Yes	SERVICE_UI_CONFIG	Indicates the type of the TAS vertex.
sourceTasVertices: Identify the specific vertices to apply association.					
	attributeFilters		Yes. (Filter definition for product and the type of vertex to be considered for compaction are mandatory.)		An array of possible attribute filters.
		attributeName	Yes	String	Indicates the name of the attribute. For example: product, type
		attributeValues	Yes	Array of Strings	Indicates the value of the attribute. For example: ["APM"]
		operator	Yes	String	Indicates the operator for the value to be compared to filter the vertices. Example: IN
	layer		Yes	String	Indicates the layer of the vertex where this association has to be applied. Can be CUSTOM/ INFRASTRUCTURE_UIM/ INFRASTRUCTURE, and so on, based on the TAS layer to which the data is ingested.
matchingAttributeNames			Yes	Array of Strings	Indicates the name of the actual attribute which is used for comparing and associating across source and target.
targetTasVertices: Identifies the specific vertices to apply association.					

Attribute Name			Mandatory	Allowed Values	Description
	attributeFilters		Yes. (Filter definition for product and the type of vertex to be considered for the association are mandatory.)		An array of possible attribute filters.
		attributeName	Yes	String	Indicates the name of the attribute. For example: product, type
		attributeValues	Yes	Array of Strings	Indicates the value of the attribute. For example: ["AppDynamics"]
		operator	Yes	String	Indicates the operator for the value to be compared to filter the vertices. For example: IN
	layer		Yes	String	Indicates the layer of the vertex where this association has to be applied. Can be CUSTOM/ INFRASTRUCTURE_UIM/ INFRASTRUCTURE and so on based on the TAS layer to which the data is ingested.
matchingAttributeNames			Yes	Array of Strings	Indicates the name of the actual attribute which is used for comparing and associating across source and target.
semantic			No	String	Indicates the type of association to be created between the source and target vertices. For example: contains

Relation Support

This section defines the relations (edges) to be used to connect the topology vertices. This helps in creating the topology edges if it must be created from different url or payloads compared to the url or payload of the vertices.

```
{
```

```
    "appdynamics": {
        "definition": {
            "resource_category": null,
            "uploadedBy": "RESTMON",
            "updatedBy": "",
            "version": "2.0",
            "defaults": {
                "port": 443,
                "interval": 60,
                "httptimeout": 30000
            },
            "auth": "basic",
            "xml_ns": "",
            "name": "appdynamics",
            "type": "https"
        },
        "urls": [],
        "topology": [],
        "alarms": [],
        "metrics": [],
        "calculated_methods": [],
        "calculated_metrics": [],
        "groups": [],
    "relations": [
         {
            "xml_ns": "",
            "url": "relations",
            "isDeltaTopology": "true",
            "group": "relations",
            "layer": "CUSTOM",
            "attributes": {
               "oi": {
                  "parent_ci_unique_id": "$['relations'][*]['end1Id']",
                  "child_ci_unique_id": "$['relations'][*]['end2Id']",
                  "product": "Microfocus UCMDB"
               }
            }
         }
      ]
    }
}
```


Attribute Name	Mandatory	Attribute Type	Description
product	No	String	Indicates the product name.
parent_ci_unique_id	Yes	String	Identifies the specific parent vertices to apply the relation.
child_ci_unique_id	Yes	String	Identifies the specific child vertices to apply the relation.

Attribute Name	Mandatory	Attribute Type	Description
semantic	No	String	Indicates the type of association to be created between the source and target vertices. For example: contains
isDeltaTopology	No	N/A	Used if the incoming payload has only the delta data.
layer	Yes	String	Indicates the layer of the vertex where this association has to be applied. Can be CUSTOM/ INFRASTRUCTURE_UIM/ INFRASTRUCTURE and so on based on the TAS layer to which the data is ingested.
url	Yes(For Pull schemas)	String	To map the url-id with the url section id tagged with the relation url.
group	Yes	String	Indicates the group name to be monitored.

Profile

In the profile section of the schema file, you can configure values that are related to RESTMon data processing and authentication.

This section includes the following information:

Introduction to a Profile

The Profile of the schema is defined inside the *restmon.json* file under the profiles JSON Array.

Example for Pull Schema

```
{
"profile": {
"name": "zabbix",
"active": "yes",
"schema": "zabbix",
"polling_interval_secs": "300",
"topology_ttl_mins": "2880"
},
"restapiconnectdetails": {
"type": "http",
"hostname": "sampleHostName",
"port": "samplePort",
"authentication": "none",
"username": "sampleUserName",
"password": "samplePassword",
"realmdomain": "",
"token": "",
"httptimeout": "30000",
"checkcert": "no"
```

```
},
"monitored_groups": {
"Topology": "yes",
"Items": "yes",
"Events": "yes"
}
}
```

Example for Push Schema

The following table lists the different attributes in the profile section:

```
{
"profile": {
"name": "zabbixwebhook",
"active": "yes",
"schema": "zabbixwebhook",
"streaming": "yes",
"polling_interval_secs": 1,
"batch_size": 1000,
"batch_wait_time_milli": 2000,
"tenantname": "sampleTenantName"
},
"monitored_groups": {
"Events": "yes"
}
}
```

Attribute	Attribute Type	Mandatory	Description
active	String "yes" or "no"	No. Defaults to "false".	Used to identify if the data processing has to be started or not.
schema	String	Yes	The name of the schema file to be used. The combination of profile name and schema name must be unique.
polling_interval_secs	Integer (String)	No	The standard polling interval at which the schema parsing/processing to be scheduled. Ensure that the polling interval is set for a minimum of 60 seconds or more to avoid performance issues. For more information, see Polling Interval for Schema Parsing. Applicable only to the pull schemas.
alarms_polling_interval_secs	Integer (String)	No	The alarm data-specific polling interval that the RESTMon uses to schedule the subsequent schema parsing/processing. Ensure that the polling interval is set 60 seconds or more to avoid performance issues. For more information, see Polling Interval for Schema Parsing
metrics_polling_interval_secs	Integer (String)	No	The metric data-specific polling interval that the RESTMon uses to schedule the subsequent schema parsing/processing. Ensure that the polling interval is set for 60 seconds or more to avoid performance issues. For more information, see Polling Interval for Schema Parsing
topology_polling_interval_secs	Integer (String)	No	The topology data-specific polling interval that the RESTMon uses to schedule the subsequent schema parsing/processing. Ensure that the polling interval is set for 60 seconds or more to avoid performance issues. For more information, see Polling Interval for Schema Parsing.

Attribute	Attribute Type	Mandatory	Description
entity_name	String	No	The application name for the profile. The attribute can have the value that is used in the schema by giving %%entity_name%%. Multiple entity name can be given in the profile with any separator and similar logic has to followed in the schema to convert it to the Array format. For example, "entity_name" : "entity1, entity2" **Note:** Applicable only to the pull schemas.
topology_ttl_mins	Integer (String)	No	Used to clear the delta edges from DB which is having expired date. endtime is calculated as the current time + ttl time
keepalive	String		Keeps the topology entity alive as long as the RESTMon instance is up and running after the initial ingestion when the attribute is set to 'true'. DX Operational Intelligence automatically extends the TTL of the entity by the configured ttl value. Supported: yes, no
discontinueProcessingOnDataFetchFailure	Boolean		Discontinues the json parsing/processing when the url fetch fails. By default, the property is set to false. You can set the property to true when you do not want the processing to continue url fetches fail. This is applicable for poll schemas.
gracePeriodForTopologyTTLRefreshMin	Integer		Defines the grace period for refreshing the topology data in toDX Operational Intelligence. The data refresh continues as and when there is any change in the entity data. For example, if the ttl is configured as 8 hours, and the gracePeriodForTopologyTTLRefreshMin is configured as 60 minutes, RESTMon reingests topology data in to DX Operational Intelligence after 7 hours (8hours - 60 minutes) of the previous ingestion.
restapiconnectdetails.type	String	Polling - Yes Steaming - No	The type of requests made to be used for urls in schema Example: "type" : "https"
restapiconnectdetails.hostname	String	Polling - Yes Steaming - No	The host name of requests made to be used for urls in schema Example: "hostname" : "localhost"
restapiconnectdetails.instancename	String	Polling - Yes Steaming - No	The instance to which the request is made. For example, "instancename" : "localhost"
restapiconnectdetails.port	String	Polling - Yes Steaming - No	The port of requests made to be used for urls in schema. Example: "port" : "8080"
restapiconnectdetails.authentication	String	Yes(Not required for streaming or push schemas).	The authentication type of requests made to be used for urls in schema Example: "authentication" : "urltoken", "token" : "artBBOcP4KSzC2w+15I2Wd1OSNbcV2f2gOtJVN1hNhLdmsmmdyGAqHVHWRZMlRZg",
restapiconnectdetails.username	String	Polling - Yes Steaming - No	The username for authentication of requests made to be used for urls in schema.
restapiconnectdetails.password	String	Polling - Yes Steaming - No	The password for authentication of requests made to be used for urls in schema.
restapiconnectdetails.realmdomain	String	Polling - Yes Steaming - No	The domain name of the request
restapiconnectdetails.token	String	Polling - Yes Steaming - No	The token that is used for authentication of requests that are made to be used for urls in schema.
restapiconnectdetails.httptimeout	String	Polling - Yes Steaming - No	The timeout for requests made to be used for urls in the schema in ms. For example, "httptimeout" : "120000"

Attribute	Attribute Type	Mandatory	Description
restapiconnectdetails.checkcert	String "yes" or "no"	Yes(Not required for streaming or push schemas).	Specifies if checkcert is needed or not. Creates a ssl socket factory for the update of OAuth access token. Default value is **no**.
restapiconnectdetails.httpclientMaxConnections	String	Polling - Yes Steaming - No	Used to configure the number of maximum connections for the netty HTTP client to avoid the overload. For example, "httpclientMaxConnections" : 5000
restapiconnectdetails.instancename	String	Yes(Not required for streaming or push schemas).	The instance to which the requests are made.
restapiconnectdetails.password	String	Yes(Not required for streaming or push schemas).	The password for authentication of requests made to be used for urls in schema
monitored_groups	JSon Array	Yes	The monitored group contains the list of groups that are mentioned in the schema. The key is the group name and value -"yes" or "no". "yes" - if the section in schema must be taken for processing. "no" - if the section in schema need not be taken for processing. Example: "monitored_groups": { "Hosts": "yes", "Alarms": "yes", "Processes": "no", "Services": "yes" } In schema the group name is mentioned using field "group". The path to find this in alarm of the above sample schema is: schema name => alarm => [0] => group
batch_size	Integer	No	The size of batches with which processing happens. Applicable for only the push schemas. For example, "batch_size" : 3000
batch_wait_time_milli	Integer	No	The wait time given to fill up the batch size. If the wait time is exceeded, then existing batch is executed. This attribute is applicable for only push schemas. For example, "batch_wait_time_milli" : 2000
is_array_input	Boolean (String)	N	This attribute specifies the payload for processing is in an array format. This attribute is applicable for only push schemas. Example: "is_array_input" : "true"
streaming_array_size	Integer	No	The array size to be followed while processing. Applicable for only push schemas and only if is_array_input is "true". Example: "streaming_array_size" : 10

Polling Interval for Schema Parsing

The polling Interval configuration is applicable only to the pull schemas.

RESTMon enables you to schedule the polling intervals for parsing the data from schemas. You can configure the polling intervals in the profile definitions using standard and dynamic polling attributes.

- The Standard Polling Interval enables you to simultaneously schedule the data parsing intervals for various data categories. You can use the attribute `polling_interval_secs` to define the standard polling interval. RESTMon fetches and parses the data (Topology, Metrics, and Alarms) in a sequential order as per the defined polling schedule.
- The Dynamic Polling Interval enables you to schedule the data parsing intervals for each data category using the following attributes:

  ```
  alarms_polling_interval_secs
  metrics_polling_interval_secs
  topology_polling_interval_secs
  ```

You can configure both dynamic and standard polling interval attributes simultaneously or can configure one of the attributes.

```
"profile" : {
"name" : "dynatrace",
"active" : "yes",
"schema" : "dynatrace",
"polling_interval_secs" : "300",
"alarms_polling_interval_secs" : "60",
"metrics_polling_interval_secs" : "60",
"topology_polling_interval_secs" : "300",
"topology_ttl_mins" : "2880"
}
```

- If you configure both standard and dynamic polling intervals,
  - For the initial polling, RESTMon parses the data from various categories as per the specified standard polling interval in a sequential order.
  - For the subsequent polling, RESTMon parses the data as per the dynamic polling intervals defined for each data category.
- If you define only dynamic polling interval,
  - For the initial polling, RESTMon parses the data from various categories simultaneously.
  - For the subsequent polling, RESTMon parses the data as per the dynamic polling intervals defined for each data category.

For example, the following configuration enables you to schedule the parsing of topology data at every four(4) hours, and alarm and metric parsing at every 300 seconds:

```
"profile" : {
"name" : "dynatrace",
"active" : "yes",
"schema" : "dynatrace",
"polling_interval_secs" : "300",
"topology_polling_interval_secs" : "14400",
"topology_ttl_mins" : "2880"
},
```

RESTMon schedules the initial polling as per the standard polling interval. The subsequent polling for topology happens at every four(4) hours (14400 seconds). The subsequent polling for alarm and metric happens at every 300 seconds(5 minutes).

Supported Authentication Types

RESTMon supports the following authentication methods: **basic**, **digest**, **NT Lan Manager** (**ntlm)**, **oauth2**, **signature**, **aws_signature**, **token**, **urltoken**, and **bearer**. You can define the value for the authentication method under *restapiconnectdetails.authentication*. If no authentication is required, then enter the value as **none**.

The following snippets are *restapiconnectdetails* samples for some of the authentication methods:

Basic Authentication Example

```
"restapiconnectdetails" : {
"type" : "https",
"hostname" : "test012804.bpc.broadcom.net",
"port" : "443",
"authentication" : "basic",
"username" : "admin",
"password" : "3mGz1ASQbGoYPvzzvDJxqw==",
```

```
"realmdomain" : "",
"token" : "",
"httptimeout" : "120000",
"checkcert" : "no"
}
```

ntlm Example

```
"restapiconnectdetails" : {
"type" : "http",
"hostname" : "test1.forwardinc .biz",
"port" : "",
"authentication" : "ntlm",
"username" : "test.user",
"password" : "IVAMKEeNesPDveavhasdw==",
"realmdomain" : "",
"token" : "",
"httptimeout" : "120000",
"checkcert" : "no"
}
```

Url Token Example

```
"restapiconnectdetails" : {
"type" : "http",
"hostname" : "<hostname>",
"port" : "80",
"authentication" : "urltoken",
"username" : "",
"password" : "",
"token" : "ga64nG75SKr8F9+MWySOLNfxhWzFgYVeNM4",
"realmdomain" : "",
"httptimeout" : "30000",
"checkcert" : "no"
}
```

Bearer Token Example

```
"restapiconnectdetails": {
    "type": "https",
    "hostname": "test.saas.appdynamics.com",
    "port": "",
    "authentication": "Bearer",
    "username": "",
    "password": "",
    "realmdomain": "",
    "token": "JSAv1cJYRvkivKeR6SB",
    "httptimeout": "120000",
    "checkcert": "no"
  }
```

Outbound Integration

The outbound integration allows DX Operational Intelligence to communicate with various ticketing systems, notification channels, and webhooks.

Using the outbound integration, you can perform the following tasks:

- Seamlessly trigger email notifications using Policies.
- Manually trigger email notifications through Alarm Analytics.
- Manage Alarm and alarm tracking using Ticket Management.

DX Operational Intelligence supports the outbound integration with the following channels and applications:

- DX Gateway
- Channels
- Policies Overview
- Message Templates

DX Gateway

You can use the DX gateway to send alarms, integrate with CA Service Management, or integrate other products using REST APIs.

The DX Gateway package consists of On-Prem Gateway and On-Prem ITSM (CA Service Management). You can install all the components or only the required component by modifying the common configuration file.

- **On-Prem Gateway:** The On-Prem Gateway is an interface that collects alarms from DX Operational Intelligence and sends alarms to the On-Prem third-party products through channels. The On-Prem Gateway acts as a poller that collects alarm data periodically. The On-Prem Gateway and the third-party product must be installed on the same system. For more information, see the On-Prem Gateway section.
- **On-Prem ITSM:** On-Prem ITSM (Incident Manager) integrates with CA Service Management (CA SM) to track and manage alarms that are raised in DX Operational Intelligence. When an alarm occurs in DX Operational Intelligence, a ticket is created in the CA Service Management system. For more information, see the On-Prem ITSM section.

  NOTE
  RESTMon is no longer available with DX Gateway. For more information about RESTMon, see the RESTMon section.

Supported Integrations Using DX Gateway

Using DX Gateway, you can integrate DX Operational Intelligence with the following products:

- BMC Remedy

DX OI	BMC Remedy
SaaS	**BMC Remedy 20.2**

  NOTE
  Integration with the SaaS version of BMC Remedy is not supported.

- CA Service Management

DX OI	CA Service Management
SaaS	**On-Prem**

Prerequisites for DX Gateway

Before you deploy the DX Gateway, make sure you meet the minimum hardware and configuration requirements.

Review the following prerequisites before deploying DX Gateway.

Hardware Requirements

Deploy DX Gateway on systems with the following minimum resources:

- Memory - 16 GB of RAM
- CPU - 4 Cores
- Storage - 100 GB

Configuration Requirements

Before you configure and deploy DX Gateway,

- Verify that you have access to DX Operational Intelligence.
- Ensure that OpenJDK 11 is installed on the system where you want to deploy the DX Gateway.
- Also, ensure that any release of Java is installed and the JAVA_HOME environment variable points to your Java installation directory.

Download DX Gateway

You can download the DX Gateway using the Settings page in DX SaaS.

To download DX Gateway, follow these steps:

1. Login to DX Operational Intelligence.
2. Click **Settings**.
3. Click **Setup** in the **Data Sources** tab.
   The **Downloads** page is displayed.
4. Click **DX Gateway** to download the package.
   The DX Gateway package is downloaded.

Contents of the DX Gateway Package

The DX Gateway package contains both configuration files and installation scripts.

The DX Gateway package consists of the following files and folders:

- **Config**: This folder contains the **generic_config.json** file which is required to configure the common DX Operational Intelligence connection details.
- **installer.bat and installer.sh**: Use the **installer.bat** or **installer.sh** to start the installation on a Windows or Linux computer respectively.
- **Scripts**: This folder contains the scripts to start and stop the installers.
- **Onprem**: This folder contains the On-Prem Gateway configuration files.
- **Onprem-Itsm**: This folder contains the On-Prem ITSM configuration files.

Configure and Deploy DX Gateway

The common configuration file (**generic_config.json**) in DX Gateway enables you to deploy On-Prem Gateway and On-Prem ITSM simultaneously or individually based on your requirement.

- To deploy the components simultaneously, use *installer.bat* on Windows or *installer.sh* on Linux in the **DX-Gateway** folder.
- To deploy the components separately, use the start scripts that are available in the respective folders.

Prerequisites:

Before you start the deployment,

- Ensure that you have reviewed the DX Gateway Prerequisites section.
- Ensure that you have downloaded the DX Gateway package from the **Settings** page.
- For On-Prem Gateway, verify that the parameters are configured in the *onprem_gateway.properties* and the *webhook_gtw.cfg* files. For more information, see the Configure On-Prem Gateway Parameters section.
- For On-Prem ITSM, review the **On-Prem ITSM** specific requirements. This includes creating and adding the ITSM channel. For more information, see the Configure Ticket Management Channels section.

The following procedure lists only the parameters which require the user input. All other parameters in the **generic_config.json** file are auto-populated when you run the DX-Gateway installer.

Follow these steps:

1. Navigate to the ***<DX-GATEWAY_Installation_Directory>*\Config** folder.
2. Edit the following parameters in the *generic_config.json* file:

   Some of the parameters in this file are auto-populated.

   a) **dxsaas_tenant_id**: Enter the DX Operational Intelligence tenant name. For example, *example_tenant*.
   b) **dxsaas_cohort_id**: Enter the cohort ID. You can find this information on the **Settings** > **Connector Parameters** page.
   c) **dxsaas_username**: Enter the DX Operational Intelligence username. For example, *user@example.net*.
   d) **dxsaas_password**: Enter the password for the user. The application encrypts the password and sets the encryption field to **true**.

   NOTE

   - Enter the password as clear text and leave the *dxsaas_password_encrypted* field to **false**.
   - If the service account user is enabled for your tenant, provide that service account user name and password for integrations. In addition to these parameters, also ensure to update the *dxsaas_cohort_id* parameter.
3. (Optional - Only for a proxy configuration) If the environment is configured to use a proxy, then update the following parameters in the *generic_config.json* file:

   a) **proxy_enabled**: Enter true to enable the proxy configuration. Default: false.
   b) **proxy_hostname:** Enter the hostname which has proxy configured.
   c) **proxy_port**: Enter the port for the proxy hostname. Default: *8080*.
   d) **proxy_protocol**: Enter the protocol to be used, either HTTP or HTTPS.
4. Save the changes to the *generic_config.json* file.
5. Run the installer.

   - **Install the components at the same time:** Execute the *installer.bat* on Windows or *installer.sh* on Linux. These files are available in the DX Gateway folder.
   - **Install Individually:** Execute the *start* script for On-Prem Gateway and *startitsm* for On-Prem ITSM. These files are available in the **Onprem** and **Onprem-Itsm** folders respectively.
6. To begin the installation enter **S** in the command prompt as the following prompt illustrates:

   ```
   What action do you like to perform Start(S) or Stop(P) Services?
   ```
7. Enter **Y** for each of the components that you want to install.
8. Verify that the deployment is successful by viewing the ***<DX-GATEWAY_Installation_Directory>*\Logs** folder. The logs folder is created once the installation is initialized for each component. For more information, see the DX Gateway Logs section.

You have now successfully configured the components.

DX Gateway Logs

You can use the log files that are created after installation to troubleshoot issues with each individual DX Gateway component.

The **logs** folder is created once the installation is initialized for the components. The logs folder contains all the individual component logs.

> **NOTE**
> Ensure you provide the required *Read* and *Write* permissions to the folder before you trigger the installation. For example, on Linux: *chmod 755 -R DX-Gateway*

The following log files are created in the logs folder.

- DX Gateway: Logs/dxgateway.log * *This logs issues specific to the DX Gateway startup and initialization.*
- On-Prem Gateway: Logs/onpremgtw.log
- On-Prem ITSM: Logs/incidentmanager.log

On-Prem Gateway

DX Operational Intelligence uses the on-premise gateway to send alarms to third-party products.

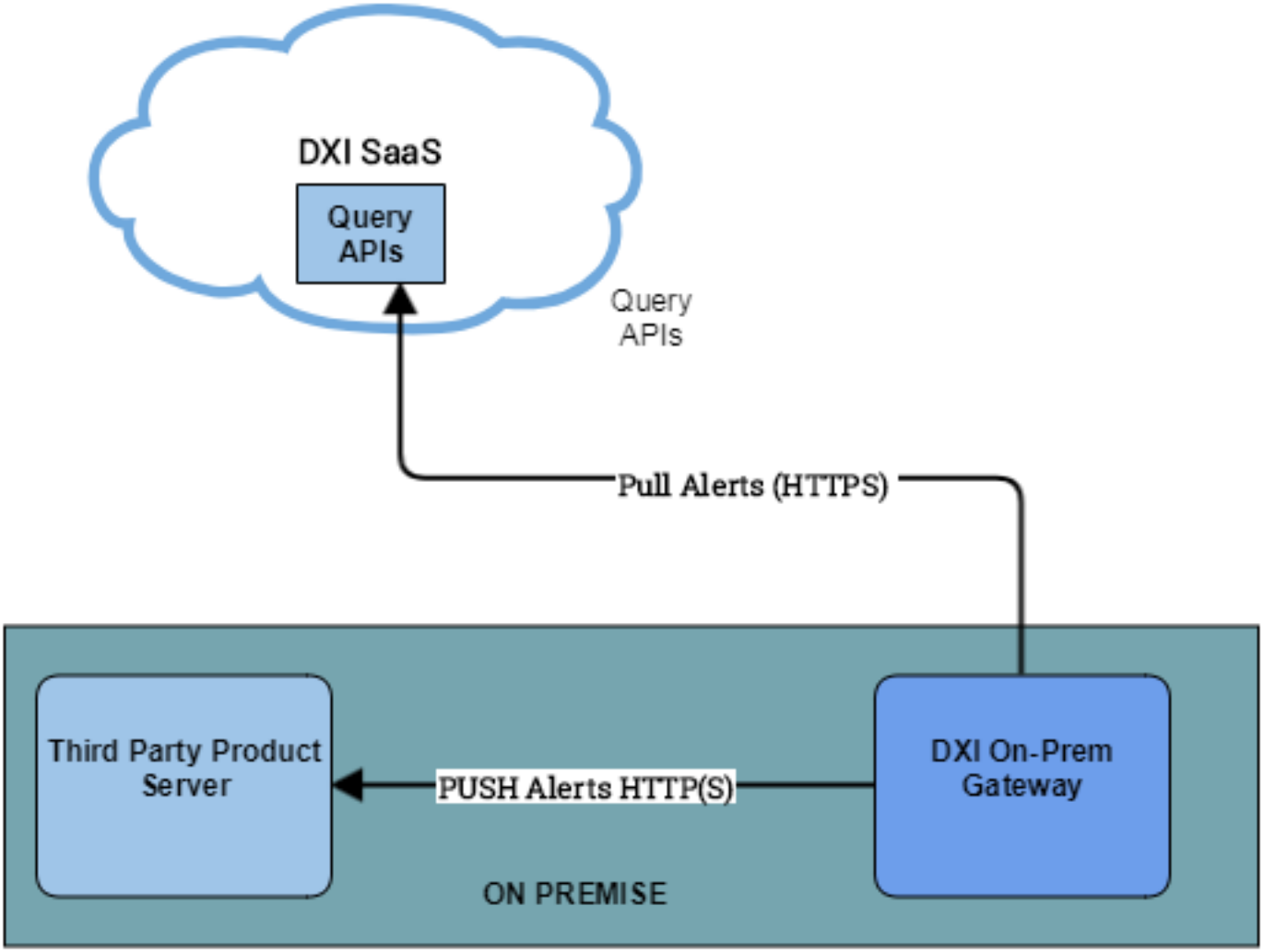
The On-Prem Gateway is an interface that collects alarms from DX Operational Intelligence and sends alarms to the On-Prem third-party products through webhook channels such as Slack. The On-Prem Gateway acts as a poller that collects alarm data periodically. The On-Prem Gateway and the third-party product must be installed on the same system.

Role: Administrator

- The On-Prem Gateway pulls alerts from DX Operational Intelligence using Query APIs through the HTTPS protocol.
- The On-Prem Gateway pushes the alerts to the third-party server that is integrated with DX Operational Intelligence using the Webhook channel.

The following architecture describes the flow of alarms from DX SaaS to the On-Prem server:

Figure 4: On-prem gateway



Video: DX SaaS Gateway Introduction

View this demonstration to understand how to push data from DX SaaS to the various on-premise products.

Video: Outbound Integration through Gateway

View this demonstration to understand how to configure On-Prem Gateway with the Slack integration.

Prerequisites to Install On-Prem Gateway

You can install the On-Prem Gateway separately or along with other components. To install separately, use the **start** script that is available in the **OnPrem** folder. To install the On-Prem Gateway along with other components, use the DX Gateway installer. Before you run the installer, you must configure the On-Prem Gateway Parameters.

NOTE
You must install the On-Prem Gateway on the same system where the third-party product is installed.

Configure On-Prem Gateway Parameters

Before installing the on-premise gateway, configure the parameters for authentication, alarm fetching, and webhooks in the onprem_gateway.properties file.

To configure the On-Prem Gateway parameters, follow these steps:

1. Navigate to the *<DX-GATEWAY_Installation_Directory>*\Onprem folder and edit the parameters as required in the *onprem_gateway.properties* file.

   Configure the following parameters for authentication:

Key	Value
doi.authn.request.tenantId	Specify the tenant ID for whom alarm has to be fetched. For example, user@mail.com
doi.authn.request.tenantPwd	Specify the tenant password. For example, text@1234
doi.authn.request.cohortId	Specify the cohort ID. For example, <string>EFF8FBC7-9C36-423D-BDCB-7BF919B67C72
doi.authn.request.pwdEncryptionEnabled	Specify true or false for the password encryption. True enables the encryption and False disables the encryption. If password encryption is enabled, the input password text is encrypted and the password is stored in the same property file. **Default value:** true
doi.authn.request.pwdEncrypted	**true** to encrypt the password text present in the property file. **false** for the password to be in plain text in the property file. By default, this parameter is false as the tenant password is in the plain text during the initial configuration. However, after the first instance, if pwdEncryptionEnabled=true, tenantPwd will be encrypted and this flag value is changed to true.
doi.env.authnBaseUrl	Specify the base URL of the authentication server. For example, https://axa-adminui.<ipaddress>.devcloud-paas.ca.com
doi.env.authnUri	Specify the URL to authenticate the tenant and acquire the token. **Note:** Do not change this value.
doi.env.alarmBaseUrl	Specify CA Digital Operational Intelligence server base URL which hosts the Alarm API. This URL can be the same as authnBaseUrl.
doi.env.alarmUri	Specify the URL to fetch the alarm. **Note:** Do not modify this value.

Configure the following parameters for fetching alarms:

Key	Value
doi.alarm.request.queryString	Specify the file/product name. For example, product: UIM AND product: Application Performance Management
doi.alarm.request.fromIndex	By default, the ranges start at 0.
doi.alarm.request.pageSize	Specify the page size indicating a maximum number of alarms to fetch per poll cycle. **Default:** 50. The value should not be too less or negative and max should not be more than 10000.
doi.alarm.request.timeFrom	Not required
doi.alarm.request.timeTo	Not required
doi.alarm.pollerDelay	Defines the fixed delay in milliseconds between two poll cycles. Default: 100 seconds, that is, 100000 ms.
doi.alarm.defaultDelayToTrigger	Specify the number of hours when the application runs for the first time. This value determines what should be the timeFrom value. If set as 4, the timeFrom value for the initial poll request will be 4 hours back from the current time.
doi.alarm.lastExecutedDateTime	Not required. The application uses this key to log the last executed date and time.

2. Edit the **webhook_gtw.cfg** file and configure the following parameters. Save the file.

Key	Value
<listeners></listeners	Defines one or more listeners where the alarm should be notified. For example, IBM Netcool, Slack, and so on.
<netcool> cla	Specify the name of the implementation class. For all webhooks, the endpoints must use class = JsonPublisher.
<netcool><filter></>

Key	Value
<netcool><1></></>	This section includes the connection configuration that is related to the webhook, such as webhook URL, auth_method, auth_key, username, and password. If auth_method is "basic", username and password field is mandatory. **Format of the section:** `<1> url = https://api.flowdock.com/v1/messages/chat/d8c89012ad9be36596f1d94096a8d25f+1 auth_method = none auth_key = username = password =</1>`

3. Save and close the files.

Proceed to install the DX Gateway using the DX Gateway installer.

On-Prem ITSM

Describes how to deploy On-Prem ITSM.

The On-Prem ITSM (Incident Manager) integration enables you to track and manage alarms raised in DX Operational Intelligence. When an alarm occurs in DX Operational Intelligence a ticket is created in the third-party system. On-Prem ITSM is part of the DX Gateway package and uses the common installer.

Follow these steps:

1. Download the DX Gateway package. For more information, see the Download DX Gateway section.
2. Create the ITSM channel. For more information, see the Configure Ticket Management Channels section.
3. Configure and deploy On-Prem ITSM using the DX Gateway common installer. For more information, see the Configure and Deploy DX Gateway section.

Verify the Deployment

After the integration, verify that On-Prem ITSM is configured correctly after the installation.

Once you configure the integration, you can monitor the log files for ticketing updates or errors.

- Monitor the **onpremnim.log** file in the **<DX-GATEWAY_Installation_Directory>\Logs** directory for ticketing errors.
- Monitor the **incidentmanager.log** file in the **<DX-GATEWAY_Installation_Directory>\Logs** directory for ITSM errors.

NOTE
Before you monitor the logs, ensure that,

- You have configured the *generic_config.json* file from the **<DX-GATEWAY_Installation_Directory>\Config** folder.
- You have used the DX Gateway common installer to deploy On-Prem ITSM.

Channels

{"URL":["https://cloudmanagement/#/settings/notifications"],"description":"concept.dita_3536f72ac0ec1d785e357fed37a3e83c4bc651e2","new":"","new_video":"","admin":"","trou
{"masterkb":"","text":"","URL":[]},"pendo":"","video":[]}

Channels enable you to configure the communication between DX Operational Intelligence and third-party integrations. When you configure channels, you can:

- Seamlessly trigger email notification using Policies.
- Manually trigger email notification through Alarm Analytics.
- Customize the information to be sent using the Message Templates.
- Manage and track alarms using Ticket Management.
- Route data from the monitoring solution to any other Webhook receiver.

After you create a channel, you can:

- **Trigger the Channels:** You can trigger the channels using **Policies** and alarm actions on the Alarm Analytics. DX Operational Intelligence supports the following channels:
  - **Ticket Management:** Use the ticket management to integrate an ITSM product to track, and manage alarms that are raised in DX Operational Intelligence.
  - **Notifications:** Notifications inform you of the DX Operational Intelligence alarms by email, or by creating and updating tickets in ITSM.
  - **Webhook:** Webhook is a fire and forget integration that enables you to route data from the monitoring solution to any other Webhook receiver. The generic Webhook framework can be used to connect to any compatible Webhook receiver using a quick and easy-to-use Webhook channel setup interface.
  - **Automic:** Automic lets you create Webhook Event objects, that enable you to integrate your external system Webhooks with the Event Engine. The Webhook Event object can listen and retrieve Webhook event payloads from external systems in the real-time.
- **Customize the Channel Information:** You can use the Message Templates to customize the content of the message to be sent through a channel when an alert occurs.

This section provides the following information:

- Prerequisites
- Configure Email Channel
- Configure Webhook Channels
- Configure Ticket Management Channels
- Troubleshoot Notifications

Prerequisites

Use the following tasks when you want to configure custom policies and message templates for the channels. These tasks are optional.

- Create Policies
- Create Message Templates

Configure Email Channel

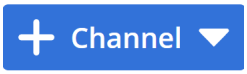
Tenant Administrators can configure email channels to send users alarm notifications.

{"URL":["https://cloudmanagement/#/settings/notifications/channel/email/create"],"description":"task.dita_8ab1526a-1db1-413c-96b9-2adbe5f52d5e","new":"","new_video":"","admin":"","troubleshooting":{"masterkb":"","text":"","URL":[]},"pendo":"","video":[]}

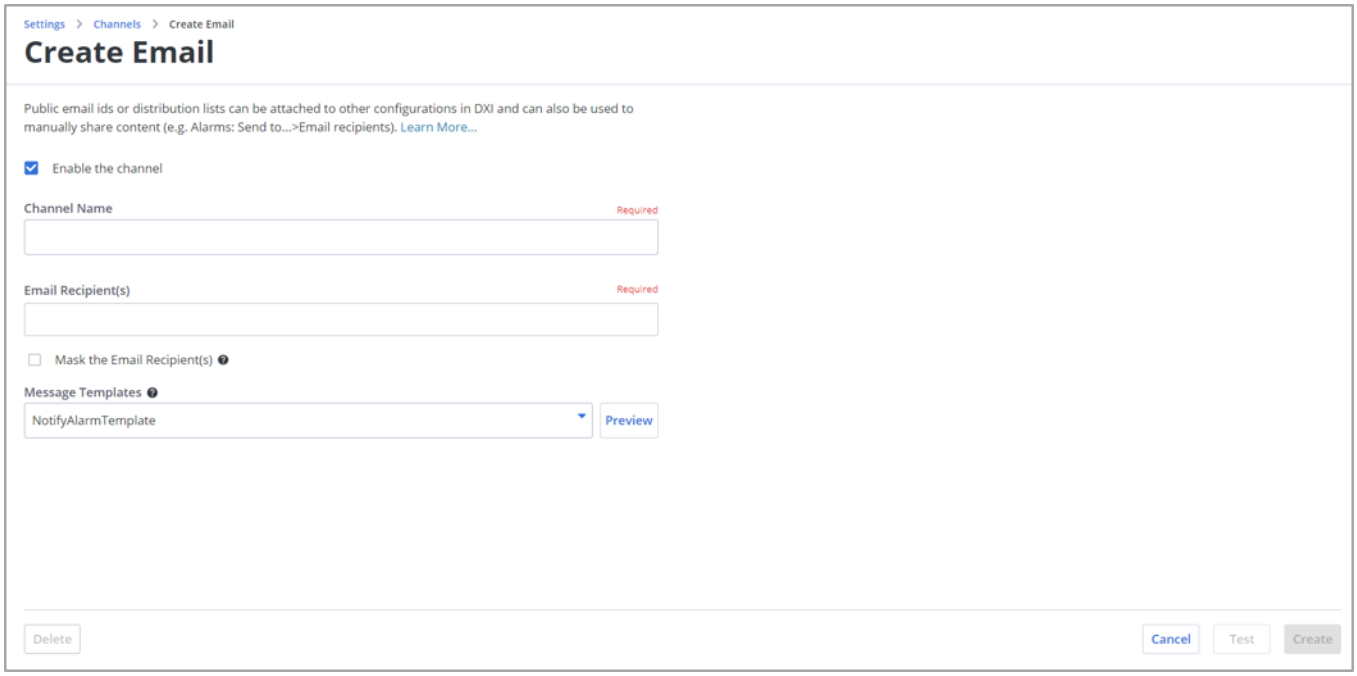
DX Operational Intelligence supports the integration with the Email notification channel. This integration enables notifying users about the alarms through emails. Emails can include alarm information such as severity, alarm message, owner, status changes, and other details.

> **NOTE**
> When you create or edit an email channel, the changes take some time to reflect.

To configure Email Channel, follow these steps:

1. Log in to DX Operational Intelligence as a Tenant Administrator and click Settings in the left navigation pane.
   The application displays the Settings page.
2. Click **Connect** in the Notification Channel Tile.
   The application displays the **Channels** page with the list of existing channels.
3. Click 

   and select Email from the drop-down.
   The Create Email Channel page appears.



4. Select the **Enable the Channel** checkbox.
   This action enables the connection between the DX Operational Intelligence and the channel.
5. Enter a unique channel name in the Channel Name field.
6. Provide one or more email IDs in the **Recipients** field. Use Comma as the delimiter to separate multiple email IDs.
   The application sends emails notifications to the specified email IDs.
7. Select **Mask** to hide the recipient IDs in the distribution list.
8. Select the message template from the **Message Template** drop-down.
   The selected message template determines the contents of the email notification that is triggered manually. For more information, see Message Templates.
9. Click **Test**.
   The application validates the channel configuration.
10. Click **Save**.
    DX Operational Intelligence completes the configuration and enables the integration with the selected Channel.

Configure Webhook Channels

Tenant Administrators can configure webhook channels to route data to a webhook receiver.

A Webhook (also called a web callback or HTTP push API) is a way for an application to provide other applications with real-time information, by piggybacking on the fundamentally decentralized nature of the web. This capability, along with their composable nature, can help build real-time complex pipelines of data to ensure information gathered is not limited by the boundaries of any single product.

Dx Operational Intelligence provides out-of-the-box, yet customizable capability for administrators to use Webhooks as a channel, to route data from the monitoring solution to any other Webhook receiver. You can use the generic Webhook framework to connect to any compatible Webhook receiver using a quick and easy-to-use Webhook channel setup interface.

As a tenant administrator, when you configure the Webhook based integration, you can map the payload to the fields of the receiving product. For out-of-the-box certified Webhook integrations, the mappings are provided out-of-the-box, but you can provide custom mappings as well.

DX Operational Intelligence supports integration with Generic Webhook and Slack:

- Configure Generic Webhook Channel
- Configure Slack Channel

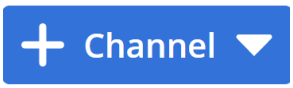
Configure Generic Webhook Channel

Tenant Administrators can configure a generic webhook channel to send data to other applications.

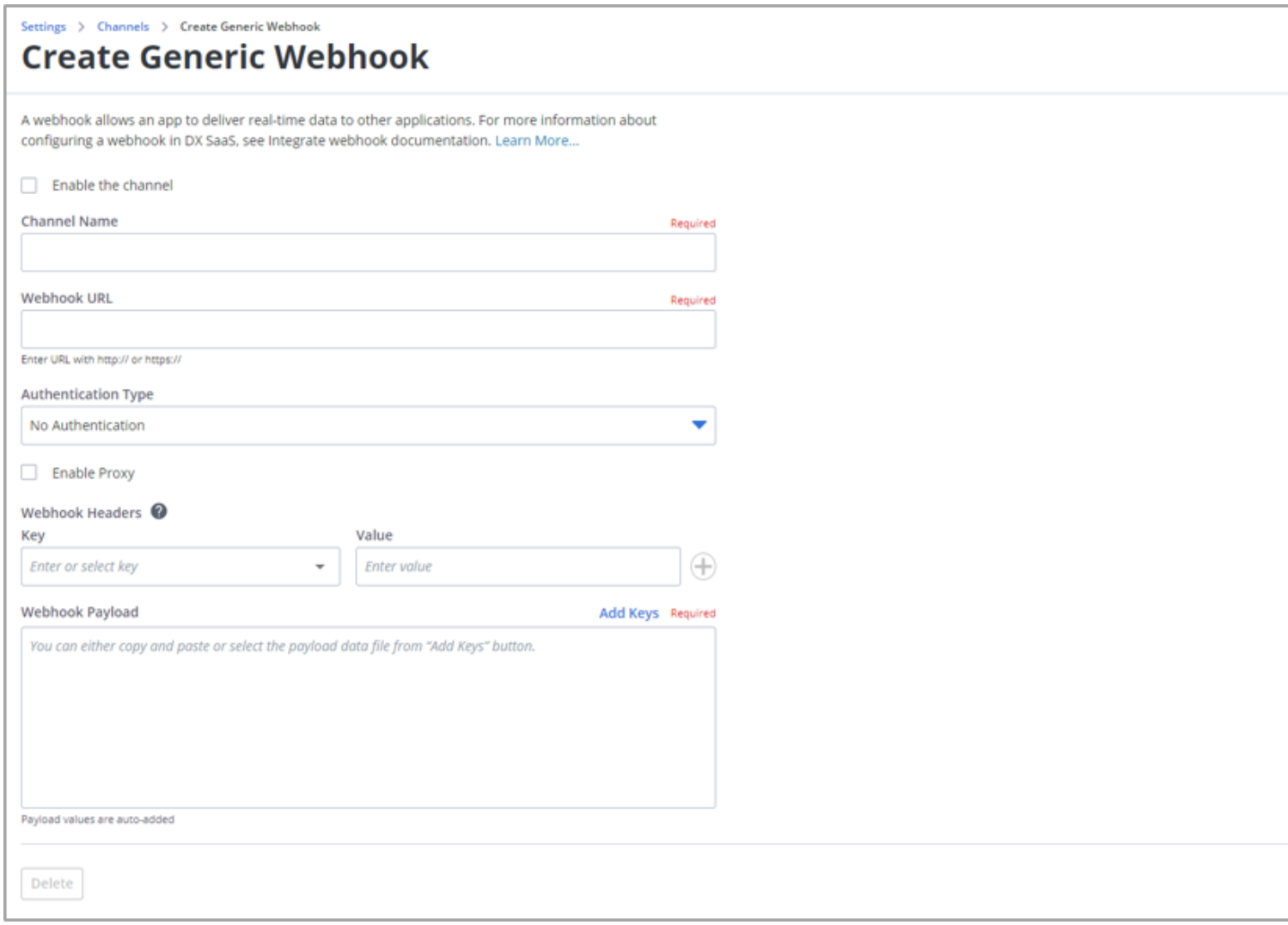
{"URL":["https://cloudmanagement/#/settings/notifications/channel/webhook/genericwebhook"],"description":"task.dita_6ecd8948-ce6b-4b80-9857-c35f8dc3c2f8","new":"","new_video":"","admin":"","troubleshooting":{"masterkb":"","text":"","URL":[]},"pendo":"","video":[]}

> **NOTE**
> When you create or edit a Webhook channel, the changes take some time to reflect.

To configure Generic Webhook channel, follow these steps:

1. Log in to DX Operational Intelligence as a Tenant Administrator and click Settings in the left navigation pane.
   The application displays the Settings page.
2. Click **Connect** in the Notification Channel Tile.
   The application displays the **Channels** page with the list of existing channels.
3. Click 

   and select Webhook (Generic) from the Select Channels Type drop-down.
   The Create Generic Webhook page appears.



4. Select the **Enable the Channel** checkbox.
   This action enables the connection between the DX Operational Intelligence and the channel.
5. Enter a unique channel name in the Channel Name field.
6. Provide the unique Webhook URL to which you intend to send the JSON payload in the Webhook URL field.
7. Select one of the following Authentication Types:
   - **No Authentication:** Select this option, if no authentication is required to connect with Webhooks.
   - **Basic Authentication:** Select this option and provide the user credentials. DX Operational Intelligence uses these credentials to establish the connection with Webhook.
   - **Token Authentication:** Select this option and provide the security token that you are using for authentication with the endpoint in the Token field.

8. Select **Enable Proxy** to route the connection through a proxy server.
9. Define Webhook Headers and Payload in the Webhook Headers section.
   a) **Key and Value:** Specify one or more key-value pairs in the Key and Value fields. You can also select the following keys and provide the values: Accept, Accept-Encoding, Cache-Control, and Content-Type. Add or delete the key-value pairs.
   b) **Webhook Payload:** Click **Add Keys** and select the attributes or you can paste the payload directly in the textbox.

      When you click **Add Keys**, the attributes are listed based on the alarm type. However, for the CI Attributes option to display the attributes, you must register the CI attributes. For more information about how to register the CI Attributes, see the Add or Update the CI Attributes section.

      The application adds the selected keys and the payload values in the Webhook Payload text box. For more information about the supported payload keys, see the Supported Payload Keys section.
10. Click **Test**.

    The application validates the channel configuration.
11. Click **Save**.

    DX Operational Intelligence completes the configuration and enables the integration with the selected Channel.

Supported Payload Keys

The payload keys that you can add are categorized based on the alarm type:

- All Alarm
- Service Alarm
- Situations

Supported Payload Keys for All Alarm

The following table lists all the filter attributes that are supported for All Alarm:

Text	Payload Key	Description	Product
APM Alarm Unique Id	${apm_alarm_unique_id}	Displays the unique ID of the alarm.	DX OI (Applicable only to the DX APM alarms)
Acknowledged	${acknowledged}	Indicates whether an alarm has been acknowledged.	Spectrum
Action Status	${action_status}	Displays the status of the action.	DX OI
Agent	${agent}	Displays the name of the DX APM agent.	DX APM
Alarm Age	${alarmAge}	Displays the age of the alarm.	Spectrum
Alarm Description	${alarm_description}	Displays the alarm description.	DX APM
Alarm Domain	${alarm_domain}	Displays the domain from which the alarm is generated.	DX APM
Alarm Name	${alarm_name}	Displays the name of the alarm.	DX APM
Alarm State	${alarmState}	Displays the state of the alarm.	Spectrum

Text	Payload Key	Description	Product
Alarm Type	${alarmType}	Displays the alarm type. **Values:** Anomaly, Application, Fault	DX APM, UIM, Spectrum, Third-party Products For UIM, Spectrum, and Third-party Products, DX OI populates this field.
Alarm URL	${alarmURL}	Displays the alarm URL.	Spectrum, UIM, ADA, Third-party Products
Alarm Unique Id	${alarm_unique_id}	Displays the unique identification number of the alarm.	All products
Alarm Update	${alarm_update}	Displays the alarm update.	DX OI
Alert External Id	${alert_external_id}	Displays the external ID for the alert.	DX APM
Annotation	${annotation}	Displays the annotation.	UIM
Anomaly Algorithm Type	${algorithm}	Displays the algorithm type of the anomaly.	
Application Name	${applicationName}	Indicates the application name in DX APM.	DX APM
Automic Jobs	${automicJobs}		DX OI
Baseline	${baseline}	Displays the baseline.	DX APM
Breached Threshold	${breached_threshold}	Displays the threshold that was breached.	DX APM
CI ID	${ci_id}	Displays the ID of CI (Configuration Item). A CI represents the component being monitored.	UIM
CI Name	${ci_name}	Displays the CI name.	UIM, Third-party Products, CAPM
CI Type	${ci_type}	Displays the CI type.	UIM, Third-party Products
CI Unique ID	${ci_unique_id}	Displays the unique ID of CI.	OI - NFA and VNA Anomalies
CS ID	${cs_id}	Displays the CS (Computer System) ID.	DX OI
CS Key	${cs_key}	Displays the CS key.	UIM
Cause Code	${causeCode}	Displays the alarm cause code which is an 8-digit, hexadecimal code that identifies the probable cause of the alarm.	Spectrum
Caution Threshold	${caution_threshold}	Displays the caution threshold.	DX APM
Channels	${channels}	Displays the channel.	DX APM
Cleared	${cleared}	Displays if cleared or not.	Cleared

Text	Payload Key	Description	Product
Closed Time	${closedTime}	Displays the closed time.	DX OI
Collection Unique Key String	${collectionUniqueKeyString}		Spectrum
Component Name	${component_name}	Displays the component name.	DX APM
Configuration Item	${configuration_item}	Displays the configuration item.	ADA, Third-party Products
Configuration Item Type	${configuration_item_type}	Displays the Configuration Item type.	UIM, Third-party Anomalies
Correlated External Id	${correlated_external_id}	Displays the correlated external ID.	DX OI
Current Trend Value	${currentTrendValue}	Displays the value of the current trend.	DX OI (Prediction Alarms)
Custom_*n* *where n represents numbers 1-10.*	${custom_n}	You can specify up to 10 custom attributes.	Custom_1 to Custom_5: UIM, Third-party Products Custom_6 to Custom_10: Third-party Products
Custom Num n	${custom_num_n}		Third-party Products
Daily Average	${dailyAverage}	Displays the daily average.	DX OI (Prediction)
Danger Threshold	${danger_threshold}	Displays the danger threshold.	DX APM
Dev Id	${dev_id}	Displays the unique identifier of the device that is involved in the alarm.	UIM
Device Global ID	${deviceGlobalID}	Displays the global ID of the device.	CAPM (CAPM Anomalies)
Device Local ID	${deviceLocalID}	Displays the local ID of the device.	CAPM (CAPM Anomalies)
Device Type	${deviceType}	Displays the type of device that is involved in the alarm.	Spectrum, UIM
Device Type Spectrum	${deviceType_spectrum}	Displays the device type in Spectrum.	Spectrum
Distribution Lists	${distributionLists}		DX APM
Doc Type ID	${doc_type_id}	Displays the ID of the doc type.	DX OI
Doc Type Version	${doc_type_version}	Displays the version of the doc type.	DX OI
Domain	${domain}	Displays the domain from which the alarm is generated.	DX OI
Dst Address	${dstAddr}	Displays the destination address.	ADA, NFA
Dst Port	${distPort}	Displays the destination port	ADA
External Id	${external_id}	Displays the external ID.	All

Text	Payload Key	Description	Product
External Ids	${external_ids}	Displays the external IDs.	N/A
Global Id	${globalID}	Displays the global ID of the alarm.	CAPM
Global ID Router	${globalID_router}	Displays the global ID of the router.	NFA
Group	${group}	Displays information about the group or groups to which the device belongs.	ADA, Spectrum, UIM
Group Id	${group_id}	Displays the group Ids to which the device belongs.	UIM
Host	${host}	Displays the host name.	All
Hub	${hub}	Displays the UIM hub from which the alarm is generated.	UIM
Intercept	${intercept}		DX OI
IP	${ip}	Displays the IP address.	All
Landscape ID	${landscapeID}	Displays the Spectrum landscape ID from which the alarm is generated.	Spectrum
Last Suppressed Severity	${last_supressed_severity}	Displays the last suppressed severity.	DX OI
Last Suppressed Timestamp	${last_supressed_timestamp}	Displays the timestamp of the last suppressed severity.	DX OI
Level	${level}	Displays the level.	UIM, Third-party Products
Location	${location}	Displays the location from which the alarm is generated.	Spectrum
Location String	${locationString}	Displays the location.	Spectrum
Maintenance	${maintenance}	Displays if the maintenance mode is on or off.	Spectrum
Management Module	${management_module}	Displays the management module in DX APM.	DX APM
Message	${message}	Displays the time and date when the alarm was last updated.	All
Metric Id	${met_id}	Displays the metric ID.	UIM
Metric External ID	${metric_external_id}	Displays the external ID of the metric.	DX APM
Metric Family	${metric_family}	Displays the metric family.	DX OI (Prediction Alarms)
Metric Group ID	${metric_group_id}	Displays the group ID of the metric.	
Metric Name	${metric_name}	Displays the name of the metric that is involved in the alarm.	ADA, Custom, UIM

Text	Payload Key	Description	Product
Metric Type	${metric_type}	Displays the type of metric involved in the alarm.	UIM, Third-party Products
Metric Unit	${metric_unit}	Displays the unit of the metric that is involved in the alarm.	UIM, Third-party Products
Metric Value	${metric_value}	Displays the value of the metric that is involved in the alarm.	UIM, Third-party Products
Metrics Index	${metrics_index}	Displays the metrics index.	DX OI
Model Name	${modelName}	Displays the name of the modeled device.	Spectrum
Model Type Handle	${modelTypeHandle}	Displays the model type flag (Visible, Instantiable, and Derivable, No Destroy, Unique, and Required).	Spectrum
Notification Data	${notificationData}	Displays the data that is sent with the alarm notification.	Spectrum
Occurrence	${occurrence}	Displays the occurrence of the event.	UIM
Origin	${origin}	Displays the UIM origin of the alarm.	UIM
Parent Device Type	${parentDeviceType}	Displays the type of the parent device.	DX OI (Prediction Alarm)
Predicted Day	${predictedDay}	Displays the predicted day.	DX OI (Prediction Alarm)
Predicted Value	${predictedValue}	Displays the predicted value.	DX OI (Prediction Alarm)
Prediction Category	${predictionCategory}	Displays the predicted category.	DX OI (Prediction Alarm)
Prediction Timestamp	${prediction_timestamp}	Displays timestamp of the prediction.	DX OI (Prediction Alarm)
Probable Cause	${probableCause}	Displays the probable cause.	Spectrum
Probe	${probe}	Displays the probe name which is generating the alarm/ notification.	UIM
Process Name	${agent_process}	Displays the name of the agent process.	DX APM
Product	${product}	Displays the product from which the alarm is generated. For example, *UIM*.	All
Product Id	${product_id}	Displays the product ID.	All
Product Version	${product_version}	Displays the product version.	All
Robot	${robot}	Displays the UIM robot from which the alarm is generated.	UIM
Role	${role}	Displays the role.	

Text	Payload Key	Description	Product
Rollup Algorithm	${rollup_algorithm}	Displays the rollup algorithm.	DX OI (Prediction Alarm)
Root Cause	${rootCause}	Displays the root cause.	Spectrum
SA Unique Id	${sa_unique_id}	Displays the unique ID of the service alarm.	DX OI
Samples	${samples}	Displays the samples.	DX OI
Service Tags	${service_tags}	Displays the service tags or user tags configured for the services.	DX OI
Services Impacted	${services_impacted}	Displays the service which is impacted by an alarm.	DX OI
Severity	${severity}	Indicates the severity of an alarm. The following severity levels are supported: • Critical • Major • Minor • Informational • Warning	All
Slope	${slope}		DX OI
Source	${source}	Displays the monitored device for which the alarm is flagged. For example, *testdevice1.example.net*	UIM
Spectro Server	${spectroSERVER}	Displays the Spectrum SpectroSERVER name.	Spectrum
Src CIDR	${srcCIDR}	Displays the source CIDR.	ADA
Src Port	${srcPort}	Displays the source port.	ADA
Start Time	${startTime}	Displays the start time.	All
Status	${status}	Displays the status.	All
Subsystem	${subsystem}	Displays the subsystem.	UIM
Subsystem ID	${subsystemID}	Displays the subsystem ID, identifying which part of the system the alarm relates to.	UIM
Summary	${summary}	Displays the summary.	Spectrum, UIM, Third-party Products
Supp Key	${supp_key}	Displays the suppression key.	UIM
Symptoms	${symptoms}	Displays the symptoms.	Spectrum
Tags	${tags}	Displays the tag that is associated with the alarm/ notification.	ADA, Spectrum, UIM, Third-party Products
Threshold	${threshold}	Displays the threshold.	DX OI (Anomaly Alarms)

Text	Payload Key	Description	Product
Time Interval	${time_interval}	Displays the time interval.	DX OI (Anomaly Alarms)
Time to Threshold	${time_to_threshold}	Displays the event violation rule that sent an alarm when a QoS metric is predicted to reach a set value within a user-defined time period.	DX OI (Prediction Alarms)
Timestamp	${timestamp}	Displays the timestamp.	All
Topology Model Name String	${topologyModelNameString}	Displays the name string of the topology model.	Spectrum
Total Points	${totalPoints}	Displays the total points.	DX OI (Prediction)
Total Suppression Count	${total_suppression_count}	Displays the total suppression count.	DX OI (Anomaly)
Trend	${trend}	Displays the trend.	DX OI (Prediction)
Troubleshooter Name	${troubleShooterName}	Displays the Spectrum Troubleshooter ID, associated with the alarm/ notification.	All
Trouble Ticket	${troubleTicket}	Displays the trouble ticket.	All
User Clearable	${userClearable}	Displays whether the alarm is user-clearable.	Spectrum
User Tag 1	${user_tag1}	Displays the custom user tag (specified in Spectrum) associated with the alarm.	UIM
User Tag 2	${user_tag2}	Displays the custom user tag (specified in Spectrum) associated with the alarm	UIM
Version	${version}	Displays the version.	DX OI
Vertex Attributes	${vertex_attributes}	Displays the vertex attributes.	DX APM
Vertex Id	${vertex_id}	Displays the vertex ID.	DX APM
Visible	${visible}		UIM

Supported Payload Keys for Service Alarms

The following table lists all the filter attributes that are supported for Service Alarm:

Text	Key	Description	Product
Acknowledged	${acknowledged}	Indicates whether an alarm has been acknowledged.	Spectrum
Alarm Type	${alarmType}	Displays the alarm type. **Values:** Anomaly, Application, Fault	DX APM, UIM, Spectrum, Third-party Products For UIM, Spectrum, and Third-party Products, DX OI populates this field.
Alarms	${alarms}	Displays the alarms.	DX OI
Annotation	${annotation}	Displays the annotation.	UIM

Text	Key	Description	Product
Closed Timestamp	${closedtimestamp}	Displays the closed timestamp.	
Health Message	${health_message}	Displays the health message.	
Last Update On Alarm Timestamp	${lastupdate_onalarm_timestamp}	Displays the last update on the alarm timestamp.	
Maintenance	${maintenance}	Displays if the maintenance mode is on or off. **Values:** true, false.	Spectrum
Message	${message}	Displays the time and date when the alarm was last updated.	All
Metric Name	${metric_name}	Displays the name of the metric that is involved in the alarm.	ADA, Custom, UIM
Network Rca	${network_rca}		
Root Cause	${rootCause}	Displays the root cause.	Spectrum
Root Cause Alarm URL	${root_cause_alarm_url}	Displays the alarm URL of the root cause.	
Root Cause Alarm Index	${rootCauseAlarmIndex}	Displays the alarm index of the root cause.	DX OI
Root Cause Host	${rootCauseHost}	Displays the host of the root cause.	DX OI
Root Cause Source	${rootCauseSource}	Displays the source of the root cause.	DX OI
Root Cause Update Timestamp	${rootcause_update_timestamp}	Displays the timestamp when the update was made to the root cause.	DX OI
SA Unique Id	${sa_unique_id}	Displays the unique ID of the service alarm.	DX OI
Service Alarm URL	${service_alarm_url}	Displays the service alarm URL.	DX OI
Service Id	${service_id}	Displays the service ID.	DX OI
Service Name	${service_name}	Displays the service name.	DX OI
Services Impacted	${services_impacted}	Displays the service which is impacted by an alarm.	DX OI
Severity	${severity}	Indicates the severity of an alarm. The following severity levels are supported: • Critical • Major • Minor • Informational • Warning	All

Text	Key	Description	Product
Start Timestamp	${starttimestamp}	Displays the start timestamp.	DX OI
Status	${status}	Displays the status.	All
Timestamp	${timestamp}	Displays the timestamp.	All
Troubleshooter Name	${troubleShooterName}	Displays the Spectrum Troubleshooter ID, associated with the alarm/ notification.	All
Trouble Ticket	${troubleTicket}	Displays the trouble ticket.	All
Trouble Ticket URL	${troubleTicketUrl}	Displays the URL for the trouble ticket.	
Version	${version}	Displays the version.	DX OI
Visible	${visible}		UIM

Supported Payload Keys for Situations

The following table lists all the payload keys that are supported for situations:

Text	Key	Description	Product
Age (In min)	${age}	Displays the age of the alarm.	
Alarms Count	${alarmsCount}	Displays the alarm count.	
Alarm Type	${alarmType}	Displays the alarm type.	UIM (Now extended to all product alarms in DX OI to indicate the alarm type. Fault for Spectrum, Anomaly for anomaly alarms, Application Performance Management for APM alarms, Service for Service Alarms, and so on.)
Closed Products	${closed_products}	Displays the closed products.	
Closure Ts	${closureTs}		
Cluster Filter	${cluster_filter}	Displays the filter.	
Cluster Id	${clusterId}	Displays the cluster ID.	
Custom*n* *where n represents numbers 1 and 2.*	${customn}	You can specify up to 10 custom attributes.	
First Alarm Start Time	${firstAlarmStartTime}	Displays the start time of the first alarm.	
Hosts	${hosts}	Displays the name of the host.	
Initial Impacted Host	${initialImpactedHost}	Displays the host that was impacted first.	
Initial Impacted Service	${InitialImpactedServices}	Displays the services that were impacted first.	

Text	Key	Description	Product
Is Closed	${isClosed}	Displays if the alarm is closed.	
Is Orphan	${isOrphan}	Displays if the alarm is an orphan.	
Is Stable	${isStable}	Displays if the alarm is stable.	
Last Alarm Timestamp	${lastAlarmTimestamp}	Displays the timestamp of the last alarm.	
Most Impacted Host	${mostImpactedHost}	Displays the host that was most impacted.	
Most Impacted Service	${mostImpactedServices}	Displays the service that was most impacted.	
Name	${name}	Displays the name.	
Noise Flag	${noiseFlag}	Displays the noise flag.	
Primary Root Cause Host	${rc_host}	Displays the host of the primary root cause.	
Primary Root Cause Service	${rc_services_impacted}	Displays the primary root cause service that was impacted.	
Primary Root Cause Source	${rc_products}	Displays the source of the primary root cause.	
Products	${products}	Displays the product.	
RCA Scores	${rca_scores}	Displays the RCA scores.	DX OI
Root Cause Count	${rootCauseCount}	Displays the product from which the alarm is generated. For example, *UIM*.	
Root Cause Message	${rc_name}	Displays the message of the root cause.	
Root Cause Relative Confidence	${rc_confidence_score}	Displays the relative confidence of the root cause.	
Root Cause Severity	${rc_severity}	Displays the severity of the root cause.	
Root Cause Sub Cluster Id	${rc_subClusterId}	Displays the sub cluster ID.	
Services Impacted	${services_impacted}	Displays the service which is impacted by an alarm.	DX OI
Severity	${severity}	Indicates the severity of an alarm. The following severity levels are supported: • Critical • Major • Minor • Informational • Warning	All
Situations URL	${situations_url}	Displays the situations URL.	
Stable Time	${stableTime}	Displays the stable time.	All

Text	Key	Description	Product
Start Time	${startTime}	Displays the start time.	All
Status	${status}	Displays the status of the alarm/ notification.	All
Sub Clusters Count	${subClustersCount}	Displays the count of the sub clusters.	
Timestamp	${timestamp}	Displays the timestamp of the alarm.	All
Troubleshooter Name	${troubleShooterName}	Displays the troubleshooters name.	All
Trouble Ticket	${troubleTicket}	Displays the trouble ticket.	All
Trouble Ticket URL	${troubleTicketUrl}	Displays the trouble ticket URL.	
Unique ID	${unique_id}	Displays the unique ID.	

Configure Slack Channel

Tenant Administrators can create webhook channels that post messages into Slack.

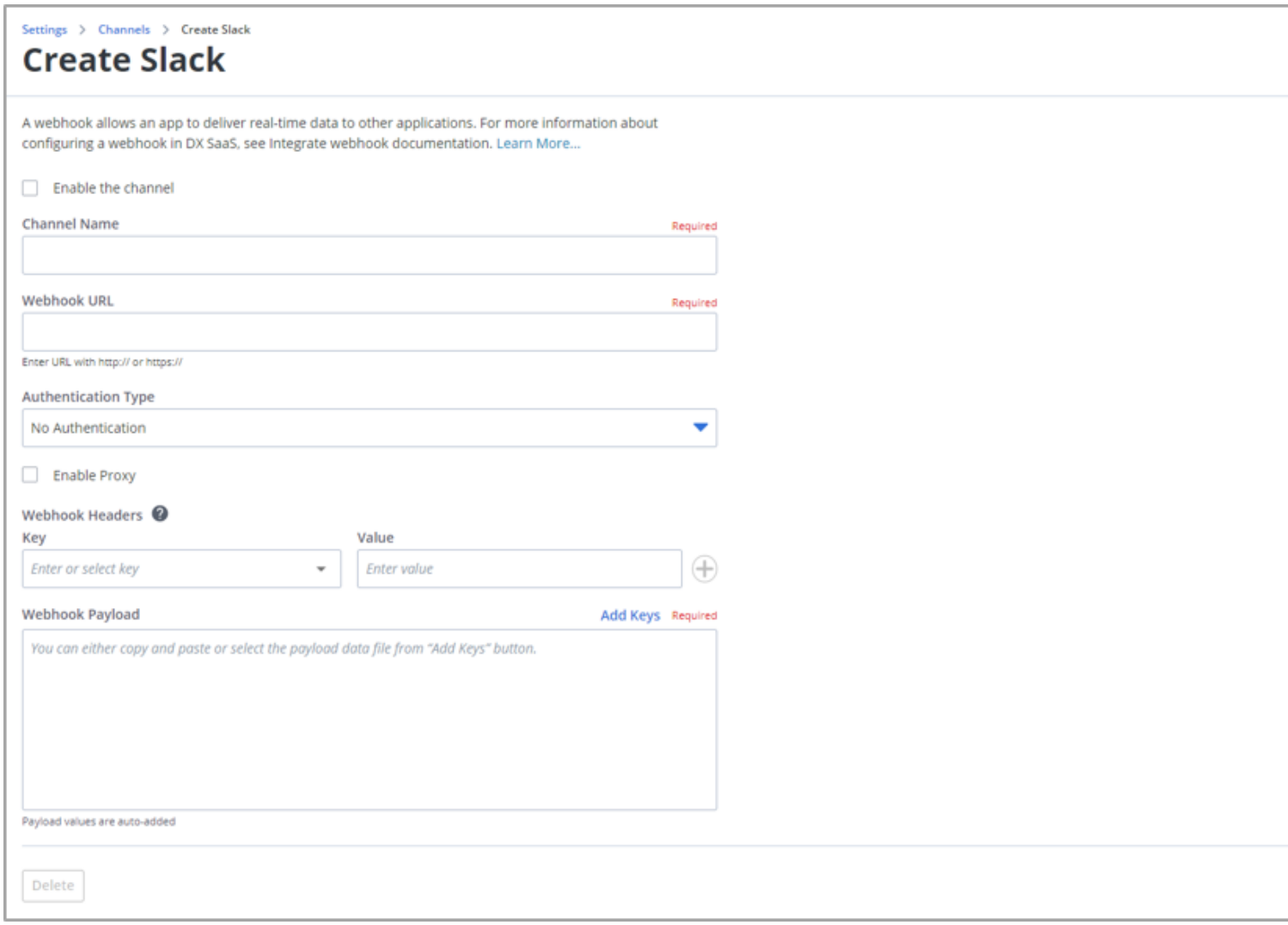
{"URL":["https://cloudmanagement/#/settings/notifications/channel/webhook/slack"],"description":"task.dita_6ecd8948-ce6b-4b80-9857-c35f8dc3c2f8","new":"","new_video":"","admin":"","troubleshooting":{"masterkb":"","text":"","URL":[]},"pendo":"","video":[]}

> **NOTE**
> When you create or edit a Webhook channel, the changes take some time to reflect.

To configure slack as a channel, follow these steps:

1. Log in to DX Operational Intelligence as a Tenant Administrator and click Settings in the left navigation pane.
   The application displays the Settings page.
2. Click **Connect** in the Notification Channel Tile.
   The application displays the **Channels** page with the list of existing channels.
3. Click

   and select Slack from the Select Channels Type drop-down.
   The Create Slack page appears.



4. Select the **Enable the Channel** checkbox.
   This action enables the connection between the DX Operational Intelligence and the channel.
5. Enter a unique channel name in the Channel Name field.
6. Provide the unique Webhook URL to which you intend to send the JSON payload in the Webhook URL field.
7. Select one of the following Authentication Types:
   - **No Authentication:** Select this option, if no authentication is required to connect with Webhooks.
   - **Basic Authentication:** Select this option and provide the user credentials. DX Operational Intelligence uses these credentials to establish the connection with Webhook.
   - **Token Authentication:** Select this option and provide the security token that you are using for authentication with the endpoint in the Token field.

8. Select **Enable Proxy** to route the connection through a proxy server.
9. Define Webhook Headers and Payload in the Webhook Headers section.
   a) **Key and Value:** Specify one or more key-value pairs in the Key and Value fields. You can also select the following keys and provide the values: Accept, Accept-Encoding, Cache-Control, and Content-Type. Add or delete the key-value pairs.
   b) **Webhook Payload:** Click **Add Keys** and select the attributes or you can paste the payload directly in the textbox.

      When you click **Add Keys**, the attributes are listed based on the alarm type. However, for the CI Attributes option to display the attributes, you must register the CI attributes. For more information about how to register the CI Attributes, see the Add or Update the CI Attributes section.

      The application adds the selected keys and the payload values in the Webhook Payload text box. For more information about the supported payload keys, see the Supported Payload Keys section.
10. Click **Test**.

    The application validates the channel configuration.
11. Click **Save**.

    DX Operational Intelligence completes the configuration and enables the integration with the selected Channel.

Supported Payload Keys

The payload keys that you can add are categorized based on the alarm type:

- All Alarm
- Service Alarm
- Situations

Supported Payload Keys for All Alarm

The following table lists all the filter attributes that are supported for All Alarm:

Text	Payload Key	Description	Product
APM Alarm Unique Id	${apm_alarm_unique_id}	Displays the unique ID of the alarm.	DX OI (Applicable only to the DX APM alarms)
Acknowledged	${acknowledged}	Indicates whether an alarm has been acknowledged.	Spectrum
Action Status	${action_status}	Displays the status of the action.	DX OI
Agent	${agent}	Displays the name of the DX APM agent.	DX APM
Alarm Age	${alarmAge}	Displays the age of the alarm.	Spectrum
Alarm Description	${alarm_description}	Displays the alarm description.	DX APM
Alarm Domain	${alarm_domain}	Displays the domain from which the alarm is generated.	DX APM
Alarm Name	${alarm_name}	Displays the name of the alarm.	DX APM
Alarm State	${alarmState}	Displays the state of the alarm.	Spectrum

Text	Payload Key	Description	Product
Alarm Type	${alarmType}	Displays the alarm type. **Values:** Anomaly, Application, Fault	DX APM, UIM, Spectrum, Third-party Products For UIM, Spectrum, and Third-party Products, DX OI populates this field.
Alarm URL	${alarmURL}	Displays the alarm URL.	Spectrum, UIM, ADA, Third-party Products
Alarm Unique Id	${alarm_unique_id}	Displays the unique identification number of the alarm.	All products
Alarm Update	${alarm_update}	Displays the alarm update.	DX OI
Alert External Id	${alert_external_id}	Displays the external ID for the alert.	DX APM
Annotation	${annotation}	Displays the annotation.	UIM
Anomaly Algorithm Type	${algorithm}	Displays the algorithm type of the anomaly.	
Application Name	${applicationName}	Indicates the application name in DX APM.	DX APM
Automic Jobs	${automicJobs}		DX OI
Baseline	${baseline}	Displays the baseline.	DX APM
Breached Threshold	${breached_threshold}	Displays the threshold that was breached.	DX APM
CI ID	${ci_id}	Displays the ID of CI (Configuration Item). A CI represents the component being monitored.	UIM
CI Name	${ci_name}	Displays the CI name.	UIM, Third-party Products, CAPM
CI Type	${ci_type}	Displays the CI type.	UIM, Third-party Products
CI Unique ID	${ci_unique_id}	Displays the unique ID of CI.	OI - NFA and VNA Anomalies
CS ID	${cs_id}	Displays the CS (Computer System) ID.	DX OI
CS Key	${cs_key}	Displays the CS key.	UIM
Cause Code	${causeCode}	Displays the alarm cause code which is an 8-digit, hexadecimal code that identifies the probable cause of the alarm.	Spectrum
Caution Threshold	${caution_threshold}	Displays the caution threshold.	DX APM
Channels	${channels}	Displays the channel.	DX APM
Cleared	${cleared}	Displays if cleared or not.	Cleared

Text	Payload Key	Description	Product
Closed Time	${closedTime}	Displays the closed time.	DX OI
Collection Unique Key String	${collectionUniqueKeyString}		Spectrum
Component Name	${component_name}	Displays the component name.	DX APM
Configuration Item	${configuration_item}	Displays the configuration item.	ADA, Third-party Products
Configuration Item Type	${configuration_item_type}	Displays the Configuration Item type.	UIM, Third-party Anomalies
Correlated External Id	${correlated_external_id}	Displays the correlated external ID.	DX OI
Current Trend Value	${currentTrendValue}	Displays the value of the current trend.	DX OI (Prediction Alarms)
Custom_*n* *where n represents numbers 1-10.*	${custom_n}	You can specify up to 10 custom attributes.	Custom_1 to Custom_5: UIM, Third-party Products Custom_6 to Custom_10: Third-party Products
Custom Num n	${custom_num_n}		Third-party Products
Daily Average	${dailyAverage}	Displays the daily average.	DX OI (Prediction)
Danger Threshold	${danger_threshold}	Displays the danger threshold.	DX APM
Dev Id	${dev_id}	Displays the unique identifier of the device that is involved in the alarm.	UIM
Device Global ID	${deviceGlobalID}	Displays the global ID of the device.	CAPM (CAPM Anomalies)
Device Local ID	${deviceLocalID}	Displays the local ID of the device.	CAPM (CAPM Anomalies)
Device Type	${deviceType}	Displays the type of device that is involved in the alarm.	Spectrum, UIM
Device Type Spectrum	${deviceType_spectrum}	Displays the device type in Spectrum.	Spectrum
Distribution Lists	${distributionLists}		DX APM
Doc Type ID	${doc_type_id}	Displays the ID of the doc type.	DX OI
Doc Type Version	${doc_type_version}	Displays the version of the doc type.	DX OI
Domain	${domain}	Displays the domain from which the alarm is generated.	DX OI
Dst Address	${dstAddr}	Displays the destination address.	ADA, NFA
Dst Port	${distPort}	Displays the destination port	ADA
External Id	${external_id}	Displays the external ID.	All

Text	Payload Key	Description	Product
External Ids	${external_ids}	Displays the external IDs.	N/A
Global Id	${globalID}	Displays the global ID of the alarm.	CAPM
Global ID Router	${globalID_router}	Displays the global ID of the router.	NFA
Group	${group}	Displays information about the group or groups to which the device belongs.	ADA, Spectrum, UIM
Group Id	${group_id}	Displays the group Ids to which the device belongs.	UIM
Host	${host}	Displays the host name.	All
Hub	${hub}	Displays the UIM hub from which the alarm is generated.	UIM
Intercept	${intercept}		DX OI
IP	${ip}	Displays the IP address.	All
Landscape ID	${landscapeID}	Displays the Spectrum landscape ID from which the alarm is generated.	Spectrum
Last Suppressed Severity	${last_supressed_severity}	Displays the last suppressed severity.	DX OI
Last Suppressed Timestamp	${last_supressed_timestamp}	Displays the timestamp of the last suppressed severity.	DX OI
Level	${level}	Displays the level.	UIM, Third-party Products
Location	${location}	Displays the location from which the alarm is generated.	Spectrum
Location String	${locationString}	Displays the location.	Spectrum
Maintenance	${maintenance}	Displays if the maintenance mode is on or off.	Spectrum
Management Module	${management_module}	Displays the management module in DX APM.	DX APM
Message	${message}	Displays the time and date when the alarm was last updated.	All
Metric Id	${met_id}	Displays the metric ID.	UIM
Metric External ID	${metric_external_id}	Displays the external ID of the metric.	DX APM
Metric Family	${metric_family}	Displays the metric family.	DX OI (Prediction Alarms)
Metric Group ID	${metric_group_id}	Displays the group ID of the metric.	
Metric Name	${metric_name}	Displays the name of the metric that is involved in the alarm.	ADA, Custom, UIM

Text	Payload Key	Description	Product
Metric Type	${metric_type}	Displays the type of metric involved in the alarm.	UIM, Third-party Products
Metric Unit	${metric_unit}	Displays the unit of the metric that is involved in the alarm.	UIM, Third-party Products
Metric Value	${metric_value}	Displays the value of the metric that is involved in the alarm.	UIM, Third-party Products
Metrics Index	${metrics_index}	Displays the metrics index.	DX OI
Model Name	${modelName}	Displays the name of the modeled device.	Spectrum
Model Type Handle	${modelTypeHandle}	Displays the model type flag (Visible, Instantiable, and Derivable, No Destroy, Unique, and Required).	Spectrum
Notification Data	${notificationData}	Displays the data that is sent with the alarm notification.	Spectrum
Occurrence	${occurrence}	Displays the occurrence of the event.	UIM
Origin	${origin}	Displays the UIM origin of the alarm.	UIM
Parent Device Type	${parentDeviceType}	Displays the type of the parent device.	DX OI (Prediction Alarm)
Predicted Day	${predictedDay}	Displays the predicted day.	DX OI (Prediction Alarm)
Predicted Value	${predictedValue}	Displays the predicted value.	DX OI (Prediction Alarm)
Prediction Category	${predictionCategory}	Displays the predicted category.	DX OI (Prediction Alarm)
Prediction Timestamp	${prediction_timestamp}	Displays timestamp of the prediction.	DX OI (Prediction Alarm)
Probable Cause	${probableCause}	Displays the probable cause.	Spectrum
Probe	${probe}	Displays the probe name which is generating the alarm/ notification.	UIM
Process Name	${agent_process}	Displays the name of the agent process.	DX APM
Product	${product}	Displays the product from which the alarm is generated. For example, *UIM*.	All
Product Id	${product_id}	Displays the product ID.	All
Product Version	${product_version}	Displays the product version.	All
Robot	${robot}	Displays the UIM robot from which the alarm is generated.	UIM
Role	${role}	Displays the role.	

Text	Payload Key	Description	Product
Rollup Algorithm	${rollup_algorithm}	Displays the rollup algorithm.	DX OI (Prediction Alarm)
Root Cause	${rootCause}	Displays the root cause.	Spectrum
SA Unique Id	${sa_unique_id}	Displays the unique ID of the service alarm.	DX OI
Samples	${samples}	Displays the samples.	DX OI
Service Tags	${service_tags}	Displays the service tags or user tags configured for the services.	DX OI
Services Impacted	${services_impacted}	Displays the service which is impacted by an alarm.	DX OI
Severity	${severity}	Indicates the severity of an alarm. The following severity levels are supported: • Critical • Major • Minor • Informational • Warning	All
Slope	${slope}		DX OI
Source	${source}	Displays the monitored device for which the alarm is flagged. For example, *testdevice1.example.net*	UIM
Spectro Server	${spectroSERVER}	Displays the Spectrum SpectroSERVER name.	Spectrum
Src CIDR	${srcCIDR}	Displays the source CIDR.	ADA
Src Port	${srcPort}	Displays the source port.	ADA
Start Time	${startTime}	Displays the start time.	All
Status	${status}	Displays the status.	All
Subsystem	${subsystem}	Displays the subsystem.	UIM
Subsystem ID	${subsystemID}	Displays the subsystem ID, identifying which part of the system the alarm relates to.	UIM
Summary	${summary}	Displays the summary.	Spectrum, UIM, Third-party Products
Supp Key	${supp_key}	Displays the suppression key.	UIM
Symptoms	${symptoms}	Displays the symptoms.	Spectrum
Tags	${tags}	Displays the tag that is associated with the alarm/ notification.	ADA, Spectrum, UIM, Third-party Products
Threshold	${threshold}	Displays the threshold.	DX OI (Anomaly Alarms)

Text	Payload Key	Description	Product
Time Interval	${time_interval}	Displays the time interval.	DX OI (Anomaly Alarms)
Time to Threshold	${time_to_threshold}	Displays the event violation rule that sent an alarm when a QoS metric is predicted to reach a set value within a user-defined time period.	DX OI (Prediction Alarms)
Timestamp	${timestamp}	Displays the timestamp.	All
Topology Model Name String	${topologyModelNameString}	Displays the name string of the topology model.	Spectrum
Total Points	${totalPoints}	Displays the total points.	DX OI (Prediction)
Total Suppression Count	${total_suppression_count}	Displays the total suppression count.	DX OI (Anomaly)
Trend	${trend}	Displays the trend.	DX OI (Prediction)
Troubleshooter Name	${troubleShooterName}	Displays the Spectrum Troubleshooter ID, associated with the alarm/ notification.	All
Trouble Ticket	${troubleTicket}	Displays the trouble ticket.	All
User Clearable	${userClearable}	Displays whether the alarm is user-clearable.	Spectrum
User Tag 1	${user_tag1}	Displays the custom user tag (specified in Spectrum) associated with the alarm.	UIM
User Tag 2	${user_tag2}	Displays the custom user tag (specified in Spectrum) associated with the alarm	UIM
Version	${version}	Displays the version.	DX OI
Vertex Attributes	${vertex_attributes}	Displays the vertex attributes.	DX APM
Vertex Id	${vertex_id}	Displays the vertex ID.	DX APM
Visible	${visible}		UIM

Supported Payload Keys for Service Alarms

The following table lists all the filter attributes that are supported for Service Alarm:

Text	Key	Description	Product
Acknowledged	${acknowledged}	Indicates whether an alarm has been acknowledged.	Spectrum
Alarm Type	${alarmType}	Displays the alarm type. **Values:** Anomaly, Application, Fault	DX APM, UIM, Spectrum, Third-party Products For UIM, Spectrum, and Third-party Products, DX OI populates this field.
Alarms	${alarms}	Displays the alarms.	DX OI
Annotation	${annotation}	Displays the annotation.	UIM

Text	Key	Description	Product
Closed Timestamp	${closedtimestamp}	Displays the closed timestamp.	
Health Message	${health_message}	Displays the health message.	
Last Update On Alarm Timestamp	${lastupdate_onalarm_timestamp}	Displays the last update on the alarm timestamp.	
Maintenance	${maintenance}	Displays if the maintenance mode is on or off. **Values:** true, false.	Spectrum
Message	${message}	Displays the time and date when the alarm was last updated.	All
Metric Name	${metric_name}	Displays the name of the metric that is involved in the alarm.	ADA, Custom, UIM
Network Rca	${network_rca}		
Root Cause	${rootCause}	Displays the root cause.	Spectrum
Root Cause Alarm URL	${root_cause_alarm_url}	Displays the alarm URL of the root cause.	
Root Cause Alarm Index	${rootCauseAlarmIndex}	Displays the alarm index of the root cause.	DX OI
Root Cause Host	${rootCauseHost}	Displays the host of the root cause.	DX OI
Root Cause Source	${rootCauseSource}	Displays the source of the root cause.	DX OI
Root Cause Update Timestamp	${rootcause_update_timestamp}	Displays the timestamp when the update was made to the root cause.	DX OI
SA Unique Id	${sa_unique_id}	Displays the unique ID of the service alarm.	DX OI
Service Alarm URL	${service_alarm_url}	Displays the service alarm URL.	DX OI
Service Id	${service_id}	Displays the service ID.	DX OI
Service Name	${service_name}	Displays the service name.	DX OI
Services Impacted	${services_impacted}	Displays the service which is impacted by an alarm.	DX OI
Severity	${severity}	Indicates the severity of an alarm. The following severity levels are supported: • Critical • Major • Minor • Informational • Warning	All

Text	Key	Description	Product
Start Timestamp	${starttimestamp}	Displays the start timestamp.	DX OI
Status	${status}	Displays the status.	All
Timestamp	${timestamp}	Displays the timestamp.	All
Troubleshooter Name	${troubleShooterName}	Displays the Spectrum Troubleshooter ID, associated with the alarm/ notification.	All
Trouble Ticket	${troubleTicket}	Displays the trouble ticket.	All
Trouble Ticket URL	${troubleTicketUrl}	Displays the URL for the trouble ticket.	
Version	${version}	Displays the version.	DX OI
Visible	${visible}		UIM

Supported Payload Keys for Situations

The following table lists all the payload keys that are supported for situations:

Text	Key	Description	Product
Age (In min)	${age}	Displays the age of the alarm.	
Alarms Count	${alarmsCount}	Displays the alarm count.	
Alarm Type	${alarmType}	Displays the alarm type.	UIM (Now extended to all product alarms in DX OI to indicate the alarm type. Fault for Spectrum, Anomaly for anomaly alarms, Application Performance Management for APM alarms, Service for Service Alarms, and so on.)
Closed Products	${closed_products}	Displays the closed products.	
Closure Ts	${closureTs}		
Cluster Filter	${cluster_filter}	Displays the filter.	
Cluster Id	${clusterId}	Displays the cluster ID.	
Custom*n* *where n represents numbers 1 and 2.*	${customn}	You can specify up to 10 custom attributes.	
First Alarm Start Time	${firstAlarmStartTime}	Displays the start time of the first alarm.	
Hosts	${hosts}	Displays the name of the host.	
Initial Impacted Host	${initialImpactedHost}	Displays the host that was impacted first.	
Initial Impacted Service	${InitialImpactedServices}	Displays the services that were impacted first.	

Text	Key	Description	Product
Is Closed	${isClosed}	Displays if the alarm is closed.	
Is Orphan	${isOrphan}	Displays if the alarm is an orphan.	
Is Stable	${isStable}	Displays if the alarm is stable.	
Last Alarm Timestamp	${lastAlarmTimestamp}	Displays the timestamp of the last alarm.	
Most Impacted Host	${mostImpactedHost}	Displays the host that was most impacted.	
Most Impacted Service	${mostImpactedServices}	Displays the service that was most impacted.	
Name	${name}	Displays the name.	
Noise Flag	${noiseFlag}	Displays the noise flag.	
Primary Root Cause Host	${rc_host}	Displays the host of the primary root cause.	
Primary Root Cause Service	${rc_services_impacted}	Displays the primary root cause service that was impacted.	
Primary Root Cause Source	${rc_products}	Displays the source of the primary root cause.	
Products	${products}	Displays the product.	
RCA Scores	${rca_scores}	Displays the RCA scores.	DX OI
Root Cause Count	${rootCauseCount}	Displays the product from which the alarm is generated. For example, *UIM*.	
Root Cause Message	${rc_name}	Displays the message of the root cause.	
Root Cause Relative Confidence	${rc_confidence_score}	Displays the relative confidence of the root cause.	
Root Cause Severity	${rc_severity}	Displays the severity of the root cause.	
Root Cause Sub Cluster Id	${rc_subClusterId}	Displays the sub cluster ID.	
Services Impacted	${services_impacted}	Displays the service which is impacted by an alarm.	DX OI
Severity	${severity}	Indicates the severity of an alarm. The following severity levels are supported: • Critical • Major • Minor • Informational • Warning	All
Situations URL	${situations_url}	Displays the situations URL.	
Stable Time	${stableTime}	Displays the stable time.	All

Text	Key	Description	Product
Start Time	${startTime}	Displays the start time.	All
Status	${status}	Displays the status of the alarm/ notification.	All
Sub Clusters Count	${subClustersCount}	Displays the count of the sub clusters.	
Timestamp	${timestamp}	Displays the timestamp of the alarm.	All
Troubleshooter Name	${troubleShooterName}	Displays the troubleshooters name.	All
Trouble Ticket	${troubleTicket}	Displays the trouble ticket.	All
Trouble Ticket URL	${troubleTicketUrl}	Displays the trouble ticket URL.	
Unique ID	${unique_id}	Displays the unique ID.	

Configure Ticket Management Channels

The ticket management integration enables you to integrate an ITSM product such as ServiceNow and CA Service Management to track and manage alarms raised in DX Operational Intelligence. When an alarm occurs in DX Operational Intelligence, the integration enables ticket creation in the ITSM system and synchronizes the Issue updates in both systems. You can synchronize the following information:

- Status of alarms and associated incidents
- The current assignee (troubleshooter) is assigned to incidents
- Severity of the incidents

DX Operational Intelligence supports integration with the following ITSM ticket management channels:

- BMC Remedy
- CA Service Management
- ServiceNow
- WolkenSoft

Supported Integrations

This section lists the integrations that are supported:

- BMC Remedy
- BMC Helix
- CA Service Management
- ServiceNow
- WolkenSoft

BMC Remedy

The following table lists the DX Operational Intelligence - BMC Remedy integration that is supported:

DX Operational Intelligence	BMC Remedy	Notes
SaaS	**BMC Remedy**	Integrate using DX Gateway. For more information, see the DX Gateway section.
On-Premise	**BMC Remedy**	Integrate directly.

BMC Helix

The following table lists the DX Operational Intelligence - BMC Helix integration that is supported:

DX Operational Intelligence	BMC Helix
SaaS	**BMC Helix**

CA Service Management

The following table lists the DX Operational Intelligence - CA Service Management integration that is supported:

DX Operational Intelligence	CA Service Management	Notes
SaaS	**On-Premise**	Integrate using DX Gateway. For more information, see the DX Gateway section.
On-Premise	**On-Premise**	

ServiceNow

The following table lists the DX Operational Intelligence - ServiceNow integrations that is supported:

DX Operational Intelligence	ServiceNow	Notes
SaaS	**SaaS**	Integrate directly.
On-Prem	**SaaS**	Integrate directly.

WolkenSoft

The following table lists the DX Operational Intelligence - WolkenSoft integrations that are supported:

DX Operational Intelligence	WolkenSoft	Notes
SaaS	**WolkenSoft - SaaS**	Integrate directly.
On-Prem	**WolkenSoft - SaaS**	Integrate directly.

Ticket Enrichment Rules

When a ticket is generated manually, the ticket is updated using the message template that is associated with the channel. And if the ticket is generated automatically, the ticket is updated using the message template that is associated with the policy. If the policy is not associated with any message template, then the message template that is associated with the channel is used.

You can enrich these tickets with additional alarm details using the enrichment rules. Basically, you map the incident fields to the alarm attributes in the enrichment rule and then associate that rule with the channel or policy. Based on this enrichment rule, DX Operational Intelligence fetches the appropriate values for the alarm attributes from the payload and updates the corresponding incident fields in the generated ticket. If the attribute values are not available in the payload, then the incident fields are updated with the default values that you specify in the enrichment rule.

The enriched tickets help you to assign incidents to the right queues, better prioritize, and resolve problems quickly.

This section provides the following information:

- Mapping Guidelines
- Create Mapping Rule

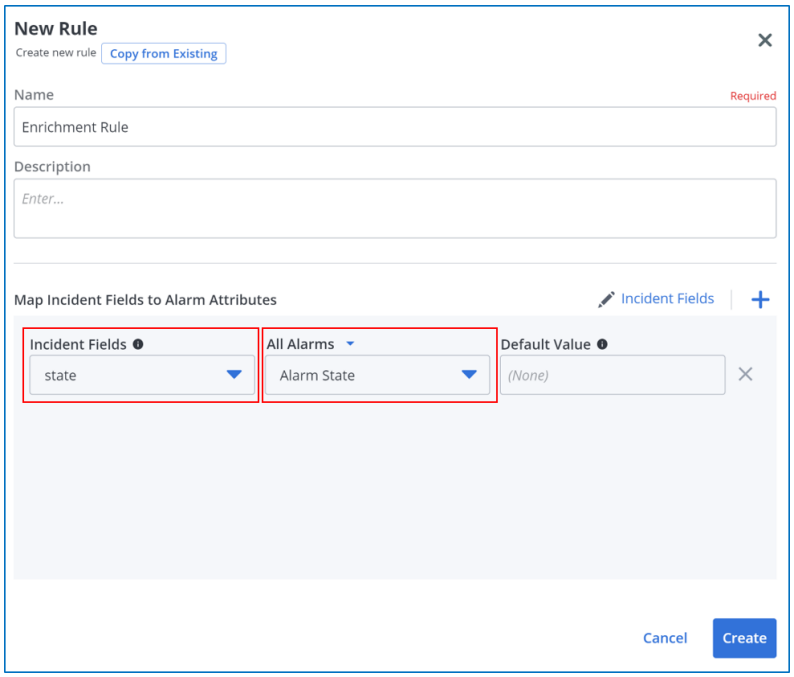
NOTE

- In this section, the word **Enrichment** is used interchangeably with **Mapping**.
- The ticket enrichment rules are supported only for ServiceNow.
- The required incident fields are onboarded to enable you to create the mapping rules. However, only the default fields are available for selection. You can add the other onboarded fields to the selection list if necessary.
- Consider the following points for CI attributes:
  - The CI attributes are supported and are available only for **All Alarms**.
  - The CI attributes are available for selection only if they are registered. You can register these attributes using the API. For more information, see the Add or Update the CI Attributes section.
  - If the CI attribute does not have any value, then the default value for that attribute is displayed in the notification.
  - If the default value is not defined, then the incident field value is blank or the field is not populated.

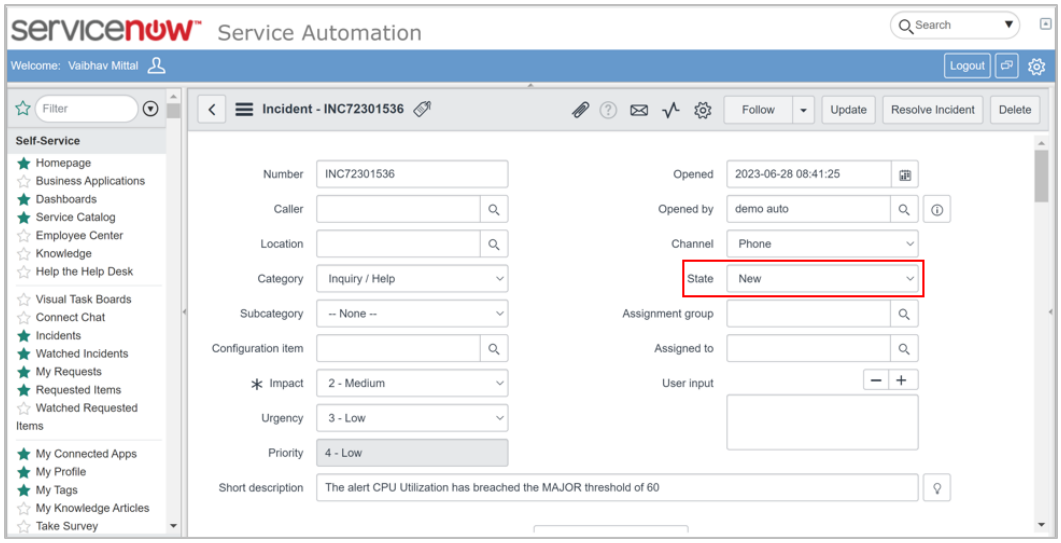
Mapping Guidelines

Follow these guidelines to map the incident fields to the alarm attributes:

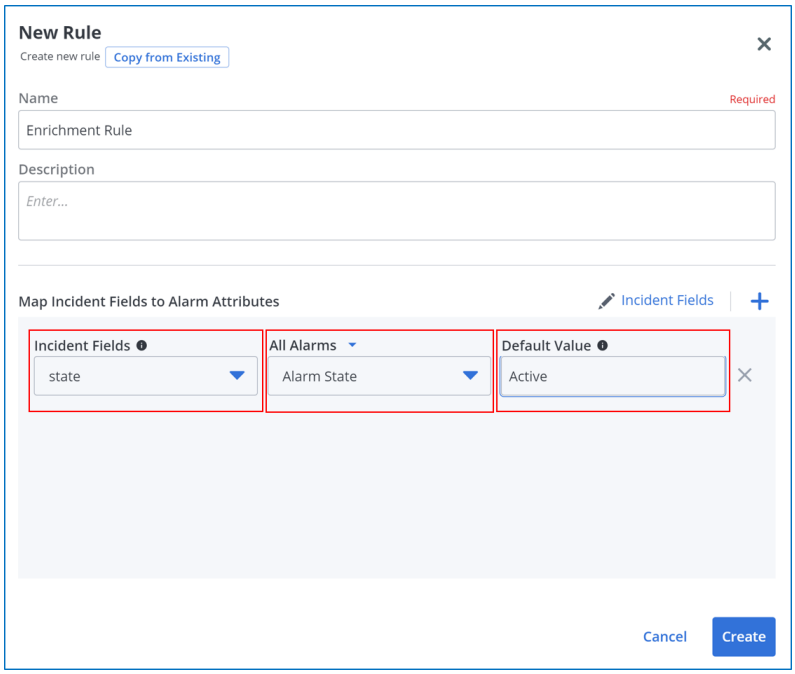
- **The alarm attribute has the value in the payload:** For example, the **Alarm State** attribute has a value in the payload. In this case, map the incident field to that alarm attribute as shown. You may leave the **Default Value** field blank.



When the ticket is generated, the **State** field in the ticket is enriched with the attribute value in the payload.

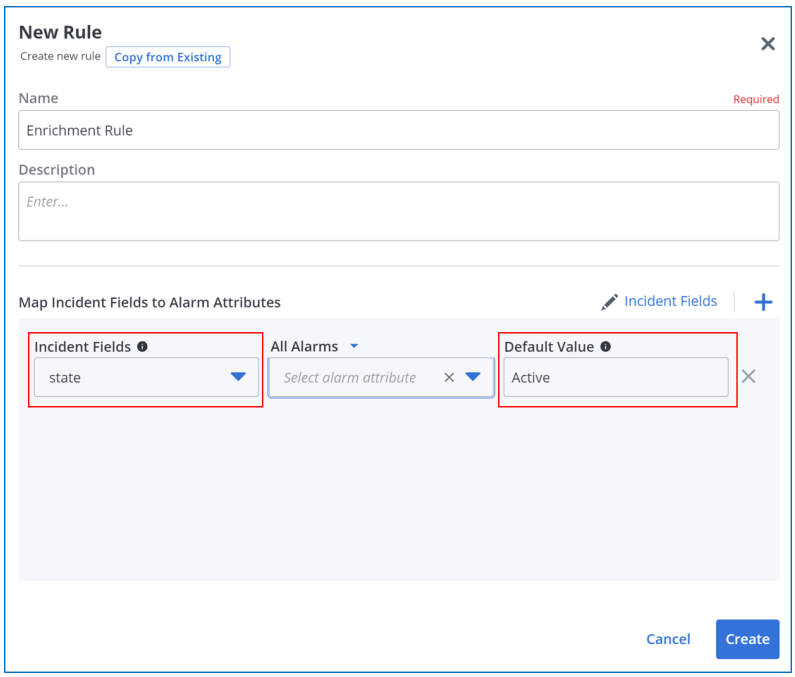


- **The alarm attribute may or may not have any value in the payload:** For example, the Alarm State attribute may or may not have any value in the payload. In this case, map the incident field to that alarm attribute, and also enter the default value as shown.



When the ticket is generated, if the attribute has the value in the payload, then the State field in the ticket is enriched with that value. If the attribute has no value, then the State field is enriched with the specified default value, that is, Active.

- **The alarm attribute does not exist in the payload:** For example, the Alarm State attribute is not in the payload. In this case, you can provide a default value to be mapped to the incident field. Select the incident field to be mapped, leave the **Alarm Attribute** value blank, and enter the default value as shown:



When the ticket is generated, the State field in the ticket is enriched with the specified default value, which is, Active.

Create Mapping Rule

You can create a mapping rule for all alarms, service alarms, or situations on the **Settings > Ticket Enrichment Rules** page.

Prerequisite:

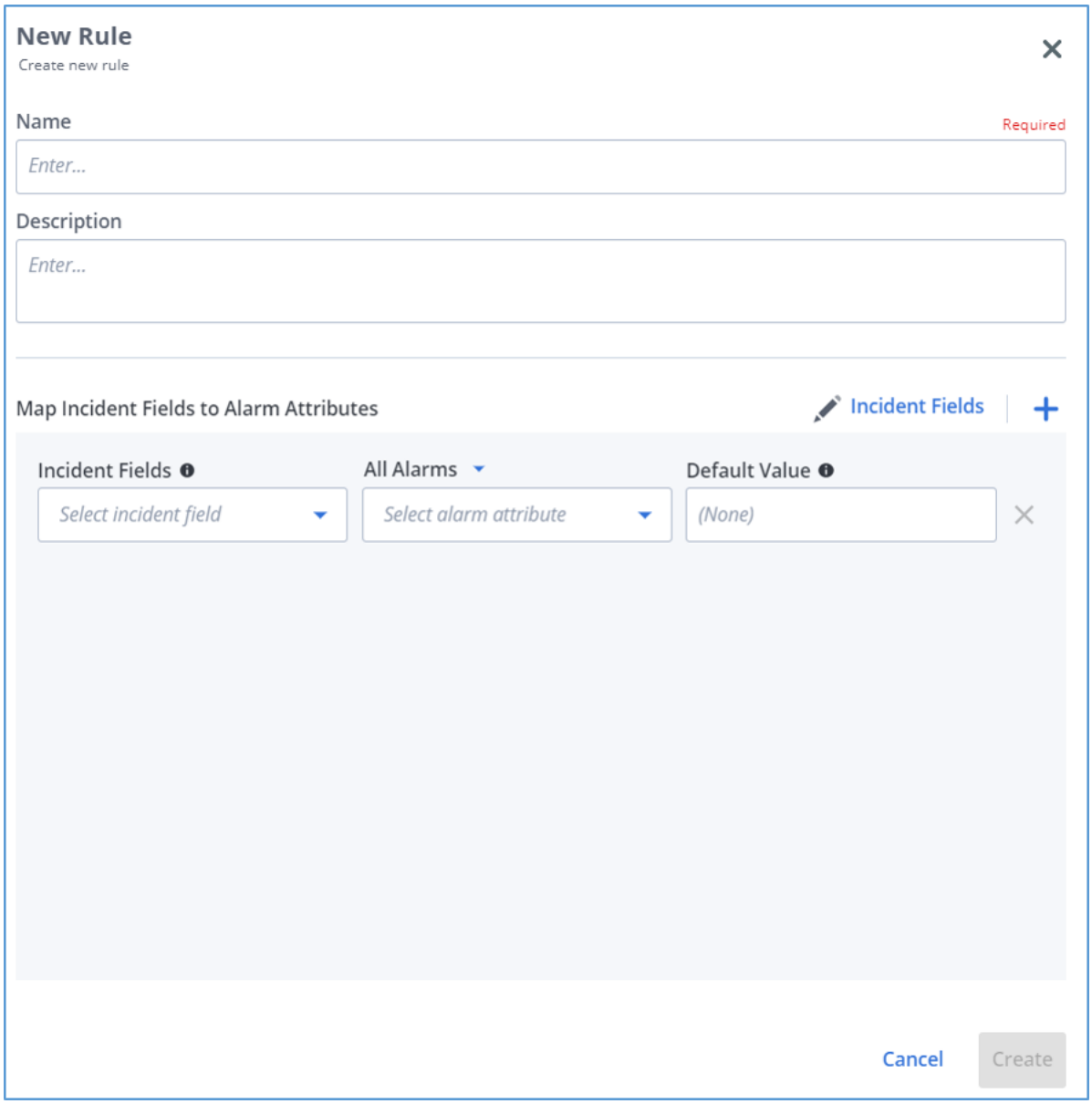
Before you create a mapping rule, ensure that:

- The ServiceNow channel is created.
- You have read the **Mapping Guidelines** section on this page.

Follow these steps:

1. Log into DX SaaS/DX Platform.
2. Click **Settings** in the left navigation pane.
3. Click the **Ticket enrichment rules** tile on the **Settings** page.

   NOTE
   By default, the **Ticket Enrichment Rules** tile is enabled only if the ITSM channel (ServiceNow) is configured. If the channel is not configured, click **Configure now** on the tile or create the channel.
4. Click **+ New Rule**.
   The **New Rule** page is displayed.



5. Provide the following information:
   - **Name:** Enter a name for the mapping rule.
   - **Description:** Enter a description for the rule.
   - **Map Incident Fields to Alarm Attributes**: Select the fields to be mapped.
     - **Incident Fields:** Click the **Incident Fields** drop-down and select the incident field to be mapped.

       NOTE
       By default, the required incident fields are onboarded. All the onboarded fields are displayed in the drop-down list. You can add more fields to this selection list, if necessary. For more information, see the **Add Incident Fields to the Selection List** section on this page.
     - **Alarms:**
       a. Select the **Alarm Category** (All Alarms, Service Alarms, or Situations).

       NOTE
       The corresponding alarm attributes are displayed. The CI attributes are available only for All Alarms.

b. Click to view the list of attributes that are available in the alarm payload.
c. Select the alarm attribute to be mapped. For more information, see the Mapping Guidelines section on this page.

- **Default Value:** Enter the default value if necessary. This value is used if the alarm attribute is not selected or the alarm attribute is unavailable in the alarm payload.

6. Click the **+** icon to add another mapping attribute.

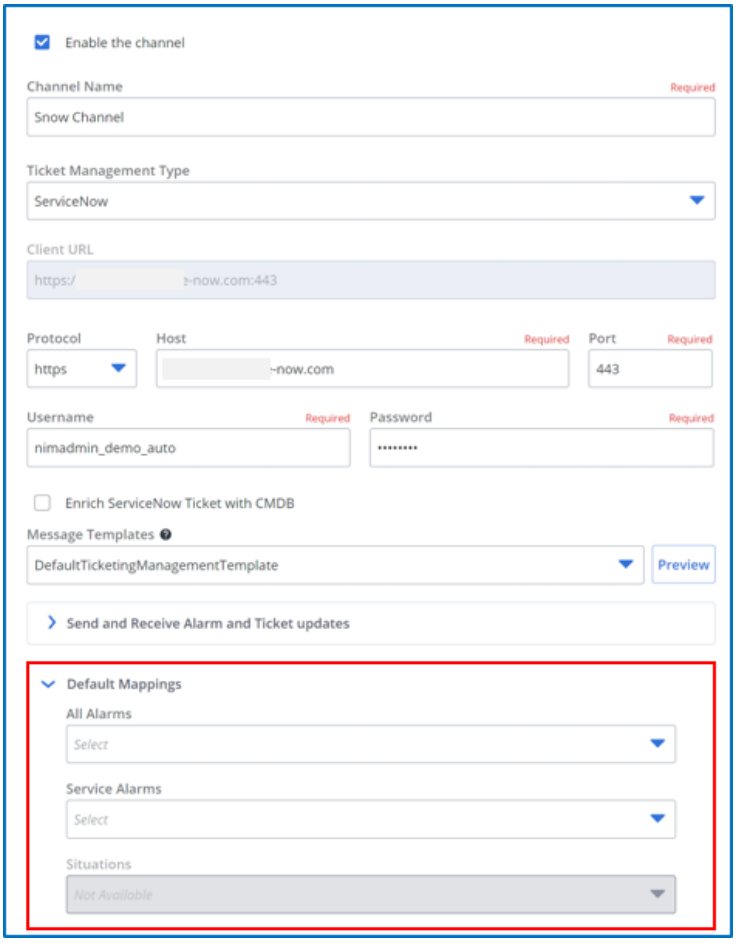
   NOTE
   You can use a field only once in a mapping rule.

7. Click **Create.**

The created mapping rule is created and is added to the list on the **Ticket Enrichment Rules** page. To **Edit**, **Delete**, and **Make a Copy** of the mapping rule, use the **Actions** button on the **Ticket Enrichment Rules** page.

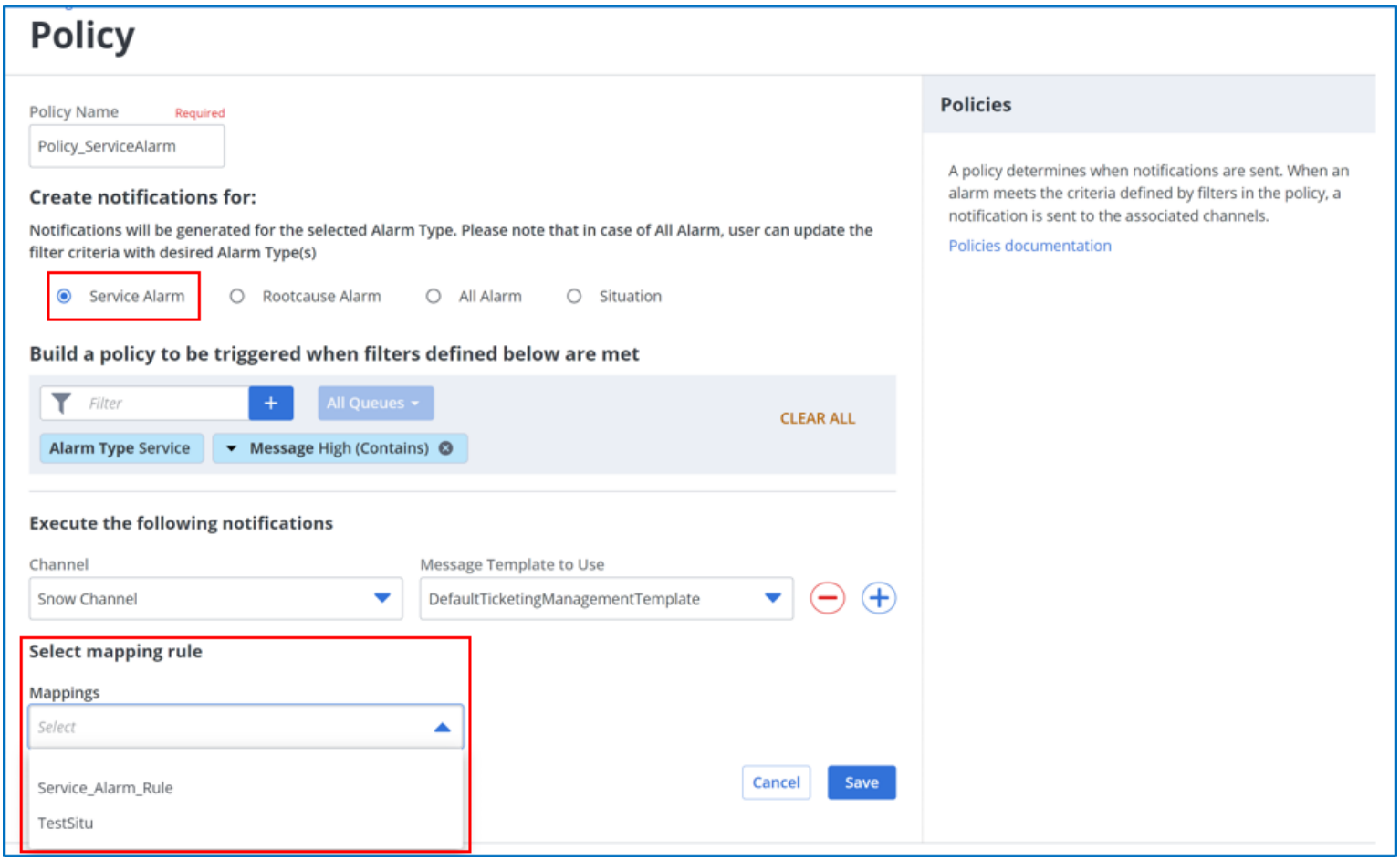
How Ticket Enrichment Works

After you create the enrichment rule, associate the ITSM channel or policy with this rule to enrich the ticket. You can associate an ITSM channel with the enrichment rule in the **Default Mappings** section on the Channels page.



NOTE
The mappings drop-down is enabled only if the mapping rules are available as shown in the illustration.

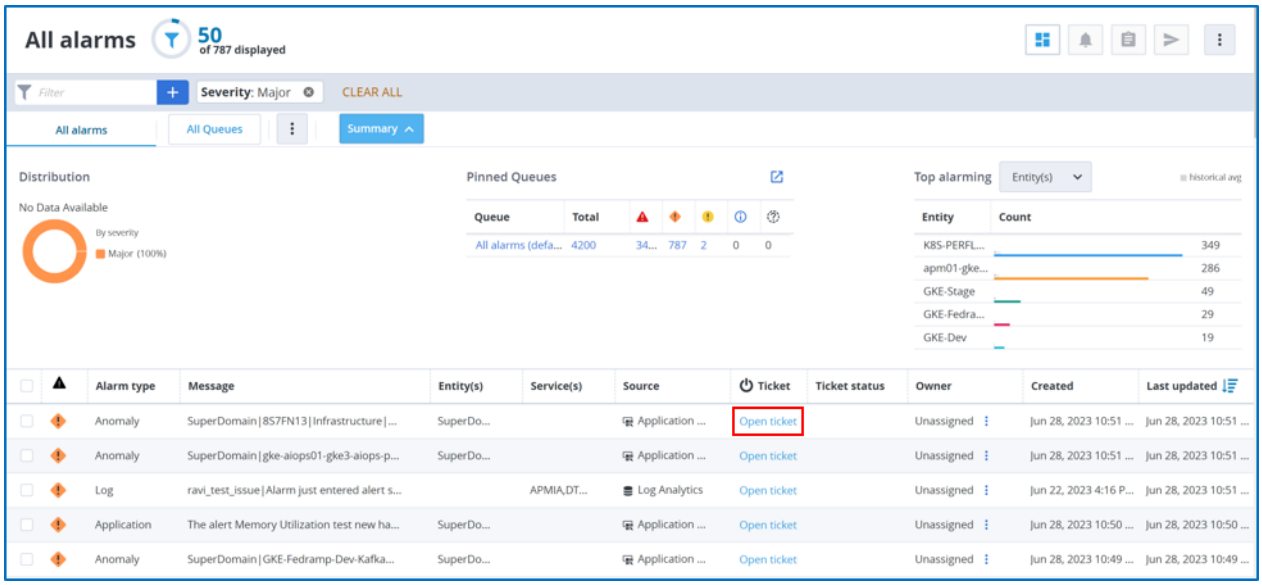
You can associate a policy with the enrichment rule in the **Select Mapping Rule** section on the Policy page. The mappings drop-down lists only the rules that are available for the selected alarm type.



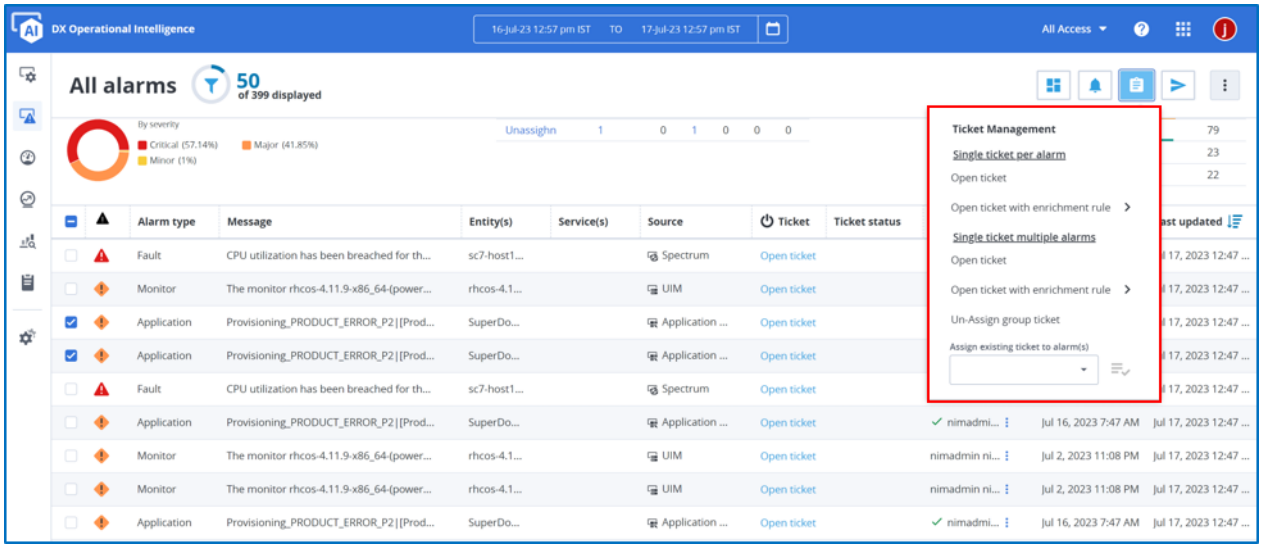
NOTE
This section is disabled if no mapping rules are available.

After you associate with the enrichment rule, the rule is applied in the following ways:

- **Manual Tickets:**
  - **Open Ticket:** If you open the ticket using this option, the mapping rule that is associated with the channel is used for the ticket enrichment. That is, the mapping rule that is selected on the ITSM channel page is used.



– **Ticket Management:** If multiple alarms with a similar root cause are ingested into DX Operational Intelligence, you can now create separate tickets for each of the alarms or you can create a single ticket for all those alarms. The **Ticket Management** icon provides the options as shown in this image:



- **Single ticket per alarm:** Creates a separate ticket for each of the selected alarms.
  – **Open Ticket:** If you open the ticket using this option, the mapping rule that is associated with the channel is used for the ticket enrichment.
  – **Open ticket with enrichment rule:** If you use this option, the mapping rule that you select in this list is used for enrichment instead of the rule that is associated with the channel.
- **Single ticket multiple alarms:** Creates a single ticket for all the selected alarms.
  – **Open Ticket:** If you open the ticket using this option, the mapping rule that is associated with the channel is used for the ticket enrichment.
  – **Open ticket with enrichment rule:** If you use this option, the mapping rule that you select in this list is used for enrichment instead of the rule that is associated with the channel.

- **Automatic Tickets:** After you create an ITSM channel, you must associate this channel with a policy to generate the tickets automatically. When the policy criteria are met, the ticket is created automatically. If the policy is associated with

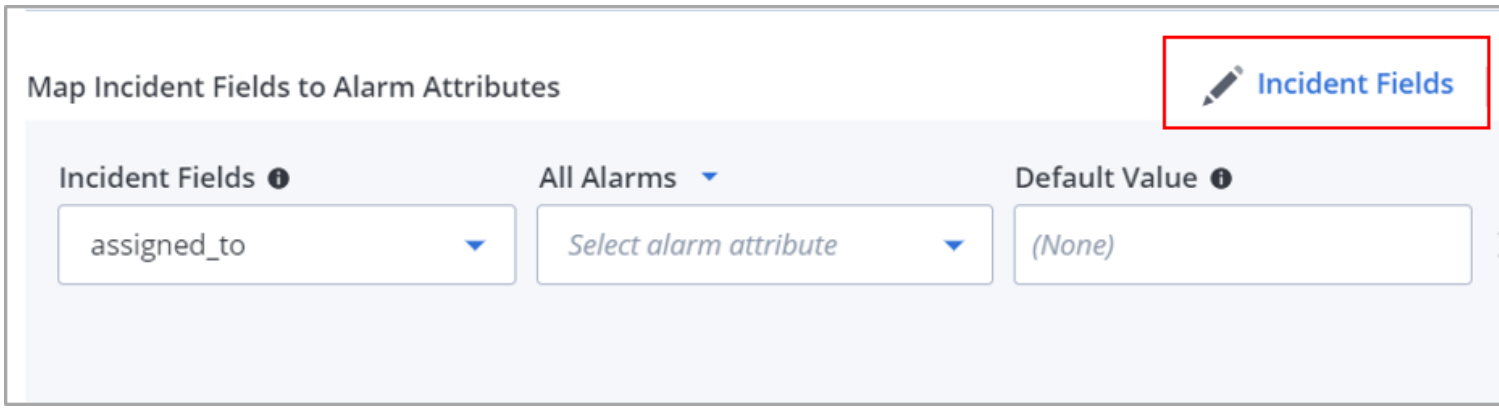
an enrichment rule, then the associated rule is used for the ticket enrichment. If the policy is not associated with any enrichment rule, then the rule that is associated with the channel is used.

Add Incident Fields to the Selection List

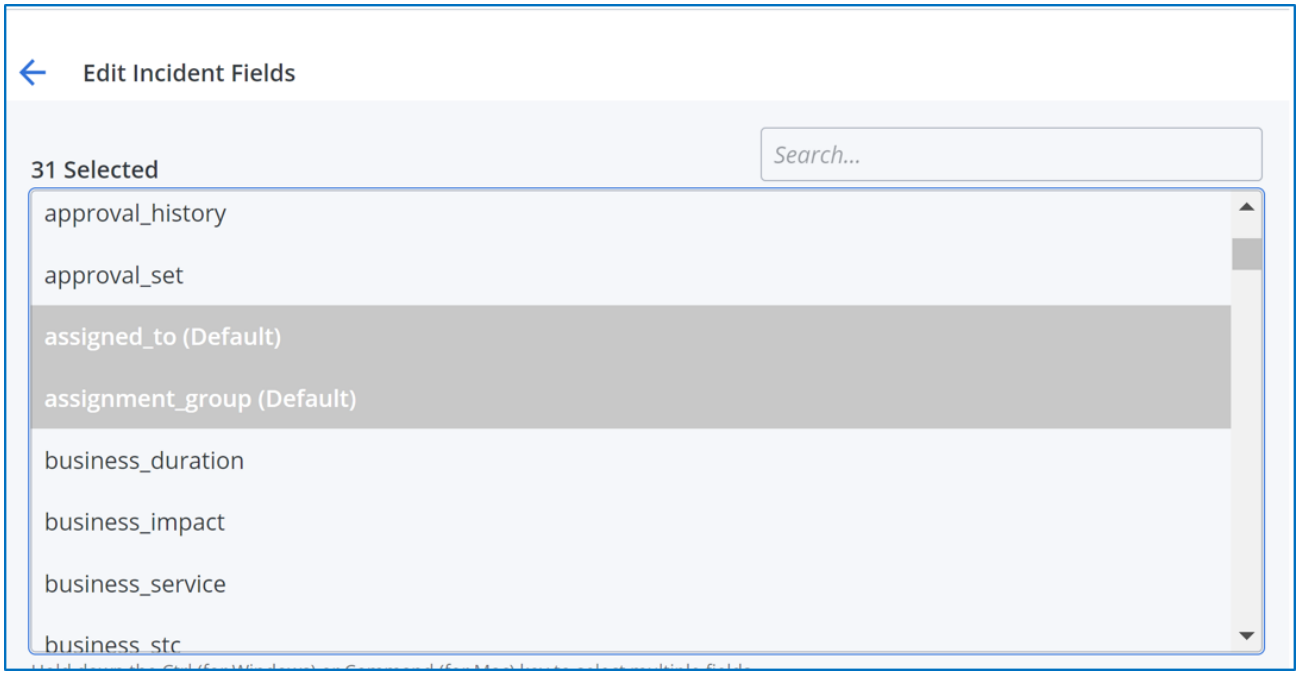
By default, the required incident fields are onboarded and are displayed in the **Incident Fields** drop-down list. You can add more incident fields to this list.

Follow these steps:

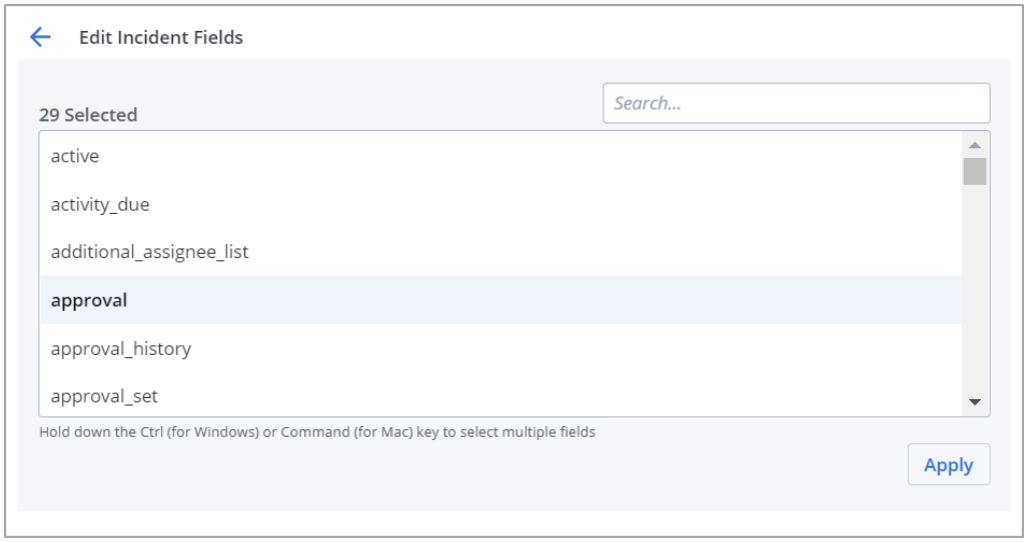
1. Click **Incident Fields** in the **Map Incident Fields to Alarm Attributes** section.



The **Edit Incident Fields** section is displayed. This section lists all the onboarded incident fields. The default fields are marked as (Default). You cannot deselect these fields.



2. Add the incident fields to the selection list.
   - To add a single field, click the field to select.
   - To add multiple fields, hold down the Ctrl (for Windows) or Command (for Mac) key to select.
3. Click **Apply.**
   The incident fields are added to the selection list. In the **Edit Incident Fields** section, the field is highlighted as shown.



You cannot unselect this field here. For more information about how to unselect this field, see the **Delete Incident Field from the Selection List** section on this page.

Delete Incident Field From the Selection List

You cannot delete the default incident fields. You can only delete those incident fields that were added manually to the selection list.

Follow these steps:

1. Delete the incident field from all the mapping rules.
   a. Navigate to the **Ticket Enrichment Rules** page.
   b. Click **Edit** under **Actions** for the rule that has this incident field.
   c. Delete the incident field in the rule.
   d. Save the mapping rule.
   e. Repeat the steps to delete the incident field from all the mapping rules.
2. Delete the incident field from the selection list.
   a. Navigate to the **Ticket Enrichment Rules** page.
   b. Click **New Rule**.
   c. Click the **Edit Incident Fields** icon to view the list.
   d. Press **Ctrl** and click the field that you want to delete.
   e. Click **Apply**.
   The field is deleted and is not displayed in the selection list. You can always add the deleted fields back to the selection list, if necessary.

Make a Copy of Mapping Rule

You can also create a rule from an existing mapping rule using the **Copy from Existing** option. This option is displayed only if the mapping rules are already created.

Follow these steps:

1. Navigate to the **Ticket Enrichment Rules** page.
2. Perform one of the following options:
   – Click **+ New Rule** and click the **Copy from Existing** button.
   – Click the ellipsis icon under **Actions** for the rule that you want to make a copy of.

The **Copy from Existing** panel is displayed.
3. Click to select the rule.
4. Click **Apply**.
5. Edit the rule as required.
6. Click **Create**.
   The mapping rule is created and the rule is added to the list on the **Ticket Enrichment Rules** page.

Delete a Mapping Rule

You can delete a mapping rule only if it is not associated with any channel or policy. If the mapping rule is associated with any channel or policy, you must first disassociate the mapping file, and then delete it.

Follow these steps:

1. Navigate to the **Ticket Enrichment Rules** page.
2. Click **Delete** under **Actions** for the rule that you want to delete.

Integration with ServiceNow

Integration of DX Operational Intelligence with ServiceNow enables you to:

- Create ServiceNow tickets for DX Operational Intelligencealarms manually or automatically (Using policies).
- Maintain the following alarm information between DX Operational Intelligence and its associated ServiceNow ticket:
  - Status of alarms
  - The current assignee (troubleshooter) assigned to tickets
  - The severity of alarms. Currently, severity can be mapped from DX OI to ServiceNow.
  - Annotation Update. If you update Annotation for a service alarm in DX OI, the associated ServiceNow ticket gets updated.
- Launch ServiceNow directly from the DX Operational Intelligence ticket.

Before You Begin ServiceNow Integration
Before you begin the integration with ServiceNow, go through the following information and complete the prerequisites:

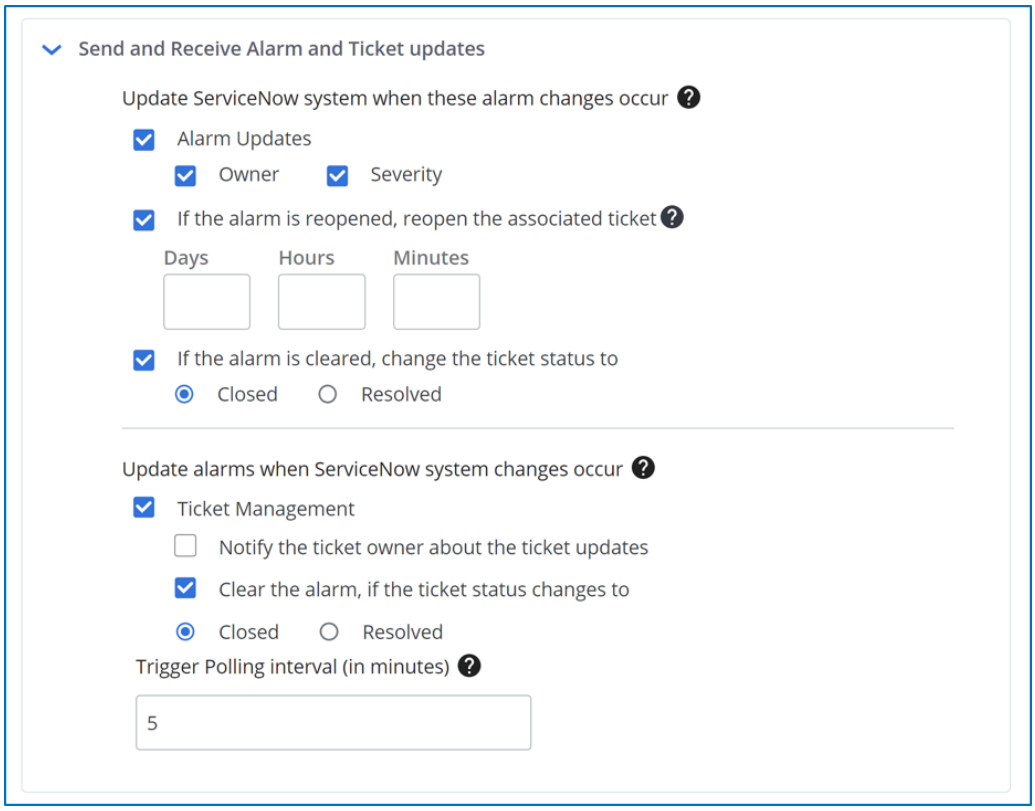
- Automatic Synchronization of Alarm and Ticket Updates
- Set Up ServiceNow

Automatic Synchronization of Alarm and Ticket Updates

Before you start the ServiceNow integration, review this section to understand how you can synchronize the alarm and ticket updates in both systems. You can configure to update any alarm updates or changes in DX Operational Intelligence to reflect in the associated tickets in ServiceNow. Similarly, any updates or changes to the tickets in ServiceNow can be updated in the associated alarms in DX Operational Intelligence.

- Alarm Updates in DX Operational Intelligence
  - Alarm Updates
  - Alarm Reopened
  - Alarm Cleared
- Ticket Updates in ServiceNow
  - Ticket Updates
  - Ticket Status Changes

You can configure this synchronization in the **Send and Receive Alarm and Ticket Updates** section while creating the channel:



Alarm Updates in DX Operational Intelligence

In the **Update ServiceNow system when these alarm changes occur** section, select the required options to send the DX Operational Intelligence alarm updates to the associated ServiceNow ticket:

- Alarm Updates
- Alarm Reopened
- Alarm Cleared

Alarm Updates

- **Owner:** Select this option to update the ticket owner when the alarm assignee changes.
- **Severity:** Select this option to update the ticket severity when the alarm severity changes. The following table describes the severity mapping in DX Operational Intelligence and ServiceNow:

Alarm Severity in DX Operational Intelligence	Severity Mapping in ServiceNow
• Critical • 1 • Danger	High
• Major • 2 • Caution	Medium

Alarm Severity in DX Operational Intelligence	Severity Mapping in ServiceNow
• Minor • 3 • Unknown • Default • Warning	Low

Alarm Reopened

An alarm is reopened in DX Operational Intelligence when:

- An DX APM alarm with the same APM alarm ID is ingested into DX Operational Intelligence.
- A Spectrum or UIM alarm with the same host and message is ingested into DX Operational Intelligence.

When an alarm is reopened, you can configure to reopen the associated ticket automatically in the ticketing system. The ticket is reopened only if the old alarm is in the **Cleared** state in DX Operational Intelligence and the associated ticket is in the **Resolved** state in ServiceNow.

Select the **If the alarm is reopened, reopen the associated ticket** option to reopen the ticket. Additionally, you can configure to reopen only those tickets that were resolved in the last 30 days, which is the maximum allowed time. If you do not specify the time period, the duration is taken as 30 days by default.

Allowed Values for this configuration: Days: 0-30, Hours: 0-23, and Minutes: 0-59 respectively.

When the ticket is reopened, the reopened alarm details are updated in the **Work Notes** section based on the message template and the original alarm details remain unchanged in the ticket description.

NOTE

- Tickets can be reopened only for raw alarms.
- If the ticket reopening fails with errors on the ticketing side due to any reason, you can use the following API to ignore the error and create a new incident. Update the **propertyValue** in the response body with the error message. For suppression of multiple failures, add comma-separated values.
  - **API Endpoint:** https://<oi_adminui_url>/oi/v2/tenantmgmtservice/<Tenant_Id>
  - **Method:** POST
  - **Request Body:**
    ```
    {"tenantID":"<Tenant_ID>","PropertyList":
    [{"propertyName":"ITSM_ServiceMgmt_Reopen_Errors","propertyValue":"<Error Message>"}]
    ```
  - **Sample Request Body:**
    ```
    {"tenantID":"4AFB2800-E0CF-4F63-A4A4-CD6AC4A522AD","PropertyList":
    [{"propertyName":"ITSM_ServiceMgmt_Reopen_Errors","propertyValue":"No Record found"}]
    ```

Alarm Cleared

Select the **If the alarm is cleared, change the ticket status to** option to change the status of the ticket to:

- **Closed:** Changes the ticket status to closed if the alarm is cleared in DX Operational Intelligence.
- **Resolved:** Changes the ticket status to resolved if the alarm is cleared in DX Operational Intelligence.

NOTE
When the underlying alarms are cleared in DX Operational Intelligence, the service alarm is cleared and the associated ticket is Closed in ServiceNow.

Ticket Updates in ServiceNow

In the **Update alarms when ServiceNow system changes occur** section, select the relevant options to update the DX Operational Intelligence Alarm data when the ticket information changes in ServiceNow:

- Ticket Updates
- Ticket Status Changes

Ticket Updates

Select the **Notify the ticket owner about the ticket updates** option to notify the ticket owner when the alarm assignee changes. **Assigned to** and **status as updates** is supported in ServiceNow.

Ticket Status Changes

Select the **Clear the alarm, if the ticket status changes to** option to clear the associated alarm when the ticket status changes to **Closed** or **Resolved**.

NOTE

- When tickets are Resolved/Closed for service alarms in ServiceNow, the associated alarm is cleared in DX Operational Intelligence and all the underlying alarms are also cleared.
- When a rootcause alarm is closed in ServiceNow, the associated rootcause alarm is cleared in DX Operational Intelligence and not the service alarm. The underlying alarms get cleared in DX Operational Intelligence only when the rootcause alarm is the cause for the service alarm.

The following table describes the types of alarms that are updated from ServiceNow in DX Operational Intelligence:

Alarm Types / Source Products	Ticket ID	Status	Assignee
Anomaly	Yes	Yes	Yes
Predictions	Yes	Yes	Yes
Service	Yes	Yes	Yes
Custom	Yes	Yes	Yes
Situation	No	No	No
APM	Yes	Yes	Yes
Spectrum	Yes	Yes	Yes
UIM	Yes	Yes	Yes
ADA	Yes	Yes	Yes
Root cause alarm	Yes	Yes	Yes

Set Up ServiceNow

To integrate DX Operational Intelligence with ServiceNow, create a user in ServiceNow and provide the required roles.

Supported ServiceNow Versions

The following ServiceNow versions are supported:

- San Diego
- Rome
- Quebec
- Paris

Create User in ServiceNow

You must create a user in ServiceNow to connect to the ServiceNow instance and also ensure that the following requirements are met:

- Provide the following roles in the **ServiceNow, Roles** list:
  - SOAP-related roles: **soap**, **soap_create**, **soap_delete**, **soap_ecc**, **soap_query**, **soap_query_update**, **soap_script**, and **soap_update**
  - Web service admin role: **web_service_admin**
  - For Get Incident & Request & Change: **odbc** role
  - For Create & Update Incident & Change: **midserver** role
  - For Create & Update Request: **catalog_admin** role
  - For Create a Comment: **u_journal_entry_user** role

    NOTE

    For more information about creating a user in ServiceNow, see the **ServiceNow** documentation.
- The user must also have the privileges for the following tables: **incident**, **sys_db_object**, **sys_user**, **sys_dictionary**, **sys_journal_field**, **task tables** for **read** permissions. These roles are assigned in the ACLs of the tables. Contact the **ServiceNow Administrator** for these roles.

  NOTE

  For information about the non-admin roles that are required to interact with ServiceNow SOAP webservices, see the **ServiceNow** documentation.
- Implement the following basic set of operations per Configuration Item (CI) in your custom or scripted web service:
  - **YES:** Indicates that the specified operation on the specified CI is available as the custom endpoint.
  - **NO:** Indicates the specified operation on the specified CI need not be made available as the custom endpoint.

  The following table lists the set of operations per Configuration Item to be implemented:

ServiceNow tables	insert	update	getRecords	getKeys	get
incident	YES	YES	YES	NO	YES
sys_journal_field	YES	NO	YES	NO	NO
task	NO	NO	YES	NO	NO
sys_dictionary	NO	NO	YES	NO	NO
sys_db_object	NO	NO	YES	NO	NO
task	NO	NO	NO	NO	NO
sys_attachment	NO	NO	YES	NO	NO
ecc_queue	YES	NO	NO	NO	NO
sys_attachment_doc	NO	NO	YES	NO	NO
sys_user	YES	YES	YES	NO	YES
sys_user_grmember	YES	NO	YES	NO	NO
sys_user_group	YES	YES	YES	NO	YES
cmdb_ci	NO	NO	YES	NO	YES

Configure ServiceNow Channel

Tenant Administrators can configure a channel to create tickets in ServiceNow.

Before you configure ServiceNow as a channel, configure the ServiceNow instance with a user role.

As a Tenant Administrator, you can configure ServiceNow Channel to enable bi-directional communication between DX Operational Intelligence and ServiceNow. DX Operational Intelligence creates a ticket in ServiceNow when an alarm is identified. DX Operational Intelligence also synchronizes any updates in alarm in the corresponding ServiceNow ticket.

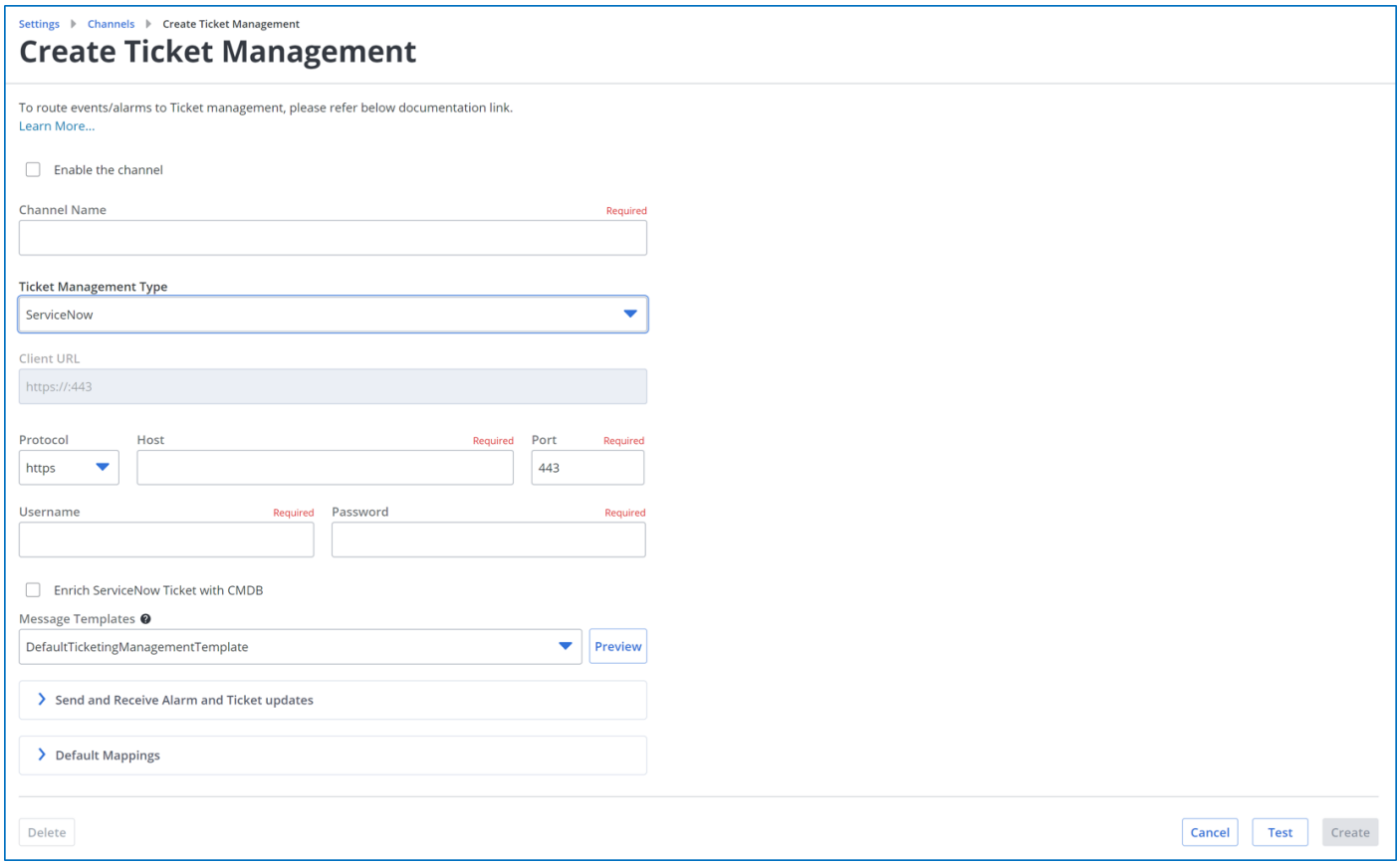
NOTE

- You can add only one ServiceNow Channel in the tenant environment.
- By default, a default policy is associated with the channel.

Complete the following tasks to configure the ServiceNow Channel:

1. Log in to DX Operational Intelligence as a Tenant Administrator and click Settings in the left navigation pane.
   The application displays the Settings page.
2. Click **Connect** in the ITSM Tile.
   The application displays the **Channels** page with the list of existing channels.
3. Click

and select Ticket Management from the Select Channels Type drop-down.
   The application displays the Ticket Management Channel page.



4. Enter a unique channel name in the Channel Name field.
5. Select **ServiceNow** from the Ticket Management Type drop-down.
6. Provide the connection details that DX Operational Intelligence uses to connect with the ticket management system:
   - **Client URL:** Populates a URL using the specified host and port.
   - **Protocol:** http or https protocol for bi-directional data transfer.
   - **Host:** Hostname of the system on which the ticket management system is hosted.
   - **Port:** Port number of the host system.
   - **User Name & Password:** User Credentials with appropriate role and permissions to access the ticket management system.
7. Select **Enrich ServiceNow Ticket with CMDB** and provide the Rest API (CMDB) URL in the CMDB URL field that appears. For more information, see the Enrich ServiceNow Ticket with CMDB section.

   CMDB lookup enriches the ticket with the following information: Location, Config Item, Category, Sub-Category, Urgency, and Assignment Group.
8. Select the template from the **Message Templates** drop-down. This message template is used when the notification is triggered manually or when the channel/template association is not specified at the policy level. You can use the **Preview** option to view the template.

   The application uses the selected template to send messages to ServiceNow when an Alarm is triggered. For more information, see the Message Templates section.
9. Define Send and Receive Rules in the **Send and Receive Alarm and Ticket Updates** section. For more information, see the Automatic Synchronization of Alarm and Ticket Updates section.
10. Select the mapping rules in the **Default Mappings** section. If no mapping rules are created, then the rules are not available for selection. You can map these rules even after you create the channel. For more information, see the Ticket Enrichment Rules section.
11. Click **Test**.

    Validates the channel configuration. The test also validates the intermediate services that are involved in the integration. "ITSM test is successful" message appears upon successful configuration.
12. Click **Save**.

    DX Operational Intelligence completes the configuration and enables the integration with the selected Channel.

Enrich ServiceNow Ticket with CMDB

When you enable the option to enrich the ServiceNow ticket with CMDB, DX OI accesses the ServiceNow CMDB to get the appropriate values and updates the ticket with those values. DX OI uses the REST API to access the CMDB. You must provide the REST API in the **CMDB URL** field. The example format of the API is as follows:

```
https://abc.service-now.com/api/now/table/cmdb_ci?
sysparm_display_value=all&sysparm_exclude_reference_link=true&sysparm_query=nameSTARTSWITH{host}%5Einstall_status
%3D1

Where {host} represents the alarm host in DX OI and is a query parameter in the CMDB REST call.
```

The response that this URL generates includes the enrichment fields. The example response is as follows:
Click to expand...


```
{
    "result": [
        {
            "sys_id": {
                "display_value": "1814376637e39bc0ad8994c543990e41",
                "value": "1814376637e39bc0ad8994c543990e41"
```

```
            },
            "location": {
                "display_value": "AP-India-Bangalore-NetMagic CoLo(IBN)",
                "value": "f5dac0eb373ea200882d83dcb3990ee4"
            },

            "subcategory": {
                "display_value": "VM",
                "value": "VM"
            },
            "category": {
                "display_value": "Compute",
                "value": "Compute"
            },
            "assignment_group": {
                "display_value": "GTSO Eng Unix",
                "value": "b27462f54fecea00a33a3285f110c7c9"
            },
            "u_tier": {
                "display_value": "2",
                "value": "2"
            }
        }
    ]
}
```


If the CMDB-based approach does not work because the response is empty and no related data is available in CMDB, you can use an alternative approach that uses a map file. This approach populates only one field: **assignment group**. To use this approach, create a map file with mappings between the DX OI service name and the corresponding assignment group. Based on this mapping, the appropriate assignment group information is populated in the ServiceNow ticket.The following snippet is an example of the mapping file:
Click to expand...


```
{
   "serviceGroupMapping":[
      {
         "serviceName":"GT Eng Unix",
         "assignmentGroup":"GT Eng Unix"
      },
      {
         "serviceName":"Hosting Backup",
         "assignmentGroup":"Hosting Backup"
      },

      {
         "serviceName":"Eng - Compute",
         "assignmentGroup":"GT Eng Lab"
      }
   ]
}
```


Name the map file that you create using this naming convention: **<16-digit-cohort-id>_snow_mapping.json**

> **NOTE**
> To get the cohort ID for the name of the map file, contact **Broadcom Support**.

After you create the file, you must send this map file to Broadcom Support so that they can upload it to the required location. After, the file is implemented, it may take a maximum of 30 minutes for the information to become available.

NOTE
To get the cohort ID for the name, use the **<es_route>/ao_dxi_tenants_1_1/_search?pretty&q=tenant_name:"?"** route, where **<es_route>** is the Elasticsearch route. In the response, the value for the **tenant_id** is the cohort ID. For example, F9664B9-B66B-4B10-B224-976BF8F4553F.

After you create the file, add the map file to the following location in the **Incident Management pod**: **/Incidentmanagement/incidentmanager/artifact**

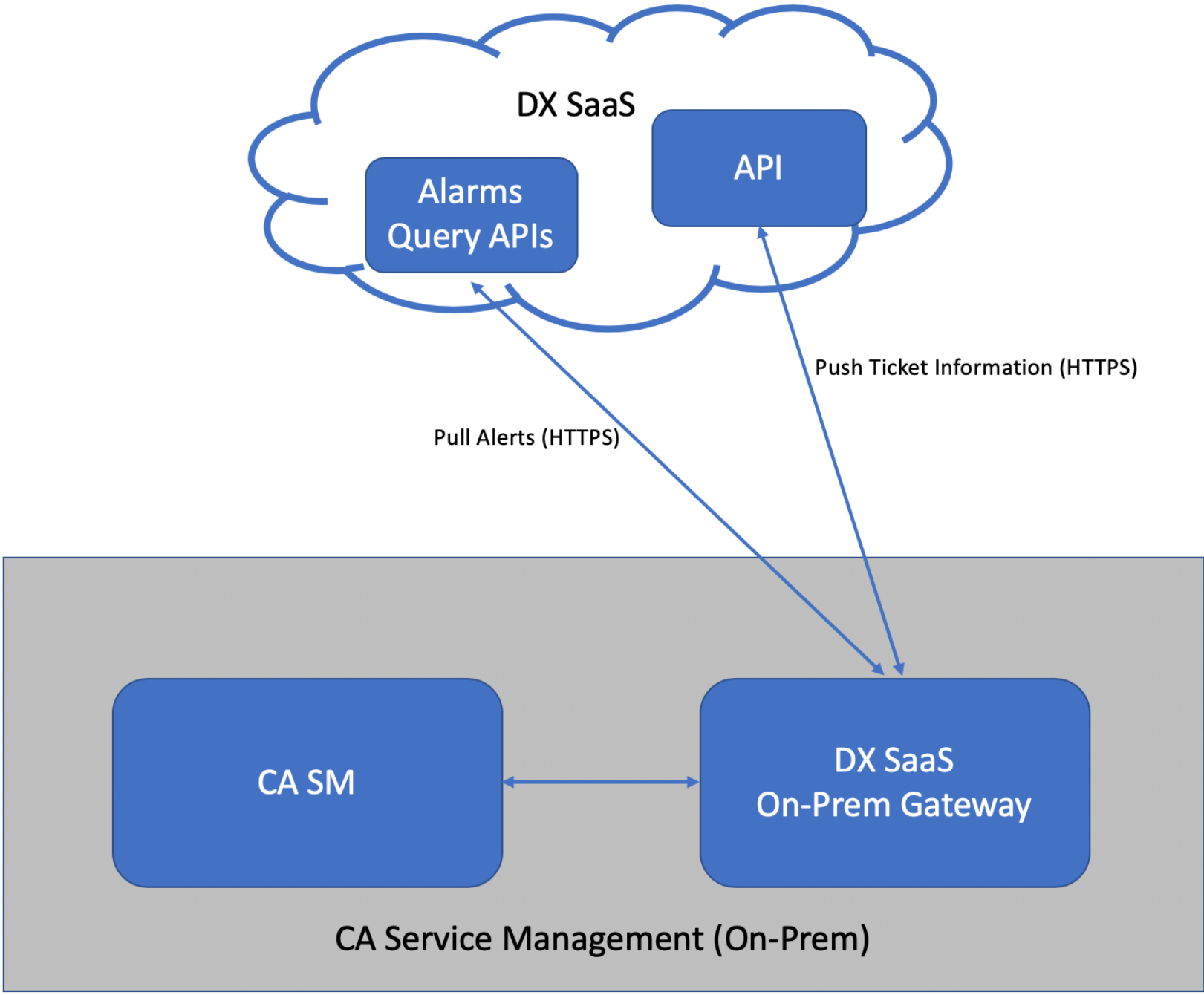
Integration with CA Service Management

You can now integrate with CA Service Management (CA SM) to track and manage alarms raised in DX Operational Intelligence.

When an alarm occurs in DX Operational Intelligence, a ticket is created in the CA Service Management system. Issue updates can be synchronized in both systems.

The CA Service Management Integration supports the following features:

Automatic Ticket Creation	You can configure DX Operational Intelligence to create tickets automatically in CA Service Management for service, root cause alarms, and situations.
Manual Ticket Creation	The Service Management integration provides you with a manual ticket creation option for users who would like to create these tickets manually after assessing the tickets. Once you create a ticket manually, the Service Management integration enables you to make updates to the tickets such as assigning the ticket to any person in your organization.
Annotation Support for Alarms	The Service Management integration solution enables you to add annotations for alarms. In order to add notes for alarms, you need to have an Administrator role. Service Management integration allows you to add annotations to alarms without creating a ticket, and once you have added the annotation, it should be updated in the ticket automatically. The updated annotation gets reflected in both DX Operational Intelligence and CA Service Management.
Bidirectional Updates Between CA Service Management and DX Operational Intelligence	The Service Management integration enables you with synchronization of alarm information across both CA Service Management and DX Operational Intelligence. The synchronized data includes severity, status, assignee, and so on. Therefore, if any changes are made to alarm severity, status, assignee, and so on in DX Operational Intelligence, then the same changes get reflected in CA Service Management. Any changes made to status and assignee in CA Service Management are reflected in DX Operational Intelligence.
Clear Alarms and Ticket Closure	The Service Management integration lets you clear alarms and close tickets manually. Therefore, all the alarms that you clear in DX Operational Intelligence and all the tickets that you close in DX Operational Intelligence get reflected in CA Service Management as well. All issue updates can be synchronized in both systems.



CA Service Management Integration Process

Perform the following steps to integrate DX Operational Intelligence with CA Service Management:

1. Review the **Before You Begin BMC Remedy Integration** section. For more information, click here.
2. Deploy On-Prem ITSM using the DX Gateway installer. For more information, click here.
   a. Download DX Gateway from the **Settings** page.

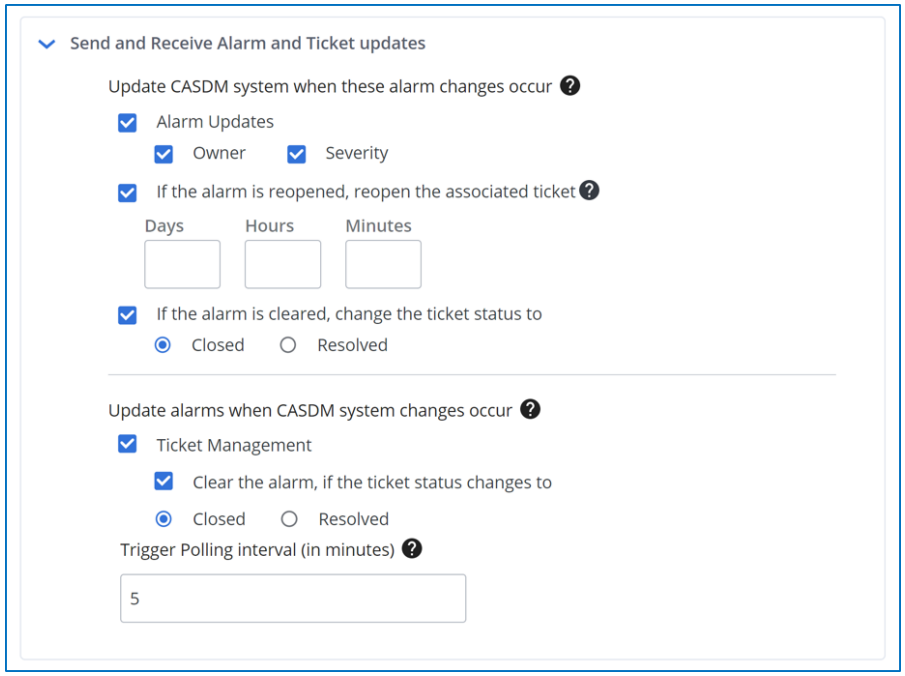
b. Create the CA SDM channel.
c. Configure and deploy On-Prem ITSM using the DX Gateway common installer.

Before You Begin CA Service Management Integration
This section describes the CA Service Management Integration

Before you configure the integration, review this section to understand how you can synchronize the alarm and ticket updates in both systems. During the channel creation, you can configure to update any alarm changes in DX Operational Intelligence to reflect in the associated tickets in CA SDM. Similarly, any changes to the tickets in CA SDM can be updated in the associated alarms in DX Operational Intelligence.

- Alarm Updates in DX Operational Intelligence
  - Alarm Updates
  - Alarm Reopened
  - Alarm Cleared
- Ticket Updates in CA SDM
  - Ticket Updates
  - Ticket Status Changes

You can configure this synchronization in the **Send and Receive Alarm and Ticket Updates** section of the **Channels > Ticket Management** page:



Before you begin the integration with CA SDM, ensure the following requirements are met:

> **NOTE**
> This integration is supported on both Windows and Linux.

- You have access to DX Operational Intelligence.
- DX Operational Intelligence is deployed on a dedicated VM with the following minimum requirements:
  - CPU - 4 Cores
  - 8-GB RAM
  - 100-GB non-OS space

Alarm Updates in DX Operational Intelligence

In the **Update CASDM system when these alarm changes occur** section, select the required options to send the DX Operational Intelligence alarm updates to the associated CA SDM ticket:

- Alarm Updates
- Alarm Reopened
- Alarm Cleared

Alarm Updates

- **Owner:** Select this option to update the ticket owner when the alarm assignee changes.
- **Severity:** Select this option to update the ticket severity when the alarm severity changes. The following table describes the severity mapping in DX Operational Intelligence and CA SDM:

Alarm Severity in DX OI	Severity Mapping in CA SM
• Major • 2 • Caution	Medium
• Critical • 1 • Danger	High
• Minor • 3 • Unknown • Default • Warning	Low

Alarm Reopened

Reopening of the associated ticket when an alarm is reopened is not supported.

Alarm Cleared

Select the **If the alarm is cleared, change the ticket status to** option to change the status of the associated ticket to:

- **Closed:** Changes the ticket status to closed if the alarm is cleared in DX Operational Intelligence.
- **Resolved:** Changes the ticket status to resolved if the alarm is cleared in DX Operational Intelligence.

> **NOTE**
> When the underlying alarms are cleared in DX Operational Intelligence, the service alarm is cleared and the associated ticket is Closed in CA SDM.

Ticket Updates in CA SDM

In the **Update alarms when CASDM system changes occur** section, select the relevant options to update the DX Operational Intelligence alarm information when the ticket information changes in CA SDM:

- Ticket Status Changes

Ticket Status Changes

Select the **Clear the alarm, if the ticket status changes to** option to clear the associated alarm when the ticket status changes to **Closed** or **Resolved**.

NOTE

- When tickets are Resolved or Closed for service alarms in CA SDM, the associated alarm is cleared in DX Operational Intelligence and all the underlying alarms are also cleared.
- When a rootcause alarm is closed in CA SDM, the associated rootcause alarm is cleared in DX Operational Intelligence and not the service alarm. The underlying alarms get cleared in DX Operational Intelligence only when the rootcause alarm is the cause for the service alarm.

The following table describes the types of alarms that are updated from CA SDM in DX Operational Intelligence:

Alarm Types / Source Products	Ticket ID	Status	Assignee
Anomaly	Yes	Yes	Yes
Predictions	Yes	Yes	Yes
Service	Yes	Yes	Yes
Custom	Yes	Yes	Yes
Situation	No	No	No
APM	Yes	Yes	Yes
Spectrum	Yes	Yes	Yes
UIM	Yes	Yes	Yes
ADA	Yes	Yes	Yes
Root cause alarm	Yes	Yes	Yes

Configure CA Service Management as Channel

Tenant Administrators can configure a channel to track and manage alarms in CA Service Management.

As a System Administrator, you can configure CA Service Management as a Channel to enable bi-directional communication between DX Operational Intelligence and CA SM.

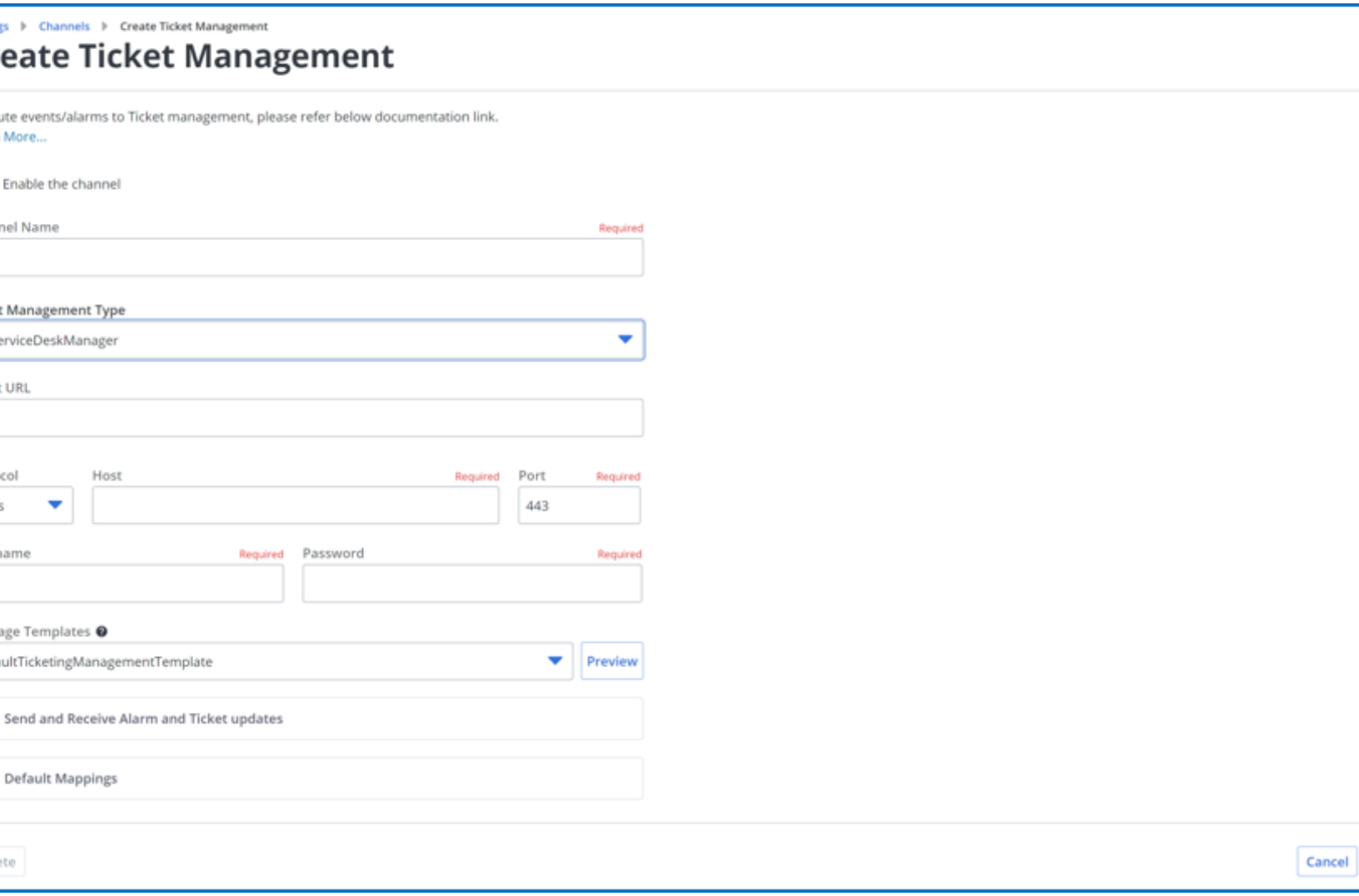
NOTE

Before you create the channel, review **CA Service Management Integration Process** in the Integration with CA Service Management section.

Complete the following tasks to configure the CA Service Management Channel:

1. Log in to DX Operational Intelligence as a Tenant Administrator and click Settings in the left navigation pane.
   The application displays the Settings page.
2. Click **Connect** in the ITSM Tile.
   The application displays the **Channels** page with the list of existing channels.
3. Click

   and select Ticket Management from the Select Channels Type drop-down.
   The application displays the Ticket Management Channel page.



4. Select the **Enable the Channel** checkbox.
   This action enables the connection between the DX Operational Intelligence and the channel.
5. Enter a unique channel name in the Channel Name field.
6. Select CAServiceDeskManager from the Ticket Management Type drop-down.
7. Provide the connection details that DX Operational Intelligence uses to connect with the ticket management system:
   a) **Client URL:** Enter the client URL.
   b) **Protocol:** http or https protocol for bi-directional data transfer.
   c) **Host:** The hostname of the system on which the ticket management system is hosted. Port: Port number of the host system.
   d) **User Name and Password:** User Credentials with the appropriate role and permissions to access the ticket management system.
8. Select the template from the Message Template drop-down. This message template is used when the notification is triggered manually or when the channel/template association is not specified at the policy level. You can use the Preview option to view the template
   For more information, see the Message Templates section.

9. Define Send and Receive Rules in the **Send and Receive Alarm and Ticket Updates** section. For more information, see the Before You Begin CA Service Management Integration section.
10. Select the mapping rules in the **Default Mappings** section. If no mapping rules are created, then the rules are not available for selection. You can map these rules even after you create the channel. For more information, see the Ticket Enrichment Rules section.
11. Click **Create**.

    A success message appears upon successful configuration and enables the integration.

Add Ticket Enrichment Rule

If you want to enrich the generated ticket, you may create a ticket enrichment rule. In the ticket enrichment rule, you can map the following out-of-the-box incident fields with the supported values that are mentioned in the table:

CA SDM Field	Supported Values (Case-sensitive)
delete_flag	true, false
in,category,@REL_ATTR	Applications, Email, Hardware, Networks, Printer, and Software
in,impact,@REL_ATTR	None, Oneperson, SmallGroup, SingleGroup, MultipleGroups, Entireorganization
in,priority,@REL_ATTR	One, HIGHPriority, MEDIUM-HIGHPriority, MEDIUMPriority, MEDIUM-LOWPriority, LOWPriority
in,severity,@REL_ATTR	Escalated, SupervisorEscal, MgrEscal, HDMgrEscalation, AllHandsEscalation
in,status,@REL_ATTR	Acknowledged, AnalysisComplete, ApprovalinProgress, Approved, Avoided, AwaitingEndUserResponse, AwaitingVendor, Cancelled, CloseRequested, Closed, ClosedUnresolved, FixinProgress, Fixed, Hold, InProgress, KnownError, Open, PendingChange, Problem-Closed, Problem-Fixed, Problem-Open, Rejected, Researching, Resolved, SA-Abandon, SA-Resolved
in,urgency,@REL_ATTR	WhenPossible, Soon, Quickly, VeryQuickly, Immediate

For more information, see the Ticket Enrichment Rules section.

Integration with BMC

This section provides the following information:

- Integration with BMC Remedy
- Integration with BMC Helix

Integration with BMC Remedy

You can integrate DX Operational Intelligence with BMC Remedy so alarms and BMC Remedy tickets are synchronized.

Integration of DX Operational Intelligence with BMC Remedy is bi-directional and any changes to alarm or corresponding BMC Remedy ticket can be synchronized in both the systems.

This integration enables you to:

- Create the BMC Remedy tickets for DX Operational Intelligence alarms manually or create tickets automatically using policies.
- Automatically synchronize updates in both the systems.
- Maintain the following alarm information between DX Operational Intelligence and its associated BMC Remedy ticket:

  - Status of alarms
  - Current assignee (troubleshooter) assigned to tickets
  - Severity of alarms. Currently, severity can be mapped from DX Operational Intelligence to BMC Remedy.
  - Annotation update. If you update annotation for an alarm in DX Operational Intelligence, the associated BMC Remedy ticket gets updated.
- Launch BMC Remedy directly from the ticket in DX Operational Intelligence.

BMC Remedy Integration Process

Perform the following steps to integrate BMC Remedy with DX Operational Intelligence:

1. Review the **Before You Begin BMC Remedy Integration** section. For more information, click here.
2. Deploy On-Prem ITSM using the DX Gateway installer. For more information, click here.
   a. Download DX Gateway from the **Settings** page.
   b. Create the BMC Remedy channel.

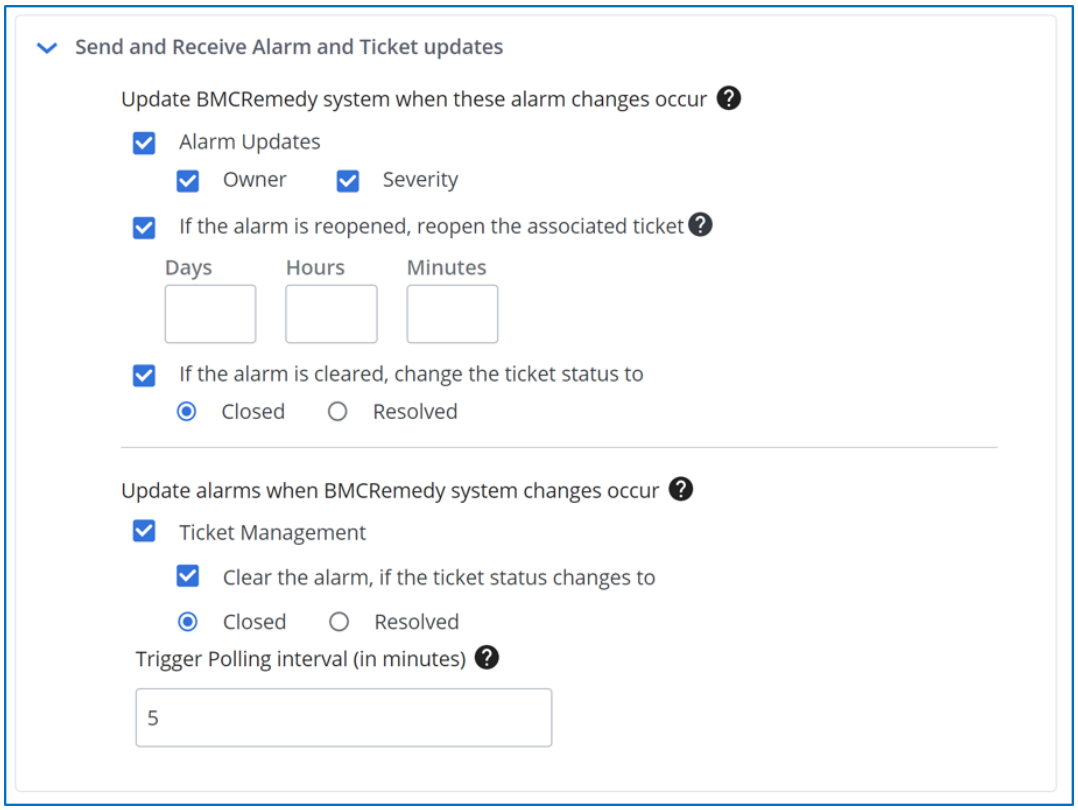
      NOTE
      For DX Operational Intelligence SaaS environments, create the channel with the protocol as http and port as 0.

   c. Configure and deploy On-Prem ITSM using the DX Gateway common installer.
3. Create the BMC Remedy channel. For more information, see the Configure BMC Remedy as Channel section.

Before You Begin BMC Remedy Integration

Before you start the integration, review this section to understand how you can synchronize the alarm and ticket updates in both systems. You can configure to update any alarm updates or changes in DX Operational Intelligence to reflect in the associated tickets in BMC Remedy. Similarly, any updates or changes to the tickets in BMC Remedy can be updated in the associated alarms in DX Operational Intelligence.

You can configure this synchronization in the **Send and Receive Alarm and Ticket Updates** section while creating the channel.



Alarm Updates in DX Operational Intelligence

In the **Update BMCRemedy system when these alarm changes occur** section, select the required options to send the DX Operational Intelligence alarm updates to the associated BMC Remedy ticket:

- Alarm Updates
- Alarm Reopened
- Alarm Cleared

Alarm Updates

- **Owner:** Select this option to update the ticket owner when the alarm assignee changes.
- **Severity:** Select this option to update the ticket severity when the alarm severity changes. The following table describes the severity mapping in DX Operational Intelligence and BMC Remedy:

Alarm Severity in DX Operational Intelligence	Severity Mapping in BMC Remedy
• Critical • 1 • Danger	High
• Major • 2 • Caution	Medium

Alarm Severity in DX Operational Intelligence	Severity Mapping in BMC Remedy
• Minor • 3 • Unknown • Default • Warning	Low

Alarm Reopened

Reopening of the associated ticket when an alarm is reopened is not supported.

Alarm Cleared

Select the **If the alarm is cleared, change the ticket status to** option to change the status of the ticket to:

- **Closed:** Changes the ticket status to closed if the alarm is cleared in DX Operational Intelligence.
- **Resolved:** Changes the ticket status to resolved if the alarm is cleared in DX Operational Intelligence.

Ticket Updates in BMC Remedy

In the **Update alarms when BMCRemedy system changes occur** section, select the relevant options to update the DX Operational Intelligence alarm data when the ticket information changes in BMC Remedy:

- Ticket Status Changes

Ticket Status Changes

Select the **Clear the alarm, if the ticket status changes to** option to clear the associated alarm when the ticket status changes to **Closed** or **Resolved**.

The following table lists the types of alarms that are updated from BMC Remedy in DX Operational Intelligence:

Alarm Types / Source Products	Ticket ID	Status	Assignee
Anomaly	**Yes**	**Yes**	**Yes**
Predictions	**Yes**	**Yes**	**Yes**
Service	**Yes**	**Yes**	**Yes**
Custom	**Yes**	**Yes**	**Yes**
Situation	**Yes**	**Yes**	**Yes**
APM	**Yes**	**Yes**	**Yes**
Spectrum	**Yes**	**Yes**	**Yes**
UIM	**Yes**	**Yes**	**Yes**
ADA	**Yes**	**Yes**	**Yes**
Root cause alarm	**Yes**	**Yes**	**Yes**

Configure BMC Remedy as Channel

Tenant Administrators can configure a channel to synchronize alarm and ticket information in BCM Remedy.

As a Tenant Administrator, you can configure BMC Remedy as a channel to enable bi-directional communication between DX Operational Intelligence and BMC Remedy.

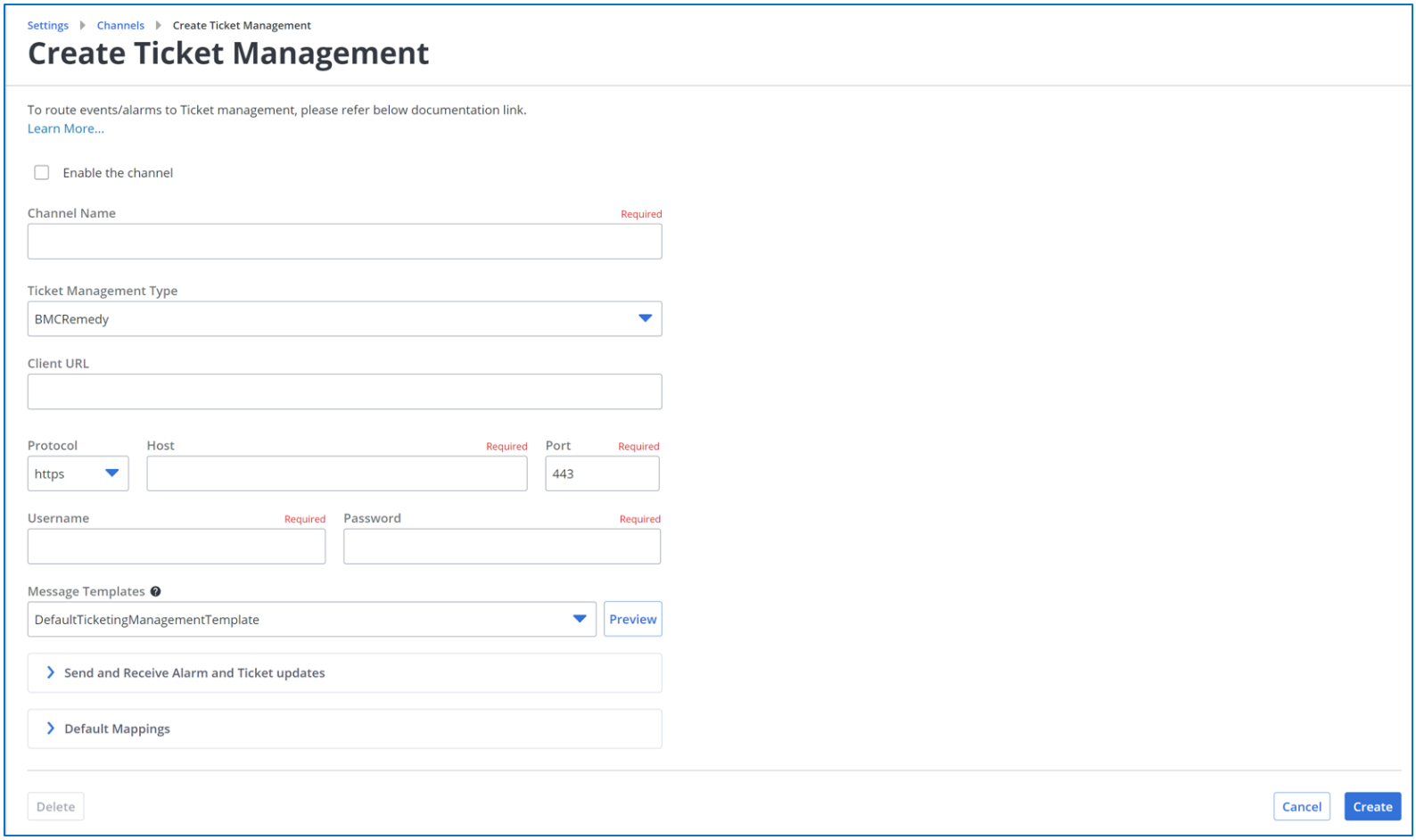
NOTE
Prerequisite:

- DX OI SaaS and BMC Remedy 20.2 integration requires DX Gateway. Download and deploy DX Gateway for this integration. For more information, see the DX Gateway section.
- Review the integration process described in the Integration with BMC Remedy section.
- Enable the Auto Assign rule in BMC Remedy so that the Assigned Group field in the incident is automatically populated. Assigned Group is a required field in BMC Remedy and DX OI does not have any configuration to store the assigned group. Contact your BMC Remedy Administrator to enable this rule. You may perform this step before or after you create the BMC Remedy channel.

Complete the following tasks to configure the BMC Remedy Channel:

1. Log in to DX Operational Intelligence as a Tenant Administrator and click Settings in the left navigation pane.
   The application displays the Settings page.
2. Click **Connect** in the Notification Channel Tile.
   The application displays the **Channels** page with the list of existing channels.
3. Click

   and select Ticket Management from the Select Channels Type drop-down.
   The application displays the Ticket Management Channel page.



4. Select the **Enable the Channel** checkbox.
   This action enables the connection between the DX Operational Intelligence and the channel.

5. Enter a unique channel name in the Channel Name field.
6. Select **BMC Remedy** from the **Ticket Management Type** drop-down.
7. Provide the connection details that DX Operational Intelligence uses to connect with the ticket management system:
   - Protocol: http or https protocol for bi-directional data transfer. **(For DX OI SaaS)** Create the channel with the protocol as HTTP and port as 00.
   - Host: Hostname of the system on which the ticket management system is hosted
   - Port: Port number of the host system
   - User Name and Password: User Credentials with appropriate role and permissions to access the ticket management system.
8. Select the template from the **Message Templates** drop-down. This message template is used when the notification is triggered manually or when the channel or template association is not specified at the policy level. You can use the Preview option to view the template

   For more information see Message Templates.
9. Define Send and Receive Rules in the **Send and Receive Alarm and Ticket Updates** section. For more information, see the Before You Begin BMC Remedy Integration section.
10. Select the mapping rules in the **Default Mappings** section. If no mapping rules are created, then the rules are not available for selection. You can map these rules even after you create the channel. For more information, see the Ticket Enrichment Rules section.
11. Click **Create**.

    DX Operational Intelligence completes the configuration and enables the integration.

Integration with BMC Helix

You can integrate DX Operational Intelligence with BMC Helix so alarms and BMC Helix tickets are synchronized.

Integration of DX Operational Intelligence with BMC Helix enables you to:

- Create BMC Helix tickets for DX Operational Intelligence alarms manually or automatically (using policies).
- Maintain the following alarm information between DX Operational Intelligence and its associated BMC Helix ticket:
  - Status of alarms
  - Current assignee (troubleshooter) assigned to tickets and vice versa
  - Annotation updates to alarms in DX Operational Intelligence, and the associated BMC Helix ticket.
- Associate the channel with the mapping rules to enrich the ticket. For more information, see the Ticket Enrichment Rules section.
- Launch BMC Helix directly from the ticket in DX Operational Intelligence.

BMC Helix Integration Process

Perform the following steps to integrate BMC Helix with DX Operational Intelligence:

1. Review the Before You Begin BMC Helix Integration section.
2. Create the BMC Helix channel.
3. Create Tickets.

Before You Begin BMC Helix Integration

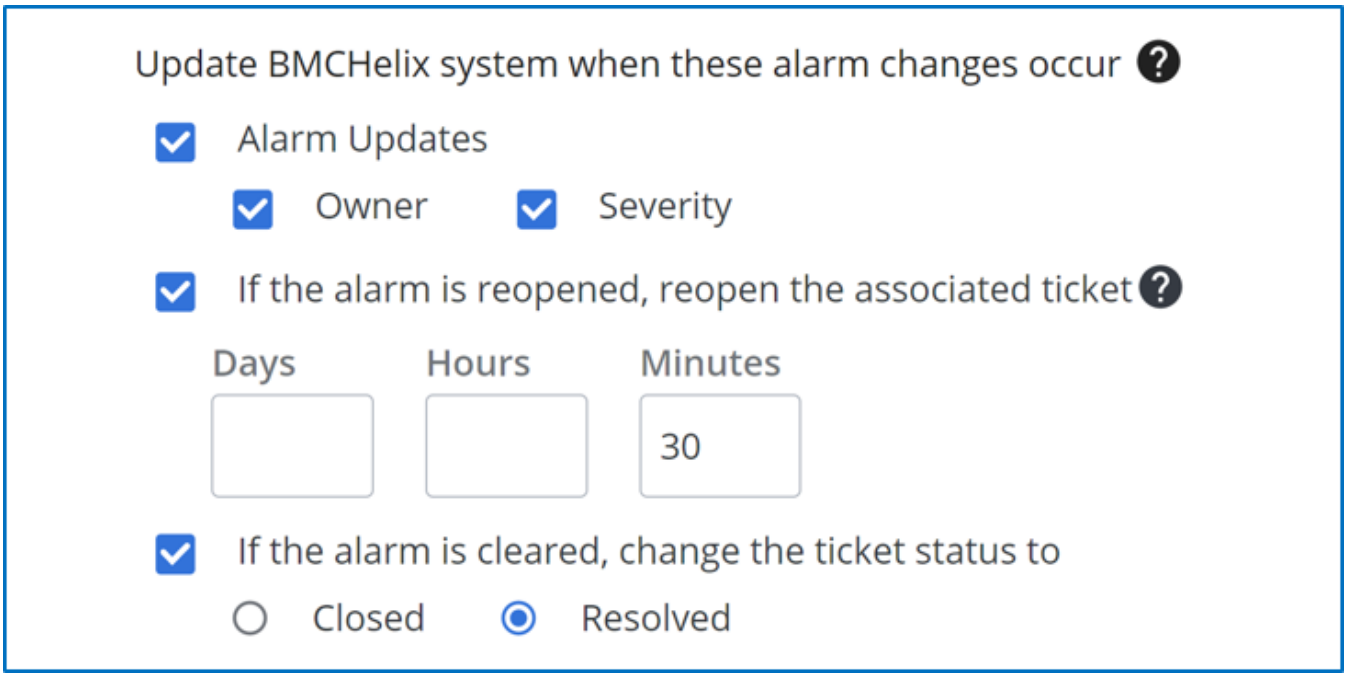
Before you start the integration with BMC Helix, review this section to understand how you can synchronize the alarm and ticket updates in both systems. You can configure to automatically update the alarm updates or changes in DX Operational Intelligence to reflect in the associated tickets in BMC Helix and vice versa.

You can configure this synchronization in the **Send and Receive Alarm and Ticket Updates** section while creating or editing the channel.

- Alarm Updates in DX Operational Intelligence
  - Alarm Updates
  - Alarm Reopened
  - Alarm Cleared
- Ticket Management in BMC Helix
  - Ticket Status Changes
  - Trigger Polling Interval

Alarm Updates in DX Operational Intelligence

In the **Update BMCHelix system when these alarm changes occur** section, select the required options to send the alarm updates in DX Operational Intelligence to the associated BMC Helix ticket:



- Alarm Updates
- Alarm Reopened
- Alarm Cleared

Alarm Updates

Select the alarm updates to be automatically updated in the corresponding tickets:

- **Owner:** Select this option to automatically update the ticket owner in BMC Helix when the alarm owner changes in DX Operational Intelligence. For example, if the ticket owner is changed in DX Operational Intelligence, the owner is automatically updated in the associated ticket in BMC Helix.
- **Severity:** Select this option to automatically update the ticket severity in BMC Helix when the alarm severity changes in DX Operational Intelligence. For example, if the alarm severity is changed to **Critical** from **Major**, the severity in the associated ticket is automatically changed to **High** in BMC Helix.

The following table describes the severity mapping in DX Operational Intelligence and BMC Helix:

Alarm Severity in DX Operational Intelligence	Severity Mapping in BMC Helix
• Critical • 1 • Danger	High
• Major • 2 • Caution	Medium
• Minor • 3 • Unknown • Default • Warning	Low

Alarm Reopened

Select the **If the alarm is reopened, reopen the associated ticket** option to automatically reopen the ticket in BMC Helix. For example, a DX APM alarm named **Alarm1** that is in the **Closed** state exists is DX Operational Intelligence. When another DX APM alarm named **Alarm2** with the same APM alarm ID is ingested into DX Operational Intelligence, this **Alarm2** is considered similar to **Alarm1**. DX Operational Intelligence searches for the ticket for **Alarm1**, reopens that ticket, and assigns it to **Alarm2**.

A ticket is reopened only if the old alarm **Alarm1** is in the **Closed** state in DX Operational Intelligence and the associated ticket is in the **Resolved** state in BMC Helix.

NOTE

- Tickets can be reopened only for the type All Alarms (raw alarms).
- Tickets can be reopened only for alarms from the following source products: DX APM, Spectrum, and UIM.
- A DX APM alarm is considered similar to an older DX APM alarm if the alarm ID is the same as the older alarm. That is, if Alarm2 with the same APM alarm ID as Alarm1 is ingested into DX Operational Intelligence, Alarm2 is considered similar to Alarm1.
- A Spectrum or UIM alarm is considered similar to an older alarm if the host and message are the same as the older alarm. That is, if Alarm2 with the same host and message as Alarm1 is ingested into DX Operational Intelligence Alarm2 is considered similar to Alarm1.

You can also configure to reopen only those tickets that were resolved in the last 30 days, which is the maximum allowed time. If you do not specify the time period, the duration is taken as 30 days by default. **Allowed Values for this configuration:** Days: 0-30, Hours: 0-23, and Minutes: 0-59 respectively.

The alarm details of the reopened ticket are updated in the **Comments** section in BMC Helix based on the message template associated with the BMC Helix channel. However, the original alarm details remain unchanged in the ticket **Description** section.

NOTE

If the ticket reopening fails with errors on the ticketing side due to any reason, you can use the following API to ignore the error and create a new incident. Update the **propertyValue** in the response body with the error message. For suppression of multiple failures, add comma-separated values.

- **API Endpoint:** https://<oi_adminui_url>/oi/v2/tenantmgmtservice/<Tenant_Id>
- **Method:** POST
- **Request Body:**

```
{"tenantID":"<Tenant_ID>","PropertyList":
[{"propertyName":"ITSM_ServiceMgmt_Reopen_Errors","propertyValue":"<Error Message>"}]
```

- **Sample Request Body:**

```
{"tenantID":"4AFB2800-E0CF-4F63-A4A4-CD6AC4A522AD","PropertyList":
[{"propertyName":"ITSM_ServiceMgmt_Reopen_Errors","propertyValue":"No Record found"}]
```

Alarm Cleared

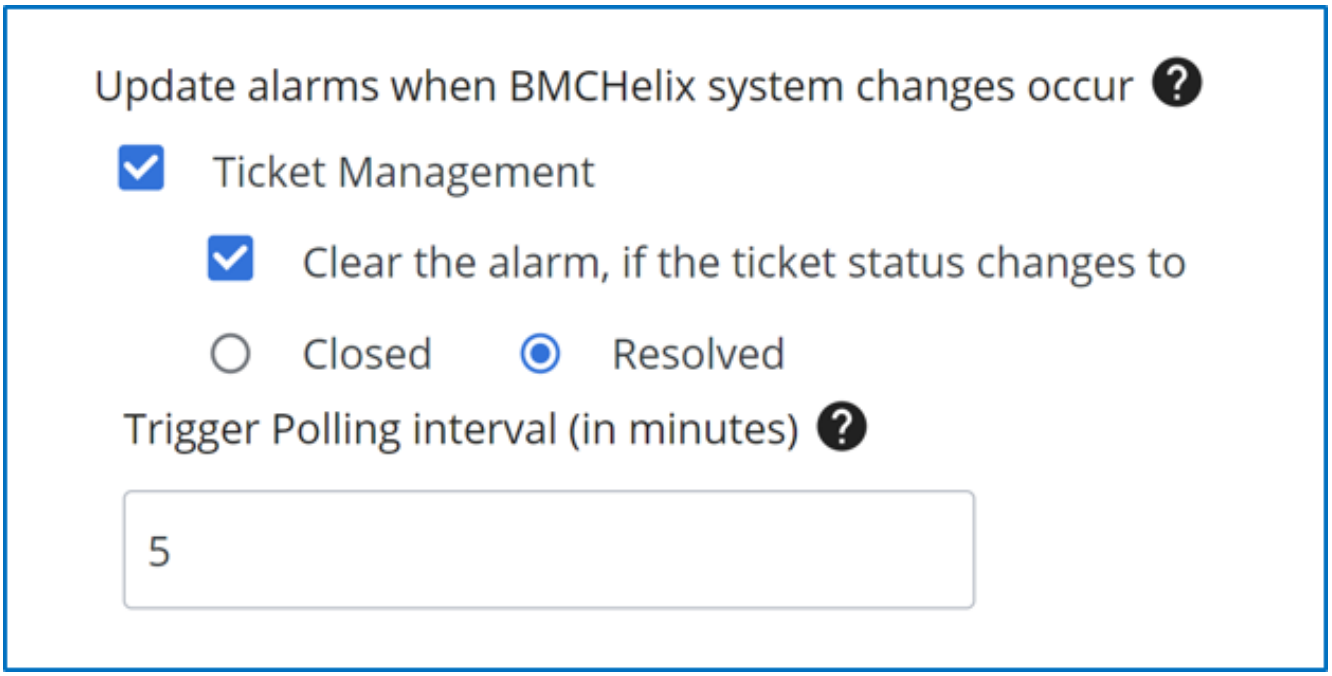
Select the **If the alarm is cleared, change the ticket status to** option to change the status of the ticket to **Resolved** in BMC Helix. For example, if the alarm is cleared in DX Operational Intelligence, the ticket status is automatically changed to **Resolved** in BMC Helix.

NOTE

- **Closed:** Indicates that the incident resolution is verified.
- **Resolved:** Indicates that the incident is resolved.

Ticket Management in BMC Helix

In the **Update alarms when BMCHelix system changes occur** section, select the options to automatically update the alarm data in DX Operational Intelligence when the ticket information changes in BMC Helix.



- Ticket Status Changes
- Trigger Polling Interval

Ticket Status Changes

Select the **Clear the alarm, if the ticket status changes to** option to clear the associated alarm in DX Operational Intelligence when the ticket status changes. For example, when the ticket status changes to **Closed** or **Resolved** in BMC Helix, the alarm is automatically cleared in DX Operational Intelligence.

NOTE
If the alarm is automatically cleared because the ticket status changed to **Closed** or **Resolved**, the **Creator** column in the **Lifecycle Events** tab of the alarm displays as **ITSM**.

Trigger Polling Interval

The polling interval is the frequency of polling the tickets in BMC Helix and passing the updates to DX Operational Intelligence. Default, Minimum, and Recommended value: 5 minutes

Configure BMC Helix as Channel

As a Tenant Administrator, you can configure BMC Helix as a channel to enable bi-directional communication between DX Operational Intelligence and BMC Helix.

Enable Auto Assign Rule

Assigned Group is a required field in BMC Helix to create an incident. You must enable the Auto Assign rule before or after you create the channel in one of the following ways:

- Enable the **Auto Assign** rule in BMC Helix so that the **Assigned Group** field in the incident is automatically populated. Contact your BMC Helix Administrator to enable this rule.
- Create a ticket enrichment or mapping rule for the required alarm type in DX Operational Intelligence.

  NOTE
  You can create this enrichment rule even after you create the channel. But, ensure to associate the rule with the BMC Helix channel.

  Follow these steps:

  a. Click **DX SaaS > Settings** in the left navigation pane.
  b. Click the **Ticket Enrichment Rule** tile.
  c. Click **+ New Rule**.
  d. Enter a name and description for the rule.
  e. Provide the following information in the **Map Incident Fields to Alarm Attributes** section:
     - **Incident Fields:** Click and select **Assigned Group** as the incident field.
     - **Alarm Type:** Select the alarm type (All Alarms, Service Alarms, and Situations) and the alarm attribute to be mapped to the Assigned Group field.
     - **Default:** Enter a valid assigned group. If the alarm attribute is not selected or if the alarm attribute is not available in the alarm payload, then this default value is used as the assigned group.

       NOTE
       For values that you can use in this field, see the Assigned Group field in BMC Helix.
  f. Click **Create**.
     The mapping rule is created. When the ticket is created, the users in the assigned group are assigned to the ticket.

     NOTE
     For more information, see the Ticket Enrichment Rules section.

Create the Channel

Perform the following steps to create BMC Helix as a channel. You can associate the channel with the enrichment or mapping rule while creating the channel or after you create the channel.

Follow these steps:

1. Log in to DX Operational Intelligence as a Tenant Administrator.
2. Click **Settings** in the left navigation pane.
   The application displays the Settings page.

3. Click **Connect** in the **Notification Channel** tile.
   The application displays the Channels page with the list of existing channels.
4. Click

   and select **Ticket Management** from the **Select Channels Type** drop-down.
   The application displays the Ticket Management Channel page.
5. Select the **Enable the Channel** checkbox.
   This action enables the connection between the DX Operational Intelligence and the channel.
6. Provide the following information:
   – **Channel Name:** Enter a unique channel name.
   – **Ticket Management Type:** Select **BMCHelix** from the drop-down.
   – Provide the connection details that DX Operational Intelligence uses to connect with the ticket management system:
     - **Client URL:** Enter the client URL.
     - **Protocol:** Select http or https protocol for bi-directional data transfer. **(For DX OI SaaS)** Create the channel with the protocol as HTTP and port as 443.
     - **Host:** Enter the hostname of the system on which the ticket management system is hosted.
     - **Port:** Enter the port number of the host system.
     - **User Name and Password:** Enter the user credentials for the ticket management system.
7. Select the template from the **Message Templates** drop-down. This template is used when the notification is triggered manually or when the channel or template association is not specified at the policy level. You can use the Preview option to view the template.

   NOTE
   For more information see the Message Templates section.
8. Define the rules to synchronize the alarm and tickets updates in the **Send and Receive Alarm and Ticket Updates** section. For more information, see the Before You Begin BMC Helix Integration section.
9. Select the Assigned Group mapping rule that you had created earlier in the **Default Mappings** section. If the mapping rule is not available, you can associate the rule after you create the channel. When a ticket is created manually, the mapping rules associated with the channel are applied for all the tickets.

   NOTE
   For more information, see the Ticket Enrichment Rules section.
10. Click **Create**.
    DX Operational Intelligence completes the configuration and enables the integration.

Create Tickets

After you have configured the BMC Helix channel, you can submit a ticket manually or you can create tickets automatically using policies.

NOTE
Consider the following points about ticket creation:

- Ensure that users have the following privileges in BMC Helix:

  - Access to Create, Update, Close, and Resolve Incidents
  - Access to Add Comments
- Ensure that the ticket enrichment rule is created and is also associated with the BMC Helix channel.
- For manual ticket creation, the default mapping rule that is associated with the channel is applied. However, for automatic ticket creation, the mapping rule that is associated with the policy is applied.
- BMC Helix tickets are created for all the alarm types.
- A Ticket ID is associated with an alarm on the successful creation of a ticket.
- You can view the alarm attributes for an alarm ticket in the **Description** field in BMC Helix. You can view any updates or delta information on alarms in the **Comments** section in BMC Helix.
- If a policy is associated with the alarm, a ticket gets created for that alarm when the policy criteria are met. Whenever there is an update for the underlying alarm, the alarm and its associated ticket get updated.

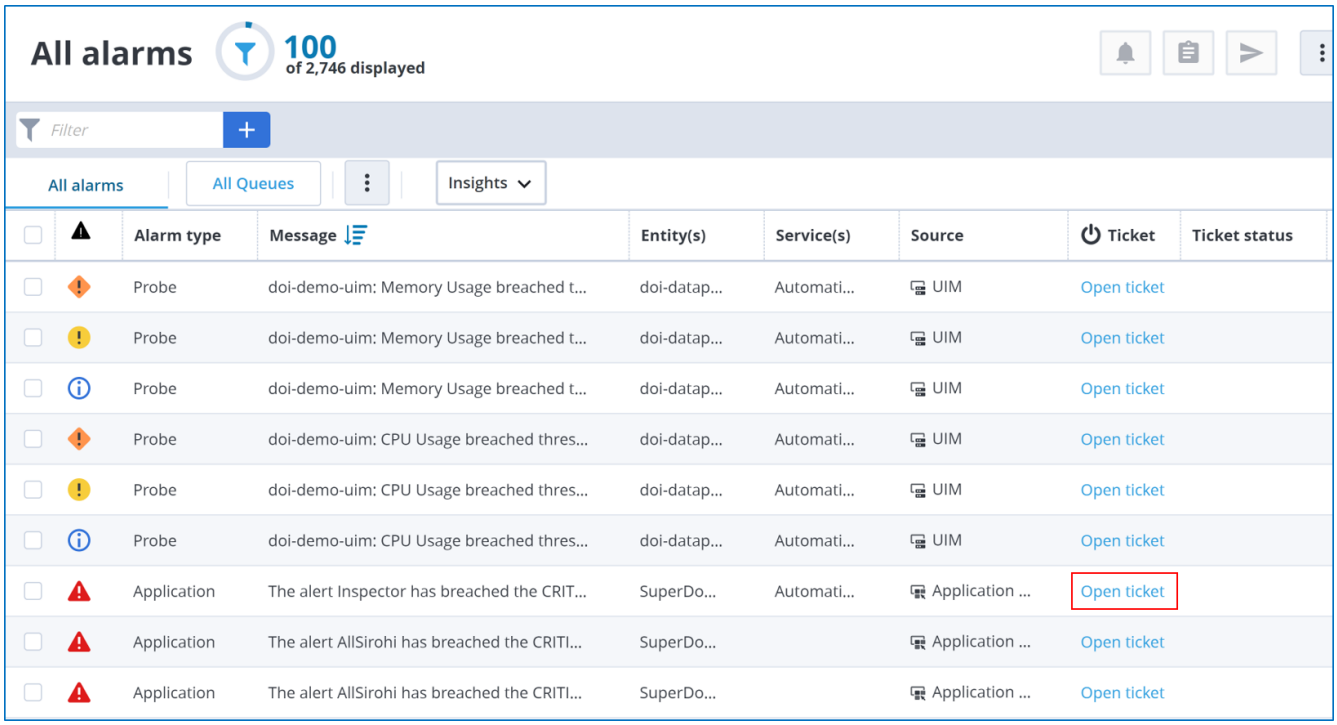
Create Tickets Manually

You can create a ticket for a single alarm or multiple alarms at the same time in DX Operational Intelligence. For a single alarm, the ticket is created using the default mapping rule that is associated with the BMC Helix channel. However, for multiple alarms, you can create the ticket with the default mapping rule that is associated with the channel or you can select the enrichment rule specifically.

Ensure to select the enrichment rule that was created for the Assigned Group to be able to create the ticket.

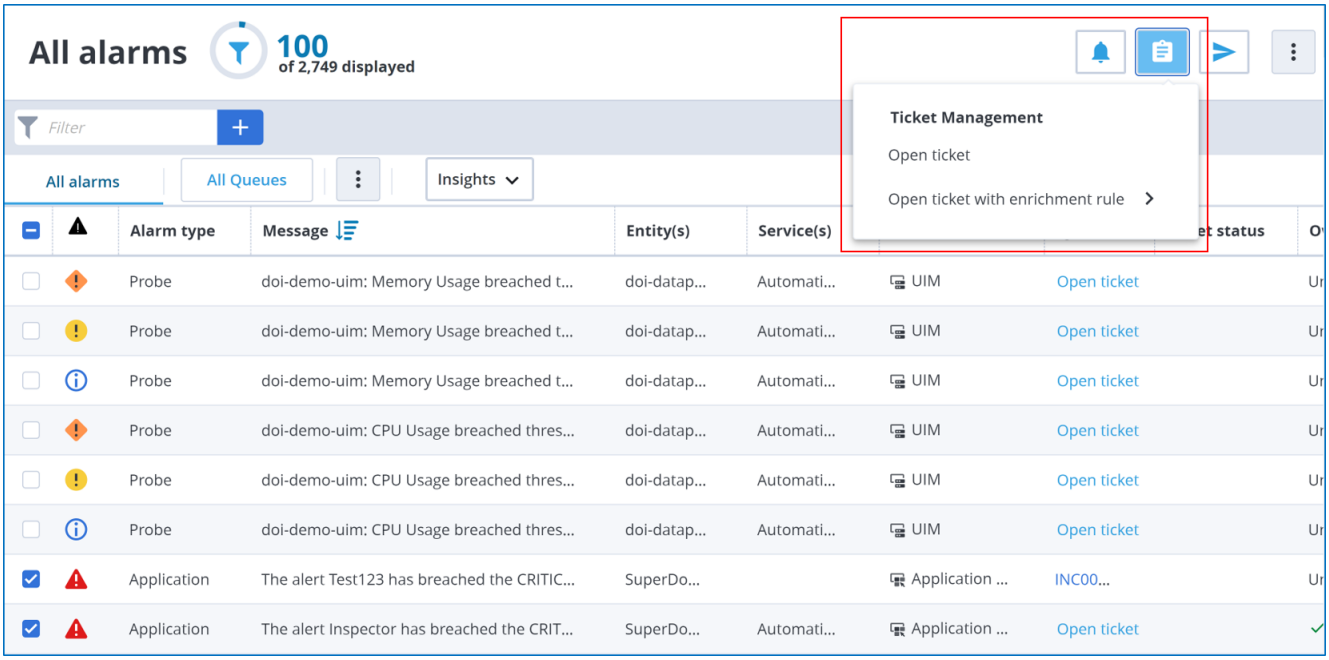
Follow these steps:

1. Log in to DX Operational Intelligence.
2. Open the **Alarms Analytics** page.
3. Open the ticket.
   - **(For Single Alarm)**
     - Click **Open ticket** for the alarm. A ticket is created in BMC Helix and the Ticket ID is displayed in DX Operational Intelligence. The following image illustrates the open ticket option on the **All Alarms** page.



– (**For Multiple Alarms**)

a. Select the required alarms, and click the **Ticket Management** icon that is displayed in the top-right corner.



b. Open the ticket using one of the following options:

- **Click Open ticket.** Use this option to open the ticket with the default mapping rule that is associated with the channel.
- **Click Open ticket with enrichment rule:** Use this option to open the ticket with the enrichment rule that you select here.

The ticket ID is generated.

4. Click the ticket ID to navigate to the BMC Helix instance.
   You can view the alarm attributes for the ticket in the **Description** field in BMC Helix. You can view any updates or delta information on alarms in the **Comments** section.

Create Tickets Automatically Using Policies

To create a ticket automatically in BMC Helix, create a policy and associate the policy with the mapping rule in DX Operational Intelligence. You must associate the policy with the enrichment rule that was created for the Assigned Group to be able to create the ticket.

When the policy criteria are met, a ticket for the alarm is automatically created in BMC Helix.

> **NOTE**
> For more information about how to configure policies, see the Configure Policies section.

Integration with WolkenSoft

You can integrate DX Operational Intelligence with WolkenSoft so alarms and WolkenSoft tickets are synchronized.

Integration of DX Operational Intelligence with WolkenSoft is bi-directional and any changes to alarms or corresponding WolkenSoft tickets can be synchronized in both the systems.

This integration enables you to:

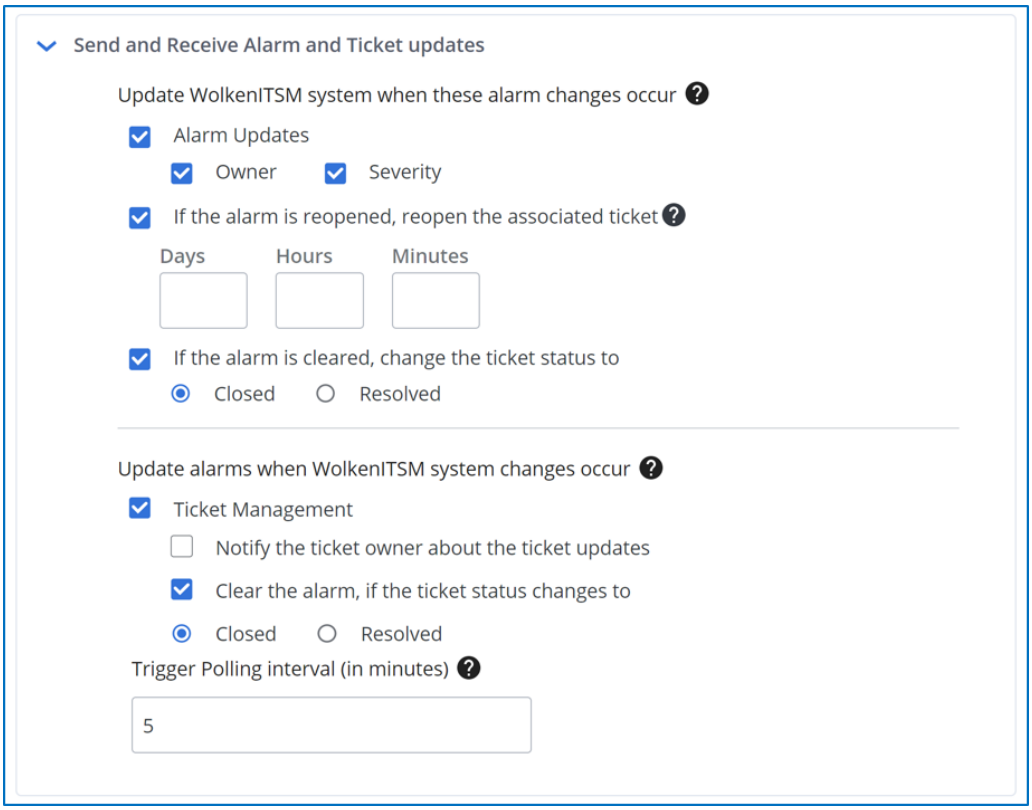
- Create the WolkenSoft tickets for the DX Operational Intelligence alarms manually or create tickets automatically using policies.
- Synchronize updates automatically in both the systems.
- Maintain the following alarm information between DX Operational Intelligence and its associated WolkenSoft ticket:
  - Status of alarms
  - Current assignee (troubleshooter) assigned to tickets
  - Severity of alarms. Currently, severity can be mapped from DX Operational Intelligence to WolkenSoft.
  - Annotation update. If you update the annotation for an alarm in DX Operational Intelligence, the associated WolkenSoft ticket gets updated.
- Enrich the WolkenSoft tickets with the optional WolkenSoft CMDB lookup.
- Launch WolkenSoft directly from the ticket in DX Operational Intelligence.

Before You Begin WolkenSoft Integration

Before you start the integration, review this section to understand how you can synchronize the alarm and ticket updates in both systems. You can configure to update any alarm updates or changes in DX Operational Intelligence to reflect in the associated tickets in Wokensoft. Similarly, you can configure to update any ticket updates or changes in WolkenSoft to reflect in the associated alarms in DX Operational Intelligence.

- Alarm Updates in DX Operational Intelligence
  - Alarm Updates
  - Alarm Reopened
  - Alarm Cleared
- Ticket Updates in WolkenSoft
  - Ticket Updates
  - Ticket Status Changes

You can configure this synchronization in the **Send and Receive Alarm and Ticket Updates** section while creating the channel.



Alarm Updates in DX Operational Intelligence

In the **Update WolkenITSM system when these alarm changes occur** section, select the required options to send the DX Operational Intelligence alarm updates to the associated WolkenSoft ticket:

- Alarm Updates
- Alarm Reopened
- Alarm Cleared

Alarm Updates

- **Owner:** Select this option to update the ticket owner when the alarm assignee changes.
- **Severity:** Select this option to update the ticket severity when the alarm severity changes.

Alarm Reopened

Reopening of the associated ticket when an alarm is reopened is not supported.

Alarm Cleared

Select the **If the alarm is cleared, change the ticket status to** option to change the status of the ticket to:

- **Closed:** Changes the ticket status to closed if the alarm is cleared in DX Operational Intelligence.
- **Resolved:** Changes the ticket status to resolved if the alarm is cleared in DX Operational Intelligence.

> **NOTE**
> When the underlying alarms are cleared in DX Operational Intelligence, the service alarm is cleared and the associated ticket is Closed in WolkenSoft.

Ticket Updates in WolkenSoft

In the **Update alarms when WolkenITSM system changes occur** section, select the relevant options to update the DX Operational Intelligence Alarm data when the ticket information changes in WolkenSoft:

- Ticket Updates
- Ticket Status Changes

Ticket Updates

Select the **Notify the ticket owner about the ticket updates** option to notify the ticket owner when the alarm assignee changes.

Ticket Status Changes

Select the **Clear the alarm, if the ticket status changes to** option to clear the associated alarm when the ticket status changes to **Closed** or **Resolved**.

NOTE

- When tickets are Resolved/Closed for service alarms in WolkenSoft, the associated alarm is cleared in DX Operational Intelligence and all the underlying alarms are also cleared.
- When a rootcause alarm is closed in WolkenSoft, the associated rootcause alarm is cleared in DX Operational Intelligence and not the service alarm. The underlying alarms get cleared in DX Operational Intelligence only when the rootcause alarm is the cause for the service alarm.

The following table lists the types of alarms that are updated in DX Operational Intelligence:

Alarm Types / Source Products	Ticket ID	Status	Assignee
Anomaly	**Yes**	**Yes**	**Yes**
Predictions	**Yes**	**Yes**	**Yes**
Service	**Yes**	**Yes**	**Yes**
Custom	**Yes**	**Yes**	**Yes**
Situation	**No**	**No**	**No**
APM	**Yes**	**Yes**	**Yes**
Spectrum	**Yes**	**Yes**	**Yes**
UIM	**Yes**	**Yes**	**Yes**
ADA	**Yes**	**Yes**	**Yes**
Root cause alarm	**Yes**	**Yes**	**Yes**

Set Up WolkenSoft

For the integration, ensure that the following requirements are met:

- WolkenSoft is installed.
- Username and password to access WolkenSoft are available.
- If using OAuth as the authentication type, then the following information is available:
  - OAuth Client ID for WolkenSoft
  - OAuth Client Secret for WolkenSoft
  - Domain Name for WolkenSoft
  - Service Account for WolkenSoft

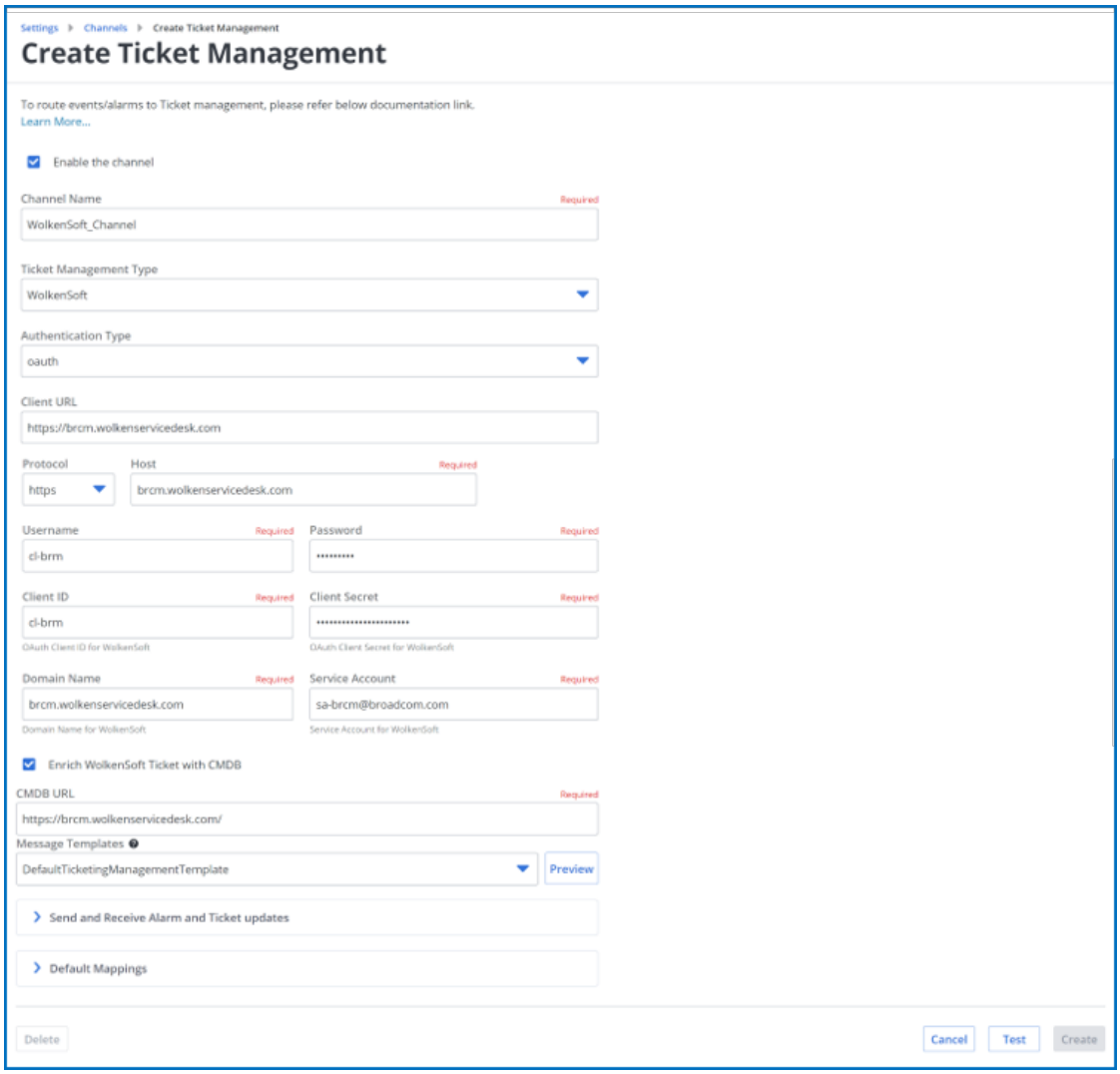
Configure WolkenSoft as Channel

As a Tenant Administrator, you can configure WolkenSoft as a Channel to enable bi-directional communication between DX Operational Intelligence and WolkenSoft.

Complete the following tasks to configure the WolkenSoft Channel:

1. Log in to DX Operational Intelligence as a Tenant Administrator and click Settings in the left navigation pane.
   The application displays the Settings page.
2. Click **Connect** in the Notification Channel Tile.
   The application displays the **Channels** page with the list of existing channels.
3. Click

   and select Ticket Management from the Select Channels Type drop-down.
   The application displays the Ticket Management Channel page.



4. Select the **Enable the Channel** checkbox.
   This action enables the connection between the DX Operational Intelligence and the channel.

5. Enter a unique channel name in the Channel Name field.
6. Select **WolkenSoft** from the **Ticket Management** Type drop-down.
7. Select the Authentication Type.
8. Provide the following information:
   - **Client URL:** Enter the client URL.
   - **Protocol:** Select the protocol for the bi-directional data transfer.
   - **Host:** Enter the hostname of the system on which the ticket management system is hosted.
   - **Username:** Enter the username for WolkenSoft.
   - **Password:** Enter the password.
   - **Client ID:** Enter the OAuth client ID for WolkenSoft.
   - **Client Secret:** Enter the OAuth client secret for WolkenSoft.
   - **Domain Name:** Enter the domain name for WolkenSoft.
   - **Service Account:** Enter the service account for WolkenSoft.
9. Select the **Enrich WolkenSoft Ticket with CMDB** checkbox. When you enable this option, DX OI accesses the WolkenSoft CMDB to get the appropriate values and updates the ticket with those values.

   a) Enter the CMDB URL.

   CMDB lookup enriches the ticket with the following information: Location, Config Item, Category, Sub-Category, Urgency, and Assignment Group.
10. Select the template from the **Message Templates** drop-down. This message template is used when the notification is triggered manually or when the channel/template association is not specified at the policy level. You can use the Preview option to view the template

    For more information see the Message Templates section.
11. Define Send and Receive Rules in the **Send and Receive Alarm and Ticket Updates** section. For more information, see the Before You Begin WolkenSoft Integration section.
12. Select the mapping rules in the **Default Mappings** section. If no mapping rules are created, then the rules are not available for selection. You can map these rules even after you create the channel. For more information, see the Ticket Enrichment Rules section.
13. Click **Test**.

    Validates the channel configuration. The test also validates the intermediate services that are involved in the integration. The "**ITSM test is successful**" message appears upon successful configuration.

    NOTE
    If the channel validation fails, you would be prompted to enter the correct information. When you click **Test** again after reentering the correct values, the following message may be displayed:

    ```
    Please fill the required fields with the valid data.
    ```

    Workaround: Perform one of the following steps:

    - Refresh the page and enter the information again.
    - Recreate the channel and ensure that you enter the correct values.
14. Click **Save**.

    DX Operational Intelligence completes the configuration and enables the integration with the selected Channel.

Troubleshoot Notifications

This video walks you through the troubleshooting steps for some of the issues:

Policies Overview

{"URL":["https:/cloudmanagement/#/settings/policies"],"description":"concept.dita_04588f01-ee42-4b5c-9cb6-56501a84c91d","new":"","new_video":"","admin":"","troubleshooting":{"masterkb":"","text":"","URL":[]},"pendo":"","video":[]}

A policy determines when notifications are sent for alarms. A notification is sent to the associated channels when an alarm meets the filter criteria in a policy. A policy includes one or more filters that define the conditions that trigger notifications.

DX Operational Intelligence provides a read-only default policy that you can link with the notification channels. This default policy has Situations as the alarm type with the following filter criteria:

- **Alarm Type:** SituationCluster, **Is Orphan:** false, **Situation Type:** Custom, Spectrum, **Severity:** Critical, Major, and **Status:** Closed (Not equals)

Supported Alarm Types

DX Operational Intelligence supports the generation of notifications for the following alarm types:

- **Service Alarm**: A service alarm is a group of alarms that affect one or more business services and are related to an incident. The service alarm is identified based on the time the alarm occurred and the root cause.

  NOTE
  Service Alarm is not available for new tenants.
- **Rootcause Alarm**: A rootcause alarm is the alarm on the topologically deepest device in the affected business service. All situations that are reported by alarms in the group are due to the identified root cause.

  NOTE
  Rootcause Alarm is not available for new tenants.
- **All Alarm**: All alarms are **raw alarms** that are generated from the source products such as DX Infrastructure Management, Spectrum, ADA, and DX APM or any custom data source.
- **Situation:** A Situation alarm is a collection of alarms representing an incident impacting applications or data center health. Situations are created using machine learning-based clustering algorithms employing time correlation, topological relationships, and natural language processing for analysis.

Policy Filters

Each alarm type has a default policy filter. Policy filters are conditions or attribute:value pairs that determine when notifications are sent. If the value in the filter matches the attribute value in the alarm, the filter condition is met. If all the filter conditions are met, then DX Operational Intelligence triggers the notification. For example, if a policy contains the filter **acknowledged:false**, and an alarm has not been acknowledged, DX Operational Intelligence sends a notification or updates the ITSM system.

When you select an alarm type on the Policy page, the default filter for that alarm type is displayed. You can use that filter or you can add more filters to the policy.

NOTE
The default filter cannot be cleared.

Default Policy Filters for Existing Tenants

The following table lists the default filter for each of the alarm types:

Alarm Type	Default Filter
Service Alarm, Rootcause Alarm	**Alarm Type**: Service
All Alarm	**Alarm Type**: Service (Not equals)

Alarm Type	Default Filter
Situation (Default Policy)	**Alarm Type:** SituationCluster, **Is Orphan:** false, and **Age (In min):** 10 (Greater than) You can delete the **Age (In min)** filter attribute only if: • **Severity: Critical** filter attribute is added • **Is Stable** filter attribute is set to **true**

Default Policy Filters for New Tenants

The following table lists the default filter for each of the alarm types:

NOTE
The Service Alarm and Rootcause Alarm alarm types are not available for new tenants.

Alarm Type	Default Filter
All Alarm	**Alarm Type**: Service (Not equals)
Situation (Default Policy)	**Alarm Type:** SituationCluster, **Is Orphan:** false, and **Age (In min):** 10 (Greater than) You can delete the **Age (In min)** filter attribute only if: • **Severity: Critical** filter attribute is added • **Is Stable** filter attribute is set to **true**

Supported Filter Operators

Operators define the type of filter match to use. Use the filter to view only those alarms with attributes matching your search criteria.

NOTE

- While filtering, the **AND** operator is used between attributes and the **OR** operator is used between the attribute values. For example,
  ```
  (Severity: Critical OR Major) AND (Alarm Type:"application" OR "fault")  AND (Message contains
   "sshd" OR Message does not contain "ALARM: [SYSTEMS]")
  ```
- Only asterisk (*) and dot (.) are supported in the policy filters.

The following table lists the supported operations:

Operator	Description
Equals	Returns results that match the specified value. Example: Alarm ID equals*120.dxi-na1.saas.example.com* Returns alarms that have the Alarm ID as 120.dxi-na1.saas.example.com.
Not equals	Returns results that do not match the value. Example: Alarm ID not equals *120.dxi-na1.saas.example.com* Returns alarms that do not have the Alarm Id as120.dxi-na1.saas.example.com
Contains	Returns results that contain the specified value. Example: Ticket ID contains *1* and *3*. Returns Ticket IDs which contain 1 and 3.
Does not contain	Returns results that do not contain the specified value. Example: Ticket ID does not contain *1*, and *3*. Returns Ticket IDs that do not contain 1 and 3.

Operator	Description
Starts with	Returns results that start with the specified value. Example: Root Cause Source starts with *UIM*. Returns Root Cause Source which starts with UIM.
Does not start with	Returns results that do not start with the specified value. Example: Root Cause Source does not start with *EMEA*. Returns Root Cause Source that does not start with EMEA.
Ends with	Returns results that end with the specified value. Example: The owner ends with *example.com*. Returns Owner whose value ends with example.com.
Does not end with	Returns results that do not end with the specified value. Example: The owner does not end with *example.com*. Returns Owner whose value does not end with example.com.
Greater than	Returns results that are greater than the specified value. Example: The Alarm Count is greater than 10. Returns Alarms where the count is greater than 10.

Create Policy

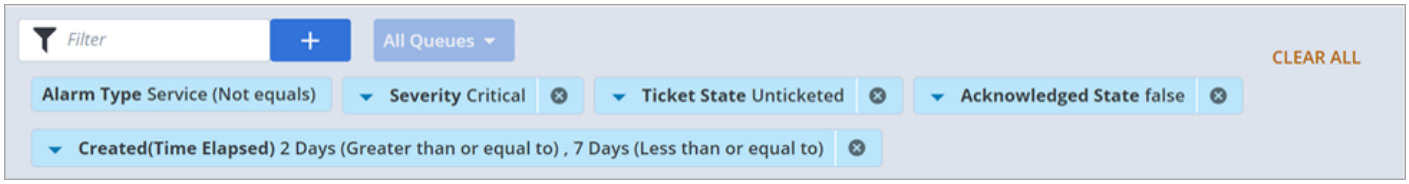
Policies determine when notifications are sent for alarms. Tenant Administrators can create policies using the Policies tile in DX Operational Intelligence.

{"URL":["https://cloudmanagement/#/settings/policies/processingpolicies"],"description":"task.dita_e58e8ad7-1b67-4c3c-bf98-afbb3942c8d3","new":"","new_video":"","admin":"","troubleshooting":{"masterkb":"","text":"","URL":[]},"pendo":"","video":[]}

The following are the guidelines for policies and notification channels that you must know before you create a policy:

- A notification channel can exist without a policy.
- A notification channel can be associated with multiple policies.
- A policy can be created before or after the notification channels are created.
- A policy can also be associated with multiple notification channels.
- Age-based policies are not supported by Automic.
- If multiple policies are satisfying the same alarm, then the notification is sent for all the policies.
- On the policy creation page, by default, the Service Alarm alarm type is selected for the existing tenants and the Situations alarm type is selected for the new tenants.
- The default policy triggers notifications for all alarms with major or critical severity. You cannot edit the default policy.
- **Trigger Alarm Notification:** For All Alarms - Updated alarms and All Alarms - Closed alarms, you can configure to send a notification only when the severity increases or when the severity changes using the additional rule attributes. For example, you can configure to send the notification only when the severity level increases from Minor to Major.
- If any alarm updates retrigger the notification when the policy criteria are met, then the notifications are sent through all the configured channels for the same alarm.
- By default, the Maintenance Mode State is always set to false to avoid alarms notifications from the devices during the maintenance. You can however configure this state to true during the creation or updation of the policies.
- If a user is deleted, the existing policy must be updated with another user.
- To get notified about alarms that were missed or not acted upon by the IT Operations teams for more than a certain period of time, you can use the new filter attribute **Create (Time Elapsed)** to create policies based on the created time of the alarms. You can also use this filter to create policies to automate the workflow of identifying any such old alarms existing in the system.

  For example, you can give a time range as shown:



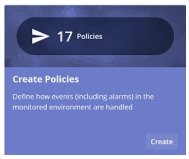
Or, you can give a relative time period:

NOTE

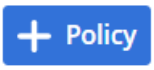
- The highest supported value is 365 days.
- 2 Days (Greater than or equal to) signifies: Current time – 2 days
- 7 Days (Less than or equal to) signifies: Current time – 7 days
- Supported time units: Days, Hours, and Minutes
- Creating age-based policies is not supported for Automic Integration.

Follow these steps:

1. Log in to DX Operational Intelligence as a Tenant Administrator.
2. Click Settings in the left navigation menu.
3. Click **Create** in the Policies tile.

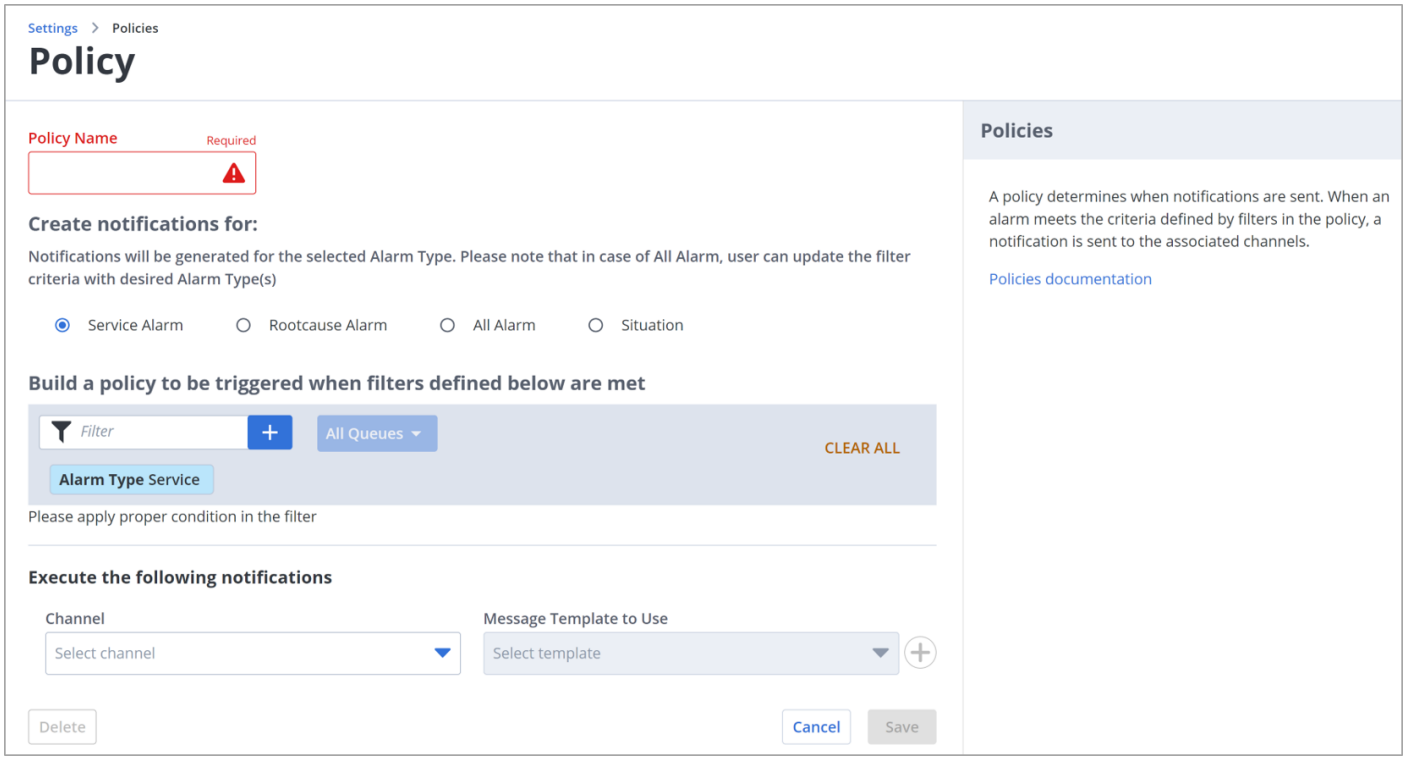


The application displays the Policies page with the list of existing policies if any.

4. Click 

.

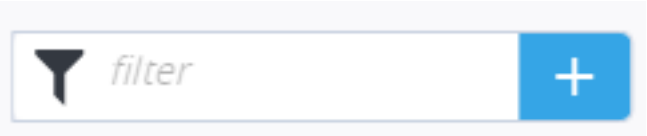
The application displays the Create Policy page. For existing tenants, the **Service Alarm** alarm type is selected. For new tenants, the **Situations** alarm type is selected. The following image illustrates the policy creation page for an existing tenant.



5. Enter a unique name for the policy in the Policy Name field.
6. Select the **Alarm Type** in the **Create notifications for** section:
   - Service Alarm (Only for existing tenants)
   - Rootcause Alarm (Only for existing tenants)
   - All Alarm
   - Situation

     NOTE

     - You can create a policy for Service Alarms and Rootcause Alarms only for existing tenants.
     - Do not create a **Service Alarm** policy and **Rootcause Alarm** policy for the same notification channel.
7. Define the filter criteria to determine when the application must trigger a notification.
   a) Click

   to add a filter.

   The attributes are grouped under **Commonly Used**, **CI Attributes**, and **All Attributes**. For the CI Attributes to be displayed, you must register the CI attributes. For more information, see the Add or Update the CI Attributes section.

   The application displays the filter attributes based on the selected Alarm Type.

b) Select the filter attributes from the displayed list, the relevant value, and the operator. The application displays the filter operators depending on the filter attribute that you have selected. For more information, see the Supported Filter Operators section.

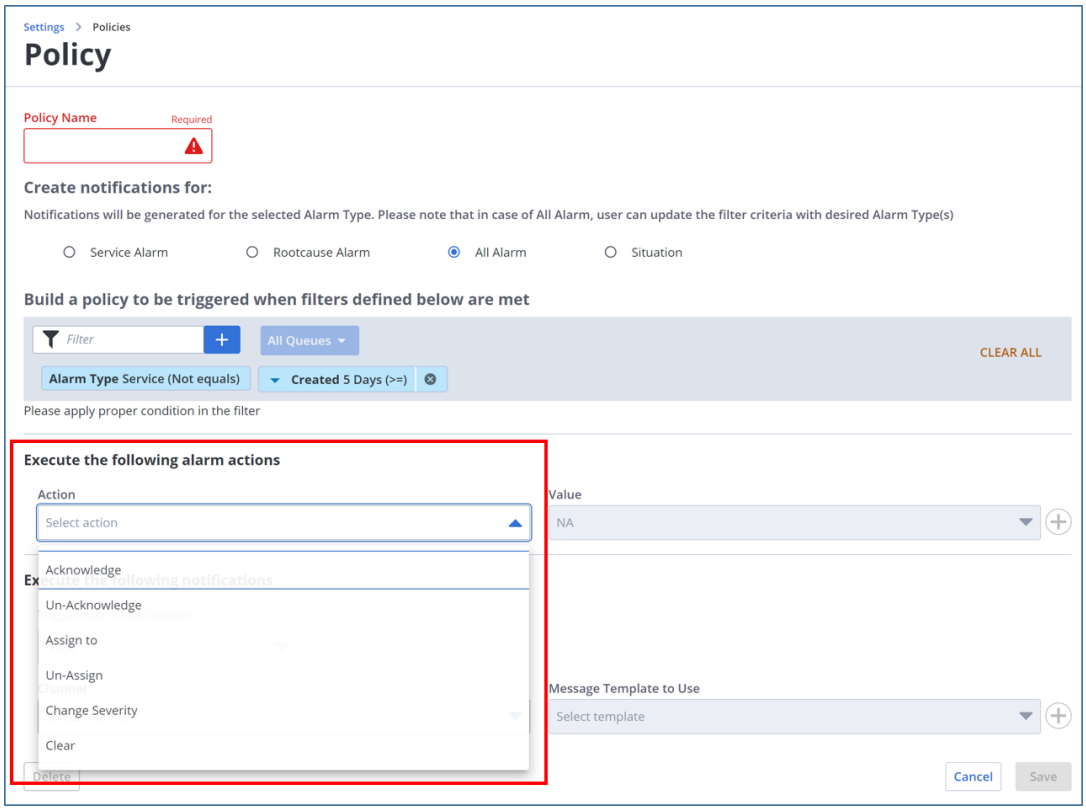
NOTE

- While filtering, the **AND** operator is used between attributes and the **OR** operator is used between the attribute values. For example,

  ```
  (Severity: Critical OR Major) AND (Alarm Type:"application" OR "fault")  AND (Message
   contains "sshd" OR Message does not contain "ALARM: [SYSTEMS]")
  ```

- For the **auto-closed alarms**, email and webhook notifications are not triggered even if the defined policy filter criteria are met. If there are any open tickets for the auto-closed alarms, those tickets get closed.

c) (**Only for Historical All Alarms**) Execute the alarm actions. To execute the actions, select the required actions from the list and provide the values. When the policy filter condition is met, the configured alarm action is executed. For example, you can create or update a policy to automatically acknowledge all alarms that are older than five days. When the policy filter criteria are met, all alarms older than five days are automatically acknowledged. The following alarm actions can be automatically performed on historical raw alarms when the filter condition is met:



- Alarms that have the created filter attribute defined in the policy are historical alarms.
- The **All Queues** and **Trigger Alarm Notification** sections are disabled if you automate the alarm actions.
- If multiple policies have the same criteria, then all the policies are randomly executed and may result in an infinite loop till the alarm is closed. So, ensure that the policy is unique.

d) (For **All Alarms**) Select the queue from **All Queues**.

NOTE

- You cannot change the options provided in the filters while using **All Queues**. For example, you cannot change the **Acknowledged** filter option (true, false) when selecting from **All Queues**.
- Only the alarm filters support asterisks but policy filters do not.
- Filters in the alarm queues and policies do not support regular expressions (Regex).

e) (For **All Alarms**) Select the rule attribute in the **Trigger Alarm Notification** field drop-down. A rule attribute determines when the application must trigger the alarm notification for the updated alarms:

- **None:** Select this option to deselect the rule attributes of an existing alarm policy. For example, if you have selected Severity Increases as the rule attribute during the policy creation, select None to clear this filter.
- **Severity Increases:** Select this option to trigger a notification whenever the severity level increases. This option is not available if the policy filter is **Severity: Critical**.
- **Severity Change:** Select this option to trigger a notification every time the severity level changes.

For more information about the Rule Attributes, see the Rule Attributes section.

f) (Optional) Select the Type (notification channel), Action (value for the notification channel), and Message template to use.

8. Click **Save.**

   The application creates the policy.

Rule Attributes

- These rule attributes apply only to the updated alarms and not the new alarms.
- The rule attributes override the policy filter. For example, if both the policy filter (**Severity: Critical**) and the rule attribute (**Severity Change**) are selected, then the rule attribute setting overrides the policy filter and sends notification every time the severity changes even though the policy filter is for Critical only.
- If you create a policy with these rule attributes (severity increase and severity change) and you associate this policy with multiple channels, then whenever the severity changes, a notification is triggered for all the associated channels.
- These rule attributes are not valid in the following cases:
  - **Only one value is supported for Severity.** If you select multiple values for Severity during the policy creation, these rules do not apply.
  - **Only the Equals operator is supported.** If you select operators such as Not equals, Contains for Severity, these rules do not apply.

Create a Policy from Alert Queue

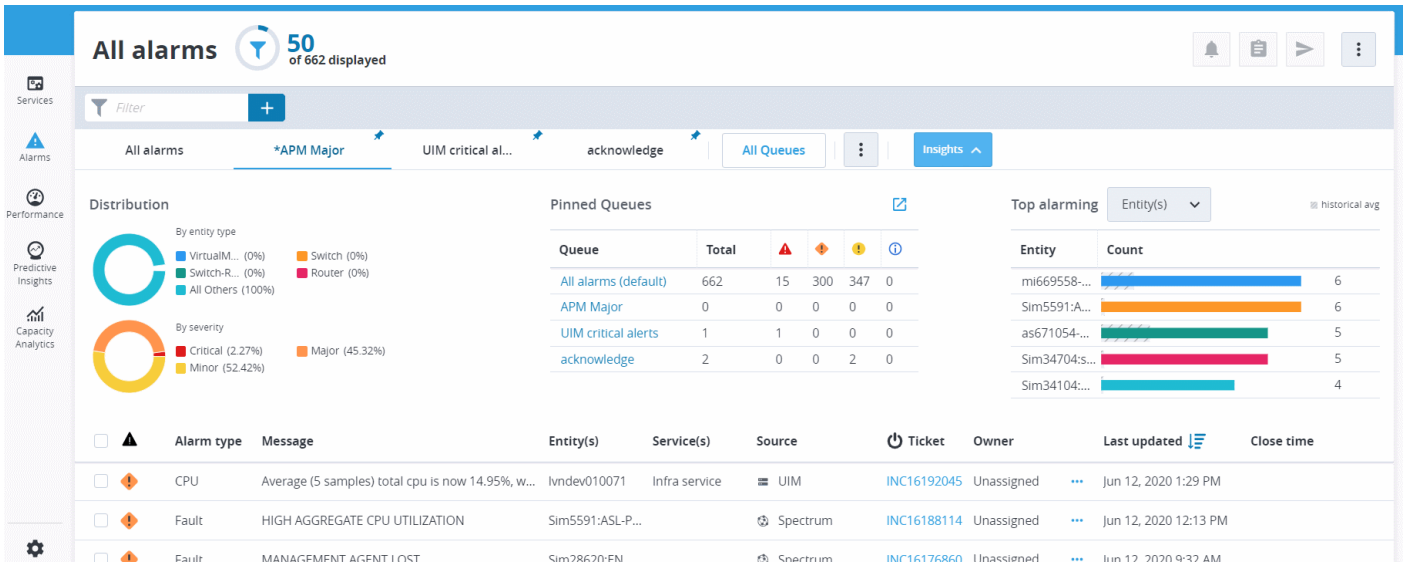
You can create a policy from the **All Alarms** page (**Alert Queue** section) in DX OI. Creating a policy from the **Alert Queue** saved view enables you to select the pre-populated view filters in your policy.

When you create a policy from the Alert Queue, review the following points:

- The filters in the DX OI Policy page are disabled. Pick only the filters from your Alert Queue saved view.
- If you delete the associated view in the All Alarms page, then the corresponding policy is also deleted.
- If you update the view with additional filter criteria, then the corresponding policy is also updated with that criteria.
- You can add multiple views to a single policy. The policy is not deleted until at least one view is associated with the policy. For example, View A and View B can be associated with Policy A. However, when you delete View A, the associated Policy A is not deleted since View B is still associated with the Policy. Instead, the filter criteria is updated to reflect the criteria only from View B.
- You can create multiple policies from the same saved view.

To create a policy for an alarm queue in DX OI, perform the following steps:

1. Click the current or required alarm queue on the dashboard.



2. Click icon and select **Create a policy**.
   The Policy page opens.
3. Follow the steps described in the Create Policy topic to create a policy.

Edit Policy

{"URL":["https://cloudmanagement/#/settings/policies/processingpolicies/*"],"description":"task.dita_109aaa5b-5743-42f2-a857-8d3d8da8f85f","new":"","new_video":"","admin":"","troubleshooting":{"masterkb":"","text":"","URL":[]},"pendo":"","video":[]}

To edit an existing policy, select the Alarm Type (Service Alarm, Rootcause Alarm, All Alarm, or Situation), modify the filters as required, and save the policy.

- If the **Alarm Type** for your existing policy is **Service Alarm** or **Rootcause Alarm**, then the respective Alarm Type is pre-populated and displayed along with the corresponding **Alarm Type: Service** filter.
- If the **Alarm Type** for your existing policy is not **Service Alarm** or **Rootcause Alarm**, then the **All Alarm** Alarm Type is pre-populated and displayed along with **Alarm Type: Service (Not equals)** filter.

  NOTE
  Service Alarms and Rootcause Alarms are not available for new tenants.

Suppress Notifications During Maintenance Schedule

You can use the **Policies** page to set up a maintenance period. You can choose to suppress alarm notifications during a maintenance period.

{"URL":["https://cloudmanagement/#/settings/policies/processingpolicies/*"],"description":"task.dita_faaefe0c-89f6-4ecd-a294-9ab5890e3e4e","new":"","new_video":"","admin":"","troubleshooting":{"masterkb":"","text":"","URL":[]},"pendo":"","video":[]}

The maintenance period stops all monitoring and metric calculations for the selected service, application, group, or device. During this period, to silence your alarms and alarm notifications, update your policy to include *maintenance* attribute to *false*.

To suppress notification during maintenance schedule, follow these steps:

1. Navigate to the Policies page and open the policy.
2. Search for the filter attribute Maintenance: false and add.
3. Click**Save**.

Supported Filter Attributes Reference

This section describes the different filter attributes that are supported based on the alarm type:

- Supported Filter Attributes for Service Alarm and Rootcause Alarm
- Supported Filter Attributes for Situations
- Supported Filter Attributes for All Alarm

Supported Filter Attributes for Service Alarm and Rootcause Alarm

The following table lists all the filter attributes that are supported for Service Alarm and Rootcause Alarm. Use these filter attributes to build a policy. The policy is triggered and a notification is sent when the defined filter is met. For example, if the filter attribute **Acknowledged State: Unacknowledged** is added to the policy, the policy is triggered and a notification is sent when this criterion is met.

NOTE
Service Alarms and Rootcause Alarms are not available for new tenants.

Filter	Description
Acknowledged State	Specify the acknowledgment state of the alarm. **Values:** true, false.
Affected Services	Specify the services which are impacted by an alarm.
Alarm ID	Specify the unique identification number of the alarm.
Alarm Status	Specify the status of the alarm. **Values:** Active, Closed.
Assign State	Specify the assigned state of the alarm. **Values:** Assigned, Unassigned.
Created (Time Elapsed)	Specify the time that has elapsed since the alarm was created. You can use this filter attribute to define a policy based on the age of the alarm. For example, specify the **Created (Time Elapsed)** as **Greater than or equal** to **30 Minutes ago** to notify when the created time for that alarm is equal to or is more than 30 minutes. • Creating age-based policies is not supported by Automatic. • The greater than or equal to operator can be used independently for an age-based policy. For example, you can select Created (Time Elapsed) and specify Greater than or equal to as 4 hours. However, the Less than or equal to operator has to be used in conjunction with the Greater than or equal to operator. For example, select Created (Time Elapsed), specify Greater than or equal to as 5 hours. Then click the + icon and specify Less than or equal as 12 hours.
Is Historical	Specify if the alarm is historical. **Values:** true, false
Maintenance Mode State	Specify the maintenance mode is on or off. **Values:** true, false.

Filter	Description
Message	Specify the alarm message.
Root Cause	Specify the root cause of the alarm.
Service Tags	Specify the service tags or user tags configured for the services. If the service has multiple tags, use the **contains** operator while creating the policy to include multiple tags. If the service has only one tag, then you may use the **equals** operator.
Severity	Specify the severity of the alarm. The following severity levels are supported: • Critical • Major • Minor • Informational • Warning
Ticket ID	Specify the ID generated by the ticketing system.
Ticket State	Specify the state of the ticket. **Values:** Ticketed, Unticketed.
Troubleshooter Name	Specify the name of the troubleshooter.
Visible	Specify if the alarm is visible. **Values:** true, false

Supported Filter Attributes for Situations

The following table lists all filter attributes that are supported for Situations. Use the filter attributes to build a policy. The policy is triggered and a notification is sent when the defined policy filter is met. For example, if the filter attribute **Acknowledged State: Unacknowledged** is added to the policy, the policy is triggered and a notification is sent when this criterion is met.

Filter Attribute	Description
Acknowledged State	Specify the acknowledgment state of the alarm. **Values:** Acknowledged, Unacknowledged.
Affected Services	Specify the services that are impacted by the alarm.
Age (In min)	Specify the age of the situation alarm. After the situation reaches the specified age, the notification channel triggers the automatic ticket creation process.
Alarm State	Specify the state of the alarm. **Values:** New, Updated, Closed.
Alarms Count	Specify the alarms count.
Annotation	Specify the annotation.
Assign State	Specify the assigned state of the alarm. **Values:** Assigned, Unassigned.
Closed Products	Specify the closed products.
Closure Ts	Specify the Closure Ts.
Cluster Previous Name	Specify the previous name of the cluster.

Filter Attribute	Description
Created (Time Elapsed)	Specify the time that has elapsed since the alarm was created. You can use this filter attribute to define a policy based on the age of the alarm. For example, specify the **Created (Time Elapsed)** as **Greater than or equal** to **30 Minutes ago** to notify when the created time for that alarm is equal to or is more than 30 minutes. • Creating age-based policies is not supported by Automatic. • The greater than or equal to operator can be used independently for an age-based policy. For example, you can select Created (Time Elapsed) and specify Greater than or equal to as 4 hours. However, the Less than or equal to operator has to be used in conjunction with the Greater than or equal to operator. For example, select Created (Time Elapsed), specify Greater than or equal to as 5 hours. Then click the + icon and specify Less than or equal as 12 hours.
Hosts	Specify the name of the host.
Initial Impacted Host	Specify the host that was impacted initially.
Initial Impacted Services	Specify the service that was impacted initially.
Initial Impacted Template	Specify the template that was impacted initially.
Is Closed	Specify if the alarm is closed. **Values:** true, false.
Is Force Closed	Specify if the alarm was force closed. **Values:** true, false.
Is Orphan	Specify if the alarm is an orphan. **Values:** true, false
Is Stable	Specify if the alarm is stable. **Values:** true, false
Maintenance Mode State	Specify if the maintenance mode is on or off. **Values:** true, false.
Message	Specify the time and date when the alarm was last updated.
Most Impacted Host	Specify the host that was most impacted.
Most Impacted Service	Specify the service that was most impacted.
Most Impacted Template	Specify the template that was most impacted.
Noisy	Specify if the alarm is noisy. **Values:** true, false
Primary Root Cause Service	Specify the primary root cause service.
Primary Root Cause Source	Specify the source of the primary root cause.
Root Cause Count	Specify the root cause count.
Root Cause Message	Specify the message of the root cause.
Root Cause Relative Confidence	Specify the relative confidence of the root cause.
Root Cause Score	Specify the score of the root cause.
Root Cause Severity	Specify the severity of the root cause.
Root Cause Sub Cluster ID	Specify the sub-cluster ID of the root cause.
Service	Specify the service which is impacted by an alarm.
Service Tags	Specify the service tags or user tags that are configured for the services. If the service has multiple tags, use the **contains** operator while creating the policy to include multiple tags. If the service has only one tag, then you may use the **equals** operator.

Filter Attribute	Description
Severity	Specify the severity level of the alarm. The following severity levels are supported: • Critical • Major • Minor • Informational • Warning
Situation ID	Specify the situation ID.
Situation Source	Specify the source of the situation.
Source	Specify the monitored device for which the alarm is flagged. For example, *testdevice1.example.net*
Stable Time	Specify the stable time.
Start Time	Specify the start time.
Subclusters Count	Specify the count of the subclusters.
Ticket State	Specify the state of the ticket. **Values:** Ticketed, Unticketed.
Timestamp	Specify the timestamp of the alarm.

Supported Filter Attributes for All Alarm

The following table lists all the filter attributes that are supported for All Alarm. Use these filter attributes to build a policy. The policy is triggered and a notification is sent when the defined policy filter is met. For example, if the filter attribute **Acknowledged State: Unacknowledged** is added to the policy, the policy is triggered and a notification is sent when this criterion is met.

Filter Attributes	Description
Acknowledged State	Specify the acknowledgment state of the alarm. **Values:** Acknowledged, Unacknowledged.
Action Status	Specify the status of the action.
Affected Services	Specify the services that are impacted by the alarm.
Agent Name	Specify the name of the DX APM agent.
Alarm Age	Specify the age of the alarm.
Alarm Description	Specify the alarm description.
Alarm Domain	Specify the domain from which the alarm is generated.
Alarm ID	Specify the unique identification number of the alarm.
Alarm Name	Specify the name of the alarm.
Alarm State	Specify the state of the alarm. **Values:** New, Updated, Closed.
Alarm Status	Specify the status of the alarm.
Alarm Type	Specify the type of alarm. **Values:** Anomaly, Application, Fault
Alarm Update	Specify the alarm update.
Alert External Id	Specify the external ID for the alert.
Application Name	Specify the application name in DX APM.
Assign State	Specify the assigned state of the alarm. **Values:** Assigned, Unassigned.

Filter Attributes	Description
Baseline	Specify the baseline.
Breached Threshold	Specify the threshold that was breached.
Cause Code	Specify the alarm cause code which is an 8-digit, hexadecimal code that identifies the probable cause of the alarm.
Caution Threshold	Specify the caution threshold.
Channels	Specify the channel.
Ci ID	Specify the ID of the Configuration Item (CI). A CI represents the component being monitored.
Ci Name	Specify the name of the CI.
Ci Type	Specify the type of the CI.
Ci Unique ID	Specify the unique ID of the CI.
Closed Time	Specify the closed time.
Component Name	Specify the name of the component.
Configuration Item	Specify the configuration item.
Configuration Item Type	Specify the configuration item type.
Created (Time Elapsed)	Specify the time that has elapsed since the alarm was created. You can use this filter attribute to define a policy based on the age of the alarm. For example, specify the **Created (Time Elapsed)** as **Greater than or equal** to **30 Minutes ago** to notify when the created time for that alarm is equal to or is more than 30 minutes. • Creating age-based policies is not supported by Automatic. • The greater than or equal to operator can be used independently for an age-based policy. For example, you can select Created (Time Elapsed) and specify Greater than or equal to as 4 hours. However, the Less than or equal to operator has to be used in conjunction with the Greater than or equal to operator. For example, select Created (Time Elapsed), specify Greater than or equal to as 5 hours. Then click the + icon and specify Less than or equal as 12 hours.
Cs ID	Specify the computer System (CS) ID.
Cs Key	Specify the Computer System key.
Current Trend Value	Specify the value of the current trend.
Custom_*n* *where n represents numbers 1-10.*	You can specify up to 10 custom attributes.
Custom Num 1, 2	
Daily Average	Specify the daily average.
Danger_Threshold	Specify the danger threshold.
Device ID	Specify the unique identifier of the device that is involved in the alarm.
Device Global ID	Specify the global ID of the device.
Device Local ID	Specify the local ID of the device.
Distribution Lists	Specify the distribution lists.
DocTypeID	Specify the doc type ID.

Filter Attributes	Description
DocTypeVersion	Specify the doc type version.
Domain	Specify the domain from which the alarm is generated.
Dst Address	Specify the destination address.
Dst Port	Specify the destination port.
Global ID	Specify the global ID of the alarm.
globalID Router	Specify the global ID of the router.
Group	Specify the group or groups to which the device belongs.
Group Ids	Specify the group Ids to which the device belongs.
Host Name (if present in the all_alarms index or if it already exists)	Specify the host name.
Hub	Specify the UIM hub from which the alarm is generated.
Intercept	Specify the intercept.
IP Address	Specify the IP address.
Landscape ID	Specify the Spectrum landscape ID from which the alarm is generated.
LastUpdatedTime	Specify the last updated time.
Level	Specify the level.
Location	Specify the location where the alarm is generated.
Maintenance Mode State	Specify if the maintenance mode is on or off. **Values:** true, false.
Management Module	Specify the management module in DX APM.
Message	Specify the alarm message.
Met Id	Specify the metric ID.
Metric External ID	Specify the external ID of the metric.
Metric Group ID	Specify the group ID of the metric.
Metric Name	Specify the name of the metric that is involved in the alarm.
Metric Type	Specify the type of metric that is involved in the alarm.
Metric Unit	Specify the unit of the metric that is involved in the alarm.
Metric Value	Specify the value of the metric that is involved in the alarm.
Model Name	Specify the model name of the device.
ModelTypeHandle	Specify the model type flag (Visible, Instantiable, and Derivable, No Destroy, Unique, and Required).
Notification Data	Specify the data that is sent with the alarm notification.
Occurrence	Specify the occurrence of the event.
Origin	Specify the UIM origin of the alarm.
Parent Device Type	Specify the type of the parent device.
Predicted Day	Specify the predicted day.
Predicted Value	Specify the predicted value.
Prediction Category	Specify the predicted category.
Prediction Timestamp	Specify the timestamp of the prediction.

Filter Attributes	Description
Probable Cause	Specify the probable cause.
Probe	Specify the name of the probe which is generating the alarm/ notification.
Process Name	Specify the process name.
Product	Specify the product from which the alarm is generated. For example, *UIM*.
Product Version	Specify the product version.
Robot	Specify the UIM robot from which the alarm is generated.
Rollup Algorithm	Specify the rollup algorithm.
Root Cause	Specify the root cause.
Samples	Specify the samples.
Service Tags	Specify the service tags or user tags configured for the services. If the service has multiple tags, use the **contains** operator while creating the policy to include multiple tags. If the service has only one tag, then you may use the **equals** operator.
Severity	Specify the severity level of the alarm. The following severity levels are supported: • Critical • Major • Minor • Informational • Warning
Slope	Specify the slope.
Source	Specify the monitored device for which the alarm is flagged. For example, *testdevice1.example.net*
Spectro Server	Specify the Spectrum SpectroSERVER name.
Src CIDR	Specify the Source CIDR.
Src Port	Specify the Source port.
Start Time	Specify the start time.
Subsystem	Specify the subsystem.
Subsystem ID	Specify the subsystem ID identifying which part of the system the alarm relates to.
Summary	Specify the summary.
Supp Key	Specify the suppression key.
Symptoms	Specify the symptoms.
Tags	Specify the tag that is associated with the alarm or notification.
Threshold	Specify the threshold.
Ticket ID	Specify the ID generated by the ticketing system.
Ticket State	Specify the state of the ticket. **Values:** Ticketed, Unticketed.
Time to Threshold	Specify the event violation rule that sends an alarm when a QoS metric is predicted to reach a set value within a user-defined time period.
TopologyModelNameString	Specify the name string of the topology model.
Total Points	Specify the total points.
Trend	Specify the trend.

Filter Attributes	Description
Troubleshooter Name	Specify the Spectrum Troubleshooter ID, associated with the alarm or notification.
User Clearable	Specify if the alarm is user-clearable.
User Tag 1	Specify the custom user tag (specified in Spectrum) associated with the alarm.
User Tag 2	Specify the custom user tag (specified in Spectrum) associated with the alarm
Vertex Attributes	Specify the vertex attributes.
Vertex ID	Specify the ID of the vertex.
Visible	Specify if the alarm is visible. **Values:** true, false

Message Templates

{"URL":["https://cloudmanagement/#/settings/messageTemplates"],"description":"concept.dita_3864ee15-2acb-4815-8503-f262f51df938","new":"","new_video":"","admin":"","troubleshooting":{"masterkb":"","text":"","URL":[]},"pendo":"","video":[]}

When an alarm policy meets the filter criteria that are defined in the policy, a notification is sent to the associated channels. A message template defines the content of the message that is sent when an alert occurs. You can use the Default Message Templates, or you can create your own Custom Message Templates with the Message Template Variables that can be reused across the notification channels.

- Default Message Templates
- Custom Message Templates
- Create a Copy of Existing Template
- Message Template Variables

Default Message Templates

DX Operational Intelligence includes default message templates for use. You may use these templates as is or you may create a copy of the templates. You cannot edit or delete the default templates.

NOTE

- To use the default templates, ensure that you have created the notification channel.
- If a message template variable has no information to be displayed, then that variable value substitution is omitted from the notification message.

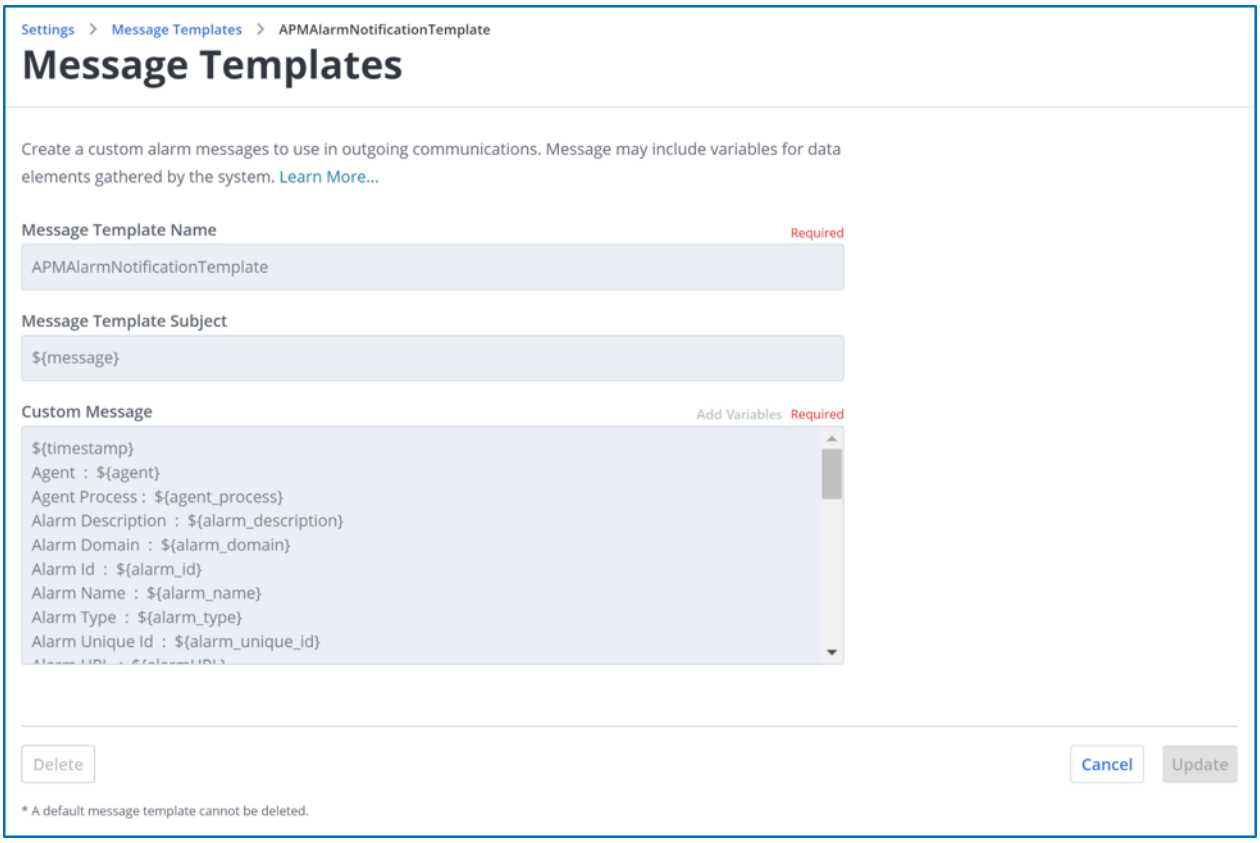
The following default message templates are available out-of-the-box:

- APM Alarm Notification Template
- Default Ticketing Management Template
- Notify Alarm Template
- Service Alarm Notification Template
- Situations Alarm Notification Template
- Spectrum Alarm Notification Template
- UIM Alarm Notification Template

APM Alarm Notification Template

The **APM Alarm Notification Template** defines the content of the message that is sent when an alert occurs in DX Application Performance Management. You cannot edit or delete this default template. However, you can make a copy of

the message template and add or delete the variables. To create a copy, click **+ Template** and then click the **Copy from Existing** button.



The following table lists the template variables that are included in the default message template:

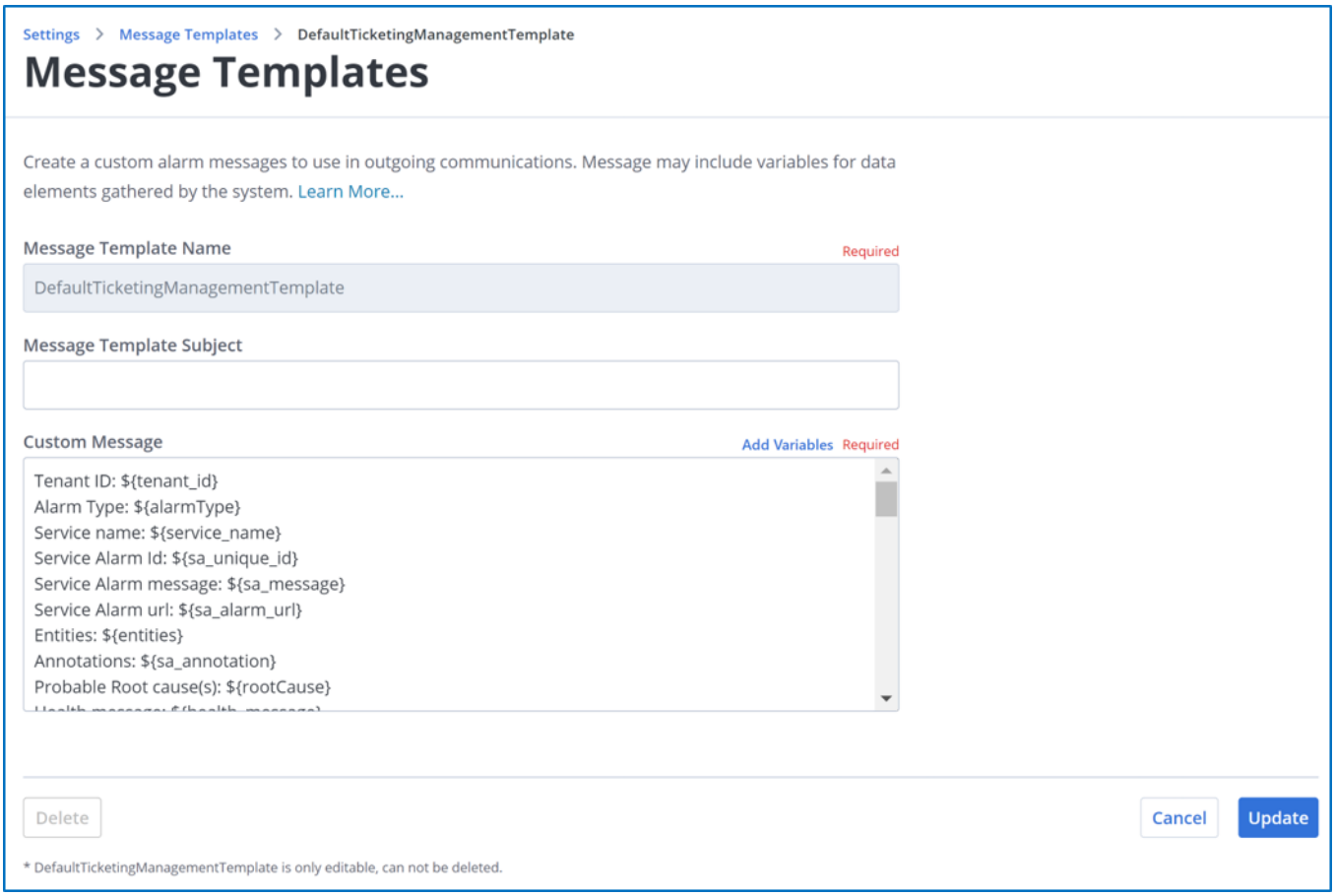
Template Variable	Description
${timestamp}	Displays the timestamp.
Agent : ${agent}	Displays the name of the DX APM agent.
Agent Process : ${agent_process}	Displays the name of the agent process.
Alarm Description : ${alarm_description}	Displays the alarm description.
Alarm Domain : ${alarm_domain}	Displays the domain from which the alarm is generated.
Alarm Id : ${alarm_id}	Displays the alarm ID.
Alarm Name : ${alarm_name}	Displays the name of the alarm.
Alarm Type: ${alarm_type}	Displays the alarm type.
Alarm Unique Id : ${alarm_unique_id}	Displays the unique identification number of the alarm.
Alarm URL : ${alarmURL}	Displays the alarm URL.
Alert Definition Link : ${alert_definition_link}	Displays the definition link for the alert.
Alert External Id : ${alert_external_id}	Displays the external ID for the alert.
Application Name : ${applicationName}	Displays the application name.
Breached Threshold : ${breached_threshold}	Displays the threshold that was breached.

Template Variable	Description
Caution Threshold : ${caution_threshold}	Displays the caution threshold.
Channels : ${channels}	Displays the channels.
CI Unique Id : ${ci_unique_id}	Displays the unique ID of CI.
Component Name : ${component_name}	Displays the component name.
Danger Threshold : ${danger_threshold}	Displays the danger threshold.
External Ids : ${external_ids}	Displays the external IDs.
Host Name : ${host}	Displays the hostname.
IP : ${ip}	Displays the IP address.
Isolation View Link : ${isolation_view_link}	Displays the link to the isolation view.
Message : ${message}	Displays the time and date when the alarm was last updated.
Metric External Id : ${metric_external_id}	Displays the external ID of the metric.
Metric Id : ${metric_id}	Displays the metric id.
Metric Name : ${metric_name}	Displays the name of the metric that is involved in the alarm.
Metric Value : ${metric_value}	Displays the value of the metric that is involved in the alarm.
Metric View Link : ${metric_view_link}	Displays the link to the Metric View.
Product : ${product}	Displays the product from which the alarm is generated.
Product Version : ${product_version}	Displays the product version.
Severity : ${severity}	Indicates the severity of an alarm. The following severity levels are supported: • Critical • Major • Minor • Informational • Warning
Status : ${status}	Displays the status.
Summary Alert : ${summaryAlert}	Displays the alert summary.
Vertex Attributes : ${vertex_attributes}	Displays the vertex attributes.
Vertex Id : ${vertex_id}	Displays the vertex ID.
Vertex Type : ${vertex_type}	Displays the vertex type.

Default Ticketing Management Template

The Default Ticketing Management Template defines the content of the message that is sent when an alert occurs for ITSM. You can edit only the subject and the custom message, but you cannot edit the message template name. Also, you cannot delete this template.

> **NOTE**
> If you update the Default Ticketing Management Template, you must update the linked notification channel for the changes to take effect.



The following table lists the template variables that are included in the default message template. To add more variables, use the **Add Variables** option.

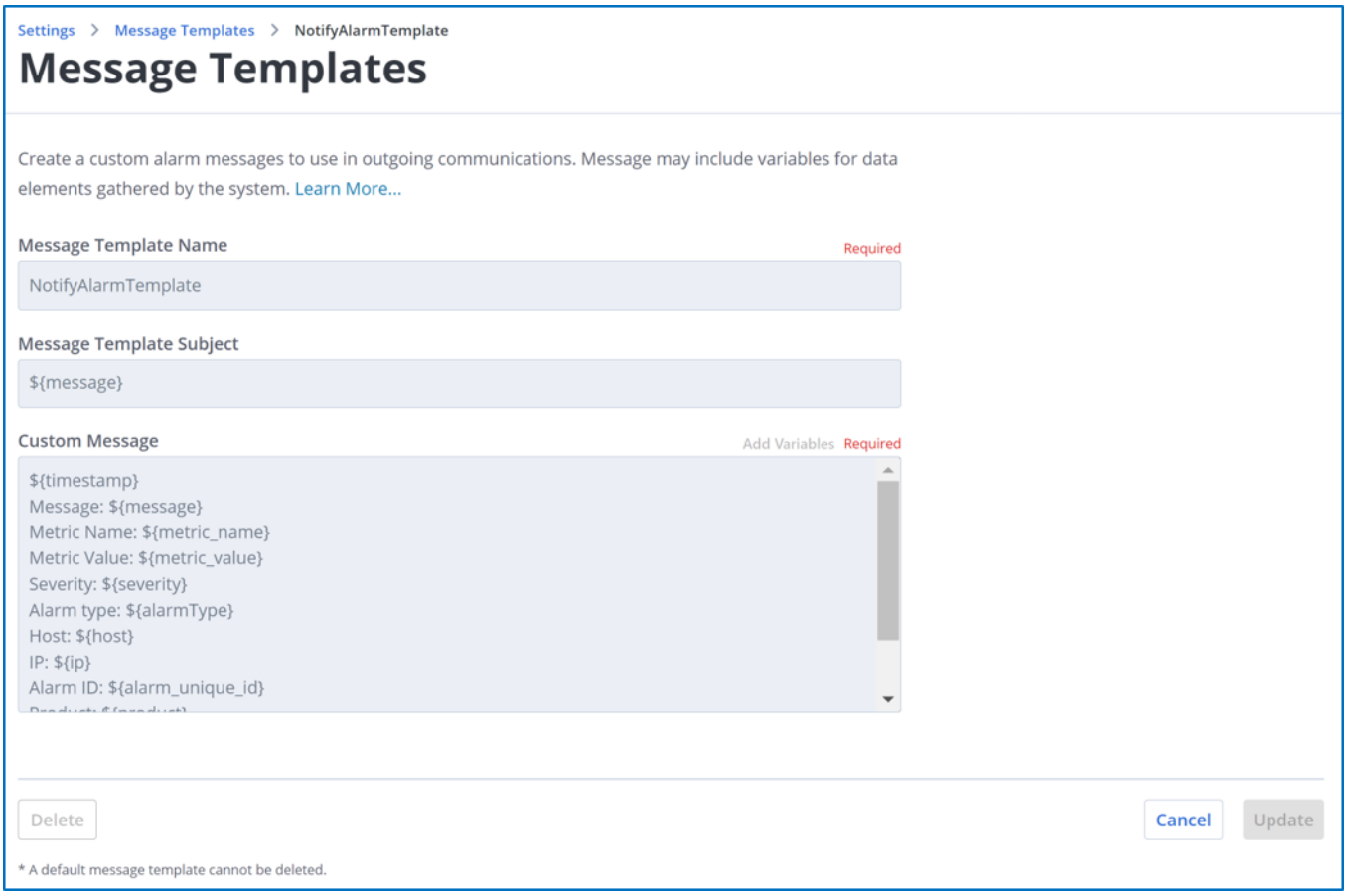
Template Variable	Description
Tenant ID: ${tenant_id}	Displays the tenant ID.
Alarm Type: ${alarmType}	Indicates the alarm type.
Service Name: ${service_name}	Indicates the name of the service.
Service Alarm Id: ${sa_unique_id}	Indicates the unique ID of the service alarm.
Service Alarm message: ${sa_message}	Displays the service alarm message.
Service Alarm url: ${sa_alarm_url}	Displays the service alarm URL.
Entities: ${entities}	Displays the entities. For the Service alarms, the ticket description contains all the entities that are affected initially. Entities that get affected later are added to the Work Notes section in ServiceNow and Working Log section in BMC Remedy.
Annotations: ${sa_annotation}	Displays the annotations.
Probable Root cause(s): ${rootCause}	Displays the probable root cause.
Health message: ${health_message}	Displays the health message.
Related alarms: ${related_alarms}	Displays the related alarms.
Situation Name: ${name}	Displays the name of the situation.
Situation Id: ${unique_id}	Displays the unique ID of the situation.

Template Variable	Description
Situation Entities: ${hosts}	Displays the situation entities.
Severity: ${severity}	Displays the severity.
Situation URL: ${situations_url}	Displays the situation URL.
Impacted Services: ${services_impacted}	Displays the services that are impacted.
Impacted Products: ${products}	Displays the products that are impacted.
Most Impacted Host: ${mostImpactedHost}	Displays the most impacted host.
Most Impacted Services: ${mostImpactedServices}	Displays the most impacted services.
SubCluster Count: ${subClustersCount}	Displays the count of the subcluster.
Ticket Logged By User: ${ticketUser}	Displays the name of the user who logged the ticket.
Email: ${emailId}	Displays the email ID.
Service Tags: ${service_tags}	Displays the service tags.

Template Variable	Description
Probable Root Cause Details:	• **Alarm Id: ${alarm_unique_id}:** Displays the alarm unique ID. • **Alarm Severity: ${severity}:** Displays the alarm severity. • **Alarm Type: ${alarmType}:** Displays the alarm type. • **Alarm Message: ${message}:** Displays the alarm message. • **Services Impacted: ${services_impacted}:** Displays the services impacted. • **Alarm url: ${all_alarm_url}:** Displays the alarm URL. • **Confidence: ${rc_confidence_score}:** Displays the root cause confidence score. • **Owner: ${troubleShooterName}:** Displays the name of the troubleshooter. • **Source Product: ${product}:** Displays the source product. • **Source Product version: ${product_version}:** Displays the source product version. • **Annotations: ${annotation}:** Displays the annotations. • **Metric Name: ${metric_name}:** Displays the metric name. • **Metric Value: ${metric_value}:** Displays the metric value. • **IP Address: ${ip}:**Displays the IP address. In the case of the OI Connector for DX APM, the IP template variable supports displaying the IP information only if present. If the DX APM agent does not send the IP address along with the host information to the Enterprise Manager for the associated APM vertices, the IP address may not be displayed. • **Host: ${host}:** Displays the host. • **Model Name: ${modelName}:** Displays the model name. • **Device Type: ${deviceType}:** Displays the device type. • **RC ClusterId: ${rc_subClusterId}:** Displays the root cause cluster ID. • **RC Host: ${rc_host}:** Displays the root cause host. • **RC Products: ${rc_products}:** Displays the root cause products. • **RC Name:${rc_name}:** Displays the root name. • **RC Severity: ${rc_severity}:** Displays the severity of the root cause. • **RC Services Impacted: ${rc_services_impacted}:** Displays the root cause services that are impacted. • **Ticket Logged By User: ${ticketUser}:** Displays the user who logged the ticket. • **Email: ${emailId}:** Displays the email ID. • **Spectro server: ${spectroSERVER}:**Displays the Spectro server. • **Service Tags: ${service_tags}:** Displays the service tags.

Notify Alarm Template

The **Notify Alarm Template** defines the content of the message that is sent when an alert occurs for **all alarms**. You cannot edit or delete this default template. However, you can make a copy of the message template and add or delete the variables. To create a copy, click **+ Template** and then click the **Copy from Existing** button.



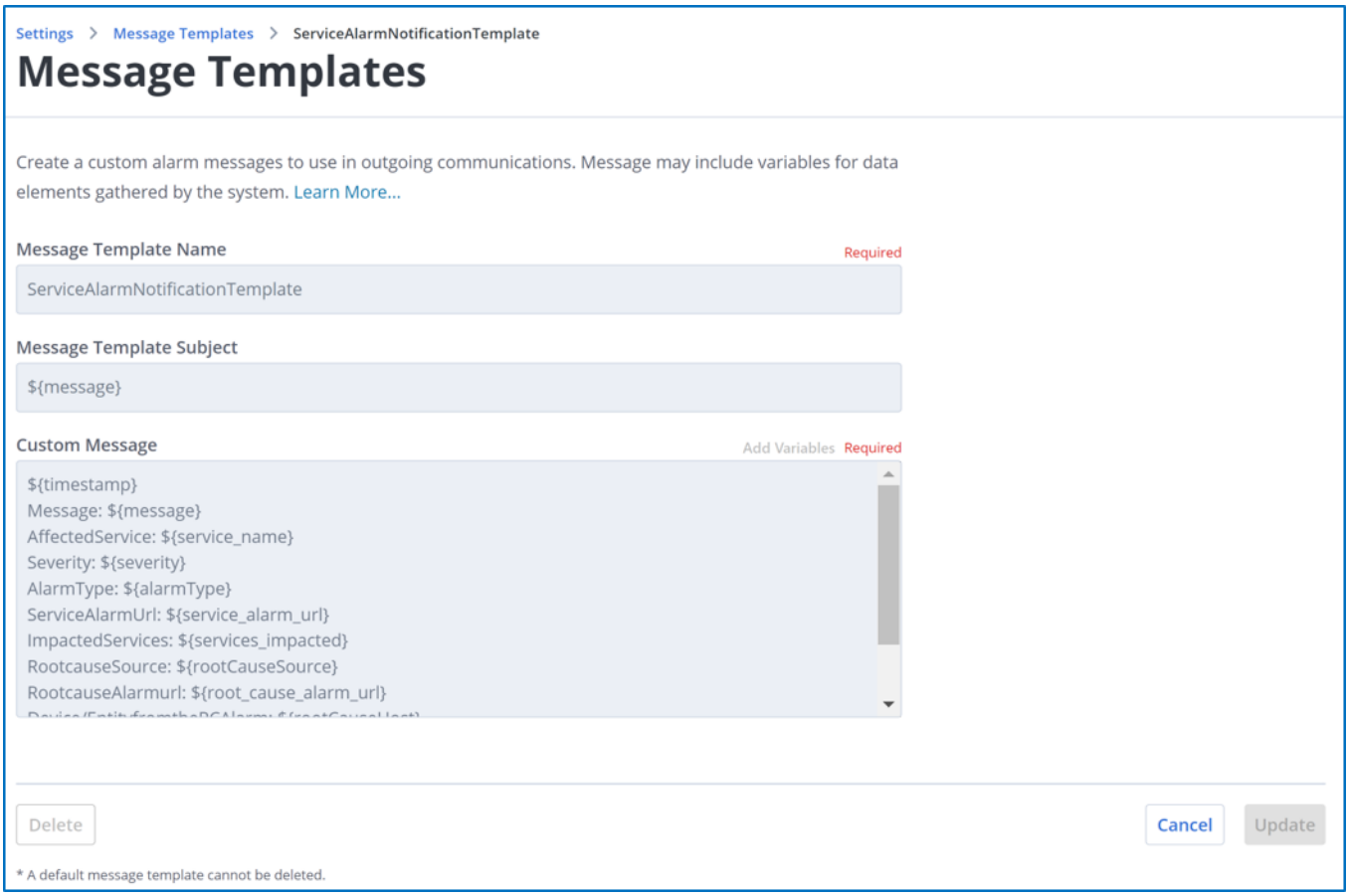
The following table lists the template variables that are included in the default message template:

Template Variable	Description
${timestamp}	Displays the timestamp.
Message : ${message}	Displays the time and date when the alarm was last updated.
Metric Name : ${metric_name}	Displays the name of the metric that is involved in the alarm.
Metric Value : ${metric_value}	Displays the value of the metric that is involved in the alarm.
Alarm Type: ${alarm_type}	Displays the alarm type.
Severity : ${severity}	Indicates the severity of an alarm. The following severity levels are supported: • Critical • Major • Minor • Informational • Warning
Host Name : ${host}	Displays the hostname.
IP : ${ip}	Displays the IP address.
Alarm Id : ${alarm_id}	Displays the alarm ID.
Product : ${product}	Displays the product from which the alarm is generated.
Product Version : ${product_version}	Displays the product version.

Service Alarm Notification Template

The **Service Alarm Notification Template** defines the content of the message that is sent when an alert occurs for a Service Alarm. You cannot edit or delete this default template. However, you can make a copy of the message template and add or delete the variables. To create a copy, click **+ Template** and then click the **Copy from Existing** button.

> **NOTE**
> The Service Alarm Notification template is not available for new tenants.



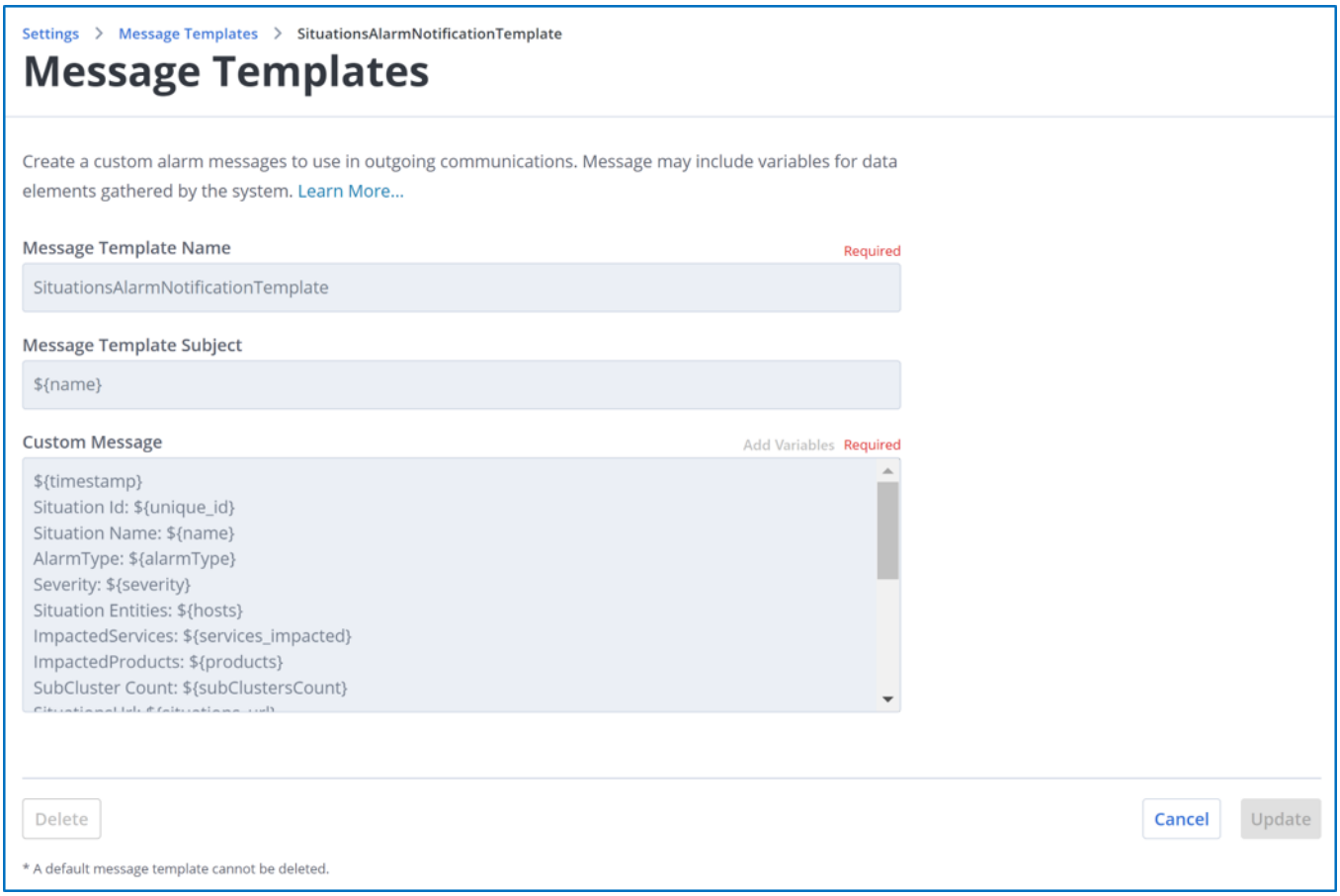
The following table lists the template variables that are included in the default message template:

Template Variable	Description
Timestamp:${timestamp}	Displays the timestamp.
Message: ${message}	Displays the message.
AffectedService: ${service_name}	Displays the name of the affected service.
Severity: ${severity}	Displays the severity.
Alarm Type: ${alarmType}	Displays the alarm type.
Service Alarm url: ${service_alarm_url}	Displays the service alarm URL.
Impacted Services: ${services_impacted}	Displays the services that are impacted.
RootcauseSource: ${rootCauseSource}	Displays the source of the root cause.
RootcauseAlarmurl: ${root_cause_alarm_url}	Displays the alarm URL of the root cause.
Device/EntityfromtheRCAlarm: ${rootCauseHost}	Displays the device or entity from the root cause alarm.

Template Variable	Description
MetricName: ${metric_name}	Displays the metric name,

Situations Alarm Notification Template

The Situation Alarm Notification Template defines the content of the message that is sent when an alert occurs for a situation. You cannot edit or delete this default template. However, you can make a copy of the message template and add or delete the variables. To create a copy, click **+ Template** and then click the **Copy from Existing** button.

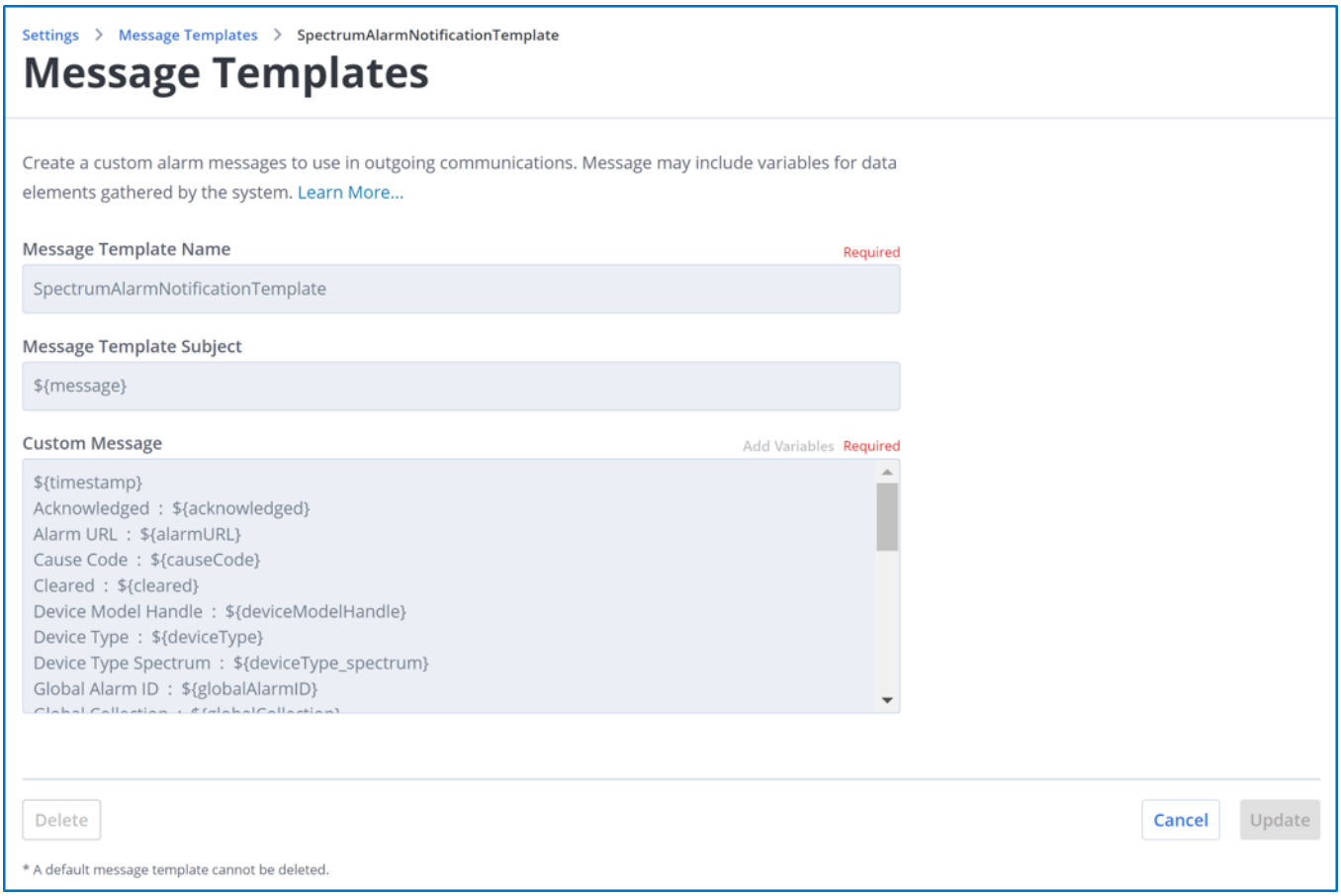


The following table lists the template variables that are included in the default message template:

Template Variable	Description
Acknowledged:${acknowledged}	Displays if the alarm was acknowledged.
Age (In min):${age}	Displays the age of the alarm.
Alarm Type: ${alarmType}	Displays the alarm type.
Alarms Count:${alarmsCount}	Displays count of the alarms.
Annotation:${annotation}	Displays the annotations.
Service Tags:${service_tags}	Displays the service tags.

Spectrum Alarm Notification Template

The Spectrum Alarm Notification Template defines the content of the message that is sent when an alert occurs in Spectrum. You cannot edit or delete this default template. However, you can make a copy of the message template and add or delete the variables. To create a copy, click **+ Template** and then click the **Copy from Existing** button.



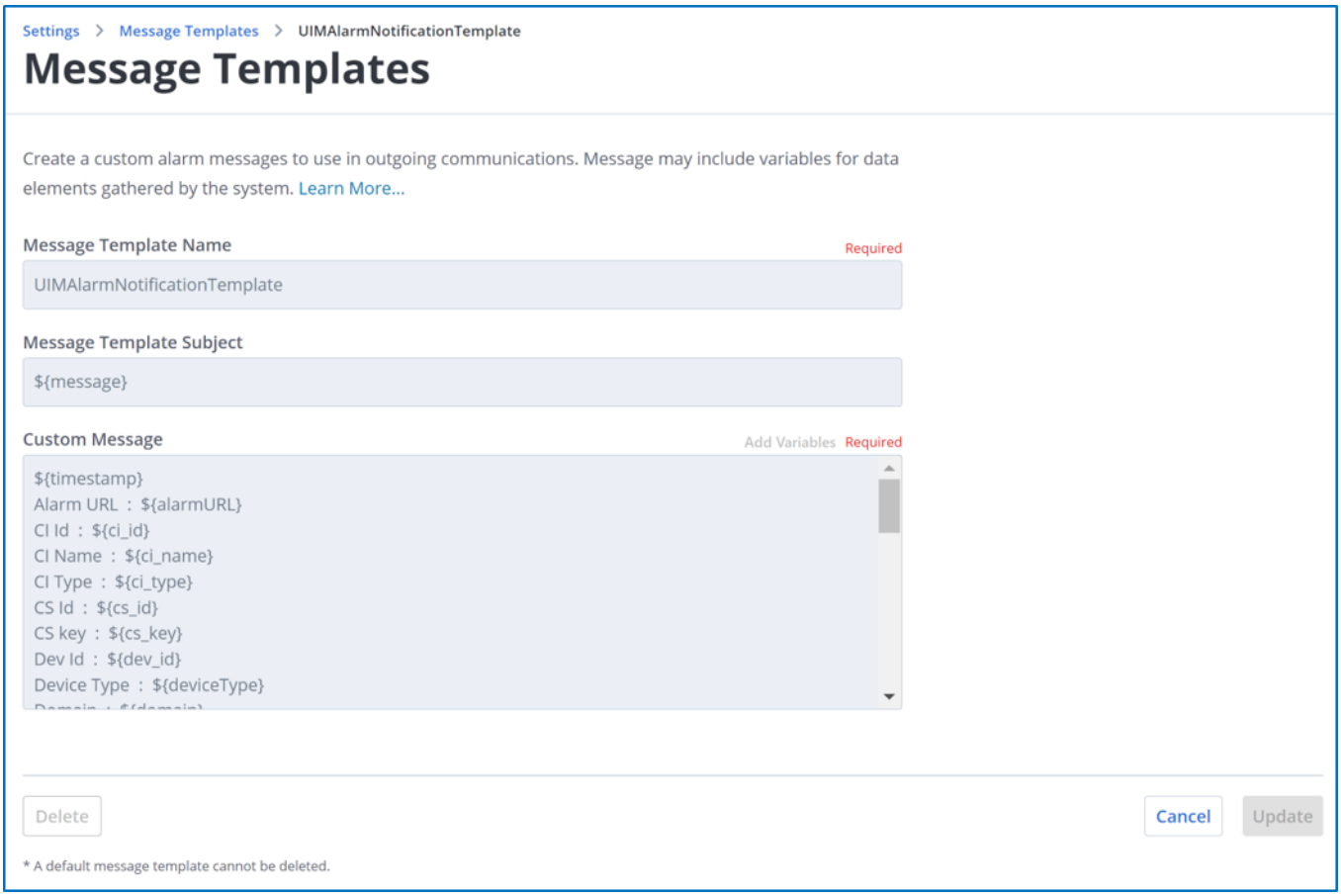
The following table lists the template variables that are included in the default message template:

Template Variable	Description
${timestamp}	Displays the timestamp.
Acknowledged:${acknowledged}	Displays if the alarm was acknowledged.
Alarm URL : ${alarmURL}	Displays the alarm URL.
Cause Code : ${causeCode}	Displays the cause code.
Cleared : ${cleared}	Displays if the alarm is cleared.
Device Model Handle : ${deviceModelHandle}	Displays the device model.
Device Type : ${deviceType}	Displays the type of device that is involved in the alarm.
Device Type Spectrum : ${deviceType_spectrum}	Displays the device type in Spectrum.
Global Alarm ID : ${globalAlarmID}	Displays the global ID of the alarm.
Global Collection : ${globalCollection}	Displays the global collection.
Global Collection Keys : ${collectionUniqueKeyString}	
Group Id : ${group_id}	Displays the group Ids to which the device belongs.

Template Variable	Description
Host Name : ${host}	Displays the hostname.
IP : ${ip}	Displays the IP address.
Landscape Id : ${landscapeID}	Displays the Spectrum landscape ID from which the alarm is generated.
Location String : ${locationString}	Displays the location.
Message : ${message}	Displays the time and date when the alarm was last updated.
Model Handle : ${modelHandle}	Displays the model handle.
Model Name : ${modelName}	Displays the name of the modeled device.
Model Type Handle : ${modelTypeHandle}	Displays the model type flag (Visible, Instantiable, and Derivable, No Destroy, Unique, and Required).
Model Type Name : ${modeltypeName}	Displays the model type name.
Product : ${product}	Displays the product from which the alarm is generated.
Product Version : ${product_version}	Displays the product version.
Severity : ${severity}	Displays the severity of an alarm. The following severity levels are supported: Critical, Major, Minor, Informational, Warning
Spectro Server : ${spectroSERVER}	Displays the Spectrum SpectroSERVER name.
Start Time : ${startTime}	Displays the start time.
Status : ${status}	Displays the status.
Topology Model Name String : ${topologyModelNameString}	Displays the name string of the topology model.
User Clearable : ${userClearable}	Displays whether the alarm is user-clearable.

UIM Alarm Notification Template

The UIM Alarm Notification Template defines the content of the message that is sent when an alert occurs in UIM. You cannot edit or delete this default template. However, you can make a copy of the message template and add or delete the variables. To create a copy, click **+ Template** and then click the **Copy from Existing** button.



The following table lists the template variables that are included in the default message template:

Template Variable	Description
${timestamp}	Displays the timestamp.
Alarm URL : ${alarmURL}	Displays the alarm URL.
CI Id : ${ci_id}	Displays the ID of CI (Configuration Item). A CI represents the component being monitored.
CI Name : ${ci_name}	Displays the CI name.
CI Type : ${ci_type}	Displays the CI type.
CS Id : ${cs_id}	Displays the CS (Computer System) ID.
CS key : ${cs_key}	Displays the CS key.
Dev Id : ${dev_id}	Displays the unique identifier of the device that is involved in the alarm.
Device Type : ${deviceType}	Displays the type of device that is involved in the alarm.
Domain : ${domain}	Displays the domain from which the alarm is generated.
Group : ${group}	Displays information about the group or groups to which the device belongs.
Group Id : ${group_id}	Displays the group Ids to which the device belongs.
Host Name : ${host}	Displays the hostname.
Hub : ${hub}	Displays the UIM hub from which the alarm is generated.

Template Variable	Description
IP Address : ${ip}	Displays the IP address.
Level : ${level}	Displays the level.
Message: ${message}	Displays the time and date when the alarm was last updated.
Met Id : ${met_id}	Displays the metric ID.
Metric Name : ${metric_name}	Displays the name of the metric that is involved in the alarm.
Metric Type : ${metric_type}	Displays the type of metric that is involved in the alarm.
Metric Unique Id : ${metric_unique_id}	Displays the unique ID of the metric.
Metric Unit : ${metric_unit}	Displays the unit of the metric that is involved in the alarm.
NIM Id : ${nimid}	Displays the NIM ID.
Occurrence : ${occurrence}	Displays the occurrence.
Origin : ${origin}	Displays the UIM origin of the alarm.
Probe : ${probe}	Displays the probe name which is generating the alarm/ notification.
Product : ${product}	Displays the product from which the alarm is generated.
Product Version : ${product_version}	Displays the product version.
Robot : ${robot}	Displays the UIM robot from which the alarm is generated.
Severity : ${severity}	Indicates the severity of an alarm. The following severity levels are supported: Critical, Major, Minor, Informational, and Warning
Source : ${source}	Displays the monitored device for which the alarm is flagged. For example, testdevice1.example.net
Status : ${status}	Displays the alarm status.
Subsystem : ${subsystem}	Displays the subsystem.
Subsystem ID : ${subsystemID}	Displays the subsystem ID, identifying which part of the system the alarm relates to.
Supp Key : ${supp_key}	Displays the suppression key.
Time Origin : ${time_origin}	Displays the time origin.
Visible : ${visible}	Displays if the alarm is visible.

Update Default Templates

You can update the default message template as per your requirements.

Consider the following points before you update the default templates

{"URL":["https://cloudmanagement/#/settings/messageTemplates/standardalertmessage/*"],"description":"task.dita_8dfb8cd6-70ec-42c0-adde-238263ed83c4","new":"","new_video":"","admin":"","troubleshooting":{"masterkb":"","text":"","URL":[]},"pendo":"","video":[]}

- To use the default templates, ensure that you have created the notification channel.
- You can edit the template subject, add or delete the template variables in the custom message, and associate this template with a channel.

NOTE

- Only in the **Service Alarm Notification** and the **Situations Alarm Notification** templates, you can edit the **Make as default** option and the **Message Template Name**.
- In the **NotifyAlarmTemplate** message template, use the **Add Variables** option to add the **Alarm type** variable.

- You can add or remove the template variables as required. You can add the variables using the Add Variable option or you can add them manually. Only the following characters are supported for custom variables that you add manually:
  - alphabets (a .. z, A .. Z)
  - numeric (0 .. 9)
  - hyphen (-)
  - underscore (_)

  If the custom variable of a message template includes any unsupported characters, testing or creating a channel using that message template fails with an error.
- You cannot delete the default templates.
- If you update the **Default Ticketing Management Template**, you must update the linked notification channel for the changes to take effect.

To update the default template, follow these steps:

1. Log in to DX Operational Intelligence as a Tenant Administrator and navigate to Settings Page.
2. Click **Create** in the Message Templates tile.

   The Message Templates page with the list of existing template appear.
3. Click the default template that you want to edit.
4. Update the following information as per your requirements:
   a) Select **Make as Default** checkbox to apply this message template to the new channels by default.
   b) Message template name: Edit the template name if necessary.

      NOTE
      You can edit only the names of the Service Alarm Notification Template and the Situations Alarm Notification Template.

   c) Enter the text that you want in the subject line of the email in the Message Template Subject field.

      NOTE
      (Only for ServiceNow) You can edit the message template subject to provide more details. For example, you can provide the subject as ***New ${severity} situation ${name} (ID: ${unique_id}) impacting ${services_impacted}.*** In ServiceNow, this subject is reflected in the **Short Description** field. If the added template variable does not have any value, then in the **Short Description** field, that variable is displayed as **NA**. For example, Impacting Services = NA.
5. Update the following information in the Custom Message Section:
   - Add Variables: You can use the variables that are mentioned in the template by default or update as per your requirements. Click the **Add Variables** option to add or remove the variables.

     For NotifyAlarmTemplate, SituationsAlarmNotificationTemplate, and ServiceAlarmNotificationTemplate message templates, use the **Add Variables** option to add the **Service Tags** variable to the custom message.

     For more information about the template variables, see Template Variables.
   - Enter the message: You can copy and paste the message. For example, to include the alarm severity and error message, create a custom message as follows. In the Custom Message section, enter the following information: Warning! A $severity alarm has been triggered.

     ```
     Warning! A $severity alarm has been triggered.
     ```

```
Alarm message: $message
```

When an alarm occurs, the following message is displayed:

```
Warning! A Critical alarm has been triggered.

Alarm message: Server trexy-ab is down.
```

NOTE
If a particular variable has no information to be displayed, then that variable value substitution is omitted from the notification message. In the case of the OI Connector for DX APM, the IP template variable supports displaying the IP information only if present. If the DX APM agent does not send the IP address along with the host information to the Enterprise Manager for the associated APM vertices, the IP address may not be displayed.

6. Select one or more channels that you want to link to this template in **Linked Notification Channels** section.
7. Click **Update**.
   The application updates the selected default template with these changes.

Custom Message Templates

In addition to the default message templates, you can create your own custom message templates using the template variables. You can edit and delete these templates. The **Delete** option for a custom message template is enabled only when a template is not associated with any channel.

- Create Custom Template
- Update Custom Template

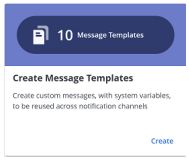
Create Custom Template

Tenant Administrators can create custom message templates using template variables.

{"URL":["https://cloudmanagement/#/settings/messageTemplates/standardalertmessage"],"description":"task.dita_eef44a55-848e-4b9f-90e5-fd04d5ef46b5","new":"","new_video":"","admin":"","troubleshooting":{"masterkb":"","text":"","URL":[]},"pendo":"","video":[]}

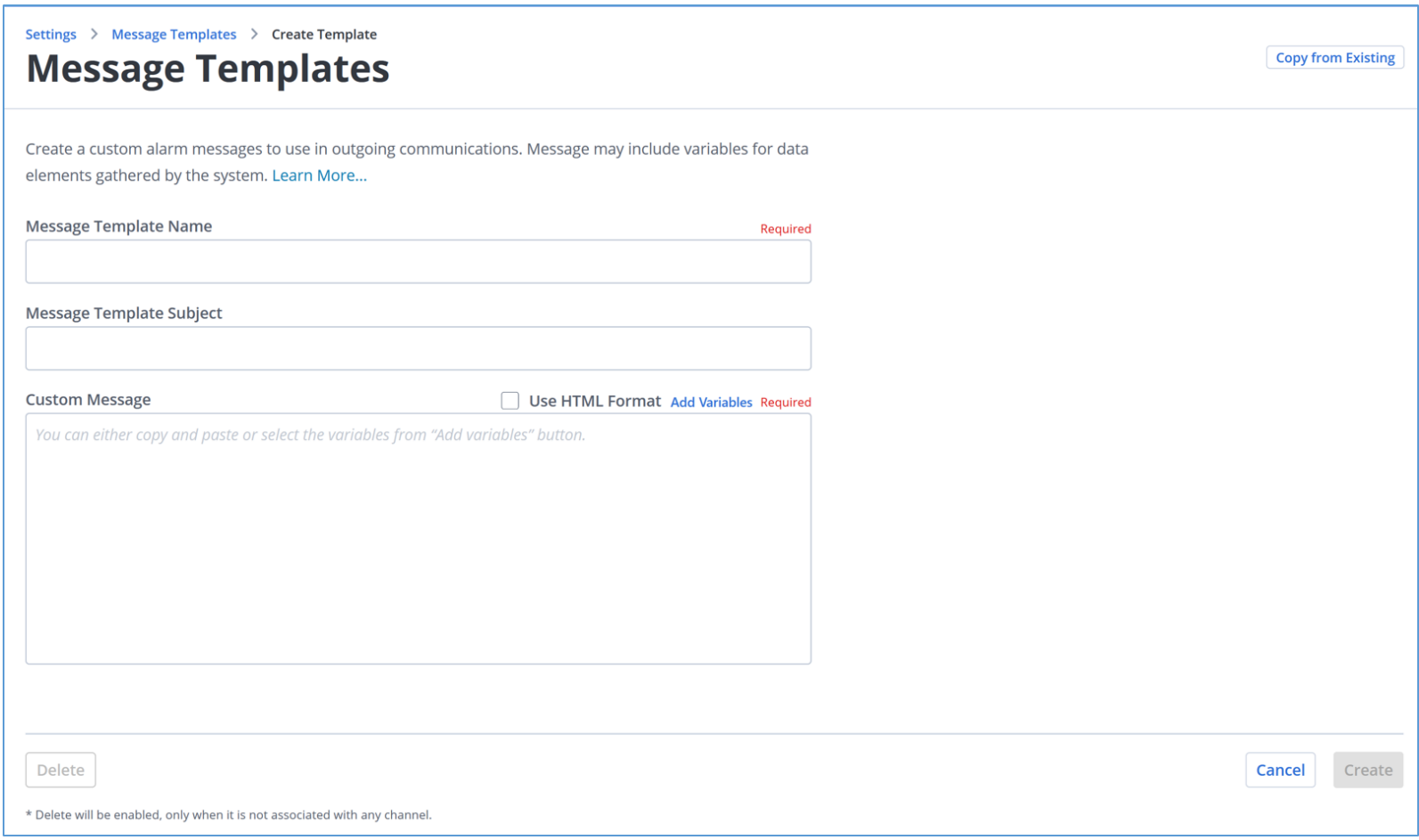
To create a custom message template, follow these steps:

1. Log in to DX Operational Intelligence as a Tenant Administrator and navigate to the Settings page.
2. Click Create in the Message Templates tile.



3. Click

.
   The application displays the Create Template page.



4. Provide a unique name for the template in the Message Template Name.
5. Enter the subject that you want in the Message Template Subject.

   (Only for ServiceNow) You can edit the message template subject to provide more details. For example, you can provide the subject as **New ${severity} situation ${name} (ID: ${unique_id}) impacting ${services_impacted}**. In ServiceNow, this subject is reflected in the **Short Description** field.

   If the added template variable does not have any value, that variable is displayed as NA in the Short Description field. For example, Impacting Services = NA.
6. Define the custom message in the **Custom Message** section. You can enter custom variables manually, or you can use the **Add Variable** option. The variables in the alarm message help the support personnel to triage and fix the issues. For more information, see the Message Template Variables section.
   - Ensure there is no space in the Tenant ID variable. Enter the Tenant ID variable as ${tenant_id}.
   - If a message template variable has no information to be displayed, then that variable value substitution is omitted from the notification message.

   a) Enter the custom variables manually.

      Only the following characters are supported for custom variables that you add manually:

- alphabets (a .. z, A .. Z)
- numeric (0 .. 9)
- hyphen (-)
- underscore (_)

If the custom variable of a message template includes any unsupported characters, testing or creating a channel using that message template fails with an error.

b) Select the **Use HTML Format** option and provide the HTML payload to send the message content in an HTML format. For example, send the message in the table format.

Consider the following points for the custom HTML payload:

- When you select this option, the **Add Variables** option is not available.
- All the HTML elements must have a closing tag.
- The HTML tags are case-sensitive.
- All the HTML tag-specific attribute values should be enclosed in double quotes.
- All the HTML elements must be properly nested.
- To use paragraphs or plain text, use the appropriate HTML tags.

c) Click **Add Variable** and select one or more variables that you want to include in the message. The message variables are grouped under the following categories: **All Alarm**, **Service Alarm**, **Situation**, and **CI Attributes**. However, for the CI Attributes to be displayed, you must register them. For more information about how to register the CI attributes, see the Add or Update the CI Attributes section.

- To add the **Service Tags** variable to the custom message in the NotifyAlarmTemplate, SituationsAlarmNotificationTemplate, and ServiceAlarmNotificationTemplate message templates, use the **Add Variables** option. For more information, see the Message Template Variables section.
- If a particular variable has no information to be displayed, then that variable value substitution is NA. In the case of the OI Connector for DX APM, the IP template variable supports displaying the IP information only if present. The IP address may not be displayed if the DX APM agent does not send the IP address along with the host information to the Enterprise Manager for the associated APM vertices.

7. Click **Create**.

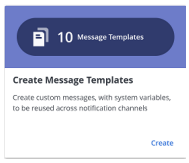
   The application creates the message template.

Update Custom Template

Tenant Administrators can update custom message templates.

To update the custom message template, follow these steps:

1. Log in to DX Operational Intelligence as a Tenant Administrator and navigate to the Settings Page.
2. Click Create in the Message Templates tile.



3. Open the message template you want to edit.

   The message variables are grouped under the following categories: **All Alarm**, **Service Alarm**, **Situation**, and **CI Attributes**. However, for the CI Attributes to be displayed, you must register them. For more information about how to register the CI attributes, see the Add or Update the CI Attributes section.

   The application displays the Edit Template page.

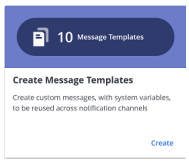
4. Update the message template.
5. Click **Update**.
   The message template is updated.

Create a Copy of Existing Template

You can create a copy of an existing default or custom message template.

Follow these steps:

1. Log in to **DX Operational Intelligence** as a Tenant Administrator.
2. Navigate to the **Settings** page.
3. Click **Create** in the Message Templates tile.



4. Click

   The application displays the Create Template page.
5. Click **Copy from Existing**.
   A list of all the available message templates is displayed.
6. Select the template you want to copy from and click **Apply**.
   The Create Template page appears with Copy appended to the template name.
7. Enter the following information:
   a) **Message Template Subject:** Enter the text you want in the email's subject line.
   b) **Custom Message:** You can copy and paste the message or use **Add Variables**. Alternatively, you can add the HTML payload to send the message in the HTML format.

      The message variables are grouped under the following categories: **All Alarm**, **Service Alarm**, **Situation**, and **CI Attributes**. However, for the CI Attributes to be displayed, you must register them. For more information about how to register the CI attributes, see the Add or Update the CI Attributes section.
8. Click **Create**.
   A copy of the message template is created.

Message Template Variables

You can use the following variables in a custom message. These variables are categorized based on the alarm type:

- All Alarm
- Service Alarm
- Situations

Template Variables for All Alarm

This section describes the template variables that are supported for All Alarms.

NOTE
If a message template variable has no information to be displayed, then that variable value substitution is omitted from the notification message.

DX Operational Intelligence

The following table lists the template variables that provide data from DX Operational Intelligence:

Text	Key	Description
Acknowledged	${acknowledged}	Indicates whether an alarm has been acknowledged.
Action Status	${action_status}	Displays the status of the action.
Agent	${agent}	Displays the name of the DX APM agent.
Agent Process	${agent_process}	Displays the name of the DX APM agent.
Alarm Age	${alarmAge}	Displays the age of the alarm.
Alarm Closure Type	${alarm_closure_type}	Displays the alarm closure type.
Alarm Description	${alarm_description}	Displays the alarm description.
Alarm Domain	${alarm_domain}	Displays the domain from which the alarm is generated.
Alarm Name	${alarm_name}	Displays the name of the alarm.
Alarm State	${alarmState}	Displays the state of the alarm.
Alarm Type	${alarmType}	Displays the alarm type. **Values:** Anomaly, Application, Fault
Alarm URL	${alarmURL}	Displays the alarm URL.
Alarm Unique Id	${alarm_unique_id}	Displays the unique identification number of the alarm.
Alarm Update	${alarm_update}	Displays the alarm update.
Alert Definition Link	${alert_definition_link}	Displays the alert definition link.
Alert External Id	${alert_external_id}	Displays the external ID for the alert.
Anomaly Algorithm Type	${algorithm}	Displays the algorithm type for the anomaly.
Annotation	${annotation}	Displays the annotation.
Anomaly Closure Type	${anomaly_closure_type}	Displays the anomaly closure type.
APM Alarm Unique ID	${apm_alarm_unique_id}	Indicates the unique alarm ID in DX APM.
Application Name	${applicationName}	Indicates the application name in DX APM.
Automic Jobs	${automicJobs}	
Baseline	${baseline}	Displays the baseline.
Breached Threshold	${breached_threshold}	Displays the threshold that was breached.
CI ID	${ci_id}	Displays the ID of CI (Configuration Item). A CI represents the component being monitored.
CI Name	${ci_name}	Displays the CI name.
CI Type	${ci_type}	Displays the CI type.
CI Unique ID	${ci_unique_id}	Displays the unique ID of CI.
CS ID	${cs_id}	Displays the CS (Computer System) ID.
CS Key	${cs_key}	Displays the CS key.

Text	Key	Description
Cause Code	${causeCode}	Displays the alarm cause code which is an 8-digit, hexadecimal code that identifies the probable cause of the alarm.
Caution Threshold	${caution_threshold}	Displays the caution threshold.
Channels	${channels}	Displays the channels.
Cleared	${cleared}	Displays if cleared or not.
Closed Time	${closedTime}	Displays the closed time.
Collection Unique Key String	${collectionUniqueKeyString}	
Component Name	${component_name}	Displays the component name.
Configuration Item	${configuration_item}	Displays the configuration item.
Configuration Item Type	${configuration_item_type}	Displays the configuration item.
Configuration Item Type	${configuration_item_type}	Displays the Configuration Item type.
Correlated External Id	${correlated_external_id}	Displays the correlated external ID.
Correlated External Ids	${correlated_external_ids}	Displays the correlated external IDs.
CS ID	${cs_id}	Displays the CS ID.
CS Key	${cs_key}	Displays the CS key.
Custom_n *where n represents numbers 1-10.*	${custom_n}	You can specify up to 10 custom attributes.
Custom Num n	${custom_num_n}	
Daily Average	${dailyAverage}	Displays the daily average.
Danger Threshold	${danger_threshold}	Displays the danger threshold.
Device Id	${dev_id}	Displays the unique identifier of the device that is involved in the alarm.
Device Global ID	${deviceGlobalID}	Displays the global ID of the device.
Device Local ID	${deviceLocalID}	Displays the local ID of the device.
Device Model Handle	${deviceModelHandle}	Displays the device model handle.
Device Type	${deviceType}	Displays the type of device that is involved in the alarm.
Device Type Spectrum	${deviceType_spectrum}	Displays the device type in Spectrum.
Direction	${direction}	Displays the direction.
Distribution Lists	${distributionLists}	
Doc Type ID	${doc_type_id}	Displays the ID of the doc type.
Doc Type Version	${doc_type_version}	Displays the version of the doc type.
Domain	${domain}	Displays the domain from which the alarm is generated.
Dst Address	${dstAddr}	Displays the destination address.
Dst Port	${dstPort}	Displays the destination port
Exists Metadata	${exists_metadata}	Displays if metadata exists.
External Id	${external_id}	Displays the external ID.
External Ids	${external_ids}	Displays the external IDs.
External Id	${externalid}	Displays the external ID.
Global Alarm ID	${globalAlarmId}	Displays the global ID of the alarm.

Text	Key	Description
Global ID	${globalID}	Displays the global ID of the alarm.
Global ID Router	${globalID_router}	Displays the global ID of the router.
Group	${group}	Displays information about the group or groups to which the device belongs.
Group Id	${group_id}	Displays the group Ids to which the device belongs.
Host	${host}	Displays the hostname.
Hub	${hub}	Displays the UIM hub from which the alarm is generated.
Ingestion Index	${ingestion_index}	Displays the ingestion index.
Initial Message	${initial_message}	Displays the initial message.
Instance Name	${instancename}	Displays the instance name.
Intercept	${intercept}	
IP	${ip}	Displays the IP address.
Isolation View Link	${isolation_view_link}	Displays the isolation view link.
Landscape ID	${landscapeID}	Displays the Spectrum landscape ID from which the alarm is generated.
Last Suppressed Severity	${last_supressed_severity}	Displays the last suppressed severity.
Last Suppressed Timestamp	${last_supressed_timestamp}	Displays the timestamp of the last suppressed severity.
Level	${level}	Displays the level.
Location	${location}	Displays the location from which the alarm is generated.
Location String	${locationString}	Displays the location.
Log Aggregation Field	${logAggregationField}	Displays the log aggregation field.
Log Alarm Definition	${logAlarmDefinition}	Displays the log alarm definition.
Log Alarm Type	${logAlarmType}	Displays the log alarm type.
Log Event Count	${logEventCount}	Displays the log event count.
Log Tie Breaker ID	${logTieBreakerId}	Displays the log tie nreaker ID.
Log Type	${logType}	Displays the log type.
Maintenance	${maintenance}	Displays if the maintenance mode is on or off.
Management Module	${management_module}	Displays the management module in DX APM.
Message	${message}	Displays the time and date when the alarm was last updated.
Message Details	${messageDetails}	Displays the message details.
Metric Id	${met_id}	Displays the metric ID.
Metric External ID	${metric_external_id}	Displays the external ID of the metric.
Metric Family	${metric_family}	Displays the metric family.
Metric Group ID	${metric_group_id}	Displays the metric group ID.
Metric ID	${metric_id}	Displays the metric ID.

Text	Key	Description
Metric Name	${metric_name}	Displays the name of the metric that is involved in the alarm.
Metric Names	${metric_names}	Displays the names of the metric that are involved in the alarm.
Metric Type	${metric_type}	Displays the type of metric that is involved in the alarm.
Metric Unique ID	${metric_unique_id}	Displays the unique ID for the metric.
Metric Unique IDs	${metric_unique_ids}	Displays the unique IDs for the metrics.
Metric Unit	${metric_unit}	Displays the unit of the metric that is involved in the alarm.
Metric Value	${metric_value}	Displays the value of the metric that is involved in the alarm.
Metric View Link	${metric_view_link}	Displays the link to the Metric View.
Metrics Index	${metrics_index}	Displays the metrics index.
Model Name	${modelName}	Displays the name of the modeled device.
Model Type Handle	${modelTypeHandle}	Displays the model type flag (Visible, Instantiable, and Derivable, No Destroy, Unique, and Required).
NASS Projection ID	${nass_projection_id}	Displays the NASS projection ID.
Network Management Lost Impacted	${networkManagementLostImpacted}	Displays the network management lost impacted.
Network Service Probable Cause	${networkServiceProbableCause}	Displays the network service probable cause.
Network Services Impacted	${networkServicesImpacted}	Displays the network services impacted.
Next Hop	${nextHop}	
Notification Data	${notificationData}	Displays the data that is sent with the alarm notification.
Occurrence	${occurrence}	Displays the occurrence of the event.
OI NASS Attribute	${oiNassAttribute}	Displays the OI NASS attribute.
OI NASS Data Type	${oiNassDataType}	Displays the OI NASS data type.
OI NASS Source	${oiNassSource}	Displays the OI NASS source.
Origin	${origin}	Displays the UIM origin of the alarm.
Parent Device Type	${parentDeviceType}	Displays the type of the parent device.
PI Metric Display Name	${pi_metric_display_name}	Dispays the PI metric name.
Policy ID	${policyId}	Displays the policy ID.
Predicted Day	${predictedDay}	Displays the predicted day.
Predicted Value	${predictedValue}	Displays the predicted value.
Prediction Category	${predictionCategory}	Displays the predicted category.
Prediction Timestamp	${prediction_timestamp}	Displays timestamp of the prediction.
Probable Cause	${probableCause}	Displays the probable cause.
Probe	${probe}	Displays the probe name which is generating the alarm/ notification.
Product	${product}	Displays the product from which the alarm is generated. For example, *UIM*.

Text	Key	Description
Product Id	${product_id}	Displays the product ID.
Product Version	${product_version}	Displays the product version.
Protocol	${protocol}	Displays the protocol.
Raw Event Payload	${rawEventPayload}	Displays the raw event payload.
Robot	${robot}	Displays the UIM robot from which the alarm is generated.
Rollup Algorithm	${rollup_algorithm}	Displays the rollup algorithm.
Root Cause	${rootCause}	Displays the root cause.
SA Unique Id	${sa_unique_id}	Displays the unique ID of the service alarm.
Same Severity Suppression Data	${same_severity_suppression_data}	Displays the same suppression data.
Samples	${samples}	Displays the samples.
Server Port	${serverPort}	Displays the server port.
Service Alarm	${service_alarm}	Displays the service alarm.
Services Impacted	${services_impacted}	Displays the service which is impacted by an alarm.
Severity	${severity}	Indicates the severity of an alarm. The following severity levels are supported: • Critical • Major • Minor • Informational • Warning
Slope	${slope}	
Source	${source}	Displays the monitored device for which the alarm is flagged. For example, *testdevice1.example.net*
Spectro Server	${spectroSERVER}	Displays the Spectrum SpectroSERVER name.
Src Addr	${srcAddr}	Displays the source address.
Src CIDR	${srcCIDR}	Displays the source CIDR.
Src Port	${srcPort}	Displays the source port.
Start Time	${startTime}	Displays the start time.
Status	${status}	Displays the status.
Subsystem	${subsystem}	Displays the subsystem.
Subsystem ID	${subsystemID}	Displays the subsystem ID, identifying which part of the system the alarm relates to.
Summary	${summary}	Displays the summary.
Summary Alert	${summaryAlert}	Displays the summary alert.
Supp Key	${supp_key}	Displays the suppression key.
Symptoms	${symptoms}	Displays the symptoms.
Tags	${tags}	Displays the tag that is associated with the alarm/ notification.

Text	Key	Description
Tenant ID	${tenant_id}	Displays the tenant ID. Enter the tenant ID variable as ${tenant_id}.
Threshold	${threshold}	Displays the threshold.
Time Interval	${time_interval}	Displays the time interval.
Time to Threshold	${time_to_threshold}	Displays the event violation rule that sent an alarm when a QoS metric is predicted to reach a set value within a user-defined time period.
Timestamp	${timestamp}	Displays the timestamp.
Topology Model Name String	${topologyModelNameString}	Displays the name string of the topology model.
Total Points	${totalPoints}	Displays the total points.
Total Suppression Count	${total_suppression_count}	Displays the total suppression count.
Trend	${trend}	Displays the trend.
Troubleshooter Name	${troubleShooterName}	Displays the Spectrum Troubleshooter ID, associated with the alarm/ notification.
Trouble Ticket	${troubleTicket}	Displays the trouble ticket.
User Clearable	${userClearable}	Displays whether the alarm is user-clearable.
User Tag 1	${user_tag1}	Displays the custom user tag (specified in Spectrum) associated with the alarm.
User Tag 2	${user_tag2}	Displays the custom user tag (specified in Spectrum) associated with the alarm
Version	${version}	Displays the version.
Vertex Attributes	${vertex_attributes}	Displays the vertex attributes.
Vertex Id	${vertex_id}	Displays the vertex ID.
Vertex Type	${vertex_type}	Displays the vertex type.
Visible	${visible}	

DX Application Performance Management

The following table lists the template variables that provide data from DX Application Performance Management:

Text	Payload Key	Description
Agent	${agent}	Displays the name of the DX APM agent.
Agent Process	${agent_process}	Displays the agent process,
Alarm Description	${alarm_description}	Displays the alarm description.
Alarm Domain	${alarm_domain}	Displays the domain from which the alarm is generated.
Alarm ID	${alarm_id}	Displays the alarm ID.
Alarm Name	${alarm_name}	Displays the name of the alarm.
Alarm Type	${alarm_type}	Displays the alarm type.
Alarm URL	${alarmURL}	Displays the alarm URL.
Alarm Unique Id	${alarm_unique_id}	Displays the unique identification number of the alarm.

Text	Payload Key	Description
Alert Definition Link	${alert_definition_link}	Displays the alert definition link.
Alert External Id	${alert_external_id}	Displays the external ID for the alert.
Application Name	${applicationName}	Indicates the application name in DX APM.
Breached Threshold	${breached_threshold}	Displays the threshold that was breached.
CI Unique ID	${ci_unique_id}	Displays the unique ID of CI.
Caution Threshold	${caution_threshold}	Displays the caution threshold.
Channels	${channels}	Displays the channel.
Component Name	${component_name}	Displays the component name.
Custom_*n* *where n represents numbers 1-10.*	${custom_n}	You can specify up to 10 custom attributes.
Danger Threshold	${danger_threshold}	Displays the danger threshold.
Distribution Lists	${distributionLists}	
Doc Type ID	${doc_type_id}	Displays the ID of the doc type.
Doc Type Version	${doc_type_version}	Displays the version of the doc type.
External Ids	${external_ids}	Displays the external IDs.
Host	${host}	Displays the hostname.
Included Alerts	${included_alerts}	Displays the included alerts.
IP	${ip}	Displays the IP address.
Isolation View Link	${isolation_view_link}	Displays the link to the isolation view.
Management Module	${management_module}	Displays the management module in DX APM.
Message	${message}	Displays the time and date when the alarm was last updated.
Metric Id	${met_id}	Displays the metric ID.
Metric External ID	${metric_external_id}	Displays the external ID of the metric.
Metric Name	${metric_name}	Displays the name of the metric that is involved in the alarm.
Metric Value	${metric_value}	Displays the value of the metric that is involved in the alarm.
Metric View Link	${metric_view_link}	Displays the link to the Metric View.
OI NASS Attribute	${oiNassAttribute}	Displays the OI NASS attribute.
OI NASS Data Type	${oiNassDataType}	Displays the OI NASS data type.
OI NASS Source	${oiNassSource}	Displays the OI NASS source.
Product	${product}	Displays the product from which the alarm is generated. For example, *UIM*.
Product Id	${product_id}	Displays the product ID.
Product Version	${product_version}	Displays the product version.
Services Impacted	${services_impacted}	Displays the service which is impacted by an alarm.

Text	Payload Key	Description
Severity	${severity}	Indicates the severity of an alarm. The following severity levels are supported: • Critical • Major • Minor • Informational • Warning
Start Time	${startTime}	Displays the start time.
Status	${status}	Displays the status.
Summary Alert	${summaryAlert	Displays the summary alert.
Tenant ID	${tenant_id}	
Timestamp	${timestamp}	Displays the timestamp.
Vertex Attributes	${vertex_attributes}	Displays the vertex attributes.
Vertex Id	${vertex_id}	Displays the vertex ID.
Vertex Type	${vertex_type}	Displays the vertex type.

Spectrum

The following table lists the template variables that provide data from Spectrum:

Text	Payload Key	Description
Acknowledged	${acknowledged}	Indicates whether an alarm has been acknowledged.
Alarm Age	${alarmAge}	Displays the age of the alarm.
Alarm State	${alarmState}	Displays the state of the alarm.
Alarm URL	${alarmURL}	Displays the alarm URL.
Cause Code	${causeCode}	Displays the alarm cause code which is an 8-digit, hexadecimal code that identifies the probable cause of the alarm.
Cleared	${cleared}	Displays if cleared or not.
Closed Time	${closedTime}	Displays the closed time.
Collection Unique Key String	${collectionUniqueKeyString}	
Device Model Handle	${deviceModelHandle}	Displays the device model handle.
Device Type	${deviceType}	Displays the type of device that is involved in the alarm.
Device Type Spectrum	${deviceType_spectrum}	Displays the device type in Spectrum.
Doc Type ID	${doc_type_id}	Displays the id of the doc type.
Doc Type Version	${doc_type_version}	Displays the version of the doc type.
Event Message	${eventMessage}	Displays the event message.
Global Alarm ID	${globalAlarmID}	Displays the global ID of the alarm.
Global Alarm ID Router	${globalAlarmId}	Displays the global alarm ID.
Global Collection	${globalCollection}	Displays the global collection.
Group	${group}	Displays information about the group or groups to which the device belongs.

Text	Payload Key	Description
Group Id	${group_id}	Displays the group Ids to which the device belongs.
Host	${host}	Displays the hostname.
IP	${ip}	Displays the IP address.
Landscape ID	${landscapeID}	Displays the Spectrum landscape ID from which the alarm is generated.
Location	${location}	Displays the location from which the alarm is generated.
Location String	${locationString}	Displays the location.
Message	${message}	Displays the time and date when the alarm was last updated.
Message Details	${messageDetails}	Displays the message details.
Model Handle	${modelHandle}	Displays the model handle.
Model Name	${modelName}	Displays the name of the modeled device.
Model Type Handle	${modelTypeHandle}	Displays the model type flag (Visible, Instantiable, and Derivable, No Destroy, Unique, and Required).
Model Type Name	${modeltypeName}	Displays the model type name.
Network Management Lost Impacted	${networkManagementLostImpacted}	Displays the network management lost impacted.
Network Services Probable Cause	${networkServiceProbableCause}	Displays the probable cause for the network services.
Network Services Impacted	${networkServicesImpacted}	Displays the network services that are impacted.
Notification Data	${notificationData}	Displays the data that is sent with the alarm notification.
Origin	${origin}	Displays the UIM origin of the alarm.
Probable Cause	${probableCause}	Displays the probable cause.
Product	${product}	Displays the product from which the alarm is generated. For example, *UIM*.
Product ID	${product_id}	Displays the product ID.
Product Version	${product_version}	Displays the product version.
Repair Person	${repairPerson}	Displays the name of the repair person.
Root Cause	${rootCause}	Displays the root cause.
Services Impacted	${services_impacted}	Displays the service which is impacted by an alarm.
Severity	${severity}	Indicates the severity of an alarm. The following severity levels are supported: • Critical • Major • Minor • Informational • Warning
Spectro Server	${spectroSERVER}	Displays the Spectrum SpectroSERVER name.

Text	Payload Key	Description
Start Time	${startTime}	Displays the start time.
Status	${status}	Displays the status.
Symptoms	${symptoms}	Displays the symptoms.
Tags	${tags}	Displays the tag that is associated with the alarm/ notification.
Tenant ID	${tenant_id}	Displays the tenant ID.
Ticket ID	${ticketID}	Displays the ticket ID.
Timestamp	${timestamp}	Displays the timestamp.
Topology Model Name String	${topologyModelNameString}	Displays the name string of the topology model.
Troubleshooter Name	${troubleShooterName}	Displays the Spectrum Troubleshooter ID, associated with the alarm/ notification.
Trouble Ticket	${troubleTicket}	Displays the trouble ticket.
Trouble Ticket URL	${troubleTicketUrl}	Displays the trouble ticket URL.
User Clearable	${userClearable}	Displays whether the alarm is user-clearable.

UIM

Text	Payload Key	Description
Alarm Type	${alarmType}	Displays the alarm type. **Values:** Anomaly, Application, Fault
Alarm URL	${alarmURL}	Displays the alarm URL.
Alarm Update	${alarm_update}	Displays the alarm update.
Automic Jobs	${automicJobs}	
Baseline	${baseline}	Displays the baseline.
Breached Threshold	${breached_threshold}	Displays the threshold that was breached.
CI ID	${ci_id}	Displays the ID of CI (Configuration Item). A CI represents the component being monitored.
CI Name	${ci_name}	Displays the CI name.
CI Type	${ci_type}	Displays the CI type.
CI Unique ID	${ci_unique_id}	Displays the unique ID of CI.
CS ID	${cs_id}	Displays the CS (Computer System) ID.
CS Key	${cs_key}	Displays the CS key.
Cause Code	${causeCode}	Displays the alarm cause code, which is an 8-digit, hexadecimal code that identifies the probable cause of the alarm.
Caution Threshold	${caution_threshold}	Displays the caution threshold.
Channels	${channels}	Displays the channel.
Cleared	${cleared}	Displays if cleared or not.
Closed Time	${closedTime}	Displays the closed time.
Correlated External Id	${correlated_external_id}	Displays the correlated external ID.
Correlated External Ids	${correlated_external_ids}	Displays the correlated external IDs.
Correlation Names	${correlationNames}	Displays the correlation names.

Text	Payload Key	Description
CS ID	${cs_id}	Displays the CS ID.
CS Key	${cs_key}	Displays the CS key.
Custom_*n* *where n represents numbers 1-10.*	${custom_n}	You can specify up to 10 custom attributes.
Device Id	${dev_id}	Displays the unique identifier of the device that is involved in the alarm.
Device Type	${deviceType}	Displays the type of device that is involved in the alarm.
Doc Type ID	${doc_type_id}	Displays the ID of the doc type.
Doc Type Version	${doc_type_version}	Displays the version of the doc type.
Domain	${domain}	Displays the domain from which the alarm is generated.
Exists Metadata	${exists_metadata}	Displays if metadata exists.
External Ids	${external_ids}	Displays the external IDs.
Group	${group}	Displays information about the group or groups to which the device belongs.
Group Id	${group_id}	Displays the group Ids to which the device belongs.
Host	${host}	Displays the hostname.
Hub	${hub}	Displays the UIM hub from which the alarm is generated.
Ingestion Index	${ingestion_index}	Displays the ingestion index.
Initial Message	${initial_message}	Displays the initial message.
IP	${ip}	Displays the IP address.
Level	${level}	Displays the level.
Link	${link}	Displays the link.
Maintenance	${maintenance}	Displays if the maintenance mode is on or off.
Message	${message}	Displays the time and date when the alarm was last updated.
Metric Id	${met_id}	Displays the metric ID.
Metric Name	${metric_name}	Displays the name of the metric that is involved in the alarm.
Metric Type	${metric_type}	Displays the type of metric that is involved in the alarm.
Metric Unique ID	${metric_unique_id}	Displays the unique metric ID.
Metric Unit	${metric_unit}	Displays the unit of the metric that is involved in the alarm.
Metric Value	${metric_value}	Displays the value of the metric that is involved in the alarm.
NIM ID	${nimid}	Displays the NIM ID.
Occurrenece	${occurrence}	Displays the occurrence.
OI NASS Attribute	${oiNassAttribute}	Displays the OI NASS attribute.
OI NASS Data Type	${oiNassDataType}	Displays the OI NASS data type.
OI NASS Source	${oiNassSource}	Displays the OI NASS source.
Origin	${origin}	Displays the UIM origin of the alarm.
Probe	${probe}	Displays the probe name which is generating the alarm/ notification.
Product	${product}	Displays the product from which the alarm is generated. For example, *UIM*.
Product Id	${product_id}	Displays the product ID.

Text	Payload Key	Description
Product Version	${product_version}	Displays the product version.
Robot	${robot}	Displays the UIM robot from which the alarm is generated.
Root Cause	${rootCause}	Displays the root cause.
SA Unique Id	${sa_unique_id}	Displays the unique ID of the service alarm.
Service Alarm	${service_alarm}	Displays the service alarm.
Services Impacted	${services_impacted}	Displays the service which is impacted by an alarm.
Severity	${severity}	Indicates the severity of an alarm. The following severity levels are supported: • Critical • Major • Minor • Informational • Warning
Size	${size}	Displays the source port.
Source	${source}	Displays the monitored device for which the alarm is flagged. For example, *testdevice1.example.net*
Start Time	${startTime}	Displays the start time.
Status	${status}	Displays the status.
Subsystem	${subsystem}	Displays the subsystem.
Subsystem ID	${subsystemID}	Displays the subsystem ID, identifying which part of the system the alarm relates to.
Supp Key	${supp_key}	Displays the suppression key.
Symptoms	${symptoms}	Displays the symptoms.
Tags	${tags}	Displays the tag that is associated with the alarm/ notification.
Tenant ID	${tenant_id}	Displays the tenant ID.
Time Origin	${time_origin}	Displays the time origin.
Timestamp	${timestamp}	Displays the timestamp.
Troubleshooter Name	${troubleShooterName}	Displays the Spectrum Troubleshooter ID, associated with the alarm/ notification.
Trouble Ticket	${troubleTicket}	Displays the trouble ticket.
User Tag 1	${user_tag1}	Displays the custom user tag (specified in Spectrum) associated with the alarm.
User Tag 2	${user_tag2}	Displays the custom user tag (specified in Spectrum) associated with the alarm
Version	${version}	Displays the version.
Visible	${visible}	

Template Variables for Service Alarms

The following table lists all the template variables that are supported for Service Alarm:

NOTE

- The message template variables for Service Alarms are not available for new tenants.
- If a message template variable has no information to be displayed, then that variable value substitution is omitted from the notification message.

Text	Key	Description	Product
Acknowledged	${acknowledged}	Indicates whether an alarm has been acknowledged.	Spectrum
Alarm Type	${alarmType}	Displays the alarm type. **Values:** Anomaly, Application, Fault	DX APM, UIM, Spectrum, Third-party Products For UIM, Spectrum, and Third-party Products, DX OI populates this field.
Alarms	${alarms}	Displays the alarms.	DX OI
Annotation	${annotation}	Displays the annotation.	UIM
Closed Timestamp	${closedtimestamp}	Displays the closed timestamp.	
Health Message	${health_message}	Displays the health message.	
Last Update On Alarm Timestamp	${lastupdate_onalarm_timestamp}	Displays the last update on the alarm timestamp.	
Maintenance	${maintenance}	Displays if the maintenance mode is on or off. **Values:** true, false.	Spectrum
Message	${message}	Displays the time and date when the alarm was last updated.	All
Metric Name	${metric_name}	Displays the name of the metric that is involved in the alarm.	ADA, Custom, UIM
Network Rca	${network_rca}		
Root Cause	${rootCause}	Displays the root cause.	Spectrum
Root Cause Alarm URL	${root_cause_alarm_url}	Displays the alarm URL of the root cause.	
Root Cause Alarm Index	${rootCauseAlarmIndex}	Displays the alarm index of the root cause.	DX OI
Root Cause Host	${rootCauseHost}	Displays the host of the root cause.	DX OI
Root Cause Source	${rootCauseSource}	Displays the source of the root cause.	DX OI
Root Cause Update Timestamp	${rootcause_update_timestamp}	Displays the timestamp when the update was made to the root cause.	DX OI
SA Unique Id	${sa_unique_id}	Displays the unique ID of the service alarm.	DX OI
Service Alarm URL	${service_alarm_url}	Displays the service alarm URL.	DX OI
Service Id	${service_id}	Displays the service ID.	DX OI
Service Name	${service_name}	Displays the service name.	DX OI
Services Impacted	${services_impacted}	Displays the service which is impacted by an alarm.	DX OI

Text	Key	Description	Product
Severity	${severity}	Indicates the severity of an alarm. The following severity levels are supported: • Critical • Major • Minor • Informational • Warning	All
Start Timestamp	${starttimestamp}	Displays the start timestamp.	DX OI
Status	${status}	Displays the status.	All
Timestamp	${timestamp}	Displays the timestamp.	All
Troubleshooter Name	${troubleShooterName}	Displays the Spectrum Troubleshooter ID, associated with the alarm/ notification.	All
Trouble Ticket	${troubleTicket}	Displays the trouble ticket.	All
Trouble Ticket URL	${troubleTicketUrl}	Displays the URL for the trouble ticket.	
Version	${version}	Displays the version.	DX OI
Visible	${visible}		UIM

Template Variables for Situations

The following table lists all the template variables that are supported for situations:

NOTE
If a message template variable has no information to be displayed, then that variable value substitution is omitted from the notification message.

Text	Key	Description	Product
Acknowledged	${acknowledged}	Indicates whether an alarm has been acknowledged.	
Age (In min)	${age}	Displays the age of the alarm.	
Alarms Count	${alarmsCount}	Displays the alarm count.	
Alarm Type	${alarmType}	Displays the alarm type.	UIM (Now extended to all product alarms in DX OI to indicate the alarm type. Fault for Spectrum, Anomaly for anomaly alarms, Application Performance Management for APM alarms, Service for Service Alarms, and so on.)
Annotation	${annotation}	Displays the annotation.	
Closed Products	${closed_products}	Displays the closed products.	
Closure Ts	${closureTs}		
Cluster Id	${clusterId}	Displays the cluster ID.	
Cluster Previous Name	${clusterPrevName}	Displays the previous name of the cluster.	

Text	Key	Description	Product
Custom*n* *where n represents numbers 1 and 2.*	${customn}	You can specify up to 10 custom attributes.	
First Alarm Start Time	${firstAlarmStartTime}	Displays the start time of the first alarm.	
Hosts	${hosts}	Displays the name of the host.	
Initial Impacted Host	${initialImpactedHost}	Displays the host that was impacted first.	
Initial Impacted Service	${initialImpactedServices}	Displays the services that were impacted first.	
Initial Impacted Template	${initialImpactedTemplate}	Displays the template that was impacted first.	
Is Closed	${isClosed}	Displays if the alarm is closed.	
Is Force Closed	${isForceClosed}	Displays if the alarm was force closed.	
Is Orphan	${isOrphan}	Displays if the alarm is an orphan.	
Is Stable	${isStable}	Displays if the alarm is stable.	
Last Alarm Timestamp	${lastAlarmTimestamp}	Displays the timestamp of the last alarm.	
Maintenance	${maintenance}	Displays if the maintenance mode is on or off. **Values:** true, false.	
Most Impacted Host	${mostImpactedHost}	Displays the host that was most impacted.	
Most Impacted Service	${mostImpactedServices}	Displays the service that was most impacted.	
Most Impacted Template	${mostImpactedTemplate}	Displays the template that was most impacted.	
Name	${name}	Displays the name.	
Noise Flag	${noiseFlag}	Displays the noise flag.	
Primary Root Cause Host	${rc_host}	Displays the host of the primary root cause.	
Primary Root Cause Service	${rc_services_impacted}	Displays the primary root cause service that was impacted.	
Primary Root Cause Source	${rc_products}	Displays the source of the primary root cause.	
Products	${products}	Displays the product.	
RCA Scores	${rca_scores}	Displays the RCA scores.	DX OI
Root Cause Count	${rootCauseCount}	Displays the product from which the alarm is generated. For example, *UIM*.	
Root Cause Message	${rc_name}	Displays the message of the root cause.	
Root Cause Relative Confidence	${rc_confidence_score}	Displays the relative confidence of the root cause.	

Text	Key	Description	Product
Root Cause Score	${rc_score}	Displays the score of the root cause.	
Root Cause Severity	${rc_severity}	Displays the severity of the root cause.	
Root Cause Sub Cluster Id	${rc_subClusterId}	Displays the sub cluster ID.	
Services Impacted	${services_impacted}	Displays the service which is impacted by an alarm.	DX OI
Severity	${severity}	Indicates the severity of an alarm. The following severity levels are supported: • Critical • Major • Minor • Informational • Warning	All
Situation Source	${situationSource}	Displays the source of the situation.	
Situations URL	${situations_url}	Displays the situations URL.	
Stable Time	${stableTime}	Displays the stable time.	All
Start Time	${startTime}	Displays the start time.	All
Status	${status}	Displays the status of the alarm/ notification.	All
Sub Clusters Count	${subClustersCount}	Displays the count of the sub clusters.	
Ticket Field Modifiers	${field_modifiers}	Displays the ticket field modifiers.	
Ticket Field Updates	${field_updates}	Displays the ticket field updates.	
Timestamp	${timestamp}	Displays the timestamp of the alarm.	All
Troubleshooter Name	${troubleShooterName}	Displays the troubleshooters name.	All
Trouble Ticket	${troubleTicket}	Displays the trouble ticket.	All
Trouble Ticket URL	${troubleTicketUrl}	Displays the trouble ticket URL.	
Unique ID	${unique_id}	Displays the unique ID.	

Proxy Configuration

If your network configuration restricts outbound traffic, using proxy, you can route all traffic through a host that has more permissive outbound policies. DX Platform enables you to set up the proxy globally, which can be used for all outbound traffic (Webhooks and Slack).

For example, your DX OI environment must communicate with Slack but your network restricts outbound traffic. Configure the proxy globally and enable the proxy for each of your outbound notification services individually (for example, Slack).

Supported Scenarios

The proxy configuration is validated for the following scenarios:

Protocol	Authentication Type	Channel
HTTP	No Auth	Webhook, Slack, ITSM
HTTP	Basic Auth	Webhook, Slack, ITSM
HTTPS	No Auth	Webhook, Slack, ITSM
HTTPS	Basic Auth	Webhook, Slack, ITSM

Set Up Proxy Configuration

You must configure the proxy at the global level after which you can enable the proxy for each individual channel.

Follow these steps:

1. Log in to DX Platform as a Tenant Administrator.
2. Click **Settings** in the left navigation pane.
3. Click **Proxy Configuration** under **Environment definition and access**.
4. Provide the following details:
   - **Protocol:** Select the data transfer type (**http** or **https**) that you want to use.
   - **Host:** Enter the IP address or hostname for the proxy server.
   - **Port:** Enter the port for the proxy server.
   - **Authentication Type:** Select the authentication type. The following authentication types are supported:
     - **No Auth:** No authorization is required.
     - **Basic Auth:** Specify the username and password.
   - **Username:** Specify the proxy user name. This field is displayed when you select Basic Auth.
   - **Password:** Specify the password for the proxy user. This field is displayed when you select Basic Auth.
5. Click **Save**.

Your proxy details are saved successfully, and you can now enable the proxy from each individual channel.

> **NOTE**
> If you edit the existing proxy configuration, you must change the password to save the changes.

Using

This section contains information about the pre-defined Blueprints available in DX Operational Intelligence. You can also understand how you could leverage the different analytical capabilities of DX Operational Intelligence:

- Service Analytics
- Alarm Analytics
- Insights
- Performance Analytics
- Capacity Analytics
- Predictive Insights
- Monitored Inventory
- DX OI - Logs
- DX Dashboards

Service Analytics

This section provides an overview of Service Analytics. Learn how an Operation Manager monitors the service health and status of service, view the impacted services, and the infrastructure and applications that are mapped to a service.

Domain management solutions monitor various aspects of a business service, including support for IT infrastructure components or the end-user experience. None of these individual solutions give you a complete, end-to-end view of service health and availability across all management domains. Operations personnel often guess how the fault or performance issues reported across the network, systems, database, or application monitoring tools actually affect key IT services, degrade service quality, or increase the risk of an outage. Similarly, service stakeholders may not understand whether IT enables them to fulfill their business objectives.

Service Analytics helps overcome these challenges by unifying the health and availability information from your domain management tools and aligning with your IT services.**Service Analytics** is a capability that provides an overview of services. You can use DX Operational Intelligence features by using the following Service Analytics components:

- **Correlation Engine** for DX Operational Intelligence inventory deduplication.
- **Services Impact** automatically correlated to underlying infrastructure for faster problem resolution.
- **Dynamic Automated Service Discovery** with attribute and graph-based traversal for creating a service.
- **Unified Topology** for the entire digital chain; physical, virtual, flow, and logical network topology for service discovery.
- **Service Management** for monitoring service health.
- **Service Alarms** for managing incoming alarms that are derived from services.
- **Flexible Service Hierarchy** to model IT services in the context of business and organization.
- Incrementally improve **Service Health**, prevent service outages, and deliver on customer experiences with complete visibility.
- **Service Personalization** enables you to build customized views and have them saved across login and support service level filtering
- **Service Level Indicators (SLI)** enable you to pick a NASS metric to visualize on the Services view.
- **Scalability Enhancements** to handle large Enterprise Service Hierarchy (20k+ services).

As an Operations Manager, you can import services and can view the following information of a service that is based on the CIs and docker instance.

- Status of service (availability and risk)
- Infrastructure and applications that are mapped to a service
- Impacted services

The following steps help the Operations Manager to create a service, determine the state of the Key Performance Indicators (KPIs) for an existing service, and identify any variations in KPI behavior of a service.

- Services User Interface
- Service Level Indicator and Service Level Objective
- Service Personalization

Access Service Analytics

You can access Service Analytics from the DX operational Intelligence landing page. You can also set the Services View as your default landing page. You can also access Service Analytics from the Alarm Analytics page.

You can access the Service Analytics view by using the following steps:

1. Log in to DX Operational Intelligence.
2. From the **DX Operational Intelligence** landing page, click

capability on the left navigation pane.
   The **Services** view appears. You can view detailed information about all the services.
3. To make the **Services** view as your default landing page in DX Operational Intelligence, click icon on the top-right, **Set as OI landing page**.

   NOTE

   - By default, the DX Operational Intelligence homepage displays only the icons of each capability. When you hover over any capability on the left navigation pane, the menu bar expands to show all the icons with the capability name. Use **Always show menu** toggle option to display all the capabilities.
   - You can also access **Service Analytics** from Alarm Analytics.

Services User Interface

The Service Analytics allows you to view the list of services and their related information on the following Services user interface:

- Service Overview Page
- Service Details Page

Service Overview Page

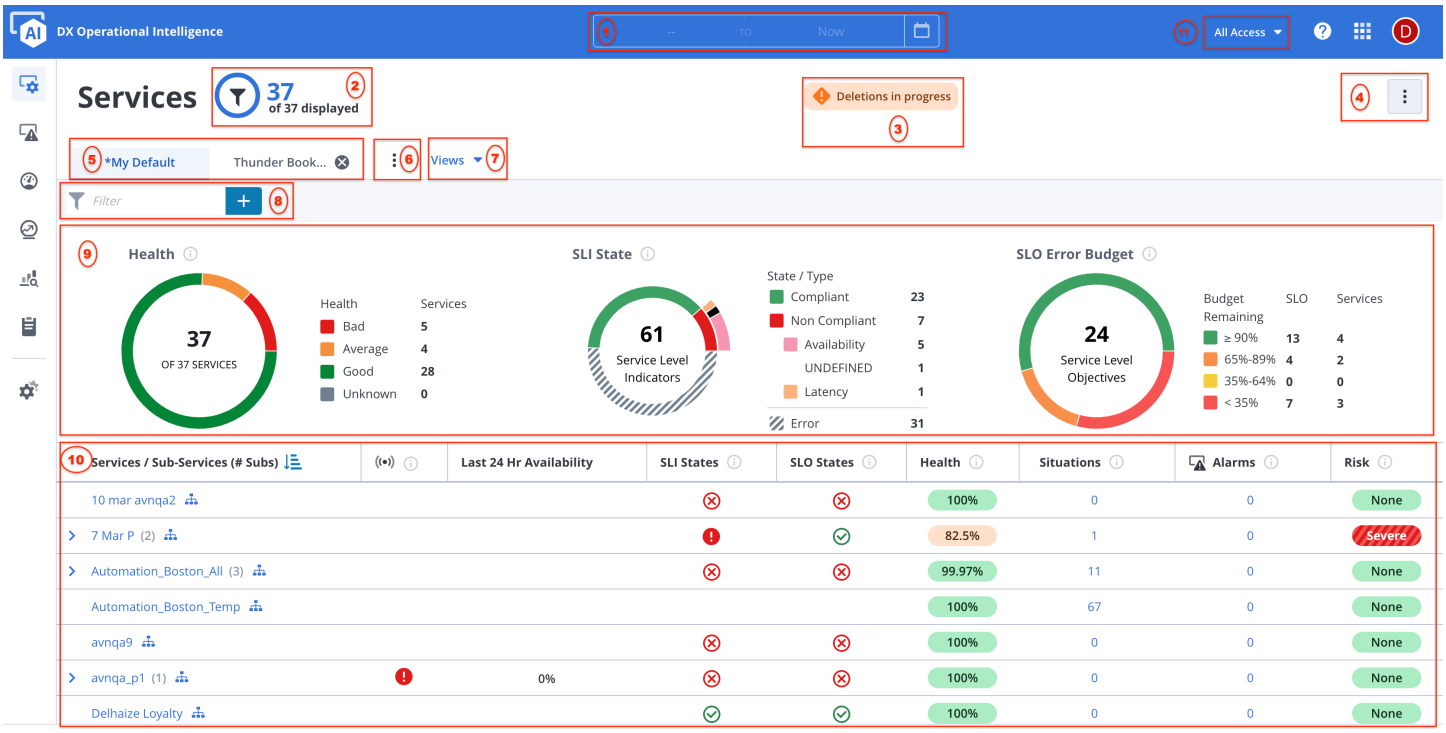
The Service Overview provides a snapshot of the business services. You view these options on this page: Service filters, Pin and Unpin filters, KPIs.

This page also gives you insights on the services that need immediate attention, and the services are at high risk of failure. You can view the total number of active services and critical services. You can further drill down to each service by clicking a service that redirects you to the Service Details view. This view displays the details of a service such as each KPI information, an option to delete a service, and an option to enable and disable the KPIs.

On the Services Overview, the metrics provide insights into all the services with color code indicating the severity of the service.

NOTE
This page displays only the current and live metrics data. The data/time picker on this page has been disabled to avoid any discrepancies in the alarm lifecycle time, that is, the creation time of the earliest alarm from all the active alarms and alarm closing time. For a historical analysis, you must go to the Service Details page or Alarm Analytics view.

The following image explains the options available on the Service Overview page:



Filter by Date/ Time Range (1)

The **Date/Time** filter is **disabled** in the **Services** view to avoid any discrepancies in the alarm lifecycle time, that is, the creation time of the earliest alarm from all the active alarms and alarm closing time.

Services Count (2)

Displays the count of filtered services from the total count of services.

Deletions in progress (3)

Click the Deletions in progress icon to view the services that are in the process of deletion.

Service View Options (4)

Use this menu to perform the following tasks:

- Refresh View
- Auto-update view
- Group Services
- Show KPIs
- Select KPIs To Display
- Customize Columns
- Create a Service
- Manage Service Templates
- Set as OI Landing Page

NOTE
For more information, see Service View Options.

Saved Services View (5)

You can save a Services view as a new tab based on your customization.

NOTE
The default view**All Services** is saved. You can customize this default view and save it as a new view.

Manage Views

( ) (6)

Use this option to manage a view. Click to perform the following actions:

	• **Save:** Saves the existing view. • **Save as..:** Saves the view as a new tab. • Reset View: Revert the changes to the last saved state. • Delete: Deletes the view tab permanently. **NOTE** For the **Default** tab, the **Delete** option resets the tab.
Views (7)	Use this option to view the list of pinned views and all views. You can also use the filter option to search for a view.
Service Filter (8)	You can use filter attributes to narrow down your services. Service Analytics is enhanced to support the filtering of services using generic preset attributes and custom attributes. Therefore, if you have any services that contain any custom metrics, you can filter them using the **Custom properties filter** and you can filter services using the generic preset attributes.
KPI Charts(9)	The KPI Charts section is configurable and can display up to six KPI charts. By default, Health, Risk, and Worst 5 services charts are displayed. You can also add SLI Error Budget, SLI State, and Availability charts. **NOTE** For more information, see View KPIs Charts.
Services Overview Panel (10)	Displays the list of services, sub-services, and their related information.
All Access (11)	By default, every user is provisioned with a universe named **All Access** that enables them to view all the data. This dropdown displays all the universes that a user has access to in addition to the All Access universe. You can select the universe for which you want to view the data. For more information, see Universes.

Service Analytics helps overcome the fault and performance challenges by unifying the health and availability information from your domain management tools and aligning it with your IT services. Service Analytics is a capability that provides an overview of services. You can import services and can view the status of a service, infrastructure, and applications that are mapped to service, and impacted services that are based on the CIs and docker instance.

{"URL":["https://digital-oi/service-analytics"],"description":"concept.dita_df56cdeb-c67b-4ad1-8e0c-a401bac6d7c7","troubleshooting":{"masterkb":"https://knowledge.broadcom.com/external/article?articleId=230635"},"video":["https://www.youtube.com/watch?v=dvPoCnvAWoE"],"customCards":[{"id":"task.dita_4f941ed7-51bd-44cc-8ac3-3aa121e436fd","type":"customize","url":"https://techdocs.broadcom.com/us/en/ca-enterprise-software/it-operations-management/dx-operational-intelligence-saas/SaaS/using/Service-Analytics/Services-User-Interface/Service-Overview-Page/Customize-Columns.html","title":"Customize KPIs Columns"},{"id":"concept.dita_627a5697-fc8b-436d-80f5-70b5266648f2","type":"use","url":"https://techdocs.broadcom.com/us/en/ca-enterprise-software/it-operations-management/dx-operational-intelligence-saas/SaaS/using/Service-Analytics/Services-User-Interface/Service-Overview-Page/Service-View-Options.html","title":"Service View Options"}]}

Service View Options

The **Services** view enables you to perform the following tasks by clicking

- **Refresh View:** This option enables you to reload the latest data.
- **Auto-update View**: By default, **Auto-update view** switch is enabled. Enable this option to refresh the table automatically.
- **Group Services:** This option enables the hierarchical service model, by grouping all the sub-services under their respective top-level services.
- **Show KPIs:** Enable this option to display the KPIs chart on the **Services Overview** page.
- **Select KPIs To Display:** Use this option to customize the KPIs chart to be displayed on the Services Overview page. Click the toggle switch to enable or disable a KPI.

  Values: Health, Availability, Risk, Worst 5, SLI State, and SLO Error Budget
- **Customize columns:** Enables you to select and deselect the columns to be displayed.

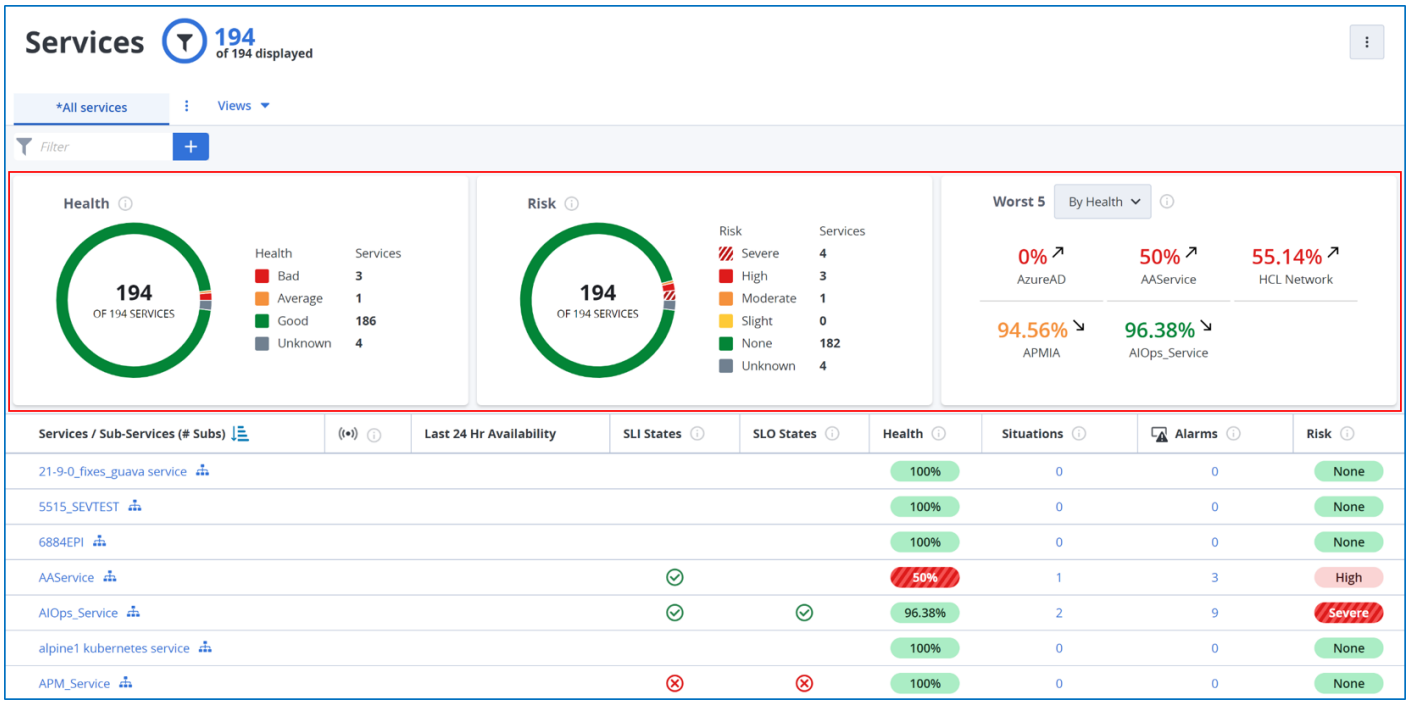
  Values: Availability, Last 24 Hr Availability, SLI States, SLO States, Location, Tags, Health, Situations, Alarms, and Risk
- **Create a Service:** Enables you to create a service using various data sources. For more information, see Create a Service icon.
- **Manage Service Templates:** Click to navigate to the service creation page.
- **Set as OI Landing Page:** Click to save the current page as the landing page.

View KPIs Chart

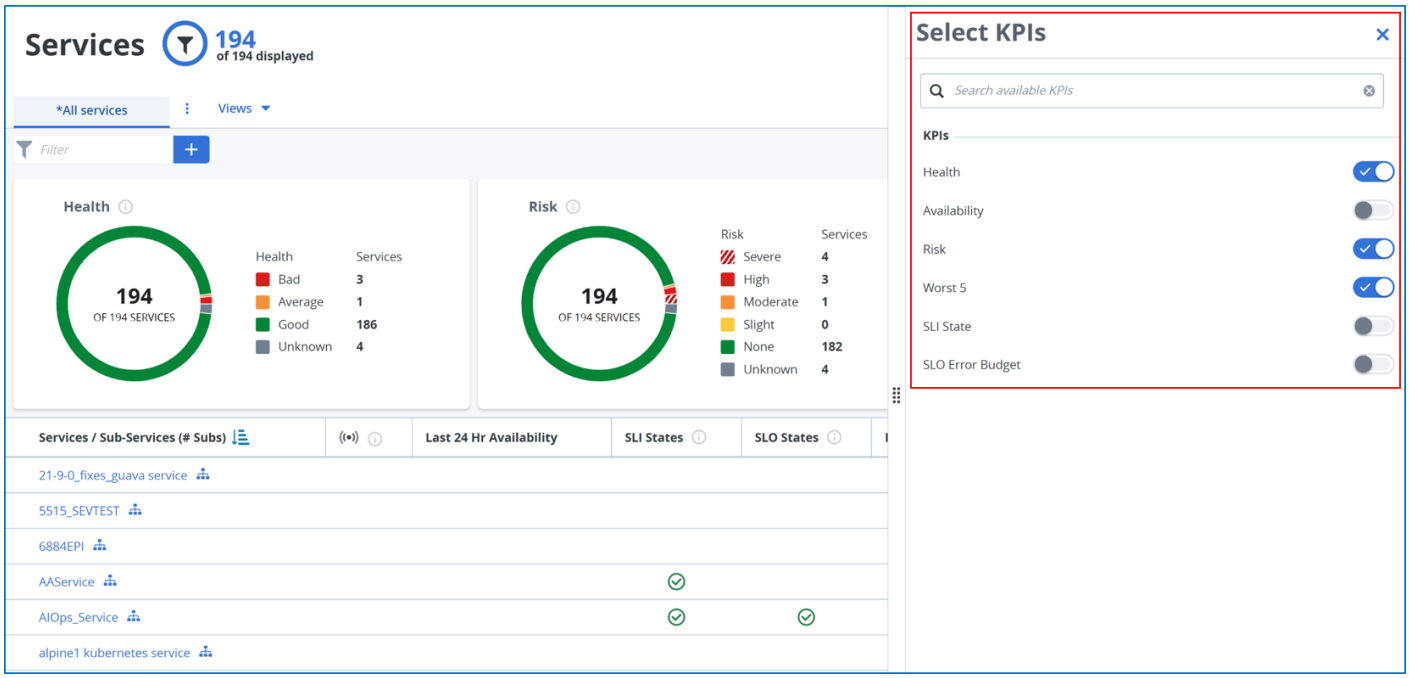
You can customize the KPIs chart to display on the Services Overview page using the following options:

- **Show KPIs:** Enable this option to display the KPIs on the Services Overview page. When you enable this option, the **Select KPIs to display** option is also enabled.

  The following image illustrates the Services page with the **Show KPIs** option enabled:



- **Select KPIs to display:** Click this option to display the available KPIs to select from.



The following steps describe how to select the KPIs to be displayed on the Services Overview page.

> **NOTE**
> By default, KPI charts for **Health**, **Risk**, and **Worst 5** appear on the **Services Overview** page when the **Show KPIs** option is enabled.

Follow these steps:

1. From the Services Overview page, click

   ,

   Select KPIs to Display.
2. Enable the required KPIs:
   - **Health:** The Health Pie Chart indicates the current health of all the services. This graph indicates the percentage of the service components that are in a functional state considering the impact on the overall service. The services are categorized based on their severity status as Bad, Average, Good, and Unknown on the Services view.
   - **Availability:** The Availability Pie Chart indicates the percentage of the service and up and operational. The services are categorized based on their severity status as Down, Good, Maintenance, and Unknown on the Services view. For example, if Payroll service risk is **severe**, then the probability of the service not being available is **high**.

     > **NOTE**
     > The **Services** view displays the Availability pie chart depending on whether the service is associated with any App Synthetic Monitor (ASM).
   - **Risk:** Risk indicates the probability of the service going down. Initially, the services that are created appear as Unknown until the data is processed.
   - **Worst 5:** The Worst 5 section provides detail of the bottom five business services that are based on the selected KPI. By default, you can view details of Health, followed by Risk, and Availability. The worst 5 section displays live data.
   - **SLI State:** Indicates the SLI state of services based on Type and State. States such as Complaint, Non-Complaint, and Error.

NOTE
The "Unknown" legend is not present for the SLO States pie chart.

– **SLO Error Budget**

NOTE
The SLI State and SLO Error Budget KPIs appear only for the tenants with SLI and SLO enabled.

Customize KPIs Columns

The Services view enables you to customize the list of KPIs columns, such as Health KPI, Availability KPI, Risk KPI.

To customize the number of columns/KPIs, perform the following steps:

1. Click

   , **Customize Columns** on the **Services View** page.

   The list of columns is displayed. **Values:** Availability, Last 24 Hr Availability, SLI States, SLO States, Location, Tags, Health, Situations, Alarms, and Risk

2. Select the required KPIs from the list.

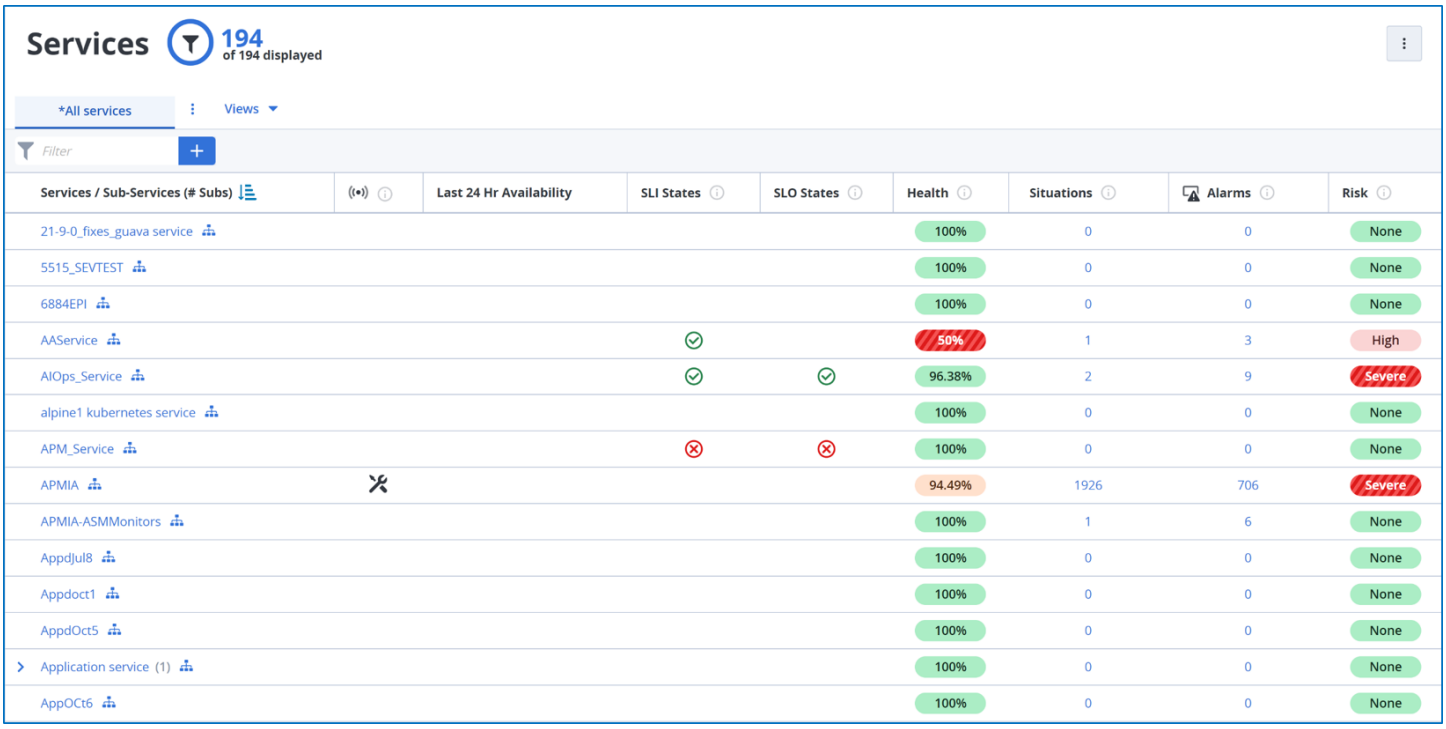
   The **Services** view displays the columns that you selected.

   The key attributes of the services are shown with colors and symbols indicating the status of each attribute.

   NOTE

   - The Risk KPI, Availability KPI, Health KPI, and all the metrics get published every 1 minute.
   - The SLI States KPI and SLO States KPI appear only for the tenants with SLI and SLO enabled.
   - In the **Services** view, all the columns support sorting of values except for **Alarms** column. You can sort the values in ascending and descending order as required

Key Performance Indicators (KPIs) for Services
The Key Performance Indicators for services give you an overview of the health of your service. DX Operational Intelligence provides the following KPIs for services.



Service/Sub-services: Indicates the name of the services or sub-services. To view the complete list of services, you need to keep scrolling down.

Availability: Availability is represented as

.

Indicates the health and current availability status of the service, and it displays the percentage of time the service is up and operational. The Availability status depends on the state of monitoring associated with a service. Availability is calculated even when the service is under maintenance.

Service Status	Monitor State	Availability	Last 24 hr availability	Availability Status (Service Details page)
Service is Active	Up	Good	100%	100%
	Down	Down	Real	0
	Maintenance	Maintenance	Unknown	Metric Unavailable
	Not attached	Unknown	Unknown	Metric Unavailable
	Inactive	Unknown	Unknown	Metric Unavailable
Service in Maintenance Window	Up	Good	100%	100%
	Down	Down	Real	0
	Maintenance	Maintenance	Unknown	Metric Unavailable
	Not attached	Unknown	Unknown	Metric Unavailable
	Inactive	Unknown	Unknown	Metric Unavailable

Last 24 hr Availability: Indicates the percentage of time the service was 100 percent available in the last 24 hours. The 24 hr availability refers to the average of 24 hours. Availability is based on the health of the service.

SLI States: Indicates the cumulative (rolled up) status of SLI associated with the service. You can click the icon to view the details of the individual SLI. You can perform the following actions on the SLI popup:

- Click the SLI name to view the Performance in the context of the service and SLI.
- Click **View metrics** to view the service and all its SLIs on the Performance Analytics page.
- Click **Related Alarms** to view the alarms in the context of the service.
- Click **View Service** to view the service details in the context of the service and SLI.

SLO States: Indicates the cumulative (rolled up) status of SLO associated with the service. You can click the icon to view the details of the individual SLI. You can perform the following actions on the SLO popup:

- Click the SLI name to view the Performance in the context of the service and SLO.
- Click **View metrics** to view the service and all its SLOs on the Performance Analytics page.
- Click **Related Alarms** to view the alarms in the context of the service.
- Click **View Service** to view the service details in the context of the service and SLO.

Location: Indicates the location defined for the service. For example, Australia.

Tags: Indicates the tag name defined for the service. Click the tags count to view all tags associated with the service.

Health: Indicates the percentage of devices that are running normally and operational within the service. The health of service is calculated based on the number of available CIs with the total number of CIs. Health of a service is determined as follows:

- Good: 96% - 100%
- Average: 76% - 95%
- Bad: 0% - 75%

> **NOTE**
> When you are creating a service for the first time, you need to wait up to 1 minute to view the details in the **Health** and **Risk** columns respectively, until then it might appear as empty or **Unknown**.

- **Health of Sub-service:** The health of service is calculated based on the weights assigned per service and individual health.

Situation Alarms (+ subs): Indicates the total number of open situations associated with the service including open situations for its child services. Selecting the value redirects to the situations page in the context of the service.

Alarms (+ subs) Indicates the alarm count, that is, the total number of Raw alarms and Anomaly alarms that are mapped to a service within the selected time range. In the alarm count, you can view the rolled-up alarm count of its sub-services. The **Alarms (+ subs)** column contains the parent alarm and child alarm count consolidated together. Clicking on the rolled-up alarm count redirects you to the **Alarm Analytics** page. For more information about Alarms, see Alarms Analytics.

> **NOTE**
> The **Alarms (+ subs)** column displays only the currently active alarms count and does not include closed alarms or prediction alarms in the alarm count.

For example, let us assume that C2 and C3 get data from CA UIM containing 20 raw alarms and 10 anomaly alarms. The alarm volume for the Mobile Banking service is 30.

Risk: Indicates the probability of the service going down. Risk is defined as the risk of service becoming unavailable in the near future unless any action is taken on the alarms and configuration items associated with that service. The risk depends on the significance of CI and the number of components on which the alarms get generated. The risk severity value ranges from severe to normal. Risk is determined as follows:

- None: 0
- Slight: 1
- Moderate: 2
- High: 3
- Severe: 4

Create a Service

Create a service by defining multi-hierarchy services using service discovery and topologies across different monitoring domains like application, infrastructure, and network.

{"URL":["http://digital-oi/service-analytics/serviceEdition/serviceAdd"],"description":"concept.dita_8106d4f8-8d5b-42ab-af72-0adc76a57c08","new":"","new_video":"","admin":"","troubleshooting":{"masterkb":"","text":"","URL":[]},"pendo":"","video":[]}

Service creation primarily consists of the following steps:

- Add Service: Add Service allows you to create a new service or create a service from the existing shared service. You can also set the service properties and create a service definition.
- Configure Service Details Properties: The Service Details panel allows you to configure the service details such as service name, tags, location, service definition filters, custom properties, service topology, KPIs, and service elements.
- Create Service Definition: Search your application from the available list of applications. Search for attributes from TAS (graph database) that helps in querying all entities that should be included in the service.
- Service Configuration: Dynamic Automated Service Discovery with attribute and graph-based traversal for creating a service.
- View Composition or Topology of the Application: Unified Topology for the entire digital chain; physical, virtual, flow, and logical network topology for the service discovery.

Configure DX App Synthetic Monitor

Before you create a service or start monitoring the service, you must configure the App Synthetic Monitor (ASM).

Follow these steps:

1. Log in to **App Synthetic Monitor**.
2. Click **Monitoring**, **Monitors**.
3. To create a monitor, click **New Monitor**.
4. In the **General** tab, enter the required parameters such as Name, URL, Time-out period.
5. In the **Locations** tab, select the **Monitor order algorithm** and enable the **Default** checkbox.
6. In the **Check Period** tab, you can schedule the date, duration, and time-period for monitoring.
7. Click **Save**.

The App Synthetic Monitor tracks the details and sends them to the OI metric store. For more, information, see App Synthetic Monitor.

Add New Service

The Add New Service allows you to create a new service or create a service from the existing shared service. You can set the service properties and create a service definition.

{"URL":["http://digital-oi/service-analytics/serviceEdition/serviceAdd"],"customLabelGetStarted":"Add Service","description":"task.dita_1308c1dc-a9d1-4a58-ae11-fdcd300ce8c6","customCards":[{"id":"task.dita_00419ce8-120c-43ff-be4d-1c92ad66004e","type":"configure","url":"https://techdocs.broadcom.com/us/en/ca-enterprise-software/it-operations-management/dx-operational-intelligence-saas/SaaS/using/Service-Analytics/Services-User-Interface/Service-Overview-Page/Create-a-Service.html","title":"Configure Service Details Properties"}]}

You can also create the service layout and its hierarchy without entities.

Follow these steps:

1. From the Services page, click

   ,

   Create Service.

   The Add Service page appears.

2. Click **Add New Service** or **Add Shared Service** based on your requirement.
   a) **Add New Service:** This option lets you add an empty group service to the layout. You can add as many services to build the service definition.
      1. Add the service details, in the **Service Details** panel.
      2. Create the service definition.
   b) **Add Shared Service:** This option lets you add an existing service and its children to the service you are creating or editing.
      1. Click the **Add Shared Service** button, select the service from the list of existing services, and edit the service details.
      2. Edit the service definition.

The service is created.

> **NOTE**
> When you create a service, the underlying topology is tagged with the service name at the same time of service creation. But, when you add a new device to the service, it takes 1 minute for the device to be associated with the service name.

Configure Service Details

The Service Details panel allows you to configure the service details such as service name, tags, location, service definition filters, custom properties, service topology, KPIs, and service elements.

Follow these steps:

1. Click the required service on the **Add Service** page.

   The Service Details panel appears.

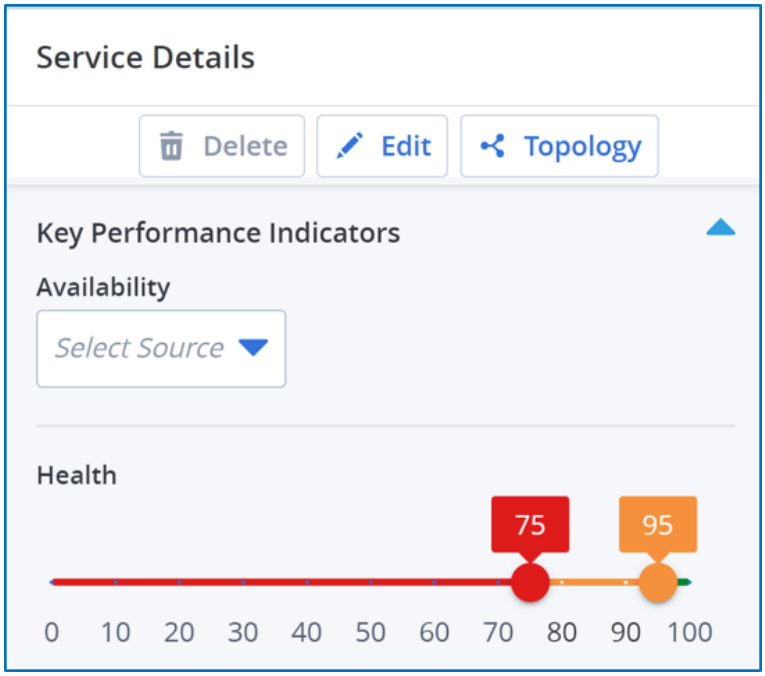
2. Enter the following properties details for a service:
   a) **Service Name:** Provide a meaningful unique service name.
   b) **Description:** Provide a description of a service.
   c) **Tags:** Specify tags to identify a service.
   d) **Key Performance Indicators:**
      - **Availability:** Enables you to select the metrics. click

        icon and select the metrics. The default source is ASM. You can also use the filters to drill down the metrics. If you want to clear the availability selection that you made, use the **Clear** option. For more information about ASM integration, see Integrate DX App Synthetic Monitor.
      - **Health:** Health indicates the percentage of devices that are running normally within the service and is calculated based on the number of available CIs with the total number of CIs.

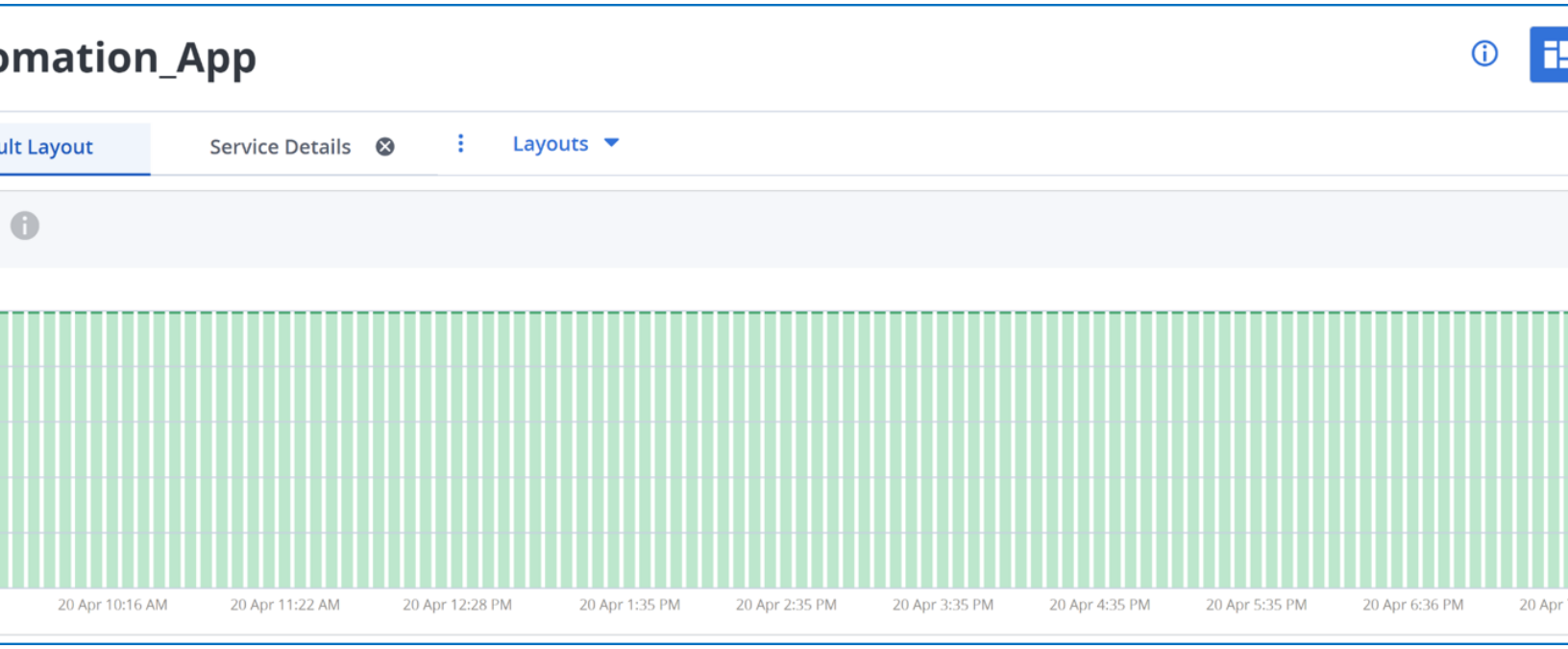
By default, the following threshold values determine the service health.

Health Category	Service Health
GOOD	95% - 100%
AVERAGE	> = 75% - < = 95%
BAD	0% - 75%

You can configure these values using the slider.



When you change these values, the **Health** widget on the **Service Details** page is updated accordingly. For example, chhange the values on the slider to 80 and 90. The Health widget displays the service health based on the new values as shown in the following image:

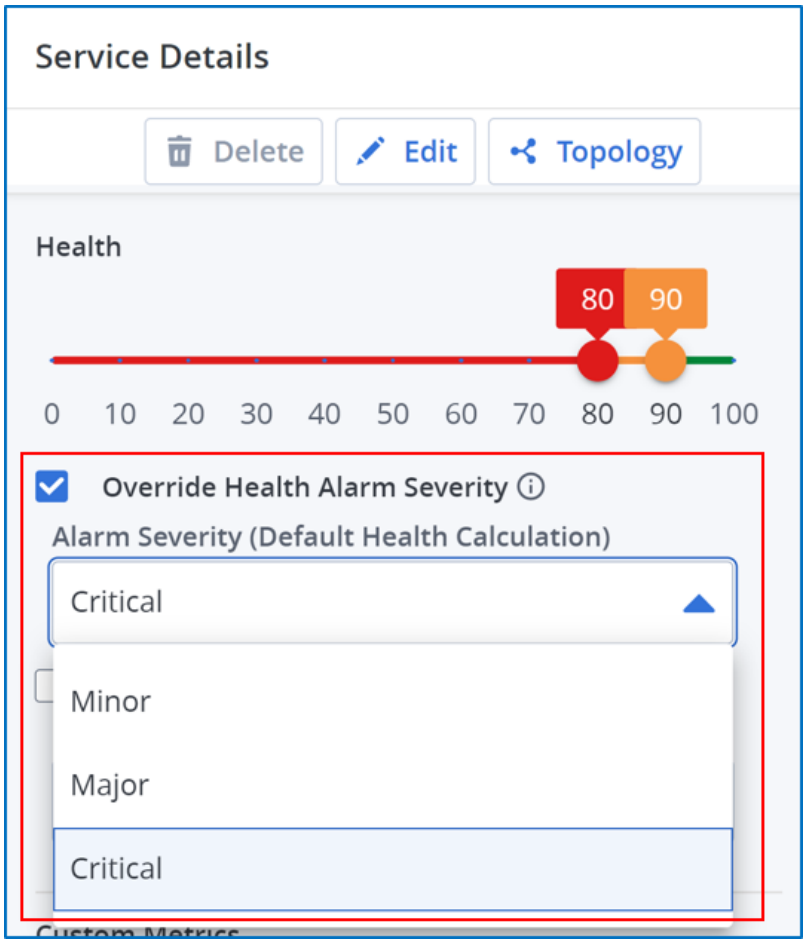


According to the reconfigured values, the service health is:

- **Good** if the percentage of devices that are running is 90% - 100%.
- **Average** if the percentage of devices that are running is > 80% but <= 90%.
- **Bad** if the percentage of devices that are running is 0% - 80%.

- **Override Health Alarm Severity:** Select this checkbox and then select the severity from the **Alarm Severity** drop-down list to override the default service health calculation which is based on critical alarms only.

  The service health is calculated based on the devices or entities that are down. By default, a device or entity is counted as down only if critical alarms are triggered against it. The major and minor alarms are not considered in this calculation. For example, if a service has 10 devices with only three devices having critical alarms, the service health is calculated based on the devices that are down. If the devices in the service do not have critical alarms, then the devices are counted as up and the service health is displayed as 100%. Using this option, you can configure the service health based on all the three alarms, critical, major, and minor and not just the critical alarms.



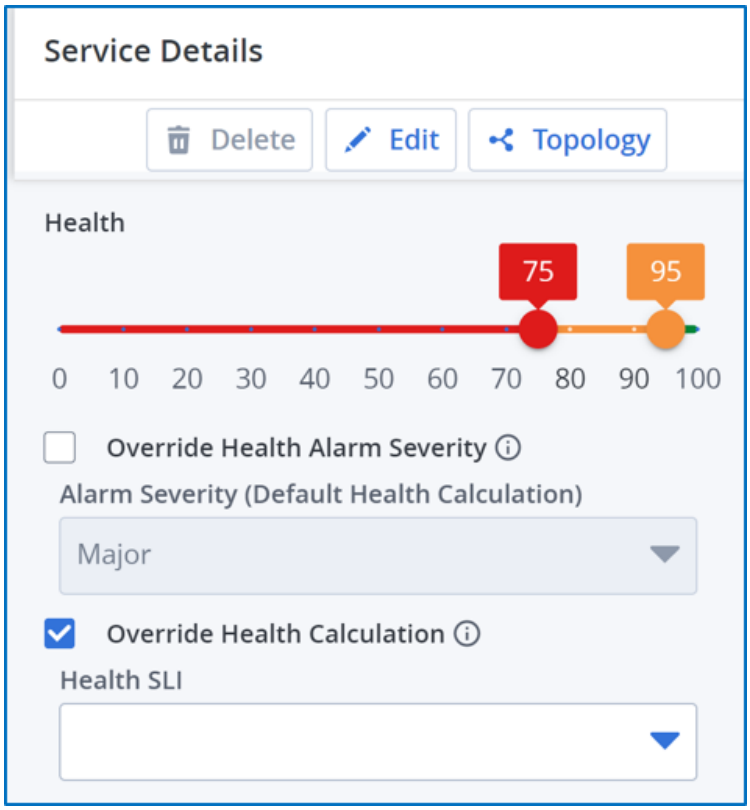
If you select:

- **Critical:** Only Critical alarms are considered for the service health calculation.
- **Major:** Major and Critical alarms are considered for the service health calculation.
- **Minor:** Minor, Major, and Critical alarms are considered for the service health calculation.

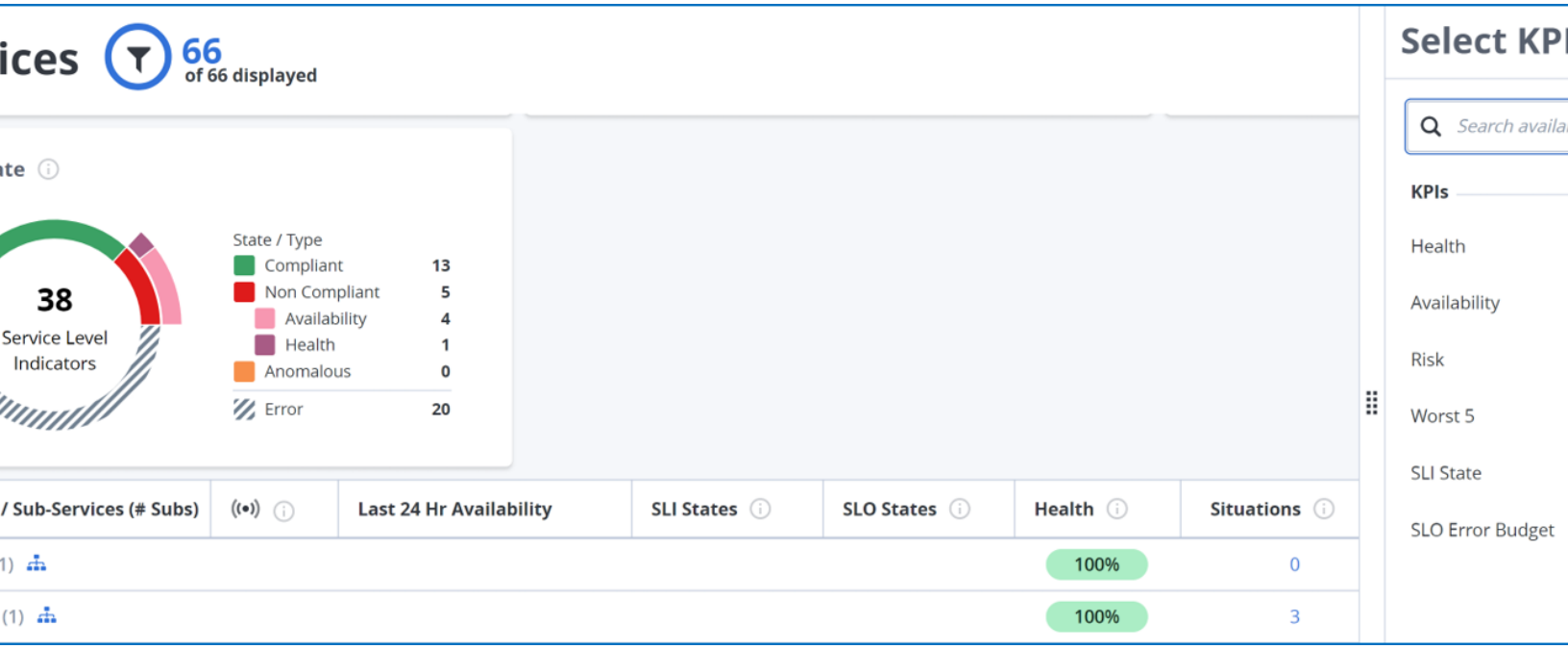
After this configuration, the **Health** widget on the **Service Details** page reflects the health based on the configuration.

- Only critical alarms are considered for service health calculation if you do not select this checkbox.
- This configuration is specific to the service. If you select this option for a parent service, then this configuration is used in the service health calculation only for the parent service and not for the child services. The service health for the child services is calculated based on the default condition, which is the critical alarms only.
- If the health SLI is selected, the default health metric is calculated with this alarm severity. For more information, see the **Override Health Calculation** section.

- **Override Health Calculation:** Select the **Override Health Calculation** checkbox and select the Health SLI. The service health is calculated based on the number of available devices and impacted devices. To calculate the service health based on metrics, you can create an SLI using the metrics. While creating the SLI, ensure to select the SLI Type as Health.



This selection overrides the metric that is used to calculate the service health with the Health SLI. The Health widget on the Service Details page displays the service health based on the Health SLI. If the **Health SLI** is created, the **SLI State** widget on the **Services** page displays Health.



- **Custom Metrics:** Allows you to select the metrics. You can also provide the custom name for the selected metric.

e) **Context for Situations:** By default, only individual services are considered for situation alarm clustering. To include all child services as a part of the Service topology select the **Include Child Services** checkbox.
f) **Custom Properties:** Enables you to assign location and custom properties. The custom properties are a key-value pair for a service and you can add up to twenty custom properties.
g) **Location:** Provide the location of a service.
h) **Selected Definition Filters:** Indicates the filters that are applied to the selected elements. This option is available only for the group service elements.

> **NOTE**
> You can also click the Topology icon and Filter icon on the **Service Details** pane to view the topology and filter information for the selected service.

3. **Manage Service Elements:** Click the filter icon to update the service definition filters. The application name filters are added under the Selected definition filters option.
4. **Topology Layout:** Click the topology icon to view the details of the topology. For more information, see View Service Topology.

Create Service Definition

Create a service definition to monitor your application by selecting a source and the corresponding base attributes on the Service Definition page.

{"URL":["https://digital-oi/service-analytics/serviceEdition/serviceAdd/filter"],"description":"task.dita_08c5920b-cfcd-4f08-a625-3f4a78953d23","new":"","new_video":"","admin":"","troubleshooting":{"masterkb":"","text":"","URL":[]},"pendo":"","video":[]}

After you have selected the **Add Service** or **Add Shared Service** option on the Add Service page, you can create a service definition.

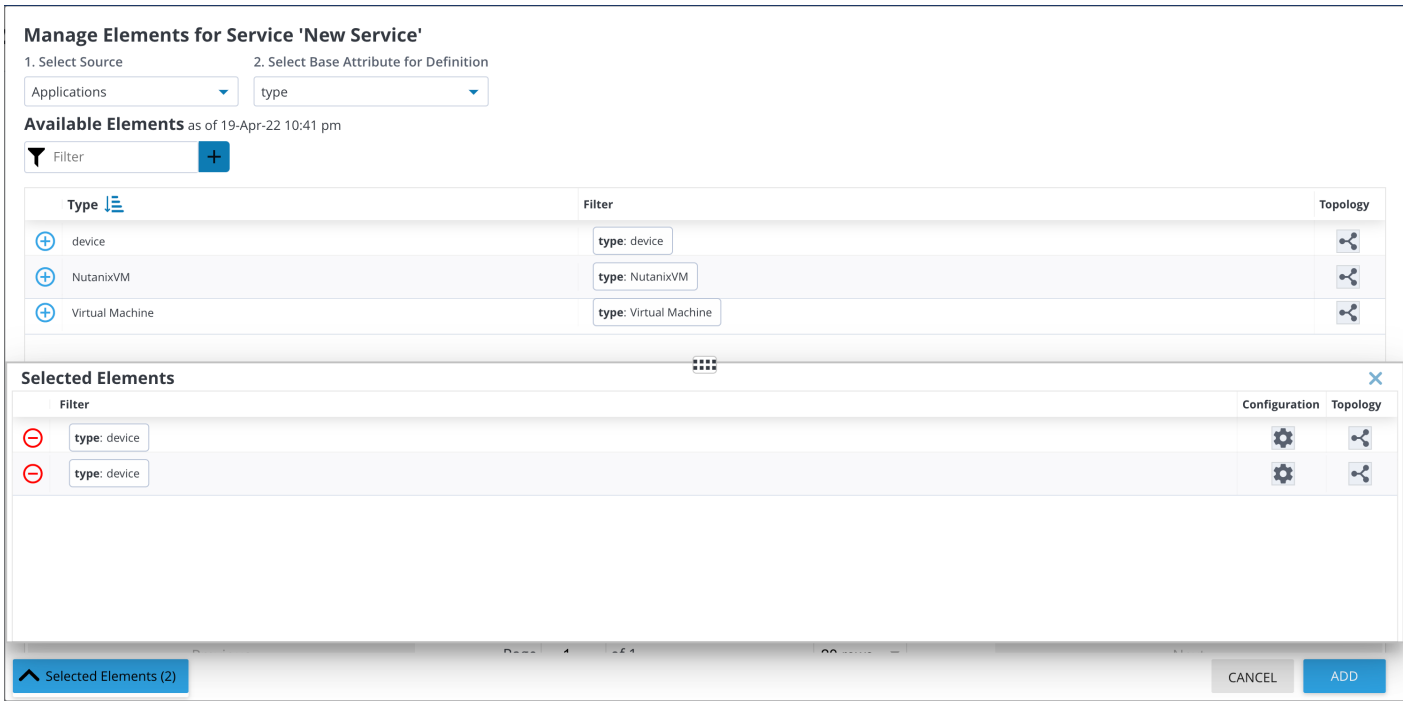
Follow these steps:

1. Click the filter icon on the service details panel.
2. Select a source from the **Select Source** drop-down list. Based on the selected source, a list of base attributes gets populated.
3. Select the corresponding **Base attribute** from the **Select Base Attribute for Definition** drop-down list.
4. The following table describes the monitoring domains and their basic attributes:

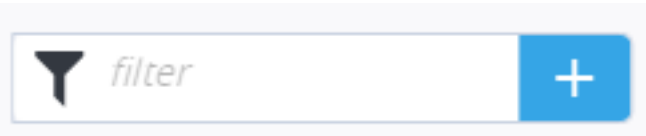
Option	
Monitoring Domain	**Basic Attribute**
Applications	applicationName
	agent
	hostname
	process
	type
	Lookup using a known entity
Infrastructure	uim.groups

Option	
	correlationNames
	primaryIpAddress
	virtualizationEnvironment
	vmware.Datacenter
	vmware.Cluster
	Lookup using a known entity
Network	spectrum.groups
	TopologyModelNameString
	capm.groups
	name
	NetworkAddress
	Lookup using a known entity
Network Services	name
	parentServiceName
	ModelHandle

After you selected the source and base attributes, the list of elements appears.



5. (Optional) Use the **Filter**



to filter the attributes.

You can use the **Operator** to drill down your search. Use the Operator such as **Equals** or **Regular expression** as per your requirement. Select the **Operator** as **Regular Expression** and Value as **Custom.*** and click **APPLY**

NOTE
This filter supports Wildcard expressions, such as .* and so on.

The filtered search results appear. You also have an option to edit the regular expression or name using the edit icon.

6. Click

next to the element.

The service name is added to the **Selected Elements** window.

NOTE
You can modify the Base attribute of the service that you created by selecting another Base attribute from the drop-down list. For example, let us assume that you want to add some agents to your service, select **Agent name** as the Base attribute.

7. Configure the service.
8. (Optional) View service topology.
9. Click **ADD.**

The application creates a service definition.

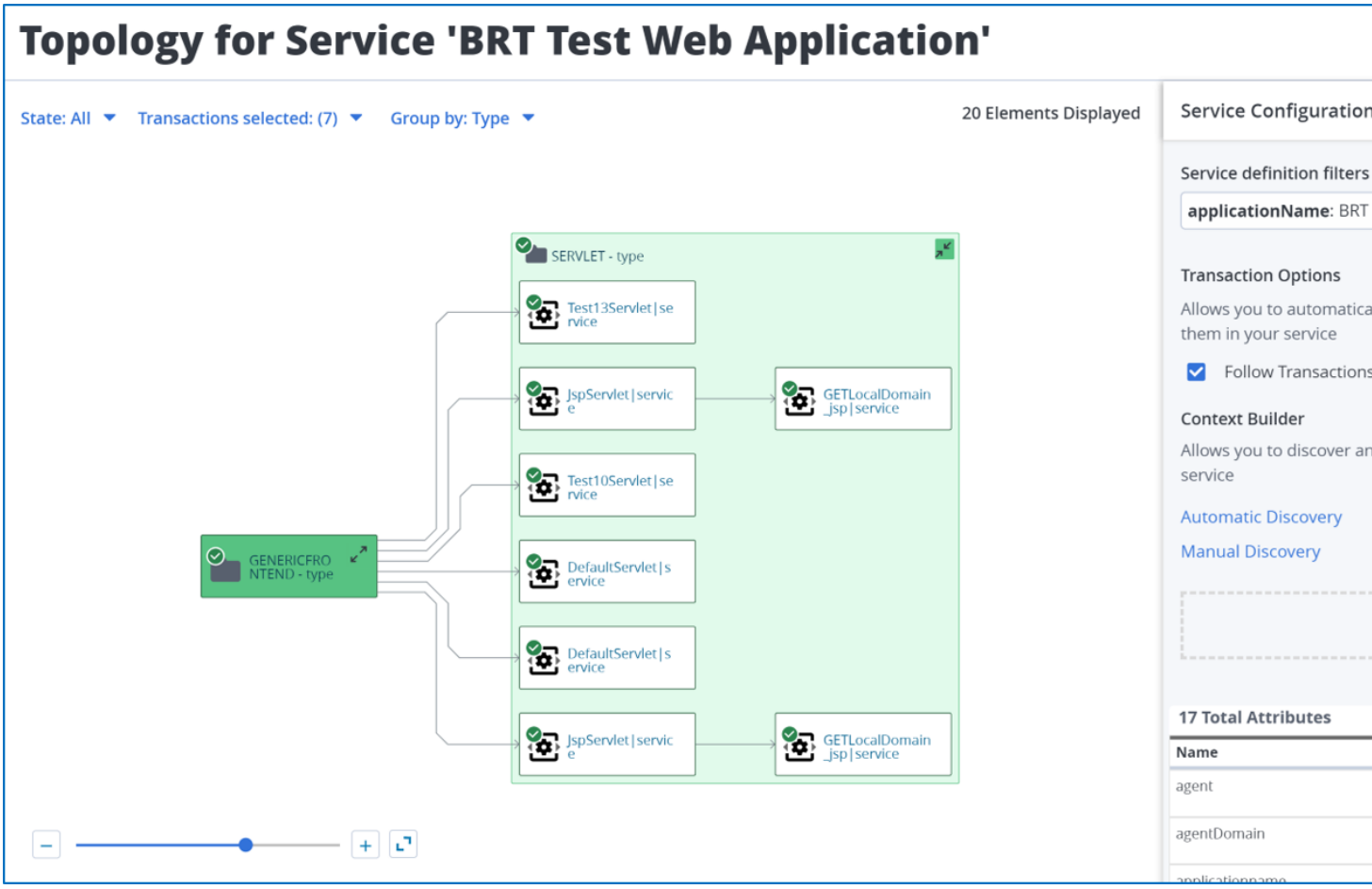
Service Configuration

The Service Configuration page allows you to build your own context view using Set Context Builder and you can also enable or disable the alarms.

{"URL":["https://digital-oi/service-analytics/serviceEdition/serviceEdit/topology"],"customLabelGetStarted":"Service Configuration","description":"task.dita_ccd02f8c-4260-4953-bad4-6703285e3062"}

Follow these steps:

1. Create a service or open an existing service.
2. Click **Edit** in the **Service Details** section.
3. Filter the elements.
4. Click + to add the required elements. The elements are added to the **Selected Elements** section.
5. Click the **Selected Elements** button to expand the dialog box.
6. Click the **Configuration** icon for the element.
7. Enter the following details in the **Service Configuration** panel.
   a) **Enable/Disable Alarm:** To enable or disable the alarm filter, click
   b) **Transaction Options:** Select the **Follow Transactions** option to follow the DX APM business transactions to other elements and to include these transactions in the service. The Topology View also includes a filter named Transactions Selected. You can use this filter to display the view for a single transaction or multiple transactions.



c) **Set Context Builder:** You can build your own context view using the following options in the Context Builder:

1. **Automatic Discovery:** Builds the context view for you automatically.
   a. Click **Automatic Discovery** and Save it. The context is saved.
2. **Manual Discovery:** Builds the context for you based on your choice.
   a. In the context menu, select **Add**
   b. To search for devices one level back, click on **Search, contains / Backward**
   c. Select the required Discovery Options:
      - App to System
      - System to Host
      - Host to System
      - System to App
      - 1-hop Network
   d. Click **Save**.

View Service Topology

Use the Service topology page to view the topology layout with color codes indicating the device availability. You can view the summary of devices, alarms, and attribute names.

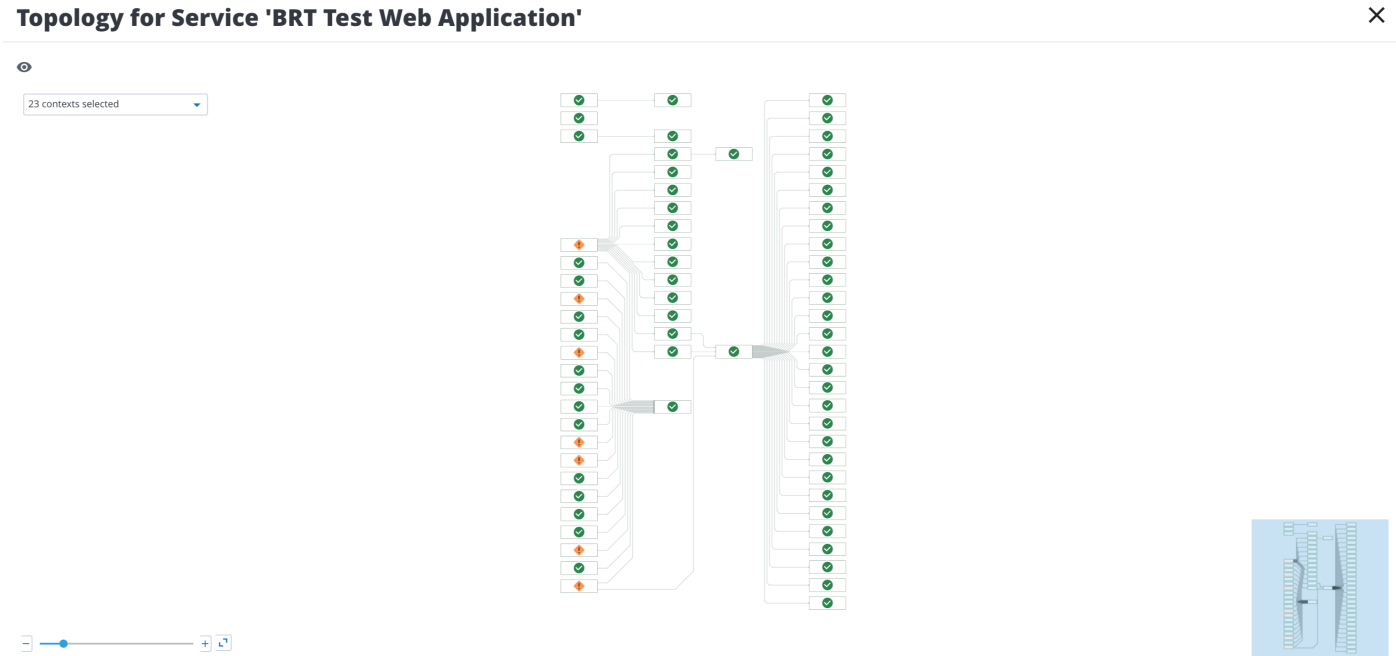
{"URL":["https://digital-oi/service-analytics/serviceEdition/serviceAdd/topology"],"description":"task.dita_821ef0fa-921f-4218-8791-7a02d9deb96f","new":"","new_video":"","admin":"","troubleshooting":{"masterkb":"","text":"","URL":[]},"pendo":"","video":[]}

You can view details of the service topology.

1. Click the **Topology**

   icon of a particular element.

   The topology layout appears for the selected



element.

NOTE
The red or green color on the topology layout does not indicate that the devices are down or up.

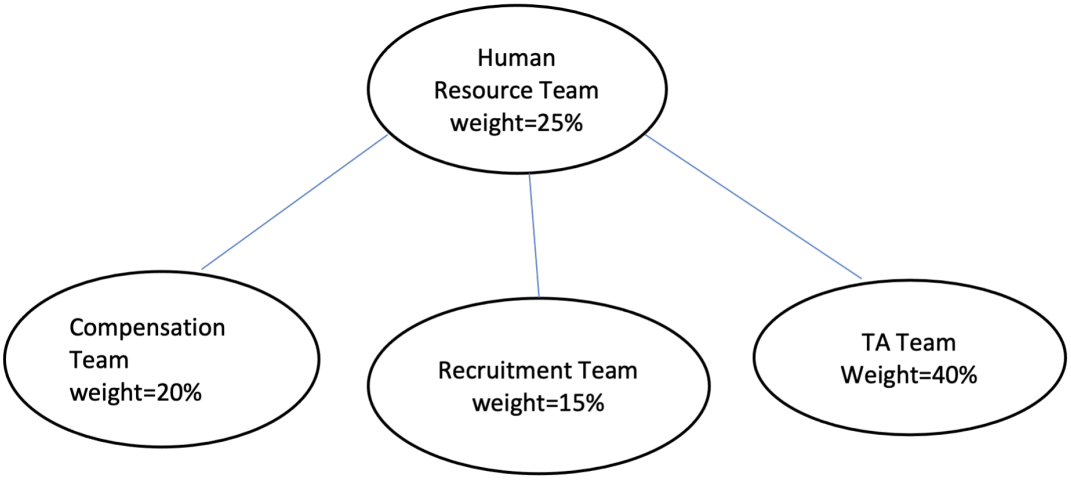
2. Click a device in the Topology hierarchy view to see the Summary of the device, Alarms, and Attributes of the device.
   - **Summary:** Contains details of Name, Type, Alert status.
   - **Alarms:** Contains details of the Severity, Date, and Time, Alarm message.
   - **Attributes:** Contains the list of attribute names and their corresponding End time values

Auto Weight

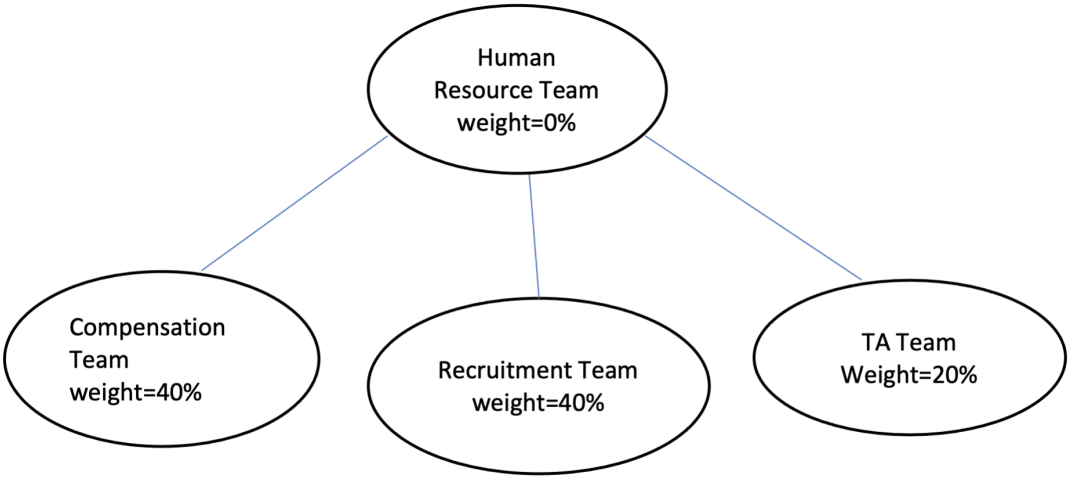
The**Auto-weight** enables you to assign weights to the sub-services. By default, Auto-weight is enabled. You can manually update the weight for each sub-service. Ensure that the total weight of the sub-service is equal to 100, else the edges appear in red color.

You can calculate the service impact propagation based on the following scenarios

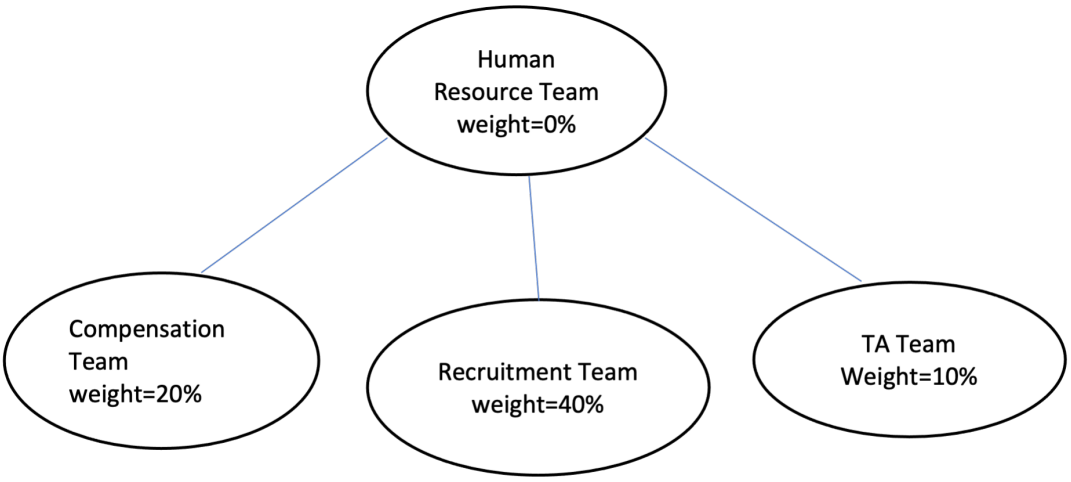
- **Scenario 1:** Parent services with topology along with their children services.
  In this scenario, the weights are distributed among children services, then the remaining percent of weight, that is (100-(sum of child services weight)) is assigned to the parent service.**For example,**the following image illustrates the weight distribution among parent and children services:



- **Scenario 2:** Parent services with own topology and 0% weight.
  In this scenario, the parent service weight is 0% as the 100% weight is distributed among child services, so when an alarm is triggered on the parent service, there will be no impact on the service hierarchy.
  For example, the following image illustrates the weight distribution among parent and children services:



- **Scenario 3:** Parent services with no topology.
  In this scenario, by default the parent service weight is 0% as it doesn't have its own topology. The alarms are not triggered on the parent service, and there will be no impact on the service hierarchy.**For example:** The following image illustrates the weight distribution among parent and children services:



Service Creation Templates

You can create a service using the **Create Service** tile on the **Settings** page or the **Create a Service** option on the Services page. However, the data is displayed only if all the details are provided. If there are any services in the hierarchy, then you are required to configure all the services in the hierarchy individually.

To help you get started with creating services, DX Operational Intelligence includes default service templates. A service template is a pre-designed layout or pattern that serves as a starting point for creating a service. The template provides a standardized structure that you can customize with the service content. A service template includes placeholders or predefined attributes that you can easily modify with the actual data. These service templates offer several benefits, including:

- **Consistency:** Templates help maintain consistency in service modeling, such as common attribute names of services, tags, custom properties, and location. You can use them across the organization level.
- **Efficiency:** Using a template saves time and effort as it eliminates the need to start from scratch. A template provides a ready-made structure, so you can focus on adding specific patterns or making necessary modifications.
- **Standardization:** Templates can enforce standard practices and guidelines at the organizational level.
- **Professionalism:** Templates are often designed by professionals or tenant administrators. Using a well-designed template can enhance the overall quality of IT monitoring.
- **Flexibility:** Templates can be customized to suit your specific needs. You can modify the templates at any time, and when they are modified, the corresponding services get updated.

This section includes the following information:

- Access Service Templates
- Default Service Templates
- Custom Service Templates
- Default Service Templates Reference

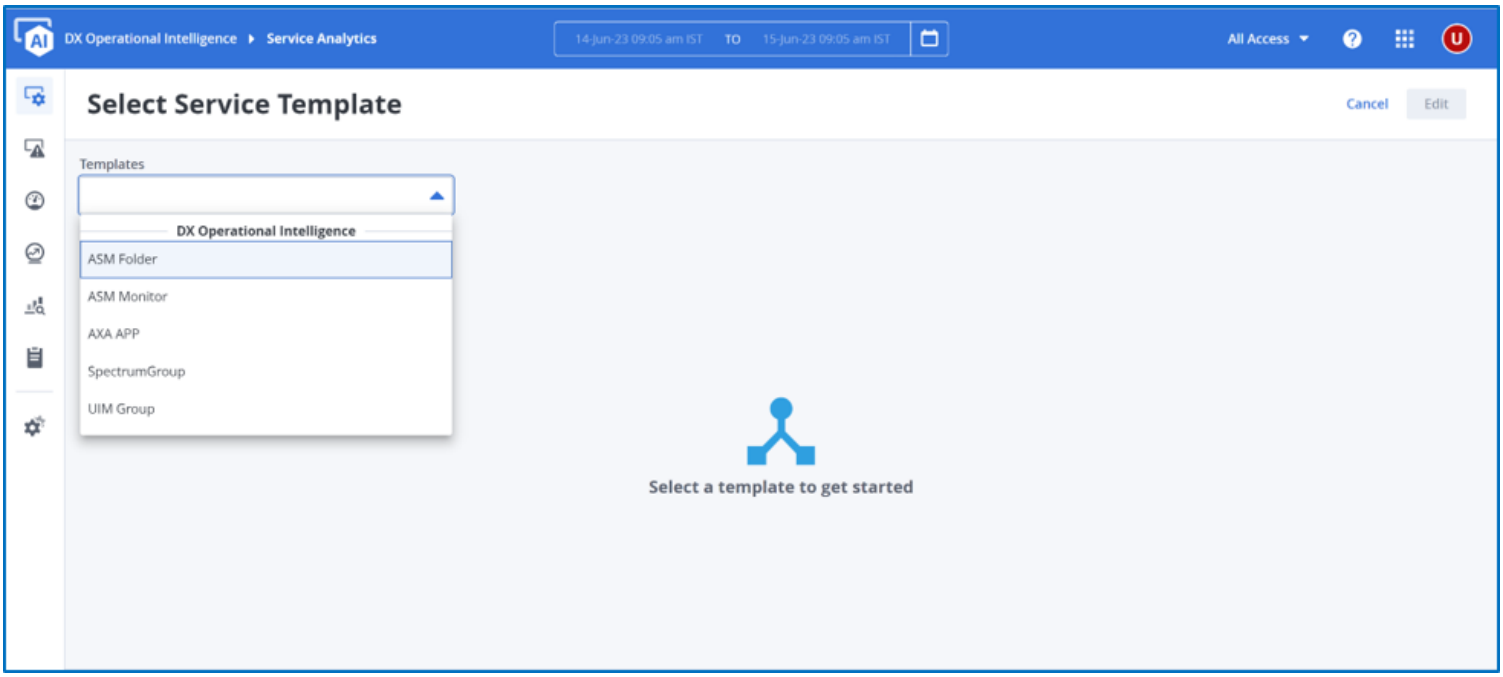
> **NOTE**
> You can use APIs to manage the service templates. For more information, see the Service Templates API section.

Access Service Templates

You can access the Service Templates in one of the following ways:

- **DX Operational Intelligence > Settings > Service Templates**
- On the **Services Overview** page, click the **ellipses** icon and click **Manage Service Templates**

The **Select Service Template** page is displayed which lists the service templates. The following image illustrates the default service templates.



The service templates are grouped under the following categories:

- **DX Operational Intelligence:** All the default templates are available under this category.
- **Custom:** All the templates that you edit and save are available under this category.

Default Service Templates

The default templates are displayed under **DX Operational Intelligence** and the templates that you edit and save are displayed under **Custom**.

The following default service templates are available out-of-the-box:

- **ASM Folder:** Use the ASM Folder template to create a service based on the ASM folders.
- **ASM Monitor:** Use the ASM Monitor template to create a service based on the ASM monitor (vertices).
- **AXA App:** Use the AXA App template to create a service based on the App Experience Analytics application. The application for which the service is being created must be integrated and monitored by Application Performance Management to be able to create the service.
- **SpectrumGroup:** Use the Spectrum Group template to create a service based on the Spectrum Groups.
- **UIM Group:** Use the UIM Group template to create a service based on the UIM group.

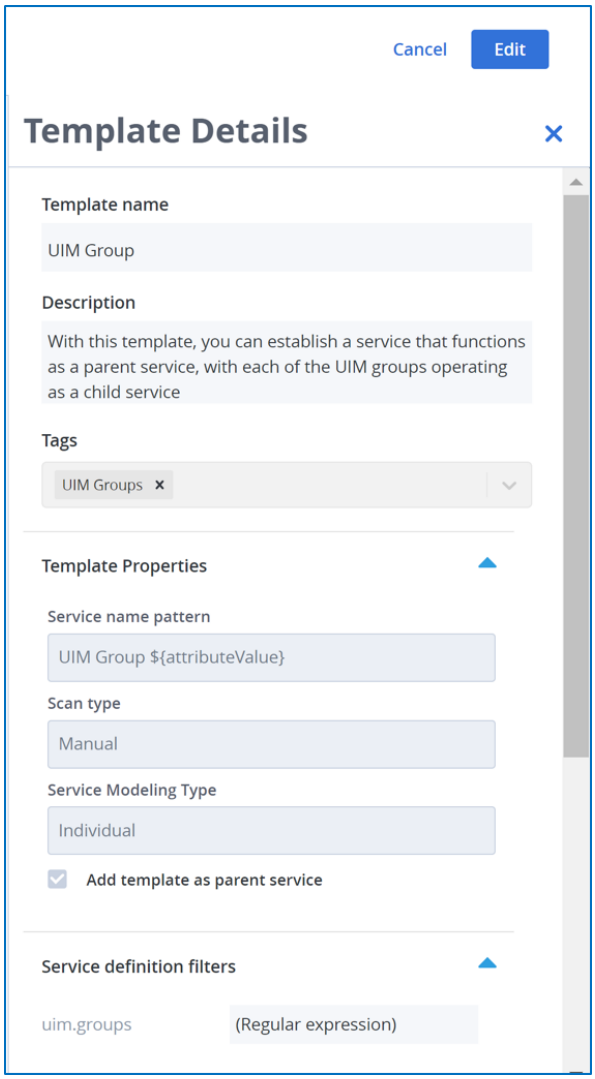
> **NOTE**
> For more information about the template details, see the Default Service Templates Reference section on this page.

Custom Service Templates

The default templates that you edit and save are considered as custom service templates. The following illustration shows how to create a custom service template using the default template.

Follow these steps:

1. Navigate to the **Service Templates** page in one of the following ways:
   - **DX Operational Intelligence > Settings > Service Templates**
   - On the **Services Overview** page, click the **ellipses** and click **Manage Service Templates**.
2. Select the required template from the list.
   The entity is displayed.
3. Click the entity to view the template details.
   The **Template Details** panel displays the pre-configured values for Template name, Description, Tags, and Template Properties.



NOTE
You may edit these values if necessary. However, you must specify the **Service definition filters** to save the template.

4. Click **Edit** on the top of the panel.
5. Click the entity again to edit the template details.
   a. (Required) **Template Name:** Edit the template name. A custom template cannot use the same name as the default template.
   b. (Optional) **Description:** Edit the description.
   c. (Optional) **Tags:** Add tags or remove the default tag.
6. Edit the following fields in the **Template Properties** section:
   – (Optional) **Service Name Pattern:** Edit the service name pattern, if necessary. Each template has a pre-defined service name pattern. If you want to edit the pattern, ensure that the pattern is unique and ${attributeValue} is included in the pattern. Examples of the patterns are: ${attributeValue} - application, application: ${attributeValue}, ${attributeValue}, and so on.
   – **Scan Type:** Select one of the following options:
     - **Manual:** Select this option to add the CIs manually. The template creates services from all the devices that are available at the time the service is created. If any new devices are added later, services are not created for those devices.

NOTE
If you delete the service or the service template, they do not have any impact on each other.

- **Automatic:** If you select this option, a service is created whenever a new CI is added. That is, a service is created automatically whenever any new device is added that matches the template definition.

  NOTE
  If you delete a service, the service is recreated if there is a match when the scan is run. However, if you delete the service template, the existing services are not deleted but no new services are created.

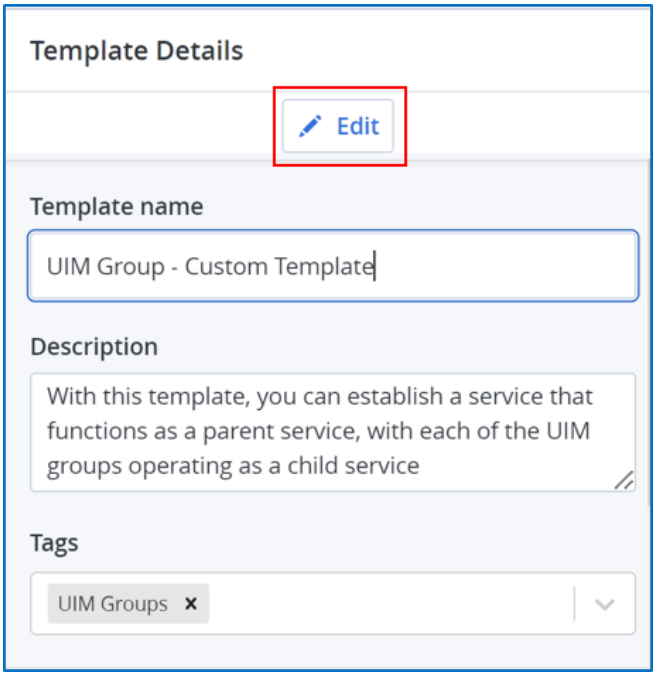
– **Service Modeling Type:** Determines how many services are created.
  - **Aggregate:** If you select this option, a single service is created with the name of the template and all the matches are part of this service. For example, if the regex pattern matches 20 groups, then a single service is created with the name of the template and all the 20 groups are part of this service. The **Add template as parent service** checkbox is disabled.
  - **Individual:** If you select this option, a service is created for every match. For example, if the regex pattern matches 20 unique groups, then 20 services are created.

    NOTE
    The **Add template as parent service** checkbox is optional. If you select this option, a parent service is created with the name of the template and all the 20 services become the children of the parent service.

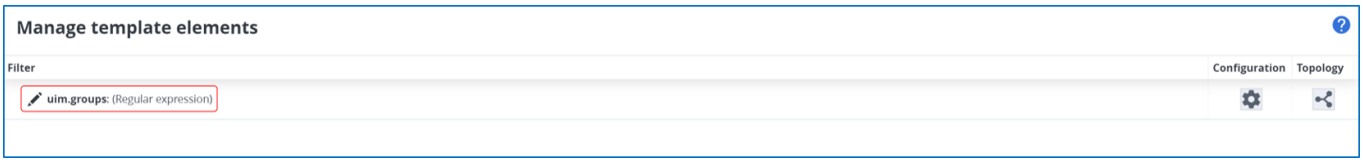
7. Configure the **Service Definition Filter**.

   NOTE
   You can save the template as a custom template only after you configure this filter.

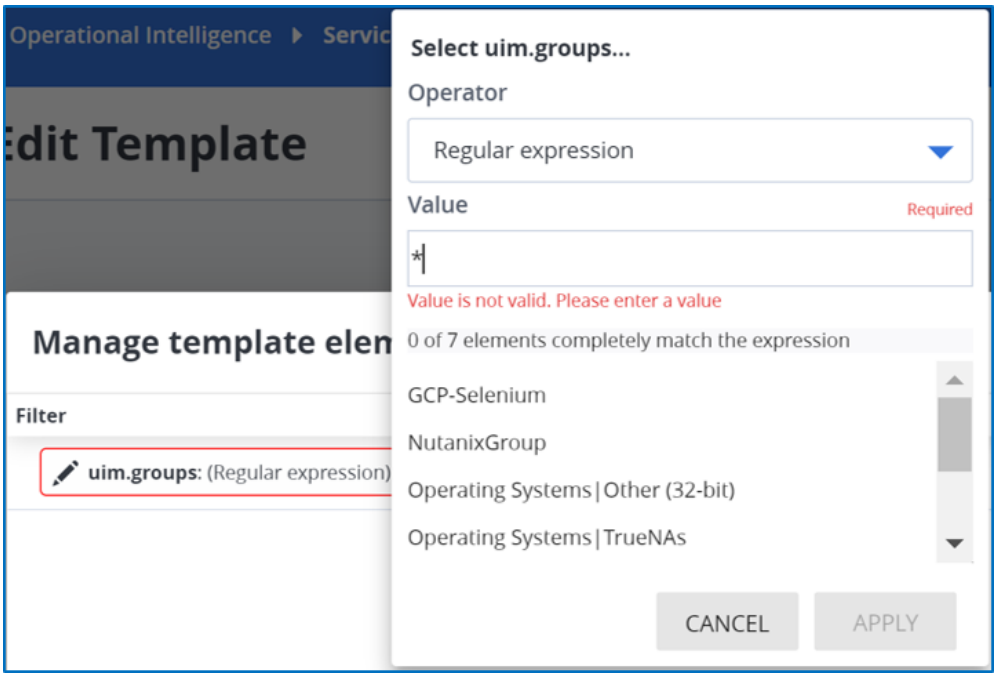
   a. Click **Edit** on the top of the panel.



   The **Manage Template Elements** dialog is displayed.



   b. Click the highlighted box to select the definition filter as shown:



c. Select the Operator and enter the Value.

NOTE

If using **Regular Expression**, enter the value as ***** to display the avialab;e elements.

8. Configure the other sections (**Key Performance Indicators, Context for situations, Custom properties,** and **Location**). For more information, see the Create a Service section.
9. Click **Save**.
   Wait for a minute for the service to be created on the Services page.

Default Service Templates Reference

This section describes the details of each of the service templates:

- ASM Folder
- ASM Monitor
- AXA App
- Spectrum Group
- UIM Group

ASM Folder

The ASM Folder service template is pre-configured with the following details:

Name	Description
Template Name	ASM Folder
Description	This template creates a service from the ASM Folder.
Tags	ASM Folders
Template Properties	
Service name pattern	Folder- ${attributeValue}

Name	Description
Scan Type	Manual
Service Modeling Type	Individual
Add template as parent service	Selected
Service Definition Filters	folder

ASM Monitor

The ASM Monitor service template is pre-configured with the following details:

Name	Description
Template Name	ASM Monitor
Description	This template creates a service from the ASM Monitor, if the ASM monitors are available.
Tags	ASM Monitor
Template Properties	
Service name pattern	Monitor- ${attributeValue}
Scan Type	Manual
Service Modeling Type	Individual
Add template as parent service	Selected
Service Definition Filters	monitor

AXA App

You can use the AXA App template to create a service based on the App Experience Analytics application. The application for which the service is being created must be integrated and monitored by Application Performance Management to be able to create the service. This template is pre-configured with the following details:

Name	Description
Template Name	AXA App
Description	This template creates a service from Axa Apps. Prerequisite that APM-AXA integration must be enabled.
Tags	Axa Application Service
Template Properties	
Service name pattern	AXA App- ${attributeValue}
Scan Type	Manual
Service Modeling Type	Individual
Add template as parent service	Selected
Service Definition Filters	AXAAppName

Spectrum Group

You can use the Spectrum Group template to create a service based on the Spectrum Groups. This template is pre-configured with the following details:

Name	Description
Template Name	SpectrumGroup
Description	With this template, you can establish a service that functions as a parent service, with each of the spectrum groups operating as a child service.
Tags	Channel, Data, Service
Template Properties	
Service name pattern	${attributeValue} Spectrum Group
Scan Type	Manual
Service Modeling Type	Aggregate
Add template as parent service	Not Selected
Service Definition Filters	spectrum.groups:(Regular expression)

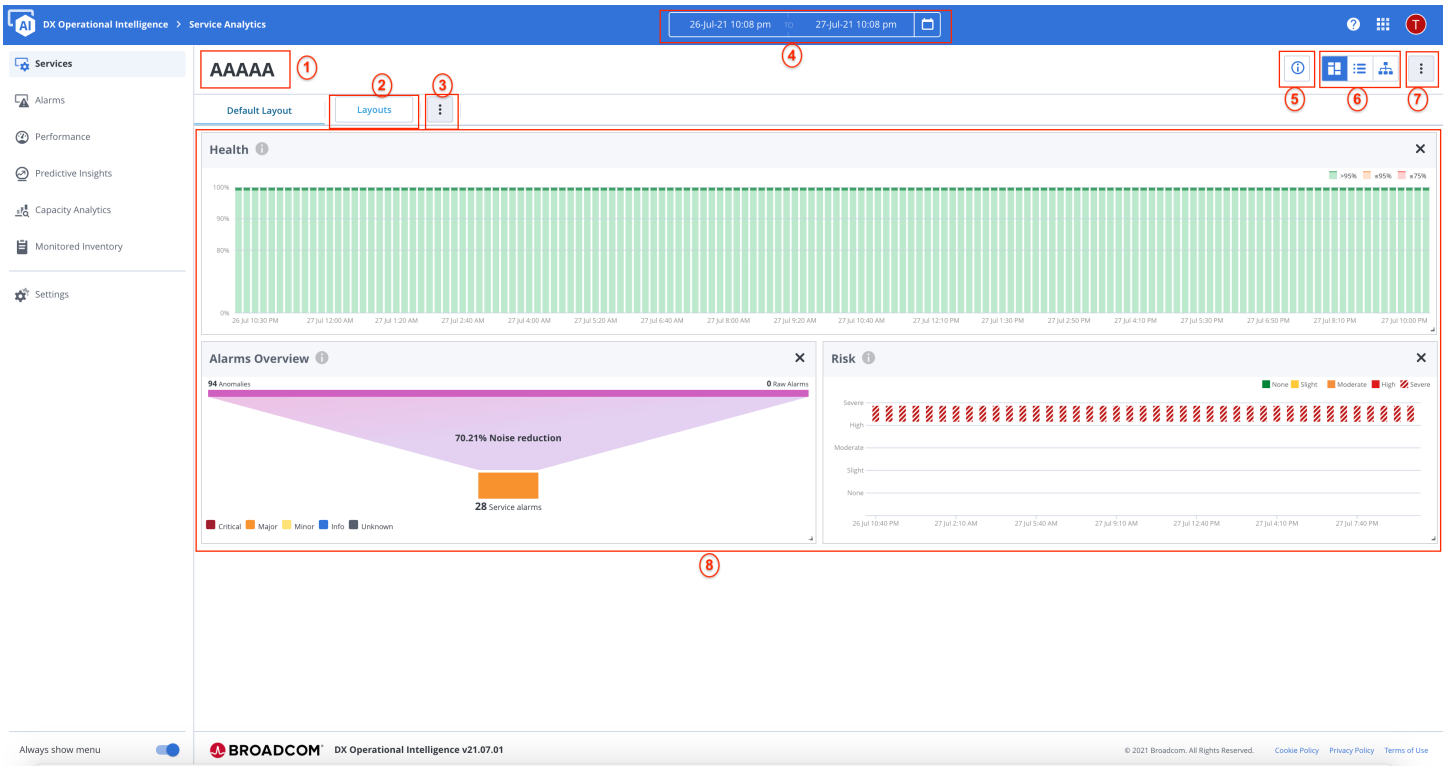
UIM Group

You can use the UIM Group template to create a service based on the UIM group. This template is pre-configured with the following details:

Name	Description
Template Name	UIM Group
Description	With this template, you can establish a service that functions as a parent service, with each of the UIM groups operating as a child service.
Tags	UIM Groups
Template Properties	
Service name pattern	UIM Group ${attributeValue}
Scan Type	Manual
Service Modeling Type	Individual
Add template as parent service	Selected
Service Definition Filters	uim.groups:(Regular expression)

Service Details Page

Service Analytics provides insights into a required service that you select on the **Services Overview** page. The Service Details view provides the complete details of each service, which includes information such as Health, Alarm overview, Risk, Availability, Operations topology, Service topology, and Custom metrics. You can leverage the Service Details view to analyze the services and find services that need immediate attention or the services that are at high risk of failure.



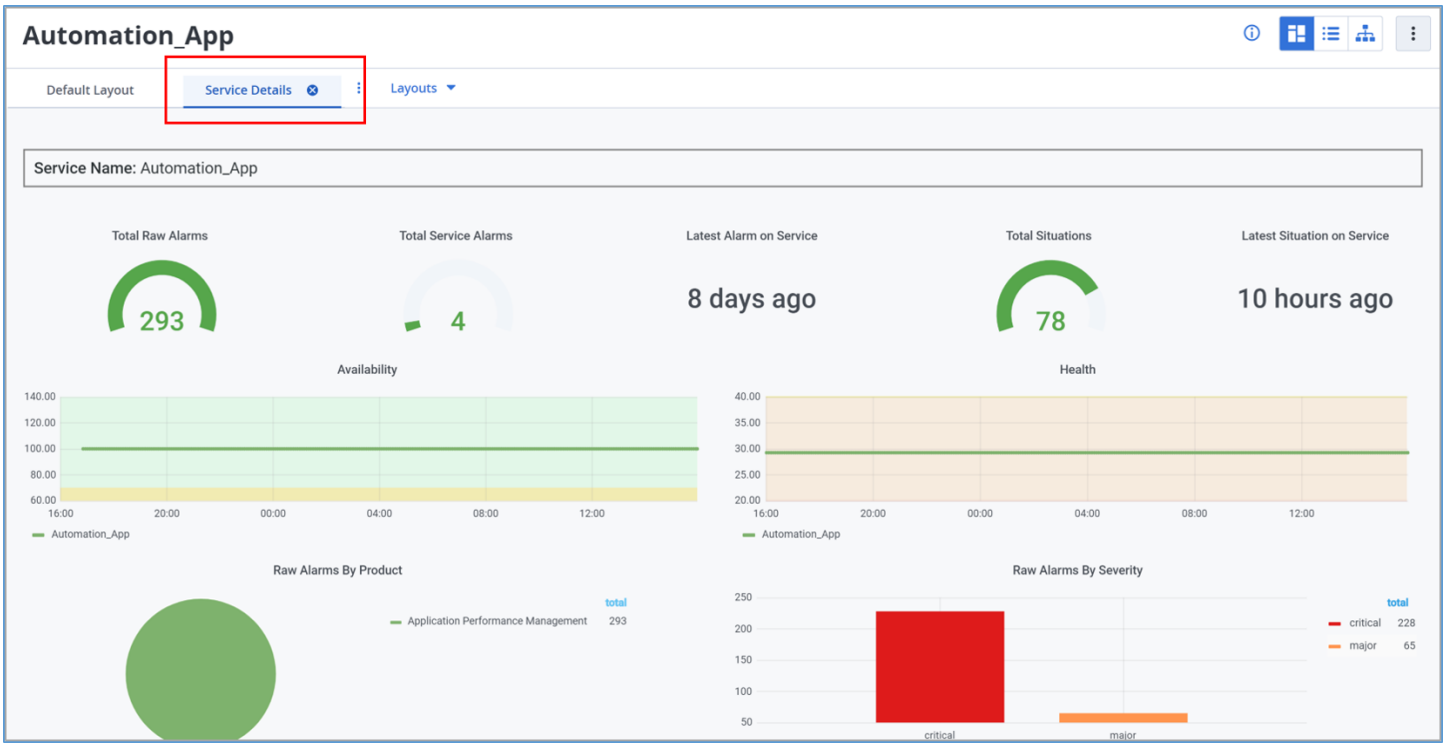
Name	Description
Service Name (1)	Displays the service name that you select on the Services Overview page.
Layouts (2)	Displays the pinned and all layouts that you saved. You can also use the filter option to search the layout.
Save/Delete Layout () (3)	Allows you to save or delete a layout.
Filter by Date/Time Range (4)	The Service details view provides a Custom filter option in the Date/Time filter, which enables you to pick a particular start date/time and a particular end date/time to help you narrow down the service details that you are looking for.
Service Details (5)	Click icon to view the service details of the service, such as service name, description, creation time, tags.

Name	Description
Charts View (6)	Use this option to view the service details in the chart view. You can perform the following actions by clicking • Edit service • Delete service • Add maintenance window • Set as OI landing page • Customize KPIs to be displayed on the Services view
Monitored Inventory View (6)	Use this option to view the monitor the inventories in the context of the service. You can perform the following actions by clicking • Edit service • Delete service • Add maintenance window • Set as OI landing page
Topology View (7)	The Topology view is a graphical representation of the relationship among services and its sub-services that support them. You can perform the following actions by clicking • Edit service • Delete service • Add maintenance window • Set as OI landing page
Three Dot Menu (7)	Use this option to perform the following actions: • Edit service • Delete service • Add maintenance window • Set as OI landing page • Select KPI Widgets
Service KPIs Layout (8)	You can view detailed information of service KPIs.

Embed DX Dashboards

You can embed a DX Dashboard to be viewed on the Service Analytics Details page. By default, the **Service Details** dashboard, which is the OOTB dashboard is embedded. However, you must pin this dashboard on the Service Analytics Details page to appear as a tab.

The following image displays the embedded Service Details dashboard:



To embed the dashboard,

- **OOTB Dashboards:** Make a copy of the dashboard and then add the **DX OI Service Details** tag on the **DX Dashboards > Dashboard Settings > General** page.
- **Custom Dashboards:** Add the **DX OI Service Details** tag on the **DX Dashboards > Dashboard Settings > General** page.

  NOTE

  - However, to maintain the service context on the Service Analytics Details page for both OOTB and custom dashboards, ensure that the service filter is added using the query variable.
  - If a user does not have access to dashboards, then the embedded dashboards are not displayed on the Service Analytics Details page.
  - The dashboard data that is displayed depends on the user role. For a custom role user, data is displayed only for the assigned universes. However, for the OOTB role user, all the data in the tenant is displayed.
  - When an OOTB role user navigates to DX Dashboards from DX Operational Intelligence, the universe that is selected in DX Operational Intelligence is not maintained in DX Dashboards. However, when the same user navigates to other DX Operational Intelligence pages using the drill-downs in the embedded dashboard, the selected universe context is maintained.
  - You can edit or delete the dashboards only in DX Dashboards but not on the Service Analytics Details page. You can only pin or unpin them.

Service Details Layout

You can select your own Service Details layouts for viewing your service KPIs. You can drag and drop your service KPI charts and organize them as required, which helps you analyze multiple service KPI charts at a time. For more information, see Service Personalization.

You can view the details of the service in the following layouts:

- Chart View
- Monitored Inventory View
- Topology View

Charts View

The Charts view provides service details and allows you to perform these actions: Edit a service, Delete a service, Add Maintenance window, Add widgets, and Manage layout.

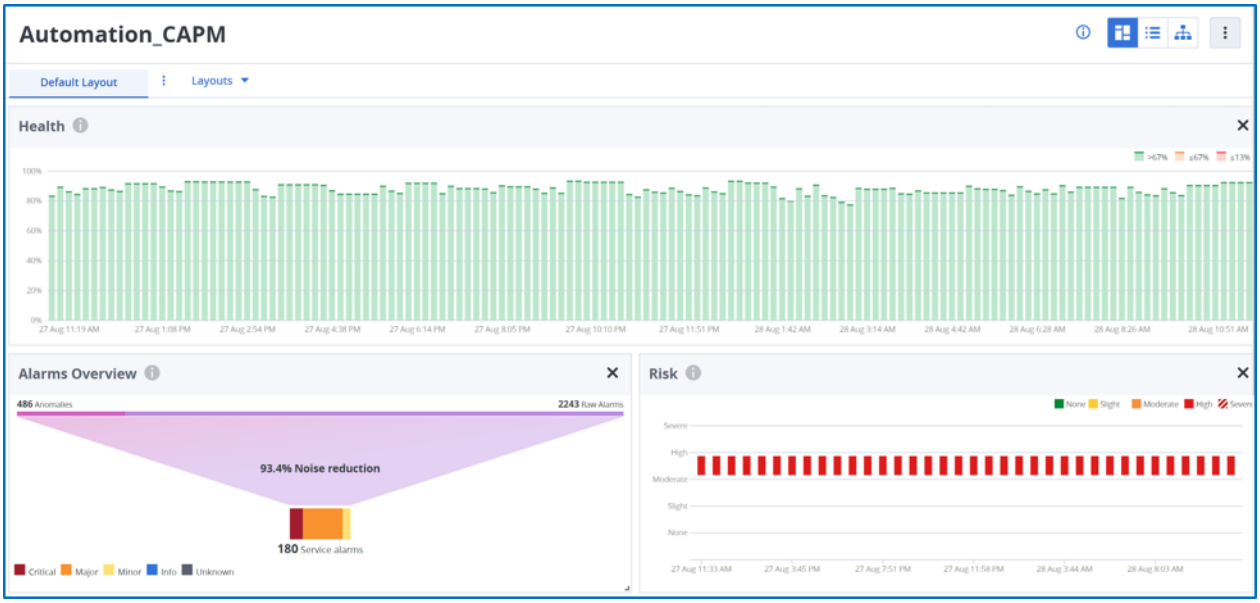
By default, the service details are visible in the Charts view. To view the service details in **Charts View**, click the icon.

- Manage Layout
- Edit a Service
- Delete a Service
- Add Maintenance Window
- Set as OI Landing Page
- Add Widgets

{"URL":["https://digital-oi/service-analytics/*/*"],"customLabelGetStarted":"Charts View","description":"concept.dita_b4c99982-cf69-4555-bce9-135aff067100","customCards":[{"type":"customize","id":"concept.dita_3355ccbe-acd8-4f41-8f01-d7bfbb9464e7","url":"https://techdocs.broadcom.com/us/en/ca-enterprise-software/it-operations-management/dx-operational-intelligence-saas/SaaS/using/Service-Analytics/Services-User-Interface/Service-Details-Page/Service-Details-Layout/Chart-View/Add-Service-Widgets.html","title":"Add Service Widgets"},{"type":"customize","id":"concept.dita_59c63824-456c-4490-9b5e-f05fb9fcce8d","url":"https://techdocs.broadcom.com/us/en/ca-enterprise-software/it-operations-management/dx-operational-intelligence-saas/SaaS/using/Service-Analytics/Services-User-Interface/Service-Details-Page/Service-Details-Layout/Chart-View/Manage-Layouts.html","title":"Manage Layouts"}]}

Manage Layouts

When you open the **Charts View**, by default, the **Default Layout** is displayed with the details for the following service KPIs: **Health, Alarms Overview, Risk, Raw Alarms, Situations, Anomalies,** and **SLI Metrics and SLOs**. You can drag and drop the service KPI widgets and organize them as required, which helps you analyze multiple service KPI charts at a time.



DX Operational Intelligence provides additional KPIs out-of-the-box that you can add to this default layout. However, you must save the default layout with a new name.

Follow these steps:

1. Open the **Charts View.**
2. **View the available KPIs:** Click the icon and then click **Customize View** as shown in the image.

The following KPIs are available:

- Alarms Overview *
- Anomalies *
- Availability
- Capacity Issues
- Health *
- Monitored Inventory
- Operations Topology
- Predictions
- Raw Alarms *
- Risk *
- Service Hierarchy
- Situations *
- SLI Metrics and SLOs *

NOTE
* Indicates that they are available in the Default Layout. For more information about each of these KPIs, see the Add Service Widgets section.

3. **Add the KPIs:** Click the toggle for the required KPI and enable the KPI.
   The KPIs are added to the layout.
4. **Save the layout:** If you have added KPIs or have edited the layout, you can save the layout.

NOTE
When you add KPIs to an existing custom layout, the layout name has * prefixed to the name. The * indicates that there are unsaved changes.

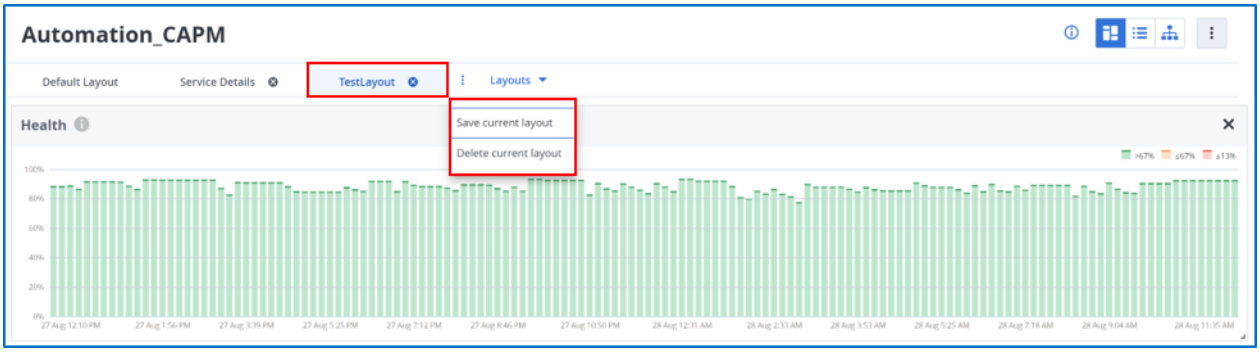
a. Click the icon and click **Save current layout**.
b. Provide the following information:

- **Make Public:** Select this checkbox to make this layout available to other users.
- **Layout Name:** Enter a name for the layout.
- Click the **Pin Layout** icon

  if you want to feature this layout as a tab.

c. Click **Save**.
   This layout is added to the **Layouts** list. If you have pinned the layout, the layout is listed under the **Pinned** section with this icon against the name. If you have not pinned the layout, you can also click the **Unpinned** icon for the layout under the **All Layouts** section.

You can perform the following actions:



- **Save Current Layout:** You can use this option to add KPIs to the default layout, or you can edit the custom layouts that you create. Select the layout as shown in the image for this option to be enabled.
- **Delete Current Layout:** You can use this option to delete the custom layouts that you create. To delete, click the tab (for example, TestLayout) to select it, click the icon, and then click **Delete current layout**. This option is enabled only when you select the tab.

  NOTE

  - You cannot delete the default layout.
  - You must pin the custom layout to display it as a tab before you delete.

Edit a Service

The Edit service allows you to modify the service and sub-service properties and topologies. You can also manage elements, and modify service or sub-service details such as service name, description, tags associated with the service, availability metrics, location, and custom properties.

You can only update the service properties (description, tags, and custom properties) and not the topology and its relationship. For more information about service properties, see Create Service.

{"URL":["https://digital-oi/service-analytics/serviceEdition/serviceEdit"],"description":"concept.dita_4233bb6f-ff3d-4ce1-be7d-63a9468f9cc0","new":"","new_video":"","admin":"","troubleshooting":{"masterkb":"","text":"","URL":[]},"pendo":"","video":[]}

Steps to Edit a Service

Follow these steps:

1. Click an existing service or sub-service on the **Service Overview** page.
2. Click the ⋮ icon, **Edit Service** option.

   The **Edit Service** page appears.
3. **Modify Service Details:** You can modify the following service details:
   a) Description
   b) Tags
   c) Service Definition Filters
   d) Key Performance Indicators
   e) Context for Situations
   f) Custom Properties
   g) Location

   NOTE

   - You cannot rename a service or cannot delete a service. By default, these two options are disabled while modifying a service.
   - You can only delete a service from the hierarchy.
4. **Add or Modify Services or Sub-Services**
   a) Click **Edit** in the **Service Details** section.
   b) Select the required **Source** and **Base Attribute for Definition**. For example, say Applications and Application name respectively.
   c) Click

   icon next to the elements.

   The **Assign element to service** pop-up appears
   d) For a new service, select **New** under the **Service** drop-down field and enter a new service name.
   e) For existing service, select **Existing** under the **Service** drop-down field and select the existing service name.
   f) Click **OK**.
   g) Click **UPDATE**.
5. **Add or Modify the Topology**

   - Click **Add Group Service** to add a new service to the topology.
   - You can modify the topology by using the drag-and-drop method on the existing services or sub-services.
6. **Update Auto Weight**

   - You can update the weights of a sub-service by clicking the percentage in the mapping. Ensure that the total weight of the sub-service is equal to 100, else the edges appear in the red color.

   NOTE
   You can also select a service or sub-service by performing the following actions from the **Service Details** pane:

- Click

  to manage all the elements.
- Click

  to view the topology layout of the service.

7. After you make the changes, Click **Update.**

Delete Service

The Delete Service option allows you to delete a parent service or child service with the respective monitoring, metric calculations associated with the service.

The service definition is deleted immediately, and the removal of service references in service alarms, service KPIs, predictions, health, and so on, is performed in the background in small batches to reduce the overhead. All the data that is associated with entities within the service such as anomalies, inventory, and so on is retained. You must fulfill the following conditions in order to delete a service:

- If the deleted service does not have any parent service, it would be considered as an individual or stand-alone service.
- If you delete a first child service level under a parent service, the next child service level moves up the hierarchy.
- If you delete a parent service that has a child service that is shared by another parent service, then that child service does not get deleted but the relationship with the parent service gets removed.
- If a child service is deleted, you must manually update the weight of the service by clicking the percentage value. Ensure that the total weight of the sub-service is equal to 100, else the edges appear in the red color.

> **NOTE**
> You cannot create a service with the same name to the one that you are deleting, you must wait until the delete operation is completed (You see a message as **Service is pending deletion**). You must create a service with another unique name.

Steps to Delete a Service

To delete a service, perform the following steps:

1. From the **Services Overview** page, click the service that you want to delete.

   The **Service Details** page appears.
2. Click

   , **Delete Service**.

   A caution message appears.
3. Select **Including underlying sub-services** option to delete the parent service and the underlying sub-services (child service).
4. Click **Delete**.

Add Service Widgets

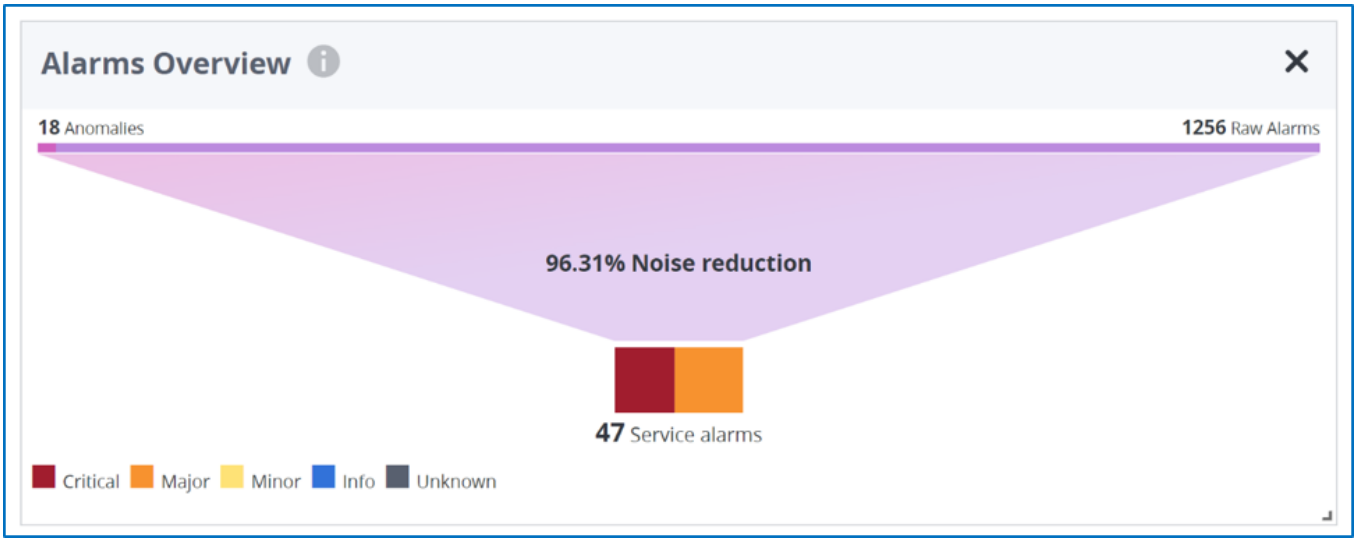
DX Operational Intelligence provides the following service widgets out-of-the-box. You can add these Service Widgets on the Service Details page:

- Alarms Overview *
- Anomalies *
- Availability
- Capacity Issues
- Health *
- Monitored Inventory
- Operations Topology
- Predictions
- Raw Alarms *
- Risk *
- Service Hierarchy
- Situations *
- SLI Metrics and SLOs

NOTE
* Indicates that these widgets appear on the Service Details page by default.

Alarms Overview

The Alarms Overview widget provides an overview of the alarms that are mapped to a service. You can view the total number of Anomalies Alarms, Service Alarms, Raw Alarms, and the percentage of noise reduction. Noise reduction is done by compressing anomalies and raw alarms into service alarm clusters. To view detailed information on alarms, click any alarm category to navigate to the Alarm Analytics view.



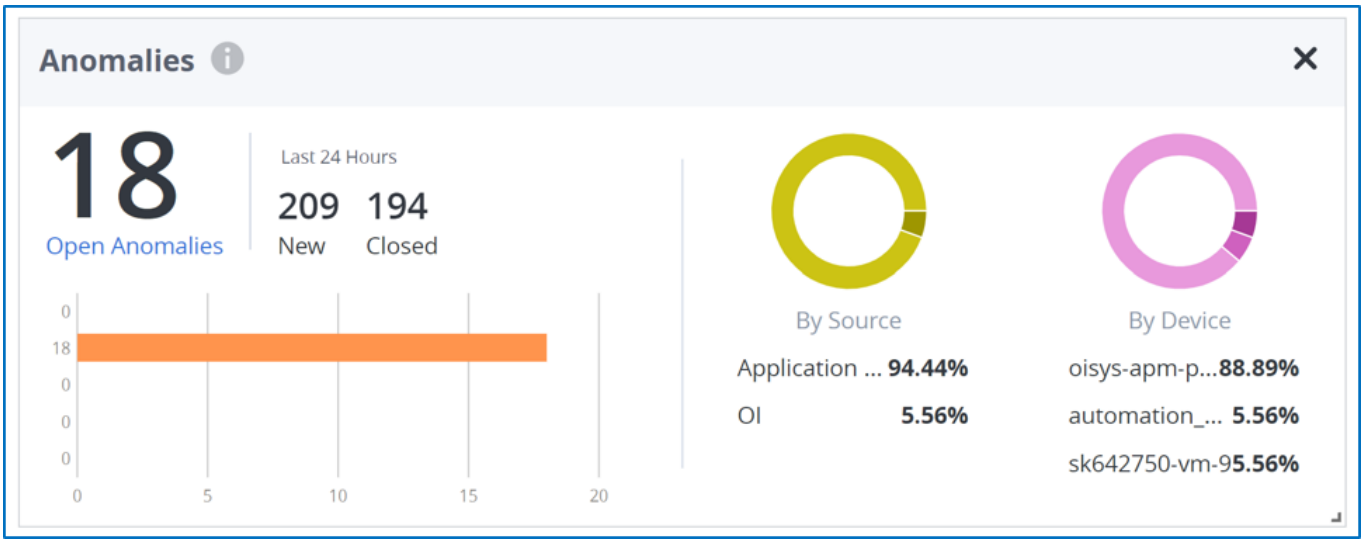
Service Alarms are alarm groups that are created based on Raw alarms and Anomaly alarms. These alarms are grouped based on the time that the alert is raised and the topology of the service (in which CI is modeled). The group describes the impact that the alarms have on the Service. One of the alarms in the group is marked as the Root Cause of the group. Root Cause alarm helps to find the issue and severity of the group. For more information, see the Service Alarms view.

NOTE

- Service Alarms are disabled for new tenants. To re-enable the Service alarms for new tenants, contact Broadcom Support.
- The Service alarm severity is the highest severity value among all the active raw alarms of the service alarm and not the root cause alarm severity.
- The **Alarms (+ subs)** column displays only the currently active alarms count and does not include closed alarms or prediction alarms in the alarm count.

Anomalies

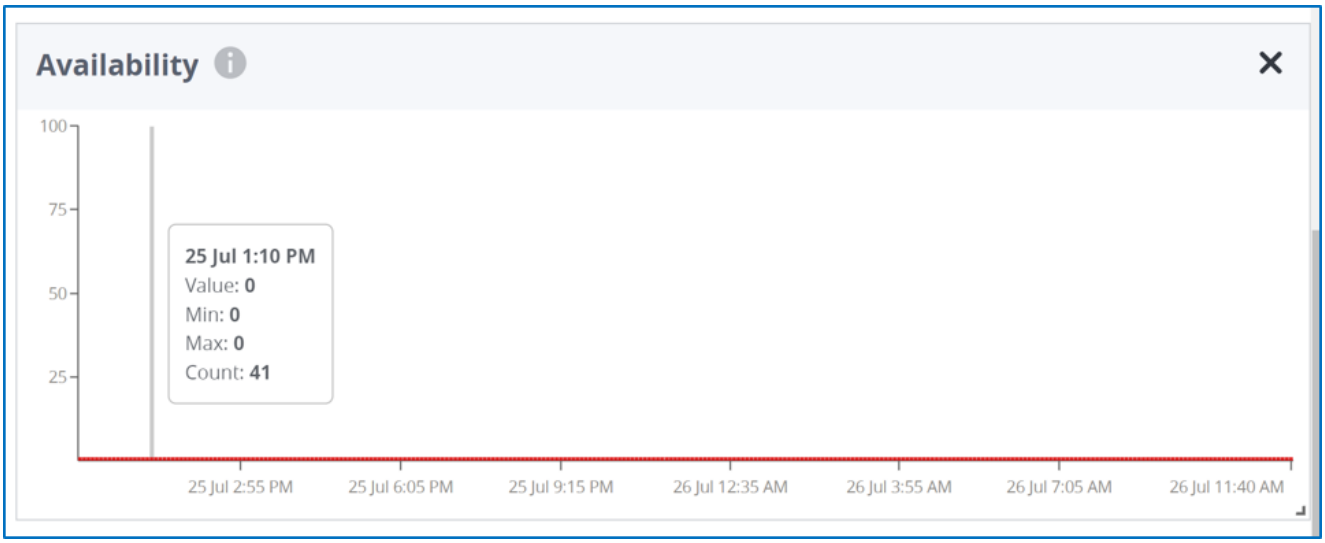
The Anomalies widget displays the total number of anomalies that are associated with the service component for the selected time period. The widget also includes the anomalies of children (if any) impacting the service.



Click the **Open Anomalies** link to navigate to the **All Alarms** page.

Availability

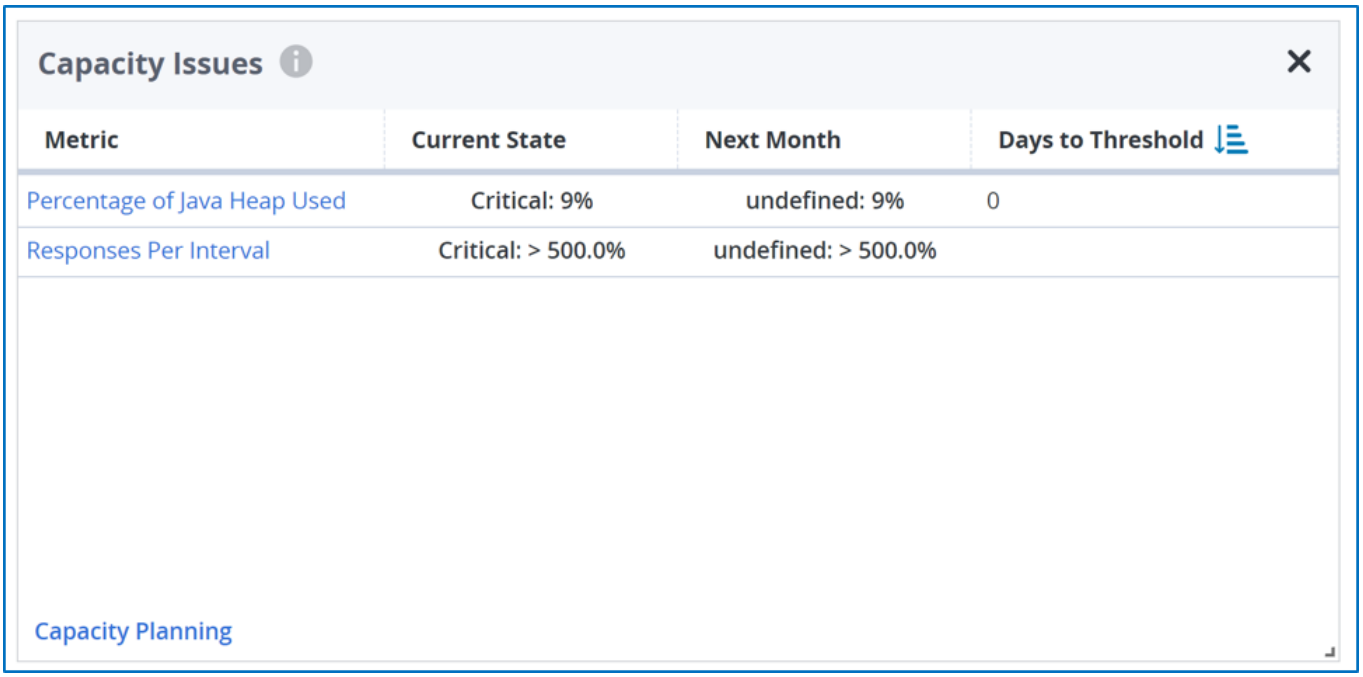
The Availability widget indicates the health and current availability status of the service, and it displays the percentage of time the service is up and operational. The Availability status depends on the state of the monitor that is associated with a service. Availability is calculated even when the service is under maintenance.



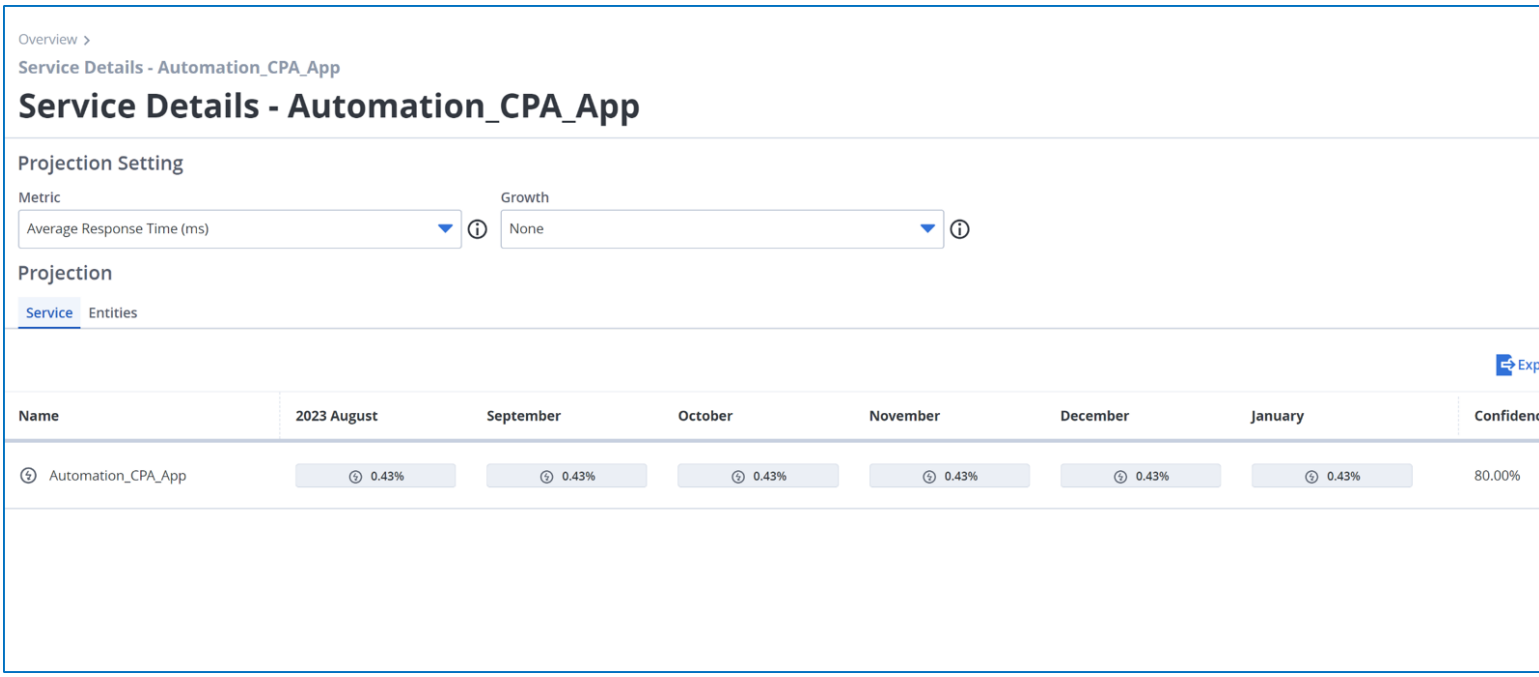
- indicates that the service is in a good health and available.
- indicates that the service is down and unavailable.
- indicates that the status of the service is unknown.
- indicates that the service is inactive.
- indicates that the service is in maintenance mode

Capacity Issues

The Capacity issues (Capacity Analytics) widget displays the capacity issues of a selected service. Also displays predictions relevant to the time context.



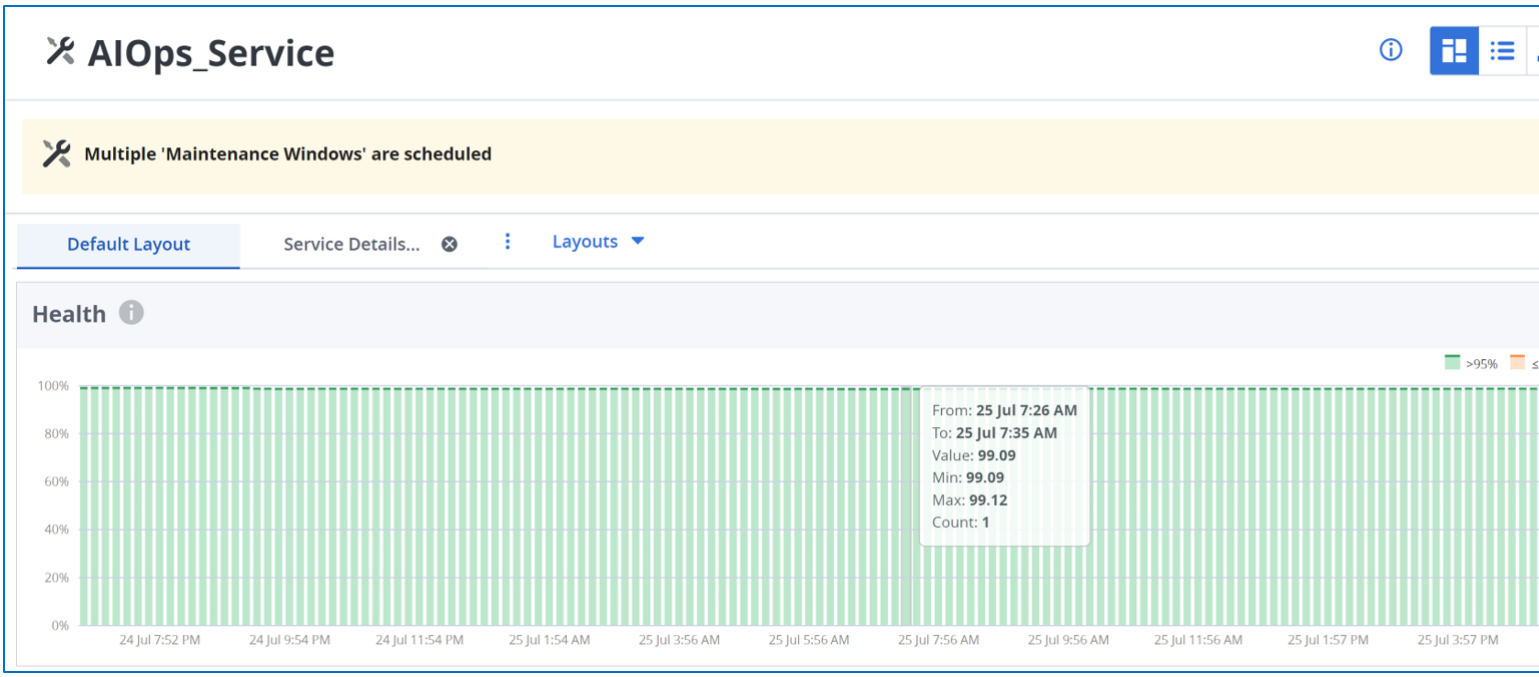
You can launch Capacity Analytics with the service filter that is applied by clicking **Capacity Planning** on the widget.



The following video explains how to view the capacity issues of a selected service:

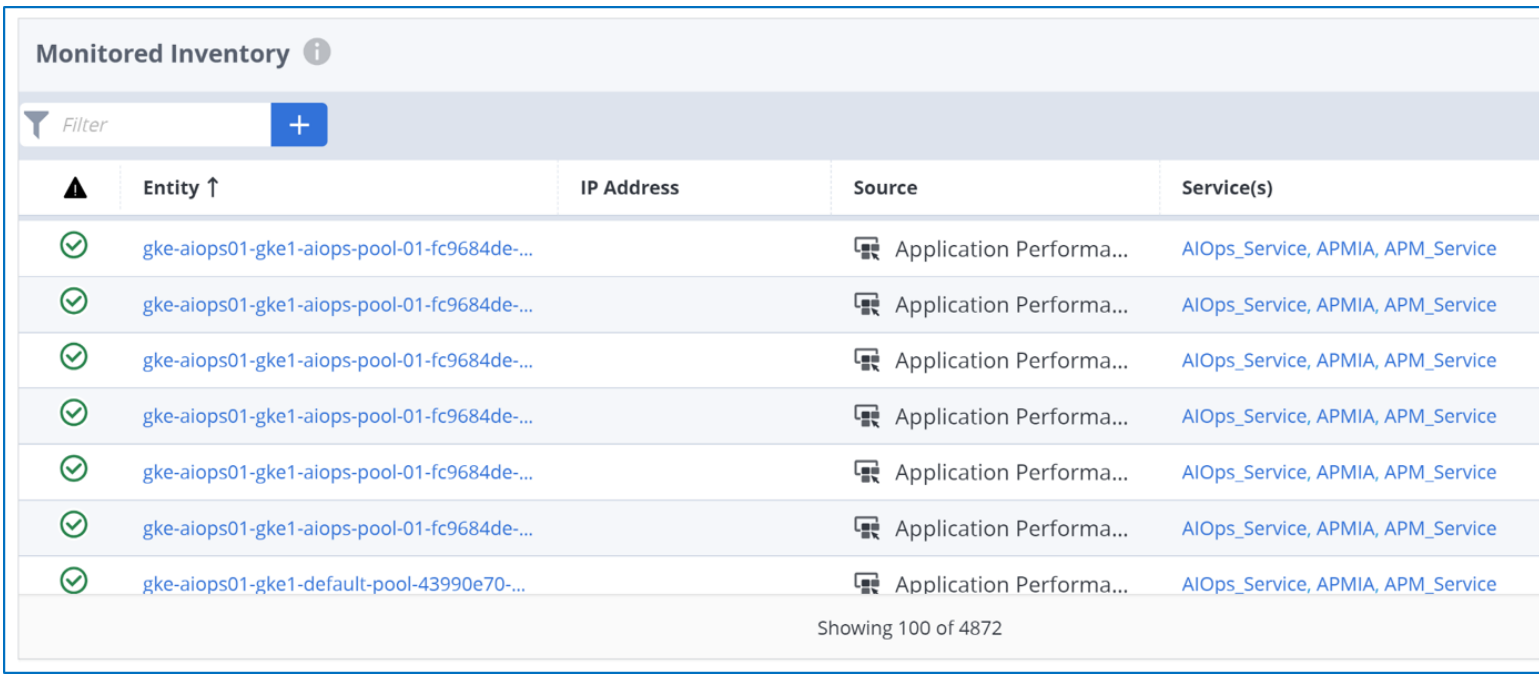
Health

The Health widget indicates the percentage of devices that are running normally and operational within the service. The health of service is calculated based on the number of available CIs with the total number of CIs.



Monitored Inventory

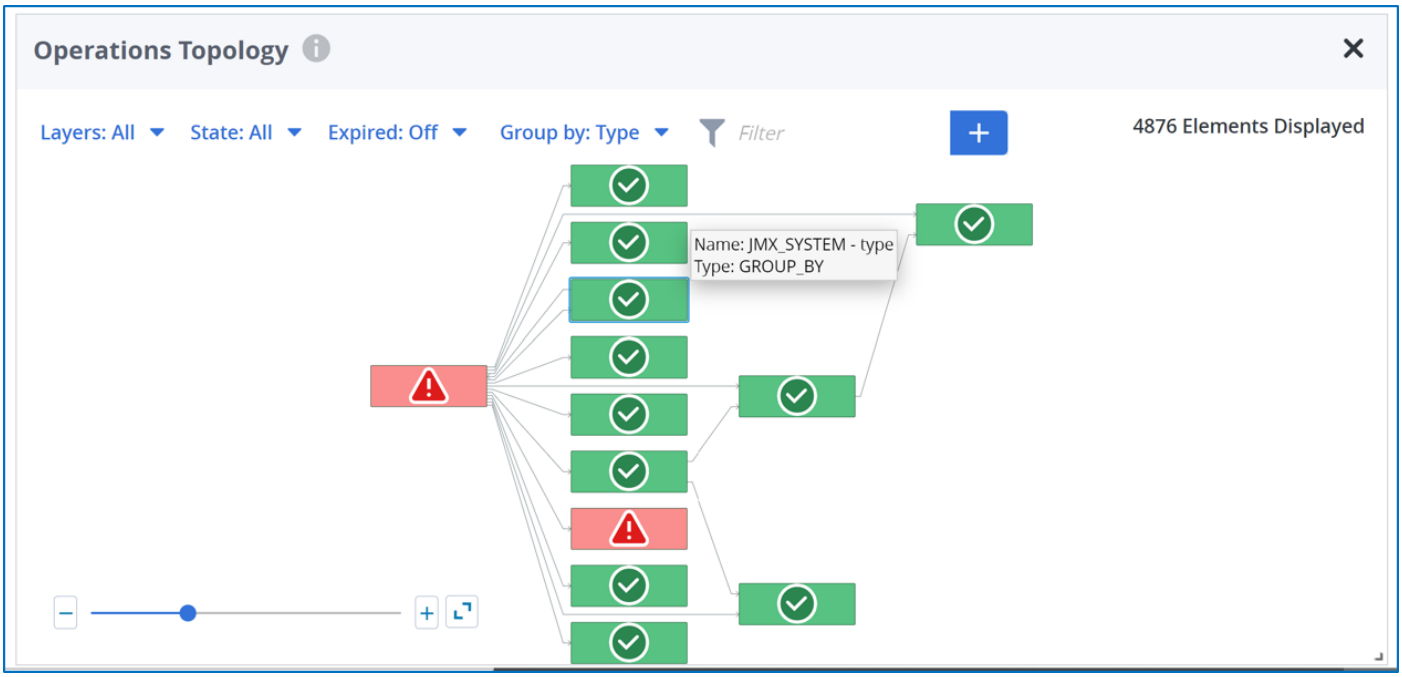
The Monitored Inventory widget provides the unified view of all entities for the selected service. Click Monitored Inventory in the widget to navigate to the Monitored Inventory page with service and filters applied. Click any entity to view the details only for that entity.



For more information, see the Monitored Inventory View section.

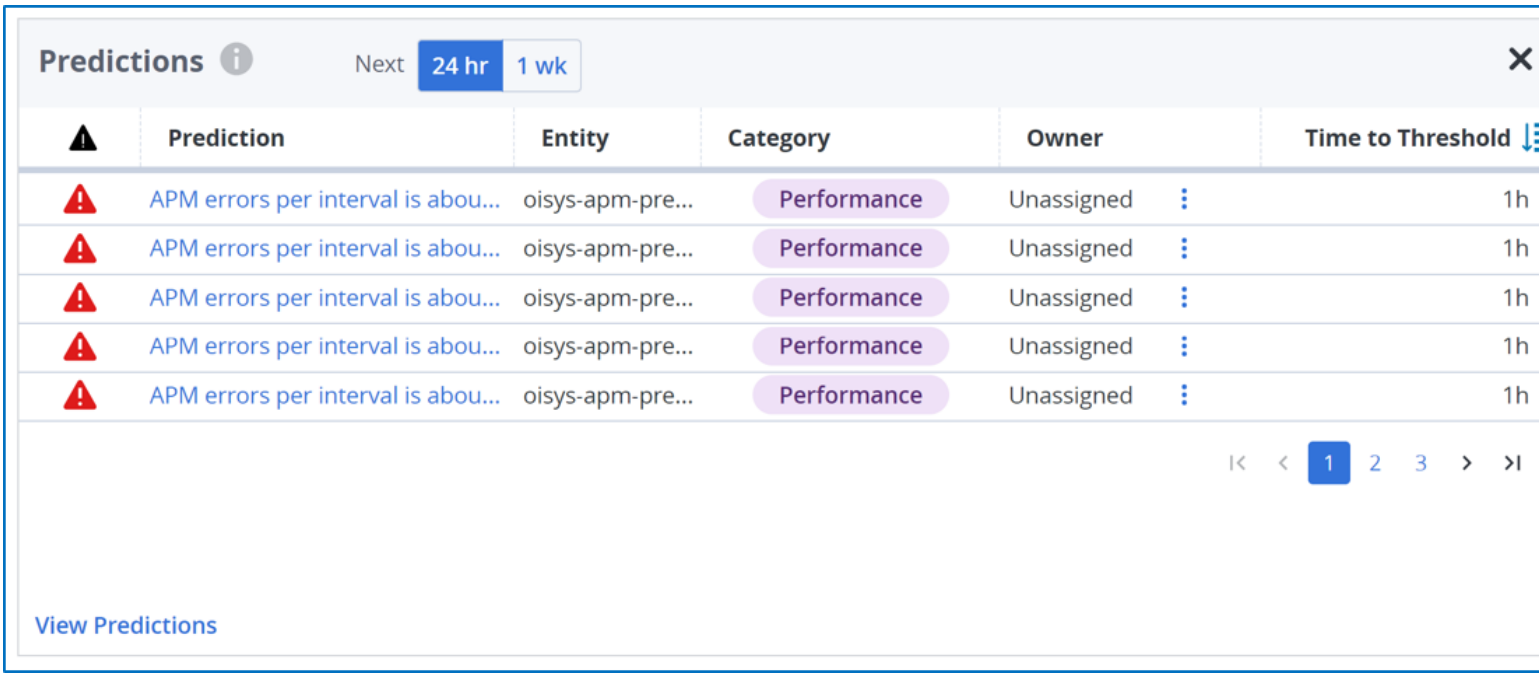
Operations Topology

The Operations Topology widget displays the topology of elements for service and its sub-services. You can select the layer visibility, state, expired, grouping, and filter the topology based on attributes. For more information, see View Topology Details.



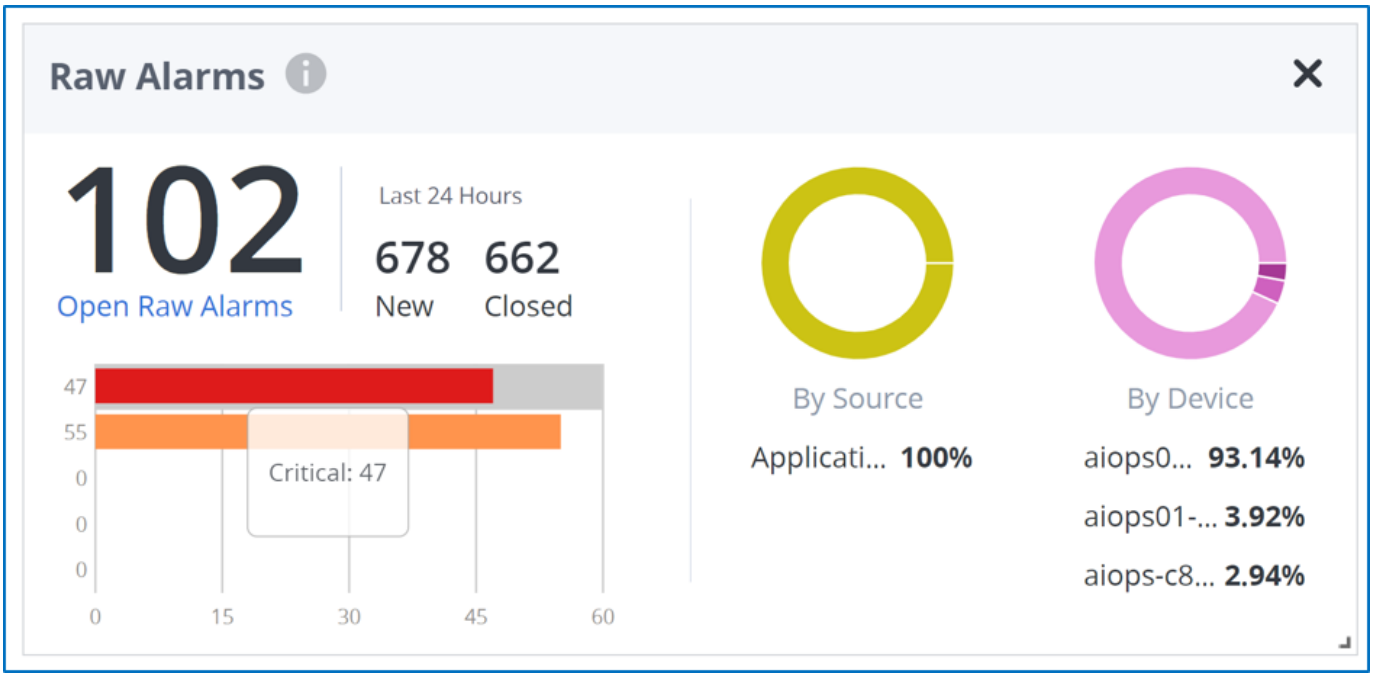
Predictions

The Predictions widget displays the predictions for the service to understand when an issue might impact the service. You can view predictions for the next 24 hours or one week.



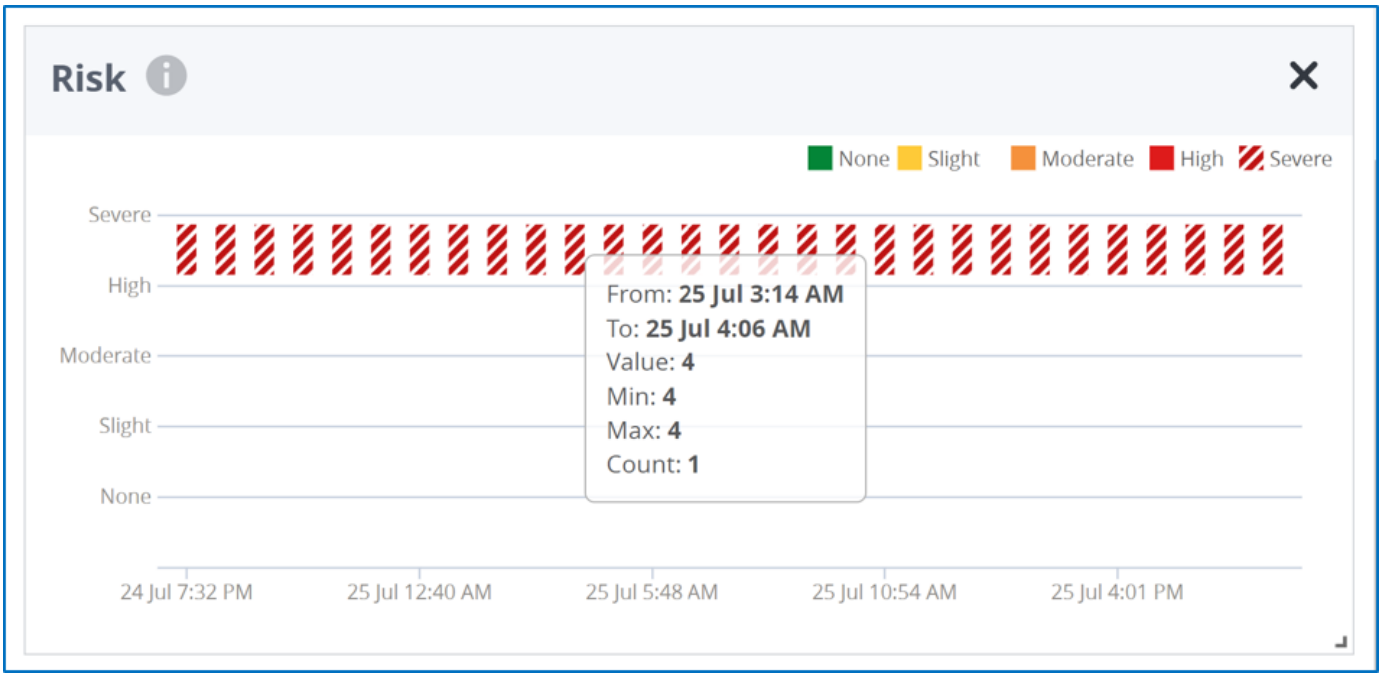
Raw Alarms

The Raw Alarms widget displays the overview of the current raw alarms that are open for service. You can also launch the All Alarms page in the context of the service. You can also view the raw alarms that are generated from the source product and generated from the device.



Risk

The Risk widget displays the probability of the service going down. Risk is defined as the risk of service becoming unavailable soon unless any action is taken on the alarms and configuration items that are associated with that service. The risk severity value ranges from severe to normal.

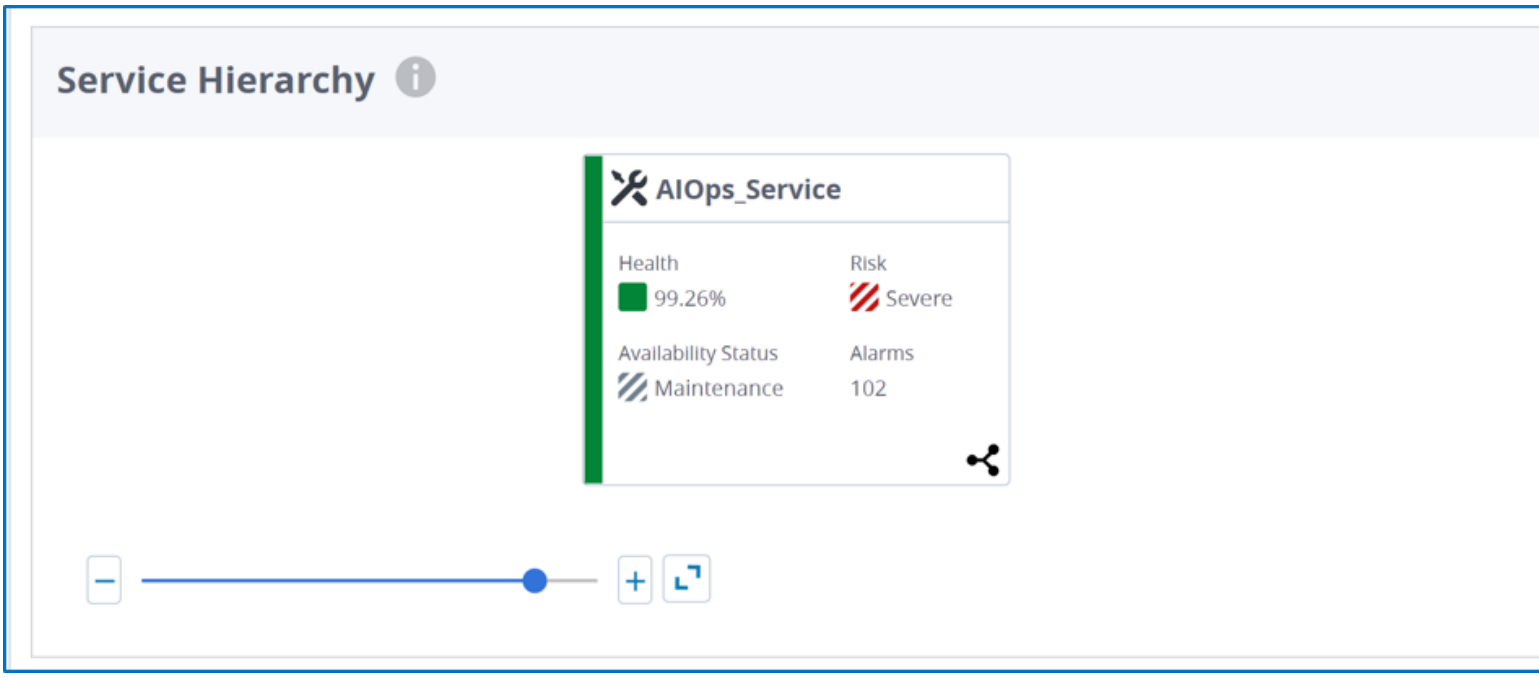


Risk is determined as follows:

- None: 0
- Slight: 1
- Moderate: 2
- High: 3
- Severe: 4

Service Hierarchy

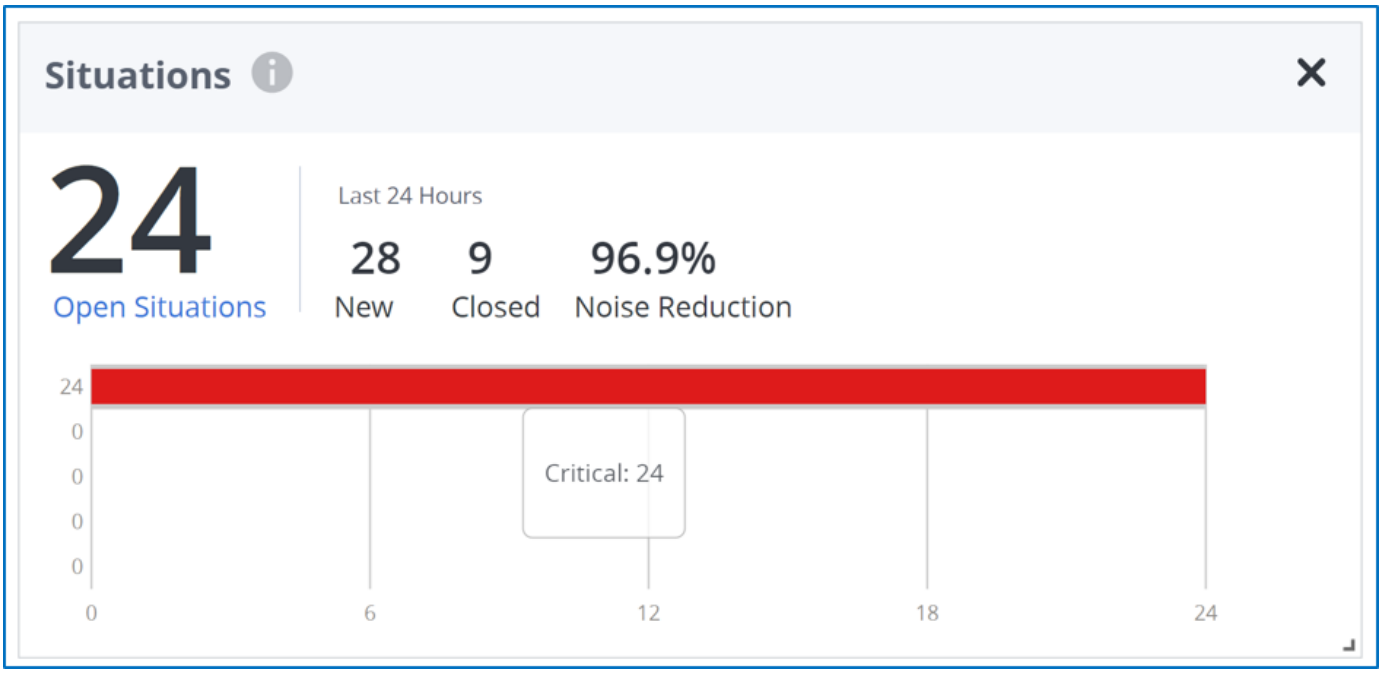
The Service Hierarchy widget indicates the hierarchical view of the relationship between the service and its sub-services.



For the Health, Risk, and Availability widgets, you can view the current value at the top of the graph and the average value on the graph based on the selected time period.

Situations

The Situations widget provides the overview of the current situations alarms that are open for service including noise reduction. You can also launch the Situations page in the context of the service.



Based on a fixed time range, you can view the following information:

- The total number of open alarms by severity.
- The total number of open situations by severity.
- Total number of entities impacted by **open** situations
- Open noise reduction by percentage.
- Total number of **new** situations that were raised in the last 1 hour.
- The total number of new situations raised in the last 24 hours.

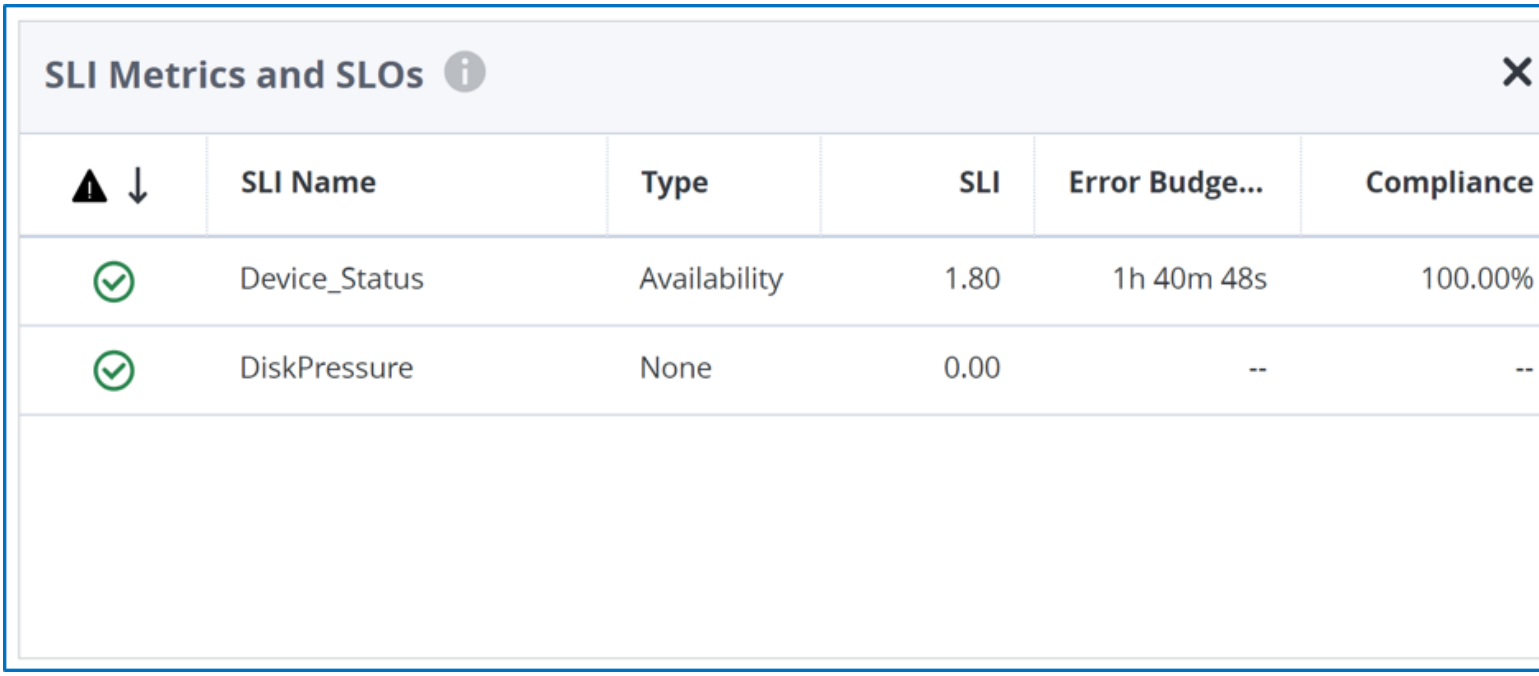
Based on the custom time-date picker, you can view the following information:

- The total number of open alarms by severity.
- The total number of open situations by severity.
- The total number of closed alarms by severity.
- The total number of closed situations by severity.

> **NOTE**
> For more information, see Situation Alarms

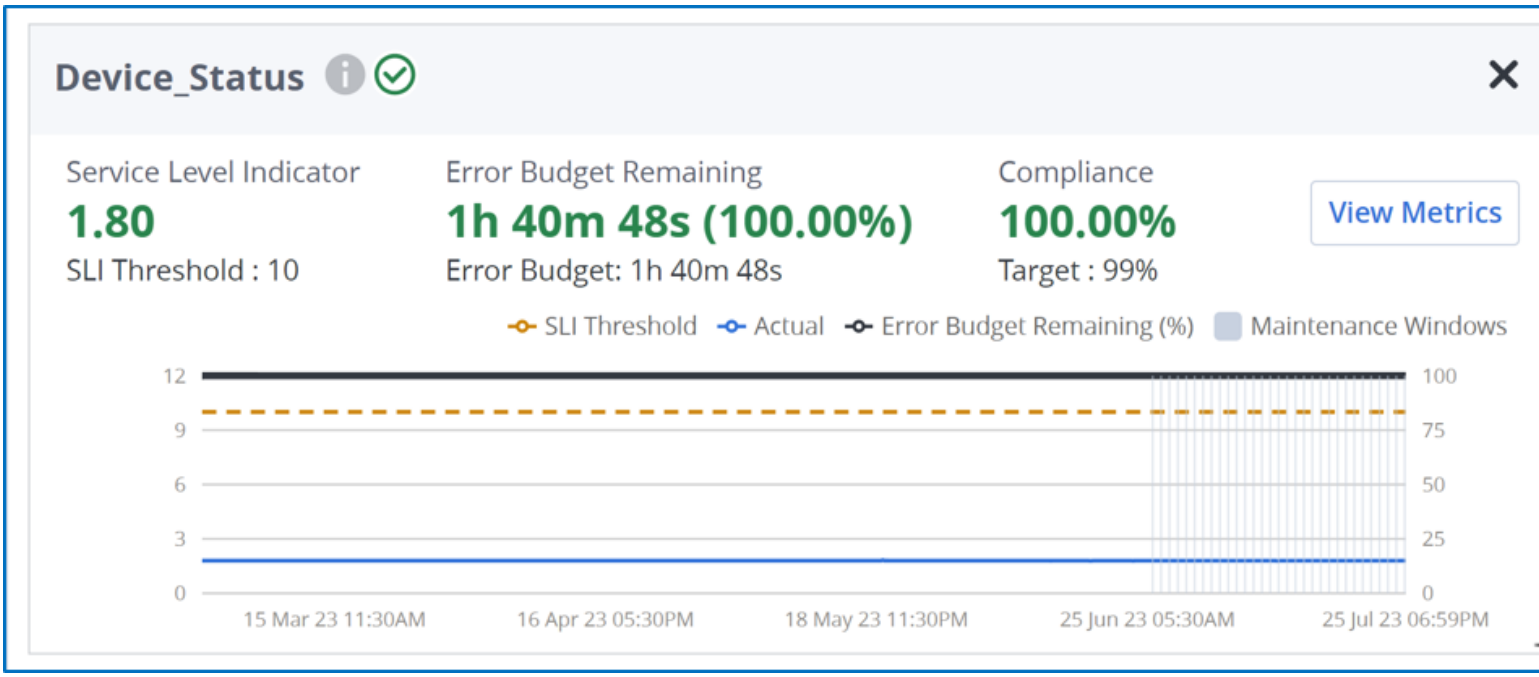
SLI Metrics and SLOs

The SLI Metrics and SLOs' widget show the list of SLI created for the Service. You can view the SLI name, type, SLI, compliance, and so on. You can also click the Fix button to fix the error for the SLI.



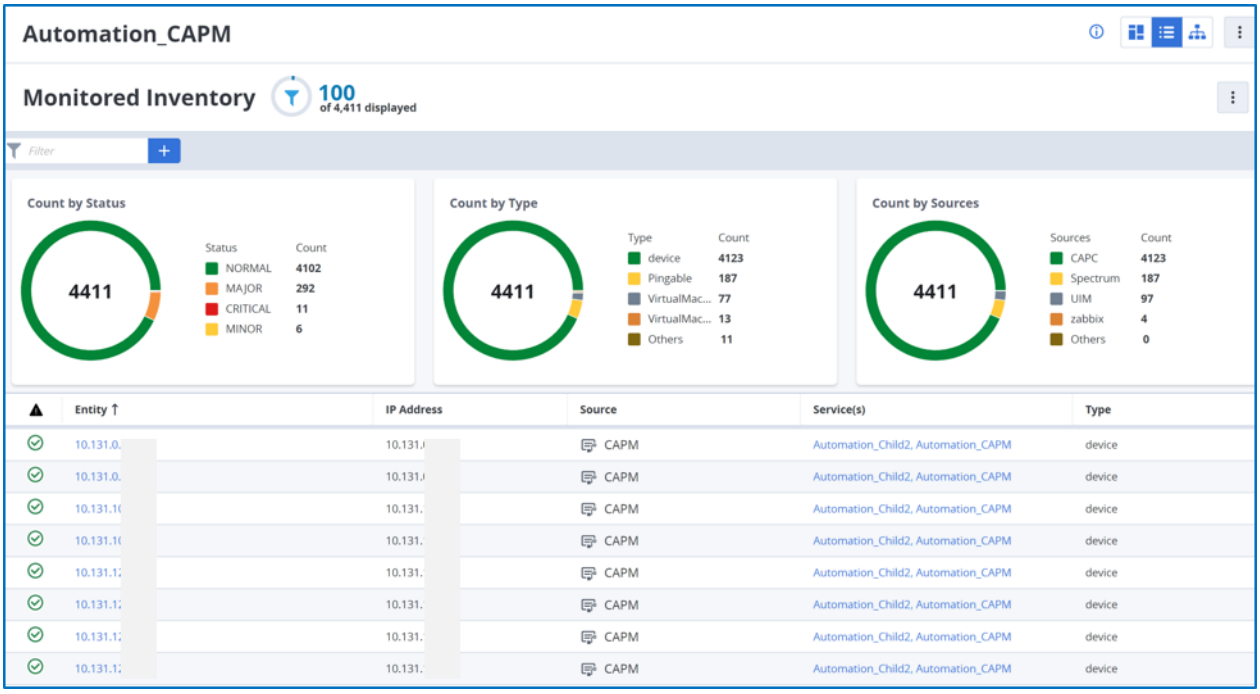
Service Level Indicators

The Service Level Indicators widget provides an overview of each SLI indicator that is defined for the service. The widget provides the SLI Threshold, Error Budget Remaining, and Compliance graph based on the selected date and time.



Monitored Inventory View

The Monitored Inventory View is a unified view of all entities for the selected service. Click the icon to view the list of entities for a service. The following image illustrates the Monitored Inventory View for a service.



Use the **filter** to search for your entities and sort the columns using the **Custom Sorting** option. You can navigate to the Service Analytics overview page, Performance Analytics, Alarm Analytics, and Capacity Analytics in the context of an entity from this table. In the **Type** column, hover over the value to display the icons.

You can also view the Monitored Inventory in the widget view. In the **Charts View**, click **Customize View** and enable **Monitored Inventory**.

> **NOTE**
> For more information, see the Monitored Inventory section.

{"URL":["https://digital-oi/service-analytics/serviceEdition/*/*/*/0/monitoredInventory"],"customLabelGetStarted":"Monitored Inventory View","description":"concept.dita_2cf11c68-281c-4470-9dc3-7e450014d49c","customCards":[{"type":"configure","id":"concept.dita_d55ce9c4-70af-471b-8ed0-428267fa8890","url":"https://techdocs.broadcom.com/us/en/ca-enterprise-software/it-operations-management/dx-operational-intelligence-saas/SaaS/using/Service-Analytics/Services-User-Interface/Service-Details-Page/Service-Details-Layout/Add-Maintenance-Window.html","title":"Add Maintenance Window"}]}

Add Maintenance Window

You can schedule a maintenance period for a required service. During the maintenance window, alarms are silenced and you will not receive any alarm notifications.

You can still monitor all the metrics such as risk, health, and availability of the devices. An alarm that raises within the maintenance time is tagged as in the Service Analytics view. You can create a maintenance schedule for once, daily, weekly, or monthly time period. You are allowed to create multiple maintenance windows.

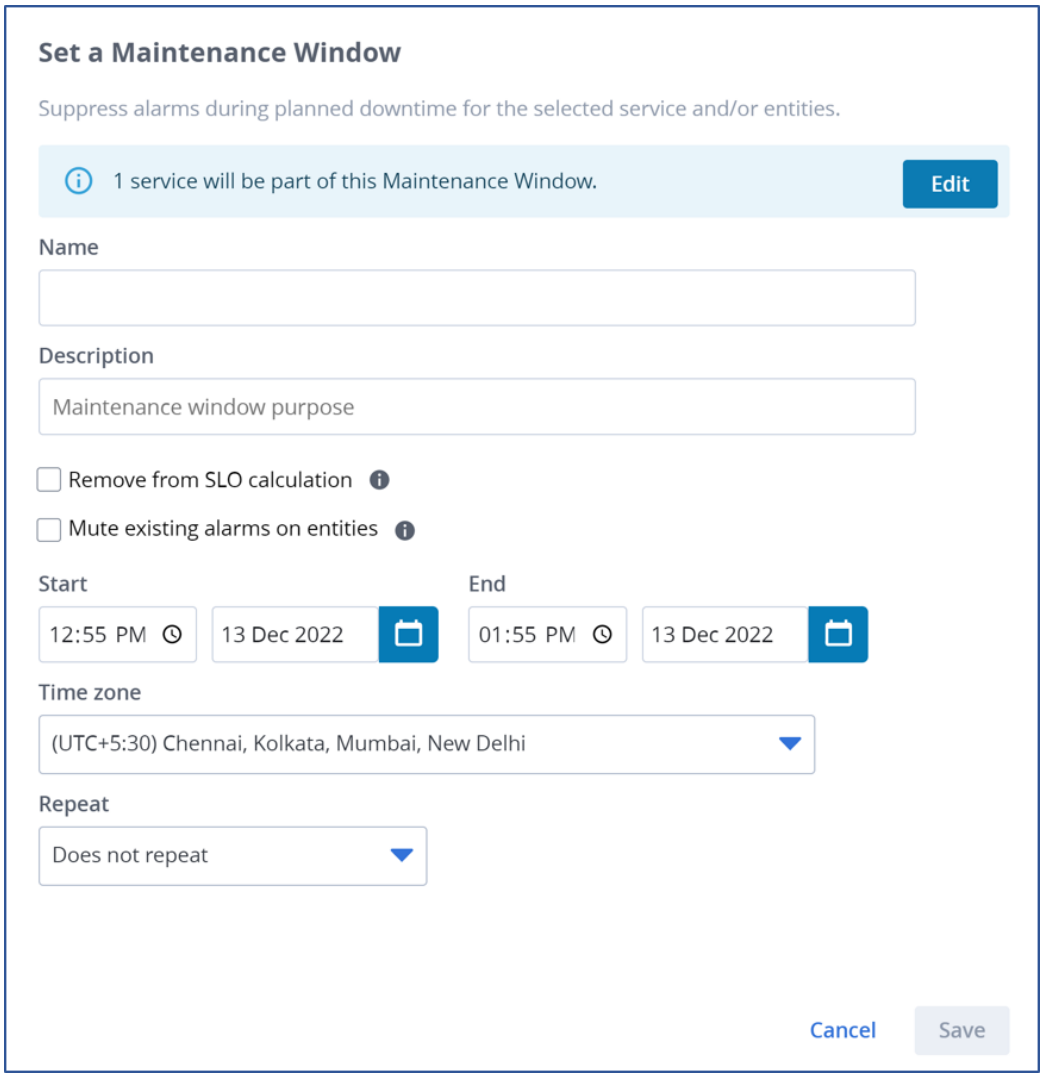
{"URL":["https://digital-oi/service-analytics/*/Services"],"customLabelGetStarted":"Add Maintenance Window","description":"concept.dita_d55ce9c4-70af-471b-8ed0-428267fa8890"}

Enable Maintenance Window

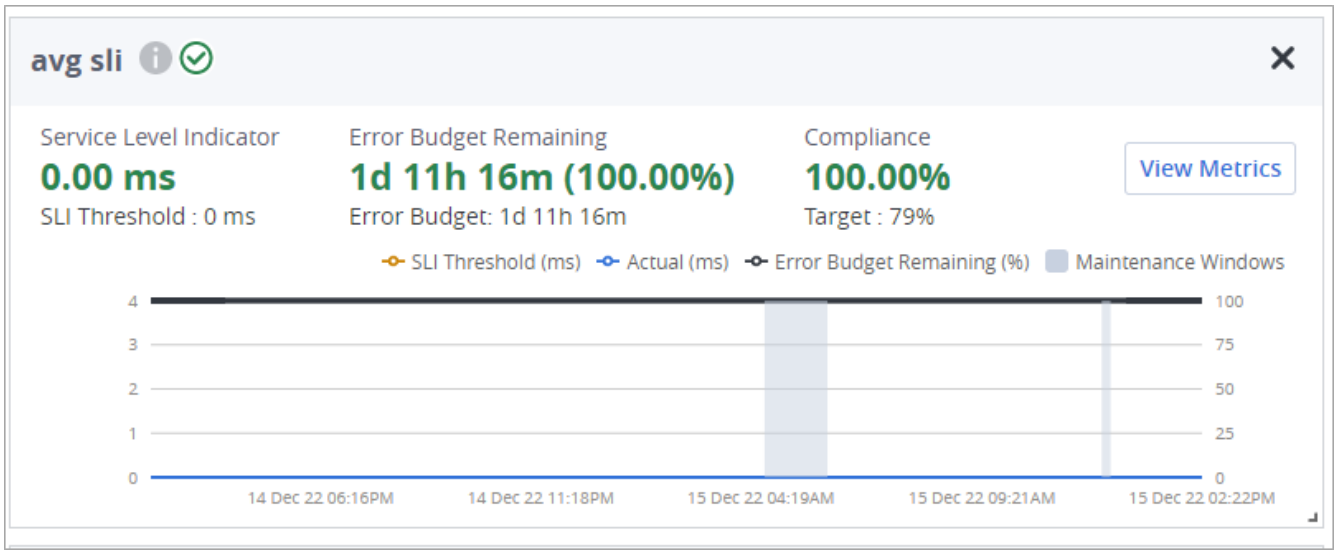
To enable the maintenance window, follow these steps:

1. Click the service for which you want to schedule the maintenance window on the **Services Overview** page.
2. Click the icon that is displayed in the top-right corner and click **Add Maintenance Window**.
   The **Choose affected entities** view opens with the **Services** tab as default.
3. Select the service based on the following selection options:
   - **Filters:** Add filters based on your requirements.
   - **Selection:**
     - **Specific:** By default, the **Specific** option is selected. This option enables you to select the list of entities with or without a filter applied.
     - **Based on Filter, Include Sub-Service:** These options are enabled after the filter is applied.
       - Select the **Based on Filter** option to select all the entities displayed on the page (with the filter criteria applied).
       - Select the **Include Sub-Service** option to select all the child services.

         > **NOTE**
         > You can also choose to add from **Agents**, **Groups**, **Entities**, and **Alarms** tabs respectively.
4. Select the service.
5. To create a maintenance window for only service, select **Add Service.** To create a maintenance window for the parent service and child sub-services, select **Add service & sub-services**.
6. (Optional) To view the selected entities, enable **Display Only Selected**.
7. Click **Continue**.
   The Set a Maintenance Window appears.



– **Name:** Enter a name for the schedule.
– **Description:** Enter a description for the schedule.
– **Remove from SLO calculation:** Select this option to exclude the SLO calculation during the maintenance window. If you select this option,
  - The SLO calculation is stopped, and only SLI is calculated during the maintenance period.
  - The Service Level Indicators widget displays this period in gray as shown:



Click the **Maintenance Windows** legend to clear the gray section on the chart. For more information, see the Add Service Widgets section.

- **Mute existing alarms on entities:** Select this option to mute the existing open alarms that were raised before or during an active maintenance period for the selected entities. When you select this option, all updates to the existing alarms which are part of this maintenance window are muted during the maintenance window and restored to their original state when the maintenance window ends. For example, the alarm is restored to the open state if the alarm was not closed during the maintenance period.

  During the maintenance period, the muted alarms are grayed out, and a maintenance icon is also displayed for those alarms. If multiple maintenance windows are scheduled at the same time on the same entity, and if this option is selected in at least one of the schedules, then the existing alarms for all the schedules are muted.

  NOTE
  If you end an active maintenance window or you delete a window, the alarms are restored to their original state.
- **Start** and **End**: Set the start and end times of the schedule. This field defines the duration of the schedule.
- **Time zone:** Select the time zone for the schedule.
- **Repeat:** Select the **Custom** option if you want a recurring maintenance schedule.
- **Every:** Set the time period based on Days, Weeks, or Months.
- **End:** Set the end time for the recurring maintenance window using the calendar. If you do not want to end the recurring maintenance window, select **Never**.

8. Click **Save.**
   The maintenance window is created for the selected service.
9. To modify an existing maintenance window, click the **Edit** option on the Maintenance Window message.
10. To delete an existing maintenance window, see the *Delete a Maintenance Window* section in Maintenance Window.

Topology Layout

The Topology layout is a graphical representation of the relationship among services and their sub-services

{"URL":["https://digital-oi/service-analytics/*/*/0/topology"],"customLabelGetStarted":"Topology Layout","description":"concept.dita_afcb07bd-b75a-4839-88d6-f2909f606c41","customCards":[{"type":"use","id":"concept.dita_ddc0cb72-3e71-4449-bc16-596cfbf9d371","url":"https://techdocs.broadcom.com/us/en/ca-enterprise-software/it-operations-management/dx-operational-intelligence-saas/SaaS/using/Service-Analytics/Services-User-Interface/Service-Details-Page/Service-Details-Layout/Topology-Layout/Alarms-for-Service.html","title":"View Alarms for Service"},{"type":"use","id":"task.dita_ed3552dd-1a8d-4e57-aa69-dd796f21d22e","url":"https://techdocs.broadcom.com/us/en/ca-enterprise-software/it-operations-management/dx-operational-intelligence-saas/SaaS/

using/Service-Analytics/Services-User-Interface/Service-Details-Page/Service-Details-Layout/Topology-Layout/View-Topology-Details.html","title":"View Topology Details"}]}

- Alarms for Service
- View Topology Details
- Topology Compaction
- Topology Traversal

The icons on the topology layout represent the object type, and arrows and the position of icons represent the relationships. To view the service detail in **Topology Layout**, click icon. The topology view provides the following functionality:

- By default, the Topology Layout displays the services and their sub-services in a high-level graphical representation.
- The deactivated services are grayed out.
- Use **Zoom-in** ( ) to see the details of the service. The service displays **Risk**, **Availability**, **Sub-service**, **Critical Service Alarms**, and **Prediction** metrics.
- Use **Zoom-out** ( ) to see the top-level tree structure.

  NOTE
  You can use the mouse wheel to zoom in and out.
- Use **Fit Content**( ) to fit the graphical representation within the screen.
- Move the topology up, down, right, or left by clicking the service and dragging it on the screen.
- Hover the mouse over Arrow or Service to see the severity of the alarm. This severity of alarm is indicated by the following color:
  - Red: Critical
  - Orange: Major
  - Yellow: Minor
  - Light Blue: Informational
  - Teal: Warning
  - Green: Good

  NOTE
  Hovering the mouse over the arrow indicates the connectivity between the service and its sub-services.
- View the weights of a sub-service for the impact calculation: **Zoom-in** to see the details of the service.
- Click the Topology icon to view the topology details for the service. For more information, see View Topology Details.
- Click the Alarms icon to view the list of alarms associated with the service. For more information, see Alarms for Service.

Alarms for Service

After you have selected a device in the topology layout, you can view the alarms for the service.

- The Alarms for Service displays the following details of the alarm.
  - Alarm Type
  - Alarm Message
  - Entity details
  - Service details
  - Source product
  - Ticket details
  - Owner details
  - Last updated details
  - Overview
  - Affected metric
  - Impacted services
  - Topology
  - Annotations

    NOTE
    For more information, see All Alarms.
- You can also view these details in Alarm Analytics by clicking

  icon.
- You can view the details of the alarm in the **Alarm List view.**By default, the Alarms for Service view opens in this

  view. The following are the list of alarm actions you can perform in the Alarm list view:
  - You can use the (bell) icon to manage alarms, acknowledge assigned alarms, and clear the assigned alarm. For more information on Alarms, see Alarm Analytics.
- You can change the alarm view by clicking

  . You can filter alarms based on the following category:
  - Situations
  - Service alarms
  - All alarms (default view)
  - Auto-update view
  - Show Maintenance alarms
  - Show closed alarms - If you want to view closed alarms, enable **Show closed alarms** switch in the

    menu. The closed alarms appear disabled with a grey text. You cannot perform any action on these closed alarms. You can also view closed alarm in the **Timeline** Tab. Make sure that you enable the Show closed alarms switch in the

menu to view the **Close time** column, which contains the details of when the alarms were closed. By default, the **Close time** column is not enabled.

– Export to Excel - You can export the alarms to an excel file. To export specific alarms, select the required alarm and click **Export**. If you do not select any alarm and click export, all alarms are exported to an excel fi

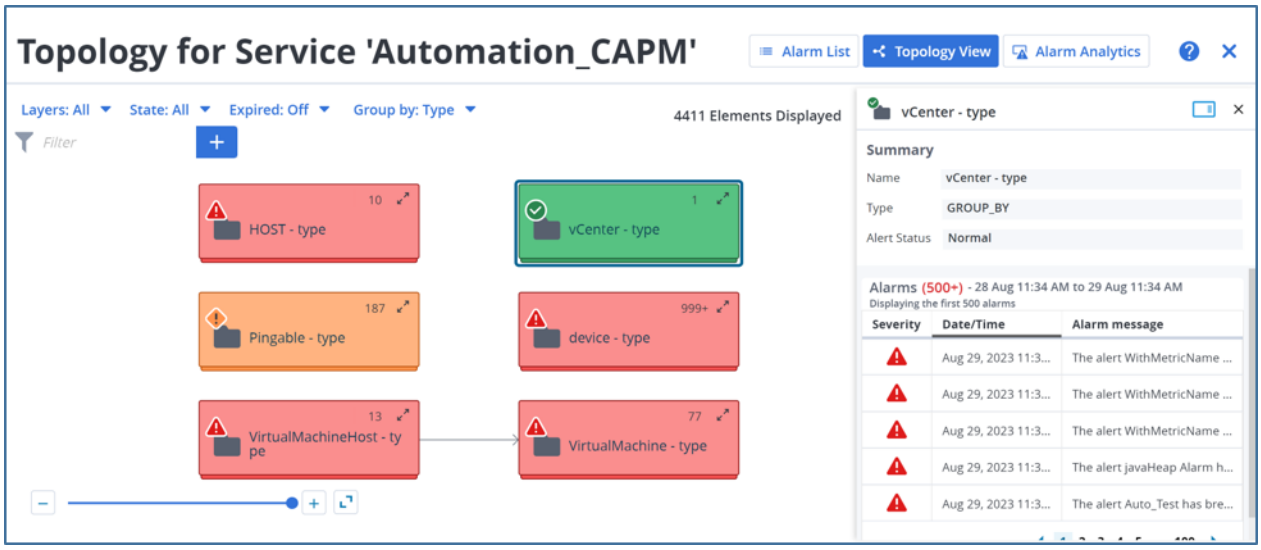
View Topology Details

The **Topology for Service** page displays the topology layout with a list of vertices and edges. The Topology page provides you with the topology hierarchy, that is, the associated elements, including the parent and child elements. You can view several details of each element such as Health, Availability status, Topology hierarchy, Alarm details, Number of users, Prediction details, and Weights.

> **NOTE**
> The sub-service would not move under the parent service automatically while using a Shared service. You must assign or move the Shared service under the parent service by using the drag-and-drop option. Shared service is represented by the
>
> icon.

To view the Topology of service, perform the following steps:

1. Select any service on the **Service Overview** page.
   The Service Details page appears.
2. Click the **Topology View** icon that is displayed in the top-right corner.
   The topology layout for the service is displayed.
3. Click the **Topology** icon that is displayed on the layout.



You can group the operational topology views using the following options:

- **Layers:** Allows you to select the required layer to be visible on the topology view screen. If there is only one layer in the view, the Layers option is not available. Select the required layer (Agent, Infrastructure, Network) to be displayed and click **Apply**.
- **State:** Allows you to select all vertices or vertices with only alarms.
- **Expired:**Allows you to view expired vertices. You can select the days ranging from one day to one week to view the expired vertices. The expired vertices appear in gray color with the expired date.

  NOTE
  By default, the Expired is set to Off.
- **Group By:** Allows you to select the views to be displayed on the topology view screen. The following two groups are available:
  - **Off:** Displays all the elements that are based on the selected Layers.
  - **Default Groupings:** The default groupings are out-of-the-box groups, which cannot be edited or deleted. You can group the topology by the following groupings:
    - Applications
    - Kubernetes Cluster
    - Kubernetes Project
    - Type (Default)
- **Custom Grouping**: The custom grouping allows you to create and save your own groupings.
  a. Click **Custom Grouping** from the **Group By** drop-down list.
  b. Enter the following details:
     - **(Optional) Name:** Set the name for your own grouping.

       NOTE
       If the Name field is left blank, grouping is applied without saving it for later use.
     - **Make Public:** Select this option to make the grouping visible to other users.
     - **Attributes:** Select the required attributes for grouping the elements.
  c. Click **Apply**. The custom grouping is added to the **Saved Groupings** list.

4. You can also view the summary of the service by clicking the service from the topology view.
5. Click on a device in the Topology hierarchy view to see the **Summary** of the device, and Alarms of the device.
   - **Summary:** Provides details of the elements such as Name, Type, Alert status.
   - **Alarms:** Provides alarm details such as Severity, Date/Time, and Alarm message.

After you view the topology, return to the **Add Elements** page.

Topology Compaction

Service Analytics supports Topology Compaction which enables you to compress several hosts under one common host or entity. Topology Compaction helps you reduce the data redundancy and view additional components in the same view. The Topology process is performed using various details such as Hostname, IP address, MAC address, and so on. For example, you have a parent service created with three sub-services. Each of the sub-services is created with different dimensions.

Let's consider the VMs sub-service which is created from a single UIM Group and the other sub-service from APM's Kubernetes. If you click on the Topology icon of the sub-service and remove all the filters. You can see three devices which are Hosts. You can use the drop-down option on the sub-service to expand and view all the associated nodes or devices that come under this sub-service. You can use the collapse option on the sub-service to collapse all the details and return to the original view.

Topology Traversal

Service Analytics supports one-hop traversal, which enables you to find all the applications and infrastructure components that are one hop away from the required node or device.

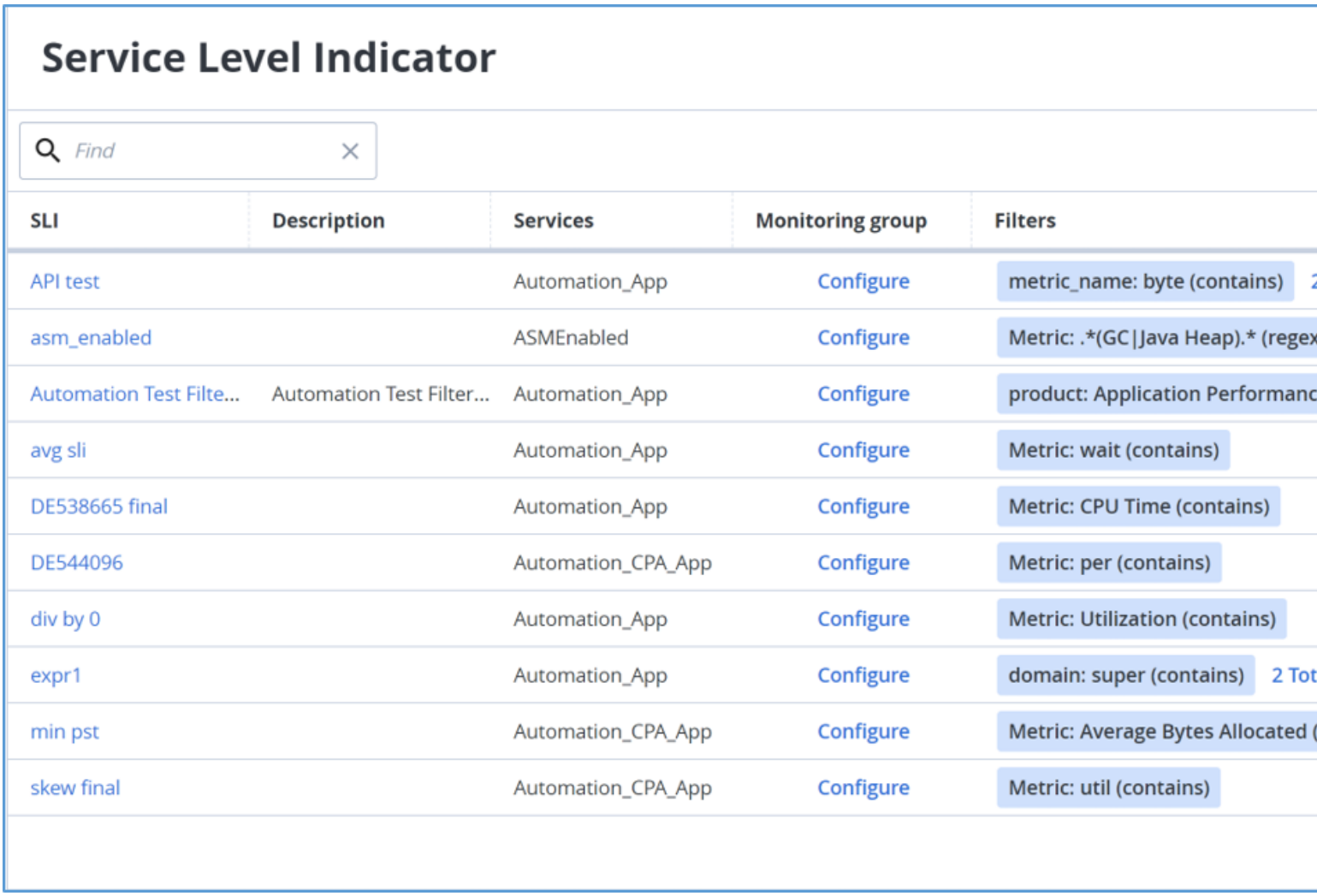
You can view all the traversals from a particular host or node by just hovering the pointer over it. The following illustration depicts the use of Topology

Traversal:

Service Level Indicator and Service Level Objective

DX Operational Intelligence automatically discovers the CIs that belongs to the service and monitors the health and risk of your services as Service Level Indicators (SLI). You can add more visibility to your service performance through the addition of SLIs and Service Level Objectives (SLOs). By using SLIs and SLOs, you can increase the performance and reliability of your services through metric-driven service-level visibility.

The following image illustrates the SLI home page:



This page provides the following information:

- SLI Name
- Description
- Services
- Monitoring Group
- Filters
- Created
- Last Editor
- last Modified

{"URL":["https://digital-oi/settings/digital-oi/settings/service-level-indicators"],"customLabelGetStarted":"Get Started with SLI/SLO","description":"concept.dita_752ef12c-f70a-4b6a-9b15-b9eabad02576"}

Service Level Indicator (SLI)

Service Level Indicator is a metric that describes an aspect of the health of a service. A good SLI is one that describes the symptoms of a service that is being delivered such as Availability or Response Time. You can add SLI to your existing

services by selecting the metric (or set of metrics), aggregation, and measurement interval. SLI form the foundation of Service Level Objectives.

Service Level Objective (SLO)

Service Level Objectives (SLO) help you to understand performance based on an objective (threshold) during a specific time (window). In DX Operational Intelligence, a Service Level Objective is a Service Level Indicator with a threshold and a time window applied. With SLO used, you can measure and track actual outcomes against defined criteria to improve processes and avoid the slow deterioration of service experience.

The SLO has two key metrics to describe the performance:

- **Compliance -** The percentage of time when your service met the criteria for performance.
- **Error Budget -** In a given time, the number of errors accumulated before exceeding the service objective.

Applying Compliance and Error Budgets in your service health reporting lets, you understand how the service has performed historically and how much budget you have left to meet the service performance goals. With these metrics, you can improve your service experience and reliability. You can calculate the compliance and error budget metrics using the rolling window or a calendar window (week, month, quarter, and year)

Rolling window: You can measure the SLO metrics using the rolling window. The rolling period uses the First in, Last out method. The oldest data in the previous calculation drops out of the current calculation, and new data replaces it.

Calendar window: The Calendar window helps you calculate the SLO based on the boundary and not start the calculation randomly. The calendar window is easier to manage and report. You can calculate the SLO metrics based on week, month, quarter, and year.

- **Week:** The evaluation begins on Sunday and ends on Saturday.
- **Month:** The evaluation begins on the first day of the month and ends on the last day of the month.
- **Quarter:** The evaluation begins on the first day of the month and ends on the last day of the quarter. For example, if the SLO metrics evaluation starts on January 1st, then it ends on March 31st.
- **Year:** The calculation starts on the first day of the year and ends on the last day of the year.

Enable the SLIs & SLOs Tile

Perform the following steps to enable the **SLIs & SLOs** tile on the **Settings page** in DX Operational Intelligence.

Follow these steps:

1. Log in to Cluster Management (**http://apmservices-gateway.<defaultSubDomain>/dxiportal**).
2. Access **Cluster Settings** from the left navigation pane in the Cluster Management console.
3. Add the **oi.tenant.sli_enabled** property with the value as **true**.
4. Log out and log back into the tenant.

Create SLIs and SLOs

{"URL":["https://digital-oi/settings/digital-oi/settings/service-level-indicators/monitoring"],"customLabelGetStarted":"Create SLI/SLO","description":"task.dita_fefc1818-2126-409e-b645-74b12ccce981"}

You can create and customize the metric to a service using multiple metric filters, intermediate derived metrics, and arithmetic expressions. Create an SLI by following this process:

- **Primary Filter:** This is the first level for the metric pipeline to select the set of metrics that must participate in the subsequently derived aggregations, thresholds, and expressions.
- **Secondary Filter:** Use the filtered metric set and select the set of metrics to perform a derived calculation (Aggregation, threshold).
- **Arithmetic Expression:** Reuse SLI or Derived metrics in the pipeline and apply arithmetic.

> **NOTE**
> **Note:** You can include up to 50k metrics for an SLI.

To create a service level indicator, follow these steps:

1. Navigate to the **Settings** page in **DX Operational Intelligence** UI.
2. Click the **Create SLIs & SLOs** tile.

   The Service Level Indicator page appears with a list of created SLIs. You can select an SLI and click a metric to view the metric chart.

   > **NOTE**
   > You can edit the existing SLI. Select the required metric and update it.

3. Click **Create SLI/SLO** button.
4. Select the Service from the service list.

   > **NOTE**
   > **Note:** You can create up to 15 SLIs for a service.

5. Apply the metric filter (also known as the Primary Filter) to configure the service level indicator. You can filter by source, metric, metadata or custom attributes. This is the first level for the metric pipeline to select the set of metrics that must participate in the subsequently derived aggregations, thresholds, and expressions.

   A list of metrics appears based on the filter applied. You can click a metric to view the metric

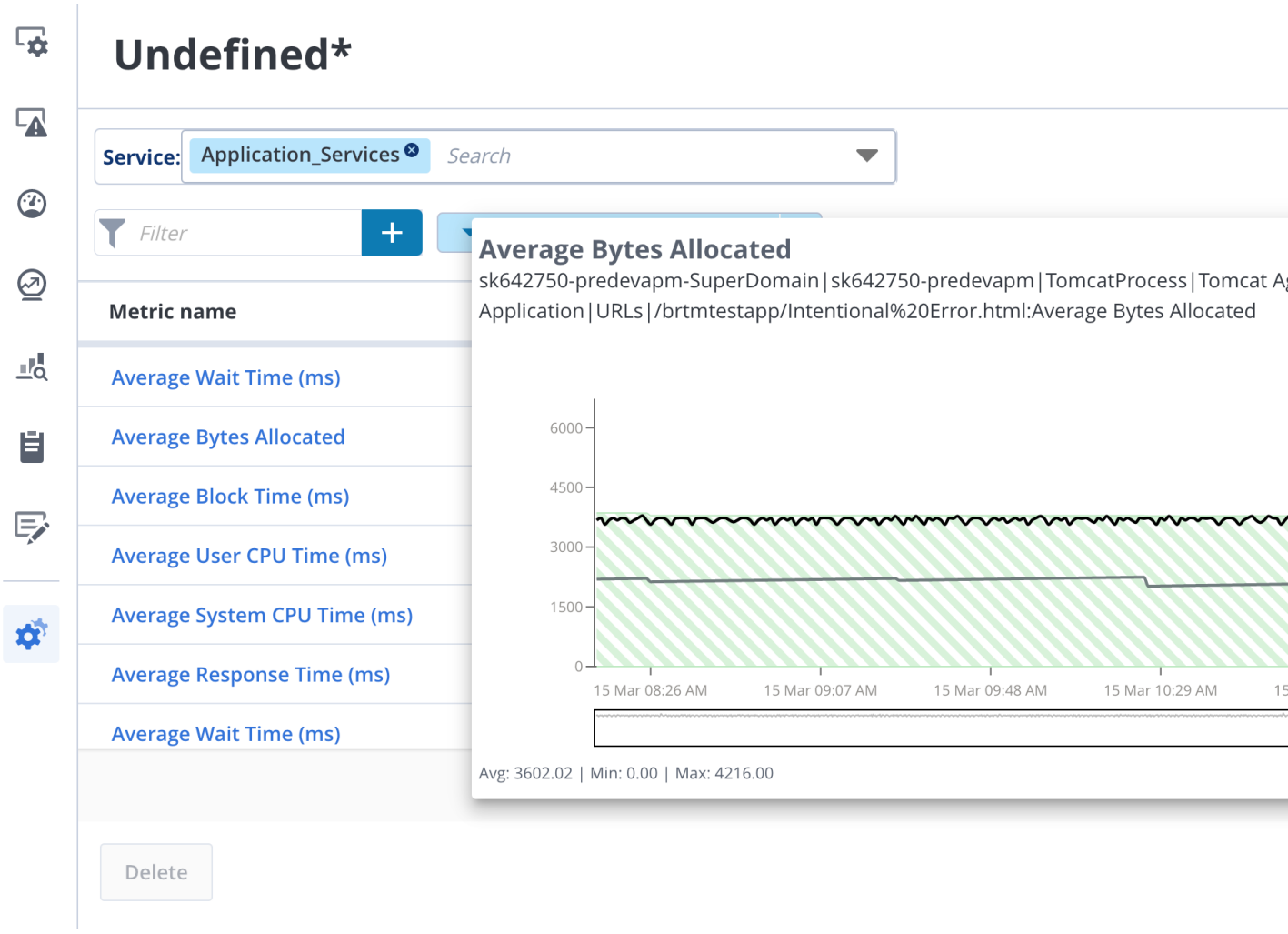


   chart.

6. **Define SLIs.** Click **+ Add Service Level Indicator.**
   a) **Publish:** You must publish the SLI for the aggregation to work.
   b) **Display in Views:** Enable this switch if you want the published SLI to appear in the views. This option is enabled only when SLI is published.
   c) Provide the following information:
      - **SLI Name:** Provide the SLI name for the service.

        NOTE
        To avoid incorrect results, choose the SLI metric name such that it does not match the input metrics filter.
      - **SLI Type:**

  - **Availability:** Choose this metric to set the SLI as an Availability metric for the selected services.
  - **Error:** Defines the amount or percentage of failed requests to a particular resource.
  - **Latency:** Defines the time period for the application to respond.
  - **None**
- **Aggregation:** Set the following aggregation type to aggregate the metrics.
  - **Average:** This function finds the weighted average of the input values and sets it to the output metric value.
  - **Boolean And:** This function applies boolean AND on the input values and sets it to the output metric value.
  - **Boolean Or:** This function applies boolean OR on the input values and sets it to the output metric value.
  - **Count:** This function provides the aggregated count of raw input metrics when the source is data.
  - **Expression:** Define the custom expression using the selected SLI. The expression can be any mathematical operation. You can provide the expression value in the **Custom Expression** field. Use **${pipeline:sli_as_value}** to denote a variable. For example, **${pipeline:sli_1}*(${pipeline:sli_2}+ ${pipeline:sli_3})**. Let us assume there are raw metrics on SLI1, and you want to add 50 in SLI2, then the expression can be written as **(${pipeline:SLI1}+50)**.

    NOTE
    DX Operational Intelligence supports ternary or conditional expressions for SLIs and SLOs.

    - ${pipeline:avg_metric1} >= ${pipeline:avg_metric2} ? Max(${pipeline:avg_metric1}, ${pipeline:avg_metric2}) : Min(${pipeline:avg_metric1} ,${pipeline:avg_metric2})
    - (${pipeline:avg_metric1} >= ${pipeline:avg_metric2} && ${pipeline:avg_metric2}>0) ? Max((${pipeline:avg_metric1} +${pipeline:avg_metric2}),10.0) : 0

    Any number in the expression should be a decimal. For example, (${pipeline:max}>= ${pipeline:min} && ${pipeline:min}>0)?Max((${pipeline:min}+${pipeline:max}),500.0):100.0.
  - **Forward:** This function is the alias of a raw metric, and you can use it as a pipeline metric (for expression function or any other pipeline function). This function forwards the raw metric into the pipeline with the same frequency as the raw metric and does not require any metric.
  - **Max:** This function finds the maximum of the input values and sets it to the output metric value.
  - **Min:** This function finds the minimum of the input values and sets it to the output metric value.
  - **Percentage:** This function works on top of boolean output and calculates the percentage of 0/1 in the incoming metrics.
  - **Sum:** This function sums the input values and sets them to the output metric value.
  - **Threshold:** This function sets the threshold conditions.

    NOTE
    Set the threshold conditions when selecting the aggregation type as the Threshold.
- **Aggregation Interval:** Set the aggregation interval for the SLI metrics data to be ingested. For example, if you set the time interval as one min, the SLI metric data is ingested every minute.

  NOTE
  The aggregation interval must be equal to or greater than the interval of the incoming raw metric (metric filter applied)
- **Unit:** Provide the unit for the SLI metric.

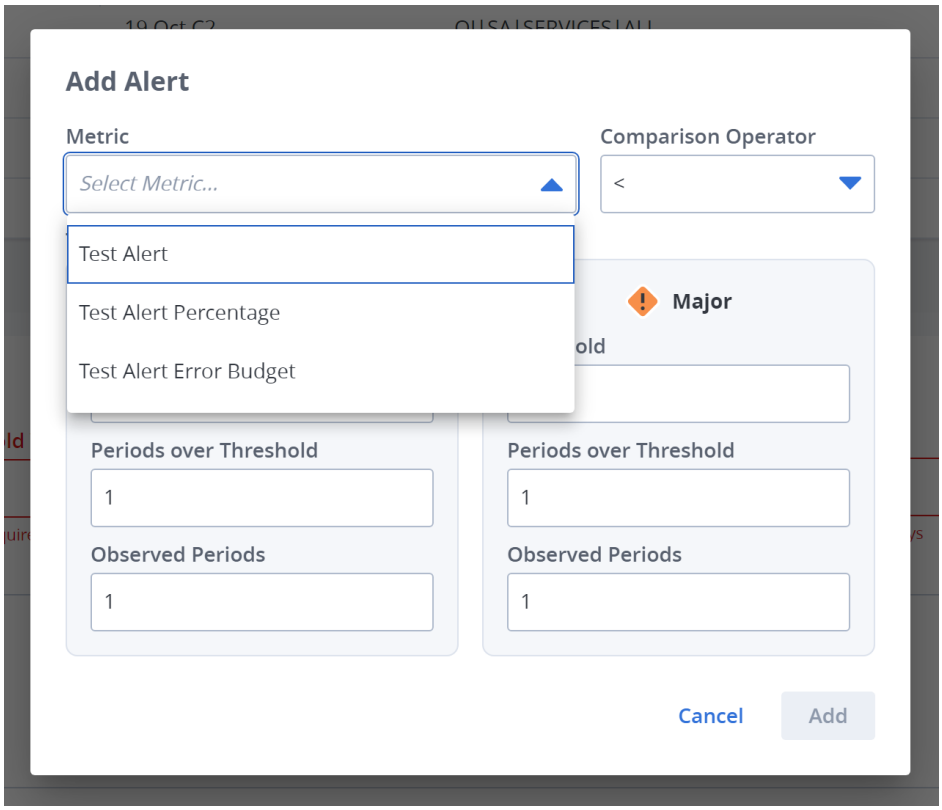
  NOTE
  A maximum of five characters is supported.
- **Skew:** Determines the wait time before emitting the output metric. Provide values in seconds.
- **As:** Specifies the alias name for the SLI and is case-sensitive. However, you can customize the name as required.
- **Description:** Provide the purpose of SLI.

7. **Select SLI Source:** Select the source for the SLI. You can select metrics with an optional secondary filter or other SLIs. Using the filtered metric set and select the set of metrics to perform a derived calculation (Aggregation, Threshold). This is the secondary filter.
   a) **Metrics**: Click **Refine metrics**: Filter the metrics further to add to the SLI.
   b) **Service Level Indicator:** Select the SLIs to use for this function.
8. **Enable SLI Threshold and SLO:** Enable this switch when you want to set the SLO for SLI.
   a) **SLI Threshold:** The name gets populated based on the SLI name.
   b) **Threshold Value:** Specifies the threshold value.
   c) **Service Level Object:** The percentage of time the SLI threshold condition is met to achieve the service objective. The threshold is calculated based on the compliance window.
   d) **Compliance Window:** Select the time window (Rolling or Calendar) to evaluate your SLO. The Rolling window uses the moving time range. For example, the aggregation interval of 5 mins creates windows of 10:00-10:05, 10:01:10:06. The Calendar window uses the fixed date. For example, the aggregation interval for monthly creates windows of Jan1:Jan31, Feb1:Feb28(29). For more information, see the Service Level Objective section.

   The SLO metrics (including Error Budget) honor the SLI creation time even when the SLO is enabled later on that SLI.
9. **Setup Alert Thresholds:** You can add an alert only for Critical and Major alarms. The alert thresholds can be set on SLI, SLO Percentage, and SLO Error Budget metrics.

   To display both the SLI and SLO metrics, ensure that the **Enable SLI Threshold and SLO** option is selected. If you do not select this option, then only the SLI metric is displayed.
   a) Select the **Setup Alert Thresholds** option and click **+ Add Alert** to open the **Add Alert** dialog.

      The Add Alert dialog opens.



b) Provide the following information:

- **Metric:** Select the metric.
- **Comparison Operator:** Select the operator.
- **Thresholds:** Enter the threshold values for the Critical and Major alarms.
- **Periods over Threshold:** Enter the number of occurrences to generate the alert.
- **Observed Periods:** Enter the number of periods within which the threshold should be reached to generate the alert.

For example, if the threshold is > 10, Periods Over is 2, and Observed Periods is 5 for a critical alarm, then a critical alarm is generated when 2 out of 5 occurrences have metric value > 10.

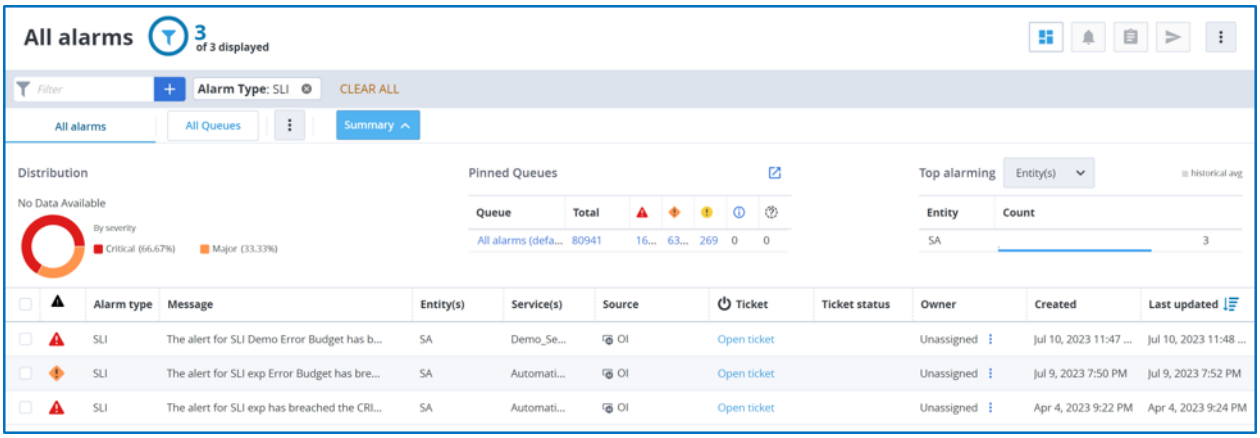
NOTE

- Observed Periods must always be equal to or greater than Periods over Threshold.
- After an existing alert is cleared, if the alert condition is met again, a new SLI alarm is not created. The alert is created only if the severity is changed.
- If the SLI name is changed or if the alert or the SLI is deleted, the existing alert does not get closed.
- If you add a service to the SLI after adding an alert, the SLI alarm is not generated for the added service.

c) Click **Add** to add the Alert Threshold. You can edit or delete the alerts if required.

10. Click **Save.**

The SLI is created for service and you can monitor the SLIs on the Service Details page. When the threshold condition is met, an alarm is generated, and you can view the details (Alarm Type is SLI, Source is OI) on the All Alarms page.

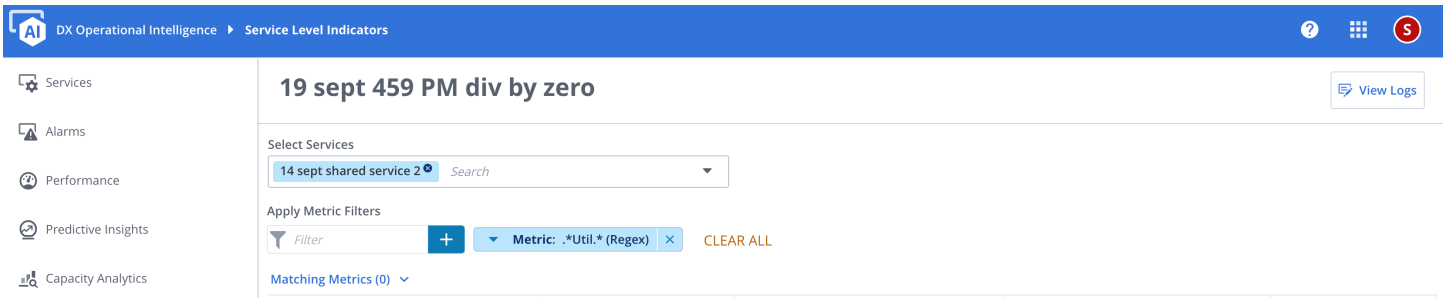


You can perform all the actions (alarm management, ticket management, trigger channel) like other alarms.

View Logs

You can view logs for the SLI pipeline errors (run time exception) using the **View Logs** option. For example, you can view logs for errors such as:

- Consecutive SLI is divided by 0
- The Aggregator fails to receive the required parameters from all the upstream functions due to incorrect skew value.



When you click the **View Logs** button, you navigate to the DX Dashboards page for the SLI that displays the error messages.

Configure Service Availability Using SLI and SLO

IT service availability can be measured and managed using Service Level Indicators (SLIs) and Service Level Objectives (SLOs). An SLI is a metric that measures some aspect of the performance or behavior of a service. SLIs can be quantitative (such as response time, throughput, or error rate) or qualitative (such as user satisfaction or system stability). An SLO is a target level of performance or behavior for a service, as defined by a specific SLI. SLOs are typically expressed as a percentage or a range of acceptable values for the SLI. For example, an SLO might specify that the response time for a service should be less than 500 milliseconds for 99.9% of requests.

By setting SLOs for important SLIs, IT teams can establish a clear set of expectations for the service and can ensure that it meets the needs of its users. SLOs also provide a way to measure the performance of the service over time and to identify areas where improvements can be made.

To ensure that a service meets its SLOs, IT teams must monitor the relevant SLIs and must take necessary action if they fall outside the acceptable range. This might involve tuning the service configuration, optimizing the code, or scaling up resources.

Overall, SLIs, and SLOs provide a way to measure and manage the availability of IT services and ensure that they are meeting the needs of their users. By monitoring SLIs and setting clear SLOs, IT teams can identify and address issues before they become major problems. They can also ensure that the service is always available and performing at an acceptable level.

This section provides the following information about how to use SLIs and SLOs to monitor the service availability:

- SLI and SLO Examples
- Configure Service Availability

SLI and SLO Examples

Suppose you are responsible to ensure the availability of a web-based ticket booking service and your team has decided to use SLIs and SLOs to monitor the service availability. Here are a few examples of SLIs and SLOs that you could use:

- **Successful Requests**
  - SLI: Successful Requests
  - SLO: 99.9% of all requests should result in a successful response within 3 seconds.

    This SLI measures the percentage of requests that are successfully processed by the ticket booking service. The SLO sets a target of 99.9% success rate for all requests, with a response time of 3 seconds or less. This means that the service should be able to handle a high volume of requests and respond quickly to them.
- **Error Rate**
  - SLI: Error Rate
  - SLO: The error rate should be less than 0.1% of all requests in a given week.

This SLI measures the percentage of requests that result in errors or failures. The SLO sets a target of less than 0.1% error rate for all requests in a given week. This means that the service should be reliable and should minimize the risk of errors or failures.

- **Uptime**
  - SLI: Uptime
  - SLO: The service should have at least 99.95% uptime in a given month.
    This SLI measures the percentage of time that the service is available and responsive to requests. The SLO sets a target of at least 99.95% uptime in a given month. This means that the service should be highly available and should minimize downtime or outages.

By monitoring these SLIs and SLOs, you can track the service availability and can take appropriate actions to address any issues or incidents that may arise.

Configure Service Availability

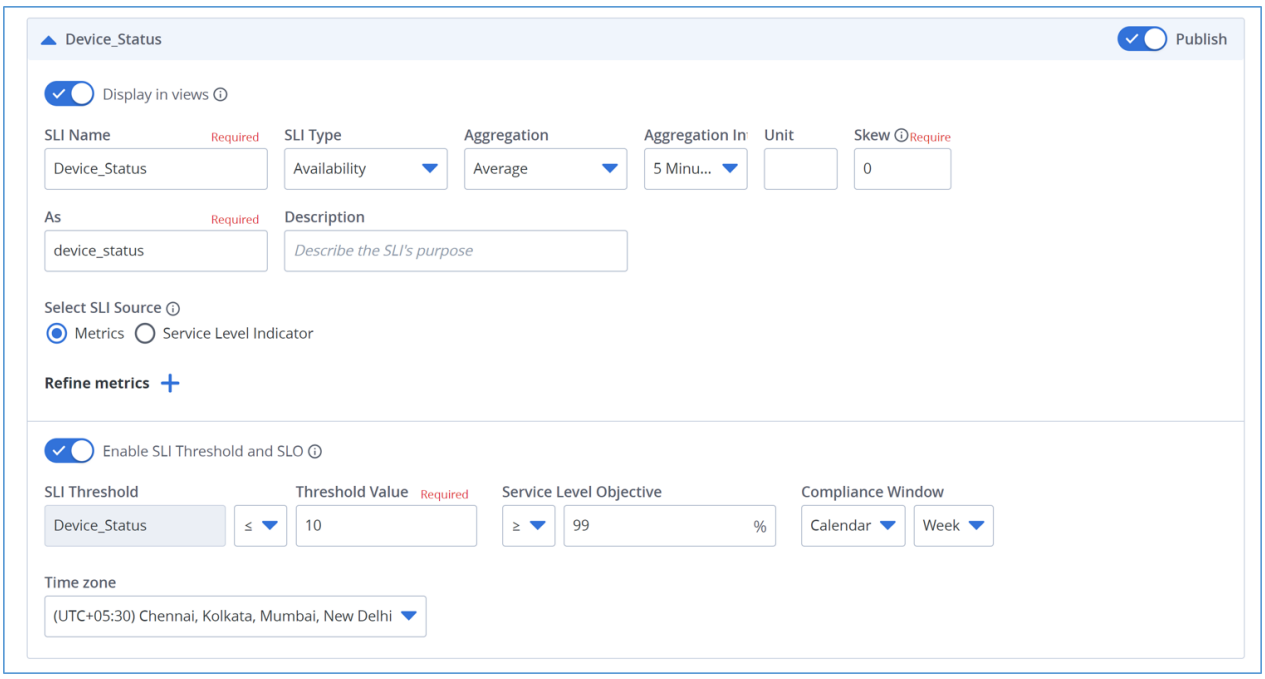
The following example illustrates how to use SLI and SLO to monitor the service availability.

Follow these steps:

1. Create a Service or Use an Existing Service. To create a service,
   a. Click **Services** in the left navigation pane.
   b. Click

   ,
   and click **Create a Service**.
   c. Click **+ Add New Service**.
   d. Click **Edit** in the **Service Details** section.
   The **Manage Elements for Service <Service_Name>** dialog box is displayed.
   e. Select **agent** from the **Select Base Attribute for Definition** drop-down list.
   All the agents are listed.
   f. Filter the available elements if necessary using the Filter section.
   g. Click **Add**.
   h. Enter the service name. For example, AIOps_Service.
   i. Click **Save**.
   Wait for a minute for the service to be displayed on the Services page.
   j. Open the service and click the **Monitored Inventory** tab.
   The inventory data is displayed.
2. Create SLI/SLO.
   a. Open the **Settings** page.
   b. Click the **SLIs & SLOs** tile.
   c. Click **+ Create SLI/SLO**.
   d. Select the service that you want to create the SLI/SLO for. For example, select AIOps_Service.
   e. Identify the metrics for which you want to configure the availability. You can select a single metric or all the metrics. For example, select Device Status as the metric.
   f. Apply the metric filter (also known as the Primary Filter) to configure the service level indicator. You can filter by source, metric, metadata, or custom attributes. This is the first level for the metric pipeline to select the set of metrics that must participate in the subsequently derived aggregations, thresholds, and expressions.
   A list of metrics appears based on the filter applied. You can click a metric to view the metric chart.
   g. Click **Add Service Level Indicator**.

h. Provide the following information: The following image illustrates sample values:



- **SLI Name:** Provide the SLI name for the service. For example, Device Status.
- **SLI Type:** Select **Availability** to set the SLI as an Availability metric for the selected services.
- **Aggregation:** Select the aggregation type to aggregate the metrics.
  - **Average:** This function finds the weighted average of the input values and sets it to the output metric value.
  - **Boolean And:** This function applies boolean AND on the input values and sets it to the output metric value.
  - **Boolean Or:** This function applies boolean OR on the input values and sets it to the output metric value.
  - **Count:** This function provides the aggregated count of raw input metrics when the source is data.
  - **Expression:** Define the custom expression using the selected SLI. The expression can be any mathematical operation. You can provide the expression value in the **Custom Expression** field. Use **${pipeline:sli_as_value}** to denote a variable. For example, **${pipeline:sli_1}*(${pipeline:sli_2}+ ${pipeline:sli_3})**. Let us assume there are raw metrics on SLI1, and you want to add 50 in SLI2, then the expression can be written as **(${pipeline:SLI1}+50)**.

    NOTE

    DX Operational Intelligence supports ternary or conditional expressions for SLIs and SLOs.

    - ${pipeline:avg_metric1} >= ${pipeline:avg_metric2} ? Max(${pipeline:avg_metric1}, ${pipeline:avg_metric2}) : Min(${pipeline:avg_metric1} ,${pipeline:avg_metric2})
    - (${pipeline:avg_metric1} >= ${pipeline:avg_metric2} && ${pipeline:avg_metric2}>0) ? Max((${pipeline:avg_metric1} +${pipeline:avg_metric2}),10.0) : 0

Any number in the expression should be a decimal. For example, (${pipeline:max}>= ${pipeline:min} && ${pipeline:min}>0)?Max((${pipeline:min}+${pipeline:max}),500.0):100.0.

- **Forward:** This function is the alias of a raw metric, and you can use it as a pipeline metric (for expression function or any other pipeline function). This function forwards the raw metric into the pipeline with the same frequency as the raw metric and does not require any metric.
- **Max:** This function finds the maximum of the input values and sets it to the output metric value.
- **Min:** This function finds the minimum of the input values and sets it to the output metric value.
- **Percentage:** This function works on top of boolean output and calculates the percentage of 0/1 in the incoming metrics.
- **Sum:** This function sums the input values and sets them to the output metric value.
- **Threshold:** This function sets the threshold conditions.

  NOTE
  Set the threshold conditions when selecting the aggregation type as the Threshold.

- **Aggregation Interval:** Set the aggregation interval for the SLI metrics data to be ingested. For example, if you set the time interval as one min, the SLI metric data is ingested every minute.

  NOTE
  The aggregation interval must be equal to or greater than the interval of the incoming raw metric (metric filter applied)

- **Unit:** Provide the unit for the SLI metric.

  NOTE
  A maximum of five characters is supported.

- **Skew:** Determines the wait time before emitting the output metric. Provide values in seconds.
- **As:** Specifies the alias name for the SLI and is case-sensitive. However, you can customize the name as required.
- **Description:** Provide the purpose of SLI.

i. Select the **Metrics** radio button as **SLI Source**.

j. Click to enable **Publish**. You must publish the SLI for the aggregation to work.

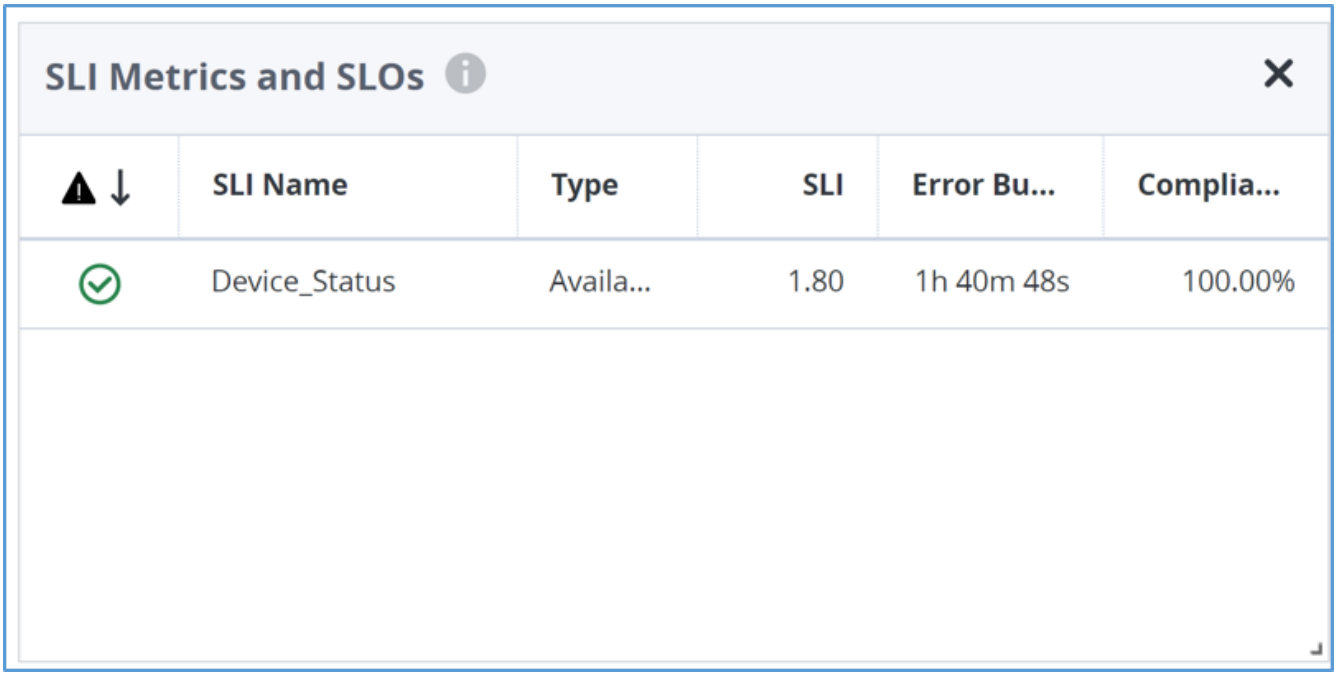
k. Click to enable **Display in views**. Enable this switch for the published SLI to appear in the views. This option is enabled only when SLI is published.

l. Click to enable the **Enable SLI Threshold and SLO** option.
   - Select the operator and enter the **Threshold Value**.
   - Select the operator and enter the **Service Level Objective**. The percentage of time the SLI threshold condition is met to achieve the service objective. The threshold is calculated based on the compliance window.

m. Select the **Compliance Window**. Select the time window (Rolling or Calendar) to evaluate your SLO.
   - **Rolling window:** You can measure the SLO metrics using the rolling window. The rolling period uses the First in, Last out method. The oldest data in the previous calculation drops out of the current calculation, and new data replaces it. The Rolling window uses the moving time range. For example, the aggregation interval of 5 minutes creates windows of 10:00-10:05, 10:01:10:06.
   - **Calendar window:** The Calendar window helps you calculate the SLO based on the boundary and not start the calculation randomly. The Calendar window uses the fixed date. For example, the aggregation interval for monthly creates windows of Jan1:Jan31, Feb1:Feb28(29). The calendar window is easier to manage and report. You can calculate the SLO metrics based on:
     - **Week:** The evaluation begins on Sunday and ends on Saturday.
     - **Month:** The evaluation begins on the first day of the month and ends on the last day of the month.
     - **Quarter:** The evaluation begins on the first day of the month and ends on the last day of the quarter. For example, if the SLO metrics evaluation starts on January 1, then it ends on March 31.
     - **Year:** The calculation starts on the first day of the year and ends on the last day of the year.

n. Click **Save**.
   Wait for a few mins for the SLI created to be synced.

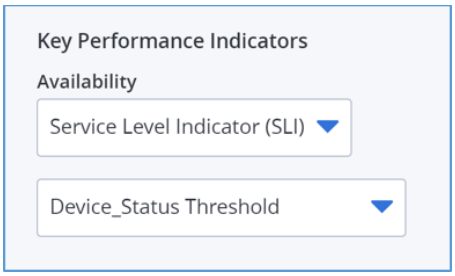
3. Configure the Availability.
   a. Verify if the **SLI Metrics and SLOs** panel is populated with data.
      a. Open the service for which you created the SLI. For example, open the service that is named AIOps_Service.
      b. Check if the **SLI Metrics and SLOs** panel is displaying the values for the metrics as shown:



      If not,
      - Click **Customize View** and check if the SLI you created is enabled under **Service Level Indicators**. If not enabled, click to enable.
      - Alternatively, wait for the metrics to be sent from the source product.

        NOTE
        If the source product is configured to send the data every 10 minutes and if the Aggregation Interval is 1 minute, then the data is sent after 11 minutes.
   b. Select the SLI Metric.
      a. Click

      and click **Edit Service**.
      b. Select **Service Level Indicator (SLI)** in the **Key Performance Indicators > Availability** section.
      c. Select **SLI Metric**. For example, select **Device_Status Threshold** as shown:



      d. Click **Update**.

4. View the data on the Services page.
   The second column indicates that the availability is being calculated.

Manage Adjustments

Compliance with an SLI/SLO indicates the performance of a service. The service owners are paid based on compliance with an SLI/SLO. To avoid any impact on compliance due to any outages, as a service owner, you can schedule maintenance windows for the services in DX Operational Intelligence. During those maintenance schedules, the Error Budget metric is not calculated for that service which ensures that you are not penalized for known maintenance schedules.

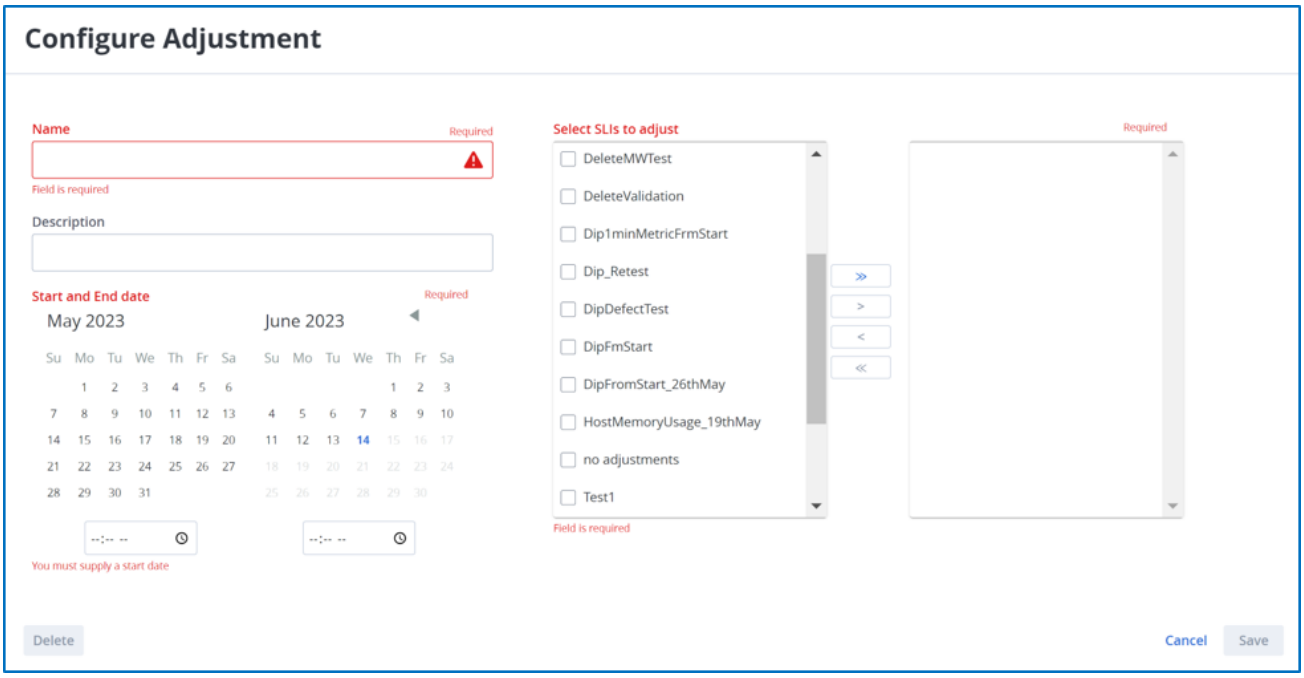
However, for any unexpected outages, the Error Budget metric dips and impacts compliance. To remove the impact of such outages on the Error Budget metric, you can make adjustments to the historical outages using the **Manage Adjustments** option that is available on the SLI/SLO page. You can add past time frames to the SLIs where the metric should not be calculated.

This section provides the following information:

- Configure the Adjustment
- Visualize the Adjustment

Configure the Adjustment

Suppose, there was an unexpected outage and the Error Budget metric dipped. You can add adjustments to remove the impact of this outage on compliance.



You can add the adjustment in the following ways:

- Add adjustment only for some time of the outage.
- Add adjustment to the whole time of the outage.
- Add individual adjustments if there were multiple outages within a time period.

Follow these steps:

1. Navigate to the Service Level Indicator page.
2. Click **+ Manage Adjustments**.
   The **Manage Adjustment** page is displayed.
3. Click **+ Add Adjustment**.
   The **Configure Adjustment** page is displayed.
4. Provide the following information:
   – **Name:** Enter a name for this adjustment.
   – **Description:** Enter the description.
   – **Start and End Date:** Select the start date and end date for the adjustment.
   – **Select SLIs to Adjust:** Select the SLI that you want to apply this adjustment to. Click **>** or **<** to add or remove one SLI. Click **>>** or **<<** to add or remove all the SLIs.
5. Click **Save**.
   The adjustment is added to the **Manage Adjustments** page with the following information:
   - **Name:** Name of the adjustment.
   - **Description:** Description of the adjustment, if added.
   - **Start Date:** Start timestamp of the adjustment.
   - **End Date:** End timestamp of the adjustment.
   - **Associated SLIs:** Total number of associated SLIs.

Visualize the Adjustment

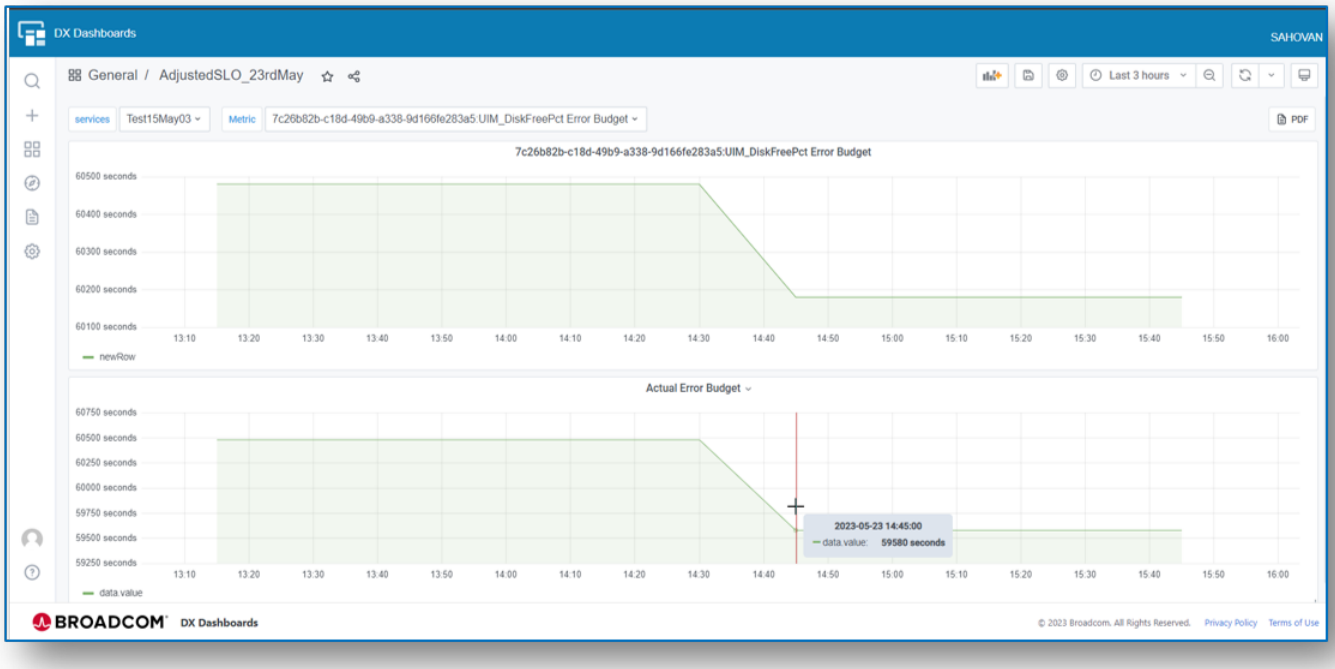
After you have added the adjustments, you can visualize these adjustments in the dashboards. The dashboard displays the following visualizations:

- **Adjusted Error Budget:** Displays the adjusted metric.
- **Actual Error Budget:** Displays the actual metric.

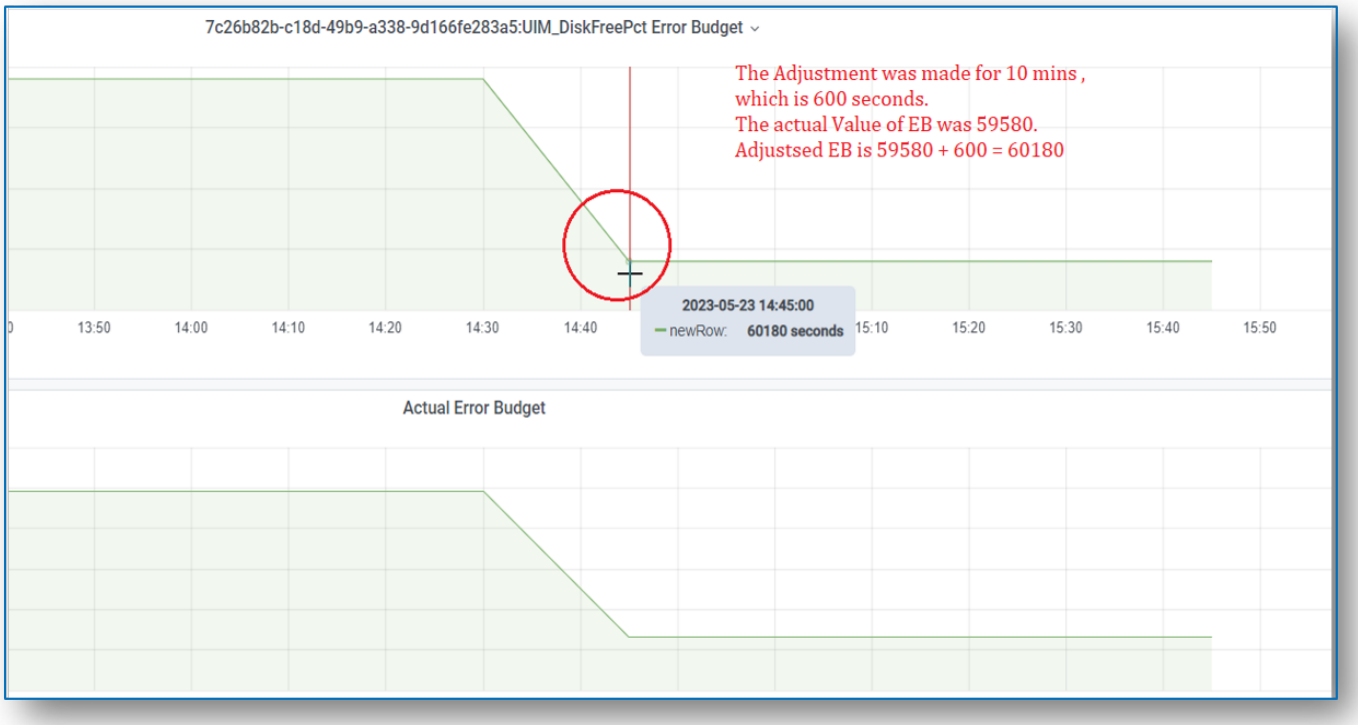
The following examples illustrate the Actual EB metric being compared with the Adjusted EB metric.

Example 1:

In this example, you can see that the adjustment is done only for 10 minutes of the dip duration though the metric dipped at 14:30.

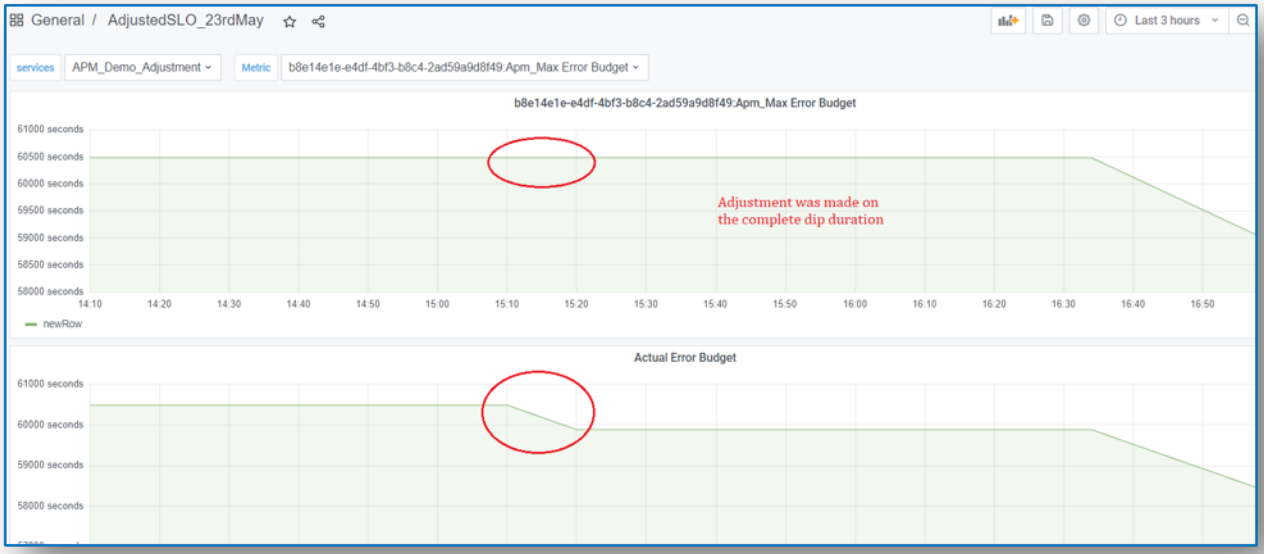


The following image displays the new value after adjustment. You can see that the metric is adjusted by adding the time in seconds to the actual EB value.



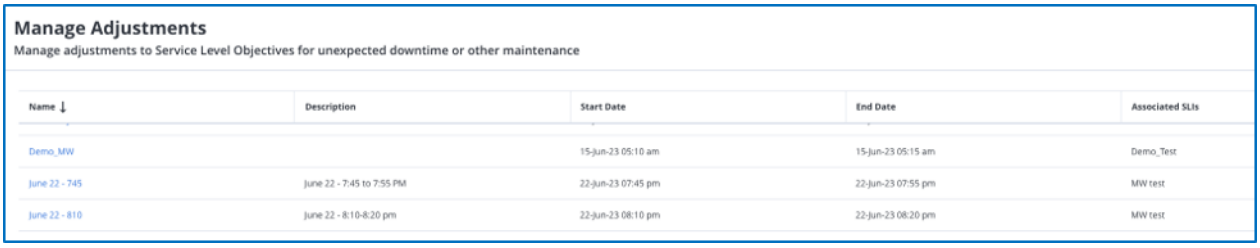
Example 2:

In this example, the adjustment is done for the entire dip duration as shown. Here is the Error Budget for the SLI Max Score. A dip was encountered for a brief duration as shown in the following image. Later it was identified that there was an unplanned maintenance during that time that resulted in the budget. After the adjustment was entered manually, the Error Budget was adjusted for the duration as reflected in the panel on the top.



Example 3:

In this example, the Error Budget dipped multiple times within a certain time period. You can add adjustments individually for each of the dips.

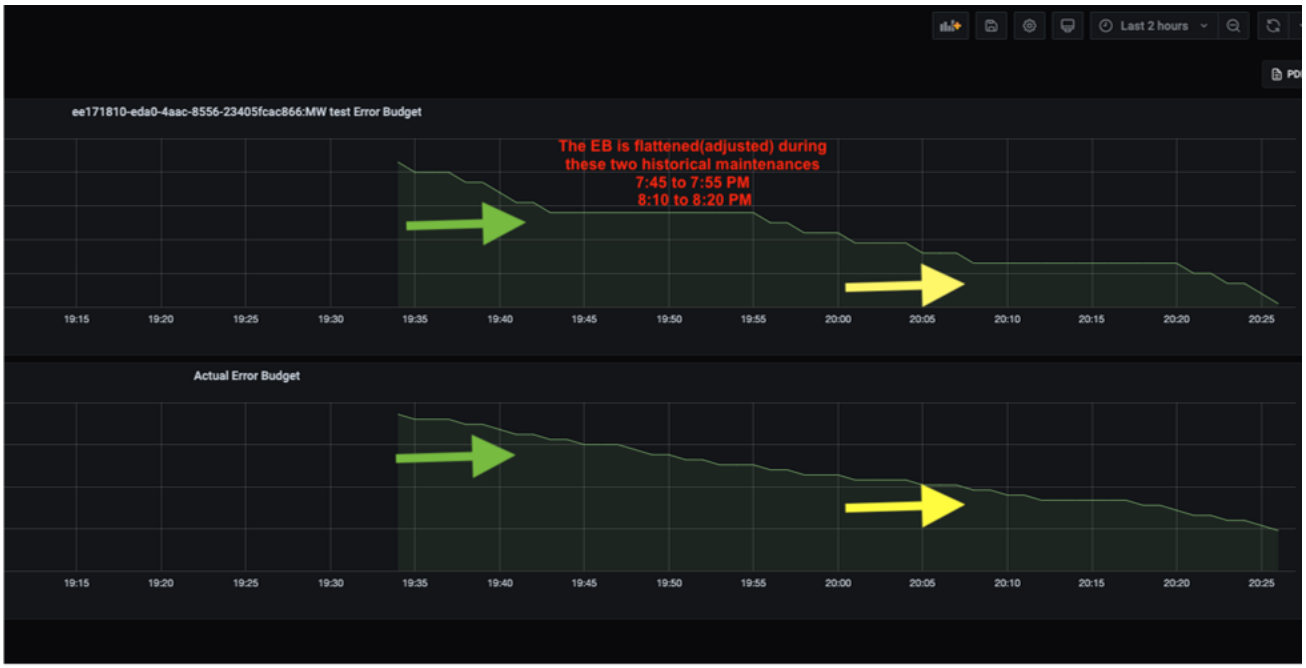


Manage Adjustments

Manage adjustments to Service Level Objectives for unexpected downtime or other maintenance

Name	Description	Start Date	End Date	Associated SLIs
Demo_MW		15-Jun-23 05:10 am	15-Jun-23 05:15 am	Demo_Test
June 22 - 745	June 22 - 7:45 to 7:55 PM	22-Jun-23 07:45 pm	22-Jun-23 07:55 pm	MW test
June 22 - 810	June 22 - 8:10-8:20 pm	22-Jun-23 08:10 pm	22-Jun-23 08:20 pm	MW test

The following image displays the adjusted Error Budget:



Error Messages

The following table explains the error messages that may appear on the SLI UI:

Error	Description
The SLI has no metrics matching	No metrics are available for the selected service.
The aggregation interval set for the SLI metric is lower than the interval of the incoming raw metric	This error appears when you have set the aggregation interval lower than the input metrics.
SLI Spec not registered because the metric count is exceeding the limit	This error occurs when you save the configuration, and the metric count exceeds 50k for an SLI.
Some of the metrics are outside the processing window.	This error appears when incoming raw metrics are not set within the scheduled aggregation interval. Due to this issue, the metrics get discarded.
Error in processing the SLI metrics	Runtime error while processing the SLI.
No active service associated with the SLI.	The service associated with the SLI gets deleted

Service Personalization

Service Personalization gives you an edge by allowing you to customize your service metrics, layout, and filters enabling you to focus on the right service metrics in a layout that works for your own needs of the organization.

You can add user-selected Service metrics (Service Indicators) to any service to improve visibility into the key service metrics impacting service health. You can add up to 10 metrics to your service. Selected service metrics are displayed on the Service Details view as a Time series chart. For improved sub-service metric visibility, when you add a parent service or sub-service as a shared service to any other service, the custom metrics that have been added to a service or sub-service also get added to the shared service.

Service Personalization adds the ability to create a frequently used layout called Service Filters. Service Filters allow you to create and retain customized layout, metrics, sort, filter, and column selection. Create multiple Service Filters for frequently used visualizations and customized flows. Service Filters are supported on the Services homepage and the Service Details view.

Personalizing Service View

Service Personalization enables you to customize your Services view instead of going through the hassles of finding certain services from the entire list of available services. This view allows you to create a service filter, view, save and delete the service filter.

You can now pin or feature the most frequently used services in the Services view by using **Filters** tab. After you pin a particular service, it gets featured as a tab in the Services view.

You can pin up to a maximum of 5 services at a time, you can drag and drop the service tabs in any order. You can also unpin a service at any point in time by just clicking on the pinned icon. The **Risk** section contains the total number of services under each risk category. For more information on risk, see Services Overview.

Create Service Filter

To create a Service Filter, perform the following steps:

1. From the **Services Overview** page, click **Filters**.

   You can view the list of existing pinned and unpinned service filters.

   - **Pinned**: This category displays up to a maximum of 5 pinned service filters. Pinned is represented by

     icon next to the service filter name and Unpinned is represented by

     icon next to the service filter name.

     NOTE
     Only the pinned service filters get featured on the **Services Overview** page homepage with all the statistics. If you want to feature or view a particular service, you need to pin that service.
   - **All filters:** This category displays the list of all the service filters which includes both pinned and unpinned service filters.

2. To pin a service filter and feature the service in the **Services Overview** page, click

   icon next to the required unpinned service filter.

   The service gets pinned automatically as a service tab in the **Services Overview** page homepage for further analysis.

3. To unpin an existing service from the filter, click

   icon next to the pinned service filter name.

After you pin certain filters for analysis, each of the pinned filters appears as a tab on the **Services Overview** page. You can also use the Health, Availability/Risk, Worst 5 pie charts on the **Services Overview** page to analyze the services, severity, and other service details.

NOTE
The **Services** view displays the Availability or Risk pie chart depending on whether the service is associated with any App Synthetic Monitor (ASM). If any service is associated with an ASM, the **Services** view displays Availability pie chart, else the Risk pie chart is displayed.

View and Modify Service Filter

You can view and edit an existing service filter by modifying the filter attributes. To edit a service filter perform the following steps:

1. From the **Services Overview**, click the **Filter** tab that you want to modify.
2. You can add more filter attributes or modify the existing filter attributes.

   NOTE
   After modifying the service filter attributes, you need to save the changes.

3. Click

   icon, and select **Save Current Filter**.
4. (Optional) You can modify the existing **Filter name** if required.
5. Click **Save**.

   NOTE
   The service filter gets saved with the latest changes.

Save Service Filter

You need to add the required filter criteria in order to save your service filter. You need to specify a name for the service filter, and you can also choose to pin the service filter while saving the service filter.

1. Click **Filters** tab.
2. Select or pin the required filters that you want to apply.
3. Click

   icon, and select **Save New Filter**.
4. Enter a **Filter name** for the new filter.
5. (Optional) If you want to pin the current filter, click

   icon.
   You can view the pinned filter as a tab.
6. Click **Save**.
   The service filter gets saved.

Delete Service Filter

To delete a current service filter that you have pinned, perform the following steps:

1. Click on the required **Service Filter** tab.
2. Click

   icon, and select **Delete Current Filter**.
3. Click **Delete** in the **Delete current filter** pop-up.
   The service filter gets deleted.

Personalizing Service Details View

Service Personalization enables you to customize your Services Details view by selecting your own layouts for viewing your service KPIs. You can create, modify, and delete your personal service details view.

You can drag and drop your service KPI charts and organize them as required. You can also add more service KPIs by clicking

option and enabling the service KPIs that you want to add to your Service Details layout. You can pin or feature the most frequently used layouts in the Service Details view by using **Layouts** tab. After you pin a particular layout, it gets featured as a tab in the Service Details view. You can pin up to a maximum of five layouts at a time, you can drag and drop the layout tabs in any order. You can also unpin a layout at any point in time by just clicking the pinned icon.

> **NOTE**
> If you create a layout that has custom metrics, then the layout is available only to that particular service. But, if you create a layout without any custom metrics, then the layout would be available across all services.

Create Service Details Layout

To create a Service Details Layout, perform the following steps:

1. From the **Services Overview** page, click a **Service**.
   The Service Details view opens with a default layout.

2. In the **Service Details** view, you can customize the default layout by performing drag and drop of the existing service KPIs.
3. To add another service KPIs, click

option.

4. Under **Select Service KPI**, enable the toggle option of the service KPIs that you want to add to you **Service Details** view.
5. You can view the list of existing pinned and unpinned layouts:
   - **Pinned**: This category displays up to a maximum of 5 pinned service layouts. Pinned is represented by

     icon next to the service layout name and Unpinned is represented by

     icon next to the service layout name.
   - **All layouts:** This category displays the list of all the service layouts which includes both pinned and unpinned service layouts.
6. To pin a service layout and feature the service in the **Services Details** view, click

   icon next to the required unpinned service layout.

   The service layout gets pinned automatically as a layout tab in the **Services Details** view for further analysis.
7. To unpin an existing service layout, click

   icon next to pinned service layout name.

   The selected service layout gets unpinned from the Service Details view.

8. Click

,

Save New Layout.

a) Enter a **Layout name** for the new layout.

b) (Optional) If you want to pin the current layout, click

icon.

You can view the pinned layout as a tab.

c) Click **Save**.

The service layout is created.

View and Modify Service Details Layout

To view and modify an existing Service Details Layout, perform the following steps:

1. From the **Service Details** page, click the **Layout** tab that you want to modify.
2. You can modify the current layout by using drag and drop, enabling or disabling the widgets.
   After making the required changes, you need to save the layout.
3. (Optional) You can modify the existing **Layout name** if required.
4. Click **Save.**
   The service details layout gets saved with the latest changes.

Delete Service Details Layout

To delete an existing service layout that you have pinned, perform the following steps:

1. Click on the required **Layout** tab.
2. Click

icon, and select **Delete Current Layout**

3. Click **Delete** in the **Delete current layout** pop-up.
   The service layout is deleted.

Service Personalization using Custom Metrics

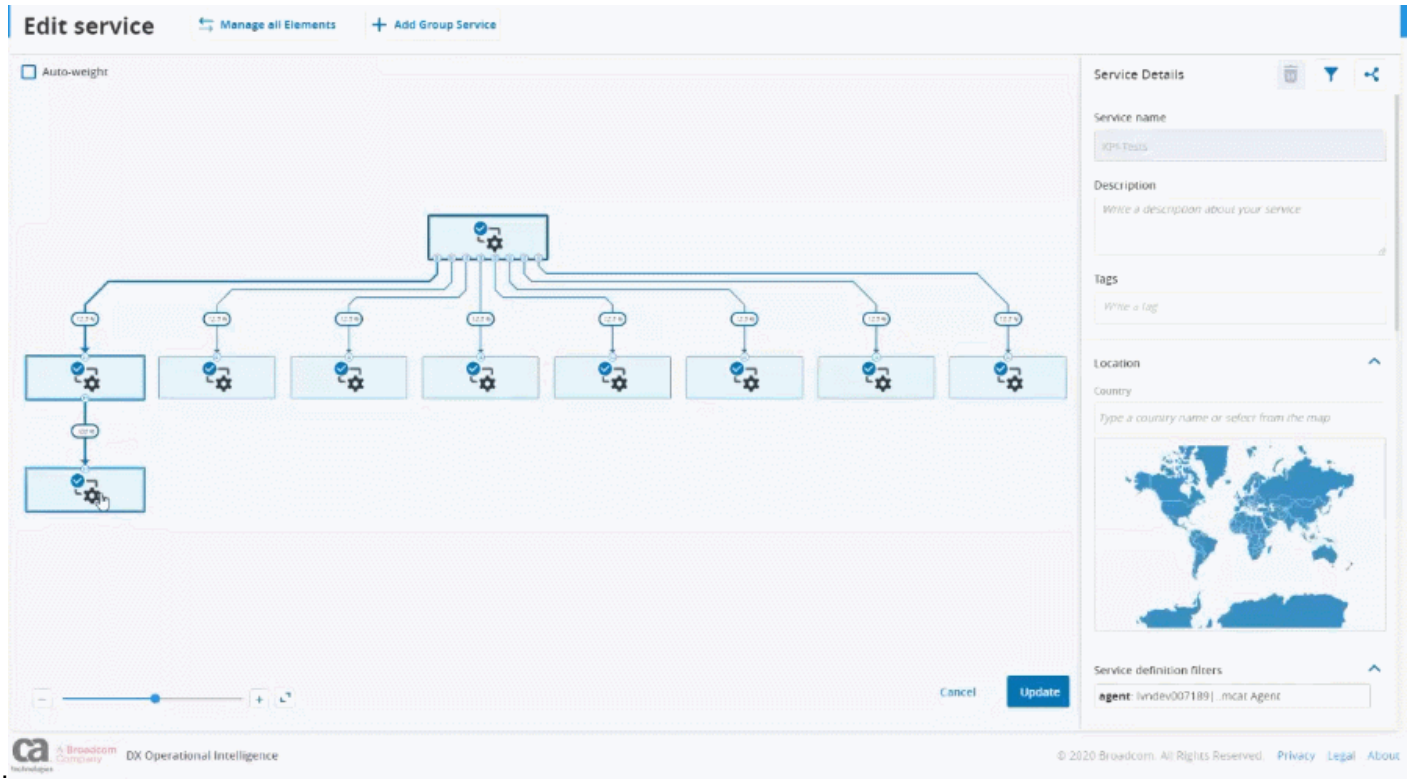
You can personalize your service by creating your own custom metrics to monitor a specific service and its associated devices.

You can prioritize a specific service along with the devices associated with that service, and monitor that service by using custom metrics. Service Personalization allows you to create up to a maximum of 10 custom metrics for a service. You can add a parent service or sub-service as a shared service to any other service, and the custom metrics that have been added to a service or sub-service also get added to the shared service.

Create Custom Metrics

To create a service personalization using custom metrics, perform the following steps:

1. In the **Services Overview**page, click an existing **Service** or **Sub-service**.
2. Click the icon, and select the **Edit service** option.
3. Click a **Service** for adding custom metrics.
   The Service Details pane opens with the details of the selected service.
4. In the **Service Details** pane, traverse down to the **Key Performance Indicators** section.



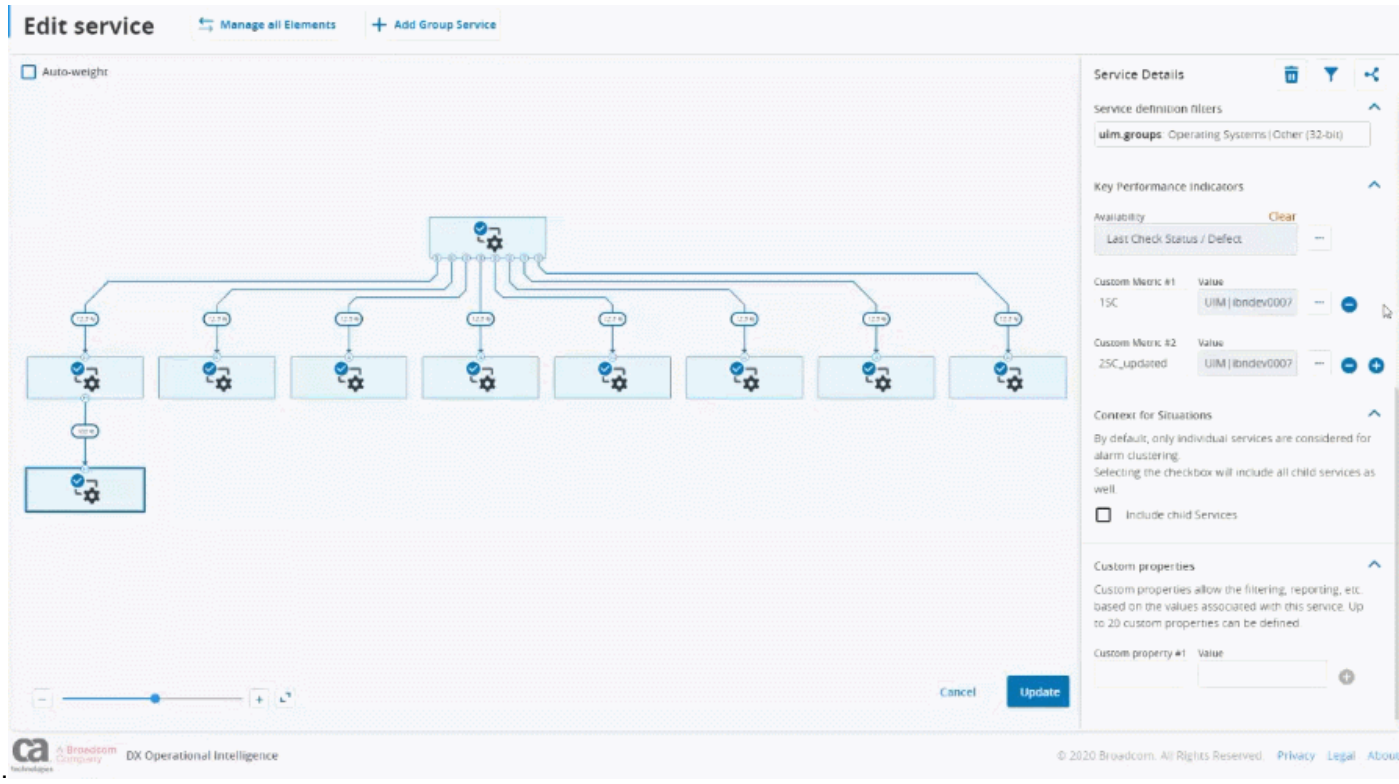
5. Under the **Key Performance Indicator** section, you can see the **Custom Metric** and **Value** option.
6. Click on the **Add**

   icon to add a Custom Metric.

   NOTE
   You can add up to a maximum of 10 custom metrics for a service. Once you add 10 custom metrics, the **Add** option gets disabled automatically.

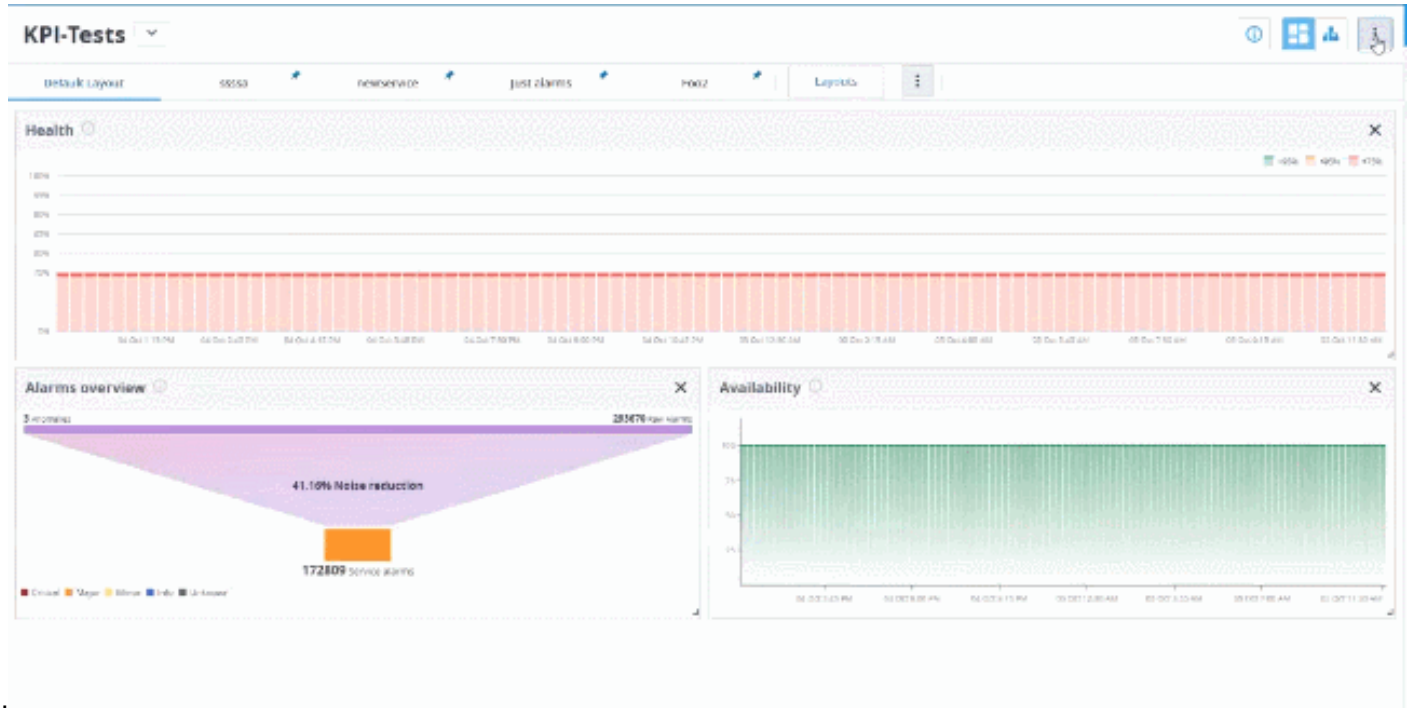
7. Enter a meaningful name for the **Custom Metric**.
8. Click

   next to the new metric name.
   The **Select 'Custom' metric** pop-up

   appears.
9. Select a product name and traverse through the hierarchy to select the required **Custom metric**.
   The selected metric gets added in the **Value** field.

10. Click **Update**.
11. Now from the updated Service Details view, click the icon and navigate to the **Select Service KPI** section. You can see the custom metric that was created under the Select Service KPI section.



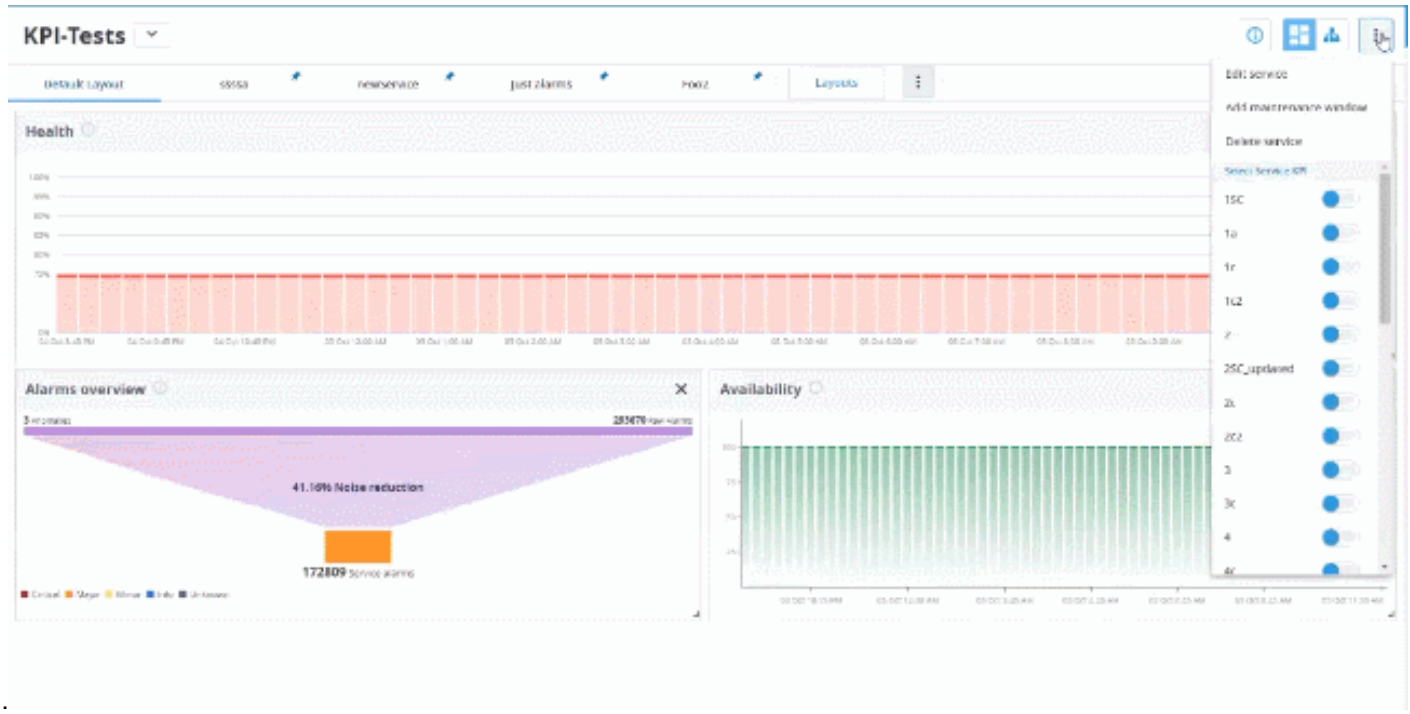
12. Enable the new custom metric using the toggle option to see the chart and monitor the service.
    You have created a new service personalization and can now monitor the specific service.

View and Modify Custom Metrics

To view and modify existing custom metrics, perform the following steps:

1. In the **Services Overview** vpage, click an existing **Service** or **Sub-service** that already has custom metrics applied.
2. Click the

   icon, and select the **Edit service** option.
3. Select a **Service** that has existing custom metrics

   applied.
4. From the **Service Details** pane, traverse down to the **Key Performance Indicators** section.
5. You can see the existing **Custom Metrics** and **Values**.
6. To remove an existing custom metric, click

   icon next to the custom metric.
7. To add another customer metric by using the

   icon next to the custom metric.
8. To modify an existing custom metric value, click on the button and select another **value**.
9. Click **Update**.
   You have updated the service personalization and can now monitor the specific service.

Alarm Analytics

Alarm Analytics in DX Operational Intelligence provides insights into service and derived alarms.

DX Operational Intelligence is a machine learning–driven, advanced analytics solution that is designed to help IT operations teams deliver a phenomenal user experience, improve the service quality, and drive operational efficiencies. DX Operational Intelligence offers an ecosystem that transforms these alarms into meaningful information which you can leverage for managing your IT platforms better. Alarms Analytics enables you to identify alarms that have a larger impact by offering extensive monitoring. You can understand the clustering of alarms that represent the context of a problem and can tie the problems to impact on services and applications.

Alarm Analytics helps you simplify your complex issues, raise the right tickets that should reach the right people by narrowing down the assignments that are based on domain, service, context, and so on, and should get quick resolutions. Alarms Analytics gives you insights on the root cause of alarms, the context, which you can leverage to create patterns and address common problems by using automation, which allows you to focus on the complex ones.

Alarm Analytics is a capability that provides an overview and insights into service and derived alarms. You can view the following information in the **Alarm Analytics** page:

- Service, raw, and anomaly alarms
- Alarm situations
- Alarms by device type and severity
- Variance in alarms for a period across devices, groups, and services
- Top five devices, groups, and services generating the most number of alarms

Using Alarm Analytics in DX Operational Intelligence, you gain the following capabilities:

- Reduce alarm noise from multiple products.
- Correlate alarms across products to identify the root cause.
- View probability bands to determine buildup to an alarm.
- Fine-tuning alarm threshold by analyzing the historical pattern.

Alarm Categories

Alarms can be classified into the following categories:

- **Anomaly Alarms:** An anomaly alarm gets generated when a metric value deviation is detected by the Data Science Engine for the configured metrics, by using machine learning algorithms. This alarm is generated when a threshold is crossed for the configured metric value.
- **Service Alarms:** A service alarm is a group of alarms that affect one or more business services and are related to an incident, which is identified by the time it occurred and its root cause. The root cause is the alarm on the topologically deepest device in the affected business service. All the situations that are reported by alarms in the group are due to the identified root cause.

  NOTE
  For new tenants the Service alarms are disabled. To re-enable the Service alarms, contact Broadcom Support.
- **Raw Alarms:** Alarms that are generated from source products such as CA Unified Infrastructure Management, DX NetOps Spectrum, CA ADA, and DX APM or any custom data source.
- **Situations Alarms:** Alarms are grouped based on context using machine learning algorithms. Clustering clubs alarms together based on distinct dimensions and groups them together for triage or further analysis. Thus, clustering enables users to filter through a huge number of alarms and analyze contextually relevant alarms.

Access Alarm Analytics

To access Alarm Analytics,

Follow these steps:

1. Log in to DX Operational Intelligence.
2. Click the Alarms icon in the left navigation panel.
   The Service Alarms page opens by default.
3. Access Alarm Analytics in the following ways:
   a. **View Service Alarms:** By default, the **Service alarms** page lists service alarms that have been generated in the last week. Alternatively, Click on the icon, click **Service alarms.**
      NOTE
      For new tenants the Service alarms are disabled. To re-enable the Service alarms, contact Broadcom Support.
   b. **View All Alarms**
      a. Click on the icon, click **All Alarms**.
   c. **View Situation Alarms**
      a. Click on the icon, click **Situations**.
   d. **View Alarms in the Context of a Service**
      a. Click the Services in the left navigation panel.
         The **Services** page is displayed.
      b. Click a service.
         The service summary page appears.
      c. Navigate to the **Alarms overview** section, where you can view the service alarms, anomaly alarms, and raw alarms for the selected service.
      d. Click an alarm type. You can now view the alarms for the alarm type for the selected service.

Alarm Severity Mapping

Incoming alarms from other products may be mapped to different severities in DX Operational Intelligence.

This section lists the alarm severity mapping and product severity mapping:

Severity Mapping

Incoming Alarm Severity Mapping	DX Operational Intelligence Severity Mapping
danger, critical	critical
major	major
warning, minor	minor
ok, good, normal, information	information
remaining other severities which are not listed above	unknown

Product Severity Mapping

DX Application Performance Management Severity Mapping

DX Application Severity Mapping Severity	DX Operational Intelligence Severity Mapping
Danger	critical
Caution	major
ok	information
unknown	unknown

DX Spectrum Severity Mapping

DX Spectrum Severity Mapping Severity	DX Operational Intelligence Severity Mapping
critical	critical
major	major
information	information
maintenance	unknown

DX Unified Infrastructure Management Severity Mapping

DX Unified Infrastructure Management Severity Mapping Severity	DX Operational Intelligence Severity Mapping
critical	critical
major	major
minor	minor
warning	minor
information	information

Third-Party Severity Mapping

Third-Party Severity Mapping Severity	DX Operational Intelligence Severity Mapping
critical	critical
major	major
minor	minor
warning	minor
information	information
Any other severity	unknown

All Alarms

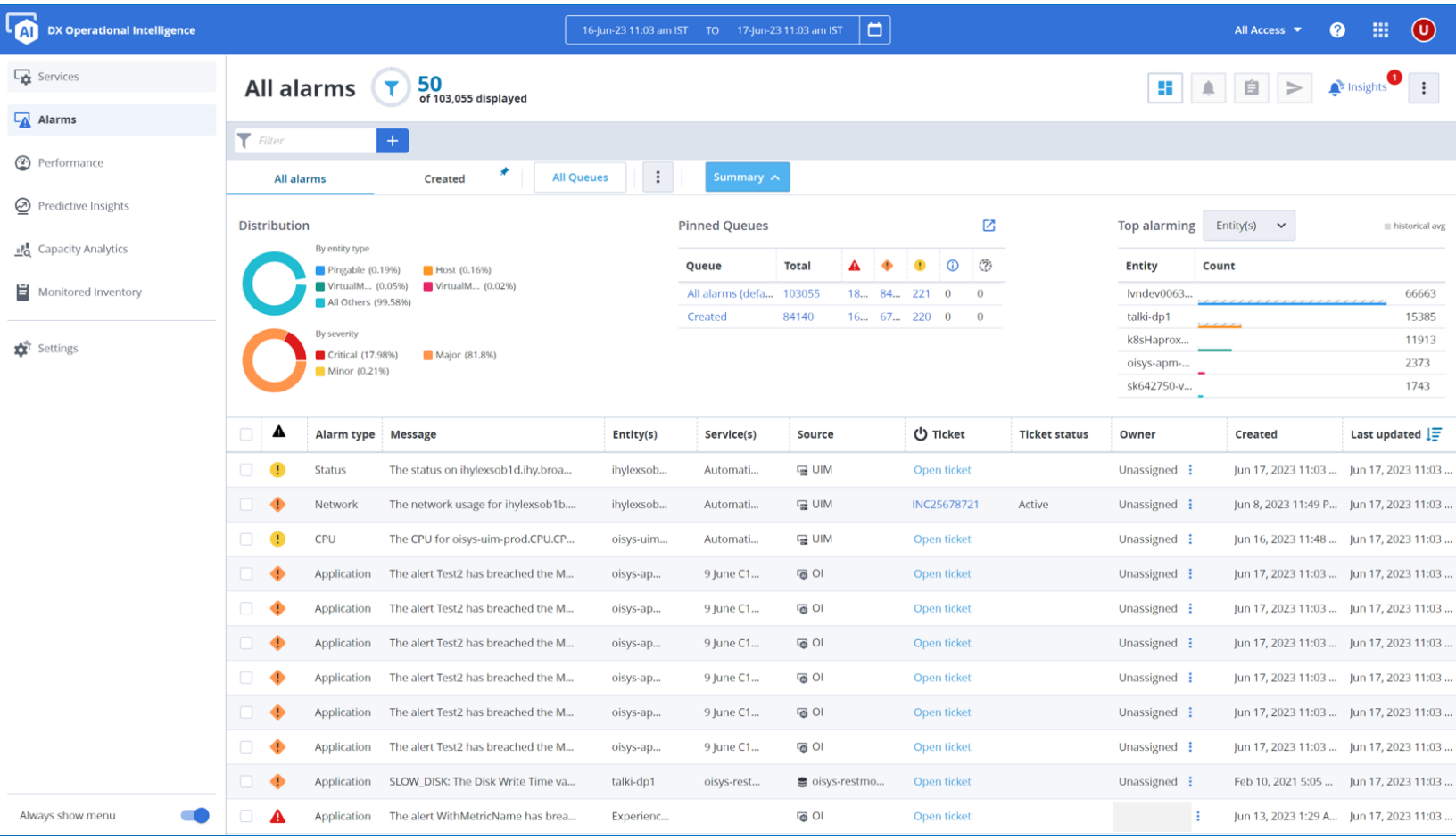
The DX Operational Intelligence All Alarms view provides an overview of your alarm data.

The All Alarms view gives you an overview of the alarms from the different products such as CA Unified Infrastructure Management, DX NetOps Spectrum, CA Application Delivery Analysis, and DX Application Performance Management or any custom data source. This page displays all the alarms that are in the open state during the selected time period irrespective of when they were opened. If the **Show Closed Alarms** option is enabled, this page also displays the alarms that are in the closed state during the selected time period.

To view All Alarms that are active or open during the selected period, click the **Change alarm views**

,

click **All alarms**.

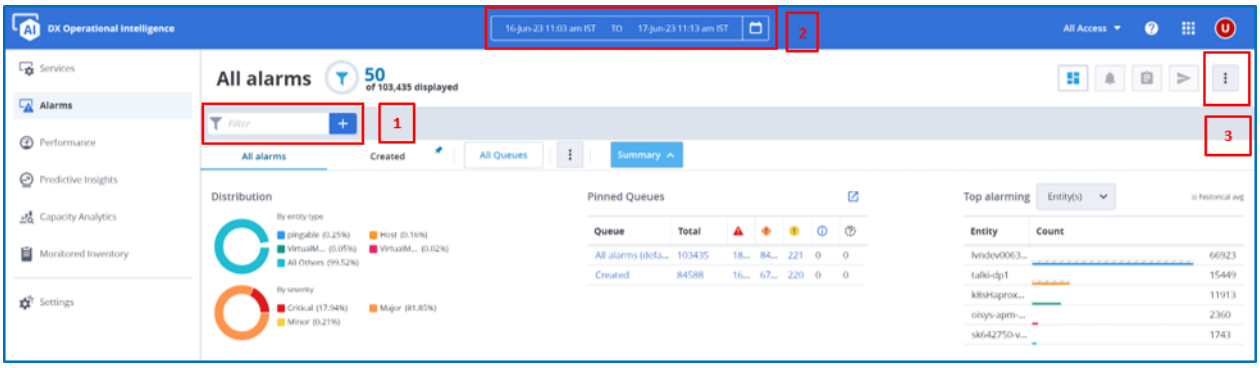


This section provides the following information:

- Filters
- All Alarms View
- Delete an Alarm

Filters

DX Operational Intelligence provides the following options to filter the alarms on all the Alarm Analytics pages:



- Alarm Attributes Filter (1)
- Date and Time Filter (2)
- Change Alarm Views (3)

Alarm Attributes Filter

You can use this filter to filter the alarms by attributes. You can either enter the attribute or you can click the icon to display the attributes.

NOTE

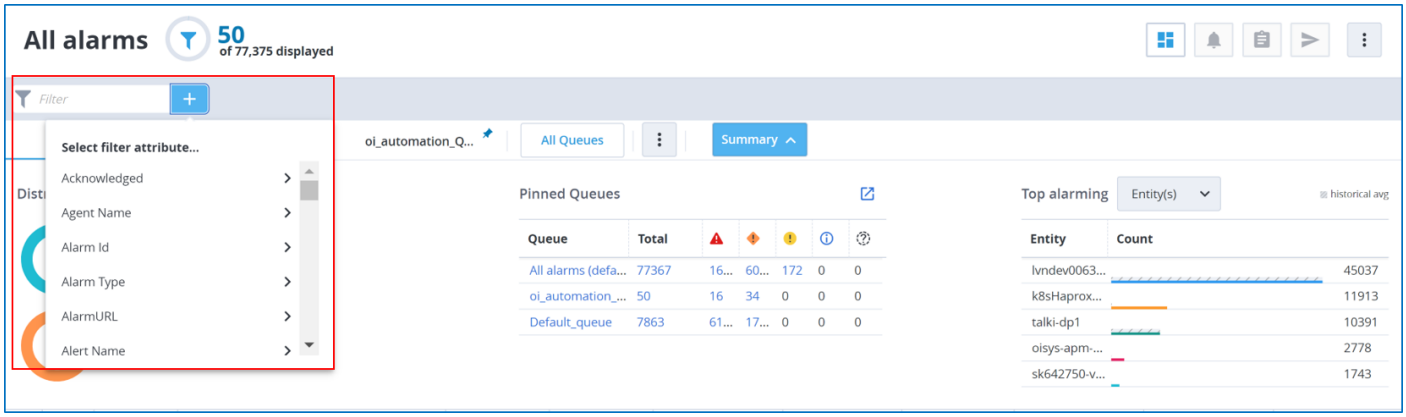
- While filtering, the **AND** operator is used between attributes and the **OR** operator is used between the attribute values. For example,

  ```
  (Severity: Critical OR Major) AND (Alarm Type:"application" OR "fault")  AND (Message contains "sshd" OR Message does not contain "ALARM: [SYSTEMS]")
  ```

- Only asterix (*) and dot (.) are supported in the filters.

Follow these steps:

1. Enter the attribute to display the attributes or click the plus icon in the **Alarm Attributes Filter**. The attributes are displayed.



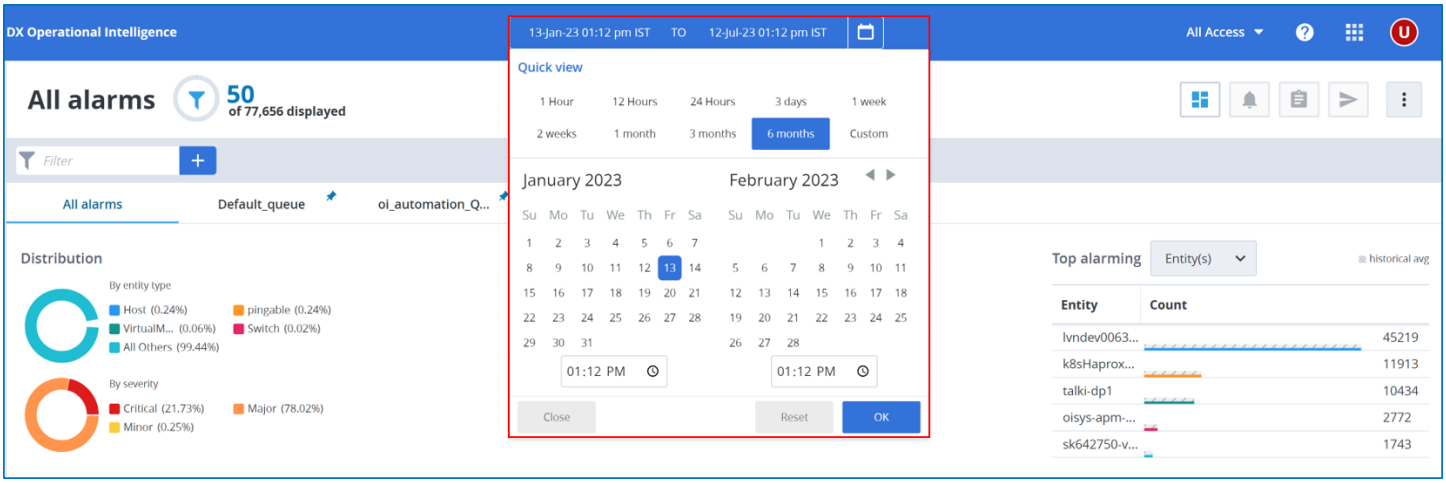
2. Select a filter attribute with any one or more operators. For example, if you want to see all critical alarms, select the attribute as **Severity** and select its value as **Critical**.

3. Click **Add** to add attributes to the alarm filter.
   The Alarms table and Insights show only the alarms that match your search criteria for the selected attributes.

Date and Time Filter

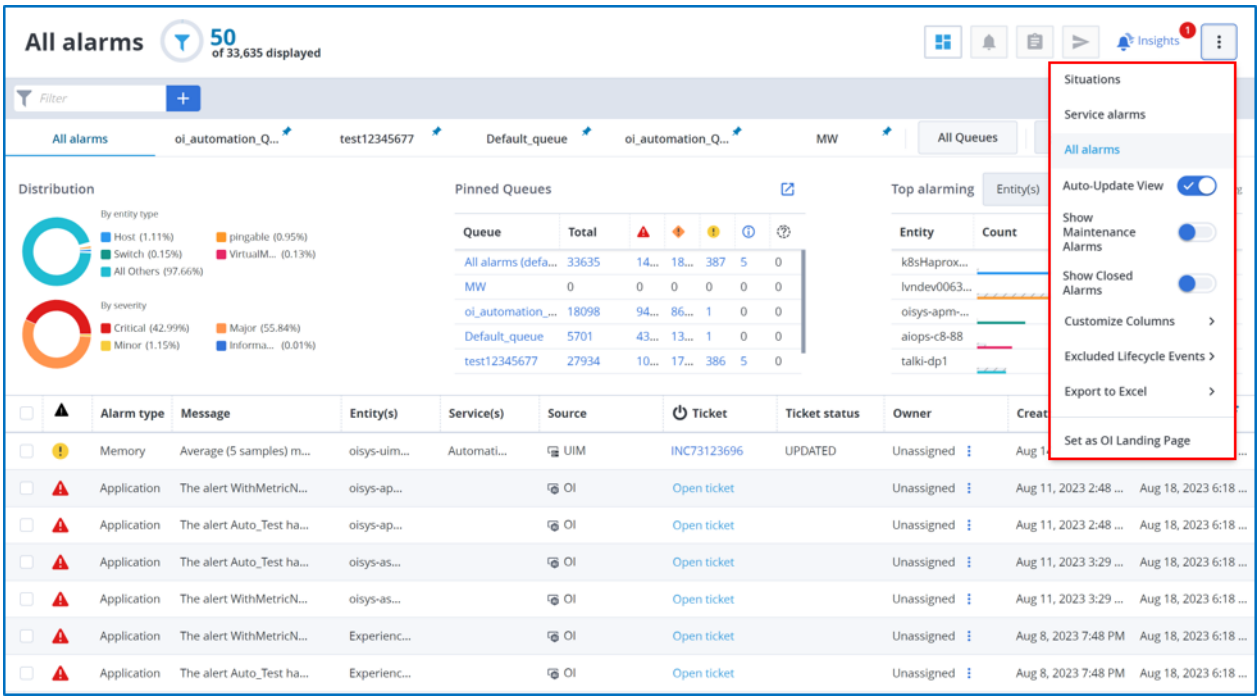
The Date and Time filter enables you to select the time period for which you want to view the active alarms (non-closed state) that have been triggered. By default, when you open the All Alarms view, the Date/Time filter displays the start date/time as the date/time of the oldest active alarm, that is, the alarm that is open for the longest period and displays the end date/time as the current date/time. The All Alarms view displays all the active alarms during this time period. You can use the Date and Time filter to search for active alarms during a specific time period such as 1 hour, 24 hours, 1 week, specific date/time.

Alarm Analytics also provides you with a **Custom** option, which enables you to pick a particular start date/time and a particular end date/time to help you narrow down the alarms that you are looking for.



Change Alarm Views

Use **Change Alarm Views** to filter alarms based on the following options for the selected time period:



- **Situations:** Displays all situation alarms.
- **Service Alarms:** Displays all service alarms.
- **All Alarms:** Displays all alarms.
- **Auto Update View:** Enable the **Auto-update view** option to refresh the Alarms table automatically.

  NOTE

  - If you want to manually update the alarm table:
    a. Disable the **Auto-update view** option.
    b. A pop-up (**Alarm view is out-of-date**) appears indicating "*Updates are pending for the current alarm view. Refreshing will update the view, but the focus may be lost. Applied filters and sorting will remain in effect after refresh*."
    c. Click **Refresh View** button to refresh the alarm table and the alarm table refreshes the data between the time interval that is set in the ALARM_REFRESH_INTERVAL environment variable and displays the last 7 days data or Select close (X), to leave the view as it is.
- **Show Maintenance Alarms:** Displays all the alarms that are in maintenance mode. These alarms are muted during the maintenance window.
- **Show Closed Alarms:** Enable the **Show closed alarms** option to view the closed alarms. The closed alarms appear disabled with a gray text. You cannot perform any action on these closed alarms. You can also view the closed alarm in the **Timeline** Tab. The **Close time** column displays the details of when the alarms were closed. This column is automatically displayed when you enabled the Show Closed Alarms option.

  ATTENTION
  By default, Auto Alarm Closure has been enabled for 30 days. Therefore, if any open alarm has no updates for more than 30 days, then the alarm gets closed automatically. If there is any update to an auto-closed alarm, the alarm gets reopened immediately.
- **Customize Columns:** The Customize Columns option enables you to customize the list of alarm columns, such as Alarm type, Severity, Service, Source. You can save the column settings of the alarm view table with the alarm information (filtering, and sorting options) to a new queue. You can add a maximum of 15 columns to the page.
- **Exclude Lifecycle Events:** This option enables you to exclude updates to the following options from displaying in the **Lifecycle Events** tab on the **All Alarms** page.

- Isolation View Link
- Metric View Link
- Alert Definition Link
- Occurrence

By default, all the options are selected. To add more fields to this list, you may contact **Broadcom Support**.

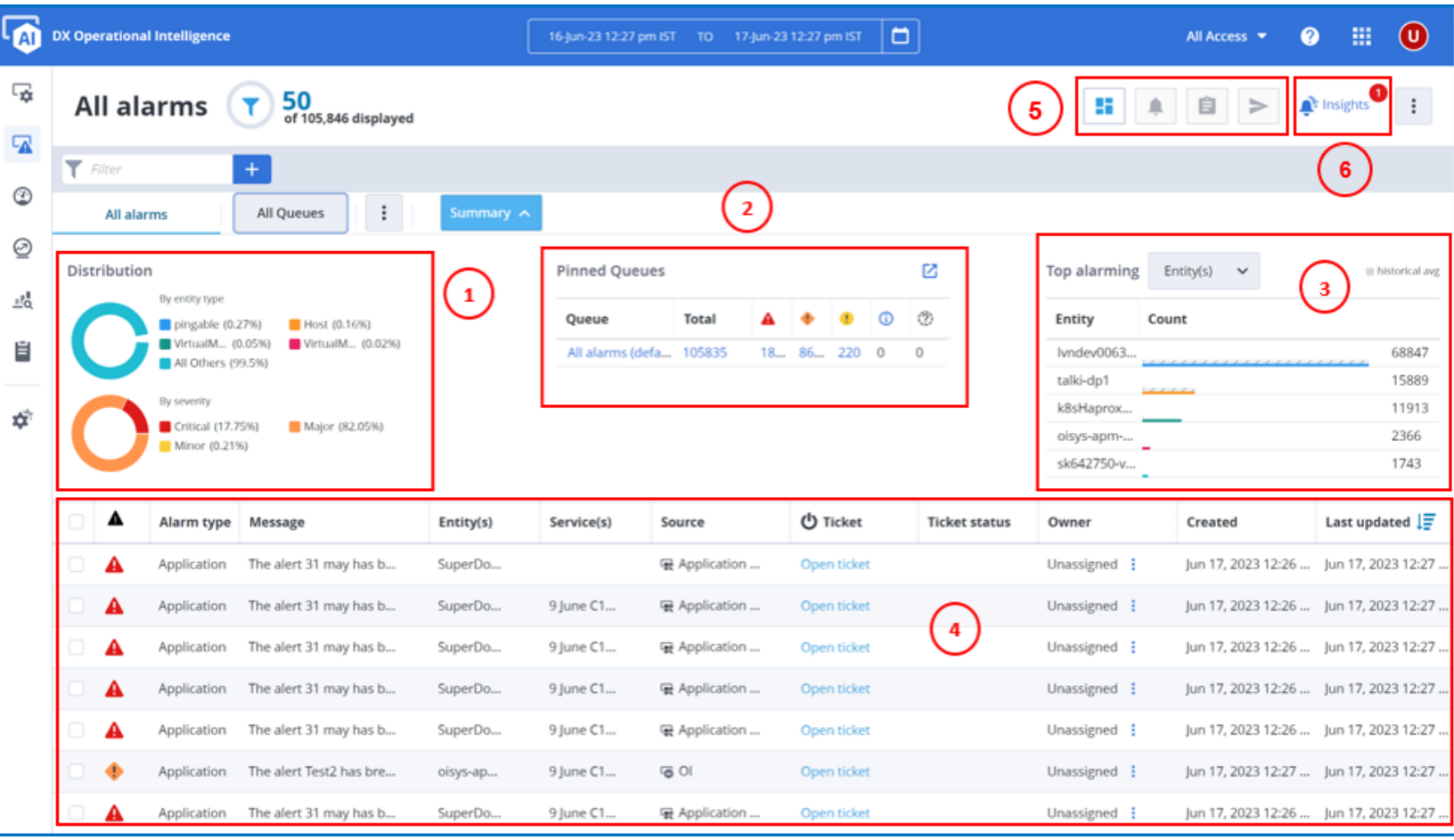
NOTE

- Updates only for the raw alarms are excluded and not for situation alarms. That is, the updates are not excluded in the **Lifecycle Events** tab on the Situations page.
- The selected options persist only for the current session.

- **Export to Excel:** This option enables you to export the alarm details to Excel. The Excel contains data that is displayed in the Alarm Table. You can export the data in the following ways:
  - **All Alarms:** Enables you to export the details of all alarms that are listed in the Alarm Table.
  - **Selected Alarms:** Enables you to export situation alarm details of specific alarms from the Alarm Table.

All Alarms View

The All Alarms view provides the information in the following sections:

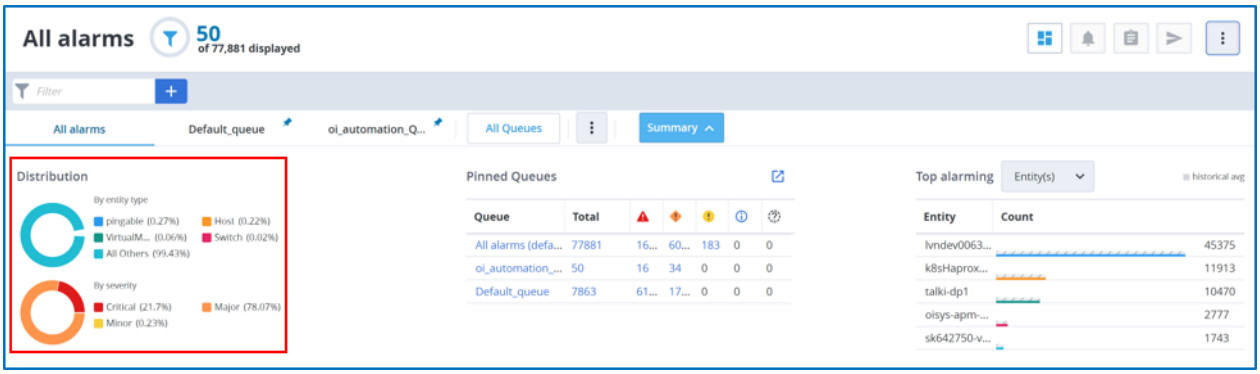


- Distribution (1)
- Pinned Queues (2)

  - All Queues
  - Save a Current Alarm Queue
  - Create a Policy for an Alarm Queue
  - Delete a Current Alarm Queue
- Top Alarming (3)
- Alarms Table (4)
  - All Alarms Details
- Alarm Actions (5)
  - Alarm Management
  - Ticket Management
- Insights (6)

Distribution

The **Distribution** section displays the distribution of alarms by entity type and severity.

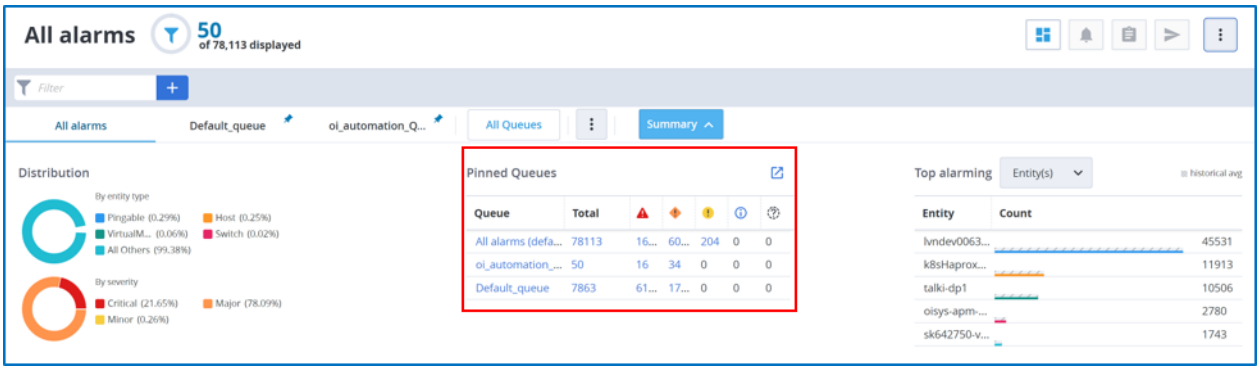


Click the legend to filter the alarms. For example, to view all the critical alarms, click the legend **Critical**. To clear the selection, click **CLEAR ALL** next to the attributes filter.

Pinned Queues

When you filter the alarms, you can save those filters as alarm queues and you can pin queues that you frequently use. The pinned queues get featured as tabs in the All Alarms view. Pinned is represented by icon next to the queue name and Unpinned is represented by icon next to the queue name. You can create any number of alarm queues but you can pin only a maximum of five queues.

The **Pinned Queue** section displays the queues that you have pinned. By default, only the **All alarms** queue is displayed in this table. After you pin a queue, this table is updated with the details of the pinned queue. For each queue, the queue name, alarm count, and alarm severity are displayed.



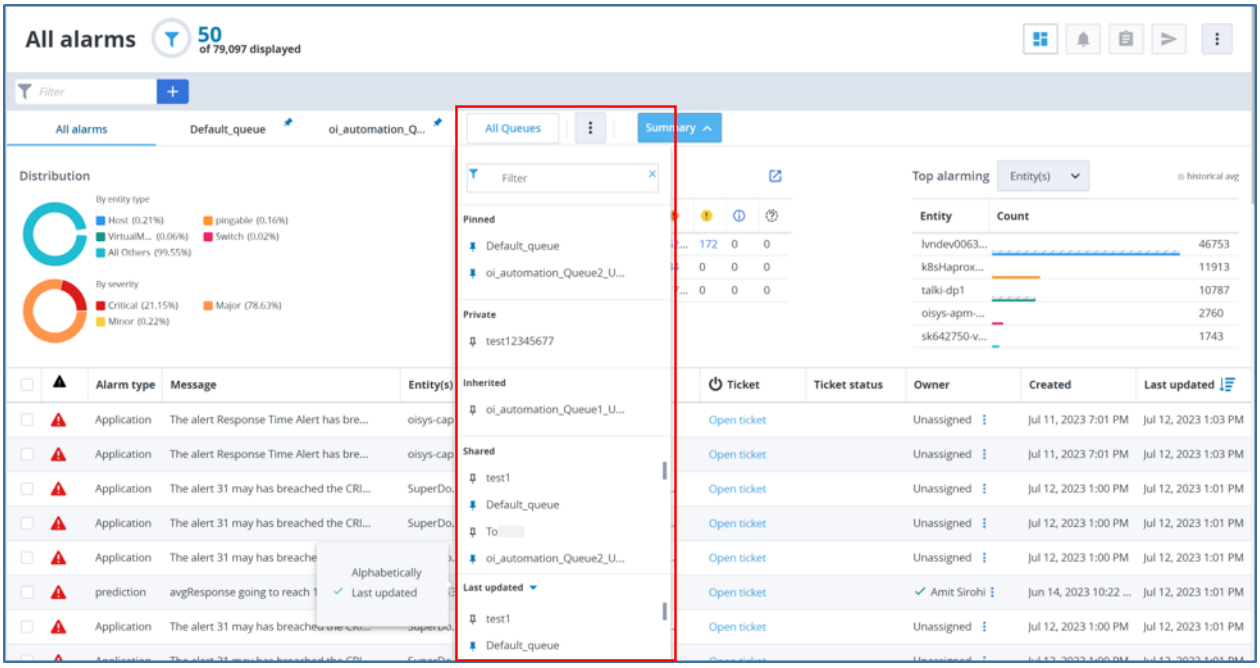
- You can sort each of the columns as per your needs.
- You can drag and drop the alarm queue tabs in any order.
- You can also unpin an alarm queue at any point in time by just clicking the pinned icon.
- To view all the queues, click the icon.

This section provides the following information:

- All Queues
- Save a Current Alarm Queue
- Create a Policy for an Alarm Queue
- Delete a Current Alarm Queue

All Queues

The **All Queues** button displays the created alarm queues under the following categories:



- **Pinned**: All the alarm queues that are pinned are displayed under this label. This category lists up to a maximum of five pinned alarm queues. When you pin a queue, the queue is automatically added to your pinned queues list and an alarm tab is also created for further analysis. Only the pinned alarm queues get featured on the All Alarms view with all the statistics. To pin an alarm queue, click the icon next to the required unpinned alarm queue. To unpin an existing alarm queue from any of the categories, click icon next to pinned alarm queue name. The selected alarm queue gets unpinned from the view.
- **Private:** All the private alarm queues are displayed under this label. These queues are available only to the user who created the queue but they are not available to other users even with the same role.
- **Inherited:** All the alarm queues that are inherited by virtue of being associated with a role are displayed under this label. For example, if you are assigned to Tenant Administrator and Power User roles, then all the queues that are shared with these roles are displayed here.
- **Shared:** All the alarm queues that are shared by the current user role with other roles are displayed under this label. For example, if you are assigned to the Power User role, then all the queues that are shared by you and the Power User role users are displayed here.
- **Last updated:** This category lists the queues that were recently updated, which includes both pinned and unpinned queues. Click the **Last Updated** drop-down to view the **Alphabetically** option.
- **Alphabetically:** This category lists the queues in the alphabetical order, which includes both pinned and unpinned queues.

Save a Current Alarm Queue

You can save (create) an alarm queue as private or you can share the queue with other roles. A private queue is available only to the user who created the queue. However, the shared queues are available to all the users who have access to the queues.

- Shared Alarm Queue
  - Manage Shared Alarm Queues
- Private Alarm Queue
  - Change Private Queue to Shared Queue
- Inherited Alarm Queue

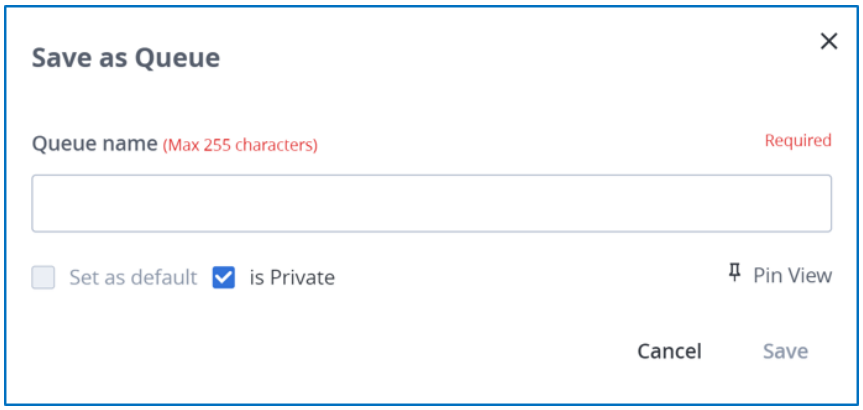
Shared Alarm Queue

You can share the alarm queues with other users through roles. All the users assigned to those roles can view, edit, and customize the shared queues depending on their privileges. When you share a queue, that queue is displayed under **Shared** for your account and for the other users, it is displayed under **Inherited** in the **All Queues** section.

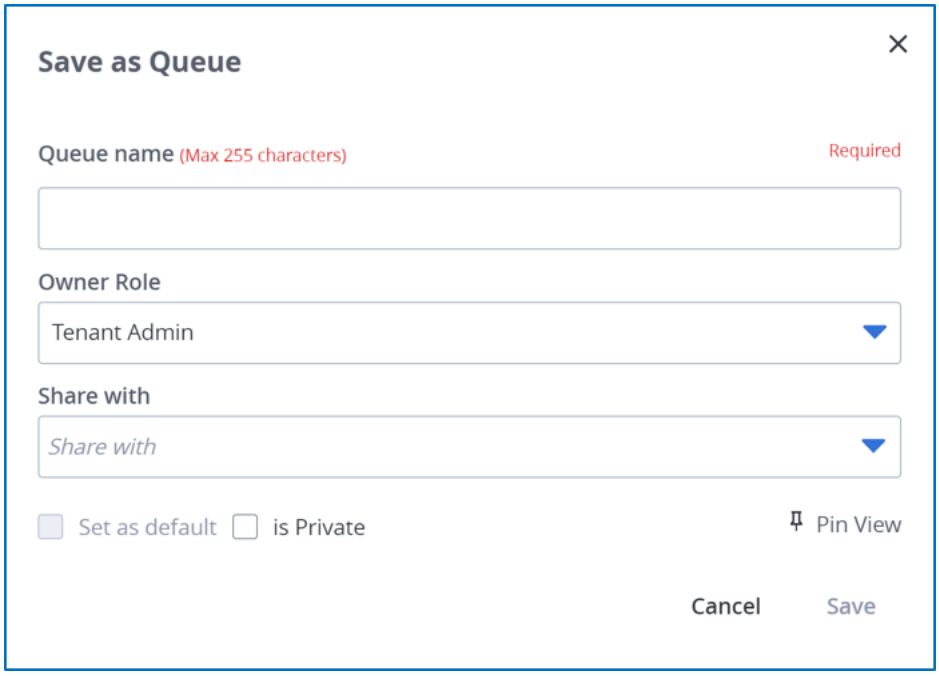
> **NOTE**
> To share the alarm queues, the user role must have the **Share Alarm Queues** privilege that is selected on the Roles page (**Roles > DX Operational Intelligence > All Features > Alarms Analytics > Alarm Queues > Share Alarm Queues**). If the user does not have this privilege, then they can only save the queue as private but they cannot share the queue.

Follow these steps:

1. Navigate to the **All Alarms** page.
2. Select the required filters that you want to apply.
3. Click the icon next to **All Queues**, and click **Save current queue**.



4. Deselect the **is Private** checkbox.
5. Provide the following information:



   – **Queue Name:** Enter a name for the alarm queue. Maximum characters allowed: 255.
   – **Owner Role:** For non-SAML based tenants, the current user role is selected. However, for SAML-based tenants, select the role that you want to provide the ownership of the queue. If you are assigned to multiple roles, then all those roles are listed. You can select the required role. All the users assigned to the owner role have access to edit, delete, and customize this queue depending on their privileges.

     NOTE
     After you save the shared queue, you cannot change the **Owner role** on the **Save as queue** dialog. However, a Tenant Administrator can change the Owner role using the **Manage Alarm Queues** tile on the **DX Operational Intelligence > Settings** page.
   – **Share with:** Select the roles to share the queue with. All the users assigned to the share with roles have access to view this queue.
   – **Set as default:** Click **Pin View** to enable this checkbox. Then, select the checkbox to make this queue as the default queue.
6. Click **Pin View** to add the queue to the **Pinned** list if you have not done it earlier.
7. Click **Save**.

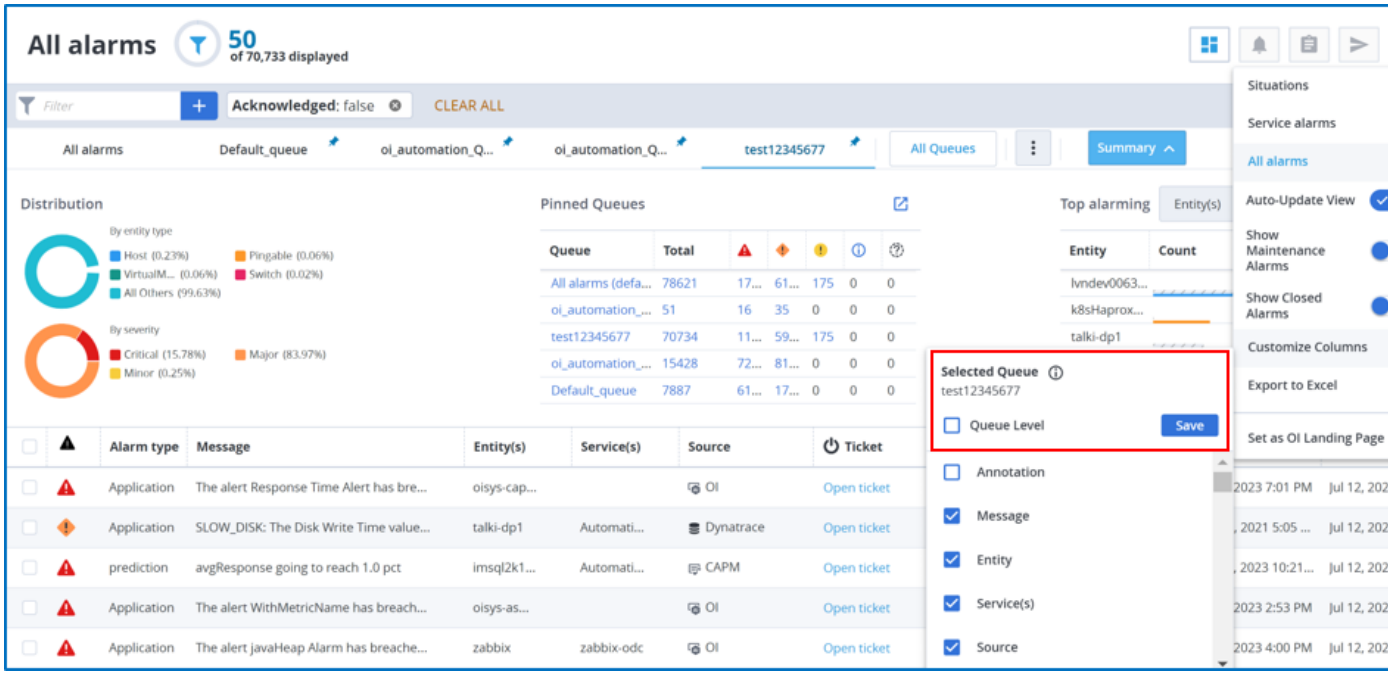
For all the **Owner role** users, the queue is displayed under **Shared** in the **All Queues** list. For example, if you created the shared queue as a Power User, then this queue is displayed under **Shared** for all the users who are assigned to the Power User role. For the **Share with** role users, the queue is displayed under **Inherited.**

The following table lists the actions that can be performed on the shared queues by each role:

Role	Privileges
Tenant Administrator	• View, save (Edit/Update), and delete the queue • Create a policy • Customize the columns for their account and also for other users at the queue level
Power User, User, Custom Role User (depends on access)	• View, save (Edit/Update), and delete the queue • Customize the columns for their account and also for other users at the queue level

NOTE

- You can customize the columns on the **All Alarms** view. Select the columns and click **Save**. To apply the customization to the other users, select the **Queue Level** checkbox and click **Save**. If the users have customized their view, user customization takes precedence over the queue-level customization.



- For all the existing queues,
  - The **Owner Role** is set to **Tenant Administrator** and all the users who are assigned to the **Tenant Administrator** role can view, edit, and delete the queue. You cannot change the **Owner role** on the **Save as queue** dialog but a Tenant Administrator can edit the queue and can update the role on the **DX Operational Intelligence Settings > Alarm Queues** page.
  - The **Share with** is set to all the roles in the tenant. All the users assigned to these roles can only view the queue.
- The default view and pinned view apply at the user level and not the queue level. If you have pinned a queue that is shared with another role, then the queue appears as pinned only for your account and not for the role user with whom you shared the queue. For example, you have shared a queue that is named

sharedqueue01 with user1 and you have also pinned the queue. This queue appears as pinned only for you and not for user1 if they have not pinned the queue.

- If a shared queue is changed to a private queue (the queue is unshared), then the queue is not visible to the users who had access to that queue. This queue is also removed from the **All Queues** list.
- If a shared queue is deleted, then the queue is not visible to the users who had access to that queue. This queue is also removed from the **All Queues** list. Any policies that are associated with the queues are also impacted.
- If the role that shared the queue is deleted, all the queues that are assigned to the role are deleted. Any policies that are associated with the queues are also impacted.

Manage Shared Alarm Queues

A Tenant Administrator can edit or delete the shared alarm queues across all the roles using the **Manage Alarm Queues** tile on the **DX Operational Intelligence > Settings** page. For example, if the ownership of the queue has to be changed, then a Tenant Administrator can update the Owner role on this page.

Follow these steps:

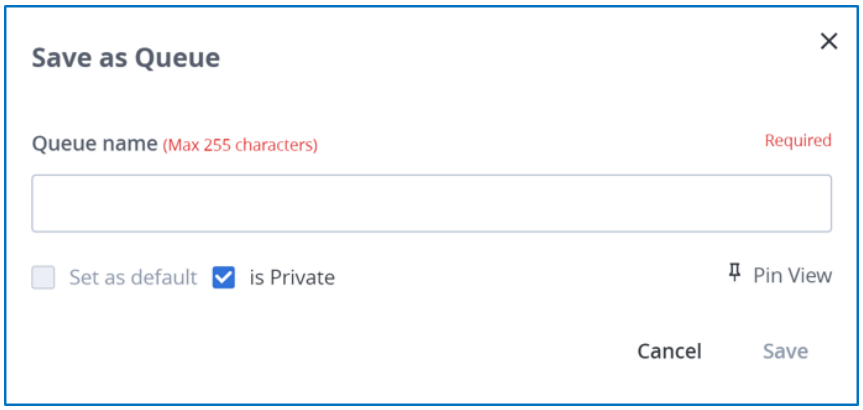
1. Open DX Operational Intelligence.
2. Click **Settings** in the left navigation pane.
3. Click the **Manage Alarm Queues** tile.
   The **Alarm Queues** page displays all the shared alarm queues.
4. Click the **...** icon under the **Actions** column for the queue:
   – **Edit:** You can edit the **Owner** role and the **Share with** role if necessary.
   – **Delete:** You can delete the queue. If the queue is deleted, then the queue is not visible to any user who had access to that queue. This queue is also removed from the All Queues list.

Private Alarm Queue

You can create an alarm queue as a private queue by selecting the **is Private** check box. A private queue is displayed only to the user who created the queue and is not available to other users even with the same role.

Follow these steps:

1. Navigate to the **All Alarms** page.
2. Select the required filters that you want to apply.
3. Click the ⋮ icon next to **All Queues**, and click **Save current queue**.
4. Provide the following information:



- **Queue Name:** Enter a name for the alarm queue. **Maximum characters allowed:** 255.
- **Set as default:** Click **Pin View** to enable this checkbox. Then, select the checkbox to make this queue the default queue.
- **Is Private:** This checkbox is selected by default. If not selected, select this checkbox to make the alarm queue private.
- Click the **Pin View** icon to add the queue to the Pinned list if you have not done it earlier.

5. Click **Save**.
   The queue is saved and displayed under **Private** in the **All Queues** list. Only the user who created the queue can edit and delete the queue and also create a policy if they have access. They can also customize the columns in the **All Alarms** view.

The following table lists the actions that can be performed on the private queues by each role:

Role	Privileges
Tenant Administrator	• View, save (Edit/Update), and delete the queue • Create a policy • Customize the columns for their account and also for other users at the queue level only if they share the queue
Power User, User, Custom Role User (depends on access)	• View, save (Edit/Update), and delete the queue • Customize the columns for their account and also for other users at the queue level only if they share the queue

Change Private Queue to Shared Queue

You can change a private queue to a shared queue. However, after you change the private queue to a shared queue, you cannot change that queue back to private.

> **NOTE**
> To share the alarm queues, the user role must have the following privilege selected on the Roles page: **Roles > DX Operational Intelligence > All Features > Alarms Analytics > Alarm Queues > Share Alarm Queues.**

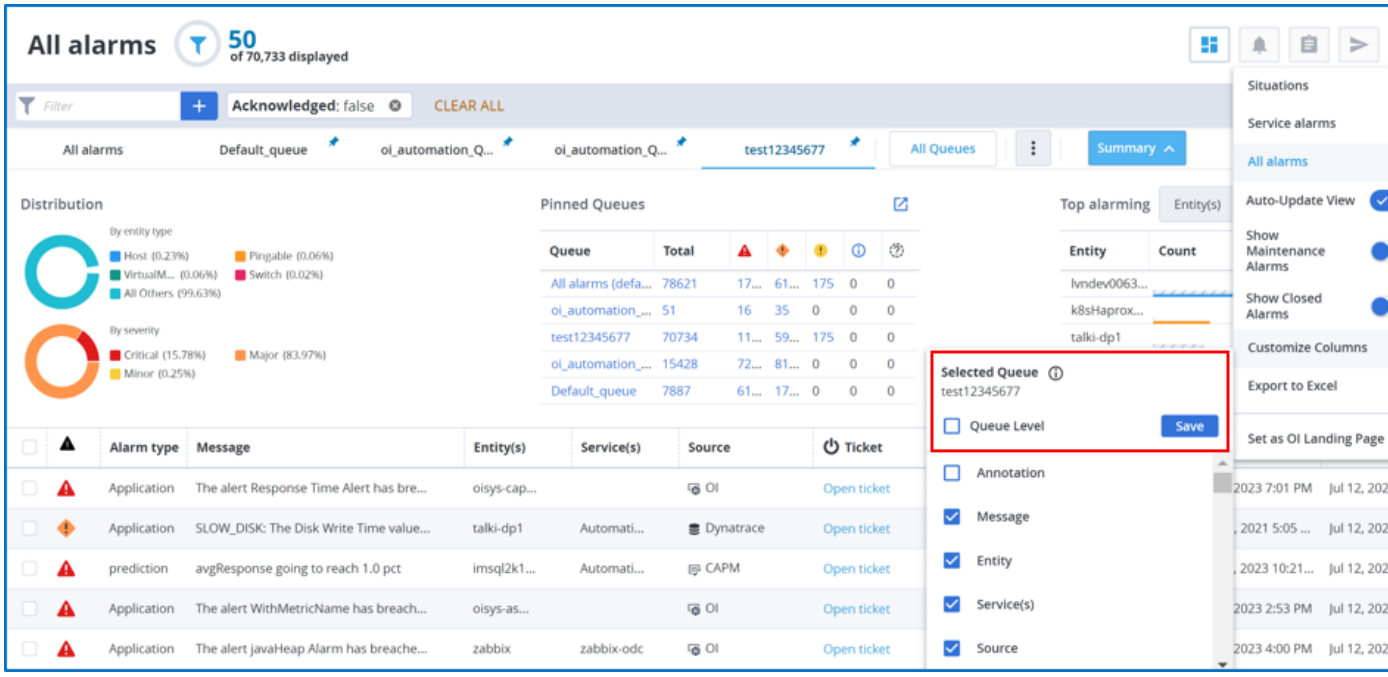
Follow these steps:

1. Open the shared queue. To open the queue, pin the queue and click the pinned tab.
2. Click **Save Current Queue**.
3. Edit the following information in the **Save as Queue** dialog:
   - **Queue Name** (if necessary)
   - **Is Private:** Deselect the checkbox to make the queue sharable.
   - **Owner Role:** For non-SAML based tenants, the current user role is selected. However, for SAML-based tenants, select the role that you want to provide the ownership of the queue. If you are assigned to multiple roles, then all those roles are listed. You can select the required role. All the users assigned to the owner role have access to edit, delete, and customize this queue depending on the owner role privileges.

     > **NOTE**
     > After you save the shared queue, you cannot change the **Owner role** on the **Save as queue** dialog. However, a Tenant Administrator can change the Owner role using the **Manage Alarm Queues** tile on the **DX Operational Intelligence > Settings** page.
   - **Share With:** Select the roles to share the queue with.
   - Set the queue as default if necessary.
   - Pin or unpin the queue if necessary.
4. Click **Save**.
   The private queue is now changed to a shared queue and appears under **Shared** in the **All Queues** list. For the users with whom this queue is shared, the queue appears under **Inherited**. These users cannot reshare the queue.

NOTE

- You can customize the columns on the **All Alarms** view. Select the columns and click **Save**. To apply the customization to the other users, select the **Queue Level** checkbox and click **Save**. If the users have customized their view, user customization takes precedence over the queue-level customization.



- If a shared queue is deleted, then the queue is not visible to the users who had access to that queue. This queue is also removed from the **All Queues** list. Any policies that are associated with the queues are also impacted.
- If the role that shared the queue is deleted, all the queues that are assigned to the role are deleted.

Inherited Alarm Queue

All the alarm queues that you inherit by virtue of being associated with roles that have queues that are shared with them are displayed under **Inherited** in the **All Queues** list. You can only view, pin, and unpin an inherited queue and you can also customize the columns only for your user account. However, you cannot save or delete the queue, or create a policy for the inherited queue.

The following table lists the actions that can be performed on the inherited queues by each role:

Role	Privileges
Tenant Administrator, Power User, User, Custom Role User (depends on access)	• View the queue • Customize the columns for their account

NOTE

- You cannot reshare the inherited queues.
- If the queue that you inherited is made private (the queue is unshared), then the queue is not visible and is removed from the **All Queues** list.
- If the queue that you inherited is deleted, then the queue is not visible and is removed from the **All Queues** list.
- If the role that shared the queue is deleted, then the queue is not visible and is also removed from the **All Queues** list.

Create a Policy for an Alarm Queue

A policy determines when notifications are sent. When an alarm meets the criteria that are defined by the filters in the policy, a notification is sent to the associated channels. For more information, see Policies. Therefore, you can create a policy to be triggered when filters or criteria that are defined are fulfilled.

NOTE

- Only the alarm filters support asterisks but policy filters do not.
- Filters in the alarm queues and policies do not support regular expressions (Regex).

To create a policy for an alarm queue, perform the following steps:

1. Navigate to the **All Alarms** page.
2. Create the alarm queue. For example,
   - **Create an Alarm Queue for Alarm State:**
     a. Apply the required filter.
        - **Acknowledged:** True, false
        - **Assign State:** Assigned, Unassigned
        - **Ticket State:** Ticketed, Unticketed
     b. Save the current queue by specifying a name for the alarm queue. You can also choose to pin the alarm queue while saving the alarm queue.
   - **Create an Alarm Queue for the Age of the Alarm:**
     a. Select the filter ***Created***. The created filter has the following options:
        - Greater than equal to
        - less than equal to

          NOTE
          You can create a policy but notifications are not triggered when you provide the value as **one** for **less than equal to** filter.
        - Created Date Range.

          NOTE
          The Create policy option is disabled for the **Created Date Range** filter.
     b. Save the current queue by specifying a name for the alarm queue. You can also choose to pin the alarm queue while saving the alarm queue.
3. Click the required alarm queue on the All Alarms view:
4. Click icon and select **Create a policy**.
   The Policy page opens.
5. Enter a **Policy Name**.
   The **Alarm Type** and **Alarm Queue** are selected by default.
   You can use the filter option to narrow down your alarm queues.
6. Select the **Trigger Alarm Notification** option.
7. Select the channel.
8. Select the message template to use.
9. Click **+** to add another channel.
10. Click **Save**.
    The policy gets created.

    NOTE
    To restrict the Maintenance alarm notifications, create a policy by setting the **Maintenance** value as **false**.

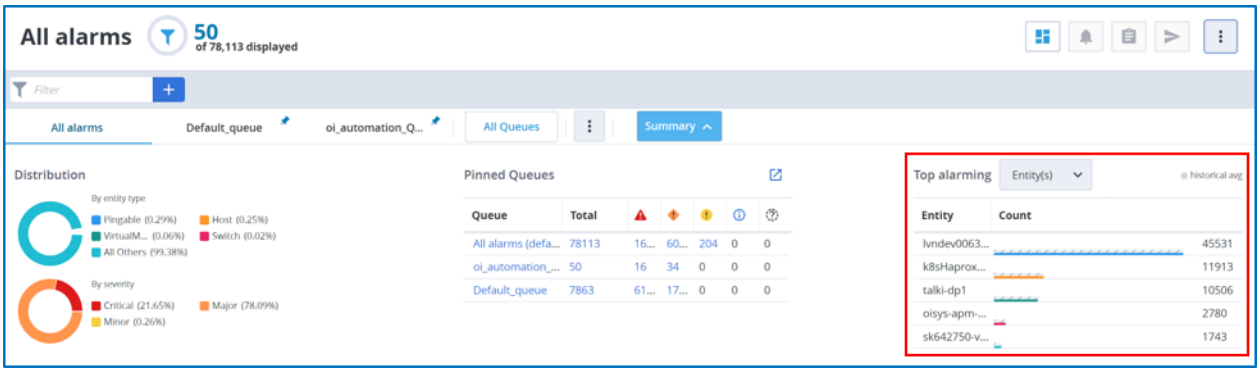
Delete a Current Alarm Queue

To delete a current alarm queue that you have pinned, perform the following steps:

1. Click the current or required alarm queue on the All Alarms view.
2. Click icon and select **Delete current queue**.
3. Enter an alarm queue name and click **Save**.
   The alarm queue gets deleted.

Top Alarming

The Top Alarming section displays the top five Entities, Groups, or Services that generate the most number of alarms. The colored horizontal bar displays the count of alarms while the underlying shaded horizontal bar shows the historical average for the entity, group, or service. The following image illustrates the top five entities:



NOTE
Click the Bar graph to filter alarms by the selected entity, group, or service. The selected filter is applied to the Alarms table. To remove all filters, click **CLEAR ALL**.

Alarms Table

The Alarms table provides details of the alarms. By default, only some of the columns are displayed. You can customize these columns using the **Customize Columns** option in the **Alarm View filter**. Only a maximum of 15 columns can be displayed at a time.

NOTE
Click the column header to sort the Alarms table in ascending or descending order. Not all Columns are available for **Service**, **Situation,** and **All Alarms** in the Alarm Table.

Column Name	Description
Column / Row-level Alarm Action	Enables you to perform row-level alarm actions or perform bulk column-level alarm actions.
Severity	Indicates the severity of an alarm. The following colors indicate the severity: • Red: Critical • Orange: Major • Yellow: Minor • Light Blue: Informational • Black: Unknown

Column Name	Description
Annotation	Displays the annotation, if added. A message icon is displayed to indicate that annotation is added. You can filter by these attributes: **Annotation: Available**, **Annotation: Unavailable**.
Alarm Type	Displays the alarm type.
Message	Displays the description for an alarm.
Entity(s)	Indicates the device or application name.
Service(s)	Displays the service which is impacted by an alarm.
Source	Displays the product from which the alarm is generated.
Ticket	Displays the ID generated by the ticketing system. Click **Open ticket** link to open a ServiceNow ticket corresponding to the alarm.
Ticket Status	Displays the ticket status.
Owner	Indicates the owner of the Alarm. If the alarm is **Unassigned**, click ⋮, select **Assign to** and assign the alarm to the respective person.
Source Timestamp	Displays the alarm timestamp for the source product updates. • This column is not displayed by default. To display this column, select **Source Timestamp** in the **Customize Columns** list. • The **Source Timestamp** column reflects the timestamp for the source product updates and the **Last Updated** column reflects the timestamp for alarm updates that are made in the DX Operational Intelligence. • For existing alarms, this column is populated only after there is an update to the alarm from the source product.
Created	Displays the alarm creation date and time. This column reflects the timestamp for alarm updates that are made in DX Operational Intelligence. If no updates are made to the alarms in DX Operational Intelligence, then the **Source Timestamp** and **Created** columns reflect the timestamp for the source product.
Last Updated	Displays the date and time when the alarm was last updated in DX Operational Intelligence. If no updates were made in DX Operational Intelligence, then the Source Timestamp and Last Updated columns reflect the timestamp for the source product.
Situation ID	Displays the situation ID.
Situation Status	Displays the status of the situation.
Situation State	Displays the state of the situation.
Situation Ticket ID	Displays the ticket ID of the situation.
Agent	Displays the name of the agent.
Cs Id	Displays the ID.
Alert Definition Link	Displays the definition link for the alert.
Sub System	Displays the subsystem.
Metric Type	Displays the metric type.

Column Name	Description
Host	Displays the host.
Cause Code	Displays the cause code.
Monitoring Host	Displays the monitoring host.
CI Type	Displays the type of CI.
Group	Displays the group name.
Dev Id	Displays the device ID.
Model Type Name	Displays the name of the model type.
Component Name	Displays the component name.
Alarm URL	Displays the alarm URL.
External IDs	Displays the external IDs.
Tags	Displays the tags.
Domain	Displays the domain.
Metric View Link	Displays the link to the metric view.
CI Name	Displays the CI name.
Application Name	Displays the application name.
Isolation View Link	Displays the link to the isolation view.
Probe	Displays the probe name.
Model Name	Displays the model name.
Custom 1 - 10	Displays the value for custom attributes.

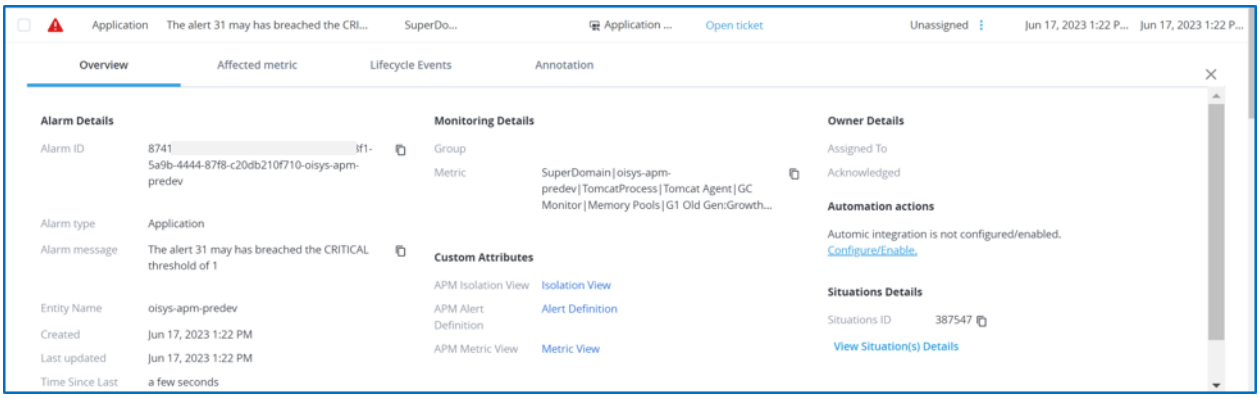
All Alarms Details

Click any row in the Alarms table, the row expands and displays the following tabs:

- Overview
- Affected Metric
- Impacted Services
- Topology
- Lifecycle Events
- Annotation
- Logs

Overview

The **Overview** tab provides additional information about the selected alarm. Properties are specific to the product or source from where the alarm originates.



The **Overview** tab displays the information under the following categories:

- **Alarm Details**

Source Product	Details
Application Performance Management	Alarm ID, Alarm Type, Alarm Message, Entity Name, Created, Last Updated, Time Since Last Update, Alarm Attributes
Spectrum	Alarm ID, Alarm Type, Alarm Message, Entity Name, Suppression Key, Created, Last Updated, Time Since Last Update, Alarm Attributes
UIM	Alarm ID, Alarm type, Alarm message, Entity name, Suppression key, created, last updated, Time since last update, Alarm Attributes

NOTE
Click the **Show Raw JSON** link to view the alarm attributes.

- **Monitoring Details**

Source Product	Details
Application Performance Management	Group, Configuration Item, Metric
Spectrum	Group, Model Type, Impacted Entities
UIM	Group, Probe, Monitoring host/robot, Source, Hub, Configuration Item, Metric, Item Type

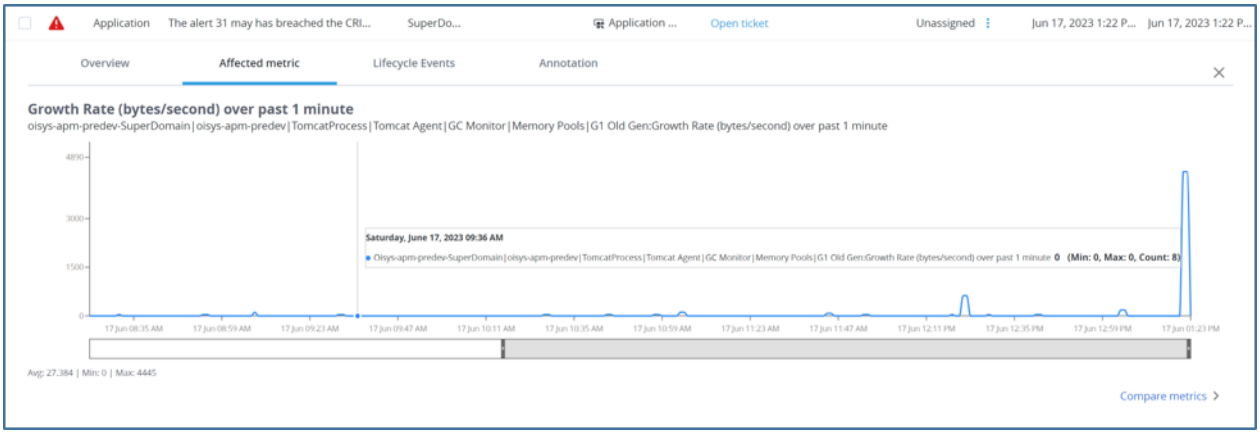
- **Custom Attributes**

Source Product	Details
Application Performance Management	APM Isolation View, APM Alert Definition, APM Metric View (Click the required link to navigate to APM.)
Spectrum	Alarm URL (Click the **Source Product Link** to navigate to Spectrum.)
UIM	Alarm URL (Click the **Source Product Link** to navigate to UIM.)

- **Owner Details** (Assigned to, Acknowledge)
- **Ticket Details** (Ticket Status and Ticket ID)
- **Automation Actions**: Displays if the Automic integration is configured or not. Click to configure or enable.
- **Service Alarm Details** (Service Alarm ID). Click the **View Service Alarms** link to navigate to the **Service Alarms** page.
- **Situation Details** (Situation ID). Click the **View Situation Details** link to navigate to the **Situations** page.

Affected Metric

The Affected Metric tab shows the metric chart of the underlying metric. If the required fields are not available in the alarm, this tab is not shown. Probability bands are shown if the metric is configured with Data Science Engine from Broadcom. If the metric is not configured with the Data Science Engine, the actual metric chart with original metric values appears. If the metric chart is not available, you must verify if the particular metric is ingested. The metric chart displays anomaly alarms when a threshold is crossed. The threshold is determined based on historical trends. The metric chart displays when a threshold is crossed. The threshold is determined based on historical trends. The alarm severity is indicated as minor, major, or severe.

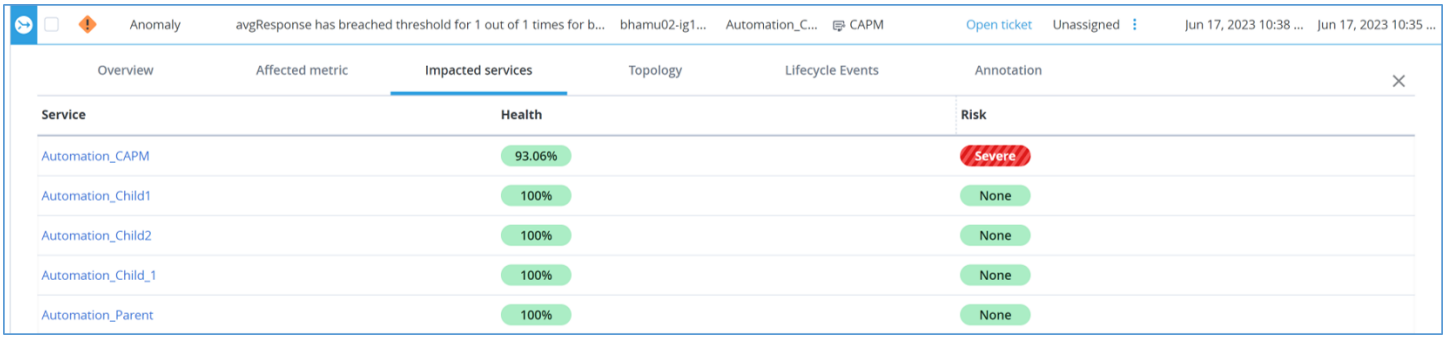


> **NOTE**
> By default, the chart time range is eight hours before the last alarm update to one hour after the last alarm update.

The Affected Metric tab has a **Compare Metrics** link that launches Performance Analytics from the context of an alarm and allows you to compare a single metric from different devices or multiple metrics from single or multiple devices. For more information about the metric charts, see the Performance Analytics section.

Impacted Services

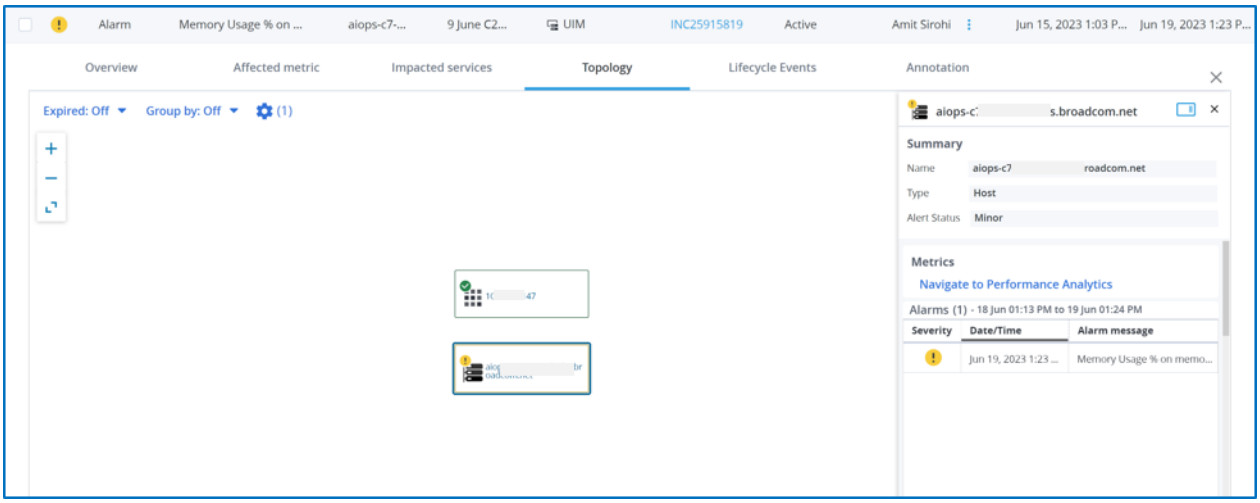
The Impacted Services tab provides details of the services that are impacted due to the selected alarm. Clicking a service redirects to the **Service Analytics** details view of that particular service. The table displays the impacted service metrics (such as users that are availing the service, actual service availability, and risk).



Topology

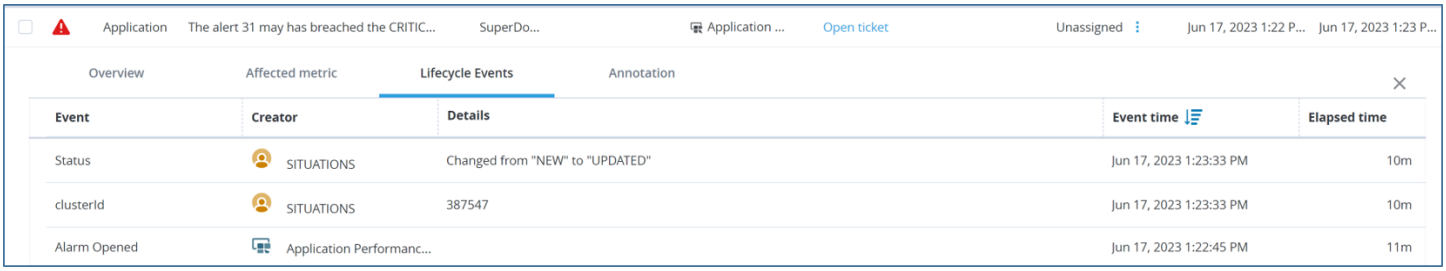
This tab provides topology details of the selected service or sub-service. You can view the topology details by clicking the topology icon

of the service. If you select a parent service and click the topology icon, the CI of its children appears along with the topology of the CIs of sub-services in different tabs. You can click a service to view the topologies associated with that service. You can also view the summary of the service by clicking the service from the Topology view. For more information on topology, see Topology Details.



Lifecycle Events

The Lifecycle Events tab provides the lifecycle of an alarm. This tab displays all the events that occurred for an alarm from the time it is created, such as created time, status updates, annotation updates, threshold changes, and alarms that are suppressed due to the maintenance window.



You can view the following details on the lifecycle events tab:

- **Event:** Displays the name of the event.
- **Creator:** Displays the name of the creator. If the event was performed manually, then the email ID is address is displayed. If the event was performed automatically, then the policy name is displayed.
- **Details:** Displays the details of the event.

  - Create Ticket (Manual and Automatic)
  - Alarm Actions such as Assign, Un-Assign, Acknowledge, and Unacknowledge (Manual and Automatic)
  - Update Status (Manual and Automatic)
  - Update through API (Create Ticket, Assign, Acknowledge, and Clear)
  - Add Annotation
  - Status (Success or Failure) of the automatic email notifications for both raw alarms and situations.
  - Status (Success or Failure) of the automatic webhook notifications for both raw alarms and situations.
  - Any updates to Southbound Gateway.

    NOTE

    The updates are displayed only for raw alarms and not situations.
  - Any updates to **earliestSourceAlarmURL** and **mostImpactedSourceAlarmURL** in the incident.
  - Any updates to the alarm fields using APIs.
  - Any alarm updates to the raw alarm or situation actions through APIs.
- **Event time:** Displays when the event occurred.
- **Elapsed time:** Displays the time elapsed.

Lifecycle Event for Maintenance Alarms

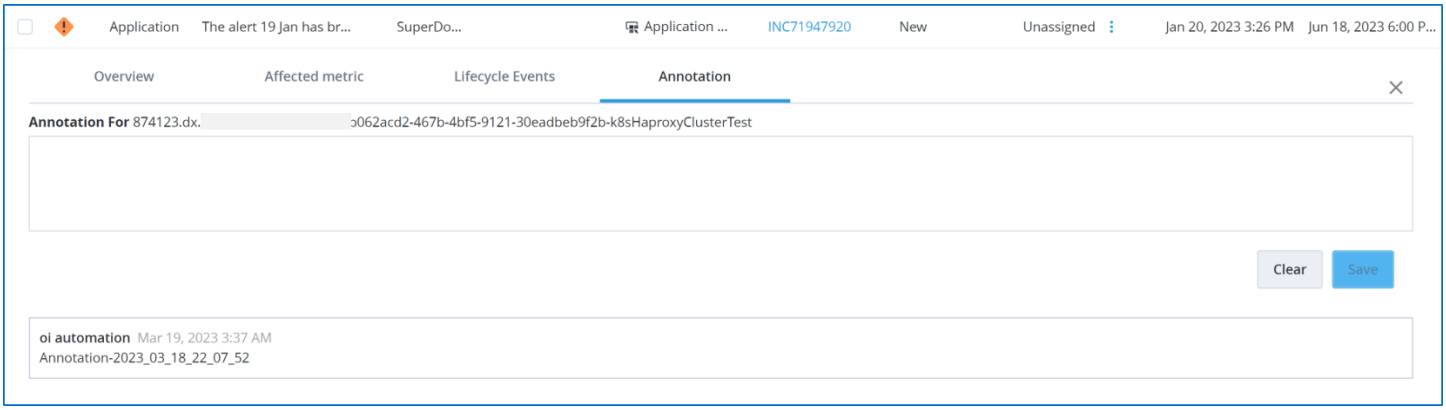
The **Details** column in the **Lifecycle Events** tab provides information on the alarms that are suppressed due to the maintenance window. Whenever the alarm is in maintenance and when the event is changed to True, a hyperlinked is added for the maintenance job which redirects to the Maintenance window. You can view lifecycle information for an alarm such as the reason for alarm suppression and details from the Maintenance Window definition.

NOTE

- If there are multiple maintenance windows for an alarm, the maintenance details are listed with a copy separated.
- If an alarm is part of multiple entities, then the service entity is taken precedence.

Annotation

The Annotation tab enables you to add any additional information or details that you want to add to the selected alarm. You can add the required information in the **Annotation** tab of the alarm and click **Save** to view the annotation as a part of the alarm details. The annotation includes the information that you added, the details of the person who added this annotation, the date, and time details.



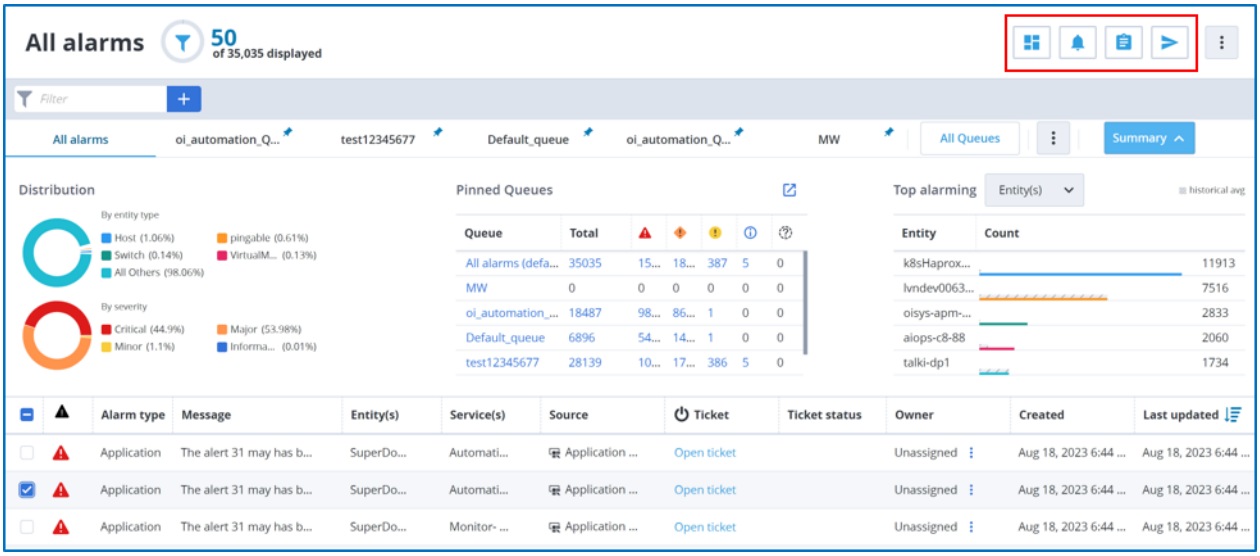
Logs

When an alarm is raised in DX Operational Intelligence by any product for a particular host and the given host is also sending syslog to Log Analytics, this tab is enabled. Click **Show Logs** to navigate to the **Log Analytics** page to visualize the logs. The Log Analytics page displays logs for the host for the last 15 minutes.

> **NOTE**
> If the log event is not enriched, then the **Logs** tab is disabled.

Alarm Actions

You can perform actions on alarms using these options:



Prerequisite:

- Integrate the source product with DX Operational Intelligence using Integration Gateway.

  > **NOTE**
  > - If the Southbound Gateway to Spectrum and UIM is configured, then only action updates are sent to the source products. Otherwise, alarm actions are within DX Operational Intelligence.
  > - Any Spectrum and UIM alarm action update in DX Operational Intelligence is sent to DX NetOps Spectrum and CA UIM products.

- **DX Dashboards (Only for All Alarms and Situations):** Click this icon to launch the **OI Alarm Metrics Overview** dashboard. This dashboard provides information about all alarms and situations at the tenant level, user level, and service level. For more information, see the OI Alarm Metrics Overview Dashboard section.
- **Alarm Management:** You can perform the following alarms actions for multiple alarms at the same time: Acknowledge, Assign to, Clear, Hide, Un-Acknowledge, Un-Assign, UnHide.

  > **NOTE**
  > For more information, see the Alarm Management section on this page.

- **Ticket Management:** Select the option to open the ticket manually.

  > **NOTE**
  > For more information, see the Ticket Management section on this page.

- **Trigger Channel:** You can notify users about an alarm directly from the Alarm Analytics page using these channels:
  - **Email:** You must configure the SMTP server to send emails to the recipient. Select one or more distribution lists to notify them about the alarm through email.

NOTE
If you do not configure the SMTP server, a success message appears but the email is not sent to the recipient.

– **Generic Webhook:** Select the configured webhook channel.
– **Slack:** Select the configured Slack channel.

Alarm Management

You can use the **Alarm Management** icon to manage alarms, acknowledge assigned alarms, and clear the assigned alarm. You can also clear the bulk alarms.

Follow these steps:

1. Navigate to the **All Alarms** page.
2. Select multiple alarms in the Alarms table and click the (bell) icon to perform bulk operations. Alternatively, click the **Ellipses** icon next to the owner of the alarm to perform a single alarm action.
   The Alarm Management menu appears. In this dialog, you can perform the following actions:
   - **Acknowledge:** Click the **Acknowledge** option to acknowledge the alarm. A green tick appears, which indicates that the alarm is acknowledged.
   - **Assign to:** Click the **Assign to** option and select the user to whom the alarm is to be assigned.
   - **Clear:** Click the **Clear** option to clear all alarms.
     NOTE
     To clear the Predictive alarms, use the Bulk Clear Alarms option.
   - **Hide:** Click the **Hide** option to hide the alarm detail in the database. The alarm in the user interface is grayed to indicate that the data is hidden in the database.
   - **Un-Acknowledge:** Click the **Un-Acknowledge** option to un-acknowledge for an alarm.
   - **Un-Assign:** Click the **Un-assign** option to remove the assignment for an alarm.
   - **Unhide:** Click the **UnHide** option to un-hide the alarm detail from the database.

NOTE

- Any alarm actions update in DX Operational Intelligence is not sent to CA ADA and DX APM products.
- For anomaly, prediction, and custom alarms, the alarm action updates are within DX Operational Intelligence.
- Clearing service alarms clears all the related alarms.
- If a root cause alarm is cleared, all the underlying alarms get closed and the service alarm gets closed.
- If a root cause alarm is cleared and the root cause alarms get generated, the service alarm does not get closed.
- Alarm action on multiple UIM or Spectrum setups is not supported.
- No alarm actions are supported for the Situation alarm.

The following table describes the supported alarm actions for different alarm types and source products:

Source Product	Alarm Actions
DX NetOps Spectrum	acknowledge, unacknowledge, ticket, assignment, unassignment, clear
DX UIM	ticket, assignment, unassignment, clear, Hide, UnHide
DX APM	acknowledge, unacknowledge, ticket, assignment, unassignment
CA ADA	acknowledge, unacknowledge, ticket, assignment, unassignment
Alarm Type	Alarm Actions
Anomaly alarm	acknowledge, unacknowledge, ticket, assignment, unassignment

Source Product	Alarm Actions
Custom alarm	acknowledge, unacknowledge, ticket, assignment, unassignment

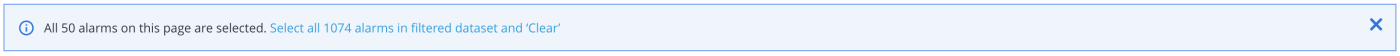
Bulk Clear Alarms

You can clear the alarms in bulk by performing these steps:

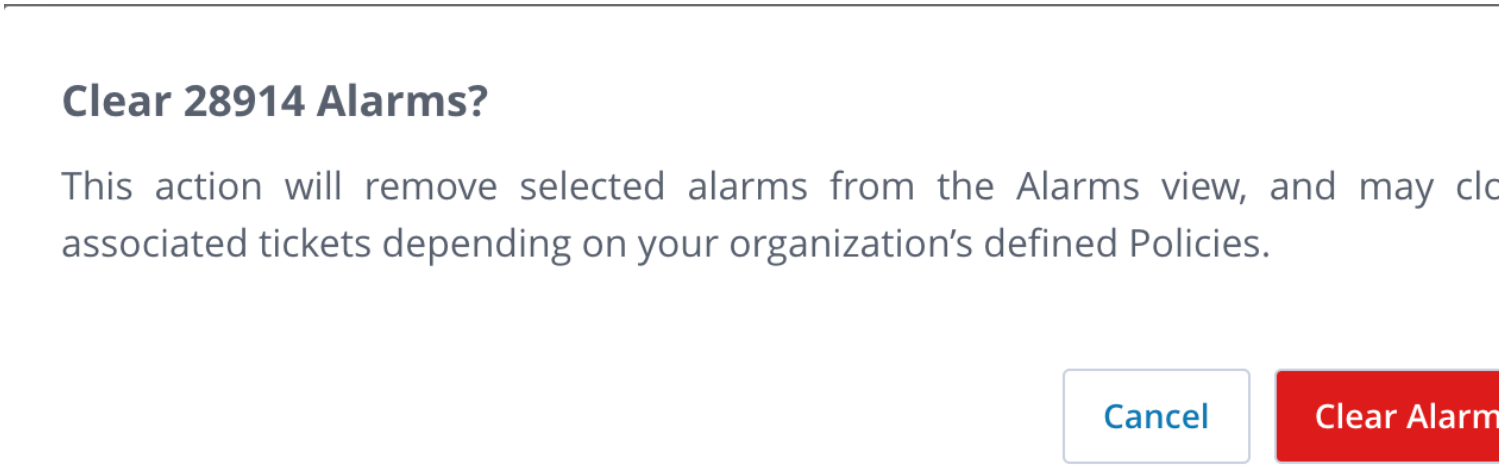
1. Select all the alarms listed on the page.

   NOTE

   If there are more than 50 alarms, by default, 50 alarms are displayed on the page.

   You see the following pop-up on top of the Alarm Analytics page: 

2. Click the hypertext to clear all the alarms. The following popup window appears:

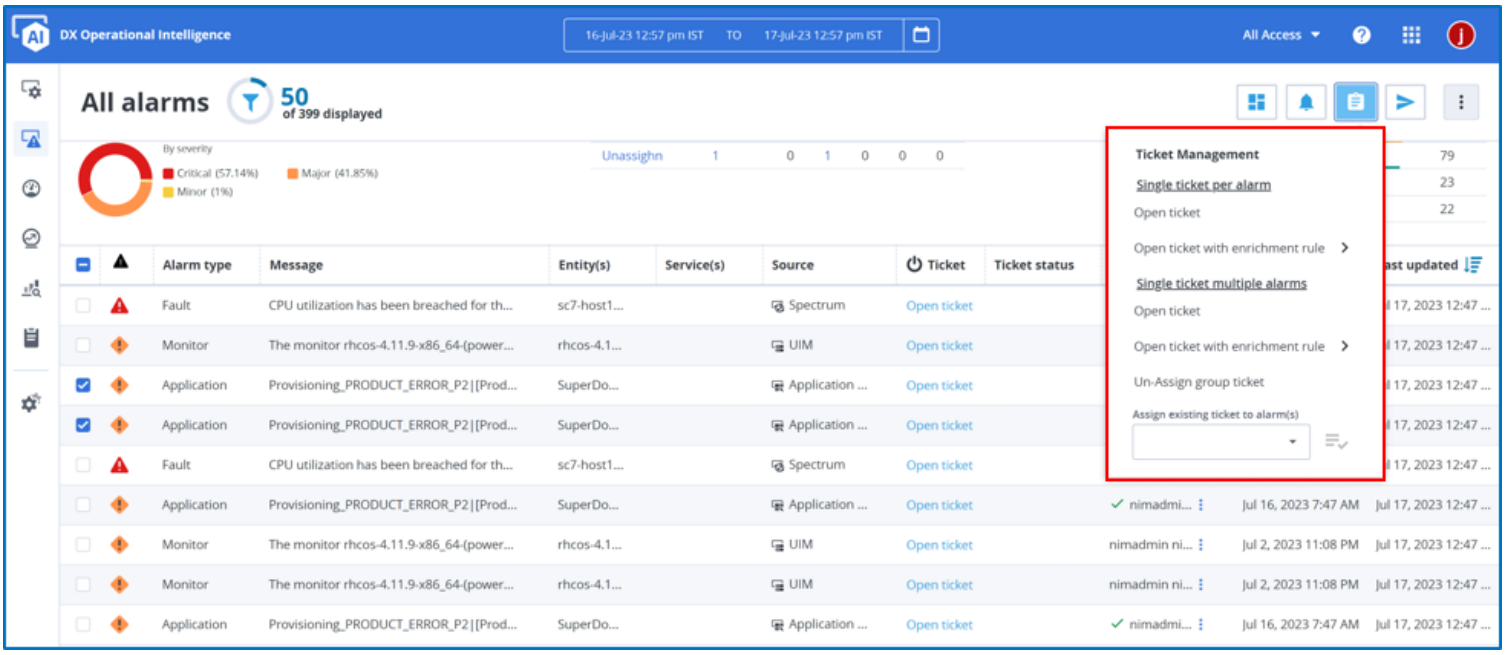


3. Click **Clear Alarms**. All the alarms get cleared.

Ticket Management

You can manually create tickets for All Alarms and Service Alarms from the Alarm Analytics page. You must configure the ITSM notification channel to manage the tickets update. You can create the tickets using one of the following options:

If multiple alarms with a similar root cause are ingested into DX Operational Intelligence, you can now create separate tickets for each of the alarms or you can create a single ticket for all those alarms using the **Ticket Management** icon shown in this image:



NOTE
The **Open Ticket** link is visible only when the ITSM notification channel is configured.

Follow these steps:

1. Navigate to the **All Alarms** page.
2. Select the alarms that you want to open tickets for.
3. Click the **Ticket Management** icon and select the required option:
   - **Single ticket per alarm:** Creates a separate ticket for each of the selected alarms. Each alarm has a different ticket ID.
     - **Open ticket:** If you open the ticket using this option, the mapping rule that is associated with the channel is used for the ticket enrichment.
     - **Open ticket with enrichment rule:** If you use this option, the mapping rule that you select in this list is used for enrichment instead of the rule that is associated with the channel.
   - **Single ticket multiple alarms:** Creates a single ticket for all the selected alarms. All the alarms will have the same ticket ID.
     - **Open ticket:** If you open the ticket using this option, the mapping rule that is associated with the channel is used for the ticket enrichment.
     - **Open ticket with enrichment rule:** If you use this option, the mapping rule that you select in this list is used for enrichment instead of the rule that is associated with the channel.

     NOTE
     Ticket Enrichment from CMDB/Enrichment Rule: For multiple alarms, the alarm with the highest severity and the oldest start time is considered for enrichment. For more information, see the Ticket Enrichment Rules section.
4. Click the ticket ID which redirects you to the ITSM ticket management system. You can view the detailed information about the ticket.

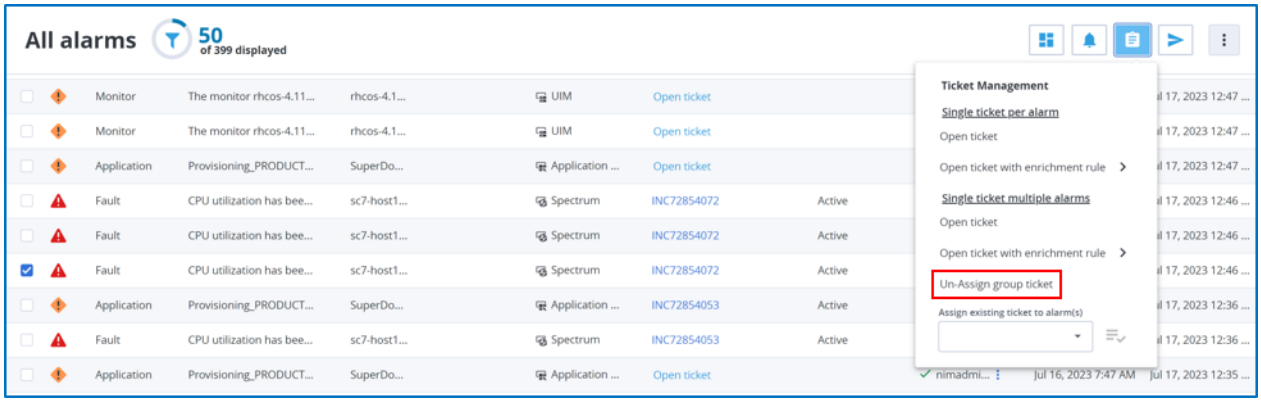
   NOTE

   - If you selected the**Single ticket multiple alarms** option, the ticket displays details for the alarm that has the highest severity and the oldest start time.

  - **Alarm ID:** Displays the alarm ID for the alarm that has the highest severity and the oldest start time.
  - **Group Member Alarms:** Displays the alarm IDs for all the alarms in the ticket group.
  - **Group Ticket URL:** Displays the alarm URL for the ticket group. Opening this URL displays all the alarms participating in this ticket group.
  - **Impact:** Displays the impact in the ticketing system which is based on the severity level of the alarm that has the highest severity and the oldest start time.
- Bi-directional updates such as synchronization of the updates, clearing alarms or closing tickets, and so on are supported.

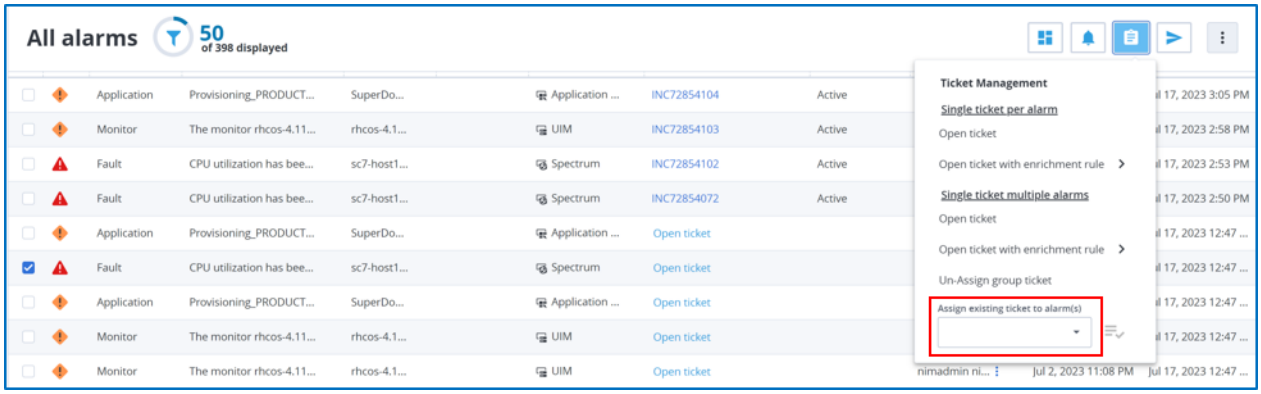
You can remove or add an alarm to the ticket group if required.

- A ticket must have at least one valid alarm linked to it. If the ticket group has only one alarm, you cannot unassign the alarm. To remove an alarm from the ticket group, select the alarm and then select **un-Assign group tickect** under Ticket Management.



  The alarm is removed from the ticket group and the **Ticket** column in the **Alarms** table is updated to **Open ticket**.
  If you remove the alarm that has the highest severity and the oldest start time from the ticket group,

  - Ticket is updated. For example, **Work Notes** in ServiceNow is updated to reflect this change. Additionally, the **Description** section is also updated to display the details of the next alarm that has the highest severity and the oldest start time.
  - The **Impact** field is automatically affected and is updated to the impact of the next alarm that has the highest severity and the oldest start time.
- To add an alarm/alarms to the ticket group, select the alarm/alarms and enter the ticket ID in the **Assign existing ticket to alarms** textbox. The alarms are added to the ticket group and the ticket ID is updated in the alarms table. In the ticketing system, the newly added alarm IDs are updated. For example, Work Notes in ServiceNow is updated with the newly added alarm IDs.



NOTE

- If **resolved** is selected in the channel configuration for bi-directional updates, you cannot assign an alarm to a resolved ticket ID. The resolved ticket IDs are not displayed. The **Ticket Status** column on the All Alarms page displays if the ticket is resolved.
- If **closed** is selected in the channel configuration for bi-directional updates, then IDs for the closed ticket are not displayed but only IDs for the resolved tickets are displayed. You can still assign the selected alarms to the resolved tickets.

- Moving an alarm to another ticket group: You can move an alarm from one ticket group to another using the **Assign existing ticket to alarms** option. However, you cannot move an alarm if the ticket group has only one alarm.
- The ticket is closed/resolved only when all the alarms in the ticket group are cleared in DX Operational Intelligence.

Insights

Click the **Insights** link to navigate to the Insights page where you can get insights into services and raw alarms. For more information, see the Insights section.

Delete an Alarm

To delete an alarm, select the alarm and click the icon. A confirmation window appears, click **Delete** button. You can perform a bulk delete operation by selecting multiple alarms and by clicking the **Delete** icon.

DX Operational Intelligence is a machine learning-driven, advanced analytics solution that is designed to help IT operations teams deliver a phenomenal user experience, improve service quality, and drive operational efficiencies. DX Operational Intelligence offers an ecosystem that transforms these alarms into meaningful information which you can leverage for managing your IT platforms better. Alarms Analytics enables you to identify alarms that have a larger impact by offering extensive monitoring. You can understand the clustering of alarms that represent the context of a problem and can tie the problems to impact on services and applications. Alarm Analytics helps you simplify your complex issues, raise the right tickets that should reach the right people by narrowing down the assignments that are based on domain, service, context, and so on, and should get quick resolutions. Alarms Analytics gives you insights on the root cause of alarms, and the context, which you can leverage to create patterns and address common problems by using automation, which allows you to focus on the complex ones.

By using Alarm Analytics in DX Operational Intelligence, you gain the following capabilities:

- Reduce alarm noise from multiple products.
- Correlate alarms across products to identify the root cause.
- View probability bands to determine buildup to an alarm.
- Fine-tuning alarm threshold by analyzing the historical pattern.

Alarms are classified into the following categories:

- **Anomaly Alarms:** An anomaly alarm gets generated when a metric value deviation is detected by the Data Science Engine for the configured metrics, by using machine learning algorithms. This alarm is generated when a threshold is crossed for the configured metric value.
- **Service Alarms:** For new tenants, the Service Alarms is disabled. To re-enable the Service alarms for new tenants, contact Broadcom Support. A service alarm is a group of alarms that affect one or more business services and are related to an incident, which is identified by the time it occurred and its root cause. The root cause is the alarm on the

topologically deepest device in the affected business service. All the situations that are reported by alarms in the group are due to the identified root cause.

- Raw Alarms: Alarms that are generated from source products such as CA Unified Infrastructure Management, DX NetOps Spectrum, CA ADA, and DX APM or any custom data source.
- Situations Alarms: Alarms are grouped based on context using machine learning algorithms. Clustering clubs alarms together based on distinct dimensions and groups them together for triage or further analysis. Thus, clustering enables users to filter through a huge number of alarms and analyze contextually relevant alarms.

{"URL":["https://digital-oi/alarms-analytics/allAlarms","https://digital-oi/alarms-analytics/allAlarms/affectedMetric","https://digital-oi/alarms-analytics/allAlarms/impactedServices","https://digital-oi/alarms-analytics/allAlarms/topology","https://digital-oi/alarms-analytics/allAlarms/lifecycleEvents","https://digital-oi/alarms-analytics/allAlarms/annotation","https://digital-oi/alarms-analytics/allAlarms/alarmTab"],"description":"concept.dita_4e6dbef3-5911-42bd-81dc-4c23ddd2620a","troubleshooting":{"masterkb":"https://knowledge.broadcom.com/external/article?articleId=226446"},"customCards":[{"id":"IPCE_PinnedQueues","type":"configure","title":"Alarm Queues"},{"id":"IPCE_Filters","type":"configure","title":"Alarm Filters"},{"id":"IPCE_AllAlarmsDetails","type":"configure","title":"Alarm Details"}]}

Alarms Inspector

When an Application alarm is generated by DX Application Performance Management, there may be other alarms generated in the same time window which are related to the source alarm. Using the Alarms Inspector, you can view these associated alarms and possible suspect alarms, their details and also the metrics, and their corresponding topology from where the alarms are generated. You can use this information to perform deep triage of an alarm.

- **Suspect Alarms:** Alarms from DX Application Performance Management that are the root cause of the incident.
- **Related Alarms:** Alarms that are related to the source alarm but are in a different location in the topology.

The Inspector also ranks the alarms based on their topological structure which helps in determining the hotspot.

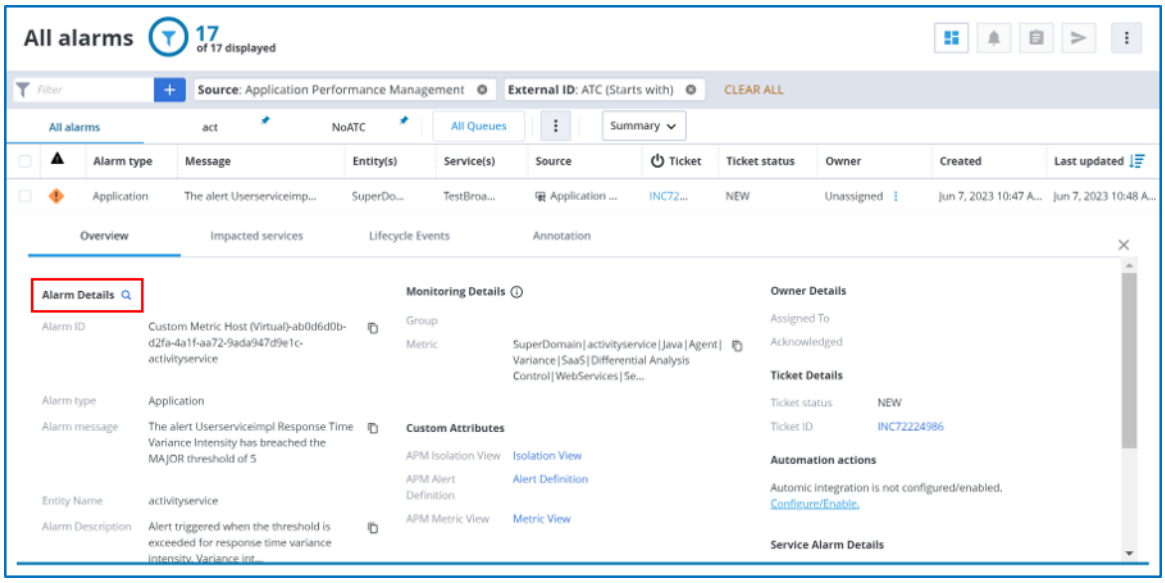
Access Alarms Inspector

You can navigate to the Alarms Inspector view from the All Alarms page.

Follow these steps:

1. Navigate to the **All Alarms** page.
2. Use the following filter on the **All Alarms** page:
   - **Source: Application Performance Management**
   - **External ID: Starts with ATC**

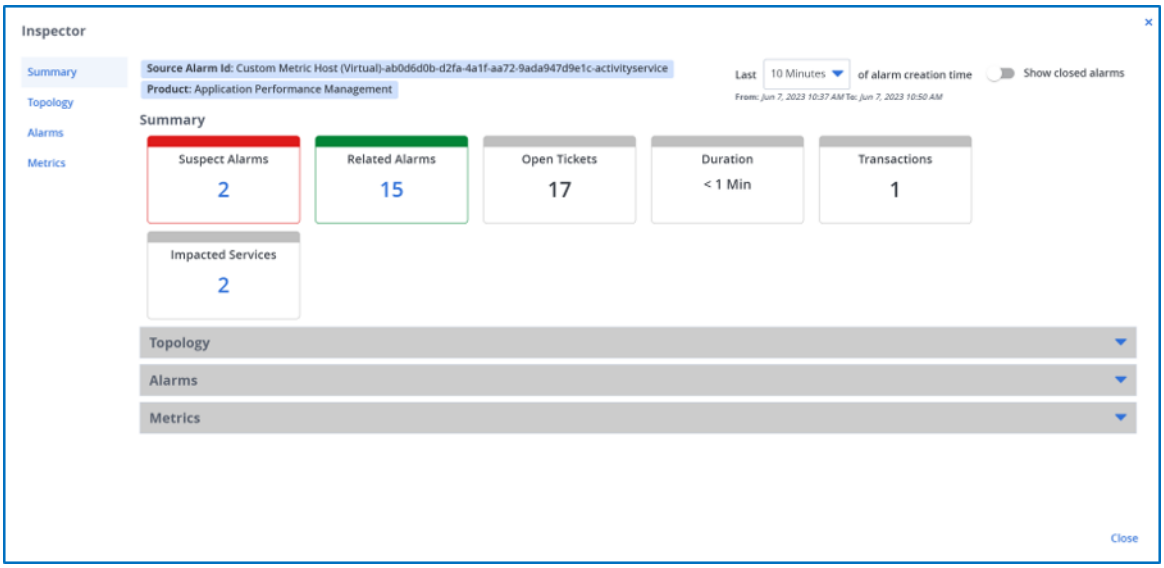
   The All Alarms page displays all the alarms associated with the application generated by DX Application Performance Management.
3. Click the required alarm from the list to display the **Alarm Details**.
4. Click the **Lens** icon to open the Inspector view.



The Inspector view opens in a window and displays all the associated alarms and their metrics to help diagnose the issue faster.

Alarms Inspector View

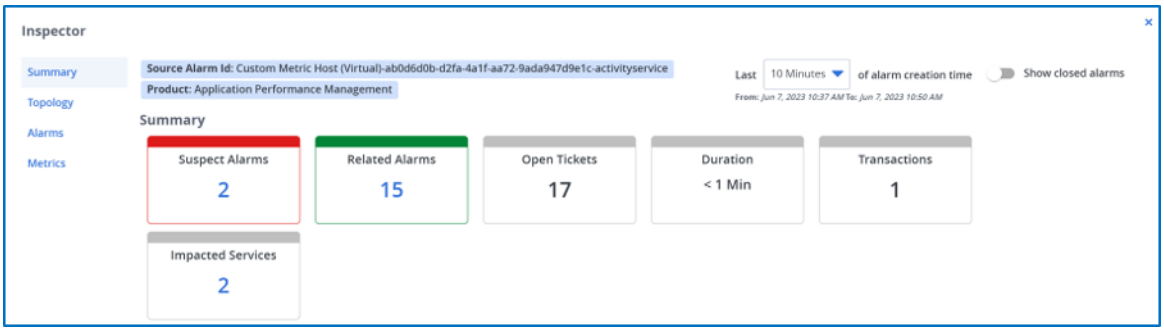
The Inspector view is divided into the following sections:



- Summary
- Topology
- Alarms
- Metrics

Summary

The Summary section provides an overview of the alarms based on the topology that is generated from the source alarm.

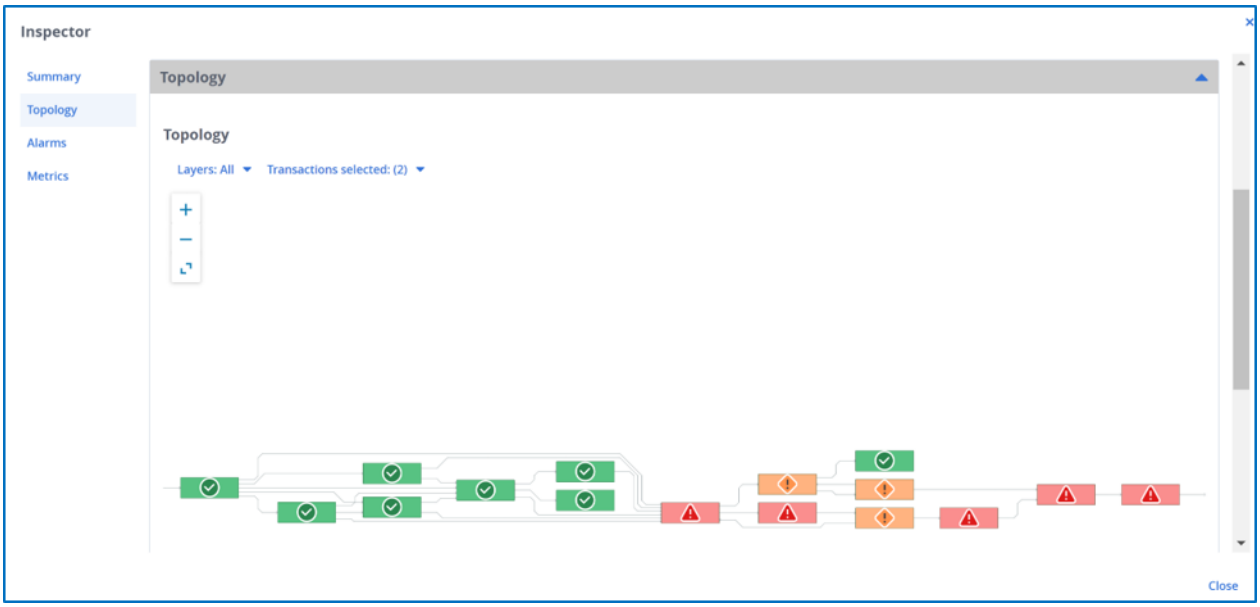


The following information is displayed in the Summary section:

- **Source Alarm Id:** ID of the selected source alarm.
- **Product:** Name of the source product.
- **Last <duration> of the alarm creation time:** Duration of the inspector window.
  - **For Open Alarms:** Duration is [Current time - Time specified]. For example, if the current time is 10:30 am and you select the last 30 minutes, then the Inspector view displays the data from 10:00 am.
  - **For Closed Alarms:** Duration is [Closed time - Time specified]. For example, if the alarm was closed at 7:30 pm and you select the last 10 minutes, then the Inspector view displays data from 7:20 pm to 7:30 am. Supported Values: 10 Minutes, 15 Minutes, 30 Minutes, 1 Hour, 2 Hours, 6 Hours, 12 Hours, and 24 Hours
- **Show Closed Alarms:** Enable this option to also include data for the closed alarms.
- **Summary:**
  - **Suspect Alarms:** Displays the number of suspect alarms. The details of these alarms are available in the Alarms section. Click the number to navigate to the Alarms section.
  - **Related Alarms:** Displays the number of related alarms. The details of these alarms are available in the Alarms section. Click the number to navigate to the Alarms section.
  - **Open Tickets:** Displays the number of open tickets for the alarm.
  - **Duration:** Duration is [Latest end time of the associated alarms - Earliest start time of the associated alarms].
  - **Transactions:** Displays the number of business transactions.
  - **Impacted Services:** Displays the number of impacted services. Click the number to view the names of the services that are impacted.

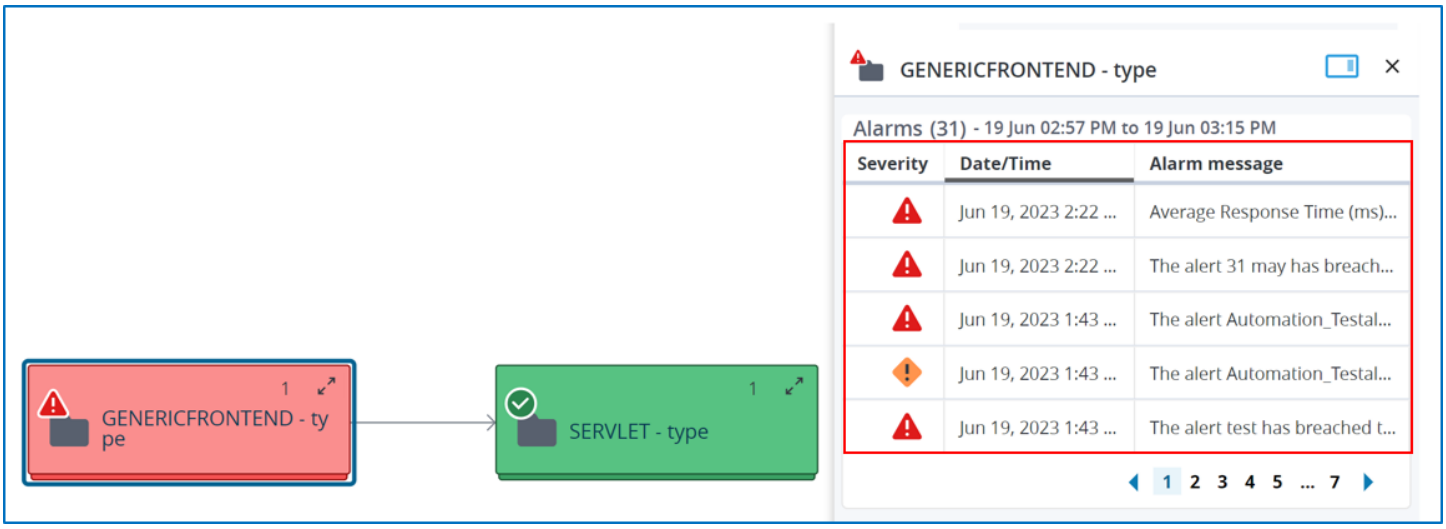
Topology

Topology is a graphical representation of how the application components are placed. In this section, you can view the business transactions and the infrastructure-related topology of the source alarm.

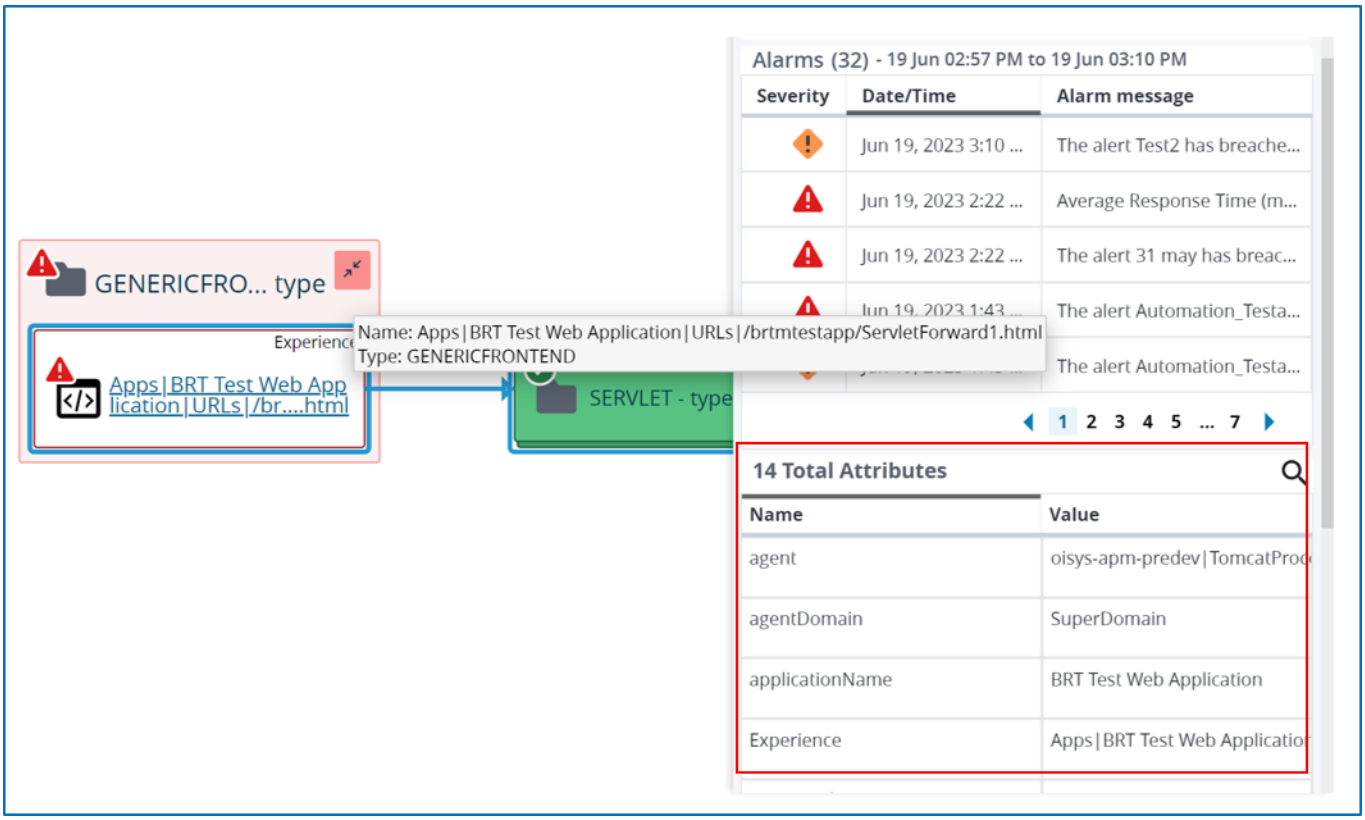


The Inspector ranks the alarms based on their topological structure and displays the entire topology associated with the alarm groups for triaging. The alarms generated on the right side are ranked higher and considered suspect alarms. All the other alarms are considered related alarms.

Click any entity to view the associated alarm details and their attributes. Click the topology to view the alarms in the group.

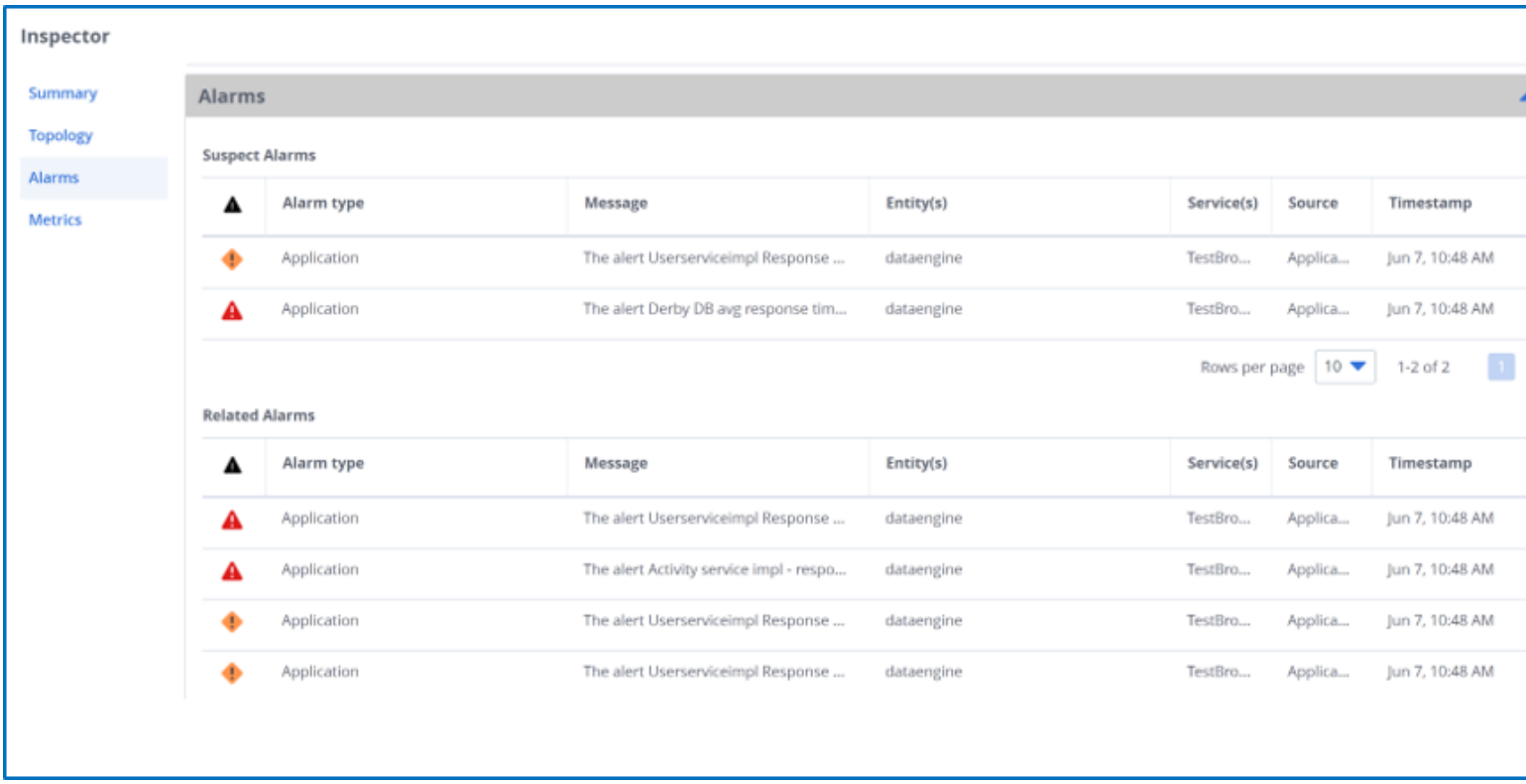


Expand the entity and click the individual vertex from the group to view the attributes.



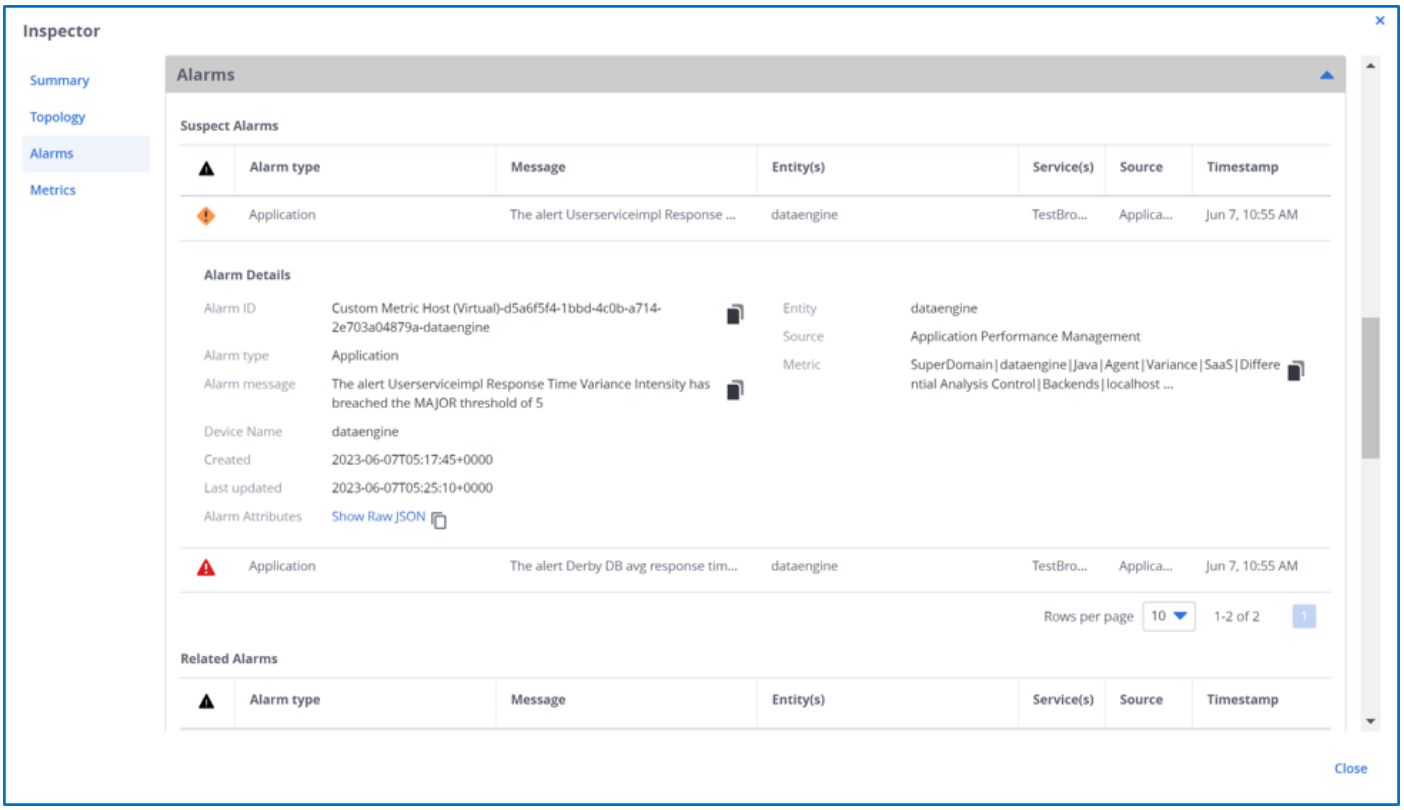
Alarms

The Alarms section provides the following details for the Suspect and Related alarms:



- Application: Name of the application Message: The alarm message. Entity: Name of the entity. Services: Name of the impacted service. Source: Name of the source product. Timestamp: Timestamp of when the alarm was generated.

The source alarm is highlighted in bold. Click any alarm to view the details for the associated alarms as shown in this illustration:

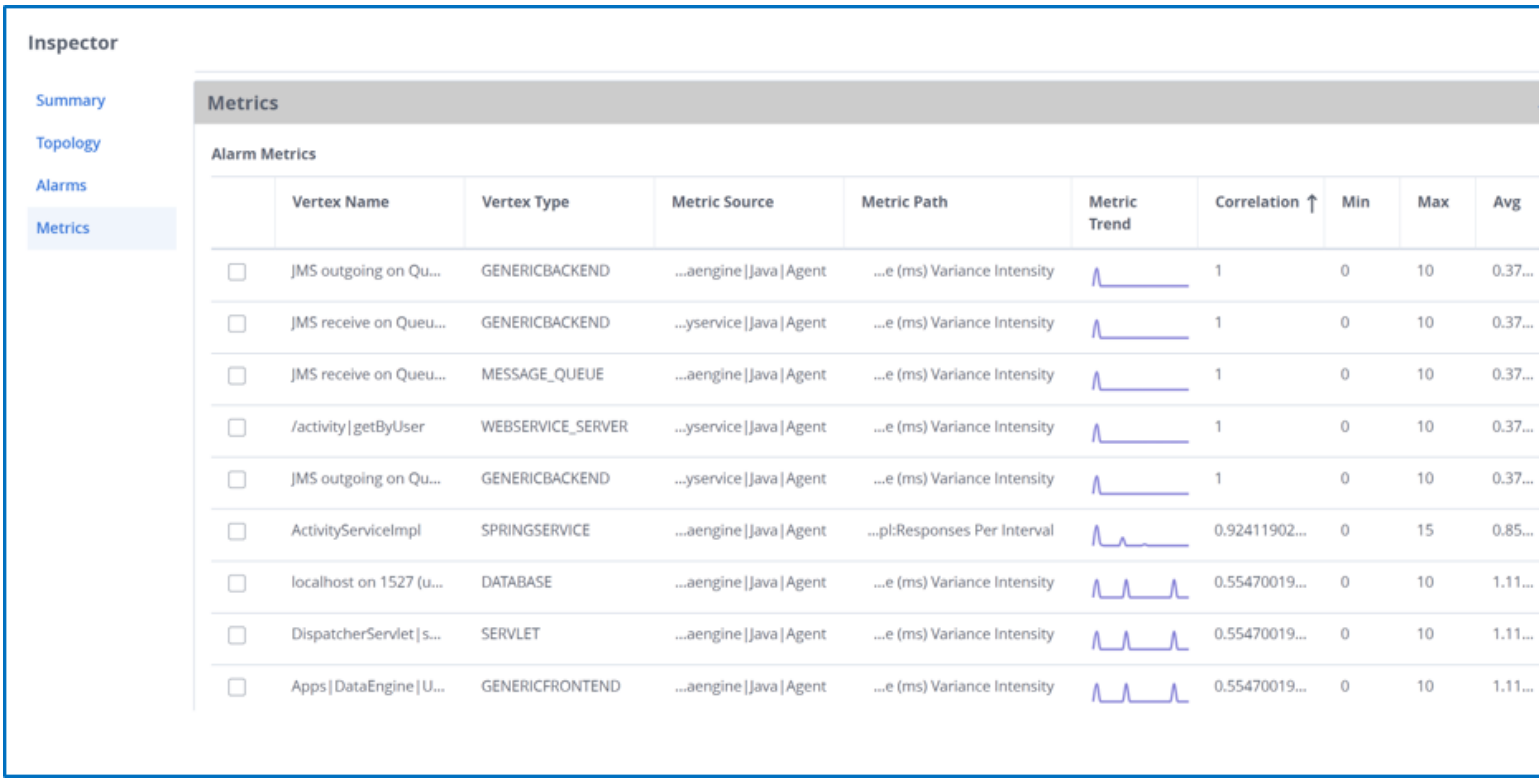


Metrics

The Metrics section is divided into the following sections:

- **Alarms Metrics:** Metrics that are associated with the generated alarms are displayed under Alarm metrics.
- **Topology Metrics:** Metrics that are associated with the entire topology are displayed under Topology metrics.

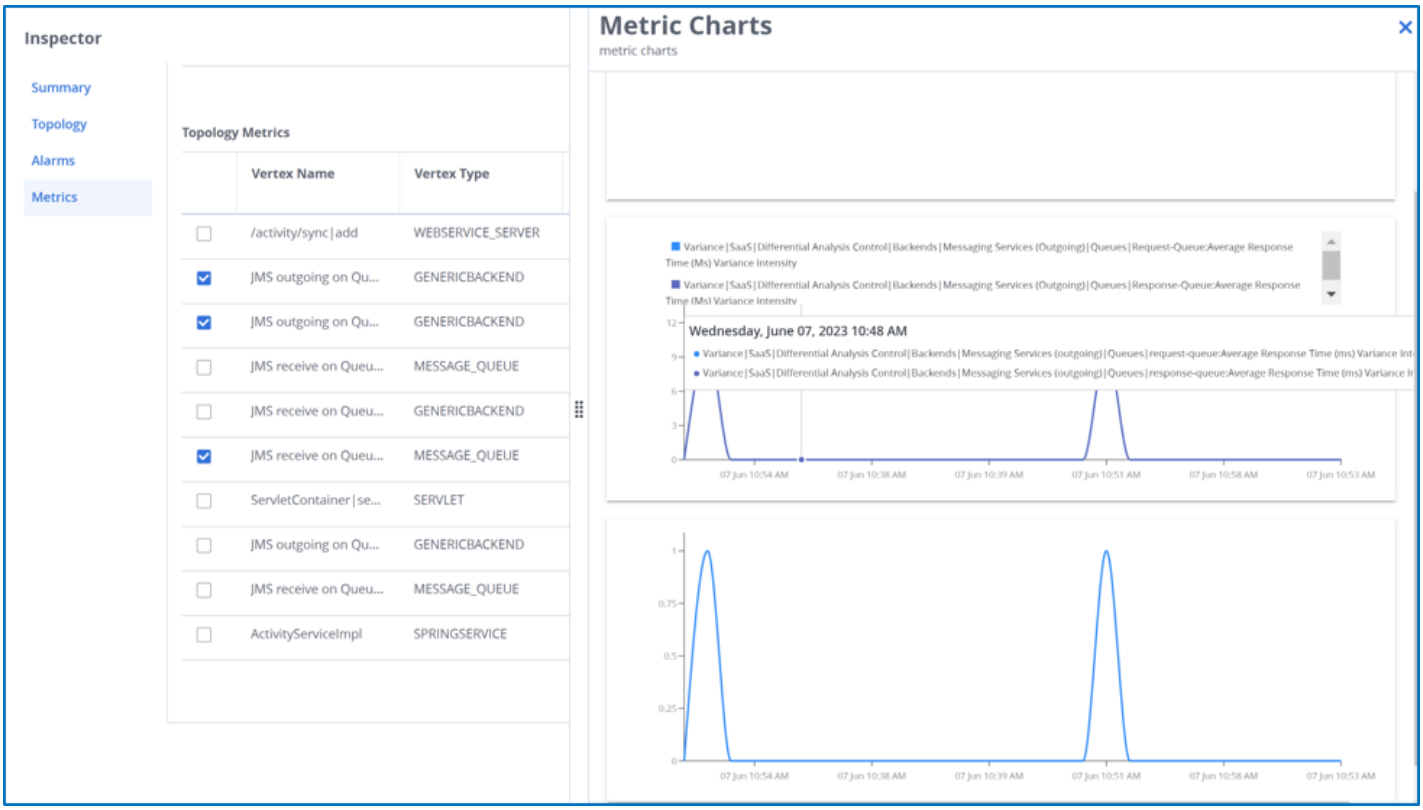
You can view the following details for the alarm and topology metrics. The following image illustrates the details of the Alarm Metrics:



Vertex Name	Vertex Type	Metric Source	Metric Path	Metric Trend	Correlation	Min	Max	Avg
JMS outgoing on Qu...	GENERICBACKEND	...aengine|Java|Agent	...e (ms) Variance Intensity		1	0	10	0.37...
JMS receive on Queu...	GENERICBACKEND	...yservice|Java|Agent	...e (ms) Variance Intensity		1	0	10	0.37...
JMS receive on Queu...	MESSAGE_QUEUE	...aengine|Java|Agent	...e (ms) Variance Intensity		1	0	10	0.37...
/activity|getByUser	WEBSERVICE_SERVER	...yservice|Java|Agent	...e (ms) Variance Intensity		1	0	10	0.37...
JMS outgoing on Qu...	GENERICBACKEND	...yservice|Java|Agent	...e (ms) Variance Intensity		1	0	10	0.37...
ActivityServiceImpl	SPRINGSERVICE	...aengine|Java|Agent	...pl:Responses Per Interval		0.92411902...	0	15	0.85...
localhost on 1527 (u...	DATABASE	...aengine|Java|Agent	...e (ms) Variance Intensity		0.55470019...	0	10	1.11...
DispatcherServlet|s...	SERVLET	...aengine|Java|Agent	...e (ms) Variance Intensity		0.55470019...	0	10	1.11...
Apps|DataEngine|U...	GENERICFRONTEND	...aengine|Java|Agent	...e (ms) Variance Intensity		0.55470019...	0	10	1.11...

- **Vertex Name:** Displays the vertex name.
- **Vertex Type:** Displays the type of vertex.
- **Metric Source:** Displays the source of the metric.
- **Metric Path:** Displays the metric path.
- **Metric Trend:** Displays the metric trend.
- **Correlation:** Displays the current correlation of the metric trend with the source metric where the source alarm is generated. Value close to 1 indicates that the current metric is highly correlated with the source metric. Value close to 0 indicates that the current metric is less correlated with the source metric.
- **Min:** Displays the minimum value of the metric during the specified time.
- **Max:** Displays the maximum value of the metric during the specified time.
- **Avg:** Displays the average value of the metric during the specified time.

In this section, you can also view the metric chart for both the alarm and topology metrics. Select the metric to view the metric chart as shown in the illustration:



Metric-Based Alert Configuration

You can configure static threshold-based alerts for metrics from APM, UIM, Spectrum, CA APM, or any third-party metrics that are ingested into DX Operational Intelligence. To configure the alerts, group the metrics by attributes using the **Metric Groups** tile on the **Settings** page and then set the Critical and Major thresholds for these groups using the **Alarms** tile on the **Settings** page.

This section provides the following information:

- Metric Groups
- Configure Alarms for Metrics Groups

Metric Groups

DX Operational Intelligence provides a few metric groups out-of-the-box for the connectors. You can edit these metric groups as required and you can also create metric groups for other metrics.

- Required Privileges
- Access Metric Groups
- Out-of-the-box Metric Groups
- Create a Metric Group

Required Privileges

By default, a Tenant Administrator and Power User have the required privileges for metric groups. However, a user with a custom role must be granted the following privileges as required.

Follow these steps:

1. Navigate to the **Settings > Roles** page.
2. Click **+ New Role**.
3. Select the required privileges under **DX Platform > Settings > Metric Groups**:
   - **View Metric Groups**
   - **Create or Update Metric Group**
   - **Delete Metric Groups**

Access Metric Groups

You can access the **Metric Groups** tile from the **Settings** page.

Follow these steps:

1. Log in to DX SaaS.
2. Click **Settings** in the left navigation pane.
3. Click the **Metrics Groups** tile under **Manage Alarms**.
   DX Operational Intelligence displays the out-of-the-box metric groups. For each of the metric groups, information such as **Group Name, Description, Filters, Creation Time, Last Editor, and Last Updated** is displayed. You can click the group name to view the metrics in the group. For each metric in the group, information such as the metric name, source, metric path, and last seen is displayed. To view the metric chart for the metric, select the metric in the list.

   NOTE
   You can sort the information by all the columns except for the Filters column.

Out-of-the-box Metric Groups

DX Operational Intelligence provides the following metric groups out-of-the-box for the connectors:

- Connector Host Memory Used Metric Group
- Connector Uptime Metric Group
- IntervalSinceLastSuccessfulPush Metric Group

You can edit and also delete these groups.

Create a Metric Group

You can use the filter to group the metrics and then create a metric group. By default, the following filter attributes are available:

- Metric
- Source
- Metadata Attributes
- Custom

The following operators are supported by the filter attributes: Equals, Not equals, Contains, Does not contain, Starts with, Does not start with, Ends with, Does not end with, and Regular Expression.

Follow these steps:

1. Navigate to the **Settings** page in DX SaaS.
2. Click **Metrics Groups** (to organize metrics into sets) under **Manage Alarms**.
   The **Metric Groups** page is displayed. This page displays out-of-the-box metric groups and any metric groups if created.

3. Click **+ New Metric Group**.
   All the metrics that are ingested into DX Operational Intelligence are displayed. The following information is displayed for each of the metrics: **Metric Name**, **Source, Metric, and Last Seen.**
4. Click to select the filter or enter the filter attribute and select the operator. For example, select the filter as Metric Name Contains CPU Utilization.
   The filtered metrics list is displayed.
5. Select the metric to display the metric chart for the last 1 hour.
6. Click **Save**.
7. Enter the **Group Name** and **Description**.
8. Click **Save**.
   The metric group is created. You can set the critical and major alarm thresholds using the **Alarms** tile.

Configure Alarms for Metrics Groups

After you create the metric group, you can set static thresholds to configure Critical and Major alarms using the **Alarms** tile on the **Settings** page. An alarm is generated whenever the thresholds are met. By default, the critical and major alarms are pre-configured for the out-of-the-box metric groups (**Connector Host Memory Used Metric Group, Connector Uptime Metric Group, and IntervalSinceLastSuccessfulPush Metric Group**). You can edit these alarms or you can create alarms.

- Required Privileges
- Access Alarms
- Configure Critical and Major Alarms for a Metric Group
- View Alarms in Alarm Analytics

Required Privileges

By default, a Tenant Administrator and Power User have the required privileges for alerting service. However, a user with a custom role must be granted the following privileges as required.

Follow these steps:

1. Navigate to the **Settings > Roles** page.
2. Click **+ New Role**.
3. Select the required privileges under **DX Platform > Settings > Alerting Service**:
   - **View Alert Config**
   - **Create or Update Alert Config**
   - **Delete Alert Config**

Access Alarms

You can access the **Alarms** tile from the **Settings** page.

Follow these steps:

1. Log in to DX SaaS.
2. Click **Settings** in the left navigation pane.
3. Click the **Alarms** tile under **Manage Alarms**.
   The Alarms page is displayed with the alarm details for the out-of-the-box metric groups and for other metric groups if created.

Configure Critical and Major Alarms for a Metric Group

The following illustration shows how to configure the critical and major alarms for a metric group.

Follow these steps:

1. Navigate to the **Settings** page in DX SaaS.
2. Click **Alarms** under **Manage Alarms**.
   The Alarms page is displayed.
3. Click **+ Alarm**.
   The **Create Alarm** page is displayed.
4. Provide the following information:
   - **Active:** Click to enable the alarm.
   - **Alarm Name:** Enter a name for the alarm.
   - **Description:** Enter a description for the alarm.
   - **Alarm Message:** The default alarm message is displayed. You can customize the alarm message using the metric attributes. Press $ to view the list of metric attributes. For example, you can add the following message using the attributes. ${metricName} on host ${hostname} has breached the ${severity} threshold of ${thresholdValue}.
   - **Alarm Type:** Enter the alarm type. If you do not specify the alarm type, then the default value is applied. **Default:** Application.
   - **Metric Group:** Select the existing metric group from the list or you can create a metric group for which you want to set the thresholds. For a new user, only the OOTB metric groups are displayed. When you select the metric group, all the metrics in the metric group are displayed in the right panel. However, the metric chart is displayed only for the top three metrics in the list. You can select a maximum of 10 metrics to be included in the preview.

     NOTE
     Data is displayed for 8min by default. You can preview the data for 1hr, 1d, 1w, and 1m.
   - **Resolution:** Select the value from the list. **Values:** 15 seconds, 30 seconds, 1 minute, 2 minutes, 6 minutes, 12 minutes, 24 minutes, 48 minutes, 1 hour, 168 minutes, and 12 hours

     NOTE

     - Resolution is the interval at which the configured thresholds are checked. For example, the metric data is pushed every 5 mins, but the resolution is set as one hour. Then for the hour, 12 data points are aggregated and then checked at hourly frequency.
     - The alert resolution must be **>=** to the interval of the raw metric data.
   - **Combination:** Select the value:
     - **Any:** An alarm is triggered if any metric in the metric group breaches the threshold.
     - **All:** An alarm is triggered if all the metrics in the metric group breach the threshold.
   - **Comparison Operator:** Select the operator. **Values:** Greater Than, Less Than, Equal To, Not Equal To

     NOTE
     This operator defines at what condition the threshold is considered as breached. For example, if the major threshold is 80 and the operator is **Greater Than**, then any value greater than 80 is considered a breach.
   - **Thresholds:**
     - **Critical:** Define the following information:
       - **Threshold:** Enter the threshold value for the critical alarm.

NOTE

- If the **Comparison Operator=Greater Than**, then the threshold for Critical Alarm must be greater than Major Alarm.
- If the **Comparison Operator=Less Than**, then the threshold for Critical Alarm must be lesser than Major Alarm.

- **Periods Over Threshold:** Enter the number of occurrences to generate the alarm.
- **Observed Periods:** Enter the number of periods within which the threshold should be reached to generate the alert.

For example, if Threshold=3, Periods Over Threshold=4, and Observed Periods=10, a critical alarm is triggered whenever four periods/datapoints of 10 periods/datapoints are greater than three.

- **Major:** Define the following information:
  - **Threshold:** Enter the threshold value for the major alarm.
  - **Periods Over Threshold:** Enter the number of occurrences to generate the alarm.
  - **Observed Periods:** Enter the number of periods within which the threshold should be reached to generate the alert.

– **Notification Policy:** Select the notification policy or you can create a policy. To create a policy, click **+ Add New Notification Policy** and provide the following information:
  - Policy Name
  - Trigger Alarm Notification
  - Channel
  - Message Template to Use

5. Click **Save**.
   The alarm is created for the metric group and is displayed on the Alarms page. The following information is also displayed:
   Enabled, Status (Major Threshold Breached, Critical Threshold Breached, and Normal), Name, Description, Metric Group, Critical Threshold, Major Threshold, Resolution, Last Editor, Last Updated, Actions (Delete).
   Click **>** next to the alarm name to expand and view the metric group filter. You can click the alarm name to view the alarm configuration and also the metrics. By default, a preview is displayed only for the first three metrics. You can select additional metrics to be included in the preview.

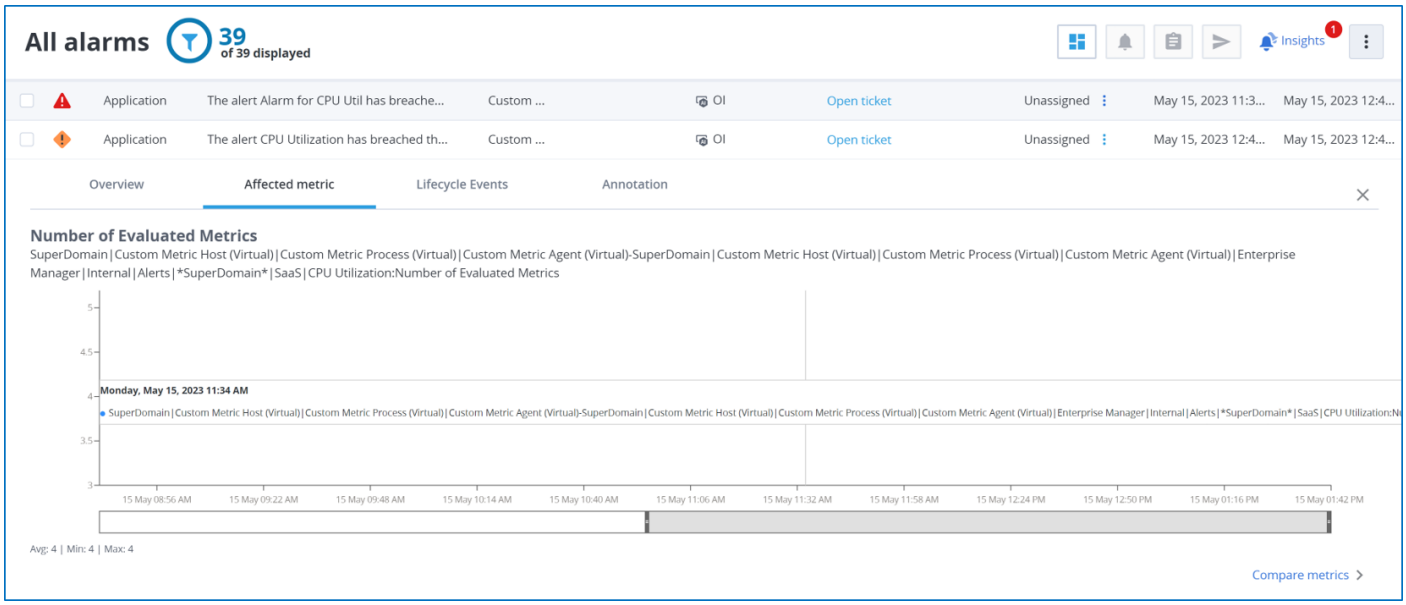
   NOTE
   You can enable or disable the alarms. If you disable the alarm, only the alarm notifications are stopped even if the problem exists but the alarm is not closed.

View Alarms in Alarm Analytics

To view the alarms for the metrics, filter the alarms by **source=OI**.

Follow these steps:

1. Navigate to the **Alarms** page.
2. Open the **All Alarms** page.
3. Filter the alarms by **source=OI**.
4. Open the alarm.
   The **Affected Metric** tab displays the details as shown.



5. Click the **Compare Metrics** link to navigate to the **Performance Analytics** page.

Situation Alarms

Situation Alarms are a collection of alarms that represent an incident. Incidents impact application or datacenter health.

Situation Alarms offers you insights into the different situations that might cause major hassles in your product and empowers you with the capability to analyze the different patterns and contain them. Situations are created using machine learning-based clustering algorithms employing time correlation, topological relationship, and natural language processing for analysis.

Situation alarms support clustering, combining the alarms together based on distinct dimensions and grouping them together for triage or further analysis. Clustering enables you to consolidate and group a huge number of alarms and analyze alarms that are contextually relevant. For example, a collection of alarms representing an incident impacting applications or data center health.

Situation alarms leverage dimensions such as Text, Active Times Series, Host, and Historical Time Series for creating situation clusters In addition, Situation alarms are now enhanced to support the Service dimension, which helps you define multi-hierarchy services using topologies across different monitoring domains. This enables an added dimension to enrich the alarms triage for the improved noise reduction of the issues. You can now perform situation analysis using Service Topology. For more information, see Service Topology.

> **NOTE**
> The tenant administrator can configure the situation clustering dimensions using the APIs. For more information, see Situation Clustering Dimensions APIs.

Clustering Types in DX Operational Intelligence

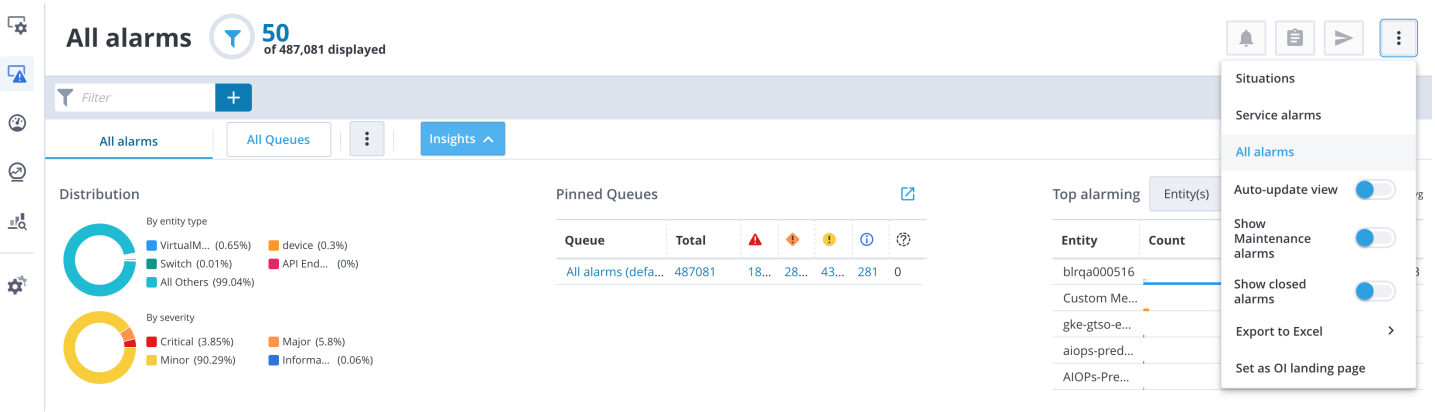
DX OI supports the following clustering types:

- Custom Situation Definitions (Custom): DX Operational Intelligence supports creating custom rules for situation clustering. Using the custom clustering type, you can define the predictable rules that meet your organization-specific business objectives.
- Algorithmic: DX Operational Intelligence provides generic global policies based on the preconfigured algorithms for situation clustering. Situation clustering based on global policies restricts organizations from adding organizational-

specific rules. If you do not configure the custom clustering type, the application uses algorithm-based global policies for situation clustering.

Access Situation Alarms

To view Situation Alarms, click on the **Alarm View Filter** on the top-right, in the context-menu select **Situations** to view the situation alarms that are created in the selected period.

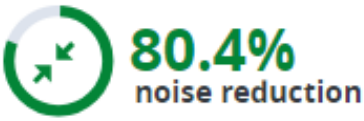


Situation Alarms User Interface

The Situation Alarm details are as follows:

- Displays the total count of situation alarms that are created from the total number of alarms.

- Displays the overall percentage of noise reduction. You can also filter and view the noise reduction for a selected filter attribute.
  For example, you can filter and view the noise reduction percentage for the selected service and the specified time range.



- The **Situations** view contains the following details:

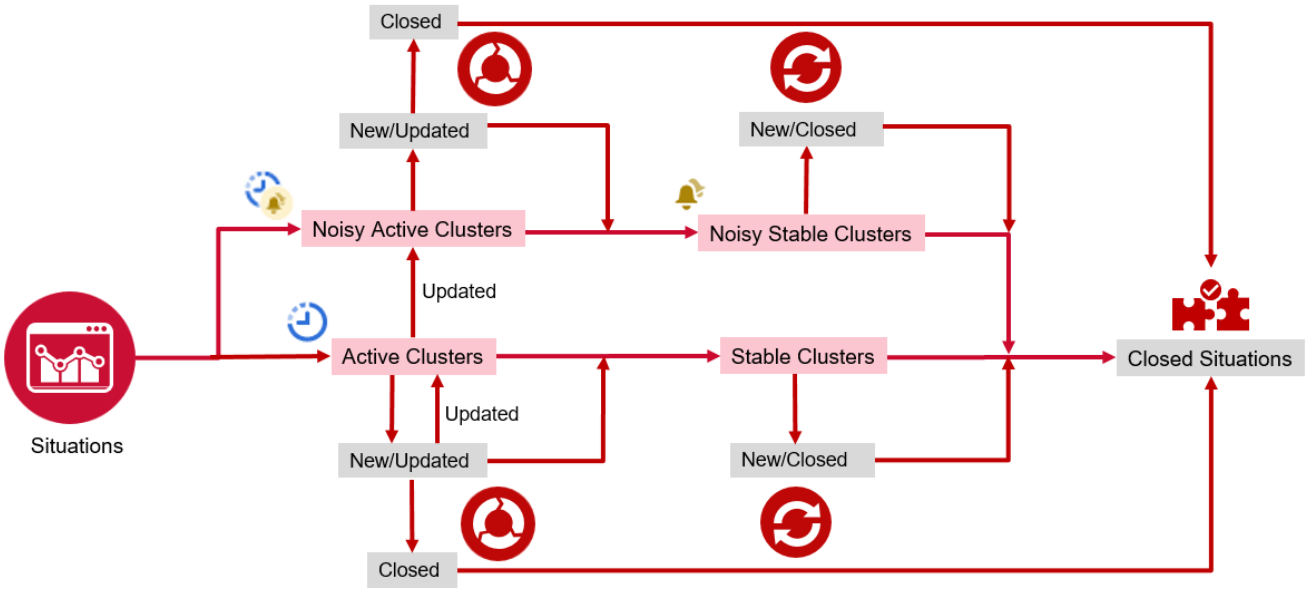
Column Name	Description
Column / Row-level Alarm Action	Enables you to perform row-level alarm actions or column-level to perform bulk alarm actions.
Severity	Indicates the severity of an alarm. The following colors indicate the severity: • Red: Critical • Orange: Major • Yellow: Minor • Light Blue: Informational • Black: Warning
Situation ID	Indicates the ID of the situation alarm
Sub-Clusters	Indicates the number of underlying alarm clusters.
State	Indicates the current state of the situation alarm.
Message	Displays the description for a situation alarm. The alarms which are in **Stable Noise Cluster** state display the dimension details of the cluster as the alarm message, which is created in the following order of precedence, provided the weights of the dimensions are the same: 1. Text 2. Time 3. Host 4. Service 5. Historical If the weights of the dimensions are different, the dimensions which have the higher weights or similarities are displayed first.
Entity	Indicates the device or application name.
Service	Displays the service that is impacted by an alarm.
Source	Displays the source product from which the alarm is generated.
Ticket	Displays the ID generated by the ticketing system. Click the **Open ticket** link to open a **ServiceNow** ticket corresponding to the alarm. For more information on tickets, see Ticket Management.
Last Updated	Displays the date and time when the alarm was last updated.

- A Situation Cluster can be in one of the following **states**:
  –

  Active Cluster: A situation cluster is in an Active state when the alarm keeps getting updates or there are reoccurrences of the alarm. Active cluster is defined to be still evolving to gather possible additional templates (sub-clusters) as well as possible root cause alarms. Active clusters are controlled by the Situation Window setting in the Situations view. If there are no further updates for a preset time period, the situation cluster transitions from an Active state to a Stable state. An Active Cluster can be in one of the following states:

- **New:** Indicates that the Active Cluster has been newly created.
- **Updated:** Indicates that the Active Cluster has been modified.
- **Closed**: Indicates that the Active Cluster has been closed.

– **Stable Cluster:** A situation cluster is in a Stable state when the alarm does not get any further updates or does not reoccur. When the Situation Window of a situation cluster has expired, and no additional changes occur to that situation cluster, then it is in a Stable state. If the situation cluster is in a Stable state and non-noisy, then no icon is displayed. A Stable Cluster can be in one of the following states:
  - **New:** Indicates that the Stable Cluster has been newly created.
  - **Closed**: Indicates that the Stable Cluster has been closed.



– **Noisy Cluster:** When there is a constant duplication of the same alarm being encountered and the situation sub-cluster exceeds the preset noise threshold value, then those sub-clusters are termed as noisy. A noisy Cluster is controlled by a threshold as the number of alarms occurring as a percentage in the last 24 hours. Once all the underlying sub-clusters of a situation cluster are marked as noisy, then the situation cluster gets marked as noisy.

  A Noisy Cluster can be in Active state or Stable state and is denoted by a bell icon. Root cause identification is not performed for noisy situation clusters.
  Root cause identification is not performed for the following scenarios:

  - Root cause identification is not calculated for situations that have noisy situation clusters.
  - Root cause identification is not calculated for situations that have a single non-noisy template.
  - Root cause identification is not calculated for situations that have exceeded the preset template limit.
  - Root cause identification is still in process, and probable cause has not been identified yet.

Monitored Inventory

The Monitored Inventory view is a unified view of all entities for the selected situation alarm. You can navigate to Monitored Inventory in the context of entities that are part of the situation alarm.

Follow these steps:

1. In the **Situations** view, select multiple alarms from the Alarms table.

2. Click the icon.
   The page navigates to the Monitored Inventory page with the situation alarm filter applied.
3. You can view the services that are associated with the entity and can navigate to the Alarm Analytics, Performance Analytics, and Service Analytics page.

For more information, see Monitored Inventory.

Alarm Actions

The Alarm actions let you perform specific actions on a situation alarm. These actions are categorized as follows on the Situations view:

- Alarm Management
- Ticket Management
- Email Notification

> **NOTE**
> You can perform specific alarm actions even at the situation cluster level. When you drill down a situation cluster, the Alarm Management and Trigger Channel actions are enabled for the situation by default.

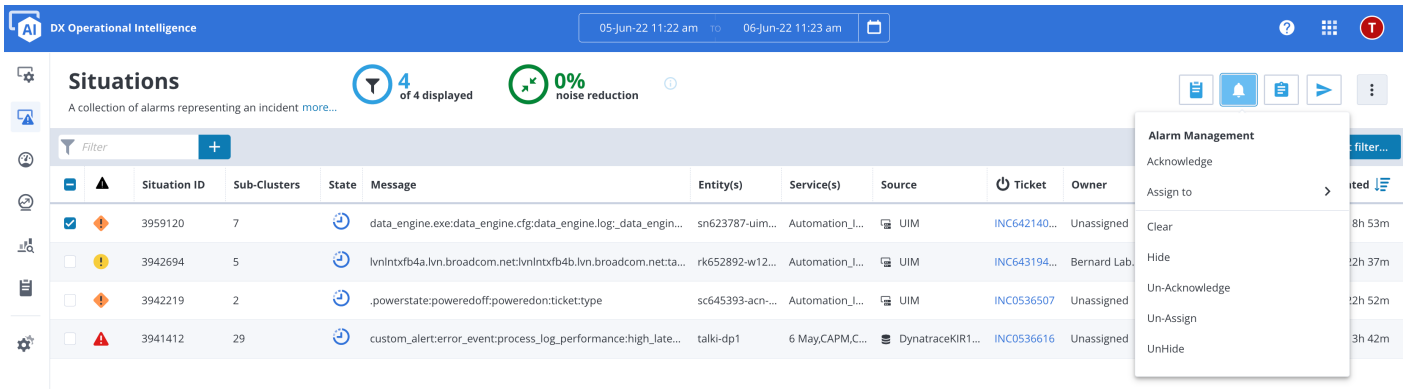
Alarm Management

You can use Alarm Management to manage stable situation alarms by performing operations such as assigning alarms to a user, clearing the assigned alarms, hiding the alarms, and un-assigning alarms from a user. You can use the icon to perform bulk alarm operations, and the icon next to the required situation alarm to perform a single alarm action. You can perform the following actions for Situation alarms:

- **Assign:** To select the user to whom the situation alarm is to be assigned.
- **Clear:** To clear one or more selected situation alarms.
- **Hide:** To hide the situation alarm details in the database. The situation alarm in the user interface is grayed to indicate that the data is hidden in the database.
- **Un-assign:** To remove the assignment for an alarm.

Follow these steps:

1. In the **Situations** view, select multiple alarms from the Alarms table, and click the icon to perform bulk operations. Alternatively, click the icon next to the **Owner** column of the required alarm to perform a single alarm action. The **Alarm Management** pop-up menu appears. In this dialog, you can perform the following actions:



2. Select the alarm action that you want to perform for one or more situation alarms. For example, say you want to assign a situation alarm to a particular user.
   a. In the **Alarm Management** pop-up, click **Assign to**.
   b. In the context menu, enter the user details or select the user to whom the situation alarm is to be assigned
      The situation alarm gets assigned to the selected user.

Ticket Management

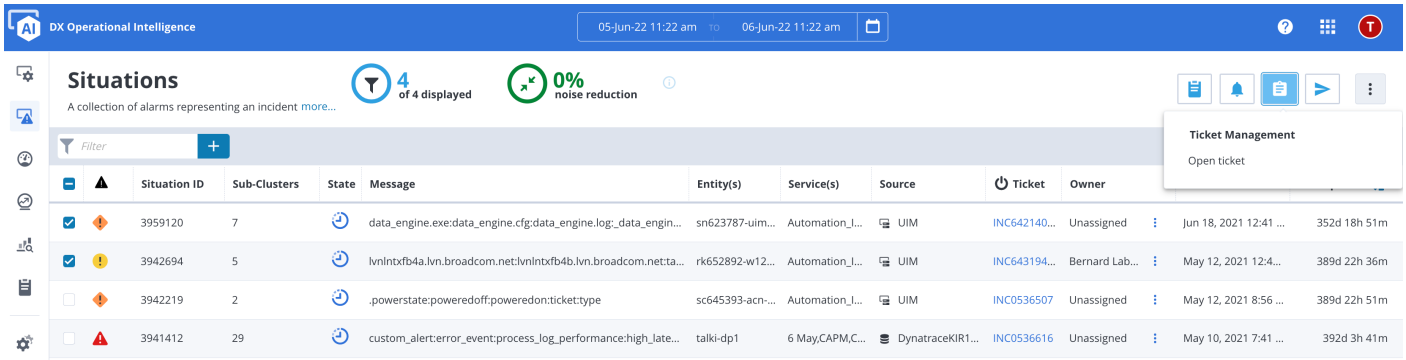
You can manage tickets in ServiceNow directly from the Situations view. You must configure the ServiceNow notification channel to manage the ticket update. You can use the icon to perform a bulk ticket operation for one or more situations, and the **Open Ticket** link next to the required situation alarm to open a ticket for that situation instantly. You can open a ticket for Active, Stable, and Noisy situation alarms.

> **NOTE**
> The **Open Ticket** link is visible only when the ServiceNow notification channel is configured. To configure the channel, see Channels

Follow these steps:

1. Select one or more Situation alarms from the table and click the (Ticket Management) icon.
2. Click **Open ticket** to open a ServiceNow ticket corresponding to the alarm.
   A ticket is created, and the ticket id is displayed.

> **NOTE**
> Alternatively, click the **Open Ticket** link under the **Ticket** column for an alarm.



3. Click the Ticket id, which redirects you to ServiceNow.

You can view the detailed information about the ticket.

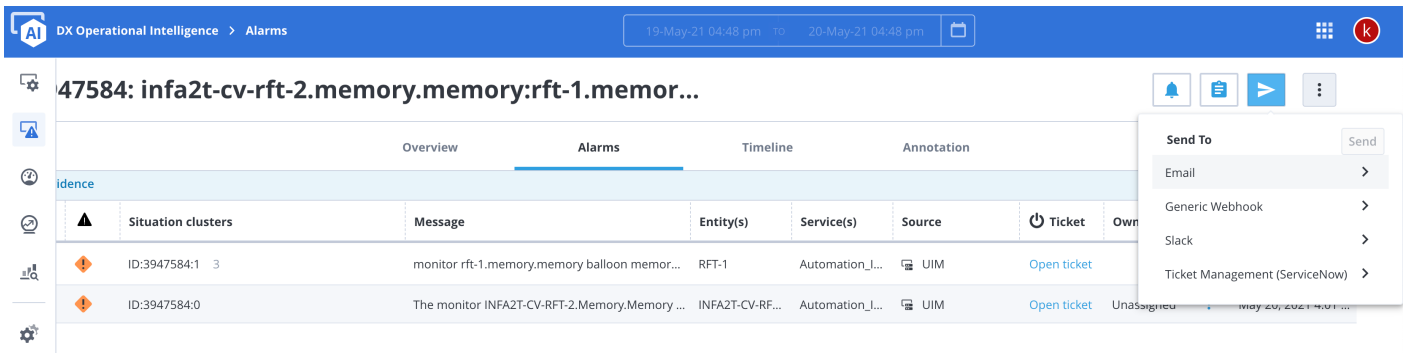
You can redirect back to the DX Operational Intelligence user interface by using the link provided in the ServiceNow ticket. Also, you can redirect back to DX Operational Intelligence by using the link sent through email, a ticket is created for the selected alarms in ServiceNow.

Email Notification

You can notify users about an alarm directly from the Situations view. You must configure the SMTP server to send emails to the recipient. For information on Email Notification configuration, see Email Notifications. You can send an email notification to one or more distribution lists to notify them about the alarm.

> **NOTE**
> If you do not configure the SMTP server, a success message appears but the email is not sent to the recipient.

1. Click the (Email Notification) icon on the **Situations** view.



2. To send an email notification about one or more situation alarms to one or more distribution lists, select **Email** or **Ticket Management (ServiceNow).**
3. Select one or more distribution lists.
4. Click **Send**.

Create a Policy for Situation Alarms

A policy determines when notifications are sent. When a situation alarm meets the criteria that are defined by the filters in the policy, a notification is sent to the associated channels. Therefore, you can create a policy to be triggered when filters or criteria that are defined are fulfilled. To create a policy for a situation alarm and to know more information on policies, see Policies.

Filters

You can filter alarms by using the various options available in the Situations view:

- **Time Filter**
- **Alarm View Filter**
- **Filter by Alarm Attribute**

Time Filters

The **Calendar** icon

enables you to select the duration for which you want to view the situation alarms that have been triggered.

The **Situations** view displays the currently active and stable alarms (alarms that are not closed) irrespective of the date/time you have selected in the filter. For example, if you select **12 Hours** in the Time filter, all the situation alarms that are active and stable get displayed. This helps you save time and focus on the active and stable situation alarms instead of going through the entire list of situation alarms.

Situation alarms also provide you with a **Custom** option, which enables you to pick a particular start date/time and a particular end date/time to help you narrow down the alarms that you are looking for.

Note:

If you still want to view all the situation alarms (active, stable, and closed) between the selected time period, then you must enable the **Show Closed Situations** option in the **Alarms** view filter.

Alarms View Filter

Use the **Alarms View Filter** to filter alarms based on the following category that is generated in the selected period:

- **Situations:** This option displays all the situation alarms.
- **Service Alarms:** This option displays all the service alarms.
- **All Alarms:** This option displays all alarms.
- **Auto Update View:** Use this option to refresh the alarms table automatically, enable the **Auto-update view** switch. Use this switch to shift the view from auto-update to manual update. To refresh the alarm table manually:
  a. Disable **Auto-update view** switch.
  b. A pop-up (**Alarm view is out-of-date**) message appears with the message "*Updates are pending for the current alarm view. Refreshing would update the view, but the focus may be lost. Applied filters and sorting will remain in effect after refresh*."
  c. Click **Refresh View** button to refresh the alarm table and the alarm table refreshes the data between the time interval that is set in the ALARM_REFRESH_INTERVAL environment variable and displays the last 7 days data or Select close (X), to leave the view as it is.
- **Situation window:** Use this option to select the time period for your situation alarms. The situation window can be 30 minutes, 6 hours, 24 hours, 3 days, or 7 days.
- **Show Closed Situations:** Use this option to view closed situation alarms by enabling the **Show Closed Situations** switch. The closed situation alarms appear disabled with a gray text and there is also a column that displays the closed time. You cannot perform any action on these closed alarms.

  NOTE
  When you enable the **Show Closed Situations** option, the **Situations** view displays the list of closed situation alarms along with the active and stable alarms.

Alarm type	Message	Entity(s)	Service(s)	Source	Ticket	Owner	Last updated	Close time
CPU	Average (1 samples) tot...	is654739-WVM...	11,12,18May...	UIM		Unassigned	Jun 5, 2020 5:46 PM	Jun 5, 2020 5:46 PM
Network	Profile is654739-wvm04...	is654739-wvm...		UIM		Unassigned	Jun 5, 2020 5:44 PM	Jun 5, 2020 5:44 PM
CPU	Average (1 samples) tot...	is654739-WVM...	11,12,18May...	UIM		Unassigned	Jun 5, 2020 5:41 PM	Jun 5, 2020 5:41 PM
Robot	Robot is654739-wvm05...	is654739-wvm05		UIM		Unassigned	Jun 5, 2020 5:39 PM	Jun 5, 2020 5:39 PM
Robot	Robot is654739-WVM03...	is654739-WVM...		UIM		Unassigned	Jun 5, 2020 5:39 PM	Jun 5, 2020 5:39 PM

NOTE
You cannot view the closed alarm in the **Timeline** tab.

- **Enable Algorithmic Situations:** Use this option to enable or disable algorithmic clustering for situations. The algorithmic clustering is enabled by default. When you disable this option, the existing active algorithmic type clusters become orphans, and no new algorithmic type clusters are formed.

  NOTE
  For a user to have access to enable or disable the Algorithmic Situations, the user must be provided with the Update Situation Tenant Configuration privilege on the Roles page.
- **Customize Columns:**The Customize Columns view enables you to customize the list of alarm columns, such as Alarm type, Severity, Service, Source, and so on. You can save the column settings of the alarm view table with the alarm information (filtering and sorting options) to a new queue. You can add a maximum of 15 columns to the page.
- **Export to Excel** This option enables you to export the alarm details to an Excel sheet. The Excel sheet contains data that is displayed in the Alarm Table. You can export in the following ways:

  NOTE
  DX Operational Intelligence exports a maximum of 10000 clusters in a single export.
  - **All Alarms** Enable you to export all alarms that are listed in the Alarm Table.
  - **All Alarms with details** Enables you to export alarm details of all the alarms that are listed in the Alarm Table which are based on the current context or view.
  - **Selected alarms** Enables you to export selected alarms from the Alarm Table.
  - **Selected Alarms with details** Enables you to export alarm details of the selected alarms that are listed in the Alarm Table which are based on the current context or view.

Filter by Alarm Attributes

You can also filter alarms by attributes using the **Alarm Attributes Filter**

. This filter allows you to view only those alarms with attributes matching your search criteria. Situation alarms support the filtering of the following attributes:

- Entity Name
- Message
- Noisy
- Service
- Severity
- Situation ID
- Source
- State
- Status

Create Alarm Filters

Creating alarm filters enables you to filter alarms based on your requirements.

Follow these steps:

1. Click the plus icon in the **Alarm Attributes Filter**.
2. Select a filter attribute with any one or more operators. For example, if you want to see all critical alarms, select the attribute as **Severity** and select its value as **Critical**.
3. Click **Add** to add attributes to the alarm filter.
   The Alarms table and Insights show only the alarms that match your search criteria for the selected attributes. The application also refreshes to display the noise reduction percentage based on the applied filter criteria.

   NOTE

   - While filtering, the **AND** operator is used between attributes and the **OR** operator is used between the attribute values. For example,
     ```
     (Severity: Critical OR Major) AND (Alarm Type:"application" OR "fault")  AND (Message contains "sshd" OR Message does not contain "ALARM: [SYSTEMS]")
     ```
   - Only asterix (*) and dot (.) are supported in the filters.
4. You can search for situation alarms using **Status** filter attribute. You can drill down the situation alarms by selecting the **Status** value from **New**, **Updated,** and **Closed**. For example:

   a. Select the filter attribute value as **Status**.
   b. Select the filter attribute value as **New**.
   c. Click **Add**.
5. You can filter the situations for a selected service and the time range using the Service filter attribute. On applying this filter, the application refreshes to display the situations associated with the selected service. The application also displays the aggregate percentage of noise reduction for the selected service and the associated child services.
   a. Select the filter attribute as **Service**.
   b. Select a service as the filter attribute value.
   c. Click **Add**.
   d. Click the Calendar icon to select the time range.

   NOTE
   DX Operational Intelligence supports navigation between Service Analytics and Situations and ensures the consistent display of Noise Reduction percentage for the selected filter criteria.

Search for Elements

You can search for elements on the Situation Alarms page using **Alarm Attributes Filter** .

Save Alarm Filters

You can filter alarms from the list of filters saved.

Follow these steps:

1. To save the current alarm filter, click the **Save current filter** button Save current filter... . The Save Filter window appears.

2. Enter the name for the alarm filter in **Alarm filter name**.
3. If you want to set the alarm filter as default, select the **Set as default** option. When you access the page next time, the **Set as default** option is enabled and the default alarm filter is applied, and the Alarms table and Insights show only alarms with the saved attributes. By default, the **Set as default** option is disabled.

   NOTE
   The default alarm filters that are created in the context of service take precedence over the default alarm filters on the Alarm Analytics view. The filters that you save on the Situation alarms page are specific only to the Situation alarms page, and these saved filters would not be available on the other alarm pages.

Edit Alarm Filters

- Add extra attributes to an existing alarm filter and click the **Update** button to update the existing alarm filter.
- To save the alarm filter with a different name, specify a different name and click **Save as**.

View Alarm Filters

To view all saved alarm filters, click the **Saved filters** button

.

NOTE
The filters that you save on the Situation alarms page are specific only to the Situation alarms page, and these saved filters would not be available on the other alarm pages.

Sorting Columns

You are provided with an option to sort the following columns in ascending or descending order:

- Severity
- Situation ID
- Sub-Clusters
- Alarm Message
- Last updated

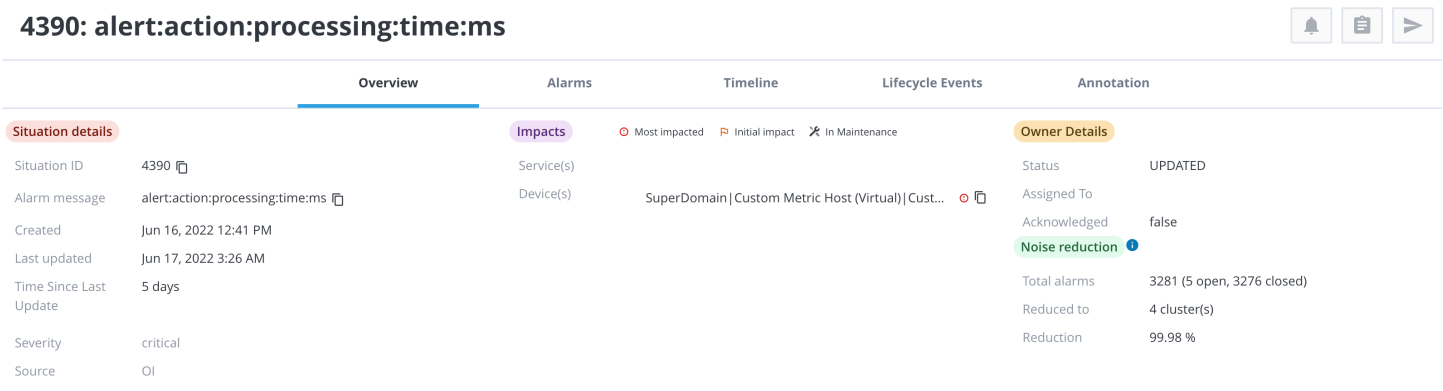
Navigating Situation Alarms

The **Situation Alarms** contains three tabs:

- Overview
- Alarms
- Timeline
- Annotation

Overview tab

This tab provides additional information about the selected alarm, and the properties are specific to the product or source from where the alarm originates. To view the details of a situation alarm, you must click on the situation alarm and must select the **Overview** tab which contains the extensive details of the situation alarm.



The Overview tab displays the following information:

- **Situation details** - This section displays details such as the ID of the situation alarm, alarm message, date/time of creation and updation, and the time since the alarms were last updated.
- **Impacts** - This section displays the details of the services and devices that are impacted due to the selected alarm. Clicking a service redirects to the Service Analytics details page of that particular service. You can copy the list of devices by clicking the Copy icon. This section also displays the list of all the impacted entities due to this alarm. The following impact legends are shown beside impacted services and entities:
  - : Most impacted
  - : Initial impact
  - : In Maintenance
- **Owner details** - This section displays the status of the owner, which user the alarm is assigned to, and whether the user has acknowledged the alarm or not.
- **Noise reduction** - This section displays details such as the total alarms open and closed the number of clusters the alarm has reduced to, and the reduction percentage.
- **Impacted Services** - This table displays the impacted service metrics (such as users that are availing the service, actual service availability, and risk).

Alarms tab

This tab provides details of the Situation Clusters and Evidence of the selected alarm. By default, when you click on any Situation alarm in the Situations view, the **Alarms** tab opens. The Alarms tab displays two sections:

- **Evidence**
- **Alarms**

Evidence

This section contains the following information:

- **Entity**: Displays the number of entities in the cluster.
- **Raw Alarms** and **Anomalies**: Displays the total number of raw alarms and anomalies that are generated.
- **Raw Alarms**: Displays the total number of raw alarms that are generated.
- **Clusters:** Displays the number of cluster alarms that are generated for the situation alarm.
- **Situation ID** and **Severity**: Displays the Situation ID of the situation alarms in the Alarm table and indicates the severity of those situation alarms.
- **Noise Reduction:** Displays the noise reduction percentage of the cluster

Alarms

Click an alarm in the situation cluster and the row expands to display more information about the alarm. When you drill down on a situation alarm, you can view unique alarms that are clustered together. The clustering of unique alarms together is known as nested clusters. The learning algorithms create sub-clusters that are more contextually relevant and correlated.

> **NOTE**
> If the sub-cluster contains only one alarm, the situation alarm directly displays the raw alarm details.

The **Alarms** cluster displays the following details:

- **Column / Row-level Alarm Action:** Select a row to perform individual actions on individual alarms or the entire column to perform bulk action.
- **Severity:** Indicates the severity of each alarm.
- **Situation Clusters:** Indicates the Situation ID of a cluster alarm and the number of sub-clusters.
  - **Sub-clusters:** Indicates the number of underlying alarm clusters. If there are any sub-clusters, the value is displayed in grey color next to the alarm, and a drop-down option appears next to the Situation cluster.
- **Message:** Indicates the alarm message.
- **Entity:** Indicates the list of devices or application names.
- **Service:** Indicates the list of service alarm names.
- **Source:** Indicates the source product of the alarm.
- **Last Updated**: Indicates the date and time when the alarm was last updated.

Topology Tab

This tab provides topology details of the selected service or sub-service. You can quickly view the topology for an item that is associated with a Situation alarm template to easily understand connected neighbors and their alarm state.

Timeline tab

This tab indicates the timeline details of the situation cluster alarm.

The Timeline tab displays the following information:

- **Evidence:** The information that is displayed in the timeline tab is the same as in the Alarm tab. For more information, see the Alarms tab.
- **Situation Clusters:** The Alarm cluster displays the following details:
  - **Column / Row-level Alarm Action** Select a row to perform individual actions on individual alarms or the entire column to perform bulk action.
  - **Severity:** Indicates the severity of each alarm.
  - **Situation Clusters:** Indicates the Situation ID of a cluster alarm and the number of sub-clusters.

- **Sub-clusters:** Indicates the number of underlying alarm clusters. If there are any sub-clusters, the value is displayed in grey color next to the alarm, and a drop-down option appears next to the Situation cluster.

– **Message** Displays the alarm message.
– **Entity** Displays the Application or Device name.
– **Source** Displays the name of the source product
– ***<Date and Time Interval>***
  - This time interval shows the journey of an alarm. You can adjust the date and time interval scroll, by dragging the horizontal scroll. You can drag and adjust the scroll on both ends of the timeline. The alarm timeline displays the alarms accordingly.

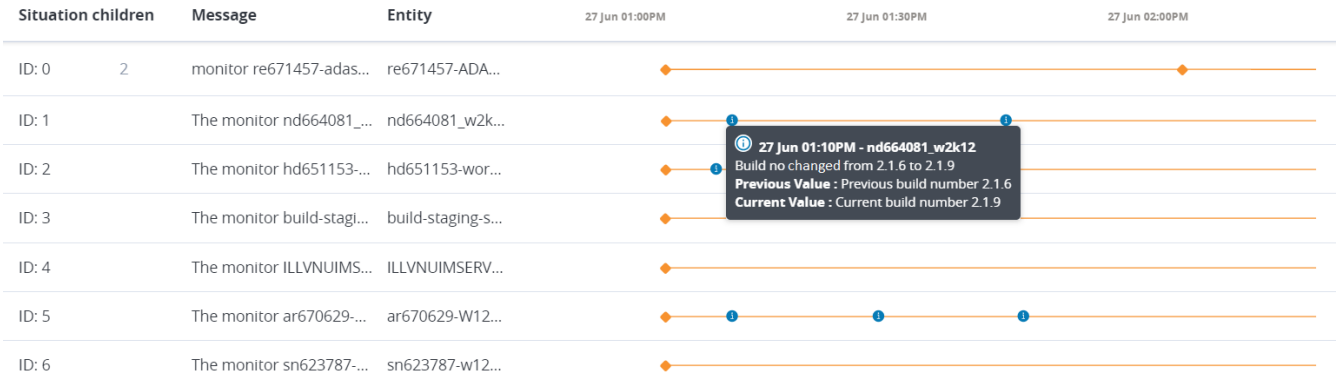
  - Click the severity icon to see the following information regarding an alarm:
    - **Created:** Displays the date and time at which the alarm got created.
    - **Alarm ID:** Displays a unique alarm ID
    - **Metric:** Indicates metric details
  - Click the **Change event** icon

to see any changes for the event at a given time.

> **NOTE**
> The Change event icon appears for an alarm when the event attribute such as hostname, IP, ci, and unique_id match the alarm attribute. The Change event icon displays the event only once for the first alarm in the cluster for a device.

Change event icon displays the following details:
- **Change event message**: This message displays the change event information on what has changed during the given time.
- **Previous value:** Indicates the previous value of an event
- **Current value:** Indicates the current value of an event

For example, in the following image, on clicking the **Change event** icon, it displays information about the change in build number from 2.1.6 to 2.1.9.



-

Metric palette switch: To view the details and metrics of the alarm, select an alarm and click the icon. This switch provides the following details:

– **Details:** This section provides the following details of the alarm:

- **Alarm ID:** Displays a unique ID
- **Alarm Message:** Displays the alarm message
- **Service impact:** Displays the list of impacted services.
- **Entity:** Displays the application or device name.
- **CI name:** Displays CI name.
- **Last Updated:** Displays the date and time when the alarm was last updated.
- **Duration:** Indicates the duration of the alarm.
- **Annotation:** Enter an annotation and click **Save**.

– **Metrics:** Displays the metric linear graph. The **Compare metrics** link launches Performance Analytics from the context of an alarm and allows you to compare a single metric from different devices or multiple metrics from single or multiple devices. For more information about the metric charts, see Performance Analytics.

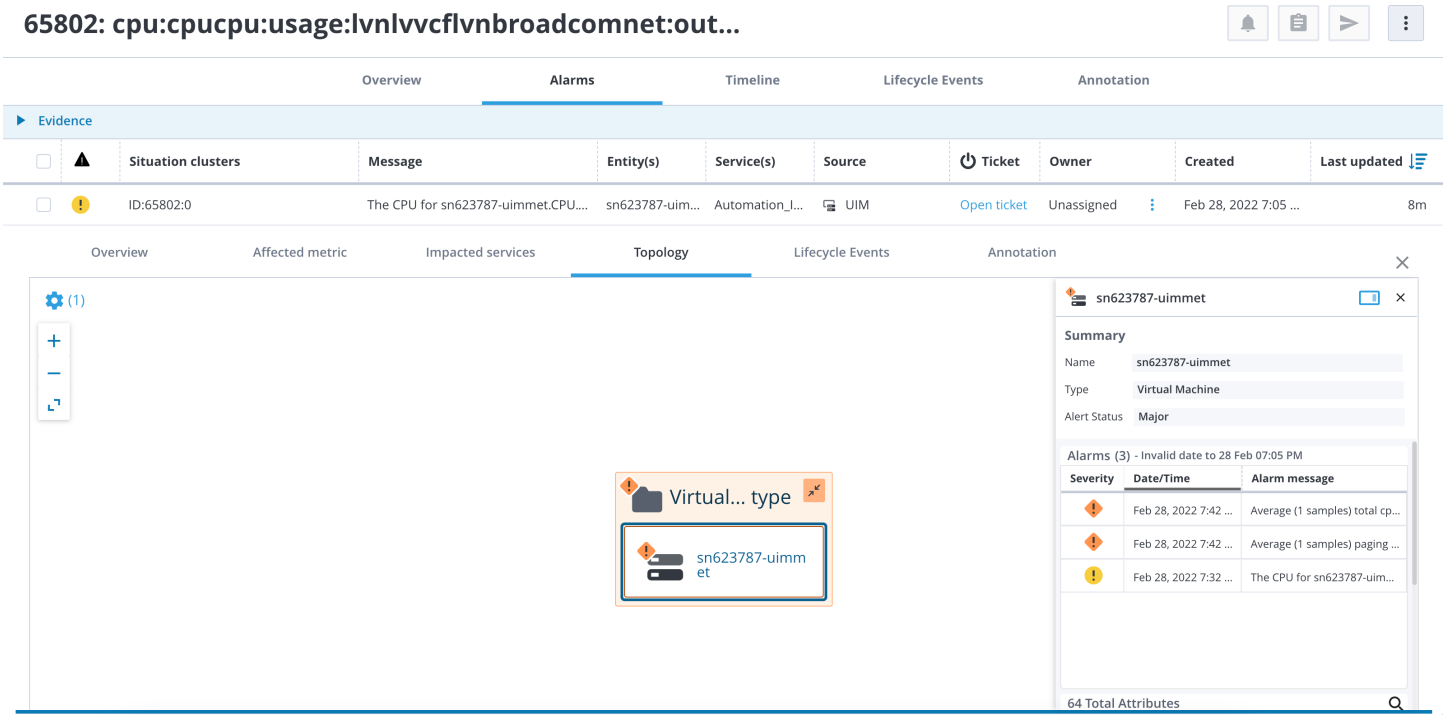
Annotation tab

This tab enables you to add any additional information or details that you want to add for a particular situation alarm. You can add details like cause, resolution, troubleshooting steps, or any other important information about a particular situation. You can view the annotation details of a situation alarm, irrespective of whether a situation is in the closed or open state. You are also provided with the ability to synchronize your annotations with any connected ServiceNow (SNOW) tickets, thus enabling the SNOW tickets to reflect the same annotation details as well.

You can add the required information in the **Annotation** tab of a particular Situation alarm and click **Save** to view the annotation as a part of the situation alarm details. The annotation stores the information that you added, the details of the person who added this annotation, date, and time details.

Topology tab

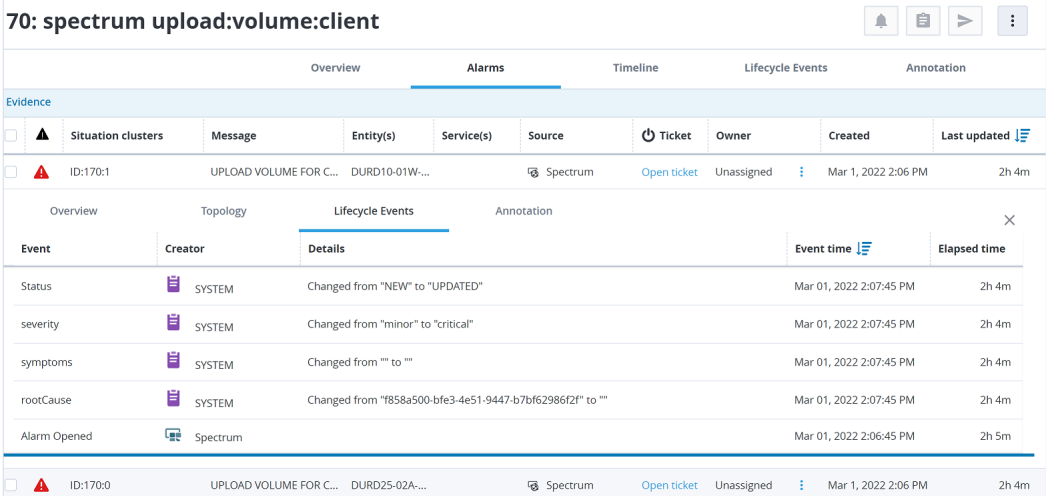
This tab provides topology details of the selected service or sub-service. You can view the topology details such as Summary and Alarms by clicking the parent service and sub-service.



Lifecycle Events tab

This tab provides the lifecycle of a situation. This tab displays the events that occurred for an alarm from the time it is created, such as created time, status updates, actions, annotation updates, threshold changes. You can view the following details on the lifecycle events tab:

- Event
- Creator
- Details
- Event time
- Elapsed time



{"URL":["https://digital-oi/alarms-analytics/clusterAlarms","https://digital-oi/alarms-analytics/clusterAlarms/overviewTab","https://digital-oi/alarms-analytics/clusterAlarms/timelineTab","https://digital-oi/alarms-analytics/clusterAlarms/lifecycleEvents","https://digital-oi/alarms-analytics/clusterAlarms/annotation","https://digital-oi/alarms-analytics/clusterAlarms/alarmTab/overviewTab","https://digital-oi/alarms-analytics/clusterAlarms/alarmTab/impactedServices","https://digital-oi/alarms-analytics/clusterAlarms/alarmTab/topology","https://digital-oi/alarms-analytics/clusterAlarms/alarmTab/lifecycleEvents","https://digital-oi/alarms-analytics/clusterAlarms/alarmTab/annotation"],"description":"concept.dita_c9c81f83-d002-46a1-8656-4e53ff3f73c1","customCards":[{"id":"IPCE_ClusteringTypes","type":"configure","title":"Clustering Types"},{"id":"IPCE_monitoredinventory","type":"configure","title":"View Monitored Inventory"},{"id":"IPCE_AlarmActions","type":"configure","title":"Alarm Actions for Situations"},{"id":"IPCE_situtaionfilters","type":"configure","title":"Situtation Filters"},{"id":"IPCE_SitutaionAlarmsTabs","type":"configure","title":"Situation Alarm Tabs"}]}

Service Alarms

Service Alarms are groups of alarms that affect one or more business services and are related to an incident. Service alarms are identified by the time the incident occurred and its root cause.

The root cause is the alarm on the topologically deepest device in the affected business service. All the situations that are reported by alarms in the group are due to the identified root cause. Service alarms are based on the entity of the alarm compared to the inventory of the services defined in DX Operational Intelligence.

> **NOTE**
> By default, the Service Alarms are disabled for new tenants. To re-enable the Service alarms for new tenants, contact **Broadcom Support**.

Access Service Alarms

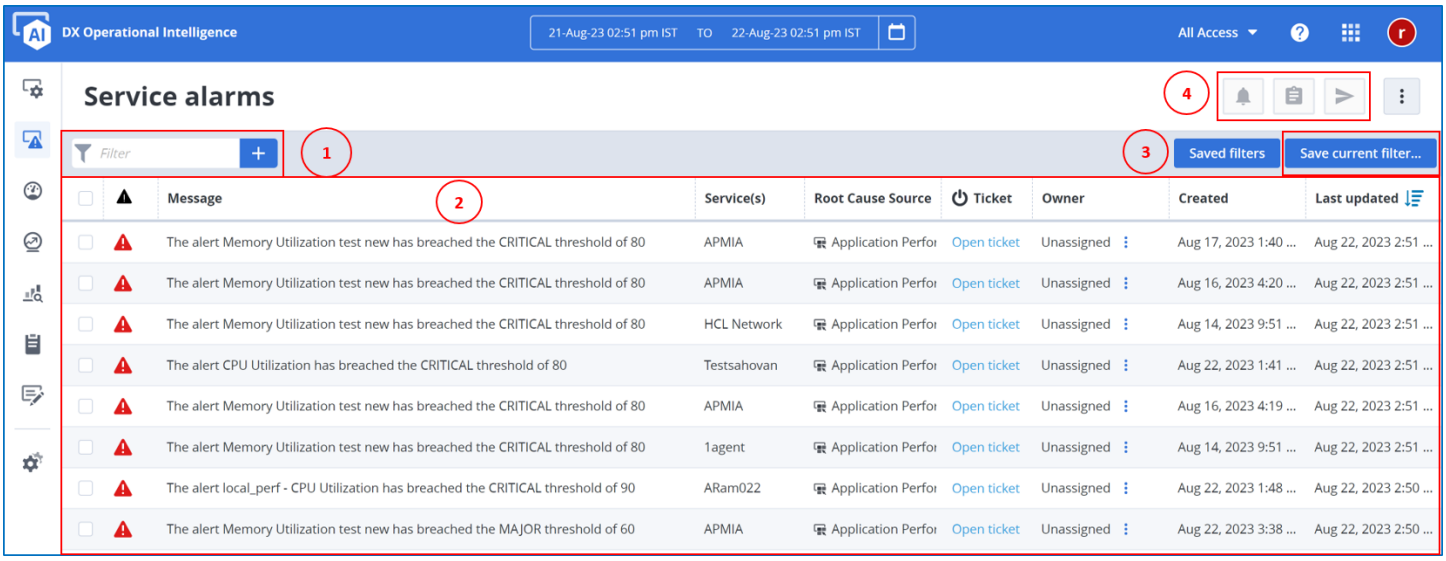
1. Log in to DX Operational Intelligence.
2. Click the

icon in the left navigation panel.
   The **Service Alarms** page opens by default.

To view the Service Alarms manually, click on the **Change Alarms Views** icon, and click **Service Alarms**.

Service Alarms User Interface

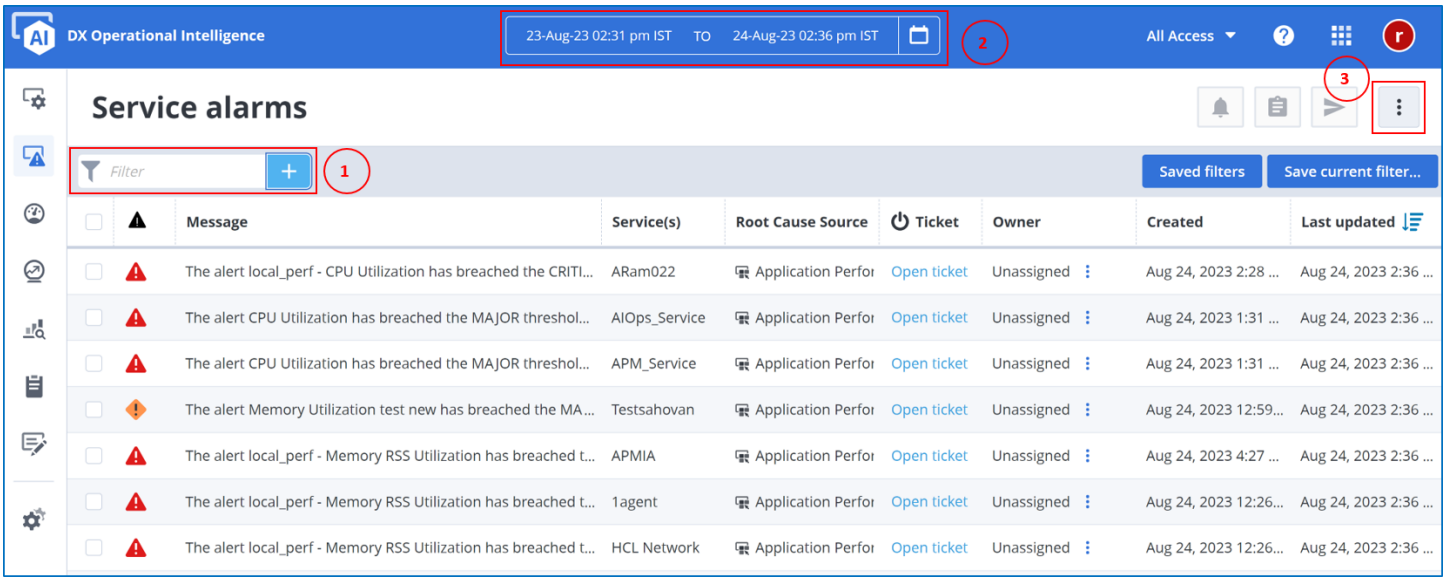
The Service alarms view displays the following details:



- **Service Alarm Filters (1)**
- **Service Alarms Table (2)**
- **Saved Filters (3)**
- **Additional Options (4)**

Service Alarm Filters

You can filter alarms using the following options available on the Service Alarms page:



- **Alarm Attributes Filter (1)**
- **Date and Time Filter (2)**
- **Change Alarm Views (3)**

Alarm Attributes Filter

You can filter alarms by attributes using the **Alarm Attributes Filter**. This filter allows you to view only those alarms with attributes matching your search criteria.

Follow these steps:

1. Enter the attribute to display the attributes or click the

   icon in the **Alarm Attributes Filter**.
2. Select the filter attributes with any one or more operators. For example, if you want to display all critical alarms, select the attribute as **Severity** and select its value as **Critical**.

   NOTE

   - While filtering, the **AND** operator is used between attributes and the **OR** operator is used between the attribute values. For example,

     ```
     (Severity: Critical OR Major) AND (Alarm Type:"application" OR "fault")  AND (Message contains "sshd" OR Message does not contain "ALARM: [SYSTEMS]")
     ```
   - Only asterix (*) and dot (.) are supported in the filters.
3. Click **Add** to add attributes to the alarm filter.
   The Alarms table and Insights show only the alarms that match your search criteria for the selected attributes.

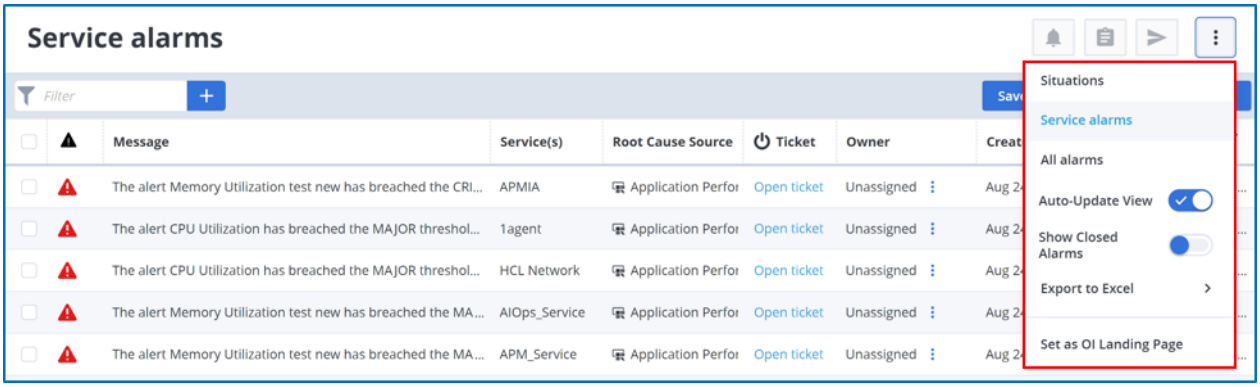
Date and Time Filter

The **Calendar** icon enables you to select the duration for which you want to view the alarms that have been triggered. By default, service alarms that have been generated in the last week are displayed.

Alarm Analytics also provides you with a **Custom** option, which enables you to pick a particular start date/time and a particular end date/time to help you narrow down the alarms that you are looking for.

Change Alarms Views

Use the **Change Alarm Views** icon to filter alarms based on the following options for the selected time period:



- **Situations:** Click this option to display all situation alarms.
- **Service Alarms:** Click this option to display all service alarms.
- **All Alarms:** Click this option to display all alarms.
- **Auto Update View:** Use this option to refresh the alarms table automatically. To refresh the alarm table manually:
  a. Disable the **Auto-update view** option.
     A pop-up (**Alarm view is out-of-date**) message appears with the message "*Updates are pending for the current alarm view. Refreshing would update the view, but the focus may be lost. Applied filters and sorting will remain in effect after refresh*."
  b. Click the **Refresh View** button to refresh the alarm table. The alarm table refreshes the data between the time interval that is set in the **ALARM_REFRESH_INTERVAL** environment variable and displays the last 7 days data or Select close (X), to leave the view as it is.
- **Show Closed Alarms:** Enable this option to view the closed service alarms. The closed service alarms appear disabled with a gray text. The alarms table also displays the closed time. You cannot perform any action on these closed service alarms. You can also view the closed alarms in the **Timeline** Tab.
- **Export to Excel:** Click this option to export the alarm details to Excel. You can export the details in the following ways:
  – **All Alarms:** Enables you to export all alarms that are listed in the Alarm Table.
  – **All Alarms with details:** Enables you to export alarm details of all the alarms that are listed in the Alarm Table based on the current context or view.
  – **Selected alarms:** Enables you to export selected alarms from the Alarm Table.
  – **Selected Alarms with details:** Enable you to export alarm details of the selected alarms that are listed in the Alarm Table based on the current context or view.

Service Alarms Table

The Service Alarms table displays the following information:

Column Name	Description
Column / Row-level Alarm Action	Enables you to perform row-level service alarm actions or column-level to perform bulk service alarm actions.
Severity	Indicates the severity of a service alarm. The following colors indicate the severity: • Red: Critical • Orange: Major • Yellow: Minor • Light Blue: Informational • Black: Warning
Message	Displays the description details of the service alarm.
Services	Displays the services that are impacted by the service alarm.
Root Cause Source	Displays the root cause source product from which the service alarm is generated.
Ticket	Displays an option to open a ticket. You can open a ticket under the following two categories: • **On Root Cause** - Use this option to open a ticket on the Root cause. • **On Service** - Use this option to open a ticket on the service.
Owner	Displays the details of the owner. If no owner is assigned, then **Unknown** is displayed. To view alarm actions, click icon.
Last Updated	Displays the date and time when the alarm was last updated.

You can sort the following columns in ascending or descending order:

- Severity
- Alarm Message
- Created
- Last Updated

Saved Filters

You can save your current filter and also use existing saved filters for quick searches.

Follow these steps:

1. Select the filter using the filter attributes.
2. Click **Save current filter** option.
   The Save Filter dialog appears.
3. Enter the name for the alarm filter.
4. Select the **Set as default** checkbox to set the alarm filter as default.

NOTE
The default alarm filters that are created in the context of service take precedence over the default alarm filters on the Alarm Analytics page. The filters that you save on the Service Alarms page are specific only to the Service Alarms page, and these saved filters would not be available on the other alarm pages.

5. Click **Save**.
   The filter is added to the **Saved Filters** list.

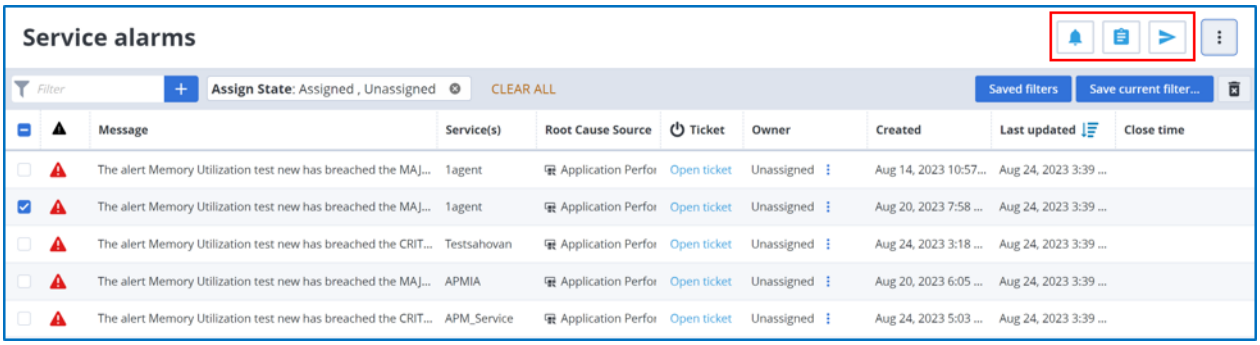
To view all the saved alarm filters, click **Saved filters**. To add attributes to an existing alarm filter, edit the filter.

Follow these steps:

1. Click the **Saved filters** button.
2. Select the saved filter in the list.
3. Select and add the required attributes.
4. Click **Update** to update the existing alarm filter.
5. Click **Save as** to save the alarm filter with a different name.

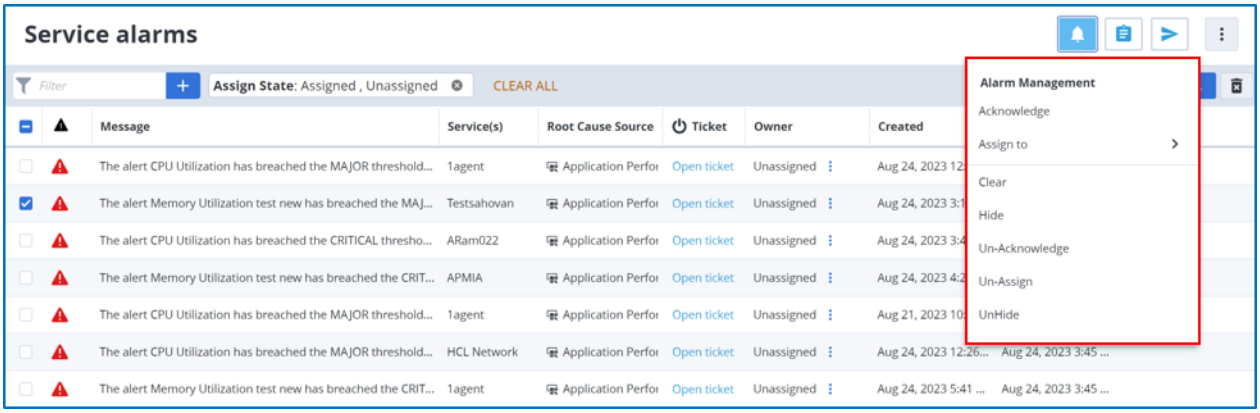
Alarm Actions

The Alarm actions let you perform a specific action on a service alarm. These actions are categorized as follows on the Service Alarms view:



Alarm Management

You can use the Alarm Management icon ( ) to manage service alarms. You can use the icon to perform bulk alarm operations, and icon next to the required service alarm to perform a single alarm action.



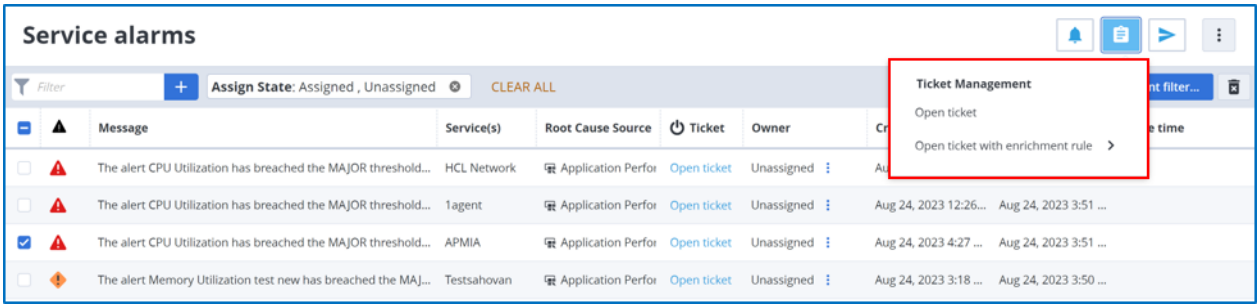
Select multiple alarms from the Alarms table and click the **Alarm Management** icon to perform the following bulk operations:

- **Acknowledge**: You can acknowledge the service alarm. A green tick appears, which indicates that the alarm is acknowledged.
- **Assign:** You can select the user to whom the alarm is to be assigned.
- **Clear:** You can clear all alarms.
- **Hide**: You can hide the alarm details in the database. The alarm in the user interface is grayed to indicate that the data is hidden in the database.
- **Un-Acknowledge**: You can un-acknowledge an alarm.
- **Un-assign:** You can remove the assignment for an alarm.
- **UnHide:** You can unhide the alarm detail from the database.

NOTE
Alternatively, click next to the **Owner** column to perform a single alarm action.

Ticket Management

You can manage tickets in the ticketing system directly from the Service Alarms view using the icon. You must configure the ITSM notification channel to manage the ticket updates.



Follow these steps:

1. Select the alarms from the table.
2. Click the **Ticket Management** icon and select the required option:
   - **Open ticket:** If you open the ticket using this option, the mapping rule that is associated with the channel is used for the ticket enrichment.
   - **Open ticket with enrichment rule:** If you use this option, the mapping rule that you select in this list is used for enrichment instead of the rule that is associated with the channel.

   Alternatively, click the **Open Ticket** link under the **Ticket** column for a single alarm.
3. Click the ticket ID which redirects you to the ITSM ticket management system. You can view the detailed information about the ticket.

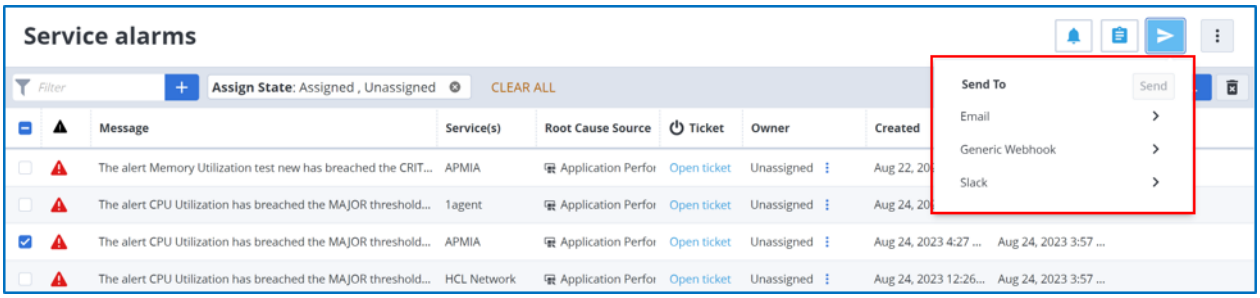
   NOTE

   - For service alarm, you can open a service alarm ticket or root cause ticket by clicking the **Open Ticket** link under the **Ticket** column.
   - The Open Ticket link is visible only when the ITSM notification channel is configured.

You can redirect back to the DX Operational Intelligence using the link provided in the ticket. You can also redirect back to DX Operational Intelligence using the link sent through email, a ticket is created for the selected alarms in the ticketing system.

Trigger Channel

You can notify users about an alarm directly from the Service Alarms view using the **Trigger Channel** icon ( ). Click the **Trigger Channel** icon and select one or more distribution lists to notify them about the alarm through email.



NOTE
Select at least one alarm in the table to enable this icon.

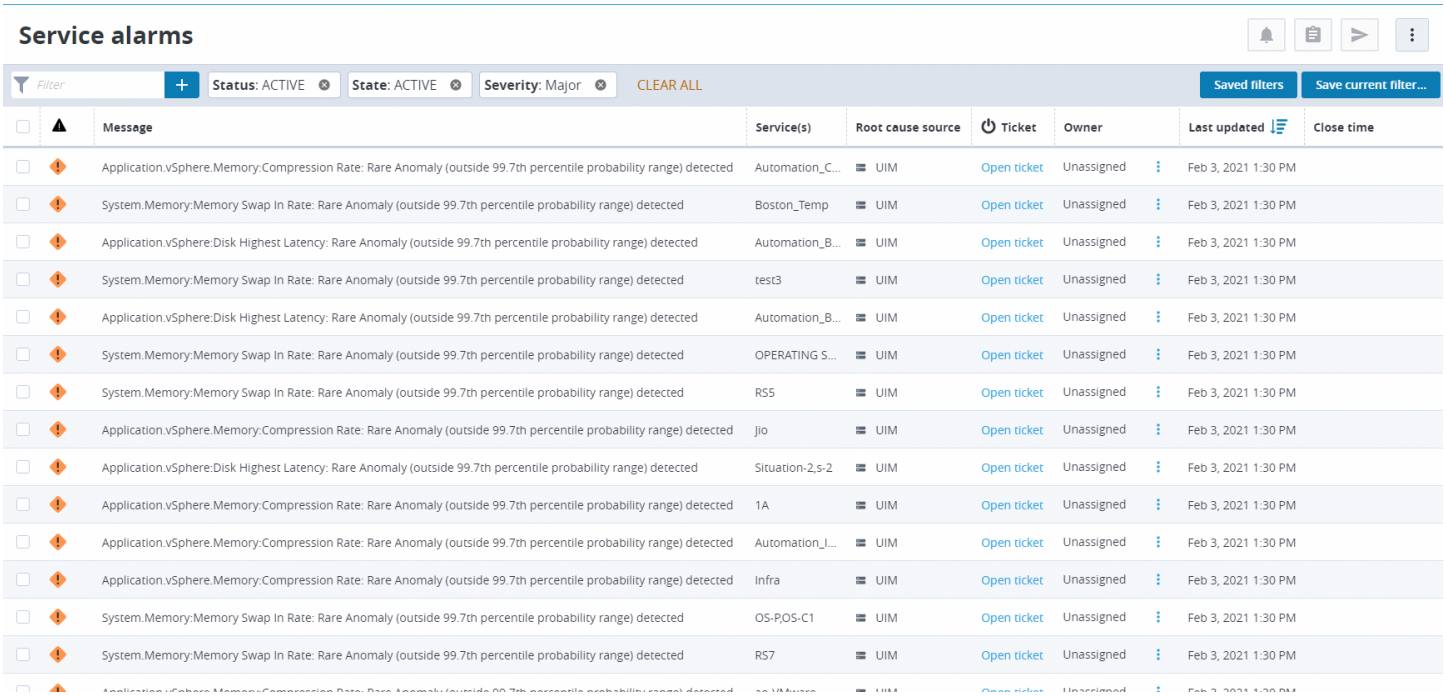
- **Email:** You can notify users about an alarm directly from the Situations view. You can send an email notification to one or more distribution lists to notify them about the alarm. Click the Trigger Channel icon, select Email, and then select the email channels from the list. For information on Email Notification configuration, see Email Notifications.

  NOTE
  You must configure the SMTP server to send emails to the recipient. If you do not configure the SMTP server, a success message appears but the email is not sent to the recipient.
- **Generic Webhook:** You can notify users about an alarm directly from the Situations view. Click the **Trigger Channel** icon, select Generic Webhook, and then select the webhook channels from the list. For information on Email Notification configuration, see Configure Generic Webhook Channel section.
- **Slack:** You can notify users about an alarm directly from the Situations view. Click the **Trigger Channel** icon, select Slack, and then select the Slack channels from the list. For information on Email Notification configuration, see Configure Slack Channel section.

Navigating Service Alarms

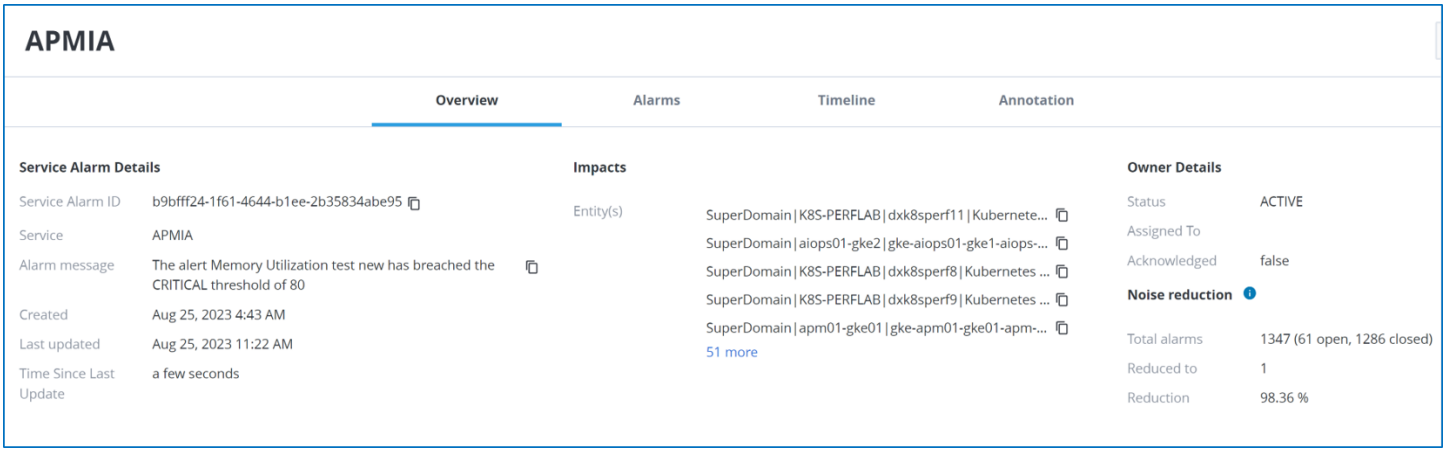
Click any service alarm in the Alarm table, and you can view the following tabs:

- **Overview**
- **Alarms**
- **Timeline**
- **Annotation**



Overview

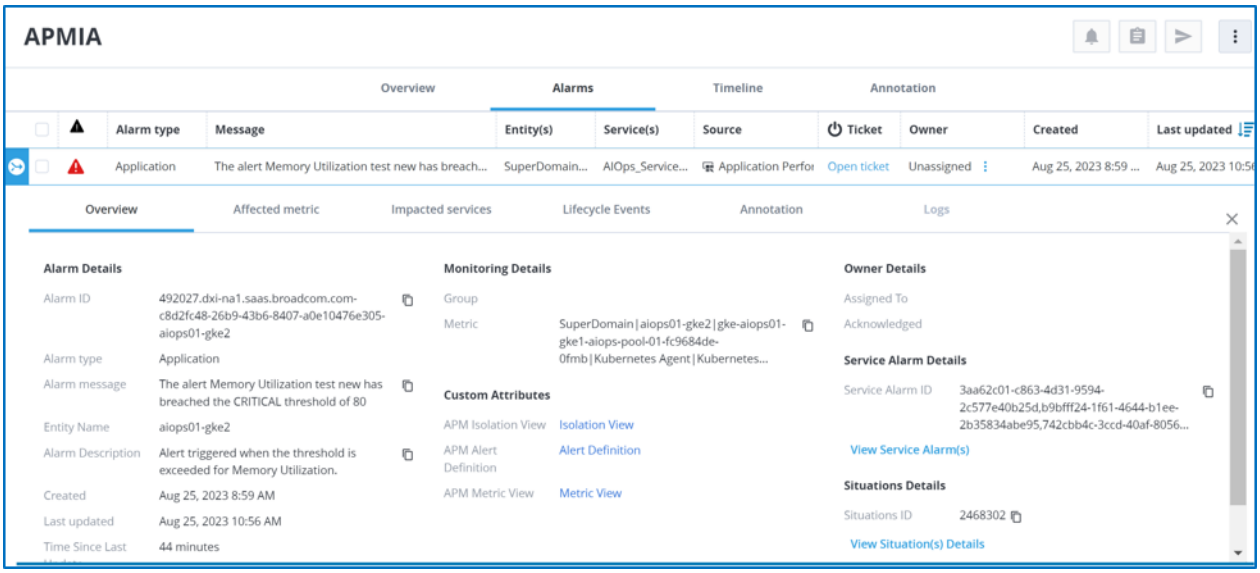
This **Overview** tab provides additional information about the selected alarm. Properties are specific to the product or source from where the alarm originates. The Overview tab displays the following details:



- **Service Alarm Details -** lists the details of the Service alarms such as Alarm ID, Alarm Type, Description of the Alarm, Suppression key, time of creation, last updated and how long since last update.
- **Impacts:** lists the details of all the impacted entities.
- **Owner details:** Provide details of the ticket such as the Status of the ticket, whom the ticket is Assigned to, and whether the alarm is Acknowledged.
- **Noise Reduction:** Displays the details such as total alarms, alarms reduced to, and reduction percentage. Noise reduction is the summation and de-duplication of related alarms.

Alarms Tab

The Alarms tab contains the list of causing raw alarms and their details. By default, when you click on a service alarm in the alarm table, the row expands to display the **Alarms** tab. The Alarms tab displays the following sub-tabs:



- **Overview**
- **Affected Metric**
- **Impacted Services**
- **Topology**
- **Annotation**

Alarms Tab - Overview

The Overview sub-tab under the Alarms tab displays the following details:

- **Alarm Details**

Source Product	Details
Application Performance Management	Alarm ID, Alarm Type, Alarm Message, Entity Name, Created, Last Updated, Time Since Last Update, Alarm Attributes
Spectrum	Alarm ID, Alarm Type, Alarm Message, Entity Name, Suppression Key, Created, Last Updated, Time Since Last Update, Alarm Attributes
UIM	Alarm ID, Alarm type, Alarm message, Entity name, Suppression key, created, last updated, Time since last update, Alarm Attributes

NOTE
Click the **Show Raw JSON** link to view the alarm attributes.

- **Monitoring Details**

Source Product	Details
Application Performance Management	Group, Configuration Item, Metric
Spectrum	Group, Model Type, Impacted Entities

Source Product	Details
UIM	Group, Probe, Monitoring host/robot, Source, Hub, Configuration Item, Metric, Item Type

- **Custom Attributes**

Source Product	Details
Application Performance Management	APM Isolation View, APM Alert Definition, APM Metric View (Click the required link to navigate to APM.)
Spectrum	Alarm URL (Click the **Source Product Link** to navigate to Spectrum.)
UIM	Alarm URL (Click the **Source Product Link** to navigate to UIM.)

- **Owner Details** (Assigned to, Acknowledge)
- **Ticket Details** (Ticket Status and Ticket ID)
- **Automation Actions**: Displays if the Automic integration is configured or not. Click to configure or enable.
- **Service Alarm Details** (Service Alarm ID). Click the **View Service Alarms** link to navigate to the **Service Alarms** page.
- **Situation Details** (Situation ID). Click the **View Situation Details** link to navigate to the **Situations** page.

Alarms Tab - Affected Metrics

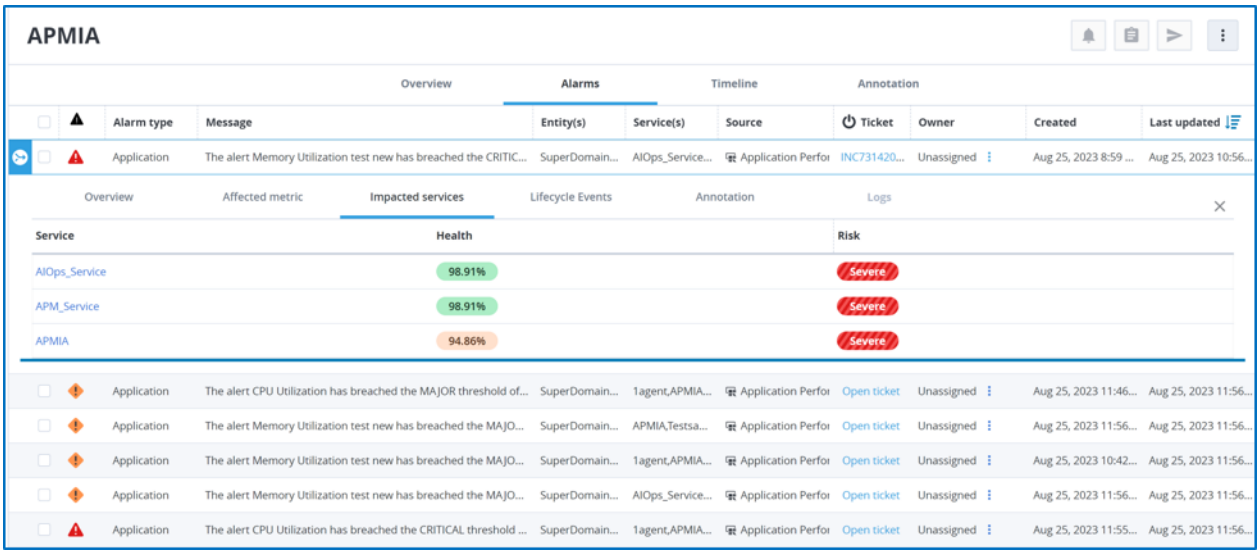
This tab shows the metric chart of the underlying metric. If the required fields are not available in the alarm, this tab is not shown. Probability bands are shown if the metric is configured with the proprietary Data Science Engine from Broadcom. If the metric is not configured with the Data Science Engine, the actual metric chart with original metric values appears. If the metric chart is not available, you must verify if the particular metric is ingested.

The metric chart displays anomaly alarms when a threshold is crossed. The threshold is determined based on historical trends. The Affected Metric tab has a **Compare Metrics** link that launches Performance Analytics from the context of an alarm and allows you to compare a single metric from different devices or multiple metrics from single or multiple devices. For more information about the metric charts, see the Performance Analytics section.

> **NOTE**
> By default, the chart time range is eight hours before the last alarm update to one hour after the last alarm update.

Alarms Tab - Impacted Services

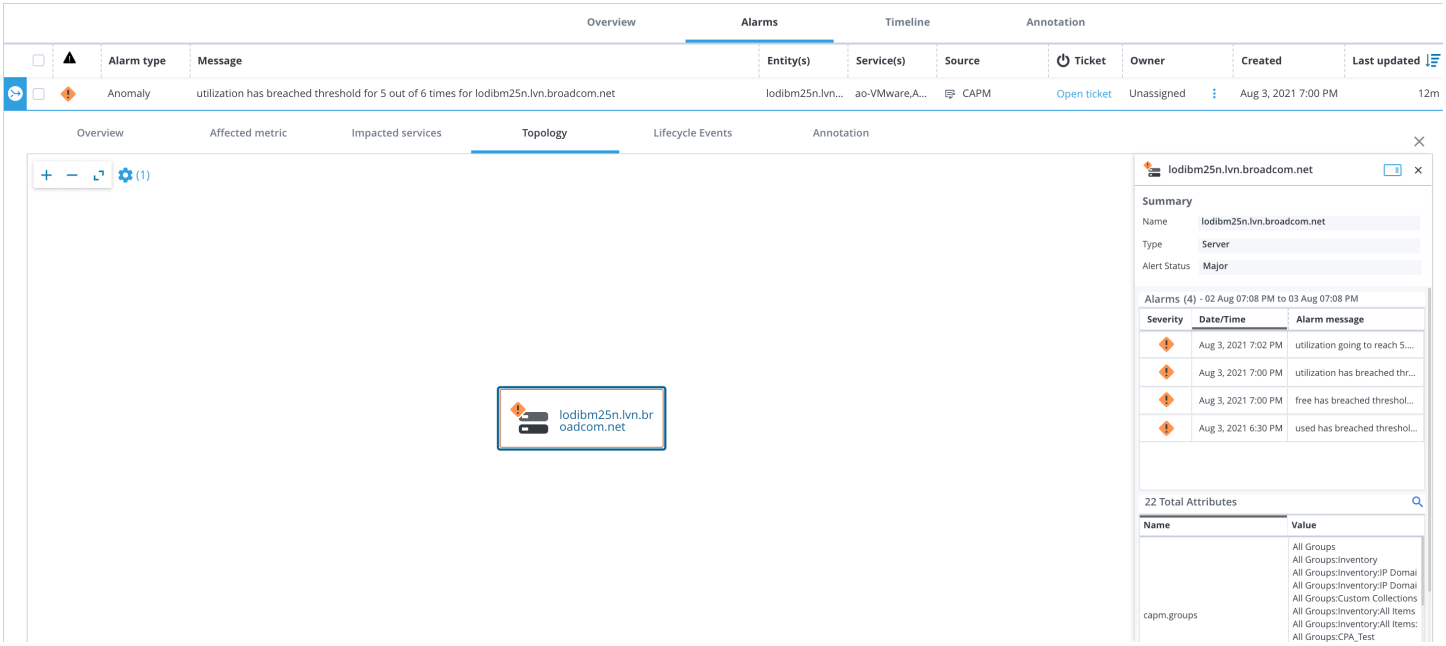
This tab provides details of the services that are impacted due to the selected alarm. Clicking a service redirects to the Service Analytics details view of that particular service. The table displays the impacted service metrics (such as users that are availing the service, actual service availability, and risk).



Alarms Tab - Topology

This tab provides topology details of the selected service or sub-service. You can view the topology details by clicking the topology icon

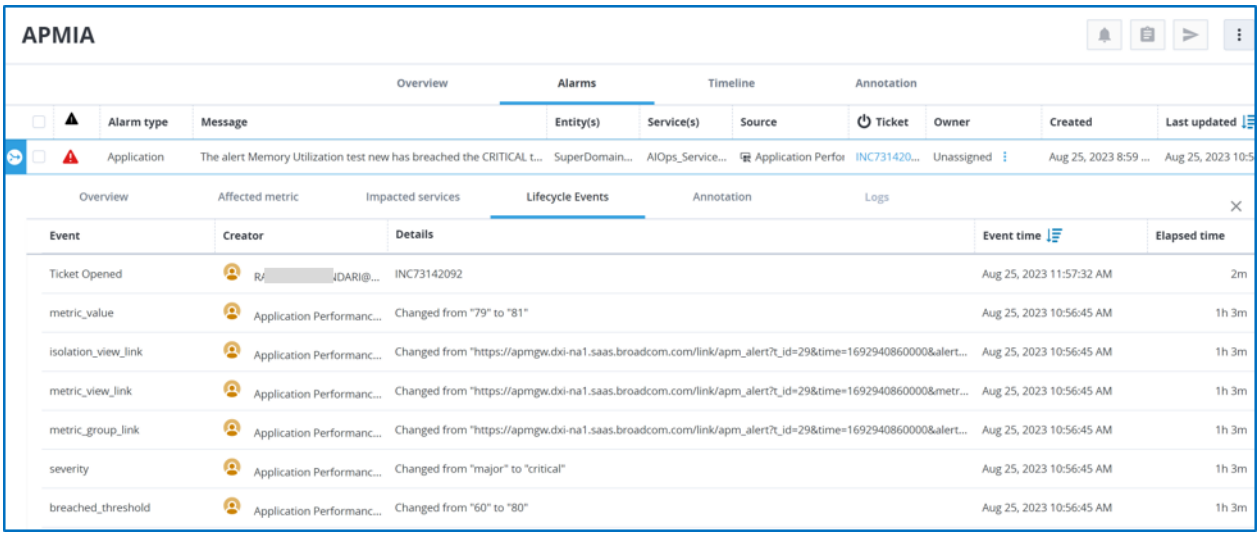
of the service. If you select a parent service and click the topology icon, the CI of its children appears along with the topology of the CIs of sub-services in different tabs.



You can click a service to view the topologies associated with that service. You can also view the summary of the service by clicking the service from the topology view.

Alarms Tab - Lifecycle Events

This tab provides the lifecycle of an alarm. This tab displays all the events that occurred for an alarm from the time it was created, such as created time, status updates, annotation updates, and threshold changes.



You can view the following details on the lifecycle events tab:

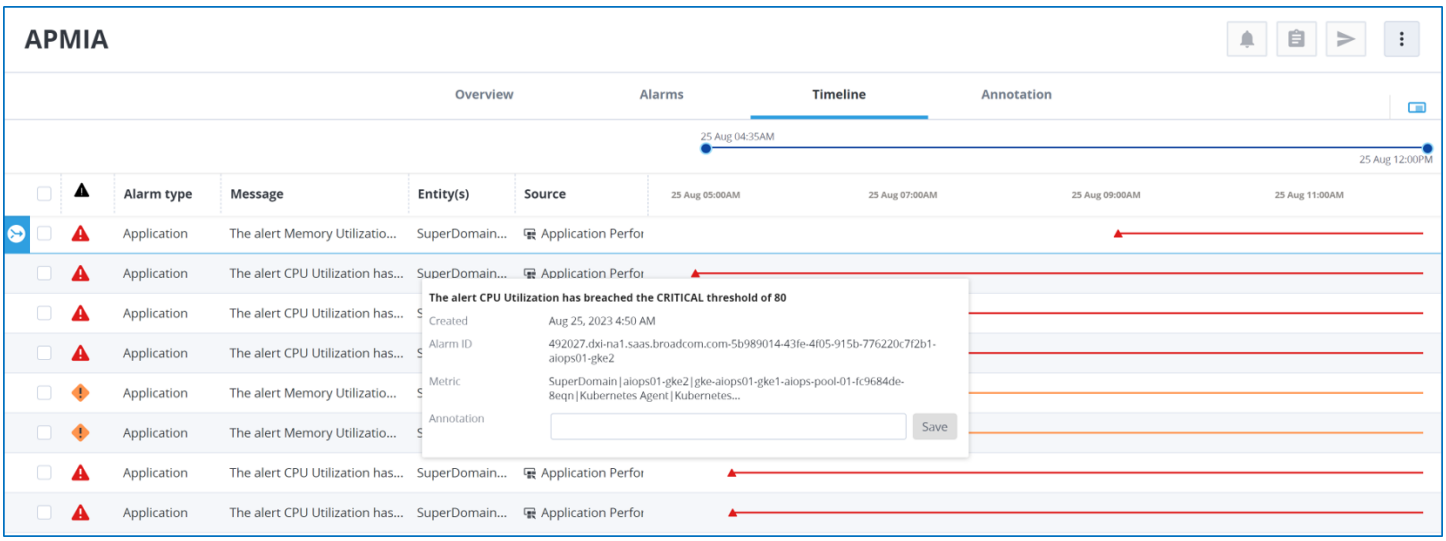
- Event
- Creator
- Details
- Event time
- Elapsed time

Alarms Tab - Annotation

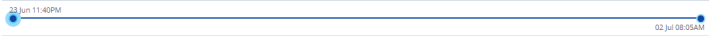
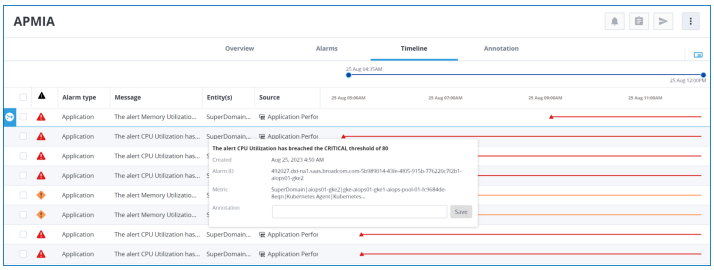
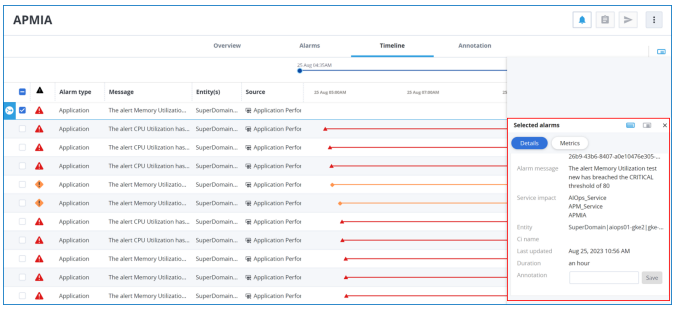
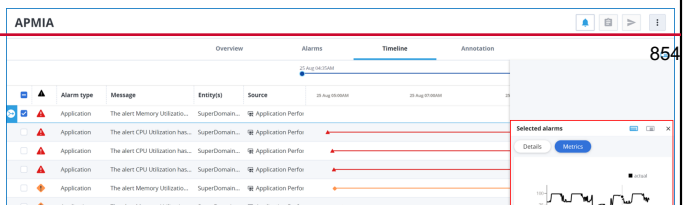
This tab enables you to add any additional information or details that you want to add to the selected alarm. You can add the required information in the **Annotation** tab of the alarm and click **Save** to view the annotation as a part of the alarm details. The annotation includes the information that you added, the details of the person who added this annotation, the date, and time details.

Timeline Tab

This tab indicates the timeline of the service alarm and the underlying raw alarms for a service alarm. The alarm on the deepest Configuration Item or device (as determined by Service Analytics) is the root cause alarm and is indicated by the icon. The **Timeline** tab displays the following information:

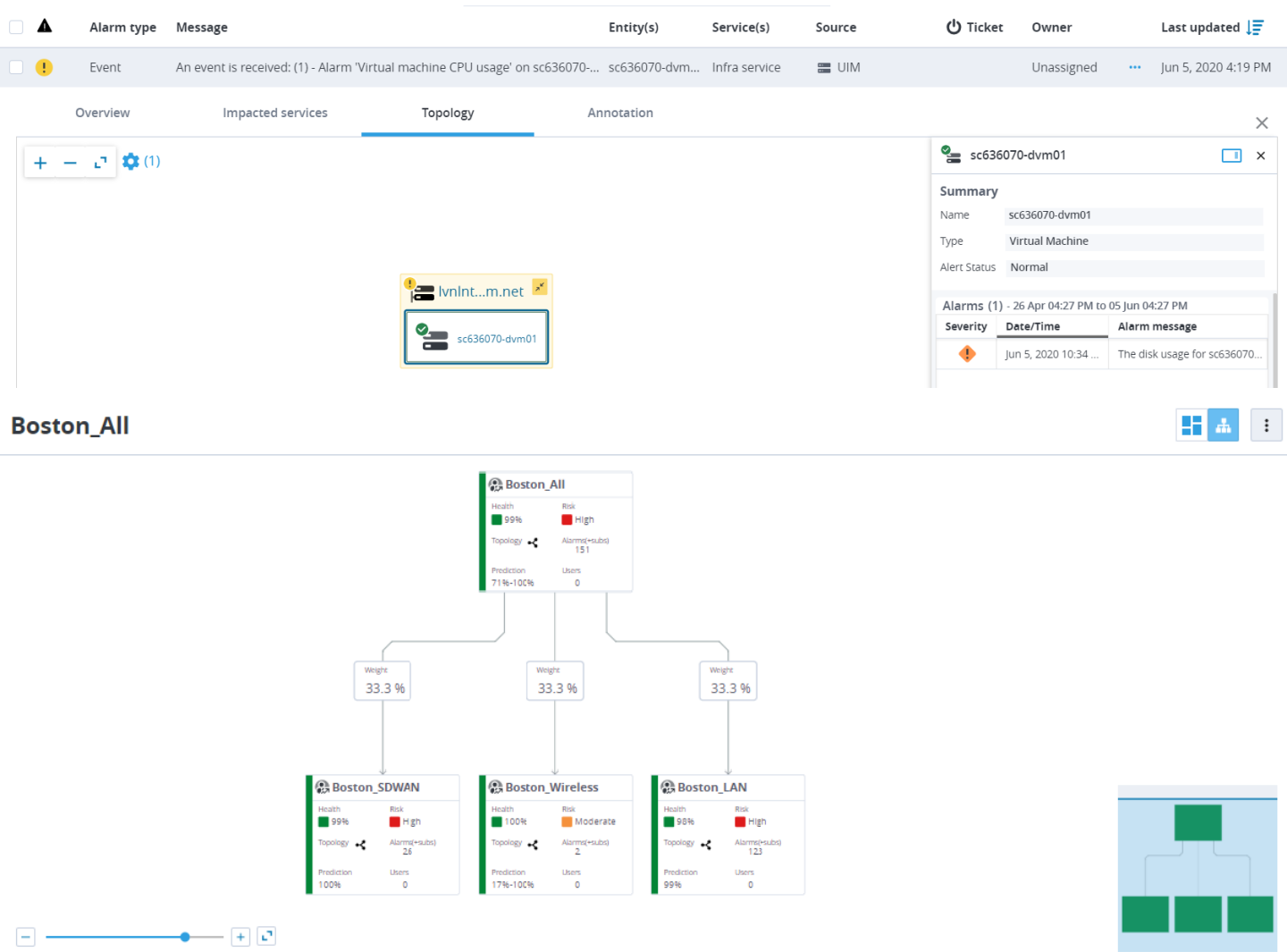


Column Name	Description
Column / Row-level Alarm Action	Enables you to perform row-level service alarm actions or column level to perform bulk service alarm action.
Severity	Indicates the severity of the service alarm.
Alarm Type	Indicates the alarm type.
Message	Displays the alarm description details.
Entities	Indicates the application or device name
Source	Displays the source product from which the service alarm is generated. This column supports the sorting operation.

Column Name	Description
<Date and Time Interval>	This time interval shows the journey of a service alarm. You can adjust the date and time interval, by dragging the horizontal scroll. You can drag and adjust the scroll on both ends of the timeline. The alarm timeline displays the alerts accordingly.  Click the diamond or small triangle to see the following information for a service alarm:  • **Created:** Displays the date and time at which the service alarm got created. • **Alarm ID:** Displays a unique service alarm ID • **Suppression Key:** Displays the suppression key details. • **Metric:** Displays the metric details of the service alarm • **Source:** Displays the source product details. • **Annotation:** You can add any additional information or details for the selected alarm and click **Save** option. **Metric Palette** To view the details and metrics of the service alarm, select an alarm and click the icon on the right end of the view. This section provides the following details: • **Details:** This tab provides the following details of the alarm:  – **Alarm ID:** Displays a unique id. – **Alarm Message:** Displays the service alarm message. – **Service impact:** Displays the list of impacted services. – **Entity:** Displays the application or device name. – **CI Name:** Displays the name of the Configuration Item. – **Last Updated:** Displays the date and time when the service alarm was last updated. – **Duration:** Indicates the duration of the service alarm. – **Annotation:** You can add any additional information or details for the selected alarm and click **Save** option. • **Metrics:** This tab displays the metric linear chart. 

Topology Tab

This tab provides the topology details of the selected service or sub-service. You can view the topology details by clicking the topology icon of the service. If you select a parent service and click the Topology icon, the CI of its children appears along with the topology of the CIs of sub-services in different tabs. You can click a service to view the topologies associated with that service. You can also view the summary of the service by clicking the service from the topology view. For more information on topology, see Topology Details.



NOTE
The **Topology** tab is displayed only when there is an inventory push from the source product, that is, the service alarm should have topology data for the **Topology** tab to be displayed.

Annotation Tab

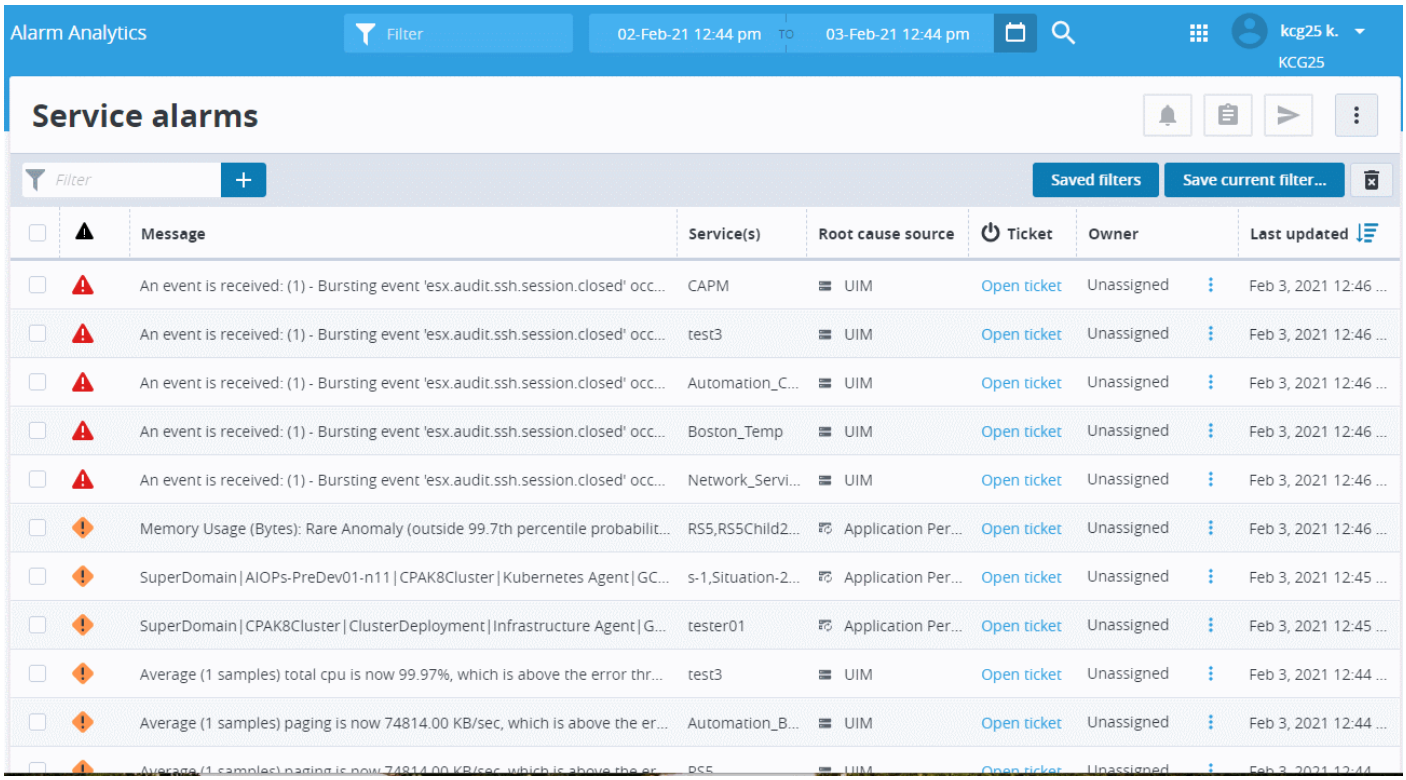
This tab enables you to add any additional information or details that you want to add to the selected alarm. You can add the required information in the **Annotation** tab of the alarm and click **Save** to view the annotation as a part of the alarm details. The annotation includes the information that you added, the details of the person who added this annotation, the date, and time details.

Delete a Service Alarm

To delete a service alarm, perform the following steps:

1. Select the required service alarm and click .
   A confirmation window appears
2. Click **Delete**.

   The service alarm gets deleted. For bulk delete operation, select multiple alarms and click the icon.



{"URL":["https://digital-oi/alarms-analytics/serviceAlarms","https://digital-oi/alarms-analytics/serviceAlarms/overviewTab","https://digital-oi/alarms-analytics/serviceAlarms/timelineTab","https://digital-oi/alarms-analytics/serviceAlarms/annotation","https://digital-oi/alarms-analytics/serviceAlarms/alarmTab/overviewTab","https://digital-oi/alarms-analytics/serviceAlarms/alarmTab/impactedServices","https://digital-oi/alarms-analytics/serviceAlarms/alarmTab/topology","https://digital-oi/alarms-analytics/serviceAlarms/alarmTab/lifecycleEvents","https://digital-oi/alarms-analytics/serviceAlarms/alarmTab/annotation"],"description":"concept.dita_727f12e0-f632-4c63-969a-c5bb4c8c57b3","troubleshooting":{"masterkb":"https://knowledge.broadcom.com/external/article?articleId=217181\",\"https://knowledge.broadcom.com/external/article?articleId=226977","text":"Service Alarms"},"customCards":[{"id":"IPCE_Alarmactions","type":"configure","title":"Service Alarm Actions"},{"id":"IPCE_filters","type":"configure","title":"Service Alarm Filters"},{"id":"IPCE_IPCE_servicealarmtabs","type":"configure","title":"Service Alarm Tabs"}]}

Anomaly Alarms

Anomaly Alarms are generated when a threshold is crossed for the configured metric value.

An anomaly alarm gets generated when a metric value deviation is detected by the Data Science Engine for the configured metrics, by using machine learning algorithms.

Anomaly Alarm Lifecycle Management

The Anomaly Alarm Lifecycle checks if there are no new anomaly alarms generated for the same metrics and automatically closes the anomaly alarms.

Anomaly Alarm Severity

The severity status of each anomaly alarm is indicated as Major.

Maintenance Window

The maintenance period stops all monitoring and metric calculations for the selected service, application, group, entities, or alarms.

Using Maintenance Windows, you can schedule a maintenance period for a Service, Agent, Group, Entity, or Alarm to perform any maintenance activity. During this period, update your policy to include *maintenance* attribute to *false* to silence your alarms and alarm notifications. For more information, see the Policies section.

However, you can still monitor the status and health of these entities. An alarm that raises within the maintenance time is tagged as on the Alarms page. You can create a maintenance schedule for once, daily, weekly, or monthly time period. To automate a maintenance schedule using APIs, see Global Maintenance APIs.

Overview

To perform maintenance activities on entities (Services, Agents, Groups, Entities, and Alarms), you must regularly schedule maintenance activities using the maintenance schedules. The maintenance schedule is a period of time that is designated to perform preventive maintenance activities that could cause a disruption of service. The **Maintenance Windows** page displays the various maintenance schedules which are currently active, inactive, or scheduled.

This page also displays the following details for each schedule:

- **Title:** Indicates the name of the maintenance schedule.
- **Description:** Indicates the description details of the maintenance schedule.
- **Status:** Indicates the current status of the maintenance schedule, which could be **Active**, **Inactive**, or **Scheduled**.
- **Start** and **End**: Indicates the start and end time of the schedule. This field defines the duration of the schedule.
- **Duration**: Indicates the duration of the maintenance schedule.
- **Recurrence:** Indicates the recurring maintenance schedule details.
- **Creator:** Indicates the name of the person who created the maintenance schedule.

Create a Maintenance Window

Before you create a maintenance window, note the following points:

- Except for Anomaly alarms, all other alarms are ingested during the maintenance period, and only notifications are suppressed. There is no change to the life cycle because of maintenance.
- When an entity is under maintenance:
  - Any new alarms or new services that are created during the maintenance period are marked under maintenance, and all the notifications are suppressed.
  - DX Operational Intelligence does not change the status of the existing alarms or service. When the entity is under maintenance, the existing alarms or existing services do not go under maintenance automatically. These existing

alarms or services are marked under maintenance only when the alarms or services are updated during the maintenance period.

- When an entity comes out of maintenance:
  - The new alarms or services do not go under maintenance.
  - The existing maintenance alarms or services still continue to be under maintenance. DX Operational Intelligence does not change the status of the existing maintenance alarms or existing services until and unless the alarms or services are updated again during the non-maintenance period.
- By default, the maintenance alarms are displayed in the **All Alarms** view with a **Maintenance** icon. You can also toggle the **Show maintenance alarms** switch that is available in the **All Alarms** view.
- By default, the maintenance services are displayed on the **Services Overview** page with a **Maintenance** icon.
- The alarms that are displayed in the **Alarms** tab are sample alarms and are for preview purposes only. Any new alarms that match the regular expression automatically go under maintenance.
- For the maintenance alarms to not appear in the default view, apply the filter **Maintenance:false** in the **All Alarms** view.
- For the maintenance services to not appear in the default view, apply the filter **Maintenance:false** on the **Services Overview** page.

Follow these steps:

1. From the **Settings** tab, click on **Maintenance Windows**.
   The Maintenance Windows page opens with all the existing maintenance schedules.
2. Click the **+ Add maintenance window** button.
   The **Choose affected entities** page opens with the **Services** tab as default.
3. Select Services, Agents, Groups, Entities, and Alarms to include in the maintenance window.
4. Selection the required options:
   - **Filter:** You can filter using the attributes to narrow down your search for a required maintenance schedule. You can either type in the **Filter** field directly and search, or type and select the filter attributes that pop up, or add a filter attribute by clicking on the **+** icon next to the **Filter** field.
     - **Services:** You can filter by Description, Location, Service, and Tags
     - **Agents:** You can filter by Agent.
     - **Group:** You can filter by Group and Source.
     - **Entity:** You can filter by Entity, IP Address, OS, Role, Source, and Type
     - **Alarms:** You can filter by Management Module, Message, and Metric Name

       NOTE

       - This filter supports regular expression and the following conditions: Equals, Not equals, Contains, Does not contain, Starts with, Does not start with, Ends with, Does not end with.
       - If you add different filter attributes in a single maintenance window, then the filter uses the AND operation to fetch the results. DX Operational Intelligence also uses the same filter criteria with the AND operation to mark the alarms for maintenance.

     Here is an example:

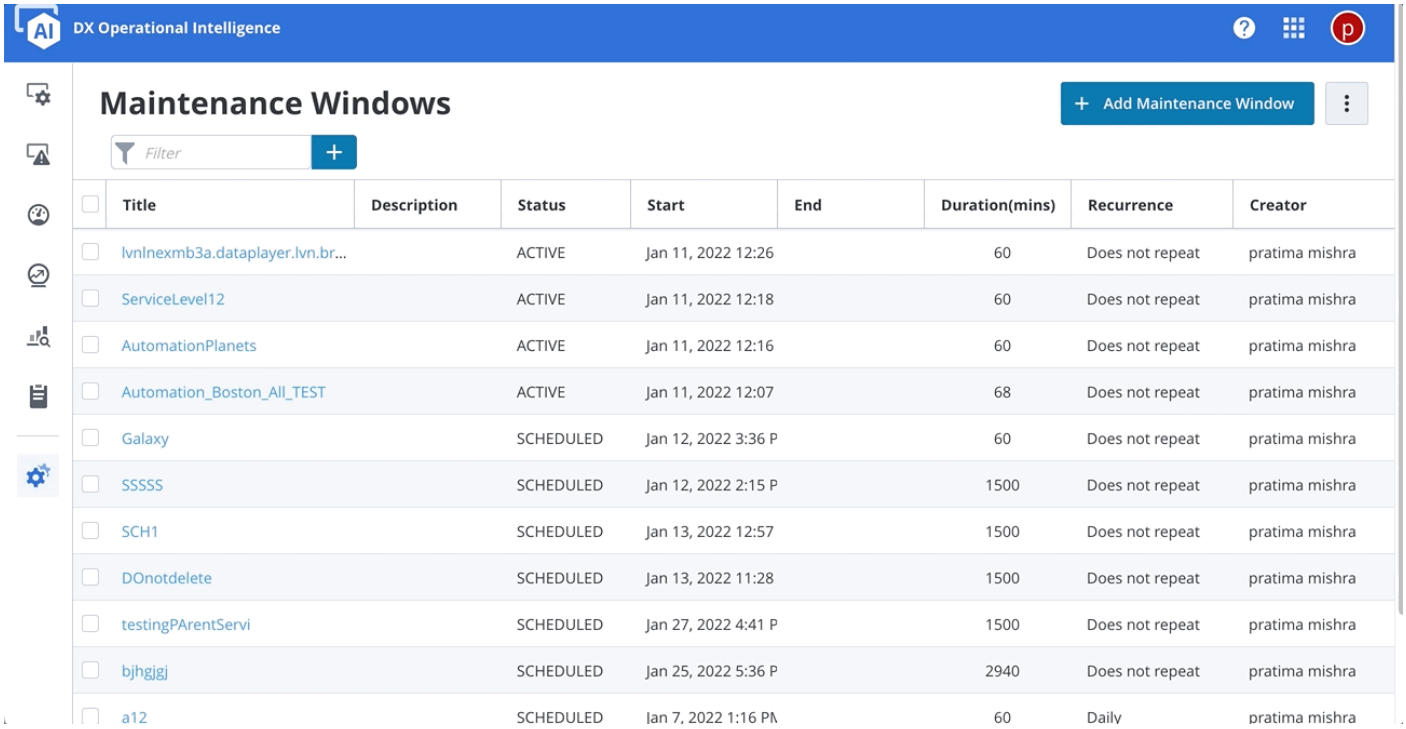
   - **Specific:** By default, the **Specific** option is enabled. Allows you to select the list of entities with or without a filter applied.
   - **Based on Filter, Include Sub-Service:** These options are enabled after the filter is applied. Selecting the **Based on Filter** option, selects all the entities that are displayed on the page (with the filter criteria applied). Select the **Include Sub-Service** option to select all the child services.

     NOTE

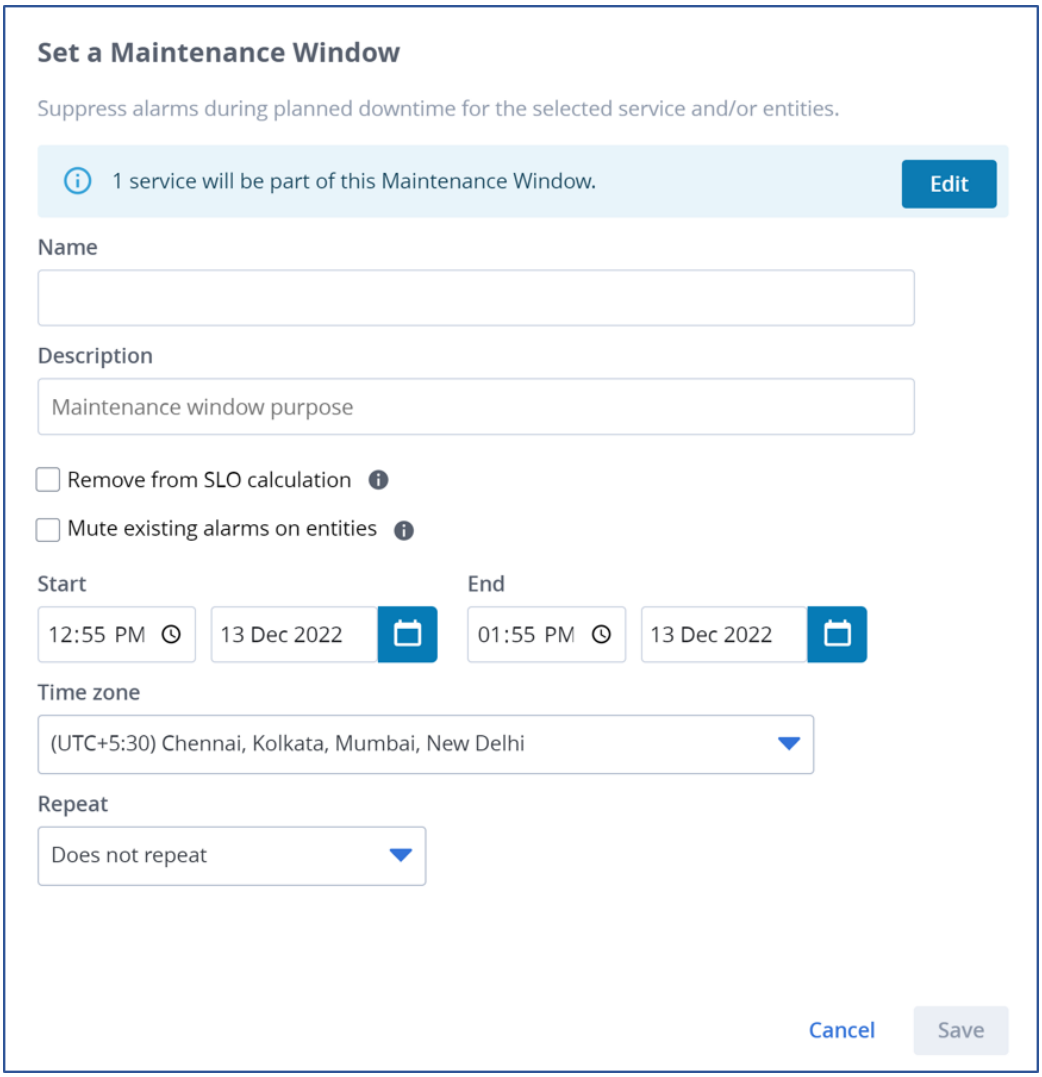
     The**Include Sub-Service** option is not displayed for Agents, Groups, Entities, and Alarms.
5. Select the services, agents, groups, entities, or alarms.

NOTE
In the context menu, if you want to create a maintenance window only for service, select **Add Service** and if you want to create a maintenance window for the parent service and child sub-services, select **Add service & sub-services**. In this example, let us select **Add Service**

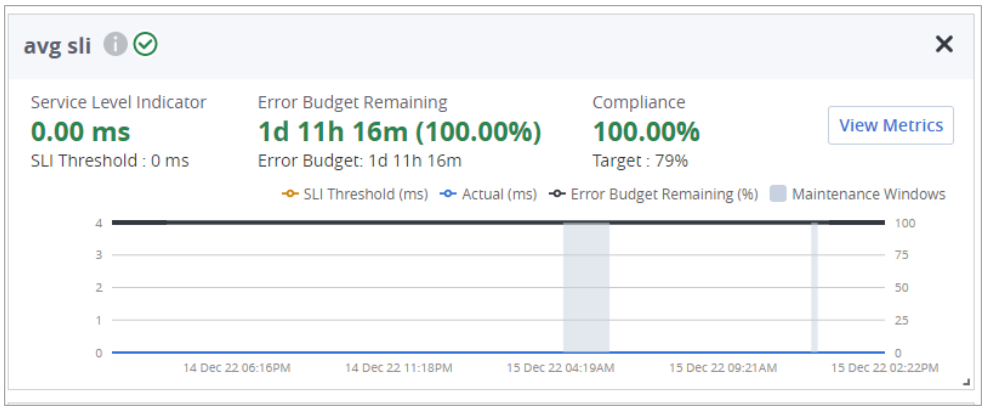
The following example shows the options available to select a service.



6. (Optional) To view the selected entities, enable **Display Only Selected** switch that is on the top-right corner of the page.
7. Click **Continue**.
   The **Set a Maintenance Window** appears.



8. Provide the following details:
   - **Name:** Enter a name for the schedule.
   - **Description:** Enter a description for the schedule.
   - **Remove from SLO calculation:** Select this option to exclude the SLO calculation during the maintenance window. If you select this option,
     - The SLO calculation is stopped, and only SLI is calculated during the maintenance period.
     - The Service Level Indicators widget displays this period in gray as shown:



Click the **Maintenance Windows** legend to clear the gray section on the chart. For more information, see the Add Service Widgets section.

– **Mute existing alarms on entities:** Select this option to mute the existing open alarms that were raised before or during an active maintenance period for the selected entities. When you select this option, all updates to the existing alarms which are part of this maintenance window are muted during the maintenance window and restored to their original state when the maintenance window ends. For example, the alarm is restored to the open state if the alarm was not closed during the maintenance period.

  During the maintenance period, the muted alarms are grayed out, and a maintenance icon is also displayed for those alarms. If multiple maintenance windows are scheduled at the same time on the same entity, and if this option is selected in at least one of the schedules, then the existing alarms for all the schedules are muted.

  NOTE
  If you end an active maintenance window or you delete a window, the alarms are restored to their original state.

– **Start** and **End**: Set the start and end times of the schedule. This field defines the duration of the schedule.

– **Timezone:** Select the timezone for the schedule.

– **Repeat:** Select the **Custom** option if you want a recurring maintenance schedule.

  NOTE
  You cannot create a recurring maintenance schedule to occur on the same day within the same schedule. For example, if you have created a maintenance schedule for a service to run from 2 PM to 3 PM, you cannot create within the same schedule a service to run from 5 PM to 6 PM. You must create a maintenance schedule for such same-day requirements.

  a. **Every:** Set the time period based on Days, Weeks, or Months.
  b. **End:** Set the end time for the recurring maintenance window using the calendar. If you do not want to end the recurring maintenance window, select **Never**.

– Click **Save**.

  Your maintenance window gets listed in the existing list of maintenance windows

Edit a Maintenance Window

To edit a maintenance window, perform the following steps:

1. From the **Maintenance Window** page, click on an existing maintenance window.
2. To modify the existing affected entities, select the entity and click **Edit**.
3. You can modify the following details:
   a. **Name:** Enter a name for the maintenance window
   b. **Description:** Enter a description for the maintenance window.
   c. **Start** and **End**: Set the start and end times for the maintenance window.
4. Click **Save**.

The changes get updated.

Delete a Maintenance Window

To delete a maintenance window, perform the following steps:

1. On the **Maintenance Window** page, click on an existing maintenance window.
2. Click on **Delete**.
3. Alternatively, you can delete a maintenance window from the **Maintenance Window** page by selecting the checkbox next to the maintenance window and clicking , **Delete** option.
   The selected Maintenance Window gets deleted.

Manage an Active Maintenance Window

You can update an existing active maintenance window by adding or removing entities and by editing the end time of the maintenance window.

Follow these steps:

1. On the **Maintenance Window** page, select the entity which is in an **active** state.
2. Click **Edit**.
3. Add or remove entities (services, agents, groups, entities, and alarms) based on your requirement.
   NOTE
   If you remove the services from an active maintenance window, the alarms get generated on that service from the time the maintenance window is updated.
4. Click **Continue**.
5. Set the **End time and date**.
   NOTE
   You cannot edit the description, start time, time zone, and recurring maintenance schedule for the active maintenance window.
6. Click **Save**.

The changes are applicable from the next maintenance window.

End an Active Maintenance Window

To end an active maintenance window, perform the following steps:

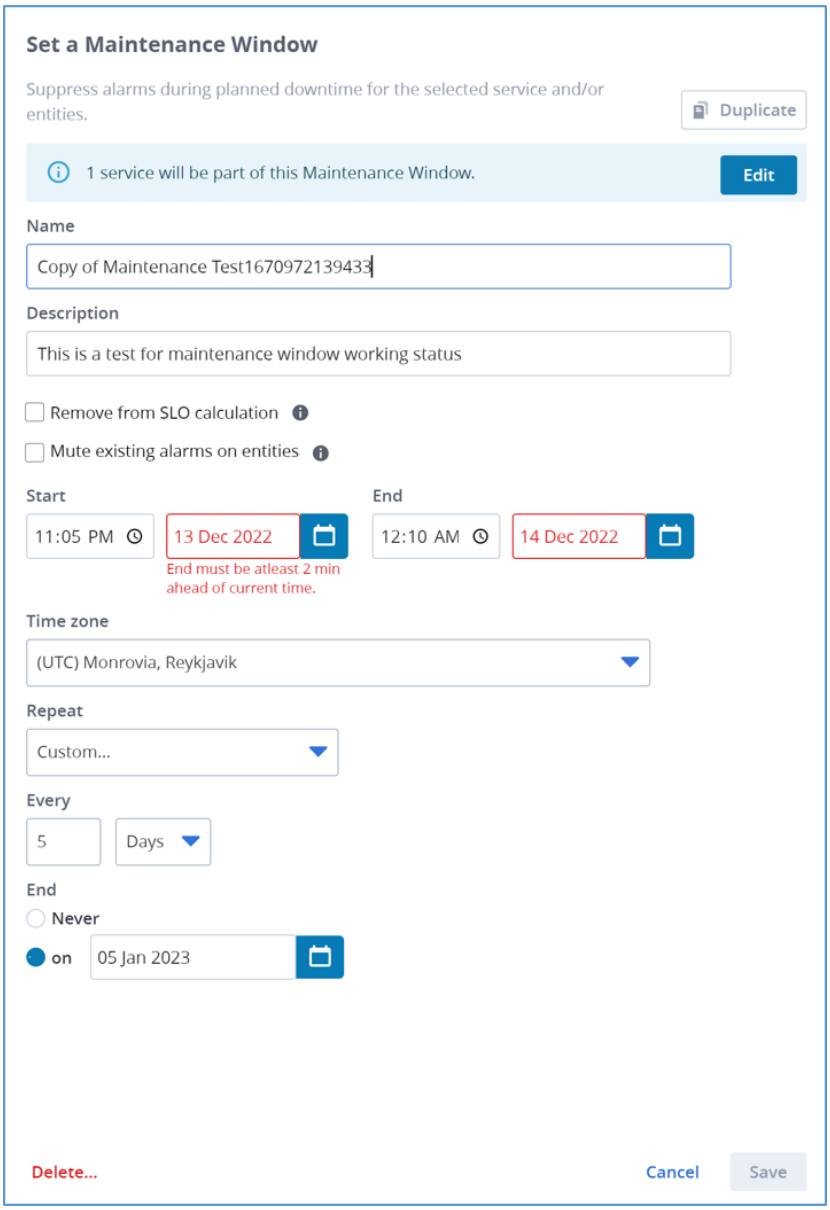
1. On the **Maintenance Window** page, select the checkbox of an existing maintenance window.
2. Click the icon at the top-right.
3. In the context menu, click on the **End 'Active' window** option.
   The selected active maintenance window ends.

Duplicate Maintenance Window

Use this feature to quickly duplicate an existing maintenance window with updates such as dates or content.

Follow these steps:

1. On the **Maintenance Window** page, click an existing maintenance window.
2. Click **Duplicate** on the **Set a Maintenance Window** page.
   The existing maintenance window name is prepended with **[Copy of] - [existing maintenance window name]**.



a. Edit the name Set the Start and End date/time.
b. Set the recurrence for the maintenance window.

NOTE
If the original maintenance window has no recurrence and the date is in the past, the **Save** button is disabled, and the dates field is highlighted in Red.

3. Click **Save**.

{"URL":["https://digital-oi/settings/maintenance-mode"],"customLabelGetStarted":"Configure Maintenance Window","description":"concept.dita_c825d2dd-e56f-49d7-8131-fd8aeeb77323"}

Audit Trail

An Audit Trail logs user activities, data access, actions taken on all the alarms such as acknowledge, close, assign, clear and so on.

DX Operational Intelligence is now enhanced to support Audit Trail. An analysis of the audit trails enables you to understand the issues that occurred and who (or what) caused the alarms and track potential security breaches. Audit Trail provides you with insight and oversight abilities that you need to increase efficiency and security in a reliable way.

Configure Automic Automation

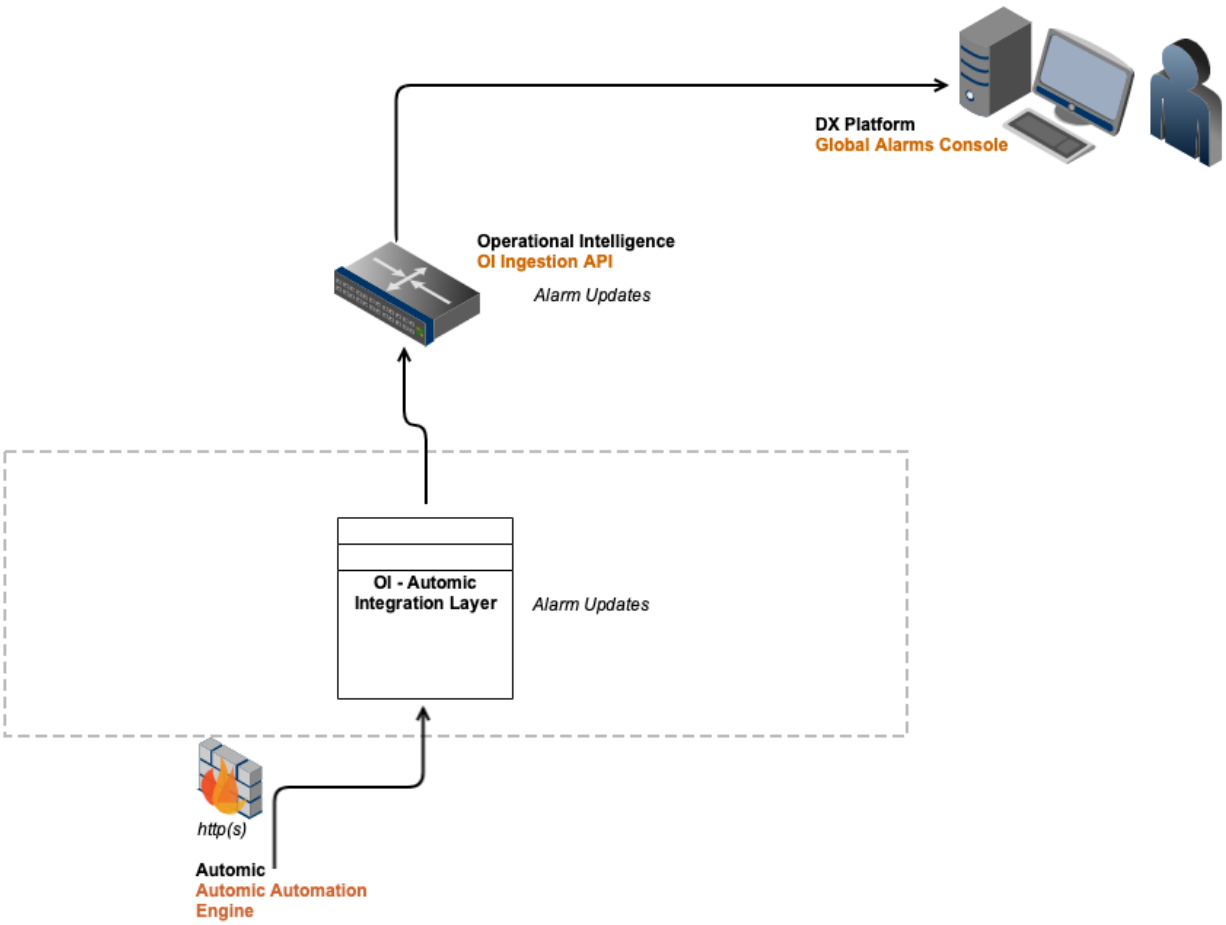
When you integrate an automation platform such as Automic Automation, you can leverage the automation features of Automic Automation for creating and executing such remediation jobs.

Content:

Overview

DX Operational Intelligence enables you to analyze and act on alarms generated by AIOps products such as APM and IM to know the root cause. The DX Operational Intelligence alarm clustering algorithm and smart Root Cause Analysis feature clusters alarms into meaningful groups and identifies the most-probable root cause - which saves a lot of effort and redirects to the appropriate points. Once the software determines the root cause of a particular alarm, you get an option to automate actions (remediation, collection, or otherwise) to be taken on the alert using an Automation platform. When you integrate an automation platform such as Automic Automation, you can leverage the automation features of Automic Automation for creating and executing such remediation jobs. This integration enables you to use a single UI (DX Platform) to view alerts, analyze their root cause, and trigger automated remediation workflows.

The following diagram illustrates how AIOps and Automic Automation integrate to provide recommended remediations.



Prerequisites

- Verify that your environment for DX Operational Intelligence and for Automic Automation is deployed.
- Verify that your Automic Automation environment has been mapped to the DX Platform attributes and can execute the jobs for an event.
- The AIOps folder is present in Automic Automation which contains the default workflows.
- Ensure that the Automic Automation payload contains the *&ALARM_JSON#|json* variable. For more information, see PromptSet.

Supported Version

DX Operational Intelligence	Automic
SaaS	On-Premise
On-Premise	On-Premise

This integration is API-based between DX Platform and Automic Automation. You may also choose to integrate using the generic webhook integration.

Configure and Deploy DX Gateway

Before you configure Automic, download and configure DX Gateway.

Follow these steps:

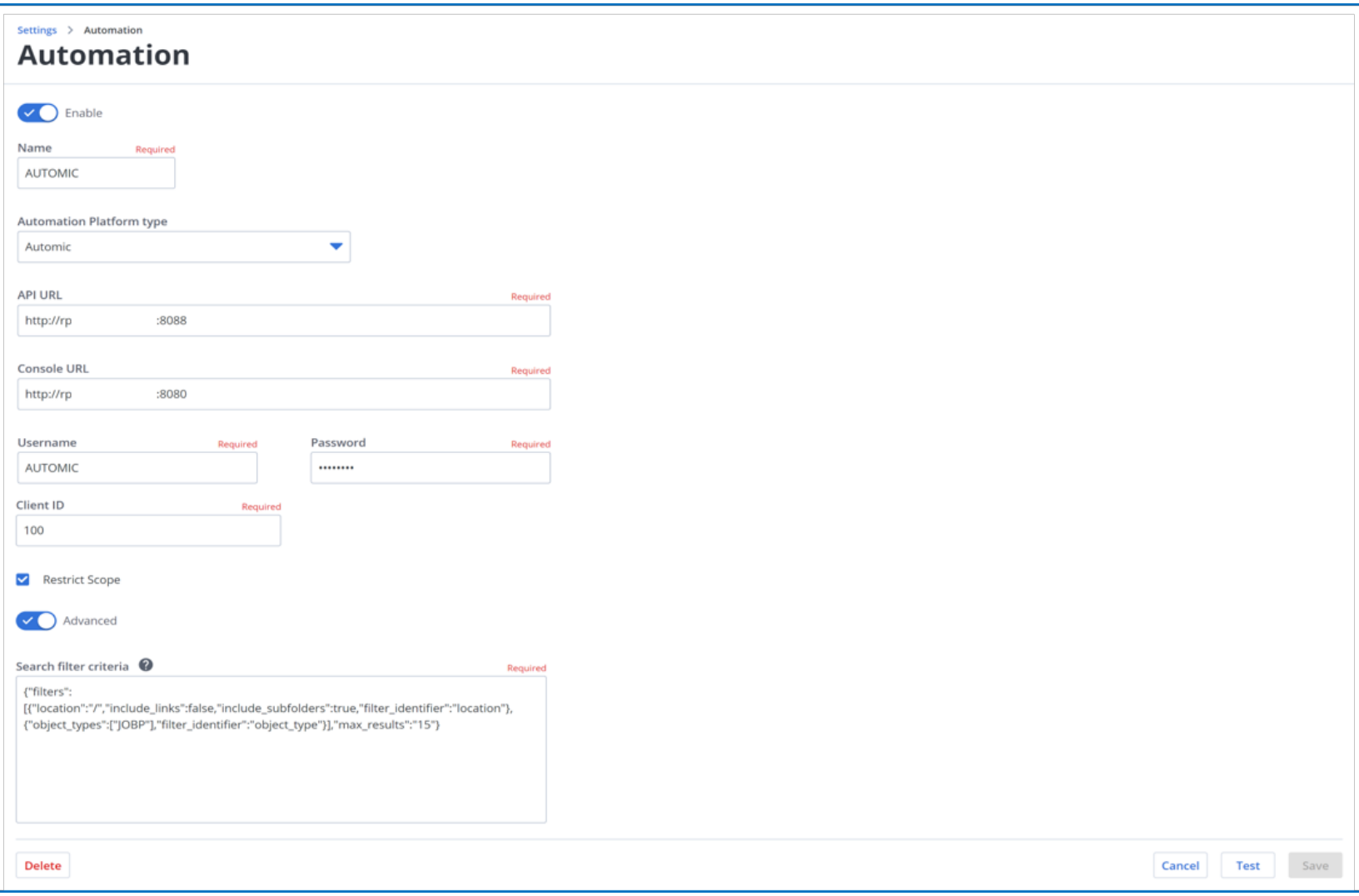
1. Log in to DX Operational Intelligence.
2. Click the **Setup Data Sources** tile on the **Settings** page.
3. Download **DX Gateway** from the **Downloads** page.
4. Go to the **DX-Gateway\Config** folder.
5. Update the following information in the **generic_config.json** file:
   – **dxsaas_gateway_ws_url:** You can find this information on the **Connector Parameters** page. Enter the value that is displayed for TAS/NASS endpoint. For more information, see the Connection Parameters page.
   – **dxsaas_gateway_token:** You can generate this token on the Tokens page. For more information, see the Tokens Management section.
6. Set the **JAVA_HOME** environment variable to point to the JRE installed on the local machine.

   NOTE

   - JAVA_HOME has to be set to Java 17 or above.
   - To confirm if the variable has the correct path set, open the command prompt and run %JAVA_HOME%.
7. Open the command prompt and navigate to the **Remediation** folder.
8. Execute the **startremediation.bat** file (for Windows) or the **startremediation.sh** script (for Unix).
9. Monitor the log file (DX-Gateway\Remediation\remediation-svc.log) for any errors.
10. Configure Automic Automation as described in the next section.

Configure Automic Automation

The following snapshot illustrates the Automation settings page.



Follow these steps:

1. Open the Launch Pad.
2. Navigate to **Settings**, and select **Automation** under **Integrations**.
3. Select the **Enable** option to activate.
4. Complete the following mandatory fields:
   - **Name** - Specify a name for the Automation connection.
   - **Automation Platform type** - Select a platform type from the list of options.
   - **API URL** - Specify the Automic Automation (Automation Engine) REST API URL. This enables DX Platform to interact (trigger or monitor executions, or search objects) with the Automic Automation Engine.
   - **Console URL** - Specify the Automic Automation (Automic Web Interface) console URL.
   - **Username** - Specify the Automic Automation user name.
   - **Password** - Specify the user's password.
   - **Client ID** - Specify the user's client number.
5. (Optional) Select the **Restrict Scope** option to reduce the list of permissions granted to the connection.
   a. If you select **Restrict Scope**, you need to specify the payloads in **Search Payloads**. For example, location or object_name.

b. To enter specific search filter criteria, select **Advanced**. The filter criteria allow advanced users to implement the same restrictive rules using free-form text area and formatted text, which improves the accuracy and focus of the recommendations. The following search filter parameters are allowed:
   - Platform
   - Application Name
   - Action Type
   - Location

   The following snippet is an example of search criteria for the location executables.

```
{
  "filters": [
    {
      "location": "/AIOPS",
      "include_links": false,
      "include_subfolders": true,
      "filter_identifier": "location"
    },
    {
      "object_types": [
        "JOBP"
      ],
      "filter_identifier": "object_type"
    }
  ]
}
```

6. Click **Test** to preview the Automic Automation connection using the settings specified (API URL, Username, Password, and Client ID). If the connection fails, you cannot save the existing settings. If the connection is successful, you can preview the workflows.

   NOTE
   If you update any of the fields API URL, Username, Password, or Client ID, then click **Test** to preview the connection before saving the settings.

7. Click **Save**, once the test connection is successful.

Set Environment Variables

Set the following environment variables:

Environment Variable	Description
SKIP_UPDATE_EXCLUDE_ATTRIBUTE_SITUATION='automicJobs,rc_subClusterName,rc_hostIp,rc_ticketUrl,annotation,ackn	By default, the following attributes are excluded when a situation is updated. If you do not want the situation to be updated based on any attribute, update this variable in the Notify Filter pod and also remove that attribute from the SKIP_UPDATE_INCLUDE_ATTRIBUTE_SITUATION environment variable.
SKIP_UPDATE_INCLUDE_ATTRIBUTE_SITUATION='rc_subClusterUrl,rc_hostIp,mostImpactedHost,mostImpactedServices,r	By default, the following attributes are included when a situation is updated. If you want the situation to be updated based on any attribute, update this variable in the Notify Filter pod and also remove that attribute from the SKIP_UPDATE_EXCLUDE_ATTRIBUTE_SITUATION environment variable.

Using Automic Automation

When you integrate an automation platform like Automic Automation, you can leverage the automation features of Automic Automation for creating and executing remediation jobs. This integration enables you to use a single UI (DX SaaS) to view alerts, analyze their root cause, and trigger automated remediation workflows.

This section contains the following topics:

Recommended Automation Actions in Global Alarms Console

Once you integrate Automic Automation, you can run automated remediation actions on any alert in the Global Alarm Console as defined during the configuration. Every time you open an alert, basic metadata is extracted from the alert and provided to the recommendation engine to get relevant actions. You can view one recommended action along with the confidence percentage information.

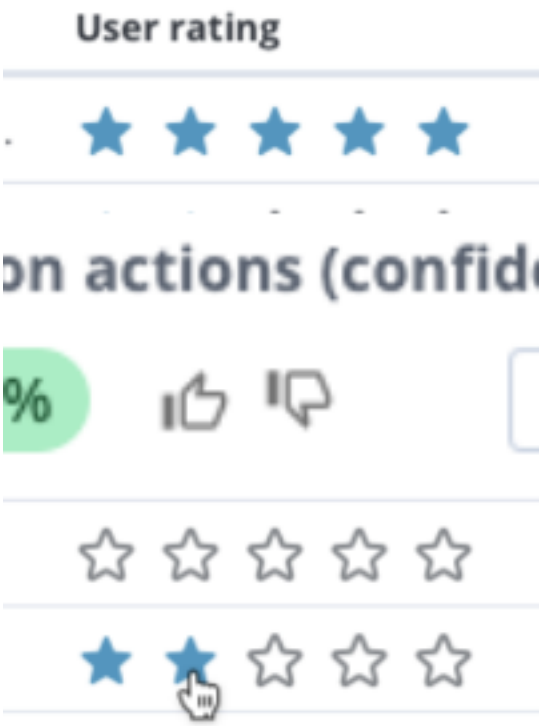
> **NOTE**
> If the Automation option has not been enabled in the Automation settings page, then you are prompted to contact the Administrator to set-up Automation Actions.

How Relevant Remedial Actions Are Recommended

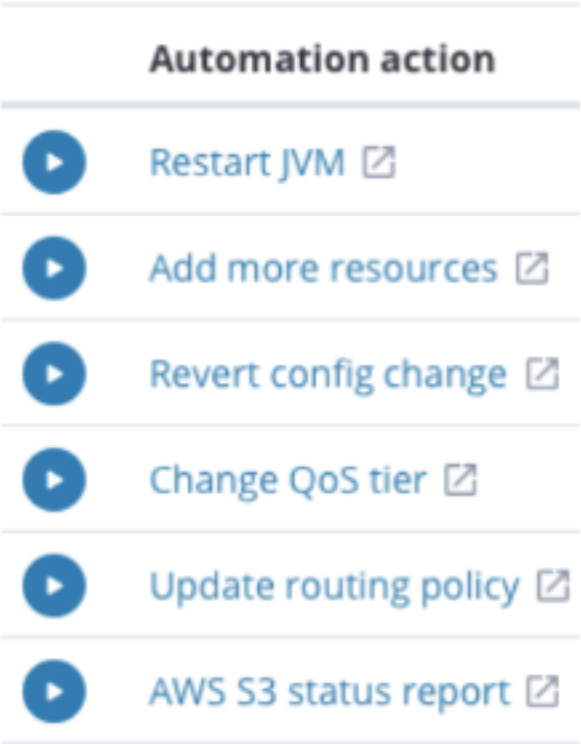
The recommended remedial actions for a specific alarm are based on a combination of explicit and implicit feedback which determines if the action is positive or negative.

Explicit feedback example - The user selects like or dislike after running a particular action, which determines if the action is positive or negative.

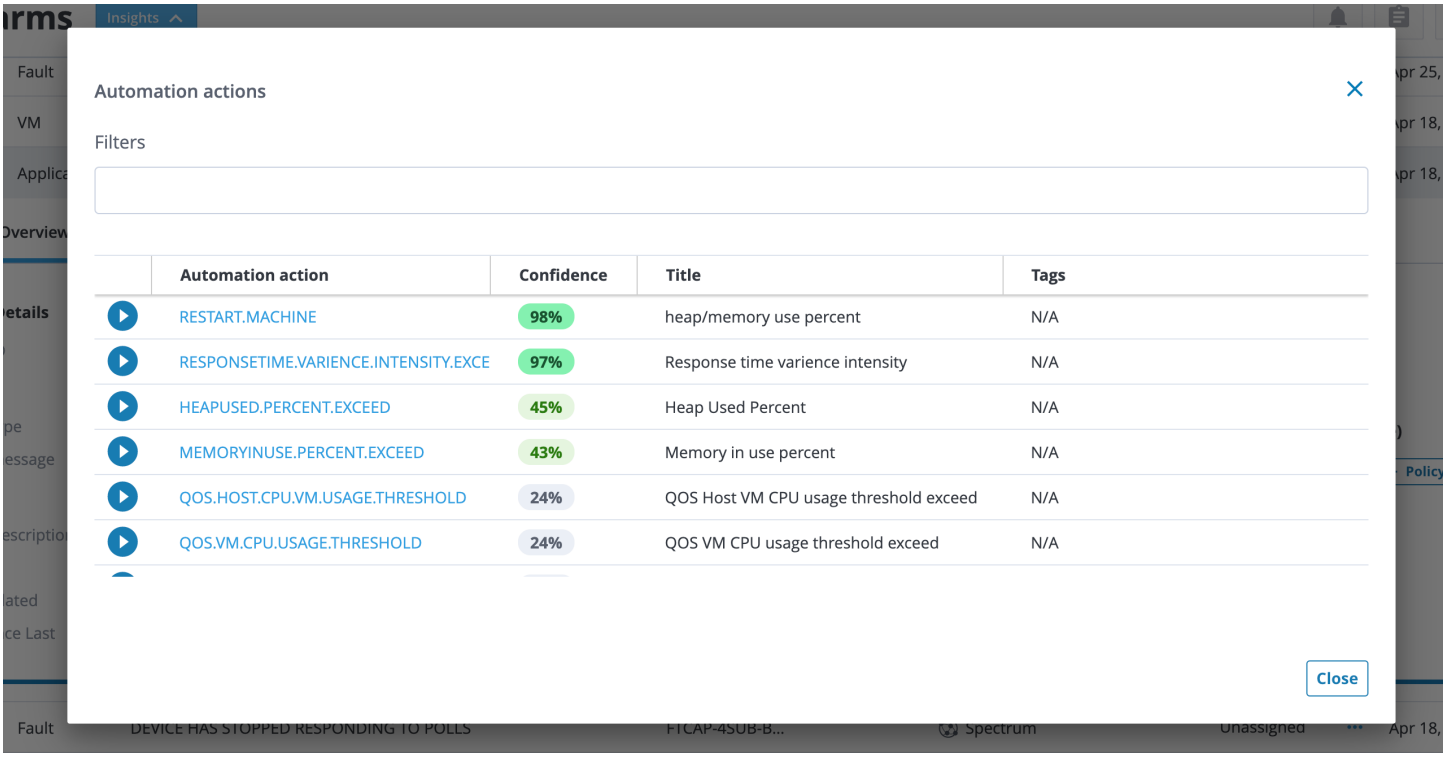
Explicit feedback example - The user rates an action on a scale of 1 to 5 which determines if the action is positive or negative.



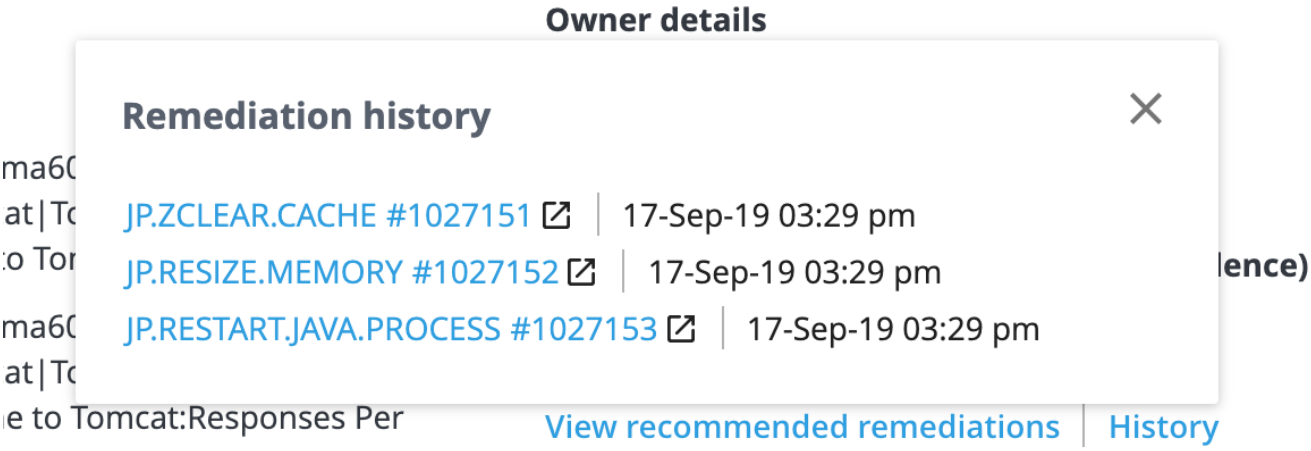
Implicit feedback example - The user initiating a particular remedial action implies that the action is positive. From the example below, if you run Change QoS tier, then the other three higher recommendations (Restart JVM, Add more resources, Revert config change) are incorrect and the feedback is captured.



The following snapshot illustrates the **Recommended Automation Actions** in the Alerts Overview tab:



All older actions are listed in the Remediation History tab along with their execution status and sorted by time of execution. The following snapshot illustrates the Remediation History tab:



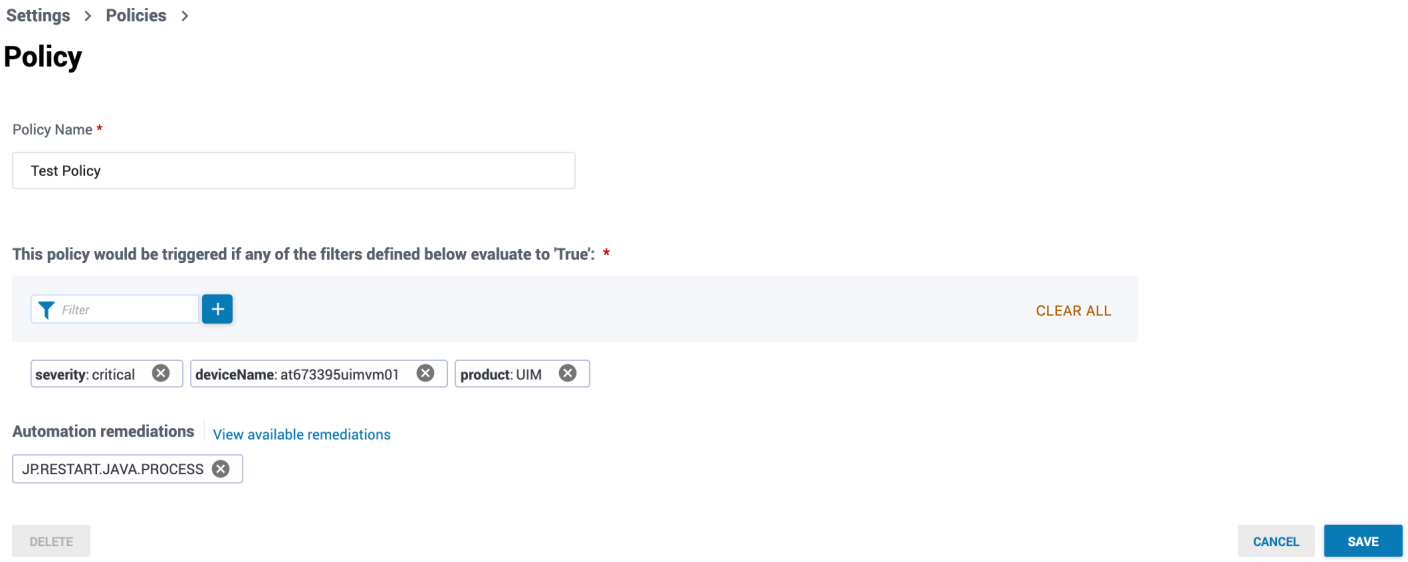
Create Policy from Workflow Recommendations

You can create a policy based on the recommended remediation actions.

Follow these steps:

1. Navigate to the Global **Alarms** Console and select any Alert. The Alerts Overview tab appears along with the Recommended Automation Actions.
2. Review the recommended action and click **+ Policy**.
3. The Policy creation page appears with the details of the alarm/recommended action automatically populated with the following information:
   - The severity of the alarm
   - Device ID of the device on which the alarm is triggered
   - Alarm title/message
   - Source product of the alarm
   - Metric for which the alarm was triggered
4. You can select additional filter attributes as follows:
   - Acknowledged
   - Action Status
   - Alarm Age
   - Alarm Description
   - Alarm Domain
   - Alarm Name
5. Select Automation Remediations to view the list of available remediations.
6. Select the remediation action you want to associate with the Policy.
7. Click **Save**.

Once the policy is defined, when any alert matches the defined filter criteria, the automation selection would be executed, along with any other channels associated with the policy. The following snapshot illustrates the Policies page:



Insights

You can configure DX Operational Intelligence to provide insights into services and raw alarms using the **Insights** tile from the **Settings** tab. All these insights are enabled by default and you can enable or disable them as required.

This section provides the following information:

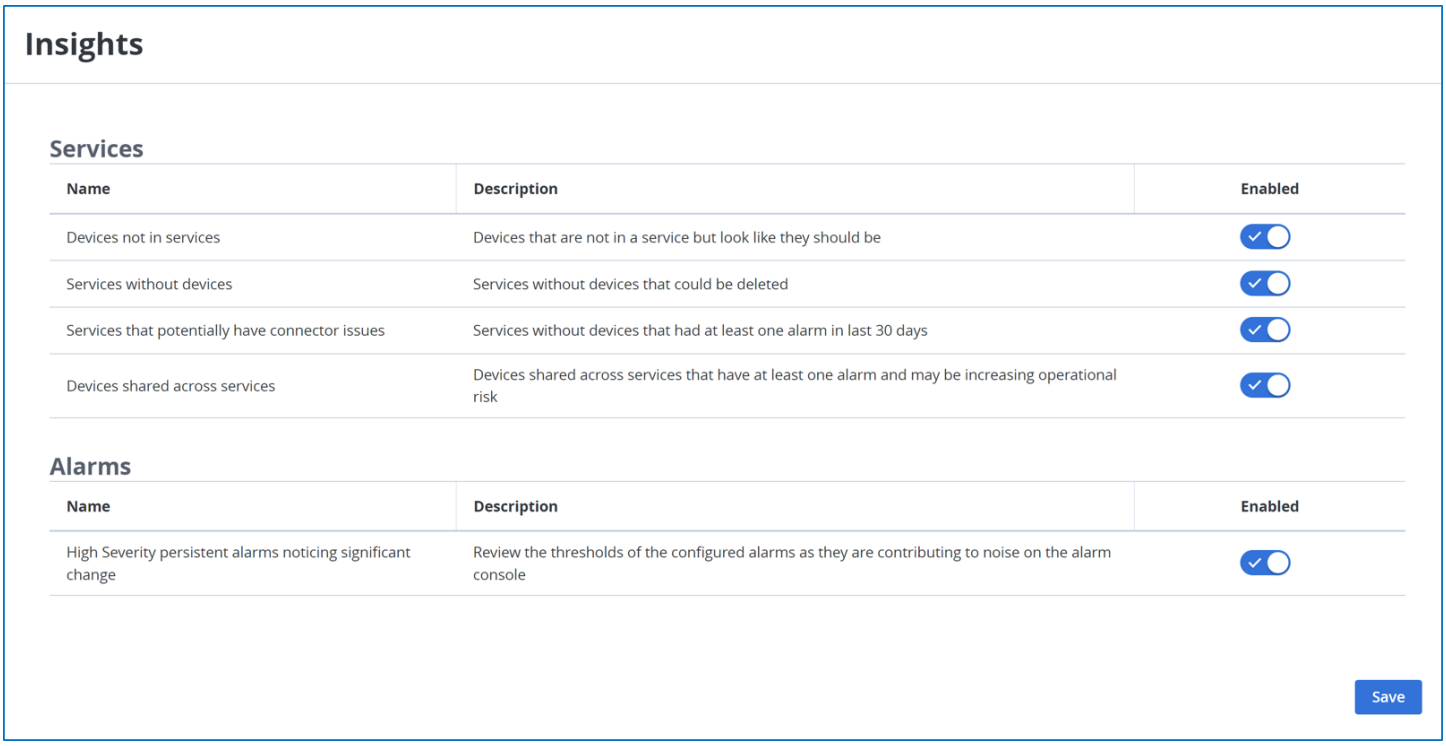
- Insights Overview
- View the Insights
- Re-enable the Insights

NOTE

- By default, this feature is disabled. To enable this feature for your tenant, contact **Broadcom Support.**
- The DX Operational Intelligence insights do not honor the universe and hence displays the same insights for all the universes.

Insights Overview

The following image illustrates the insights that are available out-of-the-box:



- **Services**
  - **Devices not in services:** Displays the devices that are not in service but look like they should be.
  - **Services without devices:** Displays the services without devices that you may delete if not required.
  - **Devices shared across services:** Displays the devices that are shared across services that have at least one alarm and may increase the operational risk.
  - **Services that potentially have connector issues:** Displays services without devices with at least one alarm in the last 30 days.
- **Alarms**
  - **High priority policies having persistent alarms:** Displays high priority policies having persistent alarms that might need updating.

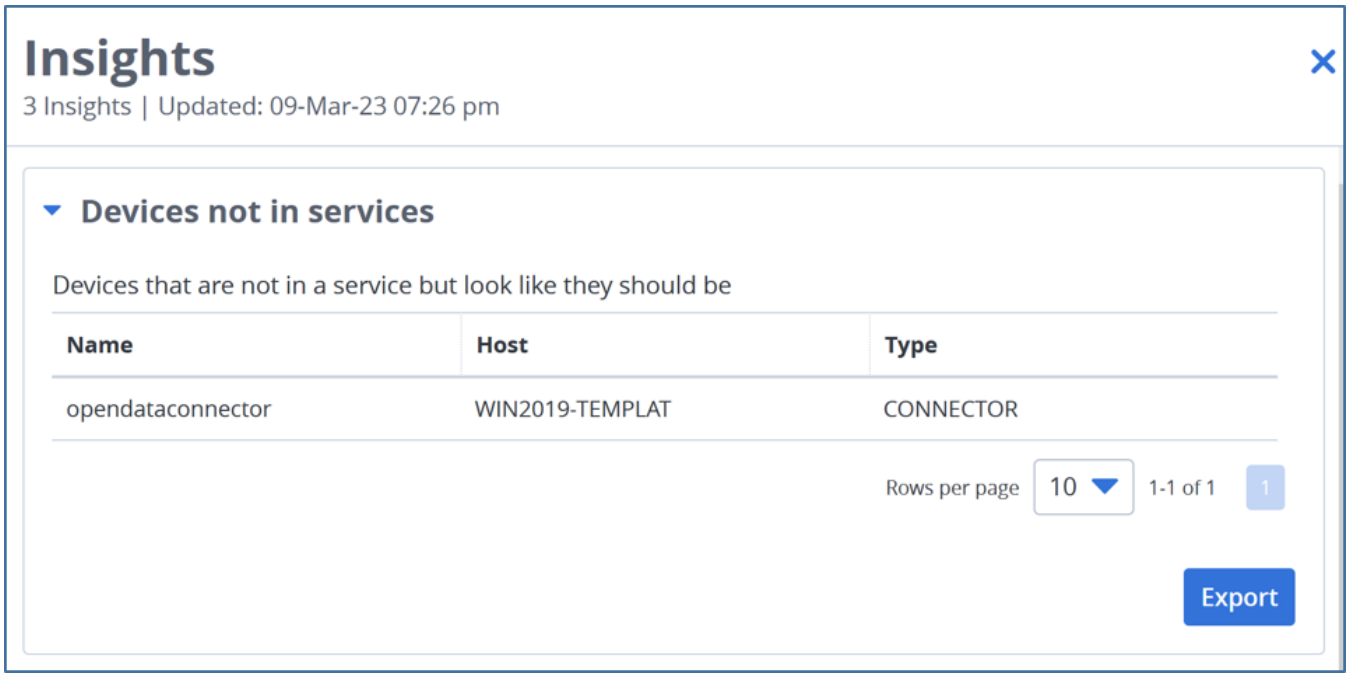
View the Insights

Information for the enabled insights is displayed on the Service Details and All Alarms pages. Click **Insights** that is displayed on the top-right corner to view the details.

- Devices Not in Services
- Services Without Devices
- Devices Shared Across Services
- Services that Potentially Have Connector Issues
- High Severity Persistent Alarms

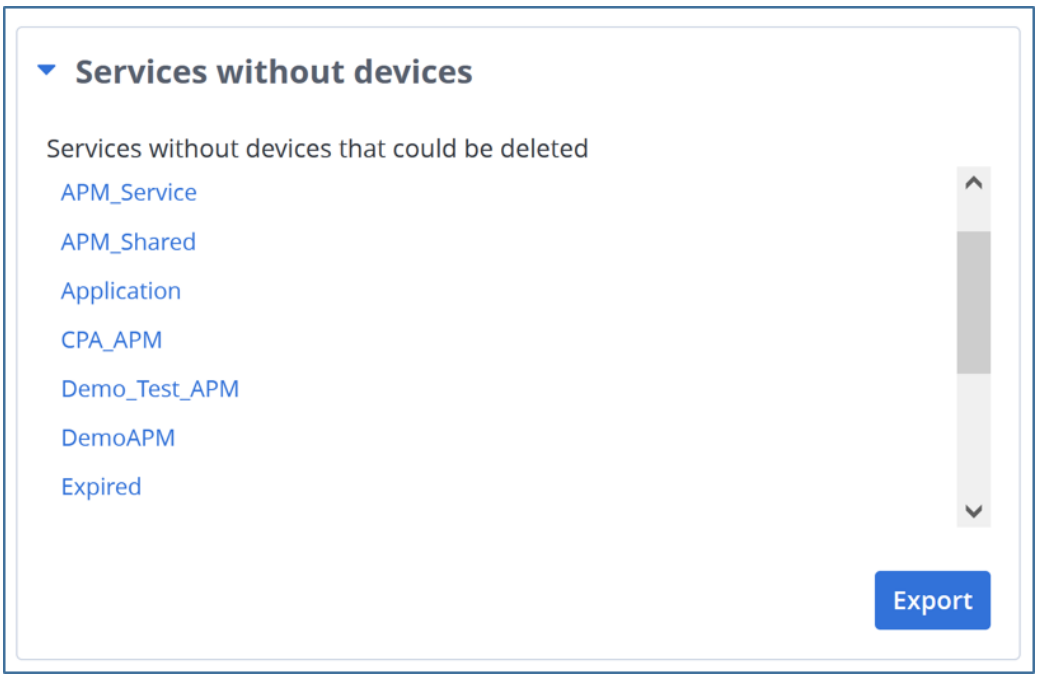
Devices Not in Services

This section displays a list of all the devices that are not in service but look like they should be. In this section, information about Agent, Name, Host, and Type are displayed for the devices. You can export this information to an Excel sheet.



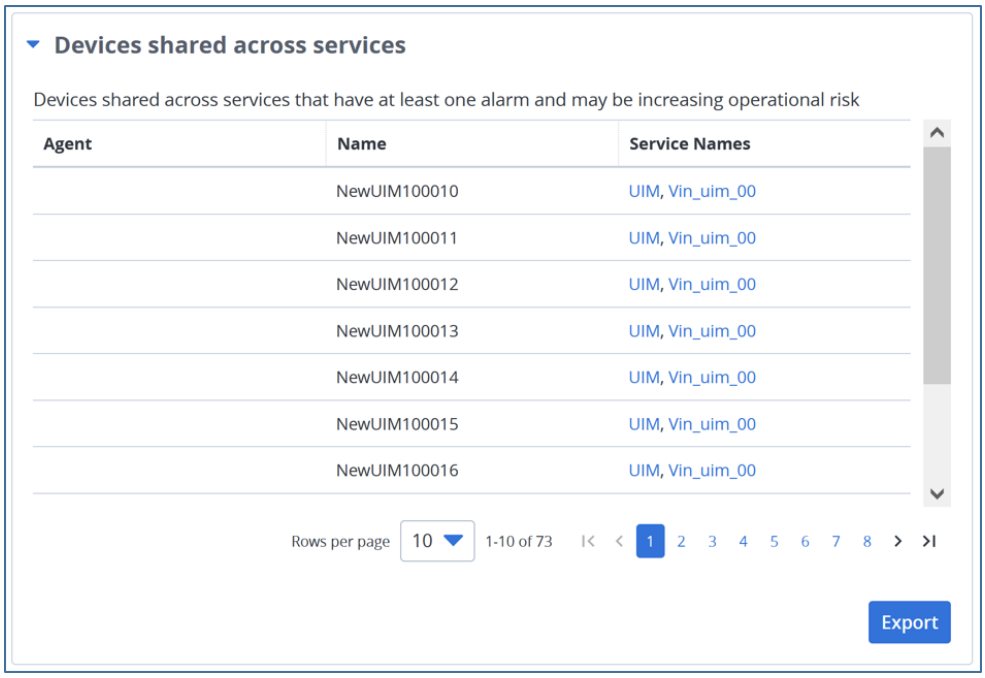
Services Without Devices

This section displays services without devices that can be deleted. Click the service name link to view the service details. You can also export this information to an Excel sheet. The exported information includes the service name and the ID.



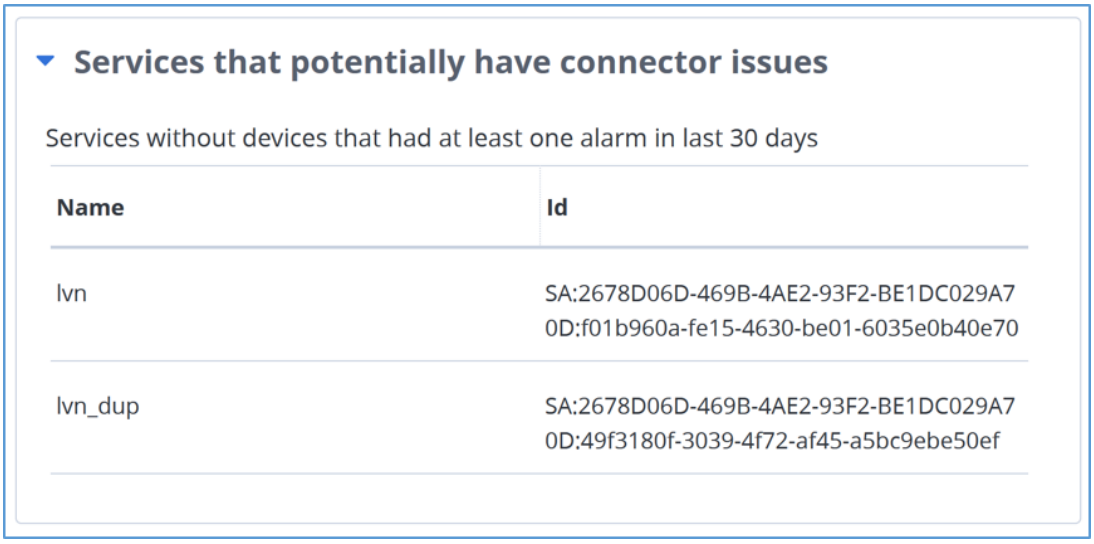
Devices Shared Across Services

This section displays devices that are shared across services that have at least one alarm and may be increasing operational risk. This section provides information about the Agent, Name, and Service Names as shown.



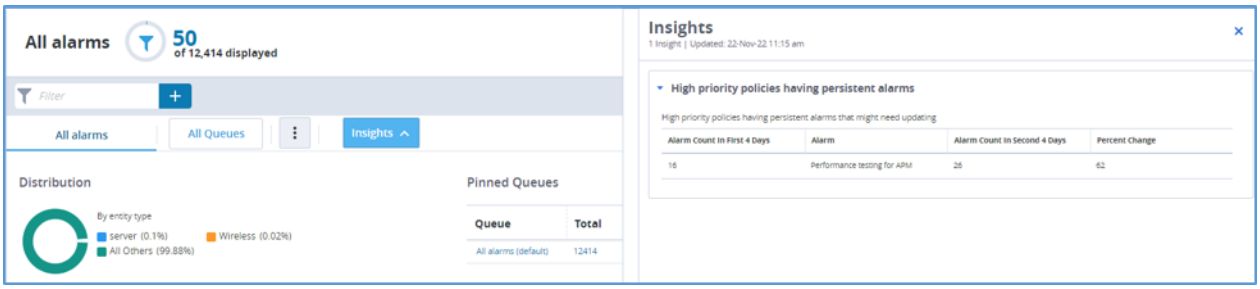
Services that Potentially Have Connector Issues

This section displays services without devices that had at least one alarm in the last 30 days. In this section, the service name and alarm ID are displayed for each of the devices.



High Severity Persistent Alarms Noticing Significant Change

This section displays high-severity persistent alarms that might need updating. Review the thresholds of the configured alarms as they may be contributing to the noise on the alarm console. They are displayed on the **Raw Alarms/All Alarms** page.



Re-enable the Insights

Perform the following steps to re-enable the insights.

Follow these steps:

1. Log in to DX Operational Intelligence and navigate to the **Settings** page.
2. Click **Enable** in the **Configure Insights** tile.
   The application displays the Insights page.
3. Re-enable the required insights.
4. Click **Save.**

Performance Analytics

Learn about Performance Analytics Overview and how it compares multiple metrics and entities for the same devices.

Performance Analytics enables you to track, aggregate, and visualize key performance indicators for any specific time period. Performance Analytics in DX Operational Intelligence allows you to:

- Detect performance bottlenecks and anomalies.
- Compare multiple metrics for the same devices, interfaces, and networks.
- Compare multiple metrics across multiple entities.
- Compare uni-variate metrics across single and multiple entities.
- Compare multivariate metrics across multiple entities.

An **Entity** can be one of the following types:

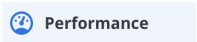
- Services
- Application
- Device

{"URL":["https://digital-oi/performance-analytics"],"description":"concept.dita_5cb05a57-5939-49e8-a499-0020ed366f70","new":"","new_video":"","admin":"","troubleshooting":{"masterkb":"","text":"","URL":["https://knowledge.broadcom.com/external/article?articleId=236621"]},"pendo":"","video":["https://www.youtube.com/watch?v=1TdJAfkPTPA&list=PLynEdQRJawmxJMNgDiLiCxa1lY8mdNGCE&index=14"],"customCards":[{"type":"use","id":"concept.dita_96a5c61b-88ea-4cf8-82bc-5de4015901c9","url":"https://techdocs.broadcom.com/us/en/ca-enterprise-software/it-operations-management/dx-operational-intelligence-saas/SaaS/using/Performance-Analytics.html","title":"Performance Analytics User Interface"}],"customCards":

[{"type":"manage","id":"topic.dita_4797b091-307e-4e15-b662-1faaa470dad4","url":"https://techdocs.broadcom.com/us/en/ca-enterprise-software/it-operations-management/dx-operational-intelligence-saas/SaaS/using/Performance-Analytics/Manage-Views.html","title":"Manage Views"}],"customCards":[{"type":"getStarted","id":"concept.dita_c6c08d69-3bff-4e16-b32b-07493ecaac99","url":"https://techdocs.broadcom.com/us/en/ca-enterprise-software/it-operations-management/dx-operational-intelligence-saas/SaaS/using/Performance-Analytics/Anomaly-Detection.html","title":"Anomaly Detection"}]}

Access Performance Analytics

You can access Performance Analytics in two ways:

1. Log in to DX Operational Intelligence, and click on 

from the left navigation pane
   a) Select the metrics from the Metric Browser pane.
2. To view the performance charts in **context of an Alarm**, perform the following steps:
   a) Click the **Alarms** icon on the navigation panel.

      The **Service alarms** page appears.
   b) Click an alarm from the list of alarms. The row expands and displays the **Alarm Details**, **Affected metrics**, **Impacted services, Topology** and **Annotation**.
   c) Navigate to **Affected metric** tab. The performance chart for the selected metrics appears.
   d) From the **Affected metric** tab, click **Compare Metrics** below the chart. The **Performance Analytics** page is launched in the context of the selected entity and alarm-affected metric.

   The metrics can be ingested into Performance Analytics from the source products listed in Compatibility Matrix. The metric store processes raw metrics along with aggregation and anomaly detection.
3.

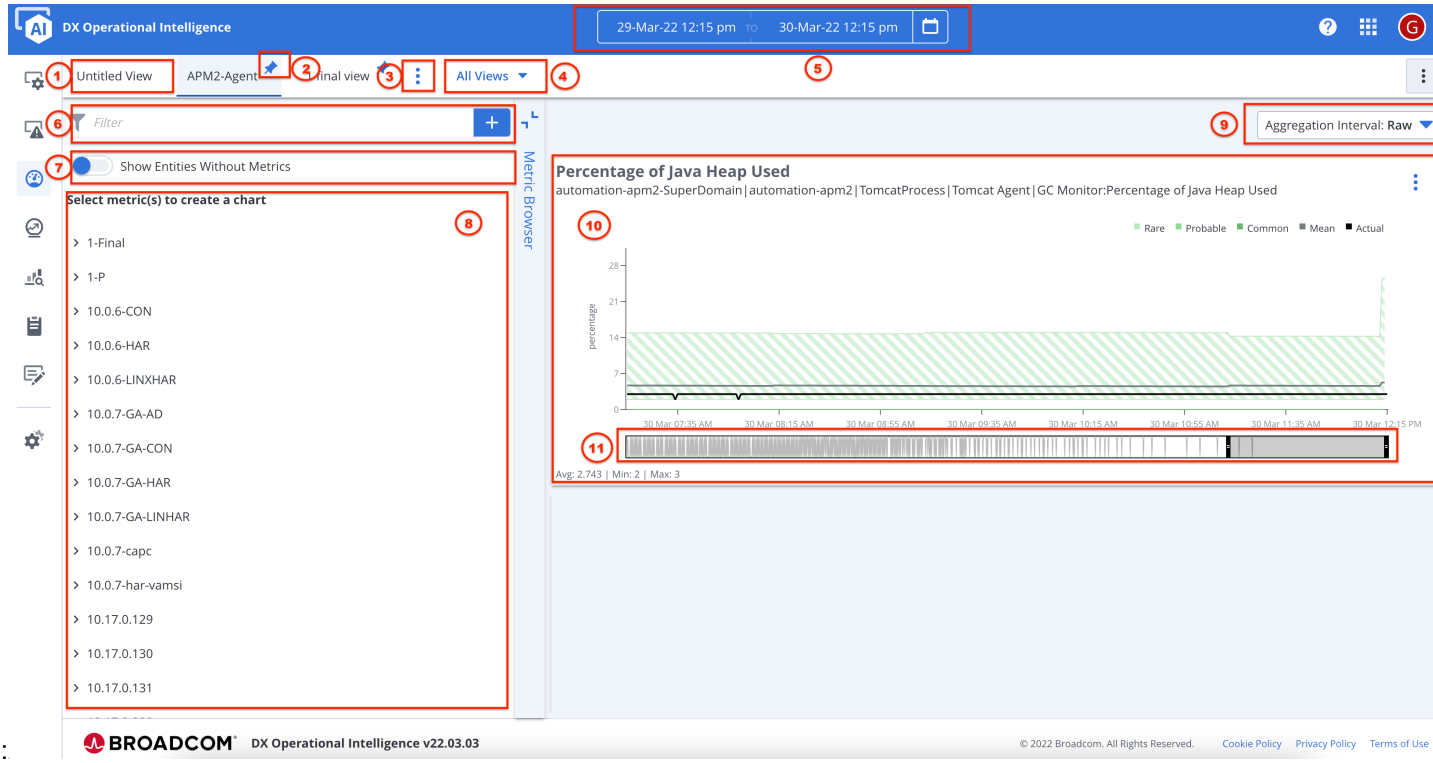
Performance Analytics User Interface

Performance Analytics User Interface allows you to select metrics from the Metric Browser pane to create a corresponding metric chart view.

When you launch the Performance Analytics page from the navigation pane for the first time, by default, you see the untitled view with the list of entities in the **Metric Browser** pane.

The greyed-out entities represent the entities with no metrics for that particular time and date. When you navigate from the Alarm Analytics page in the context of an alarm, the Performance Analytics page opens with the corresponding charts of the alarm.

The following image explains the Performance Analytics Interface:



The Performance Analytics page displays the following charts/sections:

Untitled View (1) By default, you see the untitled view with the list of entities in the **Metric Browser** pane. Click

and save the untitled view after you select the required metrics.

Pinned (2) Displays up to a maximum of five pinned views. Pinned is represented by

icon next to the view name and Unpinned is represented by

icon next to the view name. You can view all the views in the **All View** drop-down list.

Manage Views (

) (3) Use this ellipsis to save a view. Once you select some entities and metrics, you can save your selections. So, when you refer to your Performance Analytics page, you can see the saved view with metrics and metric units. For more information on the list of actions that you can perform on Save View, see Manage Views.

All Views (4) Use this option to view all the pinned views, unpinned views, and last updated views. You can also search for a view by using the **Search** filter.

Date and Time Filter (5) Click the date/time picker

icon to select the duration for which you want to view the performance analytics for the selected entities. The date/time picker also determines whether the entity is active or greyed out. If the entity have any metric active during the selected time period, the entity is expandable. For more information, see Filter by Time Range.

Filter (6) You can use filter attributes to narrow down the entities, metrics, and CIs. Select the required filter attribute and enter the corresponding values to narrow down your search. The metric filter works on the entities which are already available on the Metric Browser screen. You can also use the following metrics filters based on which the entities are enabled or greyed out:

- SourceName
- AttributeName
- DisplayName
- Metric

> **NOTE**
> We recommend reducing the number of entities first by applying entity filters.

Show Entities Without Metrics (7) Enable this toggle switch to show the entities without metrics. By default, you can view all the entities with and without metrics.

Metric Browser (8) The metric browser allows you to select entities, metrics, and configuration items. The greyed-out entities represent the entities with no metrics for that particular time and date.
You can use the **Remove All** option that is on the right corner to clear all the selected metrics. This option is enabled only when the metrics are selected in the metric browser.

Aggregation Interval (9) Use the aggregation interval drop-down to specify the interval at which data aggregation takes place. If you select **Raw** option from the drop-down, the metric charts are plotted and you cannot synchronize the metric charts. By default, **Raw** option is selected. If you select a time range within three days, then the charts are plotted on raw metrics. If you select a time range beyond three days, then the charts are plotted on 15-minute aggregates and daily aggregates.

Chart Panel (10) Selecting the metric creates a chart in the Charts panel. If only a single CI is selected from the metric type, a chart with probability bands is plotted. If you select more than one CI or multiple aggregation types are set to the entity, a line series chart is plotted. The following chart view appears based on the available data:

- Timeline Series Probability Bands
- Timeline Series (Line) Charts

Time Slider (11) The time slider appears below the chart when the space is not sufficient in the chart and the need to scroll.

Select the Metrics

1. From the **Metric Browser** pane, click the enity.

   The list of available Metrics and Configuration Items (CIs) for the selected entity appears. The hierarchy that builds up depends on the selected entity. If the selected entity has more CIs associated with it, then the hierarchy builds up in such a way that you view the list of CIs and the associated metrics of the CIs.

2. Select an **Entity** and click the drop-down arrow to view the associated **Configuration Items**.

   The list of associated CIs appears.

3. Click the drop-down arrow to view the associated **Metrics Family.**
4. Click the drop-down arrow to view the associated **Metrics** of the CIs.You can also click **Filter** at the parent level to view the associated child entities.

   NOTE
   While working on the Application context of APM sources, the APM entities that you select might not display the associated hierarchies. You must select the corresponding APM agent, then you can see the exact hierarchies and metrics.

5. Select the metric to view the performance chart.

   The performance chart appears.

Metric Charts Views

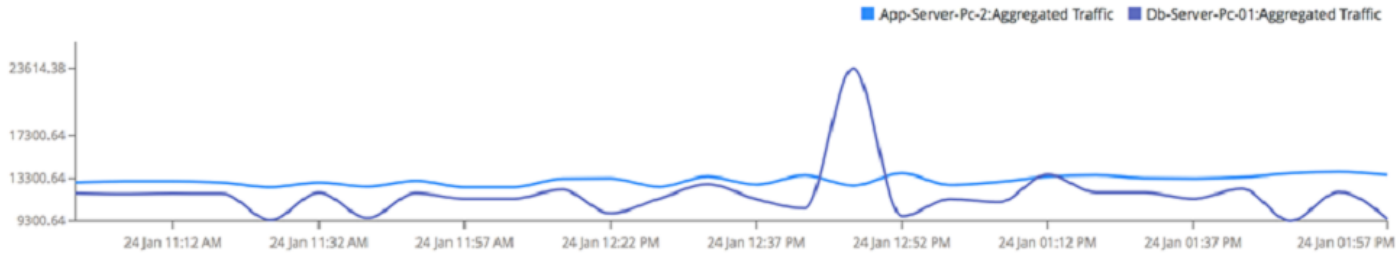
This section describes the following chart views that are displayed based on data available:

- Timeline Series Probability Bands
- Timeline Series (Line) Charts

The **Timeline Series Probability Bands** are shown when there is a single metric selected, and for that metric anomaly detection is also configured. The **Timeline Series (Line) Charts** are shown when there is more than one entity being plotted on the same chart or the selected metric does not have anomaly detection configured.

Timeline Chart

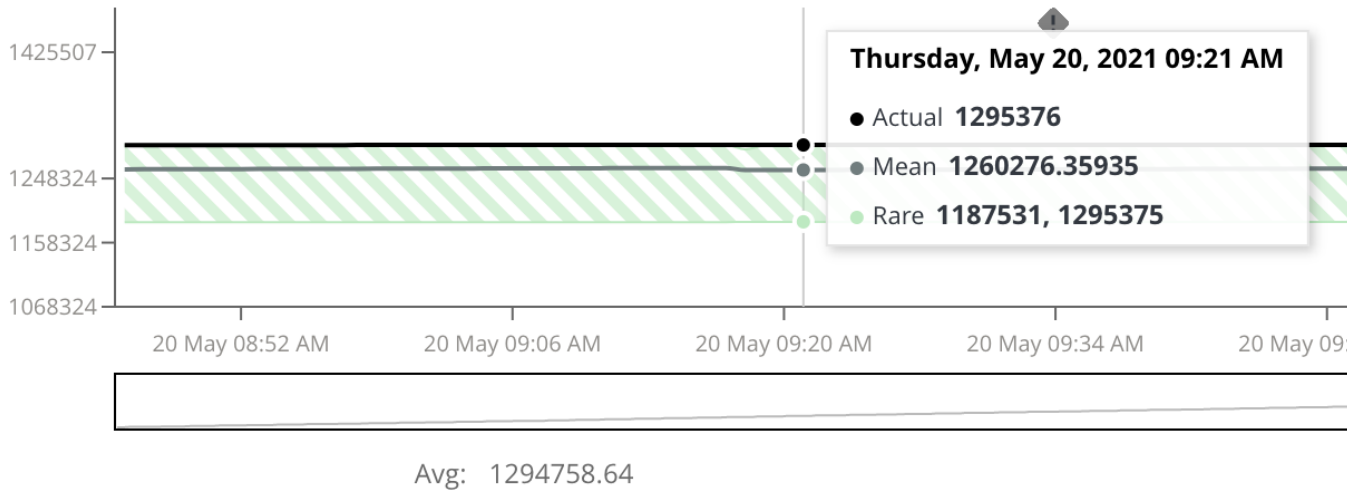
If you have selected more than one Configuration Item or aggregation type as an entity, a timeline series chart is plotted. Click the legend to view the respective series in the chart.



Probability Band Charts: Interpret the Green Highway Band

If only a single Configuration Item (CI) is selected from the metric type family, a chart with probability bands is plotted based on the duration and frequency of the data. When data for the metric is too large, a time slider is available below the chart. Use the time slider to select a time range and view details for a selected period. Three zones display the different prediction intervals present in the historical data. Scan the chart to identify points in time where an alarm was triggered, and where analytic data falls outside of the

established norm for a component on your system. For more information on anomaly detection, see Anomaly

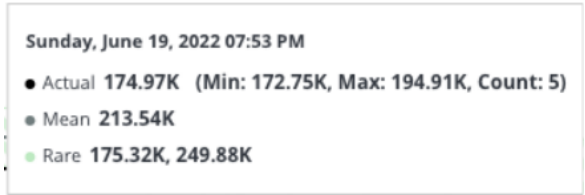


Detection.

Components of Charts

The chart combines information that is designed to help you quickly review and interpret activity for a metric in your environment. The various components of the chart are as follows:

- **Actual Data:**Actual data is illustrated as a black solid line (no alarms) or a colored solid line (alarm state) on the chart. The line is calculated from actual data for the metric. By comparing actual data and the median value, you can quickly see variations for the metric in your environment. When an alert occurs, the black line changes to a colored solid line that matches the alert severity and displays icons with the matching severity. For example, red for critical anomaly alarm, orange for major anomaly alarm, and yellow for minor anomaly alarm
- **Mean Value for the Metric:** The mean value or average for your metric is illustrated as a gray line on the chart. The mean value is displayed when there is not enough historical data.
- **Rare Data:**Rare zone data is illustrated as a light green band on the chart. The analytic place the rare zone three percentile points above or below the norm, and signal a metric behavior outside the normal range.
- You can also view the chart data for an instant when you move your mouse over the chart to display dynamic information in a popup. For example, when you pan over the zones, each data point in time is summarized in a popup:



Manage Metric Chart Views

This section gives you an overview of how you can manage your chart views on the Chart Panel.

Anomaly Detection

You can enable or disable anomaly detection for a particular chart view by clicking

.

The Anomaly detection switch appears only for the metrics that support Anomaly detection. If you disable the **Anomaly Detection** switch, the bands do not appear on the chart from the time the anomaly detection is disabled.

Show Metrics

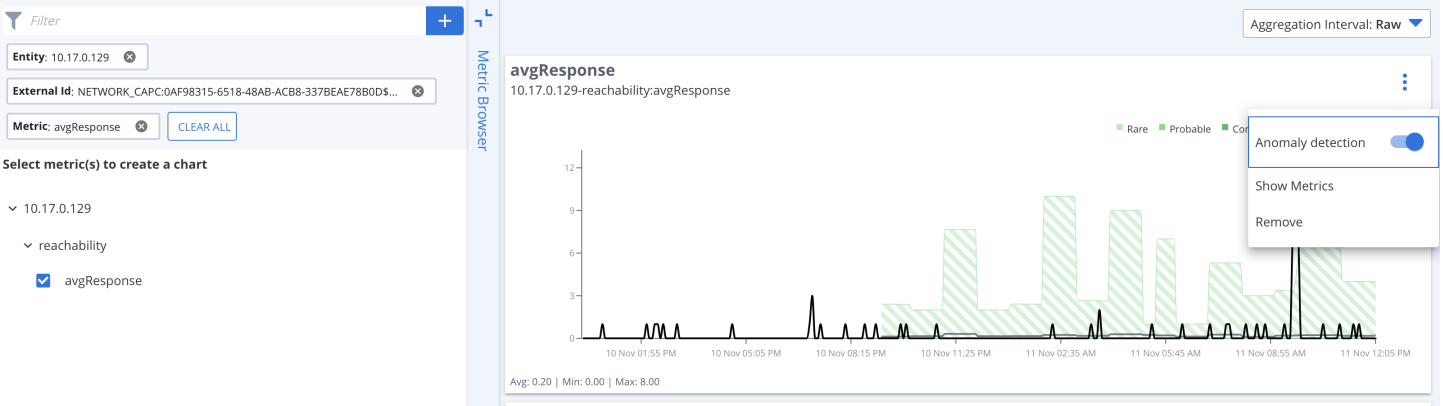
To view the metrics for a chart, perform the following steps:

1. On the Chart view, click

.

2. Click the **Show Metrics**.

The metrics associated with the chart appear on the Metrics Browser.



Remove Chart View

To remove a chart view, follow these steps:

1. On the Chart view, click

.

2. Select **Remove**.

The chart is removed from the chart panel for the view. You can then save that particular view.

Synchronize Metric Charts

You can synchronize metric charts and view the synchronized timeline for a selected metric chart or all charts.

Follow these steps:

1. Select the required metrics from the Metric Browser.

   The metric charts appear.

2. To synchronize a chart, select an **Aggregation Interval.** The metric chart is synchronized according to the specified aggregation level.

   NOTE
   Synchronization is possible only on time-based interval data, and not on raw data. By default, charts are synced and the default time interval is 10 minutes.

Filter by Time Range

Click the date/time picker

icon to select the duration for which you want to view the performance analytics for the selected entities. By default, the duration is one week.

When you are accessing the Performance Analytics page from the navigation panel, then the default duration is one week.When you access the performance analytics page in the context of an affected alarm metric, then the default duration is calculated by (-8 hours + 1 hour) from the alarm creation time. The calendar is updated to match the last updated time of the affected alarm. This range is 8 hours before the last alarm update to one hour after the last alarm update.

You can use the date/time picker filter to set the duration for your search, that was created for a specific time period such as 1 hour, 24 hours, 1 week, and so on. Performance Analytics also provides you with a **Custom** option, which enables you to pick a particular start date/time and a particular end date/time to help you narrow down the metrics that you are looking for.

NOTE
If you select a time range within 3 days, then the charts are plotted on raw metrics.

Troubleshoot Anomaly Detection

This video walks you through the troubleshooting steps for some of the issues:

Manage Views

This section gives you an overview of how you can manage your views on the Performance Analytics page.

Save View Actions

Once you select some entities and metrics, you can save your selections as a view. The units of the metrics also get stored in the saved views. You can perform the following actions in **Save View**:

- Create a View
- Edit a View
- Delete a View

Create a View

To create a new view, perform the following steps:

1. From the Performance Analytics page, select the required metrics from the **Metric Browser.**

   NOTE
   By default, you are on the**untitled view** tab.

   The metric chart view appears on the Chart panel.

2. Click

   ,

   Save.

3. Enter the view name.
4. (Optional) Click **Pin View** to pin the view. This allows you to view the saved view with metrics and metric units when you refer to the Performance Analytics page.
5. Click **Save**.

A view is created.

Edit a View

To edit an existing view, perform the following steps:

1. From the Performance Analytics page, select the required metrics from the **Metric Browser.**
2. Click

   ,

   Save

   NOTE
   You can also rename the view or save it as a new view.

3. (Optional) Enter the view name.
4. (Optional) Click **Pin View** to pin the view. This allows you to view the saved view with metrics and metric units when you refer to the Performance Analytics page.
5. Click **Save**.

The view is edited.

Delete a View

To delete an existing view, select the **View** tab, click

and**Delete**.

Anomaly Detection

Anomaly detection provides a way to identify events that deviates from the normal behavior of metrics, established based on their previous data. Anomalies or anomalous events indicate critical incidents, such as a spike in a computational resource consumption of CPU, memory, storage, network. Anomalous events can also identify the sudden increase in Java Heap or a drop in the number of user sessions, thus pointing out a glitch in the system in a proactive manner. This can be achieved with the help of a machine learning algorithm.

DX Operational Intelligence provides the tools to identify anomalous events across the network, infrastructure, and application metrics and trigger an alarm for the anomalies identified.

For more information about how to configure anomaly detection, see Configure Monitoring.

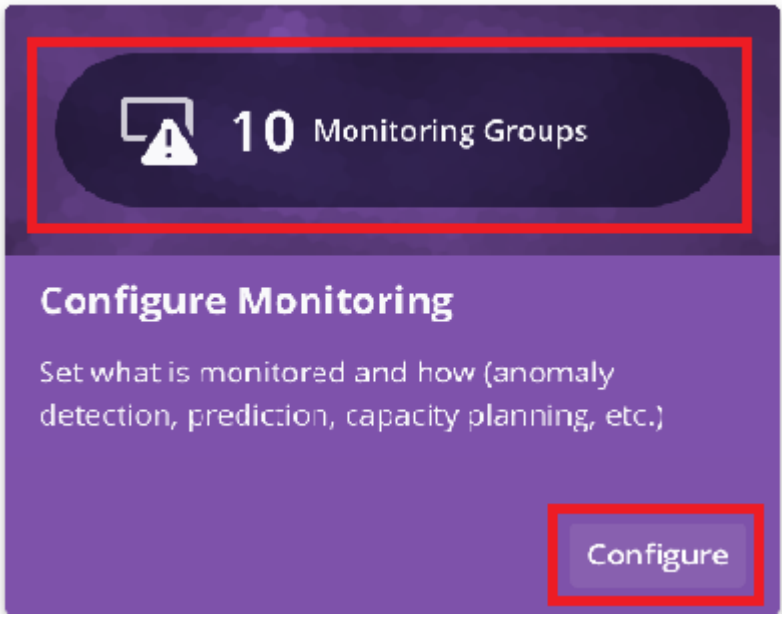
The following video explains the improvements in Anomaly Detection.

Configure Monitoring

Use the metrics groups to control and monitor anomaly detection on groups.

Configure monitoring let you create metric groups which can then be further used to configure Anomaly Detection.

Use **Configure Monitoring** tile to manage the life cycle of the metric monitoring groups and enable anomaly detection on these metric groups. Click **Configure** to configure the metric monitoring groups. You can also view the count of the configured metrics groups on the tile.



Overview

DX Operational Intelligence provides the functionality to configure the metric groups for which you will be able to enable anomaly detection from the same user interface. You can use the Metric Groups to control and monitor anomaly detection on certain metric groups. You are provided with the ability to create specific metric groups and enable anomaly detection for the specific metrics that you want to focus on, instead of having anomaly detection enabled for all the metrics.

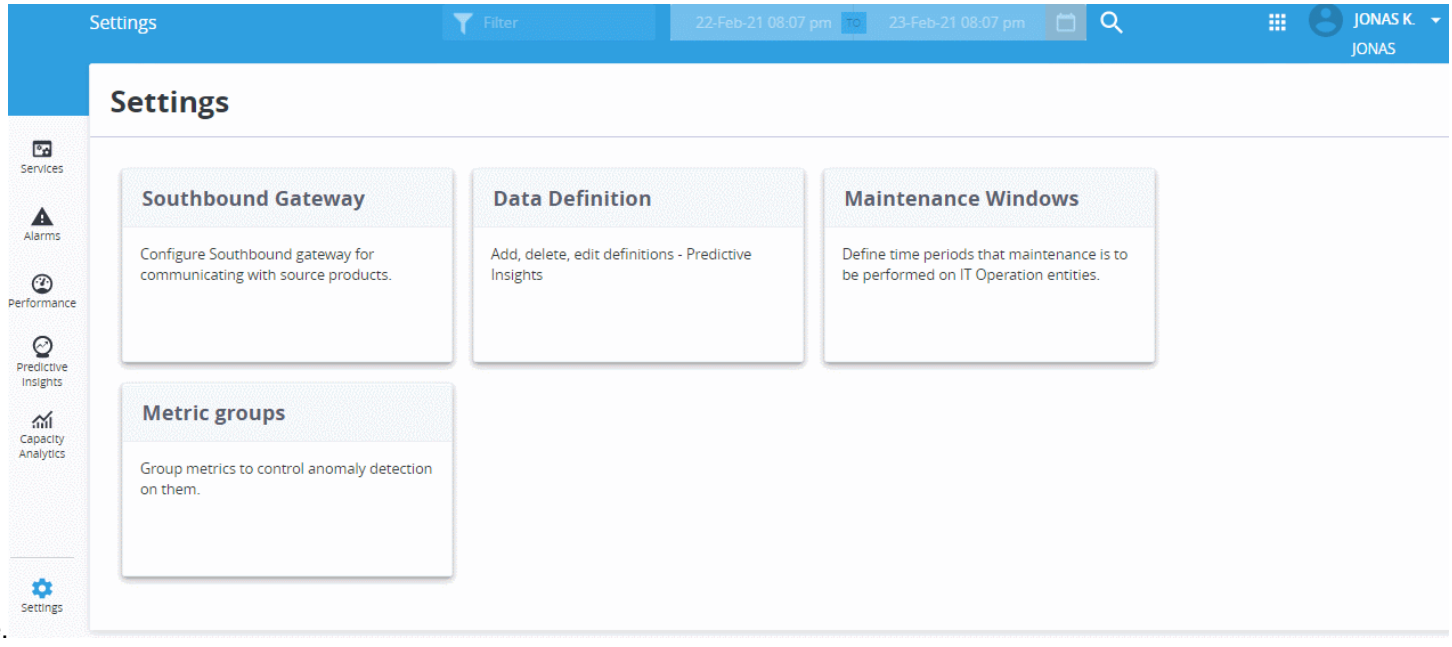
The Metric Monitoring Groups view displays the total available metrics for a tenant and the total consumed metrics of a tenant. The total predefined metric limits for anomaly detection are set at 50000 metrics for a tenant and 5000 metrics for an individual metric group. You must ensure that the total metrics enabled for anomaly detection at any given time are within 50000 metrics for a tenant and 5000 metrics for a metric group.

If you exceed the predefined metric limits for the metric group you would not be able to enable anomaly detection for that particular metric group. If you still want to create more metric groups for anomaly detection, then you must disable certain existing metric groups and bring them within the predefined metric limit. In a scenario where you have exceeded the defined metric limit for a tenant, an alarm is raised about the limit breach automatically in the Alarm Analytics view. If you do not take any action on this alarm within the expected time, then DX Operational Intelligence would disable one or more groups automatically to match the limits. Therefore, the recommendation is to manually modify the metric groups to fit into the preset limits.

> **NOTE**
> An alarm is raised only when you breach the total metric limit set for a tenant and not when you breach the total metric limit set for an individual metric group. For more information, see Anomaly Alarms and Anomaly Detection.

Access Metric Groups

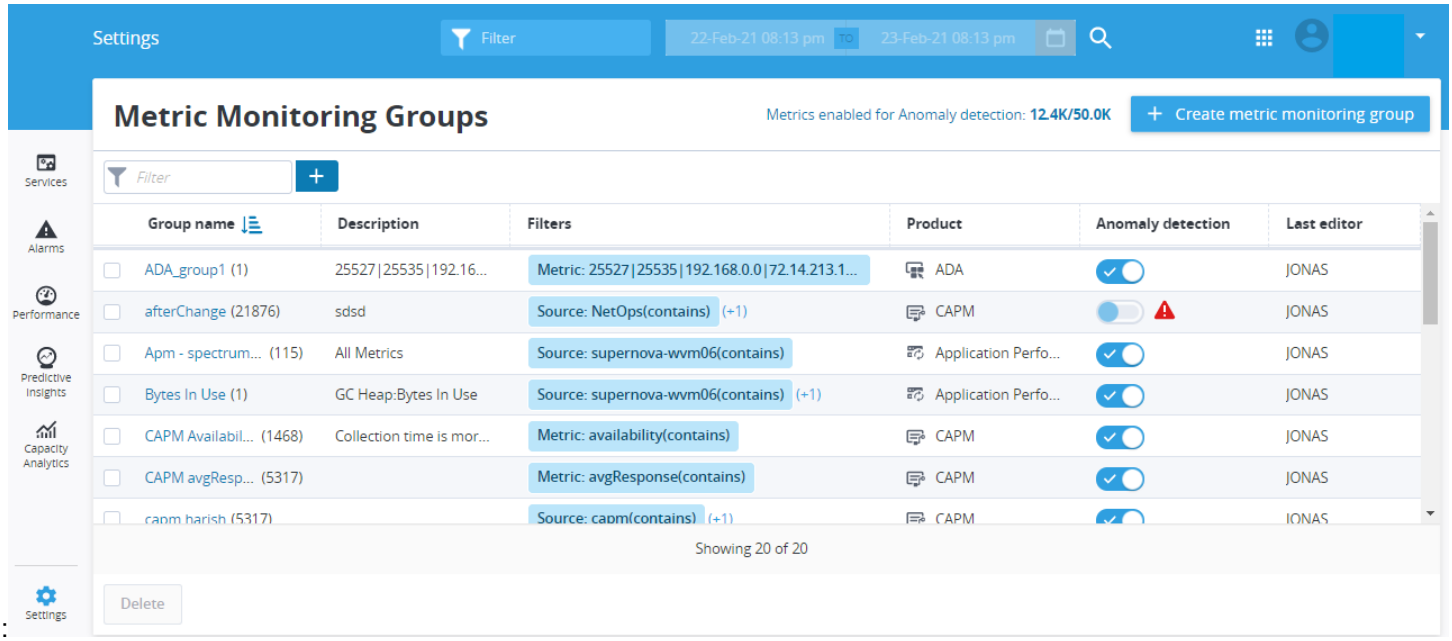
To view the Metric Groups, log into DX Operational Intelligence, and click on **Settings** icon from the left navigation pane.



Metric Monitoring Groups User Interface

The Metric Monitoring Group view displays the list of metric group names, the total number of metrics in each metric group, and that the list of metric groups that have

been enabled for anomaly detection. This User Interface displays the following details:



Column Name	Description
Column / Row-level Action	Enables you to perform row-level metric group actions or column level to perform bulk metric group actions.
Group Name	Indicates the list of metric group names. The metric group name also gives you an insight by displaying the total metrics of each group with the value in brackets. For example, UIM_test_23 (2956)
Description	Indicates the description details of each metric group.
Filters	Indicates the filters that were selected while creating the metric group. This column displays only the first selected filter for each group name, all the remaining filters are displayed as a value in brackets. If more than one filter was selected while creating a group, then you must click the number to view the remaining filters. For example, say +1.
Product	Displays the source product of each metric group.
Anomaly Detection	Indicates if the metric group is enabled or disabled for anomaly detection monitoring.
Last Editor	Indicates the name of the person who last modified the metric group.

Filters

You can filter alarms by using the filter available in the Metric Monitoring Groups view. Metric Groups support the following filters:

- **Global Search Filter**
- **Filter by Attributes**

Global Search Filters

This filter

lets you search for metric groups in the Metric Monitoring Groups table. Enter your search text to view the metric groups that match your search text. The search takes effect in all the columns in the Metric Monitoring Groups table, except for the Anomaly detection column.

Filter by Attributes

You can filter metric groups by attributes using the **Attributes Filter**

. Enter your search text to view metric groups that match your search text or use the attributes list to narrow down your search results. The search takes effect in all the columns in the Metric Monitoring Group view. **Follow these steps:**

1. Click the plus icon next to the **Metric Attributes** filter field.
2. Select the required filter attribute(s) from the list of attributes
3. Select or enter the value for the selected attribute(s).
4. Click **Add** to add attributes to the **Metric Attributes** filter. For example, if you want to see all the metric groups that have Anomaly Detection enabled, select the attribute as **Anomaly detection** and select its value as **Enabled**. The Metric Monitoring Groups table displays only the metric groups that match your search criteria for the selected attributes.

Sorting Columns

The Metric Monitoring Groups view supports sorting for the following columns:

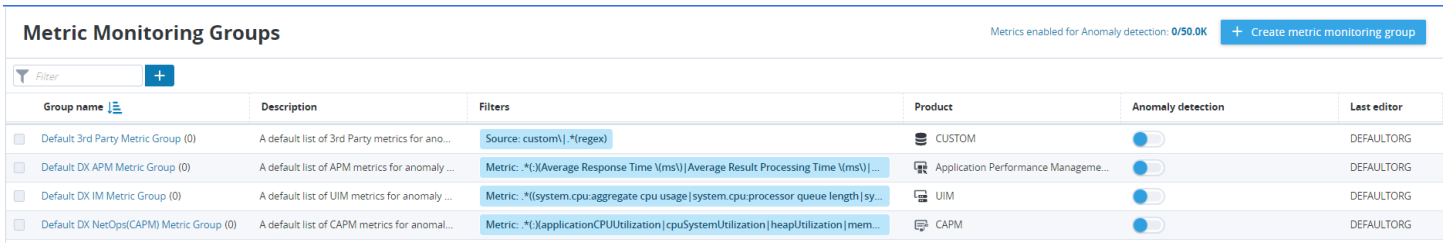
- Group name
- Description
- Product
- Last editor

Managing Metric Groups for Anomaly Detection

Default Metric Groups

Metric Monitoring Groups provide you with the following out-of-the-box metric groups when a new tenant is created. By default, these default metric groups are Read-only and disabled for monitoring. You must enable them if you want to monitor these default metric groups. The default metric groups are as follows:

- Default 3rd Party Metric Group
- Default DX APM Metric Group
- Default DX IM Metric Group
- Default DX NetOps(CAPM) Metric Group

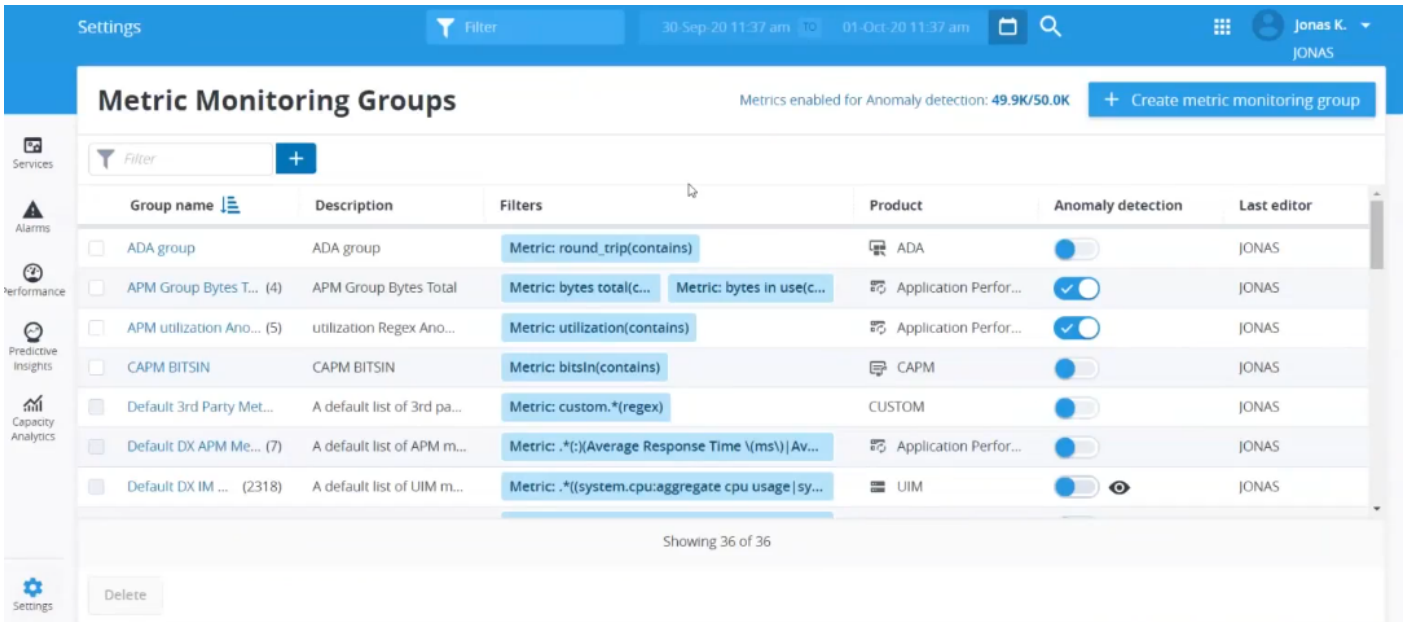


Create Metric Groups for Anomaly Detection

To monitor metric groups for anomaly detection, you must create and define the metric groups. To create a metric group, perform the following steps:

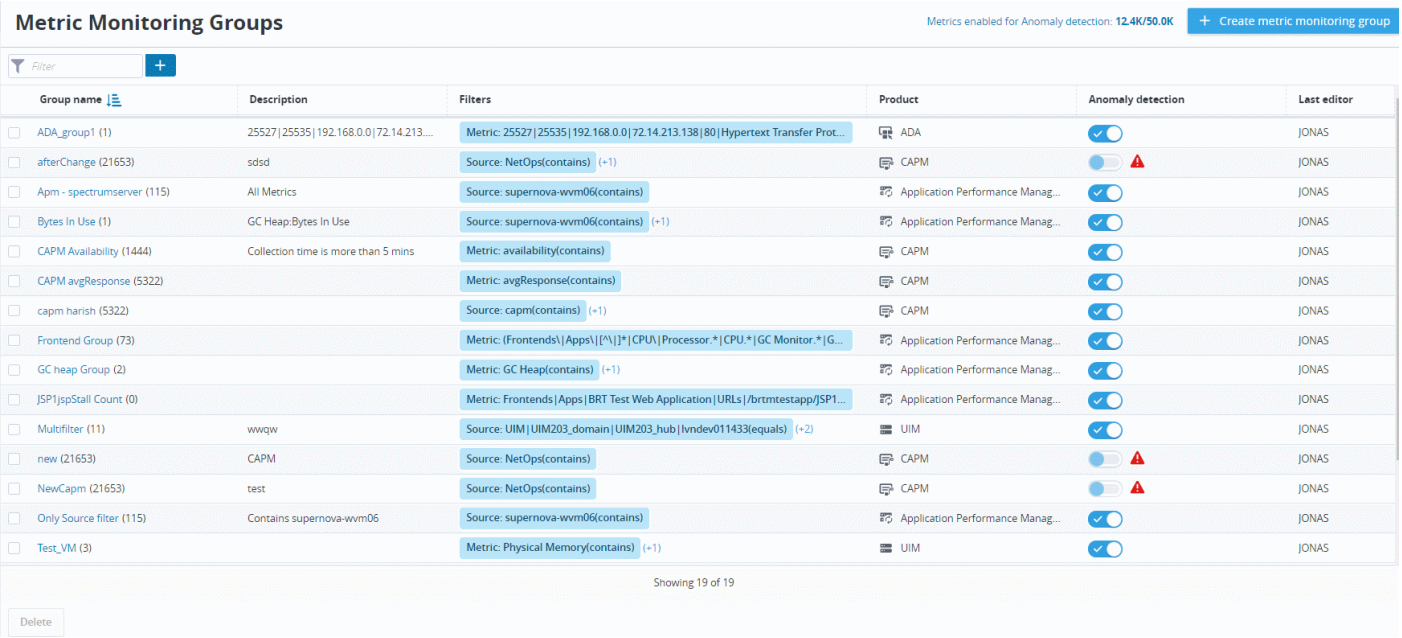
1. Log in to DX Operational Intelligence, and click on **Settings**

   icon from the left navigation pane.
2. Click on **Metric Groups**.
   The **Metric Monitoring Groups** view appears.



3. To create a metric monitoring group, click **Create Metric Monitoring Group** option.
4. Click the **Edit**

   option
   and provide a meaningful name for the metric group. Alternatively, you can provide the metric group name while saving the metric group as well.
5. Select a **Product name** from the drop-down option.
   You can see the list of available metrics for the selected product.



6. You can use the **Filter** option to drill down your search for the required metrics. You can use this filter especially when there are many metrics for a particular source product. Enter your search text to view metrics that match your search text or use the attributes list to narrow down your search results. This filter allows you to view only those metrics with attributes matching your search criteria.

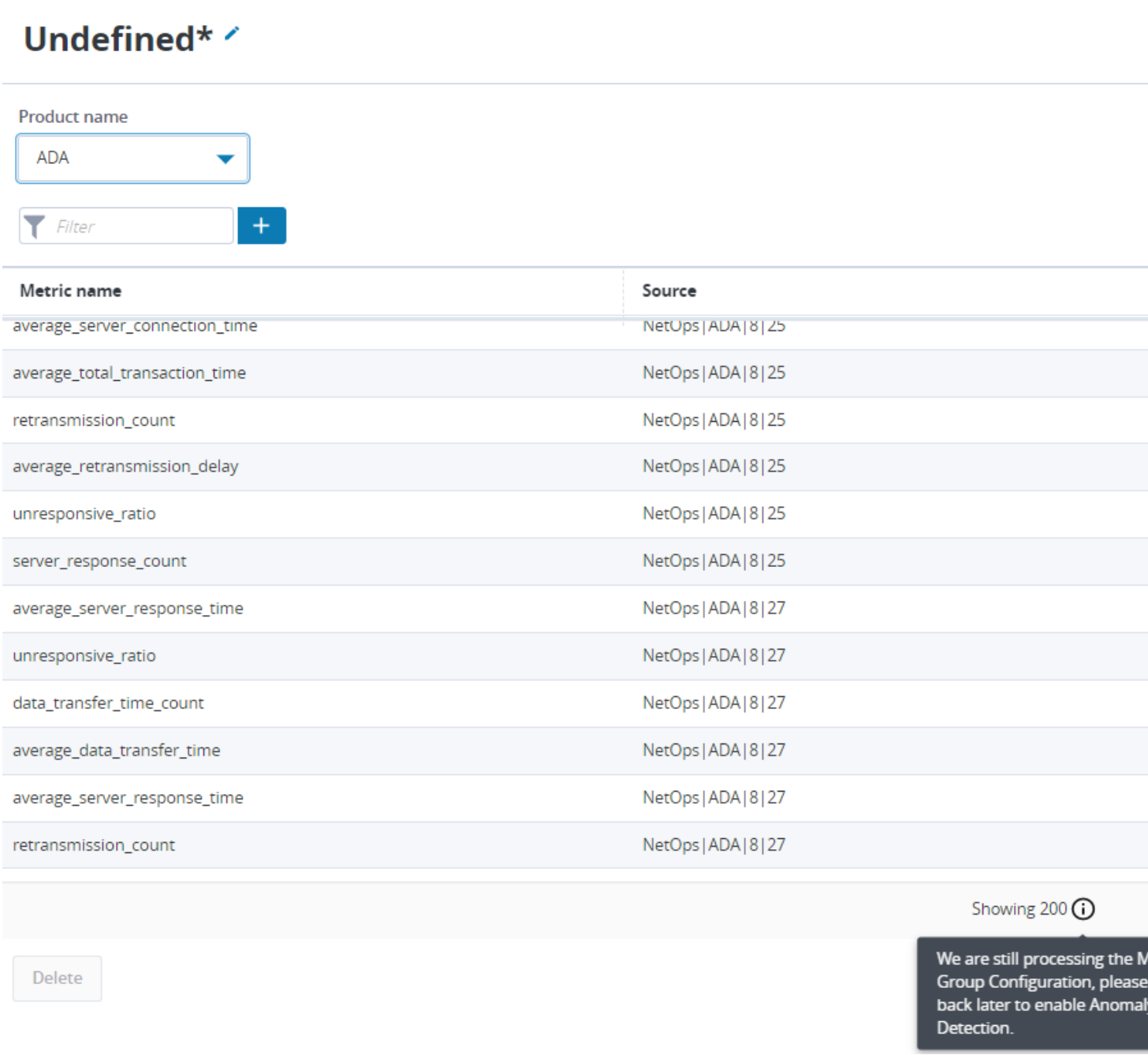
Follow these steps:

a. Click icon in the **Metrics** attribute filter field.

b. Select a filter attribute with one or more operators/options, and enter the respective value for the selected attribute.

c. Click **Add** to add attributes to the **Metrics** filter. For example, if you want to see all the metrics from the DX NetOps product, select the filter attribute as **Source**, the operator as **Contains,** and its value as **DX NetOps**. The Metric Groups table displays the metrics that match your search criteria for the selected attributes.

   The list of metrics that match the filter criteria gets displayed.

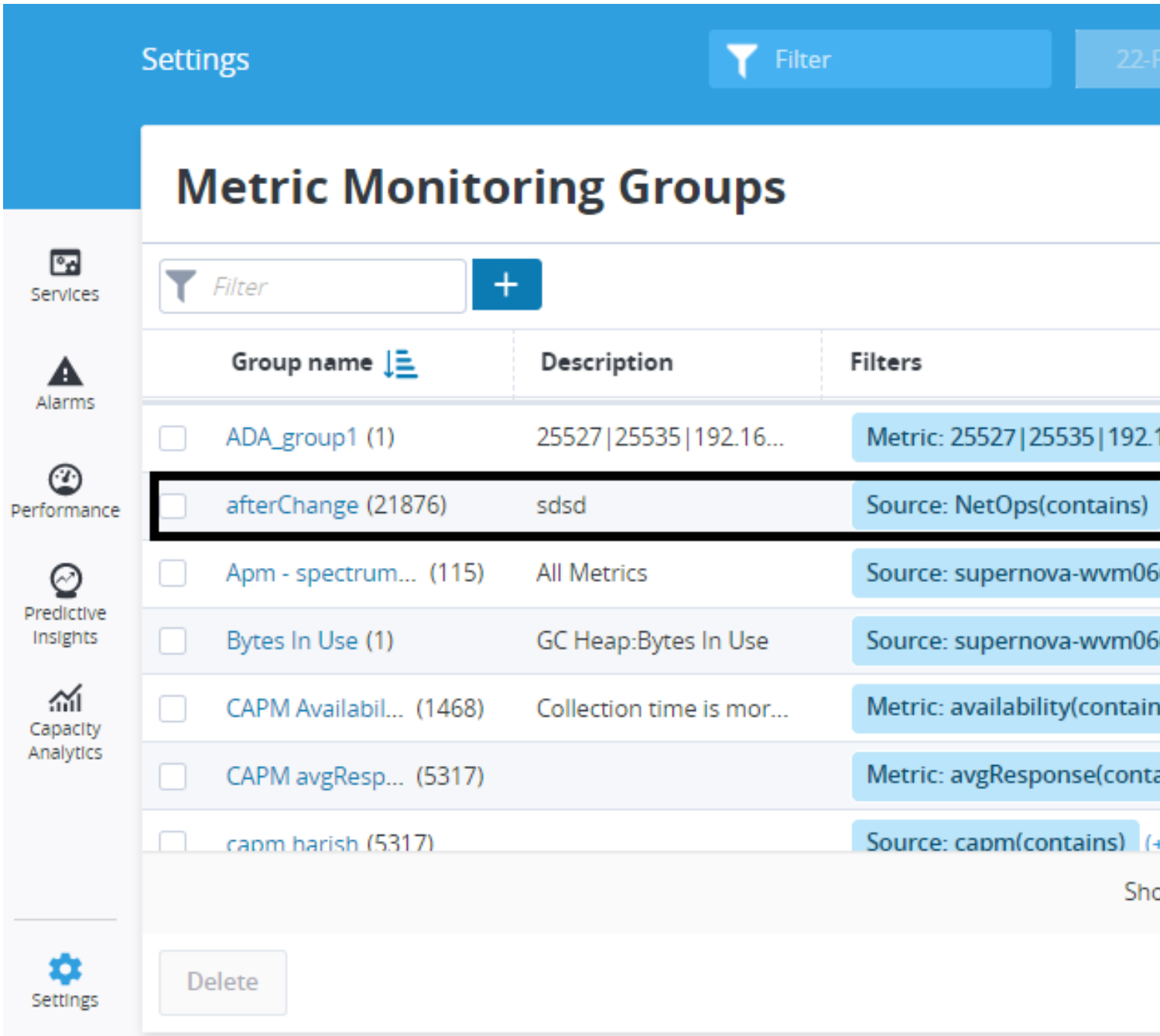


ATTENTION
In a scenario where there are too many metrics that are still being calculated, the Metric Groups display the ⓘ icon next to the metrics count. You can keep scrolling down to see more metrics getting populated, but you must wait or come back later to enable Anomaly Detection as the metrics are still being calculated. When the metric count calculation is complete, you see the total count displayed without the ⓘ icon next to the metrics count.

7. If you are creating a metric group within the defined limits, the metric group gets enabled automatically for **Anomaly Detection**. You can view this toggle option at the top-right of the view. If you have exceeded the defined limits, then the metric group gets disabled automatically. But, you can still save the metric group.

   ATTENTION

   - The total predefined metric limits for anomaly detection are set at 50000 metrics for a tenant and 5000 metrics for an individual metric group. You must ensure that the total metrics enabled for anomaly detection at any given point of time are within 50000 metrics for a tenant and 5000 metrics for a metric group.
   - If you have exceeded the limit of 5000 metrics for a metric group, then the Anomaly Detection option gets disabled automatically. The Anomaly Detection information button displays a message that you can't enable this option as it would breach the metrics limit of the tenant.
8. To save the metric group, click the **Edit** option and perform the following steps:
   a. Enter a meaningful name in the **Metric Group Name** field.
   b. (Optional) Enter the required description for the metric group in the **Description** field.
   c. Click **Next**, **Save**.
   d. To configure the time interval for the anomaly alerts and trigger conditions for the same anomaly alarms, click Configure Alarms.
9. You can see the updated list of Metric Groups. If any metric group has breached the defined metric limit, the **Anomaly Detection** toggle option gets disabled and appears as depicted below:



Configure Alarms

The Configure Alarms UI allows you to configure the time interval for the anomaly alarms and trigger conditions for the same anomaly alarms. This approach does not generate alarm storms for multiple occurrences of similar anomaly. You can perform the following actions:

- Configure the number of anomaly occurrences or the time interval for which the anomaly has to persist to trigger an alarm.
- Set the anomaly alarm trigger conditions.
- Configure the alarm message and the notification channel.

For example, A service owner is monitoring the java web services, configures the trigger condition for an alarm. Based on the trigger condition, the alarm is triggered when there is an anomaly in the java heap size that is recorded over 20 mins interval continuously.

To configure alarms, follow these steps:

1. From the **Create Metric Monitoring Group** page, click **Configure Alarms**.
2. Provide the following details:
   a. **Enable Detection:** Select this option when you want to enable anomaly detection for the anomaly alarms.
   b. **Use System Settings:** Select this option when you want to configure the alarms using the default settings. All the fields are greyed out and you can only configure the alarm notification policy.
   c. **Alarm when anomaly is (Above threshold and Below threshold):** Select the threshold option (Above threshold or Below threshold) for the anomaly alarm to generate.
   d. **Alarm if anomaly:**
      a. **Count over threshold is:** The threshold was breached X out of the last Y intervals. For example, the threshold was breached 4 out of the last 6 polled intervals.
      b. **Time over threshold is:** Within a given time window, the threshold was breached X minutes. The following example shows the time window for the anomaly alarms suppression for incoming products.

Anomaly Alarms Suppression Time Interval (in Mins)	Anomaly Alarms Time Window (in Mins)
5	5 x 12 = 60
1	1 x 12 = 1

   e. **Alert if deviation:** Select this option to reduce noise by configuring the percentage deviations or static threshold values. This configuration triggers an alarm for any minor deviation in the metric data from the band identified by the algorithm based on the historical usage. The deviation configuration works in conjunction with the existing Anomaly Alarm configurations.
      a. **Percent is greater than:** Specifies the percentage value for the deviation.
      b. **Value is:** Specifies the metric value.
   f. **Alarm Message:** Use this field to customize the alarm message. You can and use the following attributes in the alarm message.
      a. metric_name: Specifies the full name of the metric including the source and attribute name.
      b. host: Specifies the host of the metric.

      NOTE
      To get the list of supported attributes, enter ***${*** in the alarm message field.
   g. **Select Alarm Notification Policy:** You can choose the alarm policy from the existing policy list or you can create a new policy.
3. Click **Done.**

Configure Alarms

Setup Anomaly Alarms

- [x] Enable Detection

- [ ] Use System Settings

Alarm

Alarm when anomaly is

- [x] Above threshold
- [ ] Below threshold

Alarm if anomaly

Count over threshold is | 5 | in | 6 | occurrences

Alarm if deviation

Percent is > | | Value is | | |

Alarm Message

```
${metric_name} has breached threshold for
${countOverThreshold} out of ${observationCount} times for
${host}
```

Select Alarm Notification Policy

Choose an Alarm Notification Policy...

Cant find? Create New

View and Modify a Metric Group

The Metric Monitoring Groups view displays the list of Metric Groups and their respective details. You can view the Anomaly Detection column to view the metric groups that have been enabled for Anomaly Detection. If any metric group has breached the defined metric limits, then the **Anomaly Detection** toggle option gets disabled and displays the

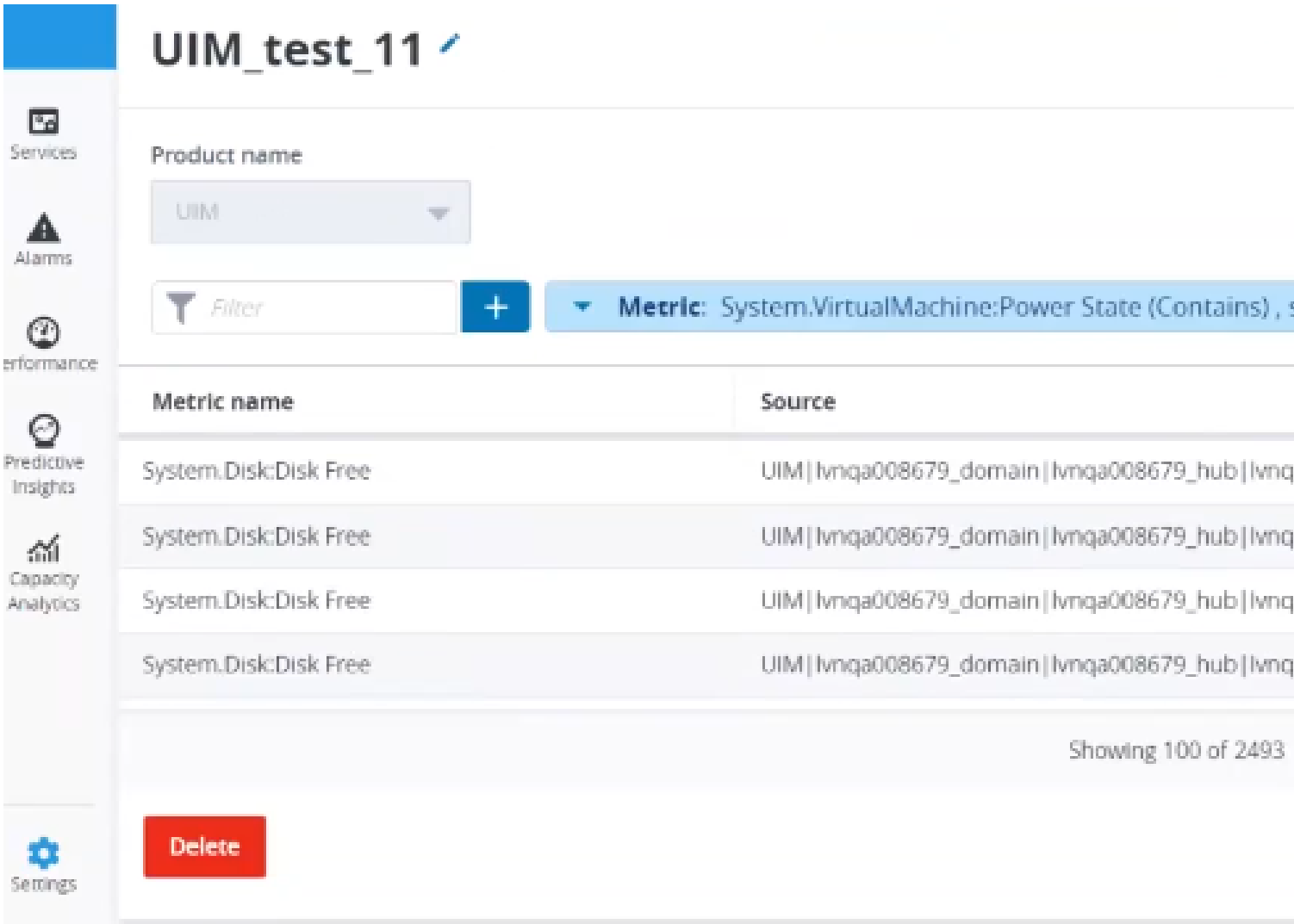
icon. You need to go to the particular metric group that has breached the limit and modify the metrics to fit into the preset limits or you need to disable the existing metric group(s) to accommodate this particular group provided it's within the defined limits. You can modify all the details in an existing metric group, except for the **Product name**, which is disabled. To view and modify an existing metric group, perform the following steps:

1. In the Metric Monitoring Groups view, click on a **Metric Group name**.
2. You can rename the Metric Group name if required.
3. You can modify the existing metrics using the metric attribute filter options.
4. Click **Save**.
   You have successfully modified the Metric Group.

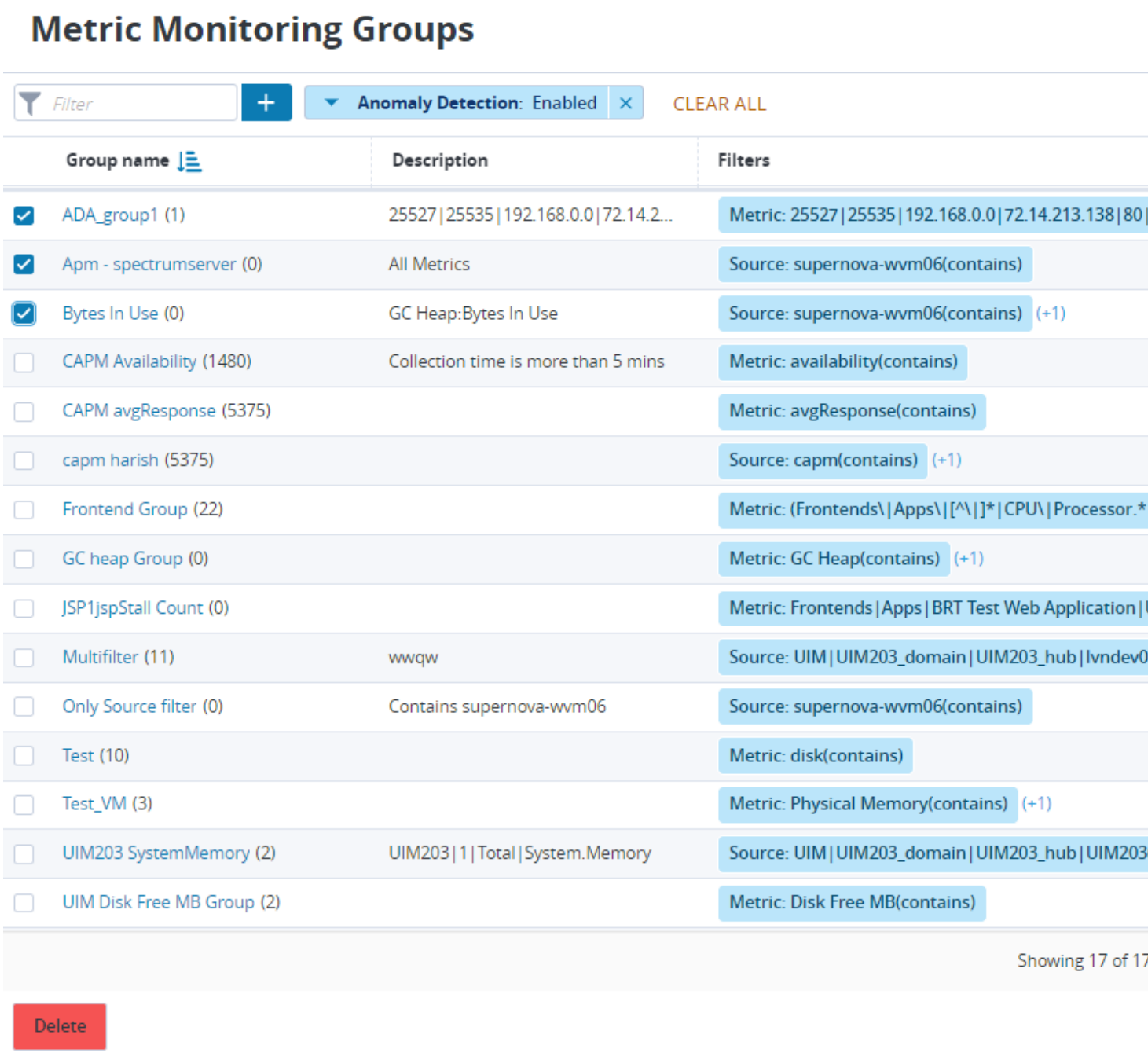
Delete a Metric Group

In a scenario where you have exceeded the defined metric limits, we recommend that you disable some metric groups from the existing list to enable the other metric groups for monitoring anomaly detection instead of deleting them. You can delete the existing metric groups using two ways:

- Click on a **Group name**, and then click **Delete** on the bottom-left of the view



- (OR) Select a **Group name** checkbox. (You can also use this option for deleting multiple metric groups)
  a. Click **Delete** at the bottom-left of the view
     You would see a pop-up requesting you to confirm the deletion.



b. Click **Delete**.

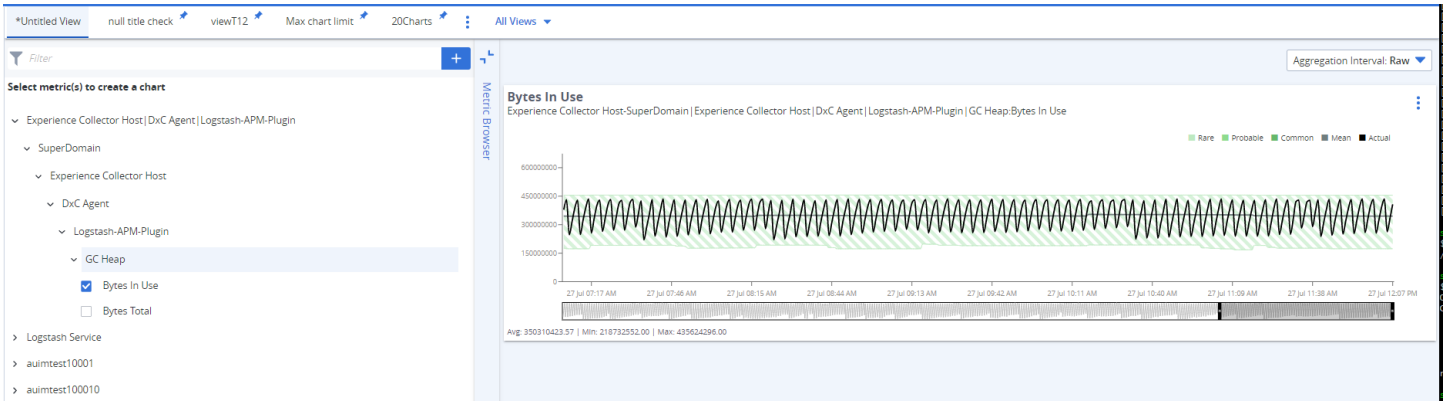
{"URL":["http://digital-oi/settings/monitoring-groups","http://digital-oi/settings/monitoring-groups/configureAlarms"],"description":"concept.dita_38fc6698-ba66-47c1-a842-ab5a13a9cf2e","new":"","new_video":"","admin":"","heroDescriptionIdentifier":"","troubleshooting":{"masterkb":"","text":"","URL":[]},"pendo":"","video":[],"customCards":[]}

How Does Anomaly Detection Works

For the configured metric groups in DX Operational Intelligence, you can enable Anomaly Detection. For more information about how to configure anomaly detection, see Configure Monitoring.

The identified anomalies for the given metrics can be viewed in Performance Analytics. In Performance Analytics, you can view the metric time series value and the anomalies that are identified by the system based on the single rare band, encapsulating the metric value across the timeline. You do not need to configure any thresholds as the algorithm learns from the historic data points of the metric. The algorithm understands the ordinary and anomalous behavior of the data to identify the correct anomalies. It also triggers alarms and notifications that are based on the configuration.

> **NOTE**
> The anomaly may trigger on individual samples within the given frequency. To understand the distribution, hover the mouse over the zones and see the minimum or maximum value.



Metric Triage Use Cases

Performance Analytics supports the following use cases:

- Use case 1: The Application owner wants to triage the performance issue for the application or a part of it
- Use case 2: The Service owner wants to triage the performance issue for the service or a part of it

Follow these steps:

1. From the **Metrics** Browser pane, select an application or service entity.

   The list of available Metrics and Configuration Items (CIs) for the selected entities appears. The hierarchy that builds up depends on the selected entity. If the selected entity has more CIs associated with it, then the hierarchy builds up in such a way that you view the list of CIs and the associated metrics of the CIs.

2. Select an **Entity** and click the drop-down arrow to view the associated **Configuration Items**.

   The list of associated CIs appears.

3. Click the drop-down arrow to view the associated **Metrics Family**.

4. Click the drop-down arrow to view the associated **Metrics** of the CIs. You can also click **Filter** at the parent level, to view the associated child entities.

   > **NOTE**
   > - The hierarchies may vary from one source product to another.

5. Select the required metric. The performance charts for the selected metrics appear on the Performance Analytics page.

6. If you select the **Calendar** duration as less than 3 days, the charts are plotted based on Raw metrics. These aggregated metrics are used to plot the chart for a duration of more than 3 days in Performance Analytics. The 15

minutes aggregation is used for up to a duration of 30 days. For greater than 30 days duration, the daily aggregation is used.

Capacity Analytics

Capacity Analytics is an analytical approach used to determine optimized resources for the continuity of operations. Capacity Analytics helps you in determining the required capacity for infrastructure resources such as CPU, memory, storage, and network for the operational continuity of the enterprise workloads.

Capacity analytics allow you to manage demands for IT resources proactively in a cost-effective manner. You can optimize the performance and efficiency of existing resources, plan for and justify any financial investments.

By using Capacity Analytics in DX Operational Intelligence, you can leverage the following benefits:

- Predict capacity for peak seasons.
- Understand when more resources are needed and plan accordingly.
- Only buy additional resources when required.
- Efficiently manage infrastructure and networks.
- Eliminate wastage of resources by identifying areas that are underutilized.

Who can use Capacity Analytics

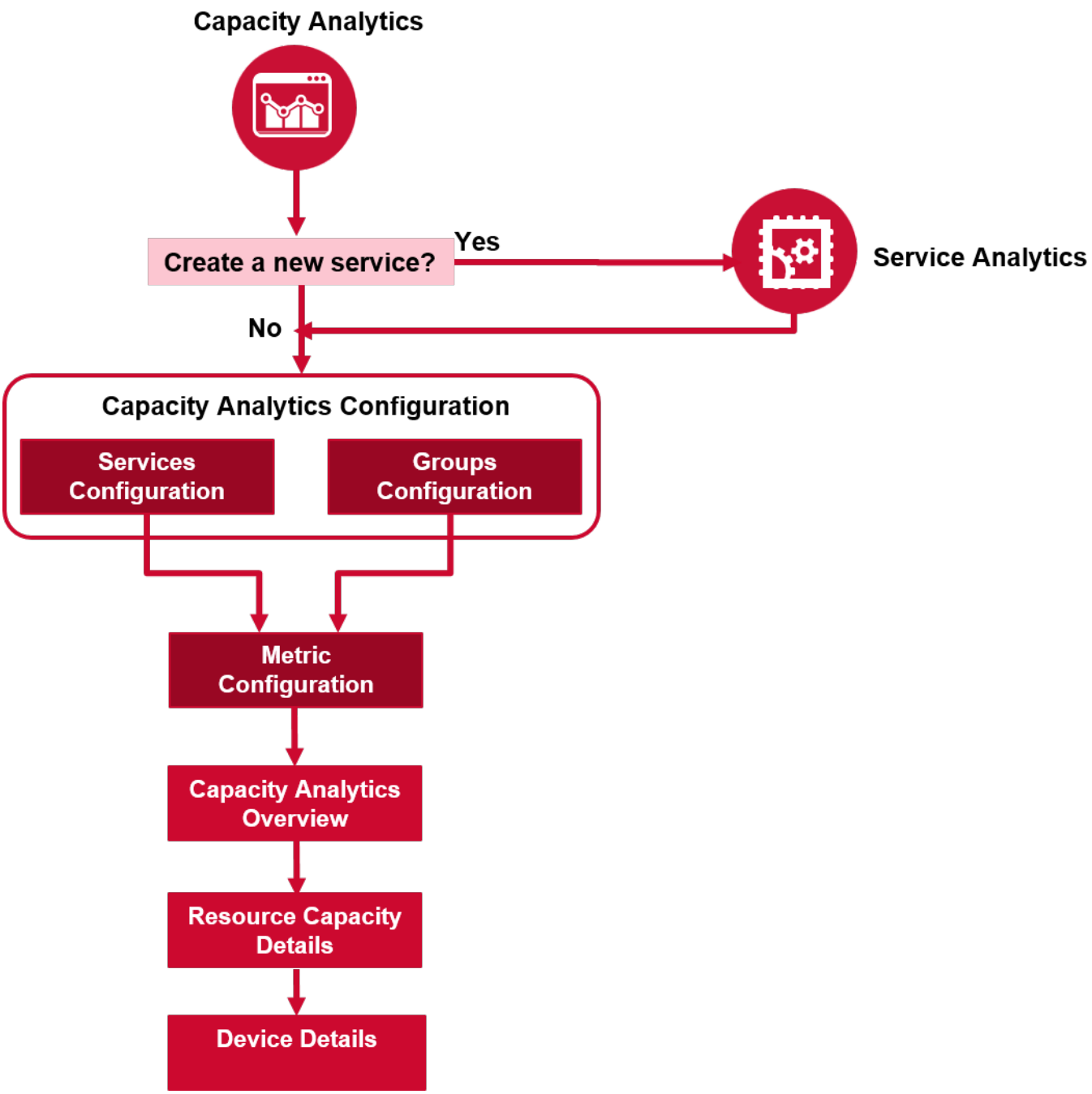
Users that have the need to forecast and manage the capacity of the enterprise infrastructure to ensure optimal performance for the workloads, while optimizing infrastructure costs can benefit from using Capacity Analytics.

The key users can be one of the following:

- Capacity Planners who are responsible for monitoring, forecasting, and managing the capacity of the infrastructure across the organization.
- Service Owners who are interested in monitoring and forecasting for their respective services and drilling down to their service topology.
- L2 operators who are interested in the capacity monitoring aspects of the devices while diagnosing infrastructure issues.

Capacity Analytics Process Flow

The following illustration depicts the process flow of Capacity Analytics.



Navigating Capacity Analytics Metrics Configuration Groups Configuration Services Configuration Service Analytics

NOTE
Click on any step in the flowchart to know more about the Capacity Analytics process.

Access Capacity Analytics

You can access Capacity Analytics from the left navigation pane in DX Operational Intelligence. The application displays the Capacity Analytics Overview page with a dashboard. You must have appropriate permissions and the login credentials to access Capacity Analytics dashboard in DX Operational Intelligence.

The Dashboard on the Capacity Analytics Overview page includes the following widgets:

- Health Chart
- Top Consumers
- Monitored Groups and Services

To view and analyze the Capacity Analytics data, complete the tasks in the following topics:

- Service Configuration
- Group Configuration
- Metrics for Capacity Analytics

Metrics for Capacity Analytics

This page gives you an overview of the various metrics that are applicable for Capacity Analytics.

Capacity Analytics supports the metrics from the following probes:

- Universal Monitoring Agent for Kubernetes
- cdm Probe
- vmware Probe
- netap-ontap Probe
- ibmvm Probe
- hpe_3par Probe

> **IMPORTANT**
> Capacity Analytics is now enhanced to display Kubernetes metrics from DX Application Performance Management (DX APM) for further analysis. But, in order to view the DX APM metrics you need to use the DX Application Performance Management Infrastructure Agent.

Universal Monitoring Agent for Kubernetes

Metric Name	Metric Type	Metric Unit	Linking Required ?	Metric to Be linked with	Description
CPU Utilization (mCore)	Usage	Count	Yes	CPU Limit (mCore)	Total CPU utilized by Node/POD/Cluster
CPU Limit (mCore)	Capacity	Count	Yes	CPU Utilization (mCore)	Total CPU available for Kubernetes Node/POD/ Cluster
Memory Usage (Bytes)	Usage	Bytes	Yes	Memory Limit (Bytes)	Total Memory Utilized by Node/POD/Cluster
Memory Limit (Bytes)	Capacity	Bytes	Yes	Memory Usage (Bytes)	Total Memory available for Kubernetes Node/POD/ Cluster

cdm Probe Metrics

Verify that the **cdm probe** is collecting the following metrics:

Metric Name	Metric ID	Description	Metric Unit	Linking Required ?	Metric to be linked with
Individual CPU System	QOS_CPU_MULTI_USAGE	The percentage of time when an individual CPU of the system was executing the kernel or operating system.	Percent	No	NA
Individual CPU Usage	QOS_CPU_MULTI_USAGE	The percentage of time for which an individual CPU of the system was used.	Percent	No	NA
Individual CPU User	QOS_CPU_MULTI_USAGE	The percentage of time for which an individual CPU of the system was executing in user mode.	Percent	No	NA
Individual CPU Wait	QOS_CPU_MULTI_USAGE	The percentage of time for which an individual CPU of the system was waiting for I/O.	Percent	No	NA
Individual CPU Steal	QOS_CPU_MULTI_USAGE	The percentage of time for which an individual CPU of the system was waiting involuntarily. During this time, the CPU is not in any of the following states: User; System; Wait; Idle	Percent	No	NA
Total CPU System	QOS_CPU_USAGE	The sum of CPU time when all CPUs of the system were executing the kernel or operating system.	Percent	No	NA
Total CPU Usage	QOS_CPU_USAGE	The percentage of time for which all CPUs of the system were used.	Percent	No	NA
Total CPU User	QOS_CPU_USAGE	The percentage of time for which all CPUs of the system were executing in user mode.	Percent	No	NA
Total CPU Wait	QOS_CPU_USAGE	The percentage of time for which all CPUs of the system were waiting for I/O.	Percent	No	NA

Metric Name	Metric ID	Description	Metric Unit	Linking Required ?	Metric to be linked with
Total CPU Steal	QOS_CPU_USAGE	The percentage of time for which all CPUs of the system were waiting involuntarily. During this time, the CPUs are not in any of the following states: User; System; Wait; Idle	Percent	No	NA
Smt	QOS_CPU_SMT	The number of simultaneous multithreading (SMT) threads enabled in a logical partition (LPAR). If there are two SMT threads, the row is displayed as "on." However, if there are more than two SMT threads, the number of SMT threads is displayed.	Count	Yes	Number of CPU
Number of CPU	QOS_CPU_NUM	This QoS is supported only on the AIX platform.	Count	Yes	Smt
Disk Usage Change (MB)	QOS_DISK_DELTA	The disk usage change in megabytes	Megabytes	Yes	Disk Usage (MB)
Disk Usage (MB)	QOS_DISK_USAGE	Aggregated disk usage in megabytes	Megabytes	Yes	Disk Usage Change (MB)
Disk Usage (%)	QOS_DISK_USAGE_PERC	Aggregated disk usage in percentage	Percent	No	NA
Disk Read (B/s)	QOS_DISK_READ_THROUGHPUT	Disk bytes read per second	Bytes/Second	Yes	Disk Read and Write (B/s)
Disk Write (B/s)	QOS_DISK_WRITE_THROUGHPUT	Disk bytes written per second	Bytes/Second	Yes	Disk Read and Write (B/s)
Disk Read and Write (B/s)	QOS_DISK_TOTAL_THROUGHPUT	Disk bytes read and written per second	Bytes/Second	Yes	Disk Read (B/s) or Disk Write (B/s)
Inode Usage (%)	QOS_INODE_USAGE_PERC	The number of free file nodes in file system in percentage.	Percent	No	NA
Disk Latency (ms)	QOS_DISK_LATENCY	This metric is the total latency of the disk.	Milliseconds	Yes	Disk Read Latency (ms) or Disk Write Latency (ms)
Disk Read Latency (ms)	QOS_DISK_READ_LATENCY	This metric is the read latency of the disk.	Milliseconds	Yes	Disk Latency (ms)

Metric Name	Metric ID	Description	Metric Unit	Linking Required ?	Metric to be linked with
Disk Write Latency (ms)	QOS_DISK_WRITE_LATENCY	This metric is the write latency of the disk.	Milliseconds	Yes	Disk Latency (ms)
Disk Usage Change (%)	QOS_DISK_DELTA_PERC	The disk usage change in percent	Percent	No	NA
Memory Usage (MB)	QOS_MEMORY_USAGE	Megabytes	Total memory usage in megabytes	Yes	Physical Memory (MB)
Memory Usage (%)	QOS_MEMORY_PERC_USAGE	Percent	Total memory usage in percentage	No	NA
Physical Memory (MB)	QOS_MEMORY_PHYSICAL	Megabytes	The size of the physical memory used on the system in megabytes.	Yes	Memory Usage (MB)
Physical Memory (%)	QOS_MEMORY_PHYSICAL_PERC	Percent	The size of the physical memory used on the system in percentage	No	NA
Swap Memory (%)	QOS_MEMORY_SWAP_PERC	Percent	Total Swap memory usage in percent	No	NA
System Memory Utilization (%)	QOS_MEMORY_SYS_UTIL	Percent	Total System memory utilization in percent. This metric is supported only on the Windows, Linux and AIX platforms.	No	NA
User Memory Utilization (%)	QOS_MEMORY_USR_UTIL	Percent	Total User memory utilization in percent. This metric is supported only on the Windows, Linux and AIX platforms.	No	NA
Total Memory	QOS_TOTAL_MEMORY_AME	MB	The size of the total memory in an LPAR in your system, in megabytes.	Yes	Memory Dxm
Memory Dxm	QOS_MEMORY_EXPANDED_DXM_AME	MB	The size of the memory deficit in an LPAR in your system, in megabytes. At times an LPAR cannot be configured with the provided memory expansion factor as it is too large, and the workload in the LPAR does not compress well. Thus, a memory deficit is created.	Yes	Total Memory

Metric Name	Metric ID	Description	Metric Unit	Linking Required ?	Metric to be linked with
Network Inbound Traffic	QOS_NETWORK_INBOUND_TRAFFIC	The total number of bytes per second received by the server.	Bytes/Second	Yes	Network Aggregated Traffic
Network Outbound Traffic	QOS_NETWORK_OUTBOUND_TRAFFIC	The total number of bytes per second sent by the server.	Bytes/Second	Yes	Network Aggregated Traffic
Network Aggregated Traffic	QOS_NETWORK_AGGREGATED_TRAFFIC	The total number of bytes per second sent and received by the server.	Bytes/Second	Yes	Network Inbound Traffic or Network Outbound Traffic
Interface Errors % (In)	QOS_INBOUND_ERRORS_PERC	Inbound errors in percent.	Percent	No	NA
Interface Errors % (Out)	QOS_OUTBOUND_ERRORS_PERC	Outbound errors in percent.	Percent	No	NA
Interface Discards % (In)	QOS_INBOUND_DISCARDS_PERC	Inbound discards in percent. This metric is not supported on the Solaris platform.	Percent	No	NA
Interface Discards % (Out)	QOS_OUTBOUND_DISCARDS_PERC	Outbound discards in percent. This metric is not supported on the Solaris platform.	Percent	No	NA
Interface Utilization % (In)	QOS_INBOUND_UTILIZATION_PERC	Inbound interface utilization in percent. This metric is not supported on the AIX platform.	Percent	No	NA
Interface Utilization % (Out)	QOS_OUTBOUND_UTILIZATION_PERC	Outbound interface utilization in percent. This metric is not supported on the AIX platform.	Percent	No	NA

vmware Probe Metrics

If you are using the **vmware** probe, activate the following QoS metrics:

Metric Name	Description	Metric Unit	Linking Required ?	Metric to be linked with
GuestMemoryUsage (in % of Memory)	This monitor indicates the guest memory utilization, also known as, active guest memory. The number can be between 0 and the configured memory size of the virtual machine. This monitor is only valid while the virtual machine is running.	Percent	No	NA
HostCpuUsage	This monitor indicates the basic CPU performance statistics. This monitor is only valid while the virtual machine is running.	Percent	No	NA
HostMemoryUsage	This monitor indicates the memory that is consumed on the host for the virtual machine. This is also known as consumed host memory. This monitor is only valid while the virtual machine is running.	MBytes	Yes	Memory
HostMemoryUsage (in % of Memory)	This monitor indicates the host memory utilization. This is also known as consumed host memory. This value is between 0 and the configured resource limit. This monitor is only valid while the virtual machine is running.	Percent	No	NA
Memory	This monitor indicates the memory size in MB.	MBytes	Yes	HostMemoryUsage
Storage Provisioned	This monitor indicates the provisioned storage space for this virtual machine.	GB	Yes	Storage Usage
Storage Usage	This monitor indicates the committed space for this virtual machine on the datastore.	GB	Yes	Storage Provisioned
CPU Latency	This monitor indicates the percentage of time the virtual machine is unable to run because it is contending for access to the physical CPU(s).	Percent	No	NA

Metric Name	Description	Metric Unit	Linking Required ?	Metric to be linked with
CPU Max Limited (% of available)	This monitor indicates the percentage of time the virtual machine is ready to run, but is not run due to maxing out its CPU limit setting.	Percent	No	NA
CPU Ready (% of available)	This monitor indicates the percentage of time that the virtual machine was ready, but could not get scheduled to run on the physical CPU.	Percent	No	NA
CPU Run (% of available)	This monitor indicates the percentage of time the virtual machine is scheduled to run.	Percent	No	NA
CPU Swap Wait (% of available)	This monitor indicates the percentage of CPU time spent waiting for swap-in.	Percent	No	NA
CPU System (% of available)	This monitor indicates the percentage of time spent on system processes for each virtual CPU in the virtual machine.	Percent	No	NA
CPU Usage (Average/ Rate)	This monitor entage of CPU usage during the interval.	Percent	No	NA
CPU Used (% of available)	This monitor indicates the total CPU usage.	Percent	No	NA
CPU Wait (% of available)	This monitor indicates the percentage of CPU time spent in wait state.	Percent	No	NA
CPU Latency	This monitor indicates the percentage of time the virtual machine is unable to run because it is contending for access to the physical CPU(s).	Percent	No	NA
CPU Limited by Max (% of available)	This monitor indicates the percentage of time the virtual machine is ready to run, but is not run due to maxing out its CPU limit setting.	Percent	No	NA
CPU System (% of available)	This monitor indicates the percentage of time spent on system processes on each virtual CPU in the virtual machine.	Percent	No	NA

Metric Name	Description	Metric Unit	Linking Required ?	Metric to be linked with
CPU Usage (Average/ Rate)	This monitor indicates the average percentage of CPU that is used during the interval.	Percent	No	NA
CPU Used (% of available)	This monitor indicates the total CPU usage.	Percent	No	NA
CPU Wait (% of available)	This monitor indicates the percentage of CPU time spent in wait state.	Percent	No	NA
CPU active average over 1 min (% of mhz*NumCpuCores)	This monitor indicates the CPU active averaged over one minute.	Percent	No	NA
CPU active average over 15 min (% of mhz*NumCpuCores)	This monitor indicates the CPU active averaged over 15 minutes.	Percent	No	NA
CPU active average over 5 min (% of mhz*NumCpuCores)	This monitor indicates the CPU active averaged over five minutes.	Percent	No	NA
CPU active peak over 1 min (% of mhz*NumCpuCores)	This monitor indicates the CPU active peak over one minute.	Percent	No	NA
CPU active peak over 15 min (% of mhz*NumCpuCores)	This monitor indicates the CPU active peak over 15 minutes.	Percent	No	NA
CPU active peak over 5 min (% of mhz*NumCpuCores)	This monitor indicates the CPU active peak over five minutes.	Percent	No	NA
CPU refused average over 1 min (% of mhz*NumCpuCores).	This monitor indicates the amount of CPU resources over the limit that were refused, averaged over one minute.	Percent	No	NA
CPU refused average over 15 min (% of mhz*NumCpuCores).	This monitor indicates the amount of CPU resources over the limit that were refused, averaged over 15 minutes.	Percent	No	NA
CPU refused average over 5 min (% of mhz*NumCpuCores).	This monitor indicates the amount of CPU resources over the limit that were refused, averaged over five minutes.	Percent	No	NA
CPU running average over 1 min (% of mhz*NumCpuCores)	This monitor indicates the CPU running averaged over one minute.	Percent	No	NA
CPU running average over 15 min (% of mhz*NumCpuCores)	This monitor indicates the CPU running averaged over 15 minutes.	Percent	No	NA

Metric Name	Description	Metric Unit	Linking Required ?	Metric to be linked with
CPU running average over 5 min (% of mhz*NumCpuCores)	This monitor indicates the CPU running averaged over five minutes.	Percent	No	NA
CPU running peak over 1 min (% of mhz*NumCpuCores)	This monitor indicates the CPU running peak over one minute.	Percent	No	NA
CPU running peak over 15 min (% of mhz*NumCpuCores)	This monitor indicates the CPU running peak over 15 minutes.	Percent	No	NA
CPU running peak over 5 min (% of mhz*NumCpuCores)	This monitor indicates the CPU running peak over five minutes.	Percent	No	NA
Memory Active (% of Memory)	This monitor indicates the amount of memory that is actively used, as estimated by VMkernel based on recently touched memory pages.	Percent	No	NA
Memory Consumed (% of Memory)	This monitor indicates the percentage of memory that is consumed by a virtual machine, host, or cluster.	Percent	No	NA
Memory Overhead (% of Memory)	This monitor indicates the percentage of memory that is consumed by the virtualization infrastructure for running the VM.	Percent	No	NA
Memory Usage	This monitor indicates the average percentage of memory that is used during the interval.	Percent	No	NA
VMCount	This monitor indicates the number of virtual machines on this host.	Count	Yes	VMCountActive
VMCountActive	This monitor indicates the number of active virtual machines on this host.	Count	Yes	VMCount
CPU Used (% of available)	This monitor indicates the total CPU usage.	Percent	No	NA
CPU Reserved Capacity (% of mhz*NumCpuCores)	This monitor indicates the total CPU capacity reserved by the virtual machines.	Percent	No	NA
CPU Usage (Average/ Rate)	This monitor indicates the average percentage of CPU that is used during the interval.	Percent	No	NA
CPU Used (% of available)	This monitor indicates the total CPU usage.	Percent	No	NA

Metric Name	Description	Metric Unit	Linking Required ?	Metric to be linked with
CPU active average over 1 min (% of mhz*NumCpuCores)	This monitor indicates the CPU active averaged over one minute.	Percent	No	NA
CPU active average over 15 min (% of mhz*NumCpuCores)	This monitor indicates the CPU active averaged over 15 minutes.	Percent	No	NA
CPU active average over 5 min (% of mhz*NumCpuCores)	This monitor indicates the CPU active averaged over five minutes.	Percent	No	NA
CPU active peak over 1 min (% of mhz*NumCpuCores)	This monitor indicates the CPU active peak over one minute.	Percent	No	NA
CPU active peak over 15 min (% of mhz*NumCpuCores)	This monitor indicates the CPU active peak over 15 minutes.	Percent	No	NA
CPU active peak over 5 min (% of mhz*NumCpuCores)	This monitor indicates the CPU active peak over five minutes.	Percent	No	NA
CPU running average over 1 min (% of mhz*NumCpuCores)	This monitor indicates the CPU running averaged over one minute.	Percent	No	NA
CPU running average over 15 min (% of mhz*NumCpuCores)	This monitor indicates the CPU running averaged over 15 minutes.	Percent	No	NA
CPU running average over 5 min (% of mhz*NumCpuCores)	This monitor indicates the CPU running averaged over 5 minutes.	Percent	No	NA
CPU running peak over 1 min (% of mhz*NumCpuCores)	This monitor indicates the CPU running peak over one minute.	Percent	No	NA
CPU running peak over 15 min (% of mhz*NumCpuCores)	This monitor indicates the CPU running peak over 15 minutes.	Percent	No	NA
CPU running peak over 5 min (% of mhz*NumCpuCores)	This monitor indicates the CPU running peak over five minutes.	Percent	No	NA
Memory Active (% of MemorySize)	This monitor indicates the percentage of memory that is actively used, as estimated by VMkernel based on recently touched memory pages.	Percent	No	NA

Metric Name	Description	Metric Unit	Linking Required ?	Metric to be linked with
Memory Balloon (% of MemorySize)	This monitor indicates the percentage of memory allocated by the virtual machine memory control driver (vmmemctl), which is installed with VMware Tools. This is the sum of all virtual machine values, plus vSphere services on the host, as a percentage of memory size.	Percent	No	NA
Memory Balloon Target (% of MemorySize)	This monitor indicates the target percentage value set by VMkernal for the virtual machine's memory balloon size.	Percent	No	NA
Memory Consumed (% of MemorySize)	This monitor indicates the percentage of memory that is consumed by a virtual machine, host, or cluster.	Percent	No	NA
Memory Granted (% of MemorySize)	This monitor indicates the percentage of machine memory or physical memory that is mapped for a virtual machine or a host.	Percent	No	NA
Memory Heap (% of MemorySize)	This monitor indicates the memory heap (% of MemorySize).	Percent	No	NA
Memory Heap Free (% of MemorySize)	This monitor indicates the mmory heap free (% of MemorySize).	Percent	No	NA
Memory Reserved Capacity (% of MemorySize)	This monitor indicates the total amount of memory reservation used by powered-on virtual machines and vSphere services on the host.	Percent	No	NA
Memory Reserved Overhead (% of MemorySize)	This monitor indicates the memory (percent) reserved for use as the virtualization overhead.	Percent	No	NA
Memory used by vmkernel (% of MemorySize)	This monitor indicates the amount of machine memory used by VMkernel for core functionality, such as device drivers and other internal uses.	Percent	No	NA
Resource CPU Usage in MHz (% of mhz*NumCpuCores)	This monitor indicates the percentage of average CPU usage.	Percent	No	NA

Metric Name	Description	Metric Unit	Linking Required ?	Metric to be linked with
CPU Capacity Entitlement	This monitor indicates the CPU resources devoted by the ESX scheduler to the virtual machines and resource pools.	Megahertz	Yes	CPU Capacity Usage
CPU Capacity Usage	This monitor indicates the CPU usage during the interval.	megahertz	Yes	CPU Capacity Entitlement
CPU Usage (% of CPUMaxUsage)	This monitor indicates the CPU usage as a percentage of maximum CPU usage.	Percent	No	NA
CPUOverallUsage (% of CPUMaxUsage)	This monitor indicates the real-time resource usage of all running child virtual machines, including virtual machines in child resource pools. This will be the percentage of current upper bound on usage.	Percent	No	NA
MemoryOverallUsage (% of MemoryMaxUsage)	This monitor indicates the real-time resource usage of all running child virtual machines, including virtual machines in child resource pools. This will be the percentage of current upper bound on usage.	Percent	No	NA
Provisioned Space	This monitor indicates the provisioned space on this datastore, in gigabytes.	GBytes	Yes	Usage
Usage	This monitor indicates the used space on this datastore, in gigabtyes.	GB	Yes	Provisioned Space
CPU Usage	This monitor indicates the aggregated CPU usage of hosts.	MHz	Yes	TotalCPU
Memory Usage	This monitor indicates the aggregated memory usage of hosts.	MB	Yes	TotalMemory
TotalCPU	This monitor indicates the aggregated CPU resources of all hosts.	MHz	Yes	CPU Usage
TotalMemory	This monitor indicates the aggregated memory resources of all hosts.	MBytes	Yes	Memory Usage
CPU Usage (% of CPUMaxUsage)	This monitor indicates the current CPU usage as a percentage of maximum CPU usage.	Percent	No	NA

Metric Name	Description	Metric Unit	Linking Required ?	Metric to be linked with
CPUMaxUsage	This monitor indicates the current upper-bound on usage. The upper-bound is based on the limit configured on this resource pool, as well as limits configured on any parent resource pool.	MHz	Yes	CPUOverallUsage
CPUOverallUsage	This monitor indicates the real-time resource usage of all running child virtual machines, including virtual machines in child resource pools.	MHz	Yes	CPUMaxUsage
CPUOverallUsage (% of CPUMaxUsage)	This monitor indicates the real-time resource usage of all running child virtual machines, including virtual machines in child resource pools. This will be the percentage of current upper bound on usage.	Percent	No	NA
MemoryOverallUsage (% of MemoryMaxUsage)	This monitor indicates the real-time resource usage of all running child virtual machines, including virtual machines in child resource pools. This will be the percentage of current upper bound on usage.	Percent	No	NA
Capacity	This monitor indicates the maximum capacity of this datastore.	GBytes	Yes	Usage
Usage	This monitor indicates the used space on this datastore, in gigabtyes.	GB	Yes	Capacity

netap-ontap Probe Metrics

If you are using the **netap-ontap** probe, activate the following QoS metrics.

Metric Name	Metric ID	Description	Metric Unit	Linking Required?	Metric to be linked with
Total Capacity	QOS_STORAGE_TOTAL_CAPACITY_AGGR	The total capacity of the referenced file system.	GB	Yes	Used Capacity
Used Capacity	QOS_STORAGE_USED_CAPACITY_AGGR	The used capacity of the referenced file system.	GB	Yes	Total Capacity

Metric Name	Metric ID	Description	Metric Unit	Linking Required?	Metric to be linked with
Used Capacity Percent	QOS_STORAGE_PERCENT_CAPACITY_USED_AGGR	The used capacity in percentage of the referenced file system.	Percent	No	NA
Snapshot Total Capacity	QOS_STORAGE_TOTAL_SNAPSHOT_CAPACITY_AGGR	The total capacity for the snapshot on the referenced file system.	GB	Yes	Snapshot Used Capacity
Snapshot Used Capacity	QOS_STORAGE_SNAPSHOT_USED_CAPACITY_AGGR	The used capacity for the snapshot on the referenced file system.	GB	Yes	Snapshot Total Capacity
Snapshot Percent Used Capacity	QOS_STORAGE_PERCENT_SNAPSHOT_USED_AGGR	The total used disk space for the snapshot in percentage on the referenced file system.	Percent	Yes	NA
Total Disk Count	QOS_STORAGE_TOTAL_DISK_COUNT	The total number of disks on the system.	Count	Yes	Failed Disk Count
Failed Disk Count	QOS_STORAGE_NUMBER_OF_FAILED_DISKS	The number of disks that are currently broken.	Count	Yes	Total Disk Count
Disk Utilization	QOS_STORAGE_DISK_UTILIZATION	The disk utilization percentage on the referenced disk drive.	Percent	No	NA
Average Latency	QOS_STORAGE_TOTAL_AVERAGE_LATENCY	The average latency for all operations in the system in milliseconds.	ms	Yes	Average Other Latency, Average Read Latency, Average Write Latency
Average Other Latency	QOS_STORAGE_AVERAGE_OTHER_LATENCY	The average latency for all other operations in the system in milliseconds.	ms	Yes	Average Latency
Average Read Latency	QOS_STORAGE_AVERAGE_READ_LATENCY	The average latency for all read operations in the system in milliseconds.	ms	Yes	Average Latency
Average Write Latency	QOS_STORAGE_AVERAGE_WRITE_LATENCY	The average latency for all write operations in the system in milliseconds.	ms	Yes	Average Latency

Metric Name	Metric ID	Description	Metric Unit	Linking Required?	Metric to be linked with
CPU Utilization	QOS_STORAGE_CPU_UTILIZATION	The percent of time that the CPU has been doing useful work since the last time a client requested the CPU Busy Time Percent.	Percent	No	NA
Total Read IOPS	QOS_STORAGE_READ_IOPS	The total read operations per second.	count/s	Yes	System IOPS
Total Write IOPS	QOS_STORAGE_WRITE_IOPS	The total write operations per second.	count/s	Yes	System IOPS
System IOPS	QOS_STORAGE_SYSTEM_IOPS	The total system operations per second.	count/s	Yes	Total Read IOPS, Total Write IOPS
LUN Percent Used	QOS_STORAGE_LUN_PERCENT_USED	The percent utilization for the LUN.	Percent	No	NA
LUN Size	QOS_STORAGE_LUN_TOTAL_SIZE	The total size of the LUN.	GB	Yes	LUN Size Used
LUN Size Used	QOS_STORAGE_LUN_SIZE_USED	The size used in GB that is in use on the LUN.	GB	Yes	LUN Size
Qtree Quota	QOS_STORAGE_QUOTA_QTREE	The quota for growth of the Qtree in GB.	GB	Yes	Qtree Space Used
Qtree Percent Used	QOS_STORAGE_PERCENT_USED_QTREE	Percent tagged space used in the Qtree.	Percent	No	NA
Qtree Space Used	QOS_STORAGE_SPACE_USED_QTREE	The storage space used by the Qtree in GB.	GB	Yes	Qtree Quota
Total Capacity	QOS_STORAGE_TOTAL_CAPACITY_VOL	The total capacity of the volume.	GB	Yes	Used Capacity
Used Capacity	QOS_STORAGE_USED_CAPACITY_VOL	The used capacity of the volume.	GB	Yes	Total Capacity
Total Snapshot Capacity	QOS_STORAGE_TOTAL_SNAPSHOT_CAPACITY_VOL	The total capacity for the snapshot in the volume.	GB	Yes	Snapshot Used Capacity
Snapshot Used Capacity	QOS_STORAGE_SNAPSHOT_USED_CAPACITY_VOL	The used capacity for the snapshot in the volume.	GB	Yes	Total Snapshot Capacity

ibmvm Probe

If you are using the **ibmvm** probe, activate the following QoS metrics.

Metric Name	Metric ID	Description	Metric Unit	Linking Required ?	Metric to be linked with
Percent Used	RESOURCE_MEM	HMC percent memory used.	Percent	No	NA
Total	RESOURCE_MEM	HMC total memory.	Megabytes	Yes	Used
Used	RESOURCE_MEM	HMC used memory.	Megabytes	Yes	Total
Cached	RESOURCE_SWAP	HMC swapspace cached.	Megabytes	No	NA
Percent Used	RESOURCE_SWAP	HMC percent swap used.	Percent	No	NA
Total	RESOURCE_SWAP	HMC total swap capacity.	Megabytes	Yes	Used
Used	RESOURCE_SWAP	HMC used swap.	Megabytes	Yes	Total
I/O Waiting	RESOURCE_CPU	HMC CPU I/O waiting percentage.	Percent	No	NA
Percent Used	RESOURCE_DISK	Percent usage for this HMC filesystem.	Percent	No	NA
Total	RESOURCE_DISK	Total available capacity of the HMC disk.	Megabytes	Yes	Used
Used	RESOURCE_DISK	Used filesystem space on the HMC disk.	Megabytes	Yes	Total
VM Total	HOST	The current VM count.	Number	Yes	VMs Stopped
VMs Stopped	HOST	Number of stopped VMs.	Number	Yes	VM Total
Assigned Memory	HOST_MEMORY	Amount of memory assigned to VMs. This figure is equal to the "Installed Memory" minus "Unassigned Memory".	Megabytes	Yes	Installed Memory
Installed Memory	HOST_MEMORY	Total amount of memory installed on the managed system.	Megabytes	Yes	Assigned Memory
CoD Processors Activated	HOST_COD	The total number of processors activated for CoD.	Count	Yes	CoD Processors Available
CoD Processors Available	HOST_COD	The number of processors available for CoD.	Count	Yes	CoD Processors Activated
CoD Processors Requested Days Available	HOST_COD	The total number of days for CoD processor requests.	Days	Yes	CoD Processors Requested Days Left

Metric Name	Metric ID	Description	Metric Unit	Linking Required ?	Metric to be linked with
CoD Processors Requested Days Left	HOST_COD	The remaining number of days available for CoD processor requests.	Days	Yes	CoD Processors Requested Days Available
Disk Bandwidth Used (%)	HOST_DISK	Network interface description.	Percent	No	NA
Global Shared Processor Pool Utilization	CPU_POOL	The average global shared processor pool utilization.	Percent	No	NA
Shared Processor Pool Utilization	CPU_MSPP	The average shared processor pool utilization.	Percent	No	NA
Storage Pool Utilization	SR	Percentage of storage space used.	Percent	No	NA
Average CPU Utilization	VIO	The average CPU utilization of this VIO Server.	Percent	No	NA
Physical Processors Consumed	VIO_CPU	The number of physical processors used by the partition	Processing Units	Yes	Current Processing Units
Processor Entitlement Consumed	VIO_CPU	Percentage of the entitled processing cycles used by the partition.	Percent	No	NA
Disk Bandwidth Used (%)	VIO_DISK	Percent bandwidth used for each disk.	Percent	No	NA
Interfaces UP	VIO_NIC_GROUP	Count of interfaces up.	Number	Yes	Interfaces DOWN
Assigned Processors	VM_CPU	Number of processors or virtual processors that are varied on for the partition.	Processing Units	Yes	Physical Processors Consumed
Physical Processors Consumed	VM_CPU	The number of physical processors used by the partition.	Processing Units	Yes	Assigned Processors
Processor Entitlement Consumed	VM_CPU	Percentage of the entitled processing cycles used by the partition.	Percent	No	NA
Assigned Memory	VM_MEMORY	Amount of memory assigned to the partition.	Megabytes	Yes	Used Memory
Used Memory	VM_MEMORY	Memory used by the partition at the time of sample.	Megabytes	Yes	Assigned memory
Disk Bandwidth Used (%)	VM_DISK	Percent bandwidth used for each disk.	Percent	No	NA

hpe_3par Probe

If you are using the **hpe_3par** probe, activate the following QoS metrics.

Metric Name	Metric ID	Description	Metric Unit	Linking Required ?	Metric to be linked with
Total Capacity	QOS_HP_3PAR_RAW_TOTAL_CAPACITY	The total raw storage capacity (in gigabytes) of the StoreServ system.	GB	Yes	Used Capacity
Used Capacity	QOS_HP_3PAR_RAW_USED_CAPACITY	The used raw storage capacity (in gigabytes) of the StoreServ system.	GB	Yes	Total Capacity
WriteCacheHit	QOS_STORAGE_NODE_WRITE_CACHE_PCT	Write Cache Hit Ratio	Percent	No	NA
ReadCacheHit	QOS_STORAGE_NODE_READ_CACHE_PCT	Read Cache Hit Ratio	Percent	No	NA
CPU Utilization	QOS_CONTROLLER_CPU_UTILIZATION	The total CPU utilization of the controller node.	Percent	No	NA
Utilization	QOS_CPG_UTILIZATION	The percentage of CPG capacity that is currently in use.	Percent	No	NA
Utilization	QOS_LOGICAL_DISK_UTILIZATION	The percentage of used capacity of the logical disk.	Percent	No	NA
Utilization	QOS_VIRTUAL_VOLUME_UTILIZATION	The percentage of used capacity of the virtual volume.	Percent	No	NA

vmware Probe Metrics

Capacity Analytics uses QoS metrics from the vmware probe.

If you are using the **vmware** probe, activate the following QoS metrics:

Metric Name	Description	Metric Unit	Linking Required ?	Metric to be linked with
GuestMemoryUsage (in % of Memory)	This monitor indicates the guest memory utilization, also known as, active guest memory. The number can be between 0 and the configured memory size of the virtual machine. This monitor is only valid while the virtual machine is running.	Percent	No	NA
HostCpuUsage	This monitor indicates the basic CPU performance statistics. This monitor is only valid while the virtual machine is running.	Percent	No	NA

Metric Name	Description	Metric Unit	Linking Required ?	Metric to be linked with
HostMemoryUsage	This monitor indicates the memory that is consumed on the host for the virtual machine. This is also known as consumed host memory. This monitor is only valid while the virtual machine is running.	MBytes	Yes	Memory
HostMemoryUsage (in % of Memory)	This monitor indicates the host memory utilization. This is also known as consumed host memory. This value is between 0 and the configured resource limit. This monitor is only valid while the virtual machine is running.	Percent	No	NA
Memory	This monitor indicates the memory size in MB.	MBytes	Yes	HostMemoryUsage
Storage Provisioned	This monitor indicates the provisioned storage space for this virtual machine.	GB	Yes	Storage Usage
Storage Usage	This monitor indicates the committed space for this virtual machine on the datastore.	GB	Yes	Storage Provisioned
CPU Latency	This monitor indicates the percentage of time the virtual machine is unable to run because it is contending for access to the physical CPU(s).	Percent	No	NA
CPU Max Limited (% of available)	This monitor indicates the percentage of time the virtual machine is ready to run, but is not run due to maxing out its CPU limit setting.	Percent	No	NA
CPU Ready (% of available)	This monitor indicates the percentage of time that the virtual machine was ready, but could not get scheduled to run on the physical CPU.	Percent	No	NA
CPU Run (% of available)	This monitor indicates the percentage of time the virtual machine is scheduled to run.	Percent	No	NA
CPU Swap Wait (% of available)	This monitor indicates the percentage of CPU time spent waiting for swap-in.	Percent	No	NA

Metric Name	Description	Metric Unit	Linking Required ?	Metric to be linked with
CPU System (% of available)	This monitor indicates the percentage of time spent on system processes for each virtual CPU in the virtual machine.	Percent	No	NA
CPU Usage (Average/ Rate)	This monitor entage of CPU usage during the interval.	Percent	No	NA
CPU Used (% of available)	This monitor indicates the total CPU usage.	Percent	No	NA
CPU Wait (% of available)	This monitor indicates the percentage of CPU time spent in wait state.	Percent	No	NA
CPU Latency	This monitor indicates the percentage of time the virtual machine is unable to run because it is contending for access to the physical CPU(s).	Percent	No	NA
CPU Limited by Max (% of available)	This monitor indicates the percentage of time the virtual machine is ready to run, but is not run due to maxing out its CPU limit setting.	Percent	No	NA
CPU System (% of available)	This monitor indicates the percentage of time spent on system processes on each virtual CPU in the virtual machine.	Percent	No	NA
CPU Usage (Average/ Rate)	This monitor indicates the average percentage of CPU that is used during the interval.	Percent	No	NA
CPU Used (% of available)	This monitor indicates the total CPU usage.	Percent	No	NA
CPU Wait (% of available)	This monitor indicates the percentage of CPU time spent in wait state.	Percent	No	NA
CPU active average over 1 min (% of mhz*NumCpuCores)	This monitor indicates the CPU active averaged over one minute.	Percent	No	NA
CPU active average over 15 min (% of mhz*NumCpuCores)	This monitor indicates the CPU active averaged over 15 minutes.	Percent	No	NA
CPU active average over 5 min (% of mhz*NumCpuCores)	This monitor indicates the CPU active averaged over five minutes.	Percent	No	NA
CPU active peak over 1 min (% of mhz*NumCpuCores)	This monitor indicates the CPU active peak over one minute.	Percent	No	NA

Metric Name	Description	Metric Unit	Linking Required ?	Metric to be linked with
CPU active peak over 15 min (% of mhz*NumCpuCores)	This monitor indicates the CPU active peak over 15 minutes.	Percent	No	NA
CPU active peak over 5 min (% of mhz*NumCpuCores)	This monitor indicates the CPU active peak over five minutes.	Percent	No	NA
CPU refused average over 1 min (% of mhz*NumCpuCores).	This monitor indicates the amount of CPU resources over the limit that were refused, averaged over one minute.	Percent	No	NA
CPU refused average over 15 min (% of mhz*NumCpuCores).	This monitor indicates the amount of CPU resources over the limit that were refused, averaged over 15 minutes.	Percent	No	NA
CPU refused average over 5 min (% of mhz*NumCpuCores).	This monitor indicates the amount of CPU resources over the limit that were refused, averaged over five minutes.	Percent	No	NA
CPU running average over 1 min (% of mhz*NumCpuCores)	This monitor indicates the CPU running averaged over one minute.	Percent	No	NA
CPU running average over 15 min (% of mhz*NumCpuCores)	This monitor indicates the CPU running averaged over 15 minutes.	Percent	No	NA
CPU running average over 5 min (% of mhz*NumCpuCores)	This monitor indicates the CPU running averaged over five minutes.	Percent	No	NA
CPU running peak over 1 min (% of mhz*NumCpuCores)	This monitor indicates the CPU running peak over one minute.	Percent	No	NA
CPU running peak over 15 min (% of mhz*NumCpuCores)	This monitor indicates the CPU running peak over 15 minutes.	Percent	No	NA
CPU running peak over 5 min (% of mhz*NumCpuCores)	This monitor indicates the CPU running peak over five minutes.	Percent	No	NA
Memory Active (% of Memory)	This monitor indicates the amount of memory that is actively used, as estimated by VMkernel based on recently touched memory pages.	Percent	No	NA

Metric Name	Description	Metric Unit	Linking Required ?	Metric to be linked with
Memory Consumed (% of Memory)	This monitor indicates the percentage of memory that is consumed by a virtual machine, host, or cluster.	Percent	No	NA
Memory Overhead (% of Memory)	This monitor indicates the percentage of memory that is consumed by the virtualization infrastructure for running the VM.	Percent	No	NA
Memory Usage	This monitor indicates the average percentage of memory that is used during the interval.	Percent	No	NA
VMCount	This monitor indicates the number of virtual machines on this host.	Count	Yes	VMCountActive
VMCountActive	This monitor indicates the number of active virtual machines on this host.	Count	Yes	VMCount
CPU Used (% of available)	This monitor indicates the total CPU usage.	Percent	No	NA
CPU Reserved Capacity (% of mhz*NumCpuCores)	This monitor indicates the total CPU capacity reserved by the virtual machines.	Percent	No	NA
CPU Usage (Average/ Rate)	This monitor indicates the average percentage of CPU that is used during the interval.	Percent	No	NA
CPU Used (% of available)	This monitor indicates the total CPU usage.	Percent	No	NA
CPU active average over 1 min (% of mhz*NumCpuCores)	This monitor indicates the CPU active averaged over one minute.	Percent	No	NA
CPU active average over 15 min (% of mhz*NumCpuCores)	This monitor indicates the CPU active averaged over 15 minutes.	Percent	No	NA
CPU active average over 5 min (% of mhz*NumCpuCores)	This monitor indicates the CPU active averaged over five minutes.	Percent	No	NA
CPU active peak over 1 min (% of mhz*NumCpuCores)	This monitor indicates the CPU active peak over one minute.	Percent	No	NA
CPU active peak over 15 min (% of mhz*NumCpuCores)	This monitor indicates the CPU active peak over 15 minutes.	Percent	No	NA
CPU active peak over 5 min (% of mhz*NumCpuCores)	This monitor indicates the CPU active peak over five minutes.	Percent	No	NA

Metric Name	Description	Metric Unit	Linking Required ?	Metric to be linked with
CPU running average over 1 min (% of mhz*NumCpuCores)	This monitor indicates the CPU running averaged over one minute.	Percent	No	NA
CPU running average over 15 min (% of mhz*NumCpuCores)	This monitor indicates the CPU running averaged over 15 minutes.	Percent	No	NA
CPU running average over 5 min (% of mhz*NumCpuCores)	This monitor indicates the CPU running averaged over 5 minutes.	Percent	No	NA
CPU running peak over 1 min (% of mhz*NumCpuCores)	This monitor indicates the CPU running peak over one minute.	Percent	No	NA
CPU running peak over 15 min (% of mhz*NumCpuCores)	This monitor indicates the CPU running peak over 15 minutes.	Percent	No	NA
CPU running peak over 5 min (% of mhz*NumCpuCores)	This monitor indicates the CPU running peak over five minutes.	Percent	No	NA
Memory Active (% of MemorySize)	This monitor indicates the percentage of memory that is actively used, as estimated by VMkernel based on recently touched memory pages.	Percent	No	NA
Memory Balloon (% of MemorySize)	This monitor indicates the percentage of memory allocated by the virtual machine memory control driver (vmmemctl), which is installed with VMware Tools. This is the sum of all virtual machine values, plus vSphere services on the host, as a percentage of memory size.	Percent	No	NA
Memory Balloon Target (% of MemorySize)	This monitor indicates the target percentage value set by VMkernal for the virtual machine's memory balloon size.	Percent	No	NA
Memory Consumed (% of MemorySize)	This monitor indicates the percentage of memory that is consumed by a virtual machine, host, or cluster.	Percent	No	NA
Memory Granted (% of MemorySize)	This monitor indicates the percentage of machine memory or physical memory that is mapped for a virtual machine or a host.	Percent	No	NA

Metric Name	Description	Metric Unit	Linking Required ?	Metric to be linked with
Memory Heap (% of MemorySize)	This monitor indicates the memory heap (% of MemorySize).	Percent	No	NA
Memory Heap Free (% of MemorySize)	This monitor indicates the mmory heap free (% of MemorySize).	Percent	No	NA
Memory Reserved Capacity (% of MemorySize)	This monitor indicates the total amount of memory reservation used by powered-on virtual machines and vSphere services on the host.	Percent	No	NA
Memory Reserved Overhead (% of MemorySize)	This monitor indicates the memory (percent) reserved for use as the virtualization overhead.	Percent	No	NA
Memory used by vmkernel (% of MemorySize)	This monitor indicates the amount of machine memory used by VMkernel for core functionality, such as device drivers and other internal uses.	Percent	No	NA
Resource CPU Usage in MHz (% of mhz*NumCpuCores)	This monitor indicates the percentage of average CPU usage.	Percent	No	NA
CPU Capacity Entitlement	This monitor indicates the CPU resources devoted by the ESX scheduler to the virtual machines and resource pools.	Megahertz	Yes	CPU Capacity Usage
CPU Capacity Usage	This monitor indicates the CPU usage during the interval.	megahertz	Yes	CPU Capacity Entitlement
CPU Usage (% of CPUMaxUsage)	This monitor indicates the CPU usage as a percentage of maximum CPU usage.	Percent	No	NA
CPUOverallUsage (% of CPUMaxUsage)	This monitor indicates the real-time resource usage of all running child virtual machines, including virtual machines in child resource pools. This will be the percentage of current upper bound on usage.	Percent	No	NA

Metric Name	Description	Metric Unit	Linking Required ?	Metric to be linked with
MemoryOverallUsage (% of MemoryMaxUsage)	This monitor indicates the real-time resource usage of all running child virtual machines, including virtual machines in child resource pools. This will be the percentage of current upper bound on usage.	Percent	No	NA
Provisioned Space	This monitor indicates the provisioned space on this datastore, in gigabytes.	GBytes	Yes	Usage
Usage	This monitor indicates the used space on this datastore, in gigabtyes.	GB	Yes	Provisioned Space
CPU Usage	This monitor indicates the aggregated CPU usage of hosts.	MHz	Yes	TotalCPU
Memory Usage	This monitor indicates the aggregated memory usage of hosts.	MB	Yes	TotalMemory
TotalCPU	This monitor indicates the aggregated CPU resources of all hosts.	MHz	Yes	CPU Usage
TotalMemory	This monitor indicates the aggregated memory resources of all hosts.	MBytes	Yes	Memory Usage
CPU Usage (% of CPUMaxUsage)	This monitor indicates the current CPU usage as a percentage of maximum CPU usage.	Percent	No	NA
CPUMaxUsage	This monitor indicates the current upper-bound on usage. The upper-bound is based on the limit configured on this resource pool, as well as limits configured on any parent resource pool.	MHz	Yes	CPUOverallUsage
CPUOverallUsage	This monitor indicates the real-time resource usage of all running child virtual machines, including virtual machines in child resource pools.	MHz	Yes	CPUMaxUsage

Metric Name	Description	Metric Unit	Linking Required ?	Metric to be linked with
CPUOverallUsage (% of CPUMaxUsage)	This monitor indicates the real-time resource usage of all running child virtual machines, including virtual machines in child resource pools. This will be the percentage of current upper bound on usage.	Percent	No	NA
MemoryOverallUsage (% of MemoryMaxUsage)	This monitor indicates the real-time resource usage of all running child virtual machines, including virtual machines in child resource pools. This will be the percentage of current upper bound on usage.	Percent	No	NA
Capacity	This monitor indicates the maximum capacity of this datastore.	GBytes	Yes	Usage
Usage	This monitor indicates the used space on this datastore, in gigabtyes.	GB	Yes	Capacity

cdm Probe Metrics

Capacity Analytics uses metrics from the cdm probe.

Verify that the **cdm probe** is collecting the following metrics:

Metric Name	Metric ID	Description	Metric Unit	Linking Required ?	Metric to be linked with
Individual CPU System	QOS_CPU_MULTI_USAGE	The percentage of time when an individual CPU of the system was executing the kernel or operating system.	Percent	No	NA
Individual CPU Usage	QOS_CPU_MULTI_USAGE	The percentage of time for which an individual CPU of the system was used.	Percent	No	NA
Individual CPU User	QOS_CPU_MULTI_USAGE	The percentage of time for which an individual CPU of the system was executing in user mode.	Percent	No	NA

Metric Name	Metric ID	Description	Metric Unit	Linking Required ?	Metric to be linked with
Individual CPU Wait	QOS_CPU_MULTI_USAGE	The percentage of time for which an individual CPU of the system was waiting for I/O.	Percent	No	NA
Individual CPU Steal	QOS_CPU_MULTI_USAGE	The percentage of time for which an individual CPU of the system was waiting involuntarily. During this time, the CPU is not in any of the following states: User; System; Wait; Idle	Percent	No	NA
Total CPU System	QOS_CPU_USAGE	The sum of CPU time when all CPUs of the system were executing the kernel or operating system.	Percent	No	NA
Total CPU Usage	QOS_CPU_USAGE	The percentage of time for which all CPUs of the system were used.	Percent	No	NA
Total CPU User	QOS_CPU_USAGE	The percentage of time for which all CPUs of the system were executing in user mode.	Percent	No	NA
Total CPU Wait	QOS_CPU_USAGE	The percentage of time for which all CPUs of the system were waiting for I/O.	Percent	No	NA
Total CPU Steal	QOS_CPU_USAGE	The percentage of time for which all CPUs of the system were waiting involuntarily. During this time, the CPUs are not in any of the following states: User; System; Wait; Idle	Percent	No	NA

Metric Name	Metric ID	Description	Metric Unit	Linking Required ?	Metric to be linked with
Smt	QOS_CPU_SMT	The number of simultaneous multithreading (SMT) threads enabled in a logical partition (LPAR). If there are two SMT threads, the row is displayed as "on." However, if there are more than two SMT threads, the number of SMT threads is displayed.	Count	Yes	Number of CPU
Number of CPU	QOS_CPU_NUM	This QoS is supported only on the AIX platform.	Count	Yes	Smt
Disk Usage Change (MB)	QOS_DISK_DELTA	The disk usage change in megabytes	Megabytes	Yes	Disk Usage (MB)
Disk Usage (MB)	QOS_DISK_USAGE	Aggregated disk usage in megabytes	Megabytes	Yes	Disk Usage Change (MB)
Disk Usage (%)	QOS_DISK_USAGE_PERC	Aggregated disk usage in percentage	Percent	No	NA
Disk Read (B/s)	QOS_DISK_READ_THROUGHPUT	Disk bytes read per second	Bytes/Second	Yes	Disk Read and Write (B/s)
Disk Write (B/s)	QOS_DISK_WRITE_THROUGHPUT	Disk bytes written per second	Bytes/Second	Yes	Disk Read and Write (B/s)
Disk Read and Write (B/s)	QOS_DISK_TOTAL_THROUGHPUT	Disk bytes read and written per second	Bytes/Second	Yes	Disk Read (B/s) or Disk Write (B/s)
Inode Usage (%)	QOS_INODE_USAGE_PERC	The number of free file nodes in file system in percentage.	Percent	No	NA
Disk Latency (ms)	QOS_DISK_LATENCY	This metric is the total latency of the disk.	Milliseconds	Yes	Disk Read Latency (ms) or Disk Write Latency (ms)
Disk Read Latency (ms)	QOS_DISK_READ_LATENCY	This metric is the read latency of the disk.	Milliseconds	Yes	Disk Latency (ms)
Disk Write Latency (ms)	QOS_DISK_WRITE_LATENCY	This metric is the write latency of the disk.	Milliseconds	Yes	Disk Latency (ms)
Disk Usage Change (%)	QOS_DISK_DELTA_PERC	The disk usage change in percent	Percent	No	NA
Memory Usage (MB)	QOS_MEMORY_USAGE	Megabytes	Total memory usage in megabytes	Yes	Physical Memory (MB)
Memory Usage (%)	QOS_MEMORY_PERC_USAGE	Total memory usage in percentage	Total memory usage in percentage	No	NA

Metric Name	Metric ID	Description	Metric Unit	Linking Required ?	Metric to be linked with
Physical Memory (MB)	QOS_MEMORY_PHYSICAL_MB	Megabytes	The size of the physical memory used on the system in megabytes.	Yes	Memory Usage (MB)
Physical Memory (%)	QOS_MEMORY_PHYSICAL_PERC	Percent	The size of the physical memory used on the system in percentage	No	NA
Swap Memory (%)	QOS_MEMORY_SWAP_PERC	Percent	Total Swap memory usage in percent	No	NA
System Memory Utilization (%)	QOS_MEMORY_SYS_UTIL	Percent	Total System memory utilization in percent. This metric is supported only on the Windows, Linux and AIX platforms.	No	NA
User Memory Utilization (%)	QOS_MEMORY_USR_UTIL	Percent	Total User memory utilization in percent. This metric is supported only on the Windows, Linux and AIX platforms.	No	NA
Total Memory	QOS_TOTAL_MEMORY_MB	AME	The size of the total memory in an LPAR in your system, in megabytes.	Yes	Memory Dxm
Memory Dxm	QOS_MEMORY_EXPANDED_DXM_AME	AME	The size of the memory deficit in an LPAR in your system, in megabytes. At times an LPAR cannot be configured with the provided memory expansion factor as it is too large, and the workload in the LPAR does not compress well. Thus, a memory deficit is created.	Yes	Total Memory
Network Inbound Traffic	QOS_NETWORK_INBOUND_TRAFFIC	The total number of bytes per second received by the server.	Bytes/Second	Yes	Network Aggregated Traffic
Network Outbound Traffic	QOS_NETWORK_OUTBOUND_TRAFFIC	The total number of bytes per second sent by the server.	Bytes/Second	Yes	Network Aggregated Traffic
Network Aggregated Traffic	QOS_NETWORK_AGGREGATED_TRAFFIC	The total number of bytes per second sent and received by the server.	Bytes/Second	Yes	Network Inbound Traffic or Network Outbound Traffic

Metric Name	Metric ID	Description	Metric Unit	Linking Required ?	Metric to be linked with
Interface Errors % (In)	QOS_INBOUND_ERRORS_PERC	Inbound errors in percent.	Percent	No	NA
Interface Errors % (Out)	QOS_OUTBOUND_ERRORS_PERC	Outbound errors in percent.	Percent	No	NA
Interface Discards % (In)	QOS_INBOUND_DISCARDS_PERC	Inbound discards in percent. This metric is not supported on the Solaris platform.	Percent	No	NA
Interface Discards % (Out)	QOS_OUTBOUND_DISCARDS_PERC	Outbound discards in percent. This metric is not supported on the Solaris platform.	Percent	No	NA
Interface Utilization % (In)	QOS_INBOUND_UTILIZATION_PERC	Inbound interface utilization in percent. This metric is not supported on the AIX platform.	Percent	No	NA
Interface Utilization % (Out)	QOS_OUTBOUND_UTILIZATION_PERC	Outbound interface utilization in percent. This metric is not supported on the AIX platform.	Percent	No	NA

ibmvm Probe

Capacity Analytics uses QoS metrics from the ibvm probe.

If you are using the **ibmvm** probe, activate the following QoS metrics.

Metric Name	Metric ID	Description	Metric Unit	Linking Required ?	Metric to be linked with
Percent Used	RESOURCE_MEM	HMC percent memory used.	Percent	No	NA
Total	RESOURCE_MEM	HMC total memory.	Megabytes	Yes	Used
Used	RESOURCE_MEM	HMC used memory.	Megabytes	Yes	Total
Cached	RESOURCE_SWAP	HMC swapspace cached.	Megabytes	No	NA
Percent Used	RESOURCE_SWAP	HMC percent swap used.	Percent	No	NA
Total	RESOURCE_SWAP	HMC total swap capacity.	Megabytes	Yes	Used
Used	RESOURCE_SWAP	HMC used swap.	Megabytes	Yes	Total
I/O Waiting	RESOURCE_CPU	HMC CPU I/O waiting percentage.	Percent	No	NA
Percent Used	RESOURCE_DISK	Percent usage for this HMC filesystem.	Percent	No	NA

Metric Name	Metric ID	Description	Metric Unit	Linking Required ?	Metric to be linked with
Total	RESOURCE_DISK	Total available capacity of the HMC disk.	Megabytes	Yes	Used
Used	RESOURCE_DISK	Used filesystem space on the HMC disk.	Megabytes	Yes	Total
VM Total	HOST	The current VM count.	Number	Yes	VMs Stopped
VMs Stopped	HOST	Number of stopped VMs.	Number	Yes	VM Total
Assigned Memory	HOST_MEMORY	Amount of memory assigned to VMs. This figure is equal to the "Installed Memory" minus "Unassigned Memory".	Megabytes	Yes	Installed Memory
Installed Memory	HOST_MEMORY	Total amount of memory installed on the managed system.	Megabytes	Yes	Assigned Memory
CoD Processors Activated	HOST_COD	The total number of processors activated for CoD.	Count	Yes	CoD Processors Available
CoD Processors Available	HOST_COD	The number of processors available for CoD.	Count	Yes	CoD Processors Activated
CoD Processors Requested Days Available	HOST_COD	The total number of days for CoD processor requests.	Days	Yes	CoD Processors Requested Days Left
CoD Processors Requested Days Left	HOST_COD	The remaining number of days available for CoD processor requests.	Days	Yes	CoD Processors Requested Days Available
Disk Bandwidth Used (%)	HOST_DISK	Network interface description.	Percent	No	NA
Global Shared Processor Pool Utilization	CPU_POOL	The average global shared processor pool utilization.	Percent	No	NA
Shared Processor Pool Utilization	CPU_MSPP	The average shared processor pool utilization.	Percent	No	NA
Storage Pool Utilization	SR	Percentage of storage space used.	Percent	No	NA
Average CPU Utilization	VIO	The average CPU utilization of this VIO Server.	Percent	No	NA
Physical Processors Consumed	VIO_CPU	The number of physical processors used by the partition	Processing Units	Yes	Current Processing Units

Metric Name	Metric ID	Description	Metric Unit	Linking Required ?	Metric to be linked with
Processor Entitlement Consumed	VIO_CPU	Percentage of the entitled processing cycles used by the partition.	Percent	No	NA
Disk Bandwidth Used (%)	VIO_DISK	Percent bandwidth used for each disk.	Percent	No	NA
Interfaces UP	VIO_NIC_GROUP	Count of interfaces up.	Number	Yes	Interfaces DOWN
Assigned Processors	VM_CPU	Number of processors or virtual processors that are varied on for the partition.	Processing Units	Yes	Physical Processors Consumed
Physical Processors Consumed	VM_CPU	The number of physical processors used by the partition.	Processing Units	Yes	Assigned Processors
Processor Entitlement Consumed	VM_CPU	Percentage of the entitled processing cycles used by the partition.	Percent	No	NA
Assigned Memory	VM_MEMORY	Amount of memory assigned to the partition.	Megabytes	Yes	Used Memory
Used Memory	VM_MEMORY	Memory used by the partition at the time of sample.	Megabytes	Yes	Assigned memory
Disk Bandwidth Used (%)	VM_DISK	Percent bandwidth used for each disk.	Percent	No	NA

hpe_3par Probe

If you are using the **hpe_3par** probe, activate the following QoS metrics.

Metric Name	Metric ID	Description	Metric Unit	Linking Required ?	Metric to be linked with
Total Capacity	QOS_HP_3PAR_RAW_TOTAL_CAPACITY	The total raw storage capacity (in gigabytes) of the StoreServ system.	GB	Yes	Used Capacity
Used Capacity	QOS_HP_3PAR_RAW_USED_CAPACITY	The used raw storage capacity (in gigabytes) of the StoreServ system.	GB	Yes	Total Capacity
WriteCacheHit	QOS_STORAGE_NODE_WRITE_CACHE_PC	Write cache hit ratio	Percent	No	NA
ReadCacheHit	QOS_STORAGE_NODE_READ_CACHE_PC	Read cache hit ratio	Percent	No	NA
CPU Utilization	QOS_CONTROLLER_CPU_UTILIZATION	The total CPU utilization of the controller node.	Percent	No	NA

Metric Name	Metric ID	Description	Metric Unit	Linking Required ?	Metric to be linked with
Utilization	QOS_CPG_UTILIZATION	The percentage of CPG capacity that is currently in use.	Percent	No	NA
Utilization	QOS_LOGICAL_DISK_UTILIZATION	The percentage of used capacity of the logical disk.	Percent	No	NA
Utilization	QOS_VIRTUAL_VOLUME_UTILIZATION	The percentage of used capacity of the virtual volume.	Percent	No	NA

netap ontap Probe Metrics

Capacity Analytics uses QoS metrics from the netap-ontap probe.

If you are using the **netap-ontap** probe, activate the following QoS metrics.

Metric Name	Metric ID	Description	Metric Unit	Linking Required?	Metric to be linked with
Total Capacity	QOS_STORAGE_TOTAL_CAPACITY_AGGR	The total capacity of the referenced file system.	GB	Yes	Used Capacity
Used Capacity	QOS_STORAGE_USED_CAPACITY_AGGR	The used capacity of the referenced file system.	GB	Yes	Total Capacity
Used Capacity Percent	QOS_STORAGE_PERCENT_CAPACITY_USED_AGGR	The used capacity in percentage of the referenced file system.	Percent	No	NA
Snapshot Total Capacity	QOS_STORAGE_TOTAL_SNAPSHOT_CAPACITY_AGGR	The total capacity for the snapshot on the referenced file system.	GB	Yes	Snapshot Used Capacity
Snapshot Used Capacity	QOS_STORAGE_SNAPSHOT_USED_CAPACITY_AGGR	The used capacity for the snapshot on the referenced file system.	GB	Yes	Snapshot Total Capacity
Snapshot Percent Used Capacity	QOS_STORAGE_PERCENT_SNAPSHOT_USED_AGGR	The used disk space for the snapshot in percentage on the referenced file system.	Percent	Yes	NA
Total Disk Count	QOS_STORAGE_TOTAL_DISK_COUNT	The total number of disks on the system.	Count	Yes	Failed Disk Count
Failed Disk Count	QOS_STORAGE_NUMBER_OF_FAILED_DISKS	The number of disks that are currently broken.	Count	Yes	Total Disk Count
Disk Utilization	QOS_STORAGE_DISK_UTILIZATION	The disk utilization percentage on the referenced disk drive.	Percent	No	NA

Metric Name	Metric ID	Description	Metric Unit	Linking Required?	Metric to be linked with
Average Latency	QOS_STORAGE_TOTAL_AVERAGE_LATENCY	The average latency for all operations in the system in milliseconds.	ms	Yes	Average Other Latency, Average Read Latency, Average Write Latency
Average Other Latency	QOS_STORAGE_AVERAGE_OTHER_LATENCY	The average latency for all other operations in the system in milliseconds.	ms	Yes	Average Latency
Average Read Latency	QOS_STORAGE_AVERAGE_READ_LATENCY	The average latency for all read operations in the system in milliseconds.	ms	Yes	Average Latency
Average Write Latency	QOS_STORAGE_AVERAGE_WRITE_LATENCY	The average latency for all write operations in the system in milliseconds.	ms	Yes	Average Latency
CPU Utilization	QOS_STORAGE_CPU_UTILIZATION	The percent of time that the CPU has been doing useful work since the last time a client requested the CPU Busy Time Percent.	Percent	No	NA
Total Read IOPS	QOS_STORAGE_READ_IOPS	The total read operations per second.	count/s	Yes	System IOPS
Total Write IOPS	QOS_STORAGE_WRITE_IOPS	The total write operations per second.	count/s	Yes	System IOPS
System IOPS	QOS_STORAGE_SYSTEM_IOPS	The total system operations per second.	count/s	Yes	Total Read IOPS, Total Write IOPS
LUN Percent Used	QOS_STORAGE_LUN_PERCENT_USED	The percent utilization for the LUN.	Percent	No	NA
LUN Size	QOS_STORAGE_LUN_TOTAL_SIZE	The total size of the LUN.	GB	Yes	LUN Size Used
LUN Size Used	QOS_STORAGE_LUN_SIZE_USED	The size used in GB that is in use on the LUN.	GB	Yes	LUN Size
Qtree Quota	QOS_STORAGE_QUOTA_QTREE	The quota for growth of the Qtree in GB.	GB	Yes	Qtree Space Used
Qtree Percent Used	QOS_STORAGE_PERCENT_USED_QTREE	Percent tagged space used in the Qtree.	Percent	No	NA
Qtree Space Used	QOS_STORAGE_SPACE_USED_QTREE	The storage space used by the Qtree in GB.	GB	Yes	Qtree Quota

Metric Name	Metric ID	Description	Metric Unit	Linking Required?	Metric to be linked with
Total Capacity	QOS_STORAGE_TOTAL_CAPACITY_VOL	The total capacity of the volume.	GB	Yes	Used Capacity
Used Capacity	QOS_STORAGE_USED_CAPACITY_VOL	The used capacity of the volume.	GB	Yes	Total Capacity
Total Snapshot Capacity	QOS_STORAGE_TOTAL_SNAPSHOT_CAPACITY_VOL	The total capacity for the snapshot in the volume.	GB	Yes	Snapshot Used Capacity
Snapshot Used Capacity	QOS_STORAGE_SNAPSHOT_USED_CAPACITY_VOL	The used capacity for the snapshot in the volume.	GB	Yes	Total Snapshot Capacity

Universal Monitoring Agent for Kubernetes

Capacity Analytics uses metrics from the Universal Monitoring Agent for Kubernetes.

Metric Name	Metric Type	Metric Unit	Linking Required ?	Metric to Be linked with	Description
CPU Utilization (mCore)	Usage	Count	Yes	CPU Limit (mCore)	Total CPU utilized by Node/POD/Cluster
CPU Limit (mCore)	Capacity	Count	Yes	CPU Utilization (mCore)	Total CPU available for Kubernetes Node/POD/ Cluster
Memory Usage (Bytes)	Usage	Bytes	Yes	Memory Limit (Bytes)	Total Memory Utilized by Node/POD/Cluster
Memory Limit (Bytes)	Capacity	Bytes	Yes	Memory Usage (Bytes)	Total Memory available for Kubernetes Node/POD/ Cluster

Services Configuration

After you complete the tasks mentioned in the Prerequisites, you must configure Services and the associated metrics in Capacity Analytics to view the Services Capacity Analytics Data.

Complete the following tasks to configure services and the associated metrics for Capacity Analytics:

- Configure Services
- Configure Associated Metrics

{"URL":["https://digital-oi/capacity-analytics/cpaservicesettings"],"description":"concept.dita_28ee49fb-fe90-432c-9dbe-efc2ac61f027","new":"","new_video":"","admin":"","troubleshooting":{"masterkb":"","text":"","URL":[]},"pendo":"","video":[]}

Configure Services in Capacity Analytics

You must configure Services to view Capacity Analytics data for those Services. The Services are populated from service hierarchies in the Services Analytics view.

{"URL":["https://digital-oi/capacity-analytics/cpaservicesettings"],"customLabelGetStarted":"Configure Services","description":"task.dita_1a28a624-ca75-4057-8936-896fa9867229"}

NOTE

- Ensure that you add a service before you get into the service configuration process. For more information on how to create a service, see Create a Service. After you add a Service, it takes 30 seconds to appear in the **Capacity Analytics** Homepage.
- Capacity Analytics displays alarms that are going to breach the threshold limit within the next 90 days. By default, all services are monitored for the threshold breach within the next 90 days.

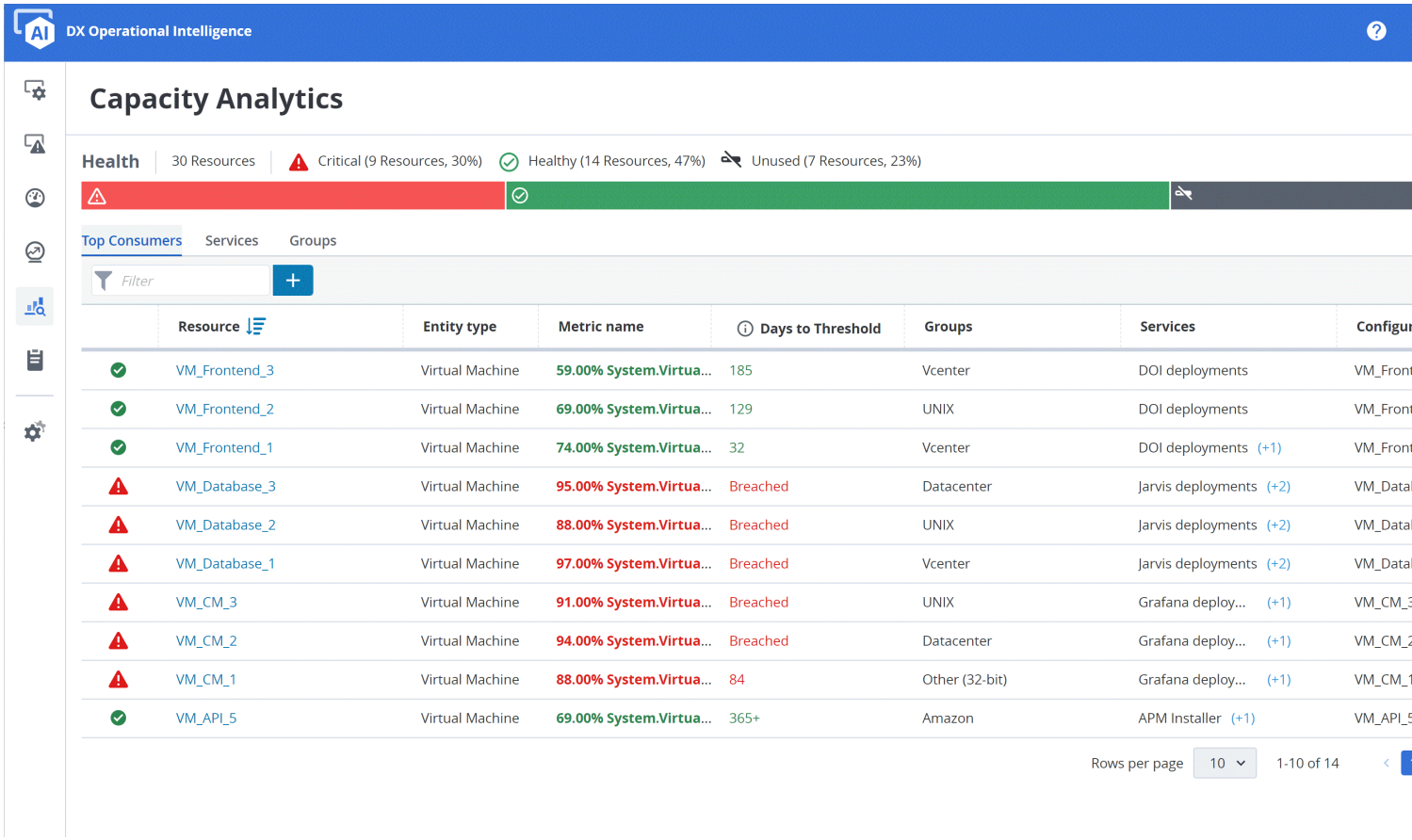
To configure the Services, follow these steps:

1. In DX Digital Operational Intelligence, click in the left navigation pane.

   The application displays the Capacity Analytics home page.

2. Click icon on the top-right of the screen, and select the **Configure Services** option.

   The application displays the Services Configuration page.

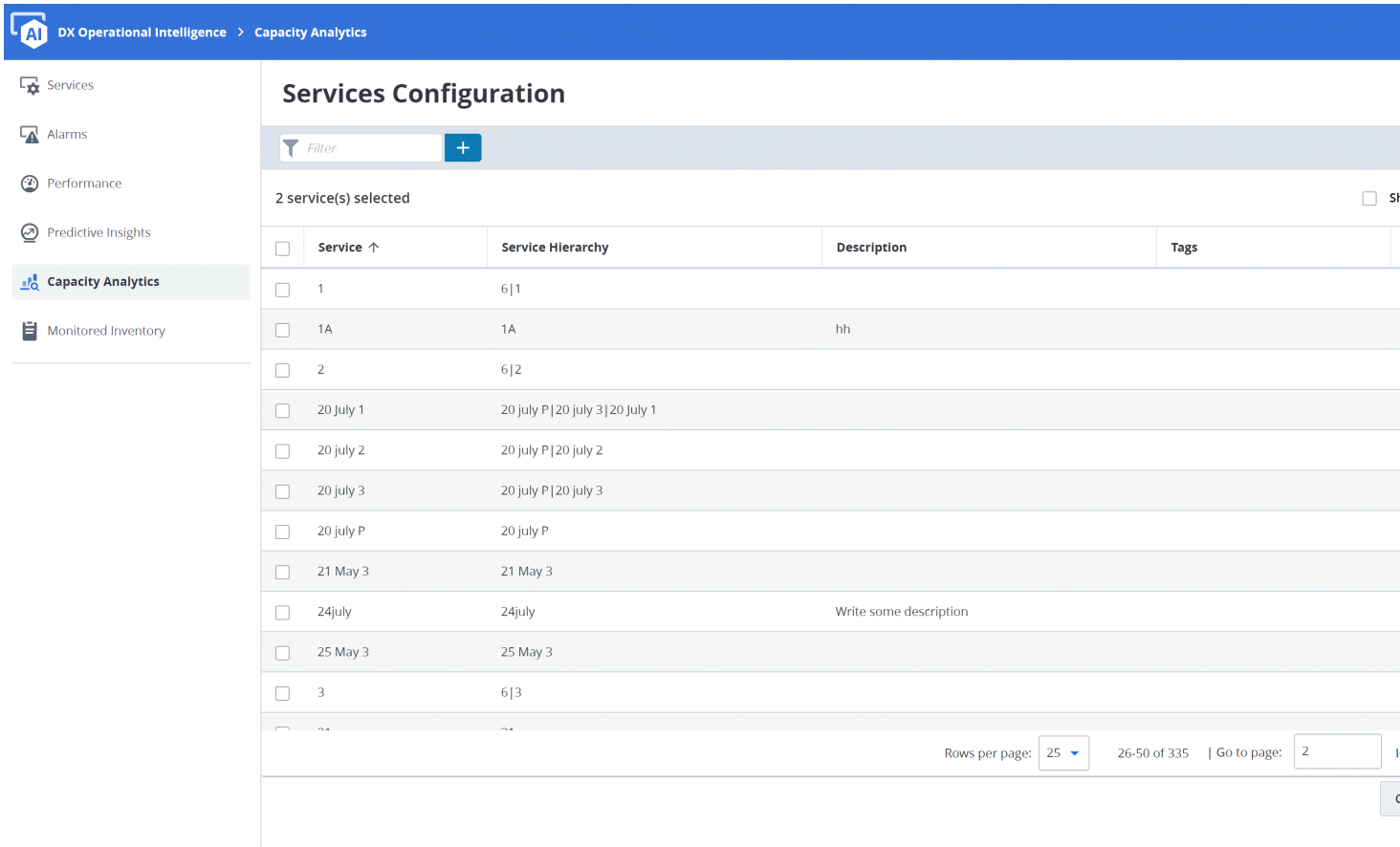


3. Select the Services that you want to configure.

   The Service names get populated as configured in the source products. The **Configuration** view displays the following details:

- Service: Indicates the name of the service.
- Service Hierarchy: Indicates the hierarchy name of the service.
- Description: Indicates the details of the service.
- Tags: Indicates the tag name that is defined for the service.
- Location: Indicates the location that is defined for the service. For example, Australia.

4. View and filter the services using the following options:
   - **Filter**: Enables you to search for services using a filter criteria. You can filter the services based on the location, service name, and the tags associated with services.
   - **Pagination**: Enables you to navigate between the paginated Service data using the Next, Previous, First, and the Last page links. You can also specify the rows that you want to view per page.



5. To view only the selected configurations, click the **Show Selected** check box on the top-right.
6. After you select the required configurations, click **Next**.

   NOTE
   If you do not perform this step, the application does not save the selected configurations.

   The application stores the selected services and redirects you to the **Metric Configuration** view forMetric Configuration.

Groups Configuration

After you followed the prerequisites, you need to configure Groups in Capacity Analytics to view the Groups Capacity Analytics Data.

You must configure Groups in order to view Capacity Analytics data for those Groups. The Groups are populated from data sources (DX IM, DX NetOps for Performance Management, and third-party data sources).

Complete the following tasks to configure services and the associated metrics for Capacity Analytics:

- Configure Group for Capacity Analytics
- Configure Metrics for Capacity Analysis

{"URL":["https://digital-oi/capacity-analytics/cpagroupsettings"],"customLabelGetStarted":"Groups Configuration","description":"task.dita_85db2497-fe58-4cee-8626-6f7a690ead51","customCards":[{"type":"configure","id":"task.dita_85db2497-fe58-4cee-8626-6f7a690ead51","url":"https://techdocs.broadcom.com/us/en/ca-enterprise-software/it-operations-management/dx-operational-intelligence-saas/SaaS/using/capacity-analytics/configure-rbac-and-groups/Configure-Groups-for-Capacity-Analytics.html","title":"Configure Groups"}]}

Configure Groups for Capacity Analytics

You must configure Groups in order to view Capacity Analytics data for those Groups.

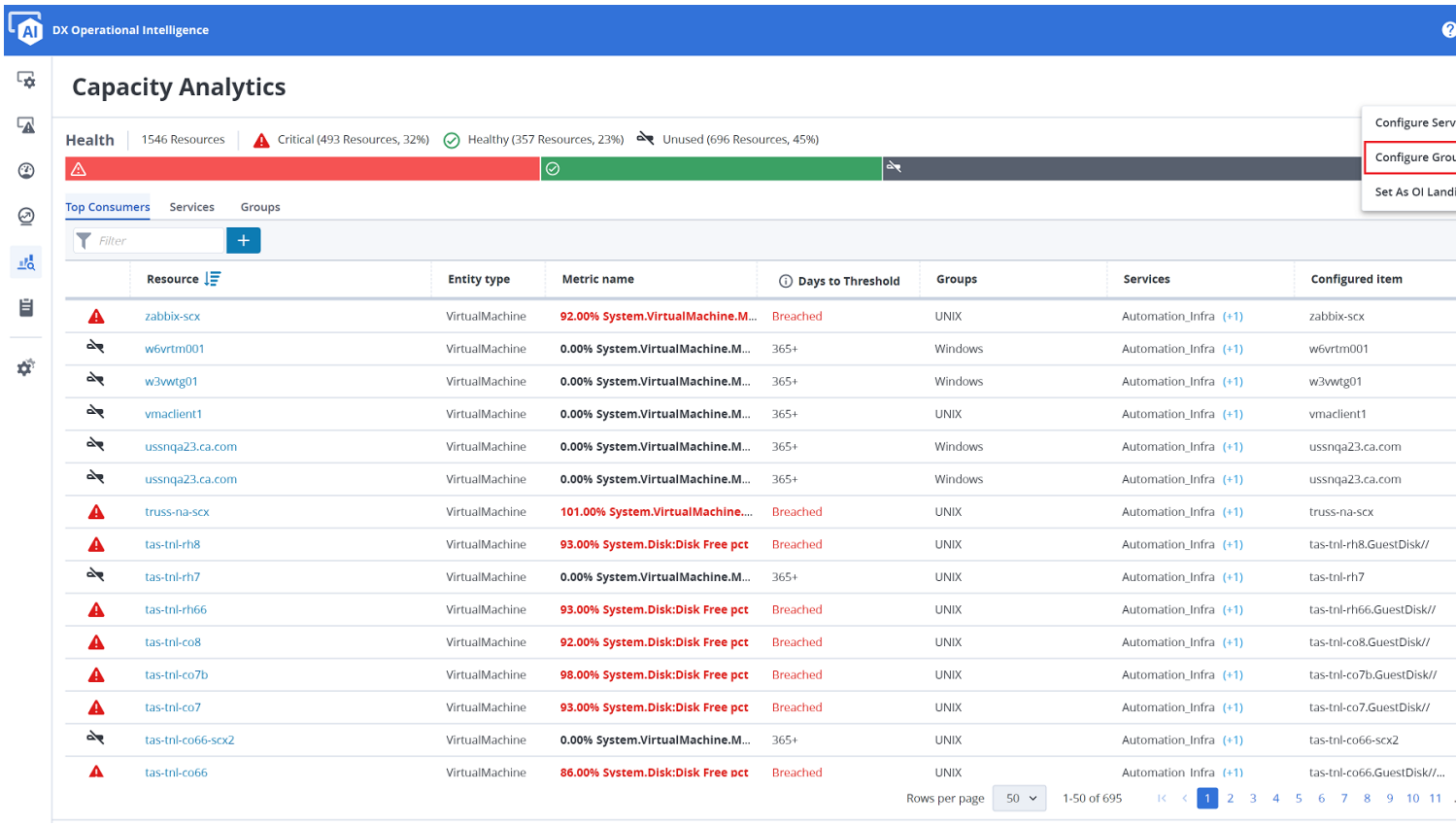
The Groups are populated from data sources (DX IM, DX NetOps for Performance Management, and third-party data sources).

NOTE

The resources that are deleted in UIM get reflected after 24 hours in the Capacity Analytics Configuration view.

To configure Groups, follow these steps:

1. In DX Digital Operational Intelligence, click in the left navigation pane.
   The application displays the Capacity Analytics home page.
2. Click icon on the top-right of the page, and click **Configure Group** option.
   The Group Configuration page appears.



3. Select the group name that you want to configure.

Monitored Technologies

4. Select monitored technologies for the selected group in the Monitored Technology column. You can also create custom monitored technologies by entering the name and clicking the **+** icon.

   NOTE
   Ensure that the Group contains the resources of the same monitored technologies type.

   The application maps the selected monitored technologies to a category in the Monitored Technologies dashboard.

   The Role field of the selected group populates the role that is associated with the selected Monitored Technologies in the Role column.

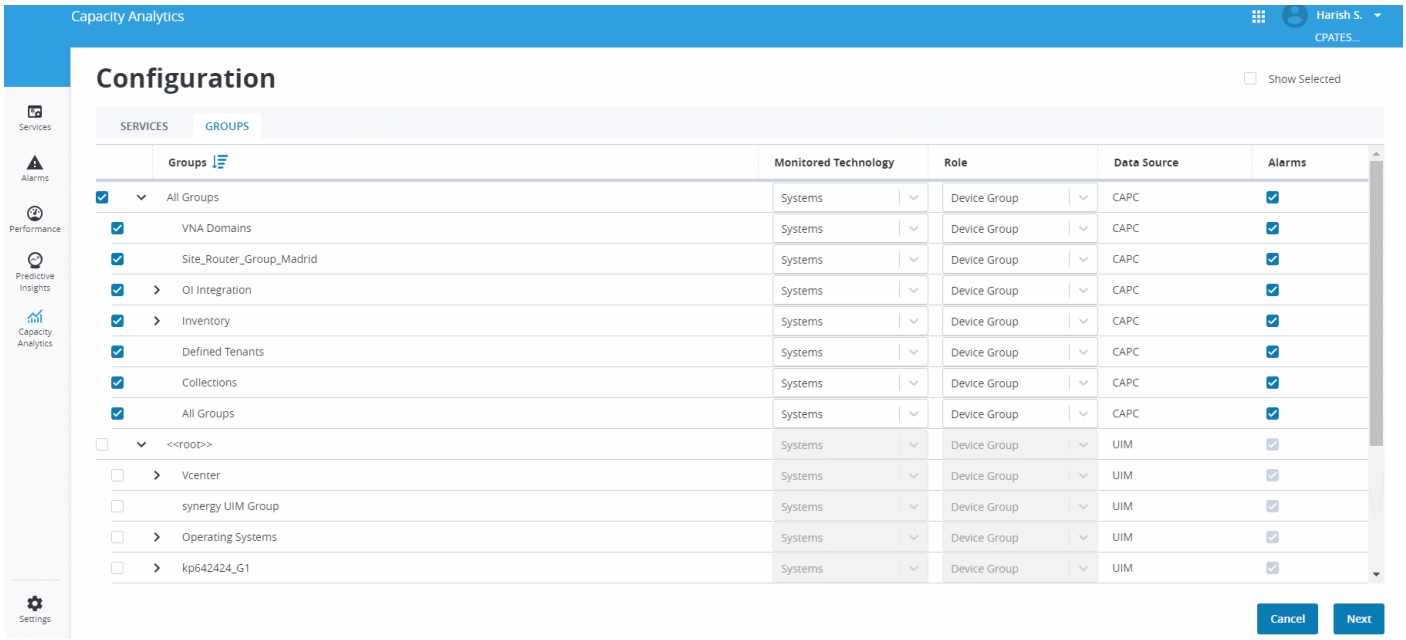
   NOTE
   The Group configuration that is applied to the parent level gets inherited from its child.

Alarm

5. Select Alarm checkbox in the Alarm column, if you want to enable alarms for the selected group.

   NOTE

   - You can enable alarms only for an individual group. By default, all groups are enabled.
   - Enabling alarm for a parent group does not enable the alarm of its child group.



The application generates the alarms for the devices that meet the threshold value. If the threshold day is less than 90 days, the prediction alarm is generated for a device on the Predictive Insights view.

6. After you select the required configurations, click **Next**.

   NOTE
   The application does not save the group configuration updates, if you do not navigate to the metric configuration step.

   The application stores the selected groups and redirects you to the **Metric Configuration** view for Metric Configuration.

Metrics Configuration

You can configure the metrics for capacity analytics in the **Metric Configuration** view.

The Metric Configuration view displays a summary (utilization and capacity) of the metric configurations that are based on the Group and Services that you select on the Capacity Analytics **Configurations** view.

NOTE
While configuring metrics for a service or group, ensure that each device is configured to monitor only one source product.

This section provides the list of tasks that you must perform to configure metrics for capacity analytics.

Metric Configuration View

The **Metric Configuration** view displays a summary (utilization and capacity) of the metric configurations that are based on the Group and Services that you select on the Capacity Analytics **Configurations** view. The page shows the total number of Groups and Services that are selected and their corresponding metrics. You must **link two metrics** to monitor the data in Capacity Analytics.

DX Operational Intelligence displays the following information for each metric:

Metric name Displays the metric name.

Display name Displays the custom display name of the metric. You can customize the display name as per your requirements.

Data Source Displays the data source name (Product) of the metric.

Context Displays the context details of the metric.

CI Type Displays the Configuration Item type details.

- Displaying the CI type eases the process of viewing, understanding, and identifying the entity or the sub-entity type for which you are configuring the metrics. This information is helpful in scenarios where you use RESTmon to collect such metrics.
- When you monitor the storage or network (for example, entity/sub-entity) the same metrics are collected for the storage entity and also for the various disks that are part of the storage as sub-entities. The availability of the CI type details during the metric configuration for Capacity Analytics enables you to make an informed decision.

Enabled Indicates whether the metric is enabled.

Threshold Displays the threshold value of the metric. You can define or edit the threshold value.

Metric unit Displays the metric unit. You can edit the metric unit.

Actions

Displays the (Edit Icon) that enables you to perform various actions on the metrics. The actions that you can perform on metrics are:

- Define the threshold value
- Edit the display name
- Edit the metric unit.

Filter Metrics in Metric Configuration View

DX Operational Intelligence supports following options to filter metrics in the Metric Configuration view:

Global Search Filter

The Global Search filter

enables you to search for metrics in the **Metric Configuration** table. Enter your search text to view the metrics that match your search text. To filter the metrics in the table based on the service context values, enter **context:<value>** in the **Filter** field.

Filter by Attribute The **Attributes Filter** field enables you to view the metrics that have one or more attributes specified in the filter criteria.

Metric Configuration supports the following attributes as filters:

- **CI Type**
- **Context**
- **Metric Family**
- **Metric Name**
- **Source Product**

To filter Metrics using filter attributes option, follow these steps:

1. Click the icon in the **Attributes Filter**.
2. Select a filter attribute.
3. Select the appropriate filter operator and specify the value.
4. Click **Add**.
   For example, You can view the metrics from **CAPM** source product by selecting the attribute as **Source Product**, operator as **Equals** and the value as **CAPM**. The Metric Configuration table and Insights show only the metrics that match your search criteria for the selected attributes.

   The application filters the metrics that match the defined filter criteria.

Enable Metrics for Capacity Analytics

You can enable metrics in Metric Configuration view for capacity Analytics.

IMPORTANT

- For DX Dashboards, you can enable all metrics with or without metric units assigned to them. We recommend that use the DX Dashboards to view projections for the metrics that are non-linked or having a non-percentage (pct) unit and are enabled for Capacity Analytics.
- For the Capacity Analytics User Interface, you must enable all metrics that have the metric unit as a percentage (pct) or linked metrics. We recommend that you have a metric unit assigned to all the metrics.

To enable the metrics for capacity analytics, turn on the toggle in the **Enabled** column for the metric that you want to enable.

Define or Update Metric Units

You must provide a metric unit to the metric before you can enable capacity planning for the metric. If a unit is not provided by the source product, you need to provide a unit for that metric. For single metrics, ensure that the metric unit provided is **pct**. For linked metrics, you can provide any metric unit.

1. Click in the Actions column of a metric for which you want to define metric unit.
   The application displays the Metric Unit column for a metric in Edit mode.
2. Specify the metric unit in the Metric Unit column.
3. Click Save.
   The application updates the metric unit for a metric.

Define Threshold for Metrics

You can define the threshold values for a metric in the Metric Configuration View. Based on the defined threshold value, the metrics are categorized as Critical, Major, Minor, Information, and Unused in the **Resource Capacity Status** dashboard on the **Capacity Analytics** view.

To define the threshold values for metrics, follow these steps:

1. Click in the Actions column of a metric for which you want to define threshold value.
   The application displays threshold field for metric in Edit mode.
2. Select the Threshold value from the drop-down and click
   ⚙
   in the Threshold column.
   The Threshold Value for CPA Usage pop-up appears.
3. Select the threshold Value for Critical in the Critical field.
4. Click Save.
   The application defines the threshold value.

Link Metrics

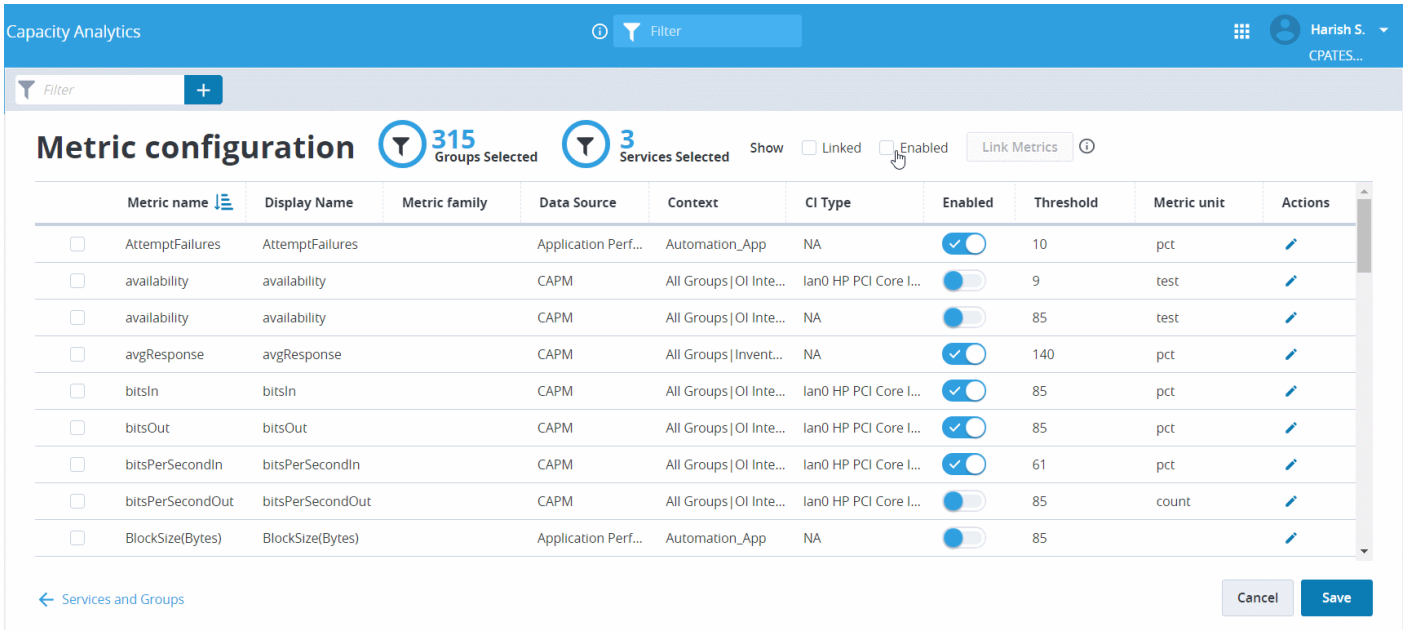
DX Operational Intelligence supports linking of two (2) metrics in the Metric Configuration View. You can link two metrics only when the following conditions are met:

- The selected metrics are from the same product and the metric family, and have the same metric units.
- The selected metrics are **Enabled**.

  NOTE
  You are not allowed to link two metrics that have percentage (pct) metric units.

1. In the **Metrics configuration** view, select two metrics by clicking the respective checkboxes.
   Ensure that the metrics belong to the same metric family and product.

2. Turn on the toggle in the **Enabled** column to enable the selected metrics for capacity analytics.



3. Select **Enabled** option to filter and view the metrics that are in enabled.
4. Click in the Actions column and change the metric units for the selected metrics in the respective Metric Unit column.

   Ensure the metric unit is same for the selected metrics.

   The application enables the Link Metrics option when the selected metrics are from the same product and metric family and have same metric units.
5. Click **Link Metrics** to link the selected metrics.

   The **Link Metrics** pop-up appears.
6. Select the **Total Metric** value and the**Utilization Metric** value from the respective field drop-down.
7. Click **Link**.

   The application links the selected metrics.
8. Click **Save**.

   The application saves the linked metrics.
9. Select **Show Linked** checkbox to view the linked metrics in the Metric Configuration View.

De-link Linked Metrics

You can de-link the linked metrics in Metric Configuration View.

To de-link the linked metrics, follow these steps:

1. In the Metrics Configuration View, select **Show Linked** view.

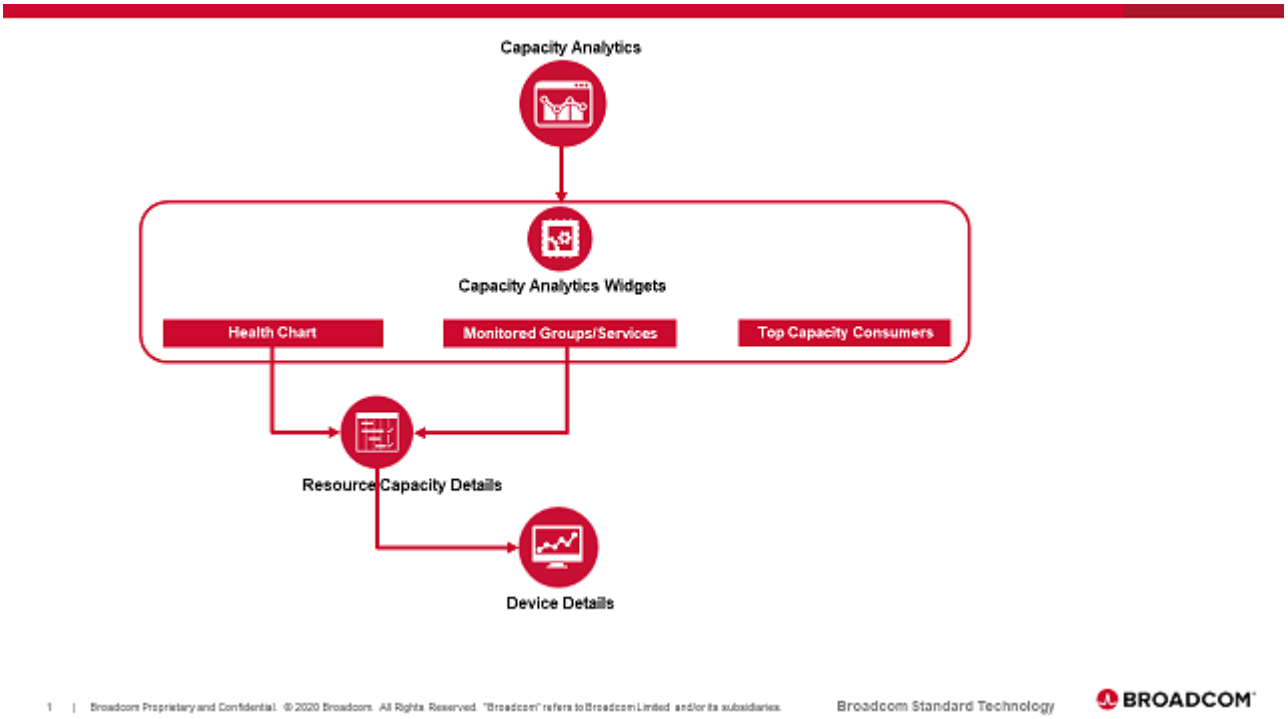
   The application displays the linked-metrics list and enables the De-Link option.

2. Select the two (2) linked metrics that you want to de-link.
3. Click the De-Link Metrics option.
   The application de-links the selected linked metrics.

Navigating Capacity Analytics

Capacity Analytics provides access to various views to analyze and optimize the resource capacity and utilization.

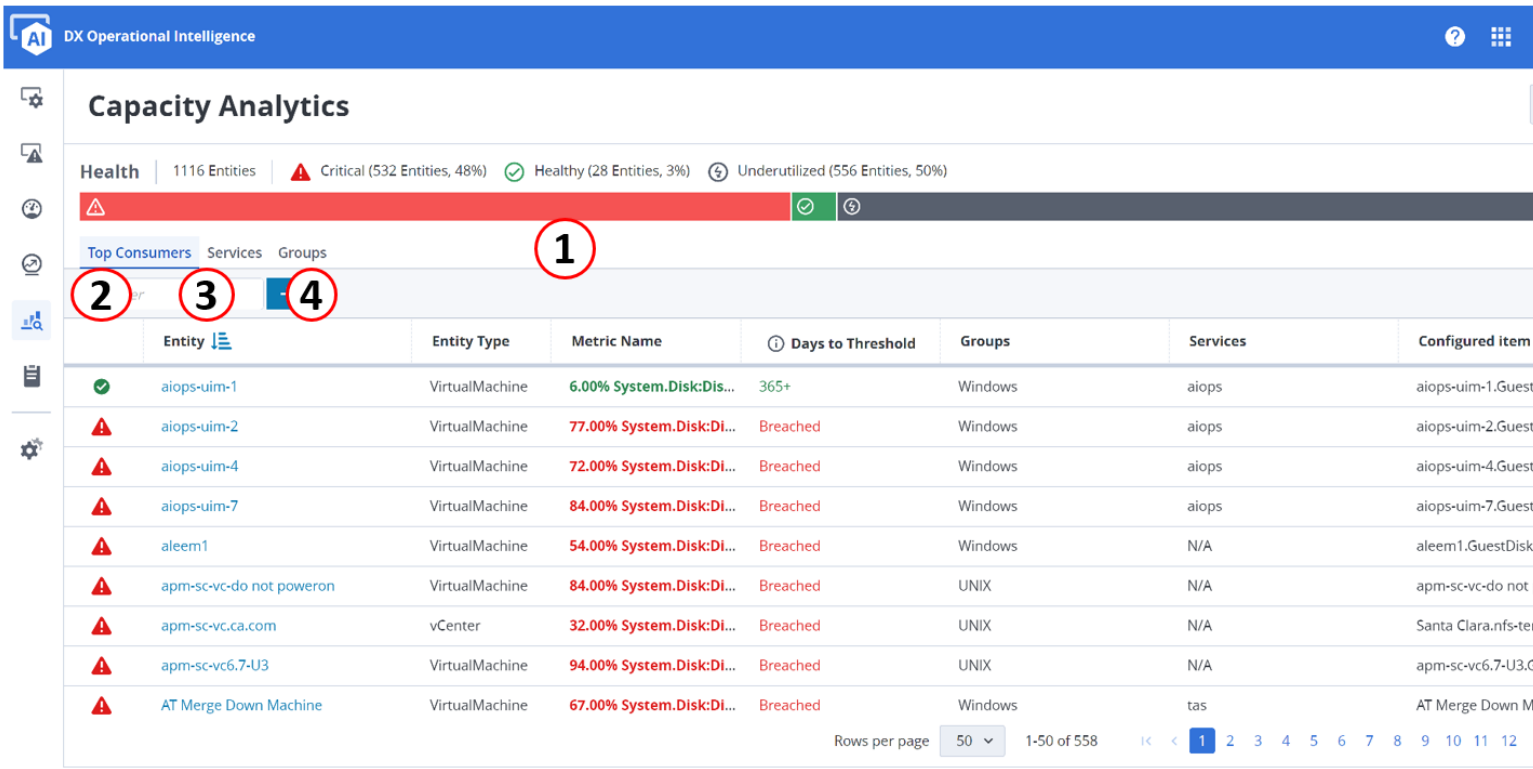
The following flowchart illustrates the information that you can view and access in Capacity Analytics:



Capacity Analytics Overview Page

Use the Capacity Analytics Overview page to view the overall health of the resources.

You can use the Capacity Analytics Overview page to identify the resources that are at risk and can exceed utilization thresholds. You can click the following tabs and charts to view and analyze the current capacity utilization of the resources and generate the capacity forecasts.



- **1**: Health Chart: Displays the health of your resources based on their current utilization.
- **2**: Top Capacity Consumers Tab: Displays the list of top capacity consumers.
- **3 & 4**: Services Tab and Groups Tab: Displays the configured groups and services in the respective tabs.

NOTE

- The prediction data that appears on the Capacity Analytics homepage is based on the selected time period under **Top Capacity Consumers** view.
- Make sure that you have the data for at least 3 hours in NASS store to view data in Capacity Analytics.

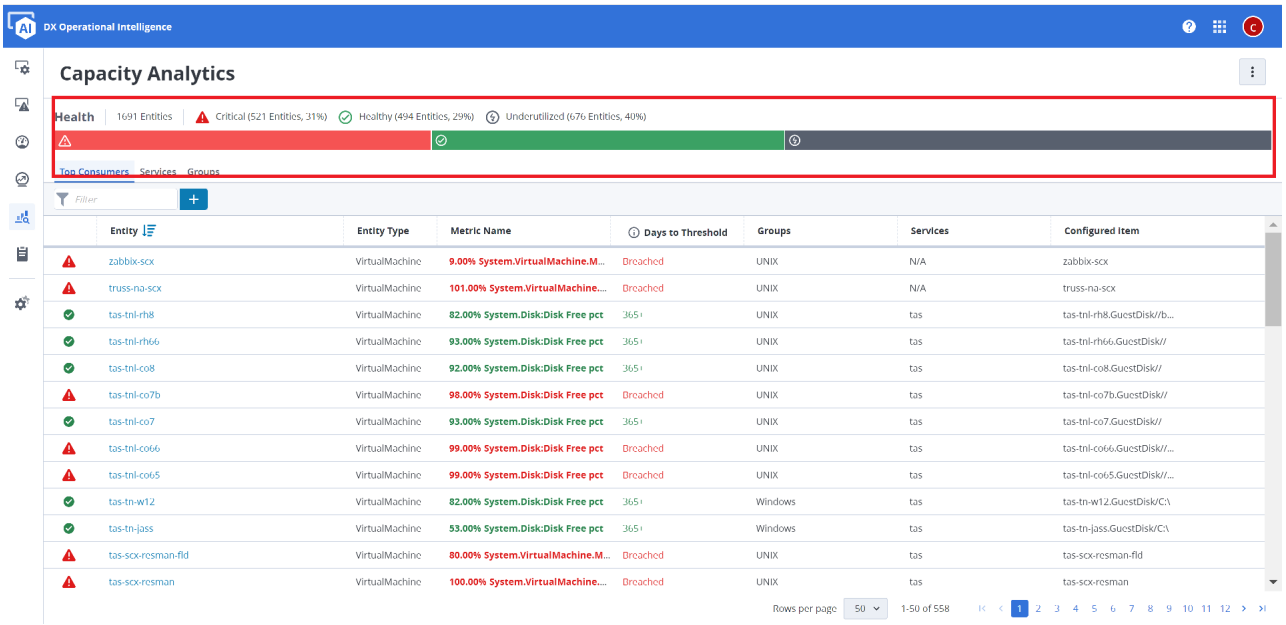
Health Chart

Using the Health Chart, you can analyze the health of the resources in terms of utilization.

The chart displays the health of your resources based on their utilization. This view categorizes the resources as per the severity levels defined for resource utilization and shows them as Critical, Healthy, and Underutilized Entities used using color codes:

As a platform administrator, you can perform the following actions:

- View the overall health of resources at a high level based on the severity level. DX Operational Intelligence categorizes the severity using color codes.
- Drill down and view the list of resources associated with a selected severity category in the Resource Health view.
- Use the 3-month actual and 6-month projection chart for each resource to analyze the resource utilization and their threshold levels. Apply the attributes and filters for in-depth analysis.
- Identify the overutilized, healthy, and underutilized resources, and take the necessary actions to resize and optimize resource utilization.



Access Resource Health View

Using the Resource Health View, you can view and analyze the current capacity status and generate short, medium and long-term capacity projections of the resources.

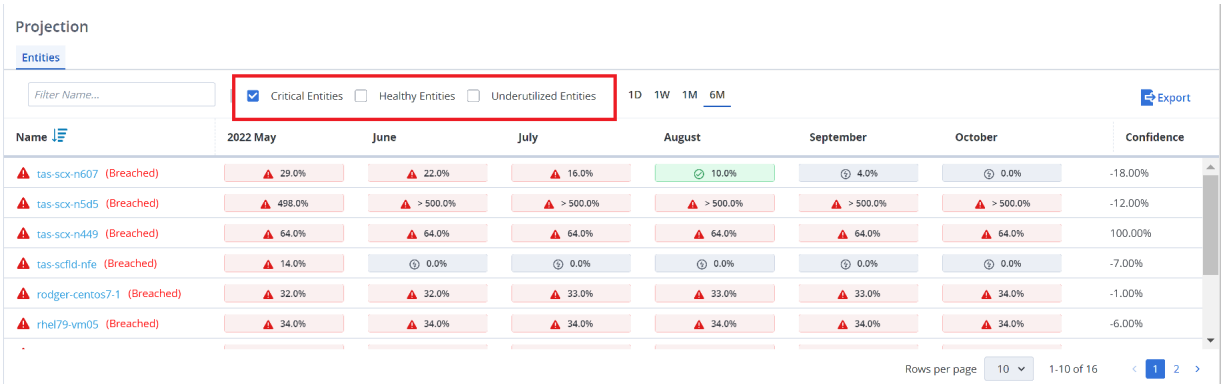
To view the resource health, follow these steps:

1. In the **Health Chart**, click the severity bar that you want to view.

   The application displays the Resources Health view with 6-Month projections for resources by selecting one of the metrics for the projection.

2. Navigate and filter the resources using the following options:

   - **Severity Based Checkboxes**: Enables you to filter the resources based on the selected Severity for the selected metric in the Metric field under Projection options.



   -

: Enables you to search and filter the data based on the text provided in the filter bar.
   - **Pagination**: Enables you to navigate between the paginated data using the Next, Previous, First, and the Last page links. You can also specify the rows that you want to view per page.

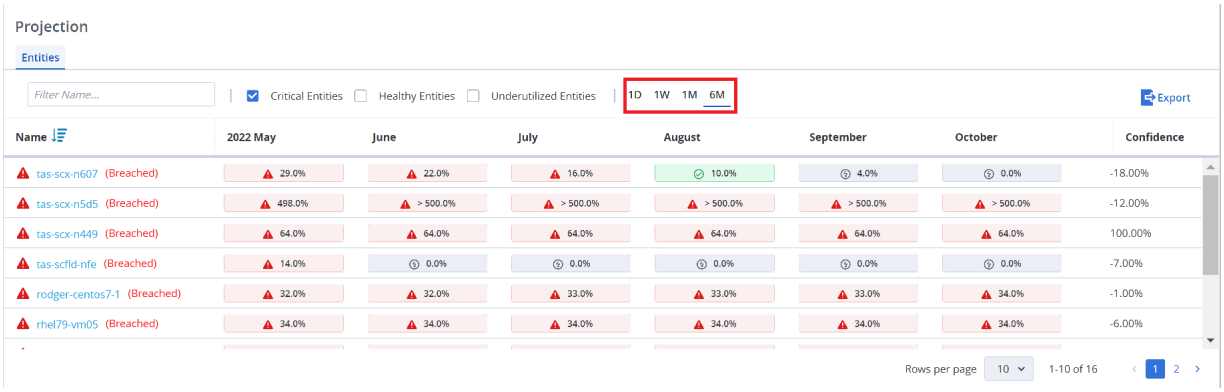
3. Click the entity name that you want to further analyze in the Projections section.
   The application displays the Device Details Summary View.
4. Click
   Export
   to export the projections to a CSV file.

Analyze Resource Health

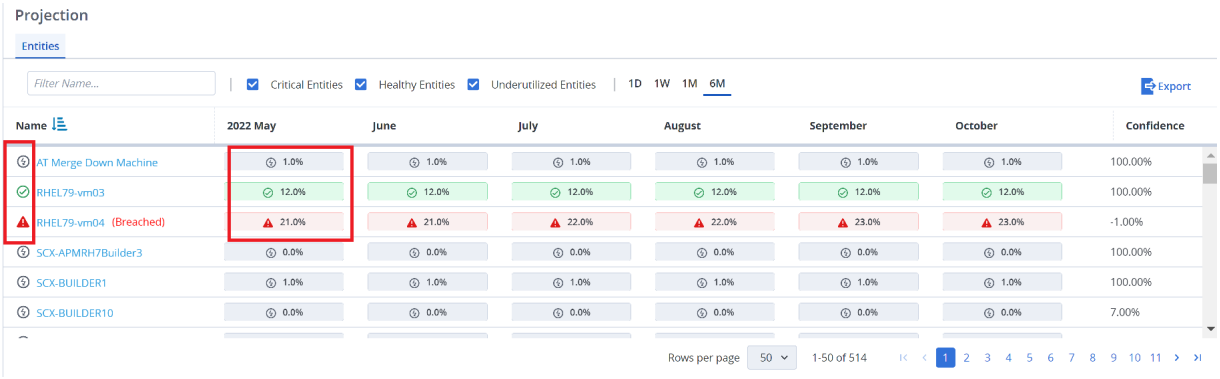
As a platform administrator, you can analyze the resource health, identify the overutilized, underutilized, and over-provisioned resources, and take necessary actions to resize and optimize the resource utilization.

DX Operational Intelligence provides the following information, forecasts, and various filters to perform the in-depth analysis seamlessly:

- **Metrics**: Includes the applicable metrics based on the selected severity level in the Metric field drop-down.
- **Resource Projections**: Provides multiple projection options to generate short, medium and long term projections. By default, displays 6-Month projection for each resource.



- **Color Codes**: Uses color codes and legends to indicate the severity of the capacity utilization:
  - : Indicates that the resource utilization is critical.
  - : Indicates that the resource utilization is healthy.
  - : Indicates that the resource is underutilized.



To analyze the resource health, follow these steps:

1. Select the following projection options while analyzing the resource projections. The projection options enable you to generate and analyze the projection variations using different combinations.
   - **Metric**: Select the metric that you want to use for projection. Use this metric to build data based on say Heartbeat Health State, Memory Balloon, Disk Space, and so on.
   - **Growth**: Select the expected workload growth rate that you want to use to model your capacity prediction. Use this input to forecast growth beyond default forecasts. For example, the use of a resource is expected to grow an extra 40 percent due to a new office. For example, select 40 percent growth to update the capacity forecasts with this growth.
2. Select the following options to view the short, medium and long term projections of the resources:
   - **1D**: Generates an hourly forecast for the next 24 hours. The application uses 24-Hour format to display the hourly projections.
   - **1W**: Generates the capacity forecast for the next seven(7) days.
   - **1M**: Generates the capacity forecast for the next 30 days.
   - **6M**: Generates the capacity forecast for the next six(6) months.
3. Click the entity name that you want to further analyze in the Projections section.

   The application displays the Device Details Summary View.
4. Click

   Export

   to export the projections to a CSV file for further analysis.

View Breached Configured Items

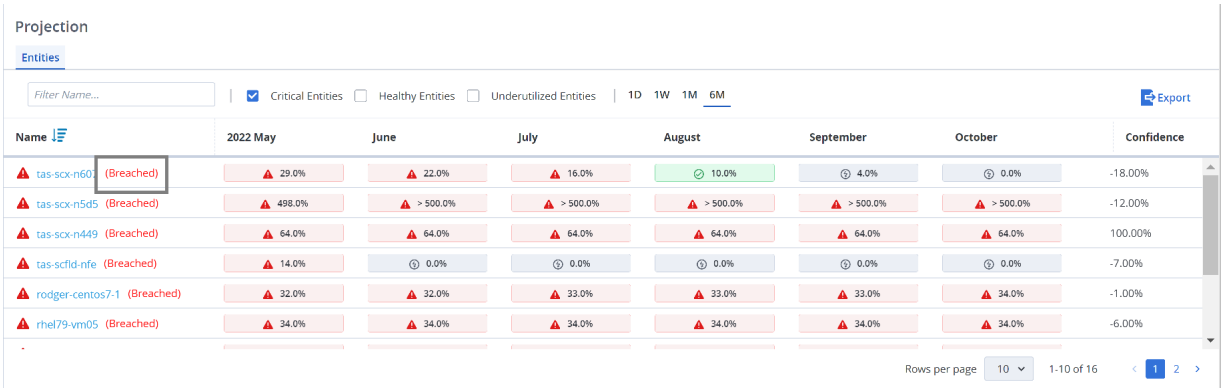
DX Operational Intelligence provides information about the breached configured item in the Resource Health view.

In the Capacity Forecast projection, the Resource Health view displays a

(Breached)

text next to the resource name. The Breached text indicates that the resource projection has a breached configured item.

To view the configuration item, hover over the Breached link.



NOTE
In the Device Details Summary view, you can generate and analyze the projection by selecting the breached configured item as a submetric.

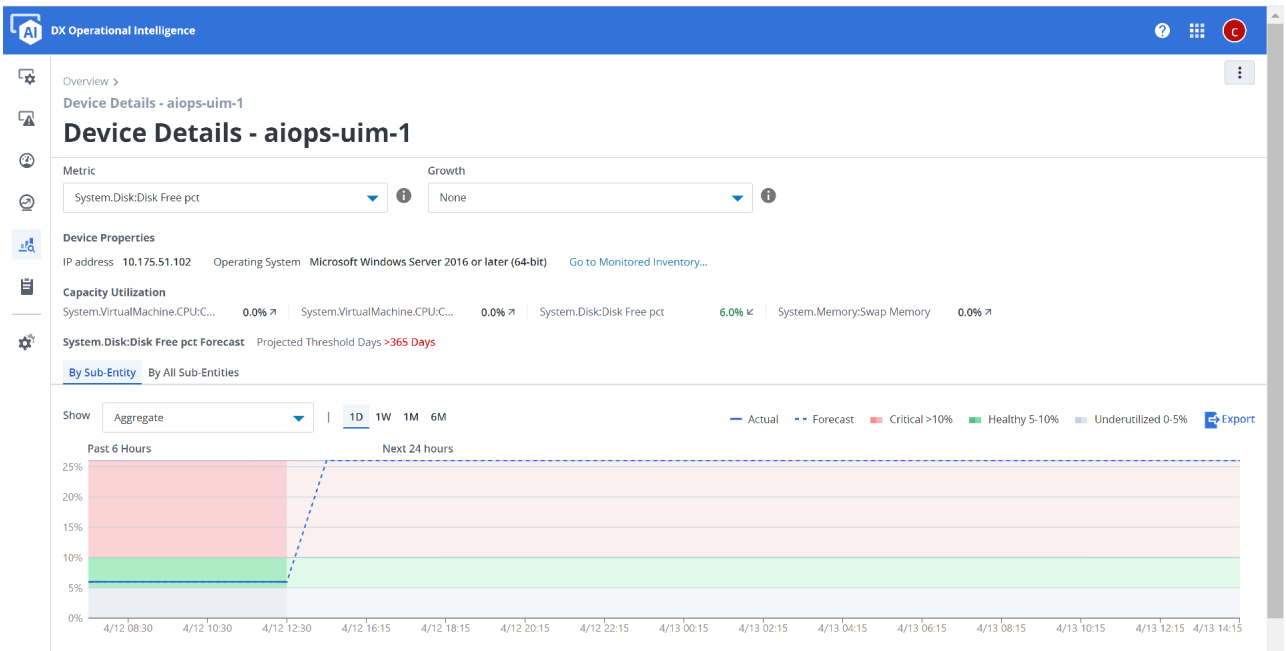
Access Device Details Summary View from Resource Health View

View the health of a single device or a resource and the hourly to 6-month forecast in the Device Details Summary view. Reset forecast when you recycle or upgrade the device.

You can access the Device Details Summary view from the Health Chart.

The Device Details Summary view provides the following details for the selected device or entity:

- **Device Properties**: Provides the device properties such as IP address, operating system. You can navigate to the Monitored Inventory widget of the selected device for additional details.
- **Capacity Utilization**: Displays the capacity utilization using the color codes of the selected device.
- **Associated Service**: Displays the list of services that are associated with the device.
- **Linked Metrics**: Displays the linked metrics that are associated with the selected metric.
- **Projections**: Generates the following projections:
  - **By Sub-Entity Projection**: Displays a 3-Month actual and 6-month threshold projection data as a line-chart for the selected entity. The projection shows the data that ranges from Unused to Critical with the Confidence level using color codes to indicate the severity.
  - **By All Sub-Entities Projection**: Displays a 6-Month threshold projection data for the entities that are associated with the service or group of the selected entity.



To access and view the Device Details Summary view, follow these steps:

1. Click the resource in the Projections section of the Resource Health view.

   The application displays the Device Detail Summary view.

   Projection Options

2. Select the following projection options while analyzing the resource projections. The projection options enable you to generate and analyze the projection variations using different combinations.
   - **Metric**: Select the metric that you want to use for projection. Use this metric to build data based on say Heartbeat Health State, Memory Balloon, Disk Space, and so on.
   - **Growth**: Select the expected workload growth rate that you want to use to model your capacity prediction. Use this input to forecast growth beyond default forecasts. For example, the use of a resource is expected to grow an extra

40 percent due to a new office. For example, select 40 percent growth to update the capacity forecasts with this growth.

Device Properties in Monitored Inventory

3. Click

in the Device Properties section to navigate to the Monitored Inventory widget.

Capacity Forecast

4. In the By Sub-Entity tab, select the following options to view the short, medium and long term capacity projections for the selected resource:
   - 1D: Generates past 6-hour actual capacity utilization and the next 24-hour capacity projection for the selected resource. The application uses a 24-Hour format to display the hourly forecast.
   - 1W: Generates past 3-day capacity utilization and the next 7-day capacity projection for the selected resource.
   - 1M: Generates past 10-day capacity utilization and the next 30-day capacity projection for the selected resource.
   - 6M: Generates past 3-month capacity utilization and the next 6-month capacity projection for the selected resource.
5. Select the submetric in the **Show** drop-down to display the projections for the selected submetric.

   By default, the projection displays the capacity forecast aggregation. The widget calculates the aggregation as an average of the associated submetrics capacity forecast.
6. Click the **By All Sub-Entities** tab to view the entity-wise capacity forecast.

Reset Forecast

7. Complete the following steps to reset the forecast for a device:

   NOTE
   You can reset the forecast for a resource when you recycle or upgrade the resource.

   a) Click and select the**Reset Forecast** option.
   b) Select the date in the **Reset From** field

      NOTE
      You can reset up to 42 days of historical data. After you reset, the historical data from the selected reset date to the current date is not accessible.
   c) Click **Apply**.

      The Device Details Summary view regenerates the forecast for the device.

Export

8. Click

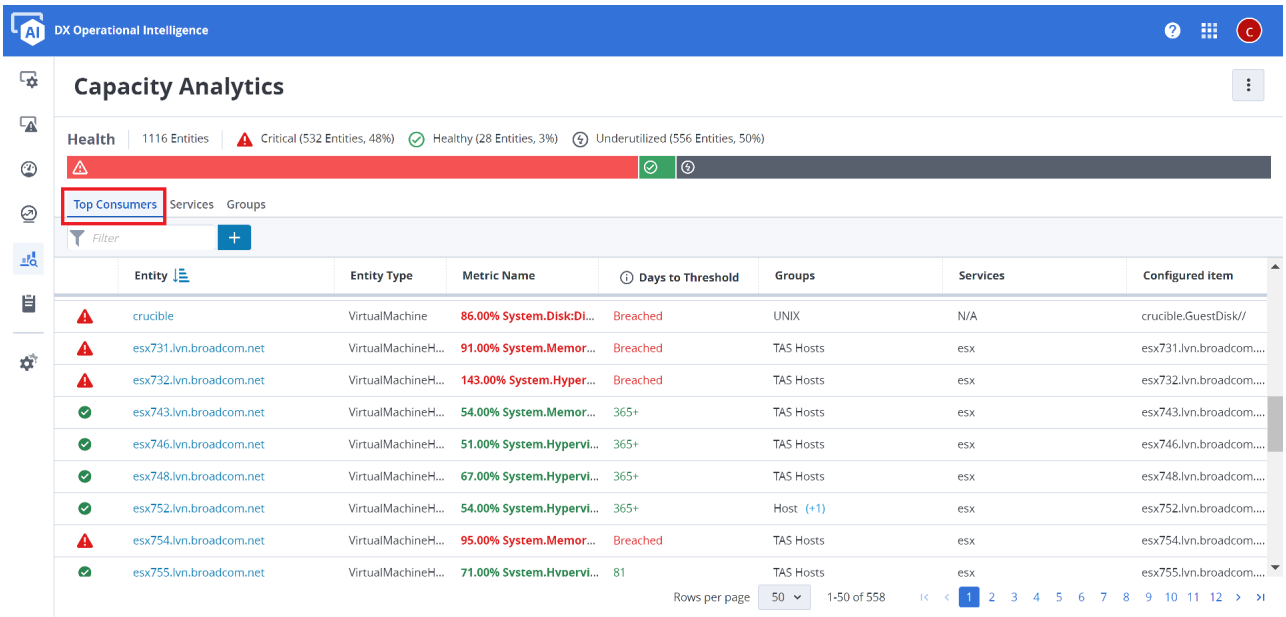
   to export the projections to a CSV file.

Top Capacity Consumers

You can use the Top Capacity Consumers View to locate resources that are at or near their capacity threshold limits.

The Top Capacity Consumers View displays the list of top resources that are at risk of consuming their capacity and reach the threshold limits within the projected time period. A resource can be a CPU, memory, disk utilization, virtual machine and so on. The View displays the following information for each resource in the Top Capacity Consumers view:

- Type of resource (Entity Type)
- Consumption percentage for the associated Metrics
- Days to reach the threshold limit (Days to Threshold)
- Associated Groups and Services
- Associated Configured items



{"URL":["https://digital-oi/capacity-analytics/topconsumers"],"customLabelGetStarted":"Top Consumers Overview","description":"concept.dita_8ed9e034-d9c4-454b-8ba9-78e995d59b47","heroDescriptionIdentifier":"Capacity Analytics is an analytical approach used to determine optimized resources for the continuity of operations. Capacity Analytics helps you in determining the required capacity for infrastructure resources such as CPU, memory, storage, and network for the operational continuity of the enterprise workloads. Capacity analytics allow you to manage demands for IT resources proactively in a cost-effective manner. You can optimize the performance and efficiency of existing resources, plan for and justify any financial investments. By using Capacity Analytics in Key definition for \"pname\" not found in the DITA map., you can leverage the following benefits: Predict capacity for peak seasons. Understand when more resources are needed and plan accordingly. Only buy additional resources when required. Efficiently manage infrastructure and networks. Eliminate wastage of resources by identifying areas that are underutilized."}

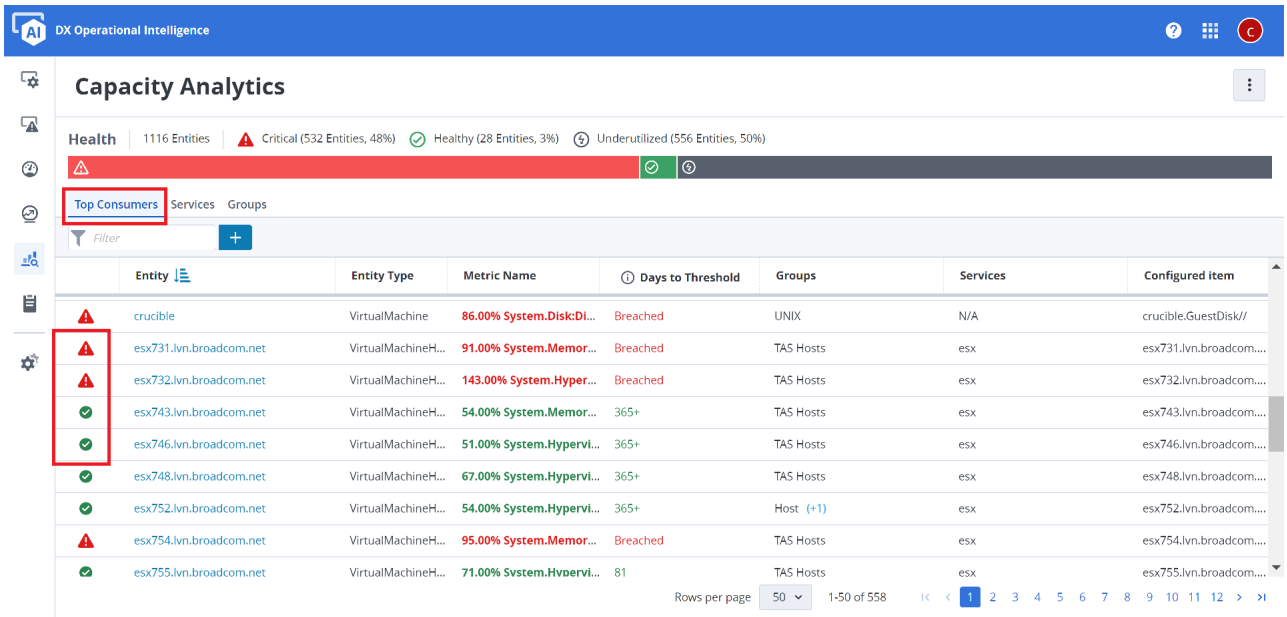
Access Top Capacity Consumers View

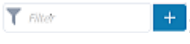
You can view the current top capacity consumers list in the Top Capacity Consumers view.

To view the Top Capacity Consumers, follow these steps:

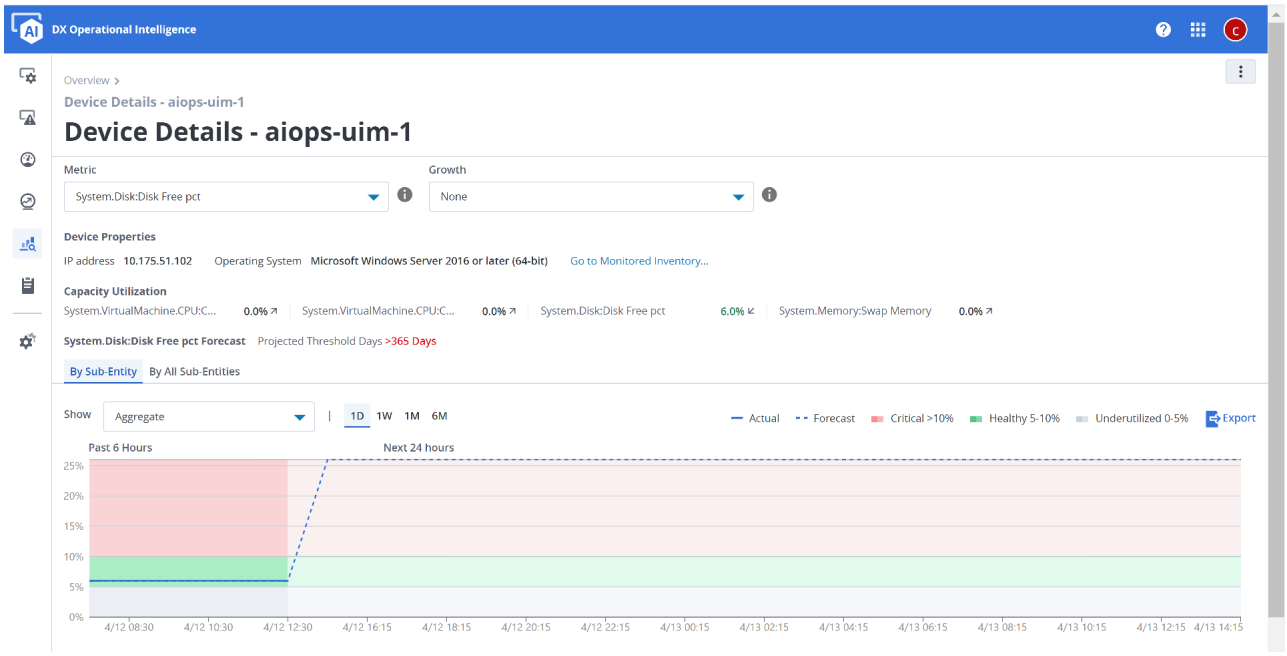
1. In the Capacity Analytics home page, click **Top Consumers** tab.

   The view displays the list of top resources that are at risk of consuming their capacity threshold limits. The view displays the severity of capacity threshold limits using the color codes:



2. Navigate and filter the services using the following options:
   -  : Enables you to search for entities using a value based filter criteria. You can define the filters using the filter attributes, operators and the values of the resource.
   - **Pagination**: Enables you to navigate between the paginated resource data using the Next, Previous, First, and the Last page links. You can also specify the rows that you want to view per page.
3. Click the resource to navigate to the Device Details Summery view.

   DX Operational Intelligencenavigates you to the Device Details Summary view for the selected resource:

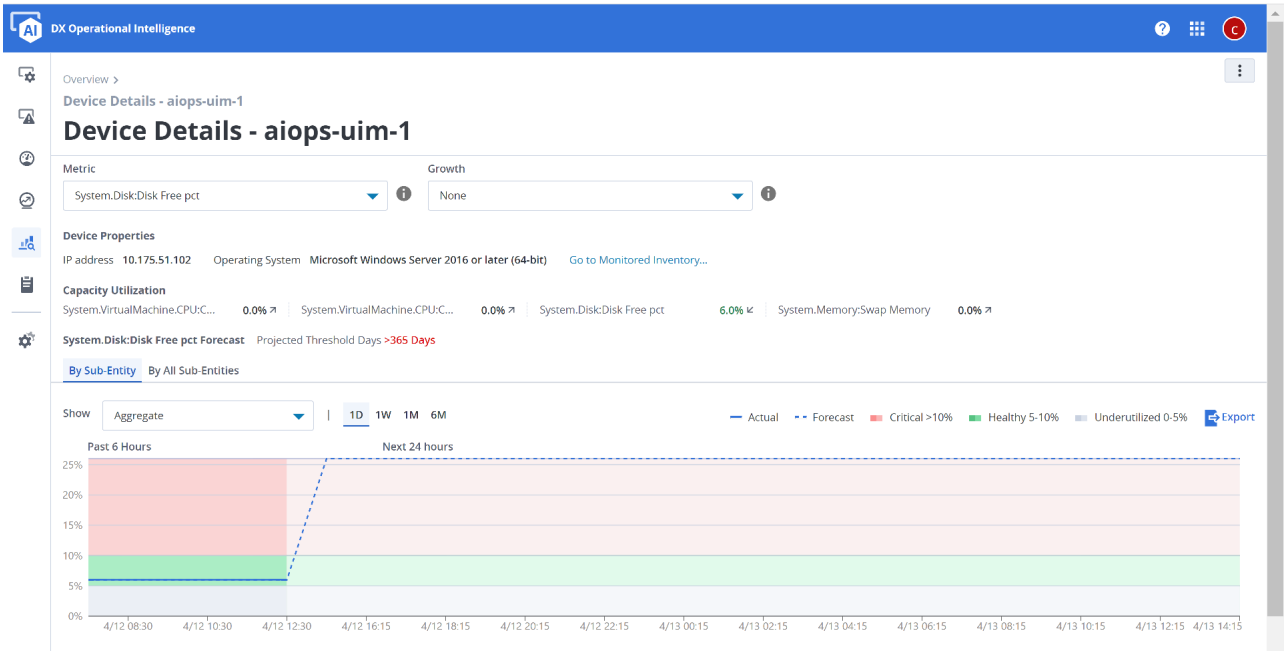


Access Device Details Summary view of Top Capacity Consumer

You can access and view the the Device Details Summary from the Top Capacity Consumers view.

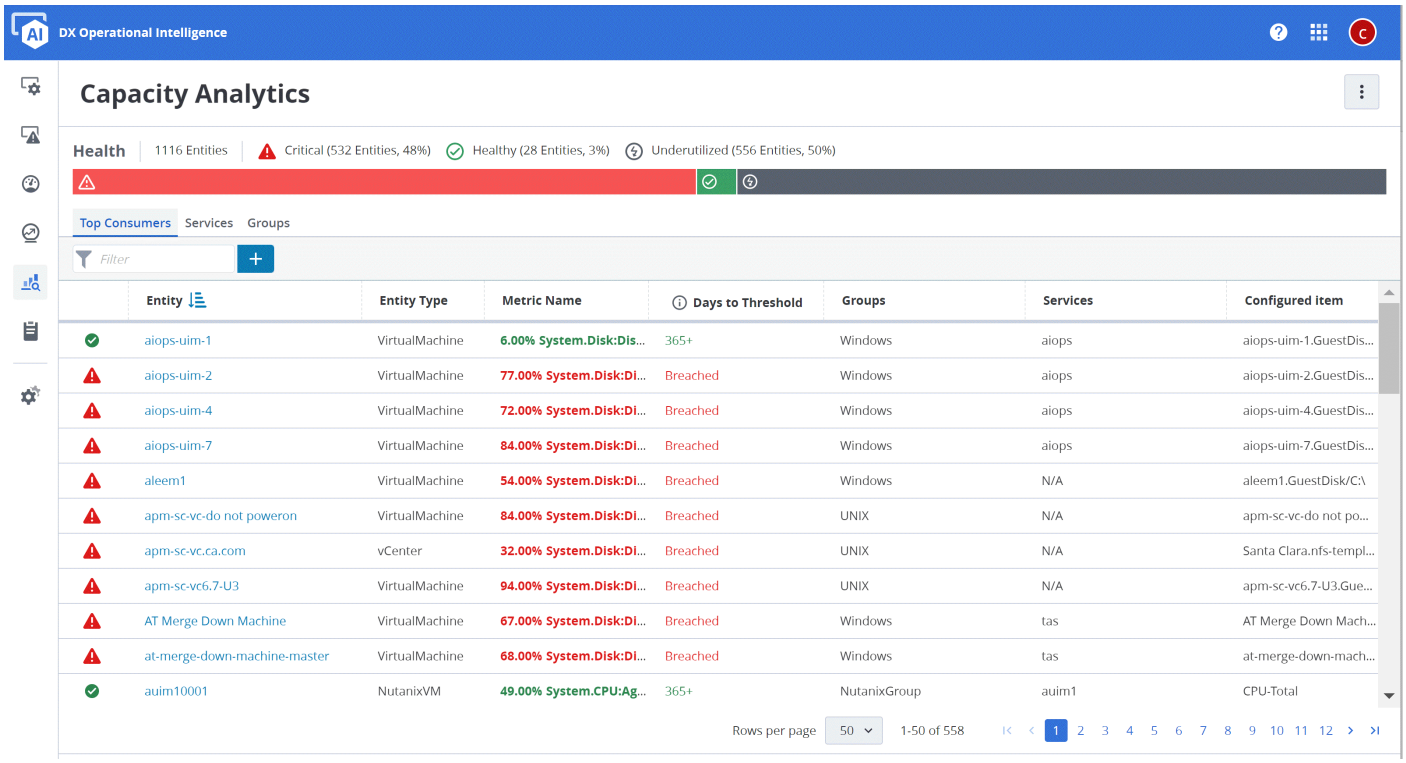
The Device Details Summary view provides the following details for the selected device or entity:

- **Device Properties**: Provides the device properties such as IP address, operating system. You can navigate to the Monitored Inventory widget of the selected device for additional details.
- **Capacity Utilization**: Displays the capacity utilization using the color codes of the selected device.
- **Associated Service**: Displays the list of services that are associated with the device.
- **Linked Metrics**: Displays the linked metrics that are associated with the selected metric.
- **Projections**: Generates the following projections:
  - **By Sub-Entity Projection**: Displays a 3-Month actual and 6-month threshold projection data as a line-chart for the selected entity. The projection shows the data that ranges from Unused to Critical with the Confidence level using color codes to indicate the severity.
  - **By All Sub-Entities Projection**: Displays a 6-Month threshold projection data for the entities that are associated with the service or group of the selected entity.



To access the Device Details Summary view of a resource in the Top Capacity Consumers view, follow these steps:

1. Click the resource name in the Top Capacity Consumers view .
   The application displays the Device Details Summary view of the resource.



Projection Options

2. Select the following projection options while analyzing the resource projections. The projection options enable you to generate and analyze the projection variations using different combinations.
   - **Metric**: Select the metric that you want to use for projection. Use this metric to build data based on say Heartbeat Health State, Memory Balloon, Disk Space, and so on.
   - **Growth**: Select the expected workload growth rate that you want to use to model your capacity prediction. Use this input to forecast growth beyond default forecasts. For example, the use of a resource is expected to grow an extra 40 percent due to a new office. For example, select 40 percent growth to update the capacity forecasts with this growth.

Device Properties in Monitored Inventory

3. Click

in the Device Properties section to navigate to the Monitored Inventory widget.

Capacity Forecast Projections

4. In the By Sub-Entity tab, select the following options to view the short, medium and long term capacity projections for the selected resource:
   - 1D: Generates past 6-hour actual capacity utilization and the next 24-hour capacity projection for the selected resource. The application uses a 24-Hour format to display the hourly forecast.
   - 1W: Generates past 3-day capacity utilization and the next 7-day capacity projection for the selected resource.
   - 1M: Generates past 10-day capacity utilization and the next 30-day capacity projection for the selected resource.
   - 6M: Generates past 3-month capacity utilization and the next 6-month capacity projection for the selected resource.

5. Select the submetric in the **Show** drop-down to display the projections for the selected submetric.

   By default, the projection displays the capacity forecast aggregation. The widget calculates the aggregation as an average of the associated submetrics capacity forecast.

6. Click the **By All Sub-Entities** tab to view the entity-wise capacity forecast.

7. Complete the following steps to reset the forecast for a device:

   NOTE
   You can reset the forecast for a resource when you recycle or upgrade the resource.

   a) Click and select the**Reset Forecast** option.

   b) Select the date in the **Reset From** field

      NOTE
      You can reset up to 42 days of historical data. After you reset, the historical data from the selected reset date to the current date is not accessible.

   c) Click **Apply**.

      The Device Details Summary view regenerates the forecast for the device.

8. Click

   to export the projections to a CSV file.

Configured Groups and Services

This Groups and Services views display the configured groups and services in Capacity Analytics.

- Services View for Capacity Analytics
- Groups View for Capacity Analytics

Services View for Capacity Analytics

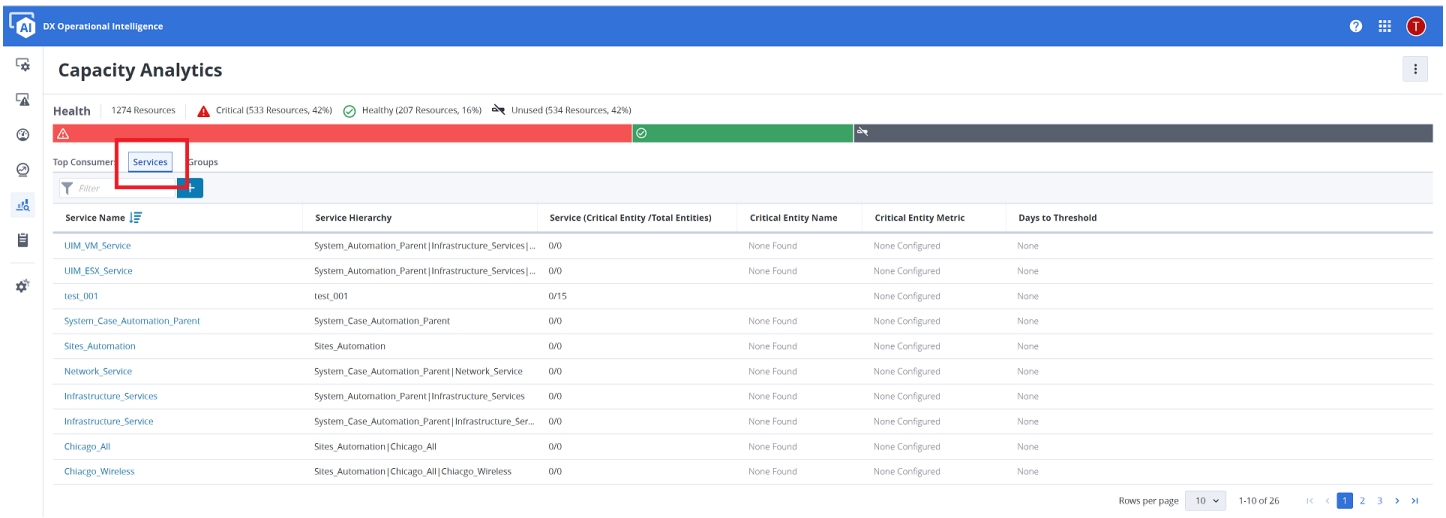
The Services tab contains the list of configured services in Capacity Analytics.

You can access the configured services in the Services tab of Capacity Analytics overview page. The Services view displays the service hierarchies of the service, critical entity count and total, entity type, metric and days to threshold information. You can sort all the columns except for Critical Entity Metric column.

- **Service Name:** Displays the name of the monitored service.
- **Service Hierarchy:** Displays the hierarchy name of the service.
- **Service (Critical Entity / Total Entities):** Displays the service count, that is, the critical entity count out of the total entities count. For example, (2/5) represents 2 critical entities out of 5 total entities.
- **Critical Entity Name:** Displays the name of the critical entity.
- **Critical Entity Metric:** Displays the name of the critical metric along with the percentage value. The color of the metric name indicates the threshold level: Red for Critical, Green for Healthy, and Grey for Underuntilized.
- **Days to Threshold:** Refers to the severity of the entity. When the entity has reached the threshold limit, the entity status text is shown as **Breached** under the **Days to Threshold** column. The color of the status text determines the current status of the entity.

  NOTE
  The **Days to Threshold** status indicates past projections, while the color of the status text indicates the current severity of the entity. Therefore if the status of a particular entity is **Breached** under the **Days to**

Threshold column, but if some action has been taken to address the threshold breach the status text appears in the corresponding color.



{"URL":["https://digital-oi/capacity-analytics/services"],"description":"concept.dita_b3036f74-5cc1-4263-b1a2-ecd0daee52d5","new":"","new_video":"","admin":"","heroDescriptionIdentifier":"","troubleshooting":{"masterkb":"","text":"","URL":[]},"pendo":"","video":[],"customCards":[]}

Access Services View

To access the Services view, follow these steps:

1. Click the **Services** tab in the Capacity Analytics home page.

   The Service view displays the list of configured services.

2. Navigate or filter the services using the following options:

   - *Filter Name...* : Enables you to search and filter the data based on the text provided in the filter bar.
   - **Pagination**: Enables you to navigate between the paginated data using the Next, Previous, First, and the Last page links. You can also specify the rows that you want to view per page.

3. Click the Service Name that you want to view and analyze further.

   The application displays the Service Details Summary view.

Access Service Details Summary View

Service Details view provides the forecast projections of the selected service.

{"URL":["https://digital-oi/capacity-analytics/service/*"],"customLabelGetStarted":"Access Service Details Summary View","description":"task.dita_b4841e10-ef28-44db-908d-a67c0f024171"}

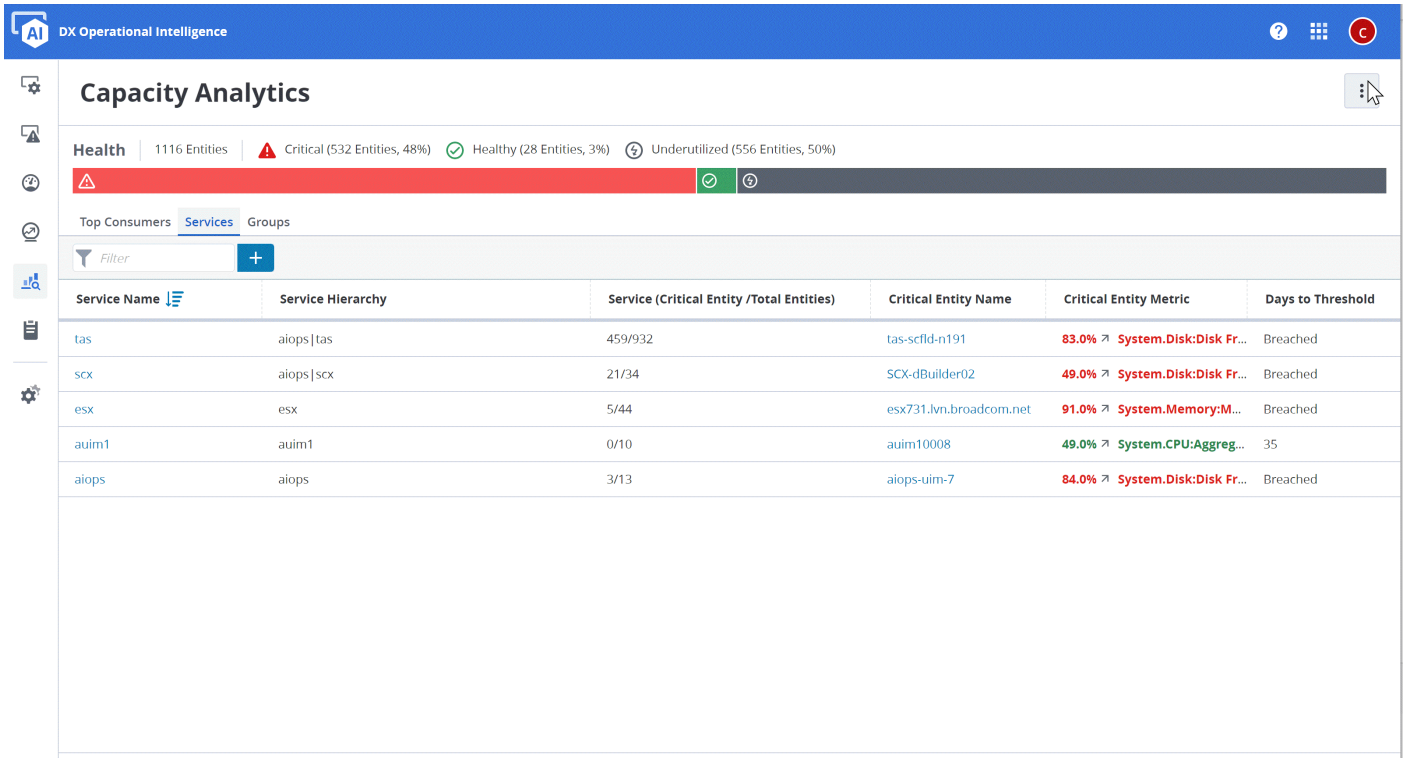
The Service Details view provides the forecast projections of the selected service. The Service Details Summary provides projections at 3 levels in different tabs:

- **Service**: Displays the 6-month projection as a rollup aggregation of the associated subservices or entities.
- **Subservices**: Displays the list of associated subservices and the 6-month projection as a rollup aggregation of the associated entities for each subservice.
- **Entities**: Displays the list of associated entities and 6-Month projection for each entity.

To access and view the Service Details view, follow these steps:

1. In the Services view, click the service that you want to view and analyze.

   The application displays the Service Details Summary view with the 6-Month projection for the selected service and the associated subservices and entities.



2. Select the following projection options while analyzing the resource projections. The projection options enable you to generate and analyze the projection variations using different combinations.
   - **Metric**: Select the metric that you want to use for projection. Use this metric to build data based on say Heartbeat Health State, Memory Balloon, Disk Space, and so on.
   - **Growth**: Select the expected workload growth rate that you want to use to model your capacity prediction. Use this input to forecast growth beyond default forecasts. For example, the use of a resource is expected to grow an extra 40 percent due to a new office. For example, select 40 percent growth to update the capacity forecasts with this growth.
3. Click the Service, Subservices, or the Entities tab in the projection section to view the projections at service, subservice, and entity levels, respectively.

   NOTE
   The Service Details view displays the Subservices and Entities tabs only when there are subservices and entities for the selected service.

4. Select the following options to view the short, medium and long term projection at Subservice and Entity level in the respective tabs:
   - **1D**: Generates an hourly forecast for the next 24 hours. The application uses 24-Hour format to display the hourly projections.
   - **1W**: Generates the capacity forecast for the next seven(7) days.
   - **1M**: Generates the capacity forecast for the next 30 days.
   - **6M**: Generates the capacity forecast for the next six(6) months.

5. Click the entity name that you want to further analyze in the Entities tab.
   The application displays the Device Details Summary View.
6. Click and then select the following options:.
   - **What-if Analysis**: Displays the What-if Analysis page. You can create and access the scenarios.
   - **Non-Percentage Metrics Dashboard**: Displays the Non-Percentage Metrics dashboard in a new tab.
7. Click

   to export the projections to a CSV file.

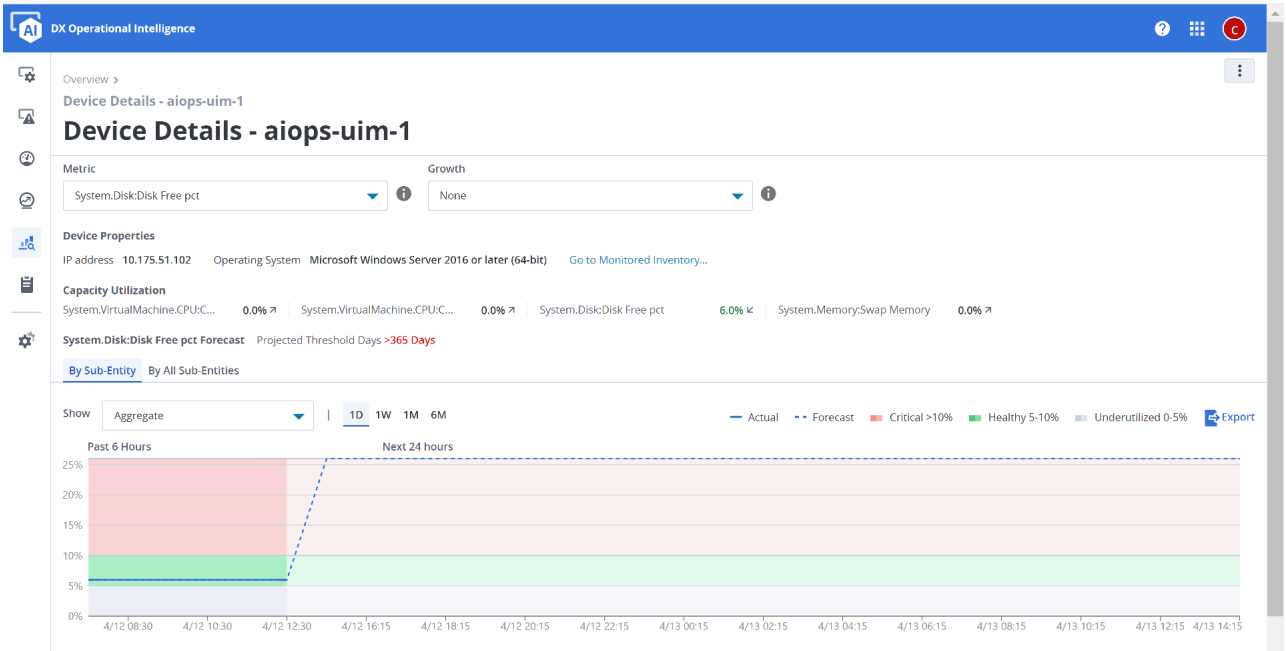
Access Device Details Summary View from Service Details View
{"URL":["https://digital-oi/capacity-analytics/devices/bysubentity"],"customLabelGetStarted":"Access Service Details Summary View","description":"task.dita_4dc3bf9d-3f98-4044-b3ea-91548e5c36b8"}

You can access the Device Details Summary view from Services Details Summary view.

The Device Details Summary view provides the following details for the selected device or entity:

- **Device Properties**: Provides the device properties such as IP address, operating system. You can navigate to the Monitored Inventory widget of the selected device for additional details.
- **Capacity Utilization**: Displays the capacity utilization using the color codes of the selected device.
- **Associated Service**: Displays the list of services that are associated with the device.
- **Linked Metrics**: Displays the linked metrics that are associated with the selected metric.
- **Projections**: Generates the following projections:
  - **By Sub-Entity Projection**: Displays a 3-Month actual and 6-month threshold projection data as a line-chart for the selected entity. The projection shows the data that ranges from Unused to Critical with the Confidence level using color codes to indicate the severity.
  - **By All Sub-Entities Projection**: Displays a 6-Month threshold projection data for the entities that are associated with the service or group of the selected entity.

To access and view the Device Details Summary view, follow these steps:



1. Click the entity in the Entities tab of the Services Detail or Groups Detail views.

   The application displays the Device Detail Summary view for the selected entity.

2. Select the following projection options while analyzing the resource projections. The projection options enable you to generate and analyze the projection variations using different combinations.

   - **Metric**: Select the metric that you want to use for projection. Use this metric to build data based on say Heartbeat Health State, Memory Balloon, Disk Space, and so on.
   - **Growth**: Select the expected workload growth rate that you want to use to model your capacity prediction. Use this input to forecast growth beyond default forecasts. For example, the use of a resource is expected to grow an extra 40 percent due to a new office. For example, select 40 percent growth to update the capacity forecasts with this growth.

3. Click

   in the Device Properties section to navigate to the Monitored Inventory widget.

4. Select the submetric in the **Show** drop-down to display the projections for the selected submetric.

   By default, the projection displays the capacity forecast aggregation. The widget calculates the aggregation as an average of the associated submetrics capacity forecast.

5. Click the **By All Sub-Entities** tab to view the entity-wise capacity forecast.
6. Complete the following steps to reset the forecast for a device:

   NOTE
   You can reset the forecast for a resource when you recycle or upgrade the resource.

   a) Click and select the**Reset Forecast** option.
   b) Select the date in the **Reset From** field

      NOTE
      You can reset up to 42 days of historical data. After you reset, the historical data from the selected reset date to the current date is not accessible.
   c) Click **Apply**.

      The Device Details Summary view regenerates the forecast for the device.
7. Click

   to export the projections to a CSV file.

What-If Analysis
You can use What-If Analysis to understand the dynamics of a scenario and the impacts of levers and dependencies in your choices.

You can use What-If Analysis to determine the effects on outcomes in a mathematical model by changing the inputs to the model in multiple scenarios. What-If Analysis enables capacity planners to perform an analysis based on the service KPIs and also in the context of the service.

What-if Analysis provides you with an option to compare the capacity for entities in different scenarios based on the service KPIs to help you plan better. What-if Analysis helps you understand the future needs of resources by simulating growth in the monitored KPI metrics and enables you to predict bottlenecks based on the 'what-if' scenarios. What-if Analysis also helps you make sense of the business metric in relation to the infrastructure metric.

What-If Analysis supports up to a maximum of three scenarios at a time. In addition, you also have an option to compare the newly created scenarios with existing scenarios. You have an option to create, modify, import, and delete a scenario respectively.

NOTE
What-if Analysis is available for only those metrics that are either linked or having percentage (pct) as their unit.

Create What-if Analysis Scenario

You need to create scenarios in order to analyze the different outcomes and impacts of your choices. You can create up to a maximum of three scenarios at a time, that is, once you have created three scenarios you can create another one only by deleting an existing scenario. To create a scenario, perform the following steps:

To create a What-if Scenario, follow these steps:

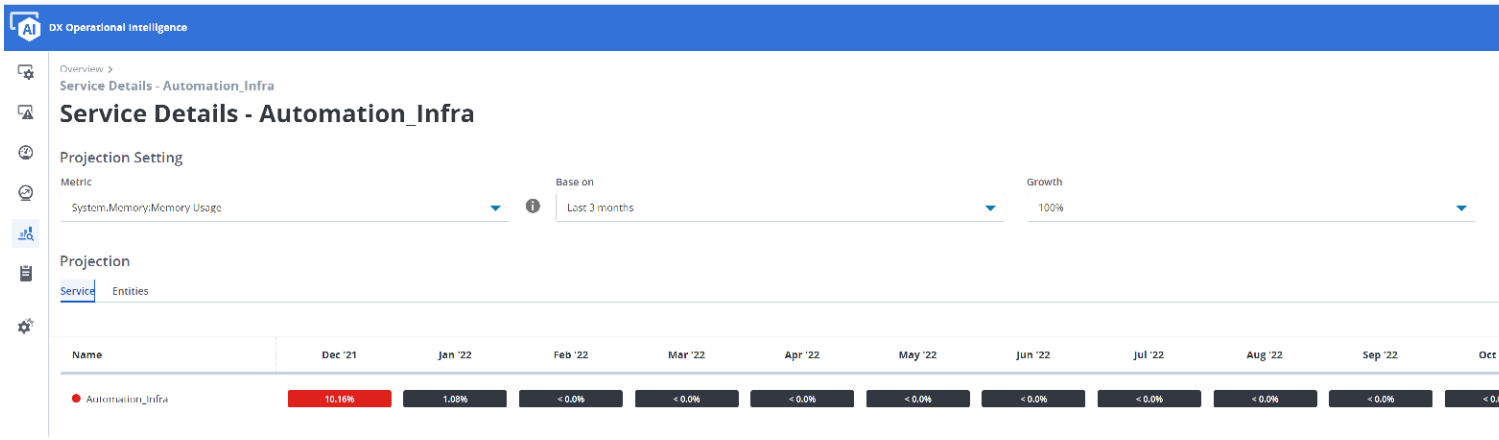
1. Click

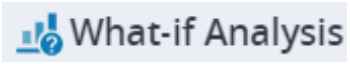
   in the left navigation pane.

   The Capacity Analytics Overview view opens.
2. Under the **Monitored Groups/Services** view, click the **Services** tab

   The **Services** tab opens by default.

3. Click the required **Service** to perform What-If analysis.



4. Click  on the top-right.
   The What-If Analysis view of the service opens.
5. Click and change the default What-if Analysis name to a meaningful name, and then click .
6. Perform one of the following actions to add a What-if Analysis scenario:

   Create one or more Scenarios:

   a) Click **Add new** icon.
   b) Click .
      The application adds a row with the scenario details to be configured.
   c) Provide the following information:
      - **Name**: Enter a meaningful name for the new scenario.
      - **KPI**: Select a KPI for the scenario.
      - **State**: Select a state for the scenario. By default, the state is selected as **All**. You can select the state as All, Critical, Healthy, or Unused.
      - **Metric**: Select a metric for the scenario. The metrics get populated based on the KPI you select.
      - **Current**: Displays the current value applicable for the scenario.
      - **Target**: Enter a target value for your analysis, but make sure that the target value is more than the current value. For example, if the current value is 100, then the target value should be more than 100.
      - **Month/Quarter/Year**: Select the forecast period for the scenario. You can select the value in terms of month, quarter, or year. Use the icon to select the exact forecast value. For example, let's select the forecast period as 6 months. The What-if Analysis view displays the data projections for 6 months.
   d) Click the icon to save the scenario.

      IMPORTANT
      After you save the What-if Analysis, you will not be able to make any further changes.

Import an Existing Scenario

a) From the new **What-If Analysis** view, click the

icon.

The existing saved analysis appears as a drop-down option.

b) Make the necessary changes in the imported scenarios.

c) Click the

icon to save the updated scenario.

What if Analysis Actions

You can perform actions actions while analyzing the What If Analysis.

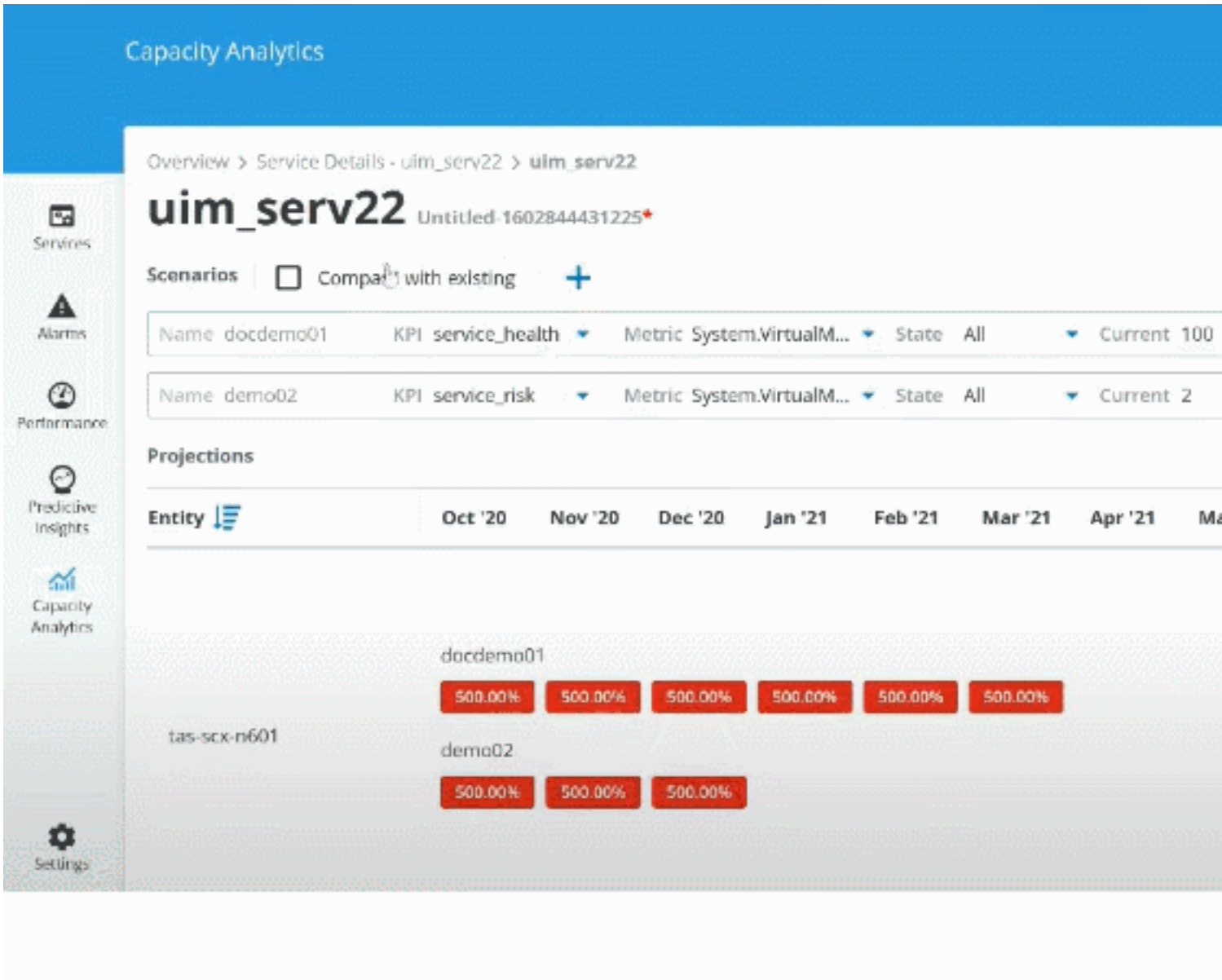
Export What If Analysis to a PDF

1. To export the What-If Analysis to a PDF format, click on the

icon on the top-right.

Compare with Existing Analysis

2. Click the **Compare with existing** checkbox to compare the newly created scenario(s) with an existing scenario.



NOTE

- To compare the newly created scenarios with existing scenarios, make sure that you already have existing scenarios that are saved and available.
- While using existing scenarios, you can only modify the **Metric** value, all the other options are Read-only.

Update Metric Values

3. (Optional) Modify the **Metric** value by using the drop-down option and click the ✓ to save the updates.

Filter by Entity

4. To **filter** the data projections by entity, enter the entity name in the **Filter by entity** filter field.

Disable Scenario

5. To **disable** a scenario data projection, click on the

   icon next to the scenario.

   Delete Scenario
6. To **delete** a scenario, click on the

   icon next to the scenario.

Import an Existing What-if Analysis

What-If Analysis also provides you with an option to import an existing What-if Analysis. You can use this option to import an existing saved analysis in to the What-If Analysis view to understand the dynamics of the existing analysis and scenarios. This view is a Read-only option, that is, you won't be able to make any modifications to the imported What-If Analysis or scenarios.

To import an existing What- If Analysis, follow these steps:.

1. From the **What-If Analysis** view, click the **Import a Saved Analysis** icon.

   The existing saved analysis appears as a drop-down option.
2. Select an existing **What-if Analysis**

   You can view the What-If Analysis and the respective scenarios for further assessment.

Groups View for Capacity Analytics

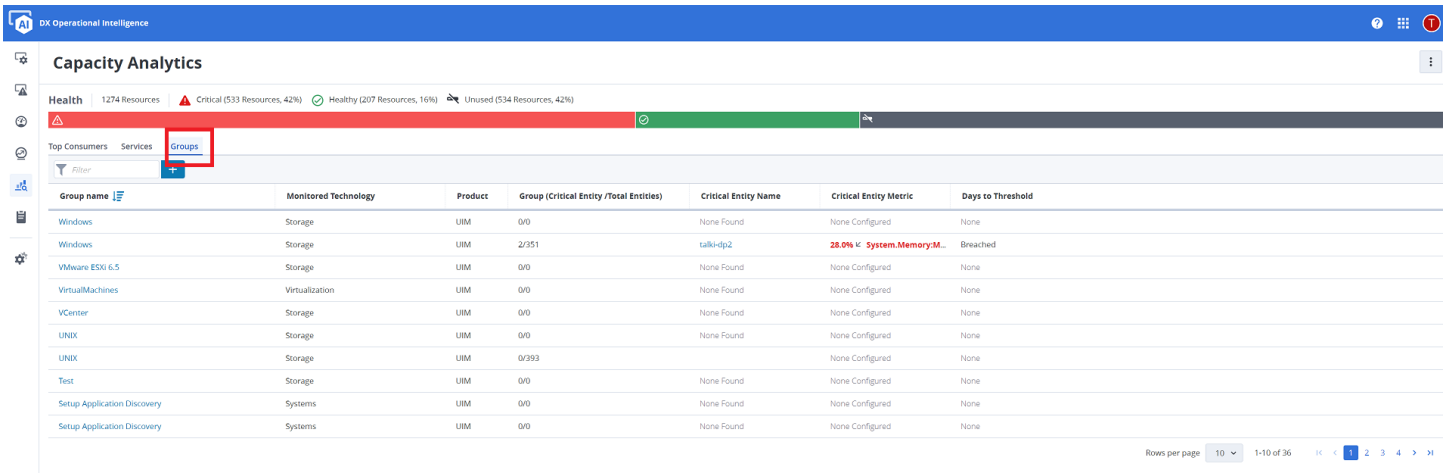
The Groups View contains the list of configured groups in Capacity Analytics.

The Groups view displays the monitored technology of the group, product name, critical entity count and total, entity type, metric, and days to threshold information.

- **Group Name:** Displays the name of the group.
- **Monitored Technology:** Displays the name of the monitored technology. This column displays the resource utilization of various platforms. The resources are categorized based on a group. Monitored Technology is applicable only for Groups. The following Group types are available:
  - System
  - Storage
  - Virtualization
  - Custom (Optional)
- **Product:** Displays the name of the source product.
- **Group (Critical Entity / Total Entities):** Displays the critical entity count out of the total entities count. For example, (2/5) represents 2 critical entities out of 5 total entities of the group.
- **Critical Entity Name:** Displays the name of the critical entity.
- **Critical Entity Metric:** Displays the name of the critical metric along with the percentage value. The color of the metric name represents the threshold level.
- **Days to Threshold:** It refers to the severity of the entity. When the entity has reached its threshold limit, the entity status text is shown as **Breached** under the **Days to Threshold** column. The color of the status text determines the current status of the entity.

  NOTE

  The **Days to Threshold** status indicates past projections, while the color of the status text indicates the current severity of the entity. Therefore if the status of a particular entity is **Breached** under the **Days to Threshold** column, but if some action has been taken to address the threshold breach the status text appears in the corresponding color.



{"URL":["https://digital-oi/capacity-analytics/groups"],"description":"concept.dita_c2b72331-6479-491d-bcd6-b08aa0b00b0e","new":"","new_video":"","admin":"","heroDescriptionIdentifier":"","troubleshooting":{"masterkb":"","text":"","URL":[]},"pendo":"","video":[],"customCards":[]}

Access Groups View

To view the Groups view, follow these steps:

1. Click the Groups tab in the Capacity Analytics Overview page.
2. Navigate or filter the groups using the following options:
   - Filter +: Enables you to search data using a value based filter criteria. You can filter the data in the widget using filter attributes, operators, and values.
   - **Pagination**: Enables you to navigate between the paginated data using the Next, Previous, First, and the Last page links. You can also specify the rows that you want to view per page.
3. Click the Group Name that you want to view and analyze further.

   The application displays the Group Details Summary view.

Access Group Details Summary View

{"URL":["https://digital-oi/capacity-analytics/group/*"],"customLabelGetStarted":"Access Group Details Summary View","description":"task.dita_10c3db00-3b7e-4032-8ceb-09741f3617ee"}

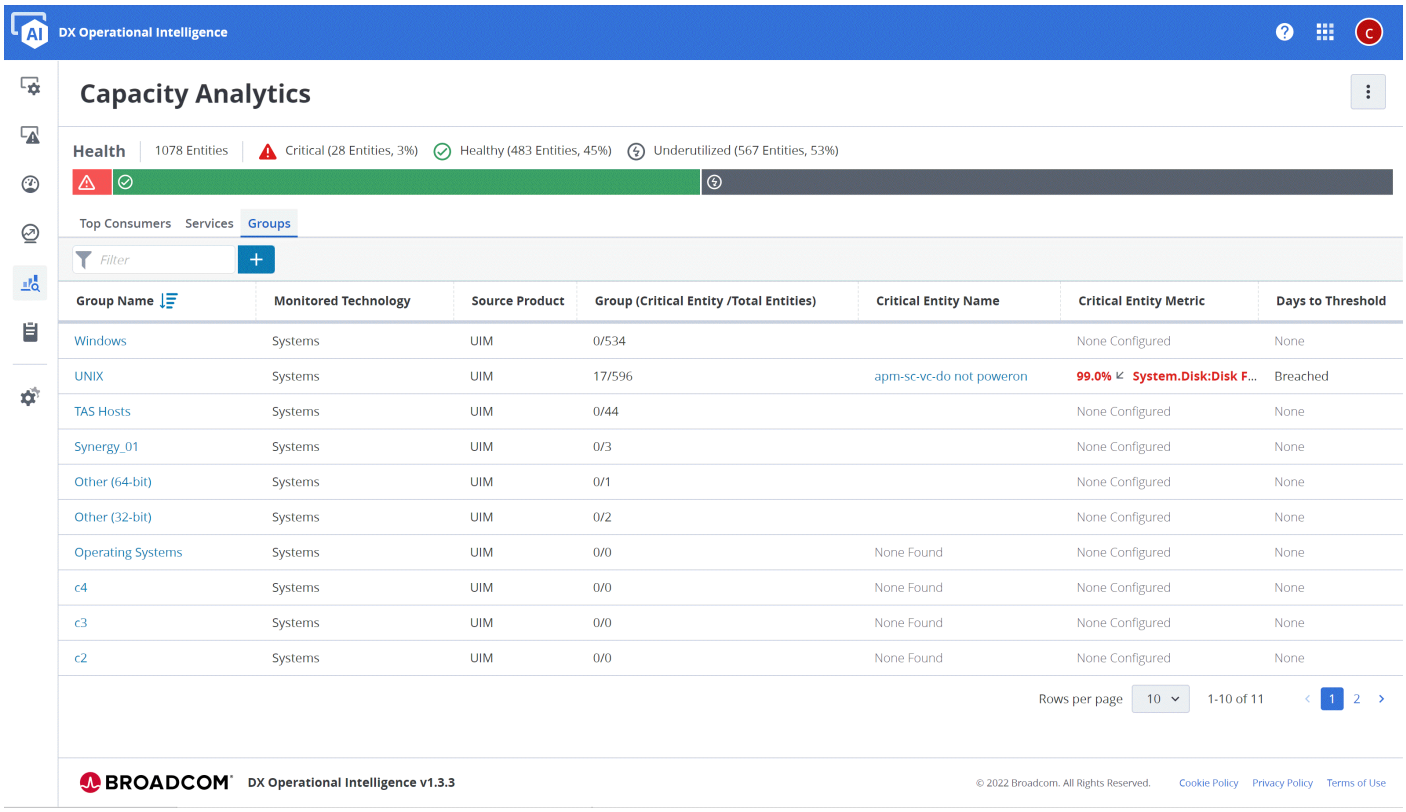
The Group Details Summary view provides the forecast projections of the selected group. The Group Details Summary provides projections at two levels in different tabs:

- **Group** : The view calculates the group-level projection as a rollup aggregation of the associated subgroups or entities in the Group tab.
- **Sub-Services**: The view calculates the subgroup-level projection as a rollup aggregation of the associated entities in the Subgroup tab.
- **Entities**: The view displays the entity-wise projections in the Entities tab.

To access and view the Group Details Summary view, follow these steps:

1. In the Groups View tab, click the group that you want to view and analyze.

   The application displays the Group Details Summary view with the 6-Month projection for selected group and the associated subgroups and entities.



2. Select the following projection options while analyzing the resource projections. The projection options enable you to generate and analyze the projection variations using different combinations.
   - **Metric**: Select the metric that you want to use for projection. Use this metric to build data based on say Heartbeat Health State, Memory Balloon, Disk Space, and so on.
   - **Growth**: Select the expected workload growth rate that you want to use to model your capacity prediction. Use this input to forecast growth beyond default forecasts. For example, the use of a resource is expected to grow an extra 40 percent due to a new office. For example, select 40 percent growth to update the capacity forecasts with this growth.
3. Click the Group, Subgroup, or the Entities tab in the projection section to view the projections at group, subgroup, and entity levels, respectively.

   NOTE
   The Group Details Summary view displays the Subgroups and Entities tabs only when the selected group has underlying subgroups or entities.
4. Select the following options to view the short, medium and long term projection at Subgroup and Entity level in the respective tabs:
   - **1D**: Generates an hourly forecast for the next 24 hours. The application uses 24-Hour format to display the hourly projections.
   - **1W**: Generates the capacity forecast for the next seven(7) days.
   - **1M**: Generates the capacity forecast for the next 30 days.
   - **6M**: Generates the capacity forecast for the next six(6) months.
5. Click the entity name that you want to further analyze in the Entities tab.

   The application displays the Device Details Summary View.

6. Click

   Export

   to export the projections to a CSV file.

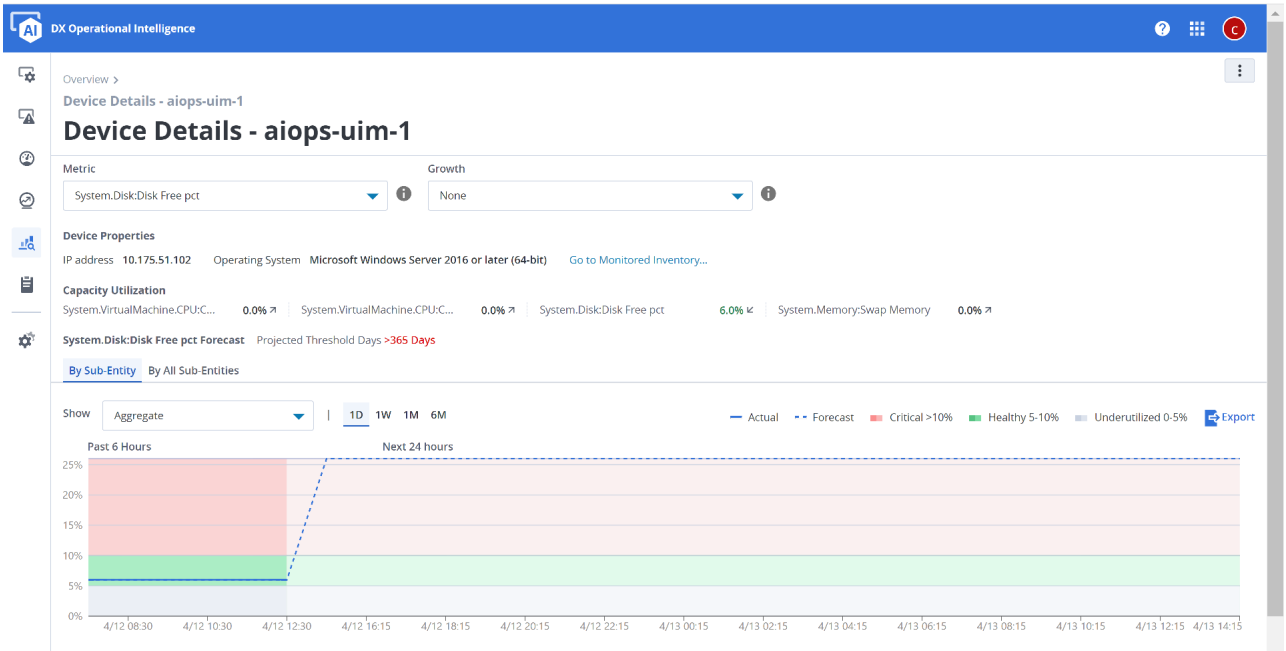
Access Device Details Summary View from Service Details View

{"URL":["https://digital-oi/capacity-analytics/devices/sub"],"customLabelGetStarted":"Access Device Details Summary View","description":"task.dita_4dc3bf9d-3f98-4044-b3ea-91548e5c36b8"}

You can access the Device Details Summary view from Services Details Summary view.

The Device Details Summary view provides the following details for the selected device or entity:

- **Device Properties**: Provides the device properties such as IP address, operating system. You can navigate to the Monitored Inventory widget of the selected device for additional details.
- **Capacity Utilization**: Displays the capacity utilization using the color codes of the selected device.
- **Associated Service**: Displays the list of services that are associated with the device.
- **Linked Metrics**: Displays the linked metrics that are associated with the selected metric.
- **Projections**: Generates the following projections:
  - **By Sub-Entity Projection**: Displays a 3-Month actual and 6-month threshold projection data as a line-chart for the selected entity. The projection shows the data that ranges from Unused to Critical with the Confidence level using color codes to indicate the severity.
  - **By All Sub-Entities Projection**: Displays a 6-Month threshold projection data for the entities that are associated with the service or group of the selected entity.



To access and view the Device Details Summary view, follow these steps:

1. Click the entity in the Entities tab of the Services Detail or Groups Detail views.

   The application displays the Device Detail Summary view for the selected entity.

2. Select the following projection options while analyzing the resource projections. The projection options enable you to generate and analyze the projection variations using different combinations.
   - **Metric**: Select the metric that you want to use for projection. Use this metric to build data based on say Heartbeat Health State, Memory Balloon, Disk Space, and so on.
   - **Growth**: Select the expected workload growth rate that you want to use to model your capacity prediction. Use this input to forecast growth beyond default forecasts. For example, the use of a resource is expected to grow an extra 40 percent due to a new office. For example, select 40 percent growth to update the capacity forecasts with this growth.
3. Click

   in the Device Properties section to navigate to the Monitored Inventory widget.
4. Select the submetric in the **Show** drop-down to display the projections for the selected submetric.

   By default, the projection displays the capacity forecast aggregation. The widget calculates the aggregation as an average of the associated submetrics capacity forecast.
5. Click the **By All Sub-Entities** tab to view the entity-wise capacity forecast.
6. Complete the following steps to reset the forecast for a device:

   NOTE
   You can reset the forecast for a resource when you recycle or upgrade the resource.

   a) Click and select the**Reset Forecast** option.

   b) Select the date in the **Reset From** field

      NOTE
      You can reset up to 42 days of historical data. After you reset, the historical data from the selected reset date to the current date is not accessible.

   c) Click **Apply**.

      The Device Details Summary view regenerates the forecast for the device.
7. Click

   to export the projections to a CSV file.

Service Key Performance Indicators (KPIs)

Capacity Analytics supports the capability to monitor and provide projections for the Service Key Performance Indicators (KPIs) configured in Service Analytics.

The Service KPIs are useful to generate projections for the metrics such as cloud service cost metric and cloud region cost metric that do not have an associated device.

Using the Service KPIs, capacity planners can use the 12-month forecast to get the projected cost for the cloud service used like storage and able to plan their budget better.

Prerequisites:

You must configure the Service KPIs based on metrics in Service Analytics before you view projections for them in correlation to the service infrastructure capacity.

For more information on configuring KPIs, see Create a Service section in Service Analytics.

View Service KPI Projections Dashboard

Capacity Analytics provides the out-of-the-box Service KPI Projection Dashboard. For more information, see. Capacity Analytics KPI Projections

You can also create custom dashboards for Service KPI projection using DX Dashboards.

Troubleshoot Capacity Analytics

This video walks you through the troubleshooting steps for some of the issues:

Predictive Insights

Predictive Insights is a capability that harnesses the power of machine learning to discover patterns and trends.

DX Operational Intelligence collects millions of data points for the different data types such as alerts, metrics, logs, events, and inventory into a single data lake. This huge data holds numerous information about the device behaviors, trends, patterns, and end-to-end data flows. These insights can be derived by applying analytics on each of these data types individually or in tandem with each other. Predictive Insights provides you with a call to action based on the predicted situations or incidents that are identified through the insights.

Based on the patterns and trends, the application predicts events that are likely to happen in the future. The events that could be predicted are:

- Performance
- Capacity

Learn about Predictive Insights in the following links:

- Predictive Insights User Interface
- Alarm Actions for Prediction Alarms
- Enable Predictive Definitions
- Predictive Insights OOB Metrics

{"URL":["https://digital-oi/predictive-insight"],"description":"concept.dita_3daedc21-f624-48f9-8ad4-6b5ce8364bba","new":"","new_video":"","admin":"","troubleshooting":{"masterkb":"","text":"","URL":[]},"pendo":"","video":[]}

Access Predictive Insights

Follow these steps:

1. Log in to DX Operational Intelligence
2. From the left-navigation page, click the

   . Depen

   on the user access role, you can view this page. By default, this page shows the predicted alarms for the next 10 days.

   NOTE

   - Predictive Insights displays the alarms that are going to breach the threshold limit within the next 10 days.
   - To view Predictive Insights data, complete the steps in Configure Data Definition topic.
   - Predictive insights is now enhanced to display Netapp/RESTmon metrics.

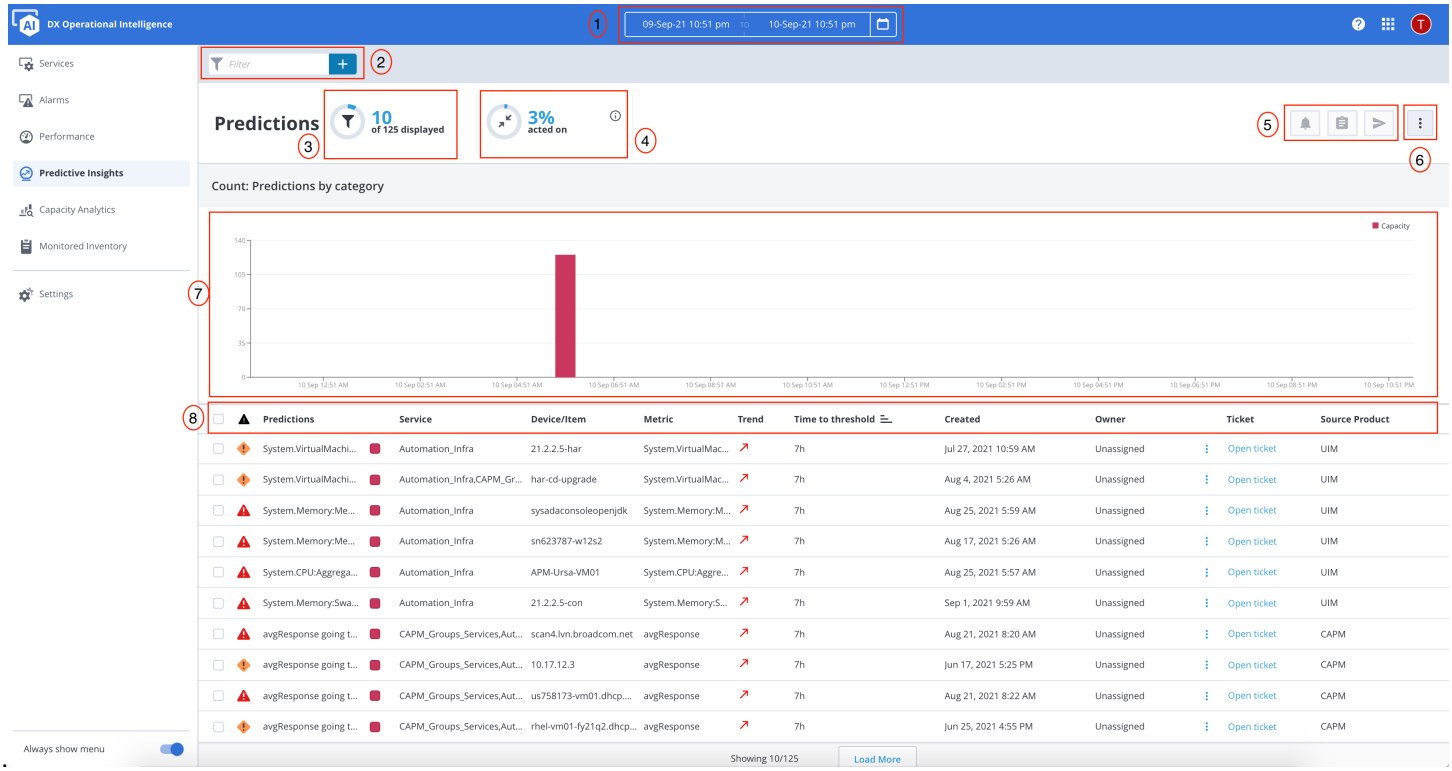
Predictive Insights User Interface

The Predictive Insights User Interface displays the total number of predictions that are displayed and also the percentage of predictions on which some action was taken.

{"URL":["https://digital-oi/digital-oi/predictive-insight"],"customLabelGetStarted":"Get Started with Predictive Inisghts","description":"concept.dita_1932af90-1e68-4af8-b6f7-d3b5d46b6250","customCards":[{"type":"configure","id":"concept.dita_20c0014b-4415-4a32-848c-059821d20ae2","url":"https://techdocs.broadcom.com/us/en/ca-enterprise-software/it-operations-management/dx-operational-intelligence-saas/SaaS/using/Predictive-Insights/Alarm-Actions-for-Prediction-Alarms.html","title":"Alarm Actions for Prediction Alarms"},{"type":"use","id":"concept.dita_01bdeff9-7826-43cb-9e38-9de87c65891d","url":"https://techdocs.broadcom.com/us/en/ca-enterprise-software/it-operations-management/dx-operational-intelligence-saas/SaaS/using/Predictive-Insights/Enable-Predictive-Definitions.html","title":"Enable Predictive Definitions"},{"type":"customize","id":"concept.dita_e031cc3b-5e2c-4576-b350-f61149ba4e39","url":"https://techdocs.broadcom.com/us/en/ca-enterprise-software/it-operations-management/dx-operational-intelligence-saas/SaaS/using/Predictive-Insights/Predictive-Insights-OOB-Metrics.html","title":"Predictive Insights OOB Metrics"}]}

The **Predictions** page depicts the future alarms graphically as a bar graph and in a table format.

This page displays the total number of predictions that are displayed and also the percentage of predictions on which some action was taken.



Search by Time Range (1)

Click the calendar icon to select the duration for which you want to view the prediction alarms. The default time range for the Predictive Insights page is 7 days from the current time. You can view predictions for any period for the next four months.

> **NOTE**
> You cannot select past date and time and future date after four months in the Predictive Insights page.

Filter by Attributes (2)

You can filter the predictions by using the attribute filter. This filter allows you to view only those alarms with attributes matching your search criteria.

Count of Predictions (3) Displays the total number of predictions.

Percentage of Predictions (4) Displays the percentage of predictions on which action was taken.

Prediction Alarm Actions (5) Allows you to perform actions such as alarm management, ticket management, channels

Three-Dot Menu (6) Use this button to enable auto-update view and set the Predictive Insights page as OI landing page.

Alarms Chart (7) The Alarms chart displays the predictive trend for alarms in a bar graph format. The graph displays prediction alarms for the following categories:

- Purple for Performance
- Red for capacity

Here the X-axis represents the date of prediction and the Y-axis represents the prediction count.

By default, the graph displays both the **Performance** and **Capacity** prediction alarms, to view only **Performance** or **Capacity** alarm, click a category or the legend in the graph.

Alarms Table (8) The Alarms page displays a table showing prediction alarms with relevant details.

To refresh the Alarms table automatically, enable the Auto-update view switch. Use this switch button to shift the view from auto-update to manual update.

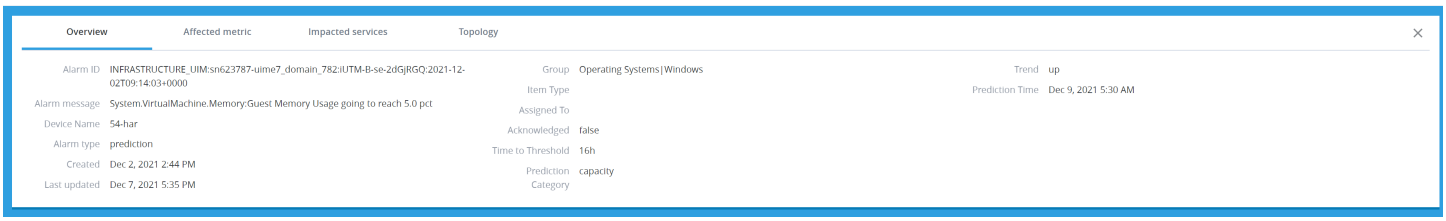
> **NOTE**
> Initially, only ten alarms prediction appears on the page. **LOAD MORE** button is enabled when there are more than ten alarms in the selected time range.

Column Name	Description
Severity	Indicates the severity of an alarm. The following colors indicate the severity: • Red: Critical • Orange: Major • Yellow: Minor • Light Blue: Informational • Teal: Warning • Green: Any other alarm that does not fall in the above categories
Predictions	Displays the prediction message.
Category	Displays the prediction alarm category. • Purple for Performance • Red for capacity
Service	Displays the service which is impacted by an alarm.
Device/Item Name	Displays the device or the component name for which the prediction alarm is generated. For example, <*DeviceName* > - <*ComponentName*>
Metric	Displays the metric for which the prediction alarm is generated.
Trend	Indicates the upward or downward trend for a metric.
Time to Threshold	Indicates the time that the metric takes to reach the threshold value. The threshold time is calculated from the current time. The time is displayed in hours. If time is greater than 24 hours, days are displayed.
Created	Displays the time when the alarm was created.

Column Name	Description
Owner	Displays the person name to whom the alarm is assigned and whether the person has acknowledged it. The alarms that have been acknowledged are indicated with a tick mark in this column. Click ⋮ to perform alarm actions (assign, acknowledge, and so on)
Ticket ID (Clickable)	Displays the ID generated by the ticketing system.
Source Product	Displays the product from which the alarm is generated.

Overview Tab

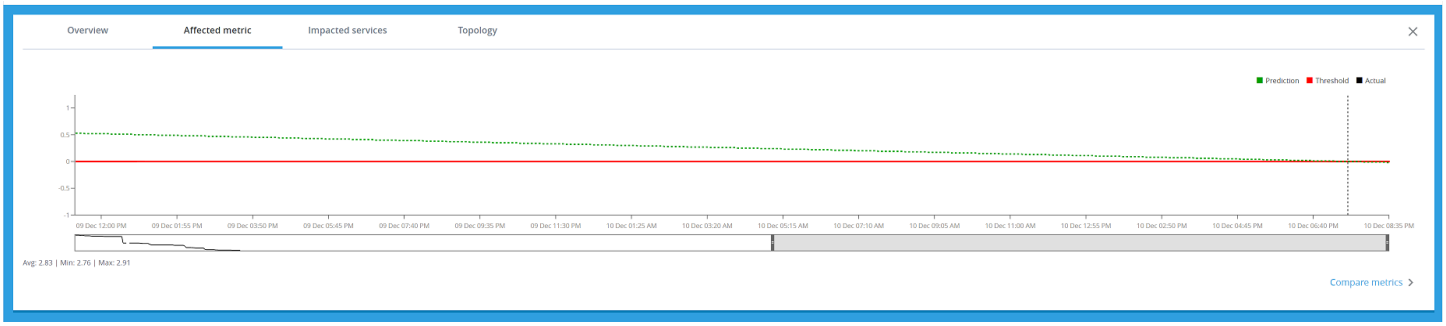
The Overview tab provides the information about the selected alarm that is useful for you to identify and triage the issue quickly. The tab contains information such as the alarm ID, alarm message, device name, creation and last updated dates, the prediction time, and so on.



Affected Metric Tab

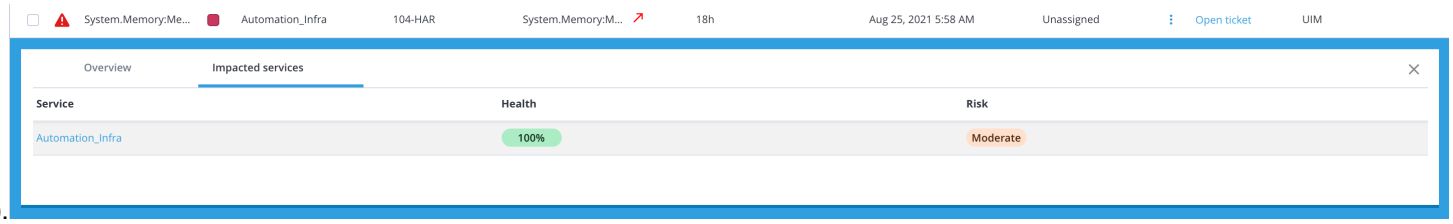
The **Affected Metric** tab shows the metric chart of the underlying metric. If the required fields are not available in the alarm, this tab is not shown. This tab shows a linear regression chart for a metric that is clickable. The metric chart displays the following values:

- The Affected Metric tab has a **Correlated Metrics** link that launches Performance Analytics from the context of an alarm and allows you to compare a single metric from different devices or multiple metrics from single or multiple devices. For more information about the metric charts, see Metric Charts Views.
- **Prediction:** The green color on the chart is the prediction chart. You can view prediction data for the last one week from the current time. Based on the past one-week data, the future trend is calculated. The future trend is until the prediction time or one week.
- **Actual**: The black color on the chart is the Actual chart. Indicates that the data is collected by DX Operational Intelligence from a data source.



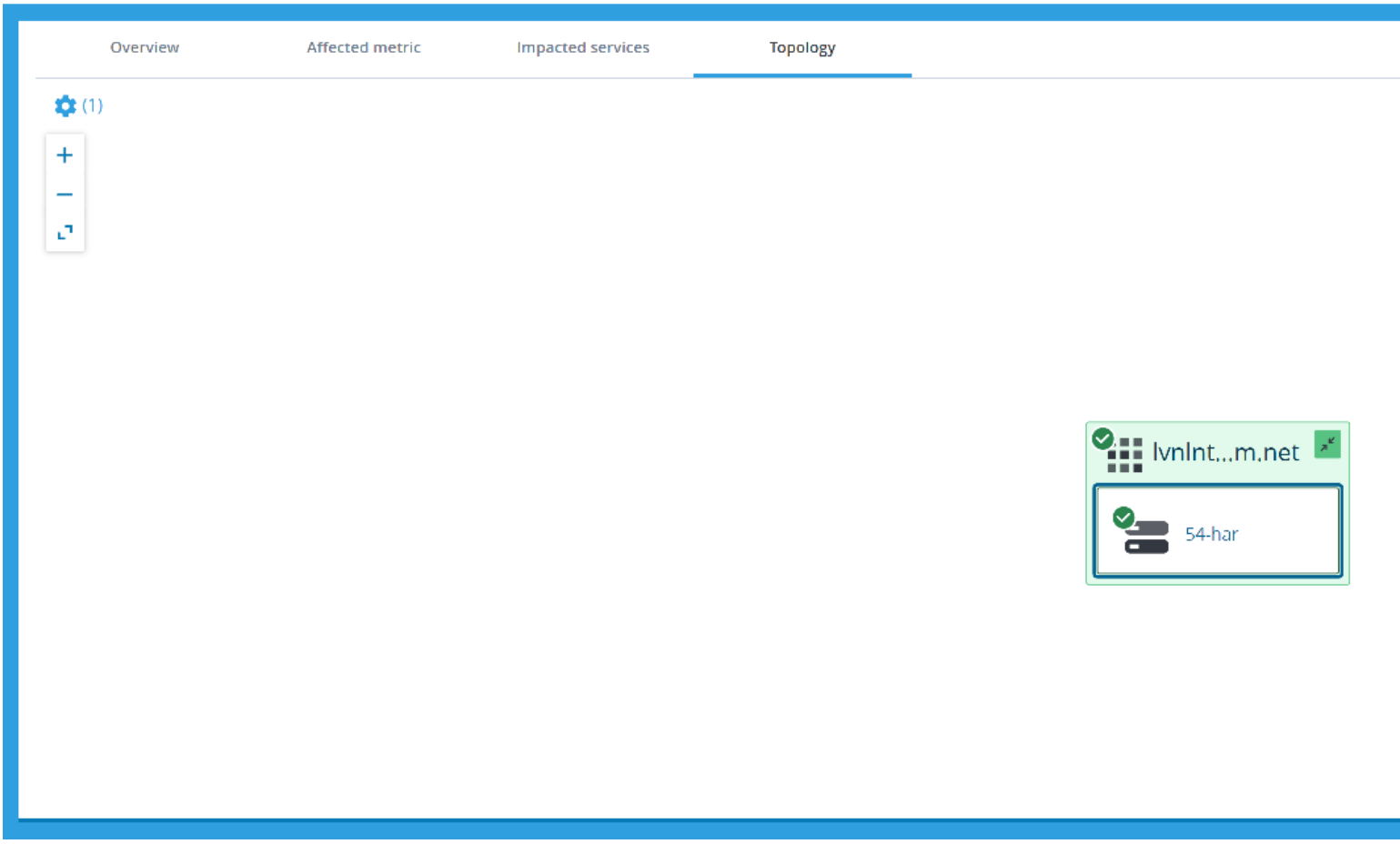
Impacted Services Tab

This tab provides details of the services that are impacted due to the selected alarm. Clicking on a service redirects to Service Analytics page. The table displays the impacted service metrics (such as users that are availing the service, actual service availability, and risk).



Topology Tab

The Topology tab displays the topology of the affected device. You can analyze the information in the context of both prediction and capacity alarms while triaging the issue.



Alarm Actions for Prediction Alarms

The alarm actions let you perform a specific action on an alarm. These actions are categorized as follows on the Predictive Insights page:

- Alarm Management
- Ticket Management
- Email Notification

Alarm Management

You can use the Alarm Management icon to manage alarms, acknowledge assigned alarms, and clear the assigned alarm.

Follow these steps:

1. Select multiple alarms from the Alarms table on the Predictive Insights page.
2. Click the **Alarm Management** icon to perform bulk operations.
   The **Alarm Management** pop-up menu appears.
3. Select the alarm action that you want to perform for one or more alarms.

- **Acknowledge:** Click to acknowledge one or more selected alarms.
- **Assign to:** Click the user to whom the alarm is to be assigned.
- **Clear:** Click to clear one or more selected alarms.
- **Hide:** Click to hide the alarm details in the database. The alarm in the user interface is grayed to indicate that the data is hidden in the database.
- **Un Assign Group Ticket:** Click to unassign a group ticket.
- **Un-Acknowledge:** Click to unacknowledge one or more selected alarms.
- **Un-assign:** Click to remove the assignment for an alarm.
- **Unhide:** Click to unhide the alarm.

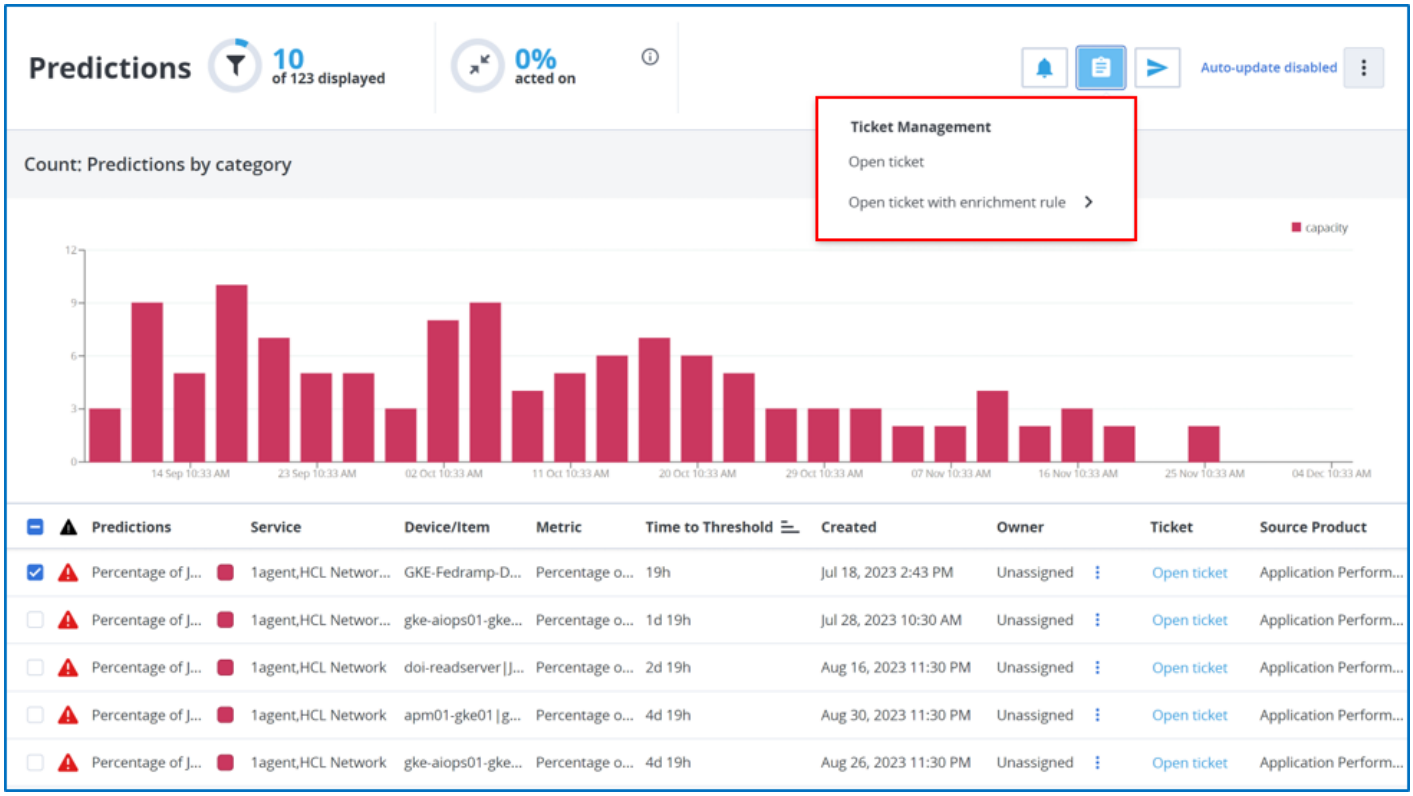
Alternatively, click the icon next to the **Owner** column of the required alarm to perform an alarm action for a single alarm.

Ticket Management

You can manage tickets directly from the Predictive Insights page.

Follow these steps:

1. Navigate to the **Predictive Insights** page.
2. Select the alarms that you want to open tickets for.
   The **Alarm Actions** section is enabled.
3. Click the **Ticket Management** icon and select the required option:



- **Open ticket:** This option uses the mapping rule that is associated with the channel for the ticket enrichment.
- **Open ticket with enrichment rule:** This option uses the mapping rule that you select in this list for enrichment instead of the rule that is associated with the channel.

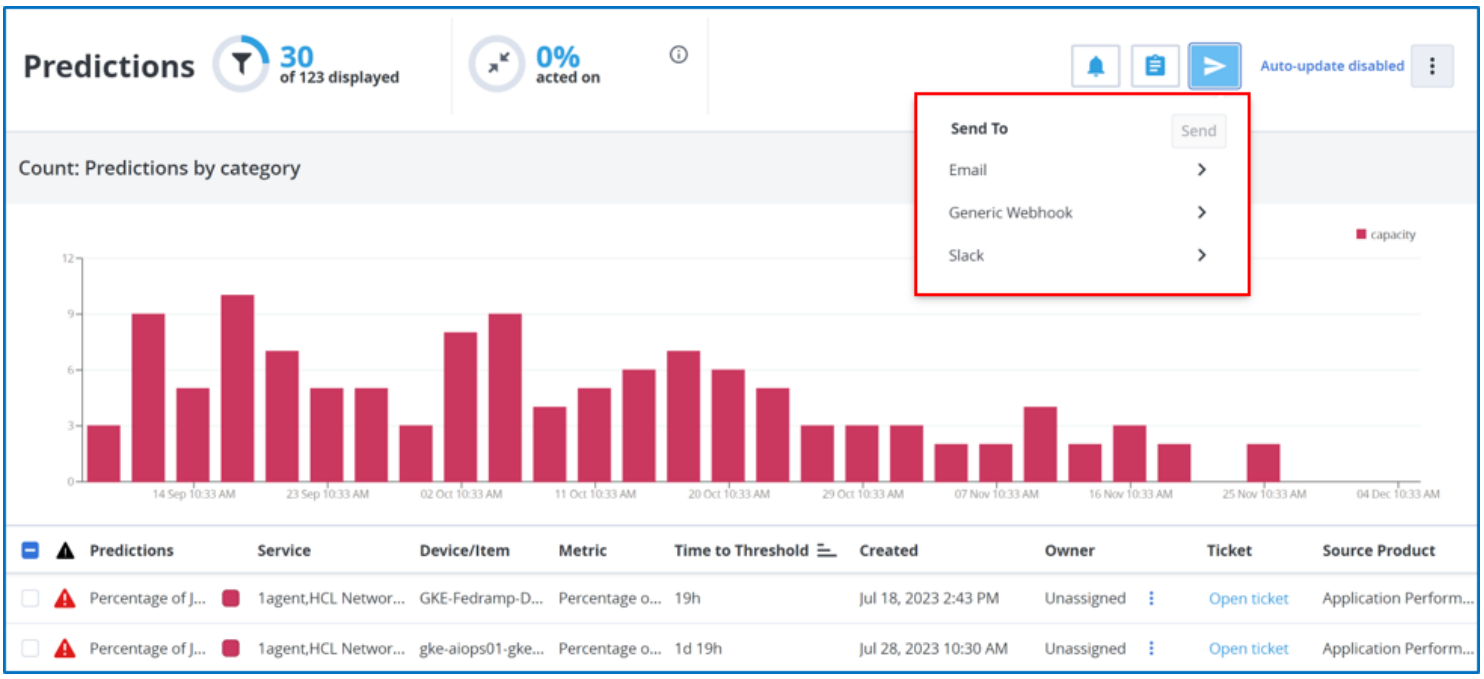
NOTE

- Alternatively, you can click the **Open Ticket** link in the **Ticket ID** column to open a ticket for a single alarm. The **Open Ticket** link is visible only when the ITSM notification channel is configured. To configure the channel, see the Channels section.
- You cannot create a ticket for an alarm with the severity as **Informational.**

4. Click the ticket ID which redirects you to the ITSM ticket management system. You can view the detailed information about the ticket. You can redirect back to the DX Operational Intelligence user interface by using the link provided in the ticket. Also, you can redirect back to DX Operational Intelligence by using the link sent through email, a ticket is created for the selected alarms in the ticketing system.

Trigger Channel

You can notify the users about alarms directly from the Predictive Insights page using the **Trigger Channel** icon. You can notify using one of the channels:



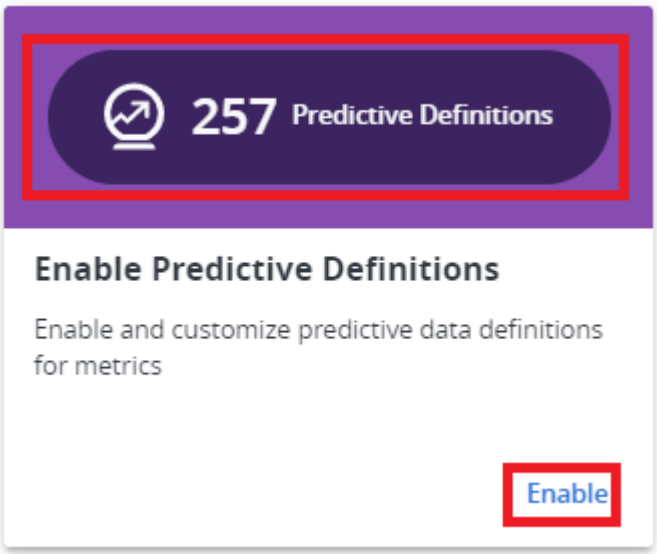
- **Email:** You can notify users about an alarm directly from the Predictive Insights page. You can send an email notification to one or more distribution lists to notify them about the alarm. Click the **Trigger Channel** icon, select **Email**, and then select the email channels from the list. For more information, see the Email Notifications section.

  NOTE
  You must configure the SMTP server to send emails to the recipient. If you do not configure the SMTP server, a success message appears but the email is not sent to the recipient.
- **Generic Webhook:** You can notify users about an alarm directly from the Predictive Insights page. Click the **Trigger Channel** icon, select Generic Webhook, and then select the webhook channels from the list. For more information, see the Configure Generic Webhook Channel section.
- **Slack:** You can notify users about an alarm directly from the Predictive Insights page. Click the **Trigger Channel** icon, select Slack, and then select the Slack channels from the list. For more information, see the Configure Slack Channel section.

Enable Predictive Definitions

The Enable Predictive Definitions tile allows you to configure the threshold value for each metric of the devices.

The **Enable Predictive Definitions** tile allows you to configure the threshold value for each metric of the devices. To enable and customize the predictive data definitions, Click **Enable** on the tile. You can also view the count of the predictive definitions on the



tile.

{"URL":["https://digital-oi/settings/data-definition"],"customLabelGetStarted":"Enable Predictive Definitions","description":"concept.dita_01bdeff9-7826-43cb-9e38-9de87c65891d"}

Data Definitions for Predictive Insights

The data definition appears based on the group you select for Predictive Insights (PI) on the **Settings** page. This page shows the metrics applicable to the groups that you select on the PI Settings page. You can set the threshold value for each metric of the devices that are categorized on the **Predict Insights** page. You can perform the following actions on this page:

- Set threshold value for metrics
- Update display name
- Set rollup algorithm

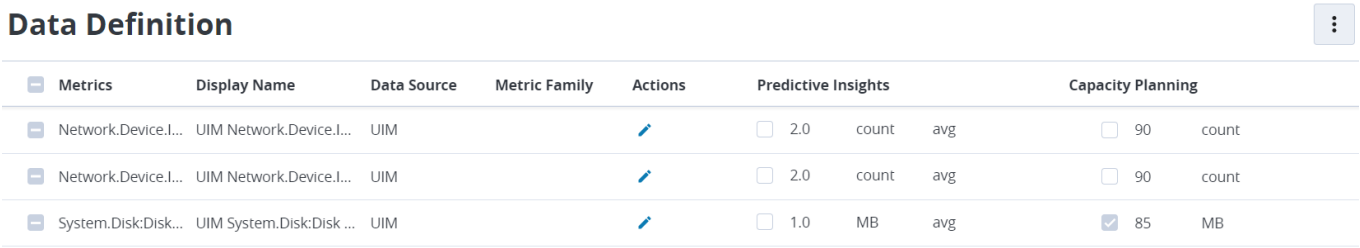
Configure Data Definition

To configure the data definition, follow these steps:

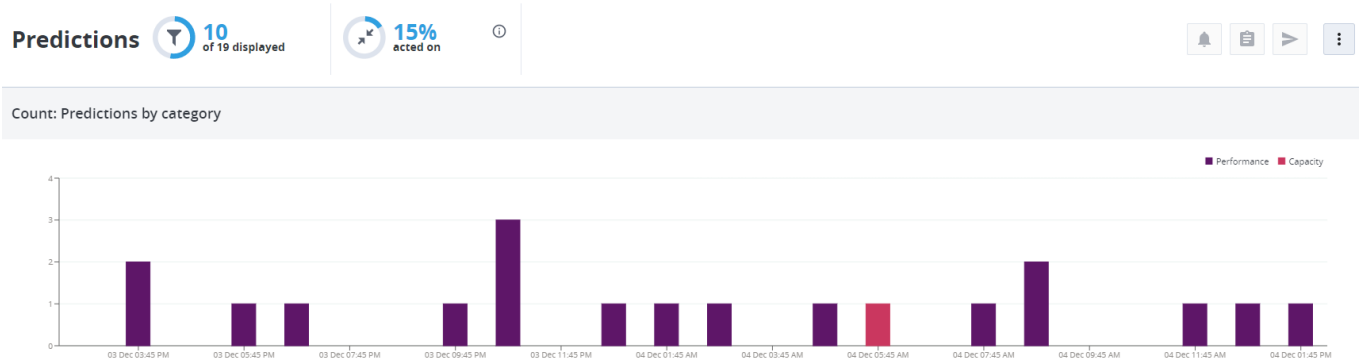
1. After you configure the groups, click **Save & Next**.
   The Data Definition page appears for that group.

2. **Set Threshold Value:** You can set the threshold value for a metric, based on which, the metrics are categorized as Performance and Capacity in the **Predictions** dashboard on the Predictive Insights page.
   a) Select **Edit** under the **Actions** column, and select the Predictive Threshold Setting under the **Predictive Insights column**.
   b) Enable and set the threshold value, and click **Save.**

      NOTE
      You can only set Critical value based on which all other values get populated.
   c) **Set Rollup Algorithm:** You can select one rollup algorithm value from the following options:
      - avg
      - sum
      - min
      - max
3. **Update display name:** Select **Edit** under the **Actions** column, and enter meaningful display name of a metric.
4. **Enable Analytics:** To display a summary (performance and capacity) of the predictions, you must select one or more metrics. To enable analytics, follow these steps:
   a) Select one or more metrics and click , **Enable Analytics**. The Metrics window appears.
   b) Select the total metric (Predictive Insights) to enable the metrics, and click Save. The metrics that are selected appear as follows:



   c) Click the **Predictive Insights** page to see the predictive insights of your selected metrics. The page appears as follows:



5. **Toggle Auto-update view:** You can use the toggle button to enable the **Auto-update view**.

Predictive Insights OOB Metrics

The Predictive Insights contains the below OOB metrics.

DX NetOps Performance Management Metrics

The following table displays the OOB metrics that are supported by DX NetOps Performance Management (DX NetOps PM):

Technology	CA PM MF/ Metric	Description	Default Threshold	Enable by Default.
LAN/WAN	portmfs/im_pctUtilization	Indicates the percentage of overall bandwidth that is utilized by LAN/ WAN.	80%	Yes
	portmfs/im_pctErrors	Indicates the percentage of errors out of good frames.	5%	Yes
	portmfs/im_pctDiscards	Indicates the percentage of frames that are discarded out of good frames.	10%	Yes
	portmfs/im_nonunicast	Indicates amount of non-unicast packets that are detected by network interface.	30000pkts	Yes
Router and Switch	portmfs/im_pctUtilization	Indicates the percentage of overall bandwidth that is utilized by routers and switch.	70%	Yes
	bufferpoolmfs/im_PercentUsedBuffers	Indicates the percentage of buffer that is utilized in the allocated buffers.	70%	Yes
	cpumfs/im_Utilization	Indicates the percentage of CPU utilized in the processor bandwidth.	70%	Yes
	portmfs/im_pctErrors	Indicates the percentage of errors out of total frames.	20%	Yes
	portmfs/im_pctDiscards	Indicates the total number of frames that are discarded out of total frames.	20%	Yes
System	cpumfs/im_Utilization	Indicates the percentage of CPU utilized in processor bandwidth.	60%	Yes
	virtualmemorymfs/im_Utilization	Indicates the percentage of the virtual memory that is utilized in the total virtual memory.	80%	Yes
	memorymfs/im_Utilization	Indicates the percentage of physical memory that is utilized in the total physical memory.	15%	Yes

	partitionsmfs/im_Percent Used	Indicates the percentage of system partition that is utilized in the partition size.	90%	Yes
	partitionsmfs/im_Percent Used	Indicates the percentage of user partition in the partition size.	90%	Yes
Wireless	Wireless Access Point/ discardsIn	Number of packets discard in.	10000pkts	Yes
	Wireless Access Point/ discardsOut	Number of packets discard out.	10000pkts	Yes
	Wireless Access Point/ errorsIn	Number of packets errors in.	10000pkts	Yes
	Wireless Access Point/ errorsOut	Number of packets errors out.	10000pkts	Yes
	Wireless Controller/ activeAccessPoints	Indicates the number of active Access Points that are associated with this Wireless Controller.	500pcs	Yes
	Wireless Controller/ clientsAssociated	Indicates the percentage of clients that are associated with the controller.	10000	Yes
SDN	SDN Tunnel / packetLossPercentage	Indicates percentage of SDN Tunnel packet loss.	15%	Yes
	SDN SLA Path / packetLossPercentage	Indicates the percentage of SDN SLA Path packet loss.	15%	Yes
	Virtual Interface/pctErrors	Indicates the percentage of the packets that has error.	20%	Yes
	Virtual Interface/ pctDiscards	Indicates the percentage of the packets that are discarded.	20%	Yes
	Virtual Interface/ pctUtilization	Indicates the percentage of the interface utilization.	70%	Yes
ADA	ADA/averageNRTT	Indicates the average Network Round Trip Time in Milliseconds.	300msec	Yes
	ADA/averageSRT	Indicates the average server response time in Milliseconds.	300msec	Yes
	ADA/averageTTT	Indicates the average total transaction time in Milliseconds.	300msec	Yes

	ADA/averageSCT	Indicates the average server connection time in Milliseconds.	300msec	Yes
	ADA/averageNCT	Indicates the average network connection time in Milliseconds.	300msec	Yes
	ADA/averageDTT	Indicates the average data transferred time in Milliseconds.	300msec	Yes
	ADA/averageEffectiveRTT	Indicates the average effective Network Round Trip Time in Milliseconds.	300msec	Yes
NFA	NFA/inbytes	Indicates the Bytes In per Minute.	750000bps	Yes

DX Application Performance Management Metrics

The following table displays the list of technologies that are used by DX Application Performance Management (DX APM):

Technology	APM Metric		Metric Description	Default Threshold	Enable by Default.
	Blame point metrics.				
		average_response_time_ms	Indicates the application's average response time in Milliseconds.	5000msec	Yes
		errors_per_interval	Indicates the number of Application/ Frontend/Backend errors.	8	Yes
	Resource Metrics				
	CPU				
		utilization_aggregate	Indicates the percentage of system CPU utilization.	80%	Yes
		utilization_process	Indicates the percentage of CPU utilized per process.	90%	Yes
		cpu_utilization_host	Indicates the percentage of CPU utilized per host.	90%	Yes
		processor_time	Indicates the percentage of CPU processor time.	90%	Yes
	GC Monitor				
		percentage_of_java_heap_used	Indicates the percentage of java heap that is used by GC Monitor.	90%	Yes
		percentage_of_time_spent_in_gc_during_last_15_minutes	Indicates the percentage of time that is spent in GC during last 15 minutes.	70%	Yes

		percentage_of_maximum_capacity_currently_used	Indicates the percentage of maximum capacity that is currently used by GC Monitor.	95%	Yes
	Others				
		tall_count	Number of Front-end Stalls	5	Yes
		memory_rounded	Indicates the percentage of total Memory Utilized.	90%	Yes
	Docker/Kubernetes				
		cpu	Indicates the percentage of CPU Utilized by Docker/ Kubernetes.	80%	no
		memory	Indicates the percentage of memory Utilized by Docker/ Kubernetes.	80%	no
		• dropped_packets_during_receive • dropped_packets_during_send	• Indicates the number of packets that are dropped. • Indicates the number of packets that are sent.	8	no
		errors_sent_and_errors_received	Indicates the number of errors that are sent and received.	8	no
		throttling_time_ns	Indicates the Throttling time in Nano seconds.	1000000ns	no

DX Unified Infrastructure Management Metrics

The following table displays the list of DX Unified Infrastructure Management (DX UIM) metrics:

Technology	Metric Name	Description	Metric ci Type	Default Threshold	Enable by Default.
CDM Probe	System.CPU:CPU Usage	Indicates the percentage of system CPU utilization.	1.5:1	90%	yes
	System. Disk:Disk Usage	Indicates the percentage of system disk utilization.	1.1:3	95%	yes
	System. Memory:Memory Usage	Indicates the percentage of system memory utilization.	1.6:2	80%	yes
	System. Memory:Physical Memory Usage	Indicates the percentage of system physical memory utilization.	1.6:7	80%	yes
	System. Memory:Swap Memory Usage	Indicates the percentage of system swap memory utilization.	1.6:9	70%	yes

Docker	Application.Container.Docker:Memory percent of available memory used	Indicates the percentage of available memory that is used by Docker.	3.56.1:12	80%	no
	Application.Container.Docker:CPU Usage	Indicates the percentage of CPU usage.	3.56.1:9	90%	no
Interface	MetricFamily.Interface:Utilization	Indicates the percentage of Interface utilization.	11.1:1	80%	no
	MetricFamily.Interface:PctErrorsIn	Indicates the percentage of errors In out of total frames.	11.1:28	5%	no
	MetricFamily.Interface:PctErrorsOut	Indicates the percentage of errors out of total frames.	11.1:29	5%	no
	MetricFamily.Interface:PctDiscardsIn	Indicates the percentage of incoming packets discarded.	11.1:25	10%	no
	MetricFamily.Interface:PctDiscardsOut	Indicates the percentage of outgoing packets discarded.	11.1:26	10%	no
	MetricFamily.Interface:PctCollisionsOut	Indicates the percentage of collisions out, out of the total frames.	11.1:65	5%	no
Wireless Interface	MetricFamily.WirelessInterface:PctDiscardsIn	Indicates the percentage of incoming packets discarded.	11.99:115	10%	no
	MetricFamily.WirelessInterface:PctErrors	Indicates the percentage of errors out of total frames.	11.99:30	5%	no
	MetricFamily.WirelessInterface:Bandwidth Utilization	Indicates the percentage of Bandwidth utilization.	11.99:23	80%	no

Monitored Inventory

The Monitored Inventory capability provides a unified inventory view of all entities available in DX Operational Intelligence. The unified view of all entities across the environment allows you to:

- Quickly locate and fix the monitoring or sub-optimally monitored devices.
- Manage device redundancies from different monitoring tools and understand the potential impact of a planned change event, which helps to plan the monitoring coverage.
- View all relevant details about the devices associated with an incident, especially their cross-domain correlated impact, and allows you to dive deeper into any aspect of those devices as required.

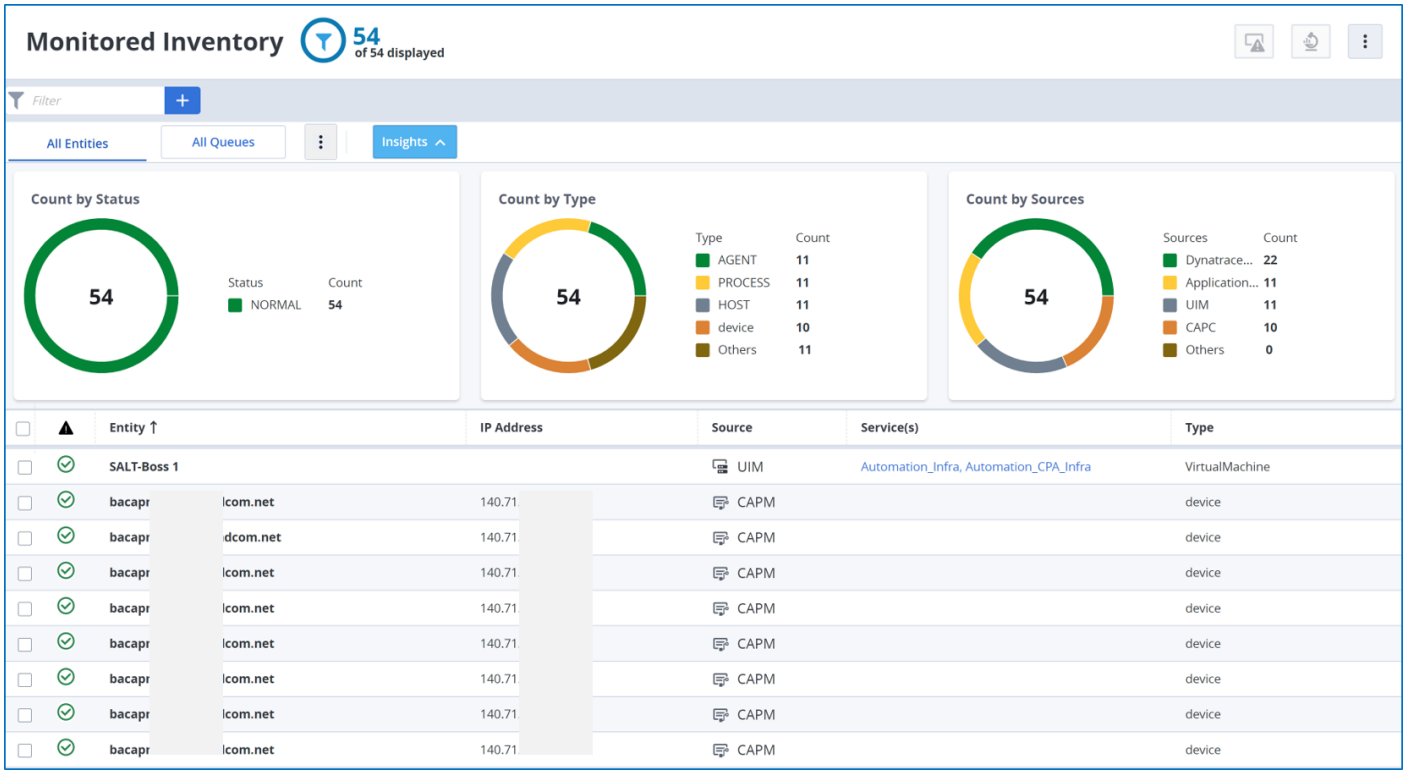
{"URL":["https://digital-oi/monitored-inventory"],"description":"concept.dita_93076ffe-a08f-40ae-b13c-7e608407954f","new":"","new_video":"","admin":"","troubleshooting":{"masterkb":"","text":"","URL":[]},"pendo":"","video":["https://www.youtube.com/watch?v=iKZJLnN8egE&amp"]}

Access Monitored Inventory

You can access **Monitored Inventory** in one of the following ways:

1. Open DX Operational Intelligence.
2. Click **Monitored Inventory** in the left navigation panel.

   The monitored inventory page is displayed.



3. You can also access the monitored inventory in the following ways:
   a) **Service Analytics:** View all entities in the context of services in the Widget View and Monitored View on the Service Details page. For more information, see Monitored Inventory View.
   b) **Situation Alarms:** Navigate from Situation Alarms to Monitored Inventory in the context of entities that are part of situation alarm. For more information, see Situation Alarms.

Monitored Inventory User Interface

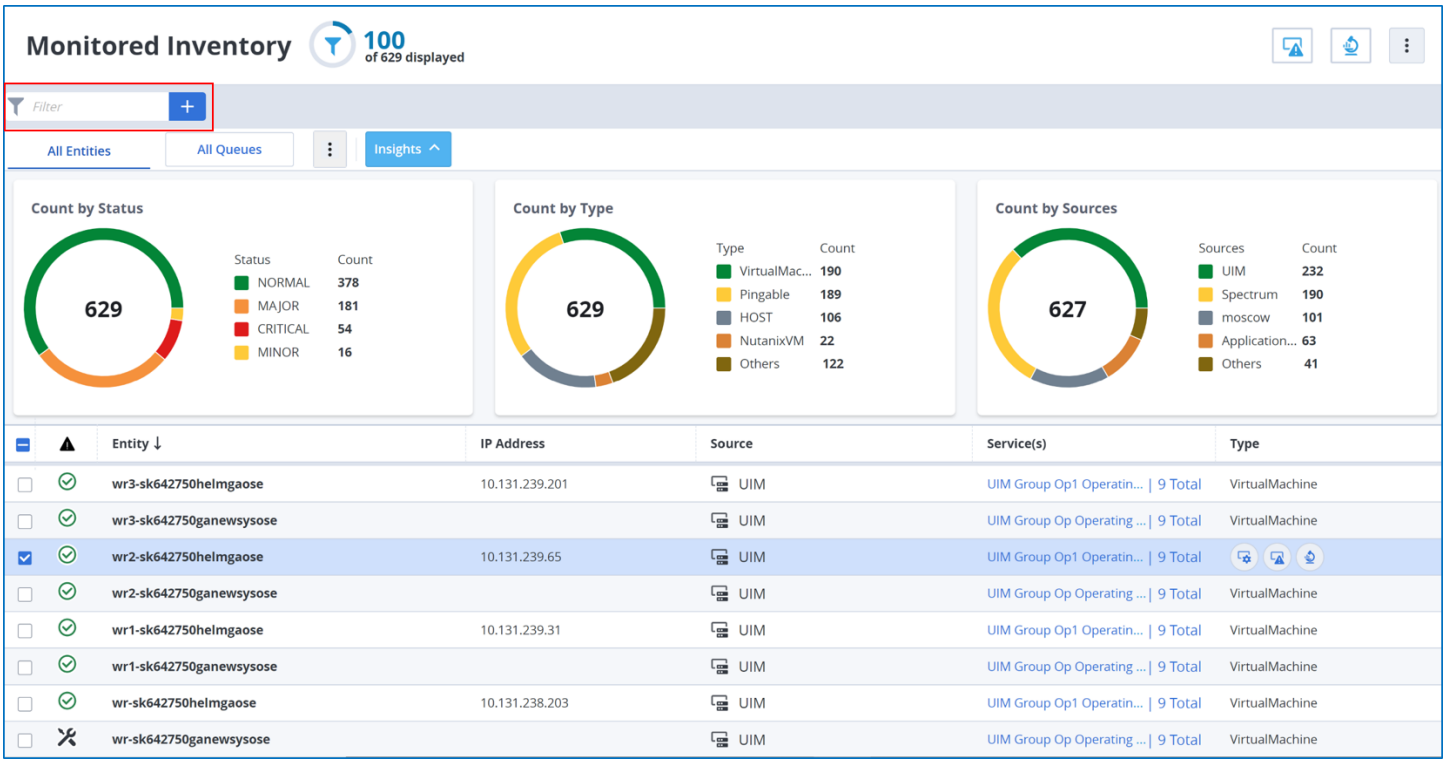
{"URL":["https://digital-oi/digital-oi/monitored-inventory"],"customLabelGetStarted":"Get Started with Monitored Inventory","description":"concept.dita_06b10393-f614-4973-8d5f-f77c3ba9c5bb"}

The **Monitored Inventory** page displays the following information:

- Filter
- All Queues
- Quick Filters
- Monitored Inventory Columns
- In-Context Navigation to Services View
- In-Context Navigation to DX Operational Intelligence Capabilities
- Additional Options

Filter

Use the filter attributes to filter the data in the monitored inventory table. You can enter the text or click the filter section to display the filter attributes.

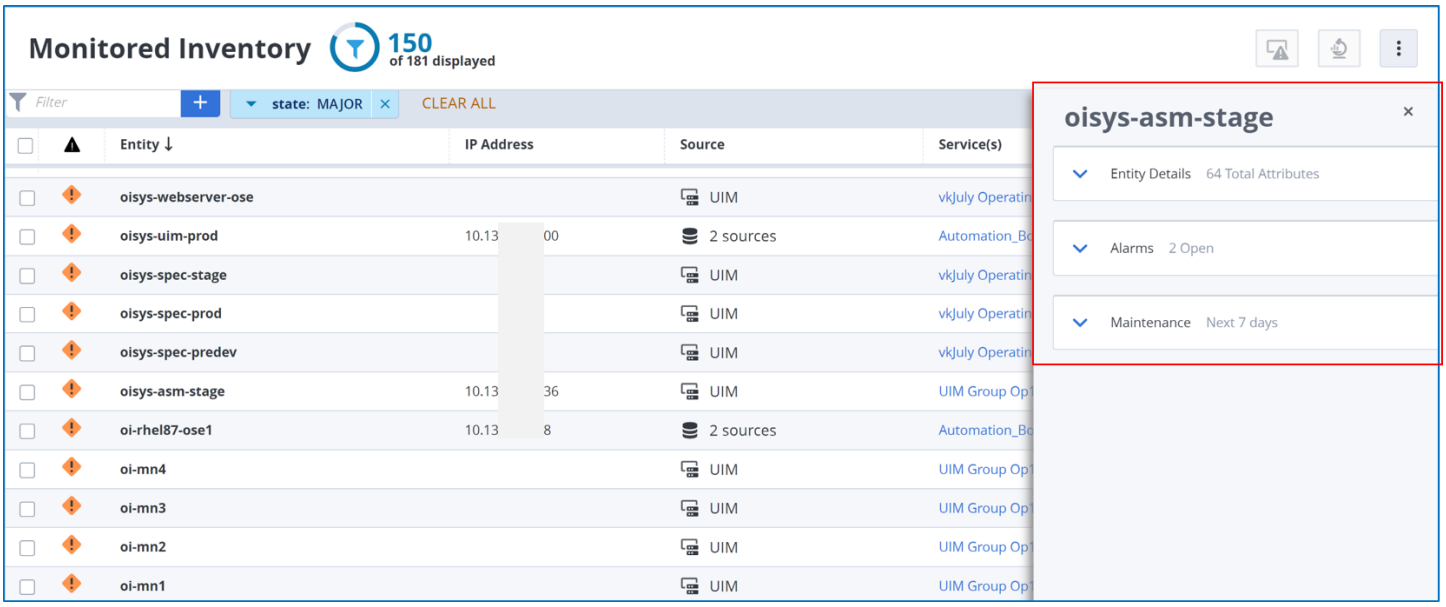


You can filter by:

- Entity
- IP Address
- Maintenance
- Service
- Situation Id
- Source
- State
- Type
- More: Click **More** to view additional attributes to filter by. This list also includes the option **Others** that you can use to filter by attributes that are not listed in this section but are available in the **Entity Details Panel** section.

Entity Details Panel

The **Entity Details** panel provides an in-context view of the selected entities to quickly understand entity details such as alarms, entity attributes, and maintenance. Click the non-hyperlinked area in any row to open the **Entity Details** panel.

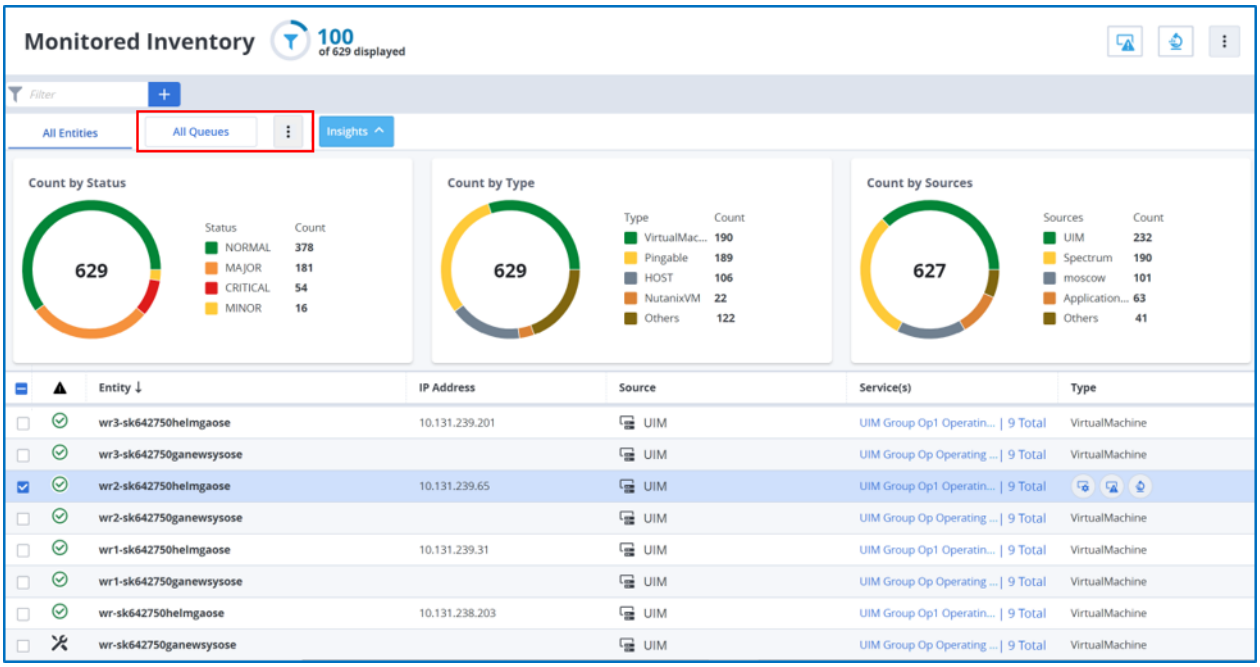


The **Entity Details** panel is categorized into:

- **Entity Details:** Displays all the attributes for the entity are displayed.
- **Alarms:** Displays details for the open alarms: **Severity, Message, Source, Created, Last Updated, Lifecycle**
  Click the message link to navigate to the **All Alarms** page.
- **Maintenance:** Displays the following information:
  - **State:** Displays if the entity is in maintenance mode.
  - **Next 3 windows:** Displays if there are any maintenance windows scheduled.

All Queues

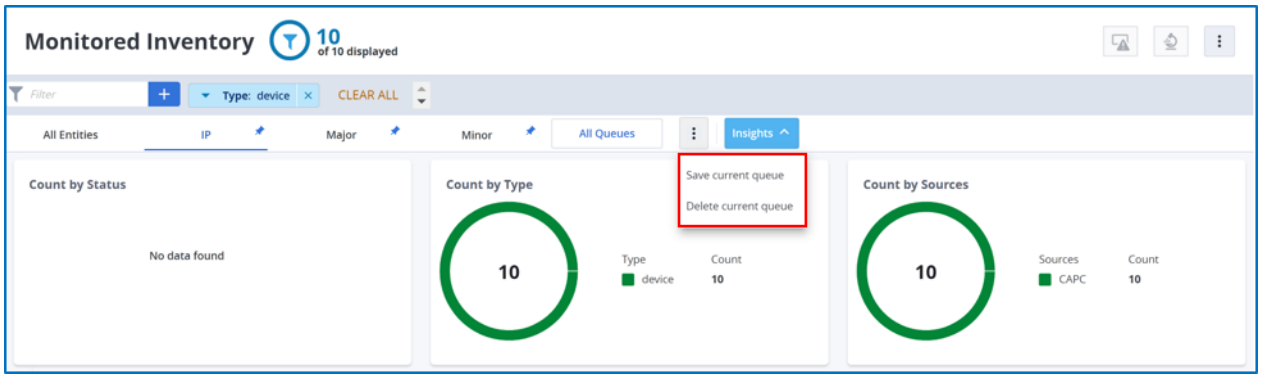
When you filter the entities, you can save the filter as a queue, and the **All Queues** dropdown lists the saved queues. You can pin the queue that you frequently use so that it gets featured as a tab. The queue is also added under **Pinned** in the **All Queues** list. A pinned queue has the icon next to the queue name and the unpinned queue has the icon next to the queue name. Click the icon to pin or unpin the queue.



You can save any number of queues but you can pin only a maximum of five queues.

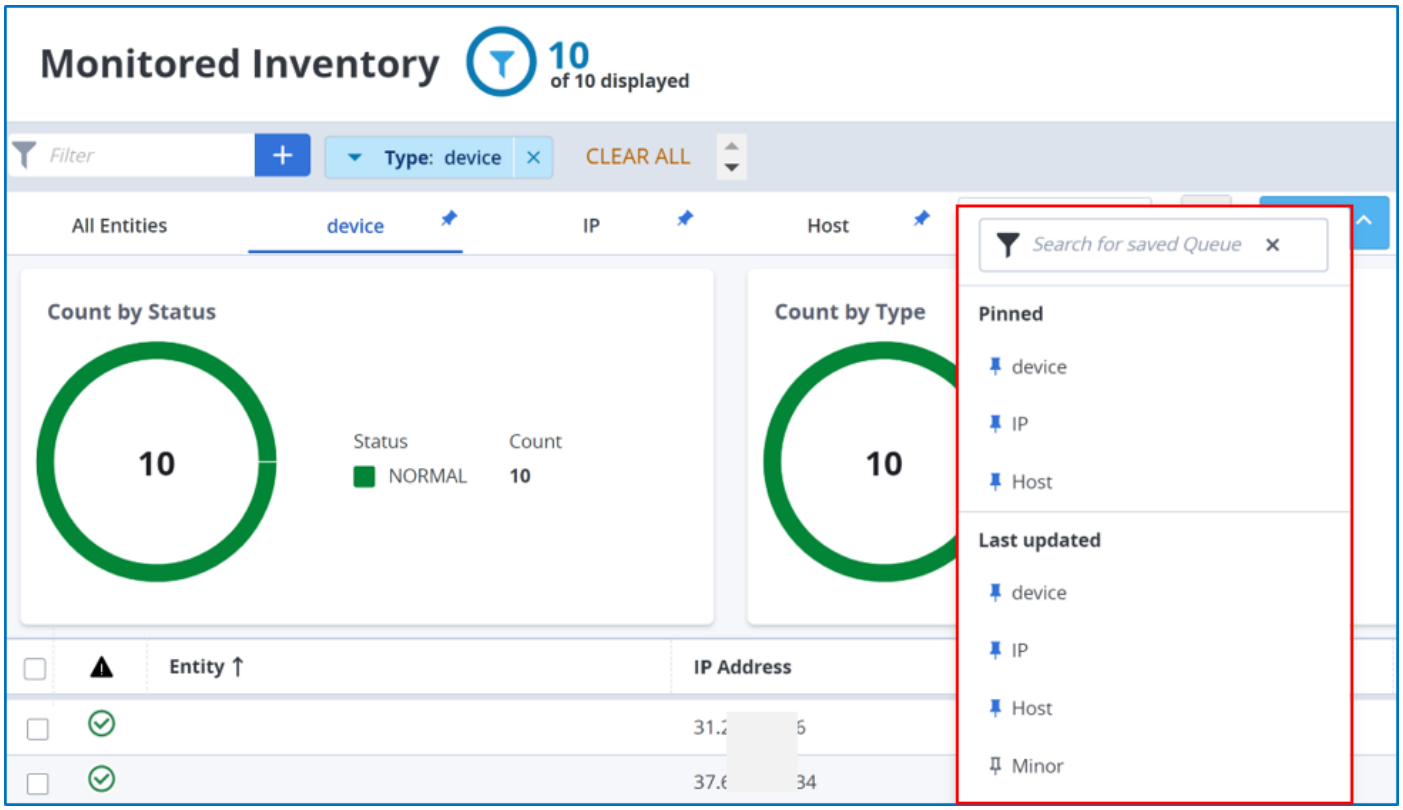
Follow these steps:

1. Navigate to the **Monitored Inventory** page.
2. Add the filter using the attributes.
3. Click the **Ellipsis** icon next to **All Queues** and select **Save Current Queue**.



NOTE
The **Save Current Queue** option is enabled only after you select the filter.

4. Provide the following information:
   - **Queue Name:** Enter a name for the queue.
   - **Set as Default:** Select this checkbox to set this queue as the default view. By default, the **All Entities** tab is the default view.
   - **Pin Queue:** Click the icon to pin the queue as a tab.
5. Click **Save**.
   The queue is added to the **All Queues** list. If you have pinned the queue, the queue is featured as a tab, and the queue is also listed under Pinned as shown in the illustration.

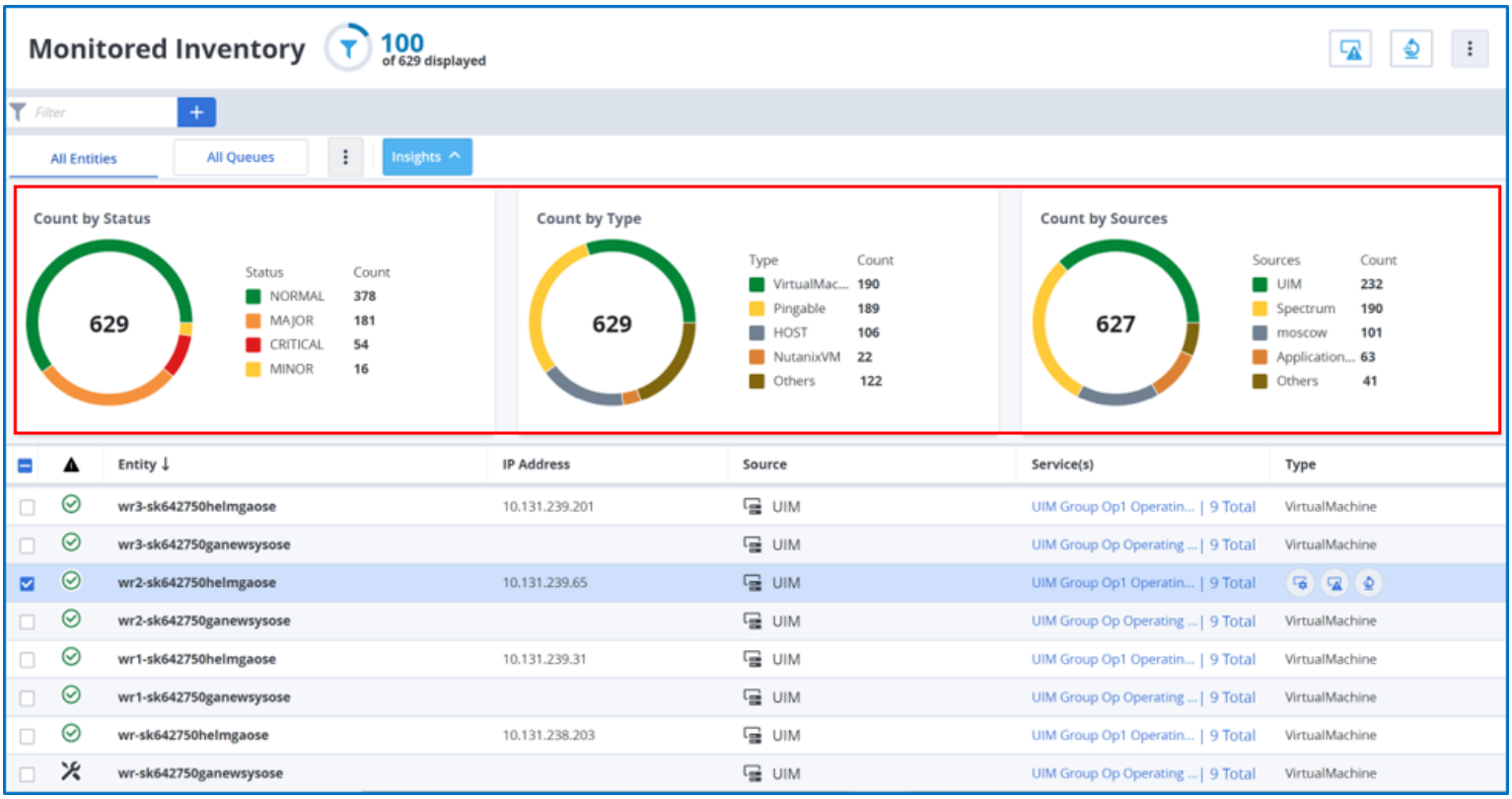


NOTE

- You can edit or delete only a pinned queue.
  - To edit, click the pinned queue (the featured tab), click the **Ellipsis** icon next to **All Queues,** and click **Save Current Queue**.
  - To delete, click the queue (the featured tab), click the **Ellipsis** icon next to **All Queues,** and click **Delete Current Queue**.
- You can set the queue as default while saving the queue. To set the queue as default later, edit the queue and select the checkbox.
- If you delete the default queue or if there are no default queues, then the **All Entities** tab becomes the default view.
- You can pin a queue while saving the queue or in the **All Queues** list and you can unpin a queue in the **All Queues** list. However, you cannot unpin the default queue.

Quick Filters

Using the status, type, and source widgets, you can understand the health of your entities at a glance and you can filter quickly to the problematic entity.



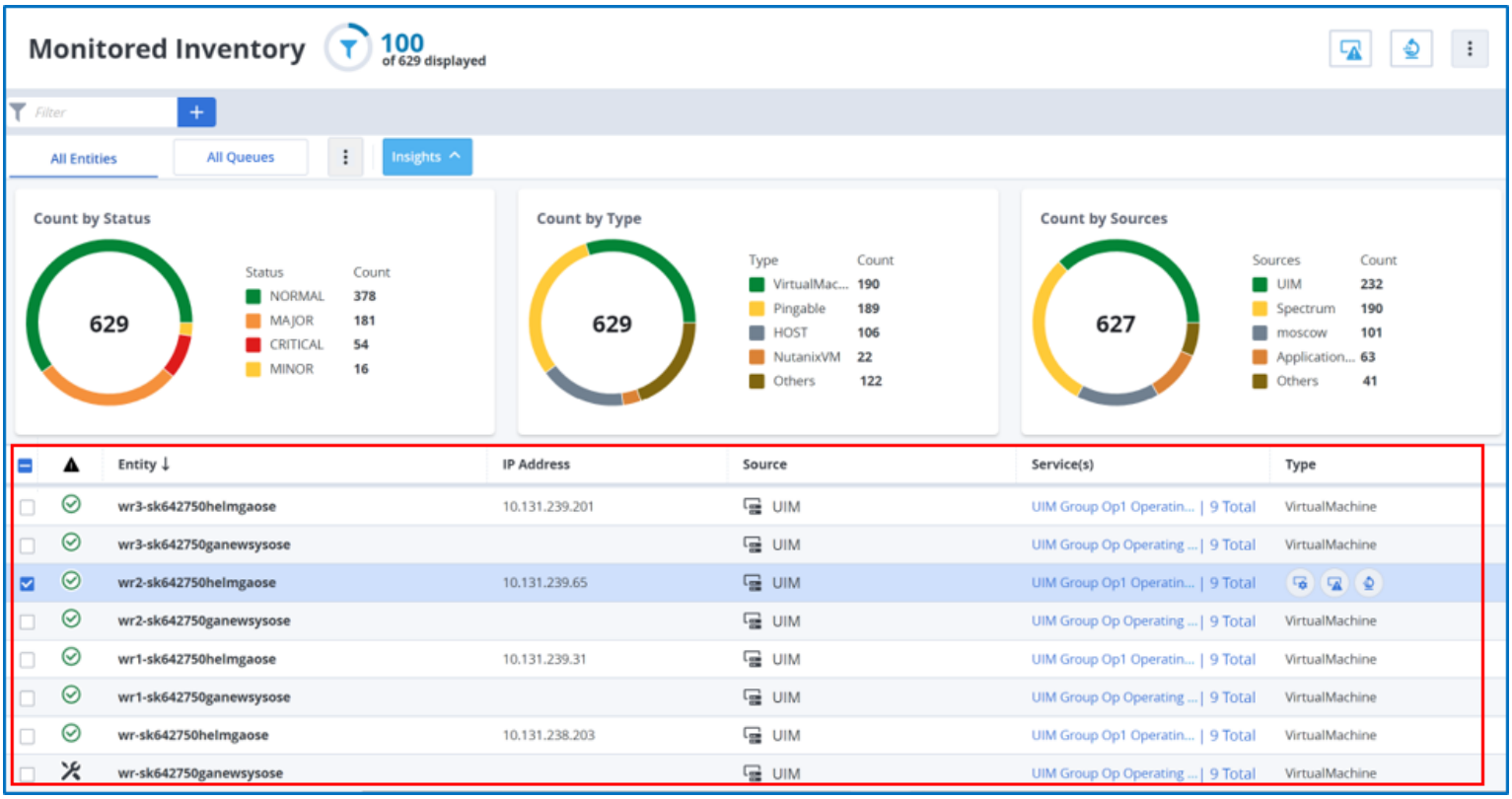
- **Count by Status:** Displays the count of the entities by status. **Values:** Normal, Major, Critical, and Minor.
- **Count by Type:** Displays the count of entities by entity type. The count is displayed for the top four types and the remaining entities are grouped under **Others**.
- **Count by Sources:** Displays the count of the entities by source. The count is displayed for the top four sources and the remaining entities are grouped under **Others**.

NOTE

- Click the **Insights** button to display or hide the widgets.
- Click the widget legend or the color on the donut chart to filter the data. To reset the view, click **CLEAR ALL** to clear the filter.

Monitored Inventory Columns

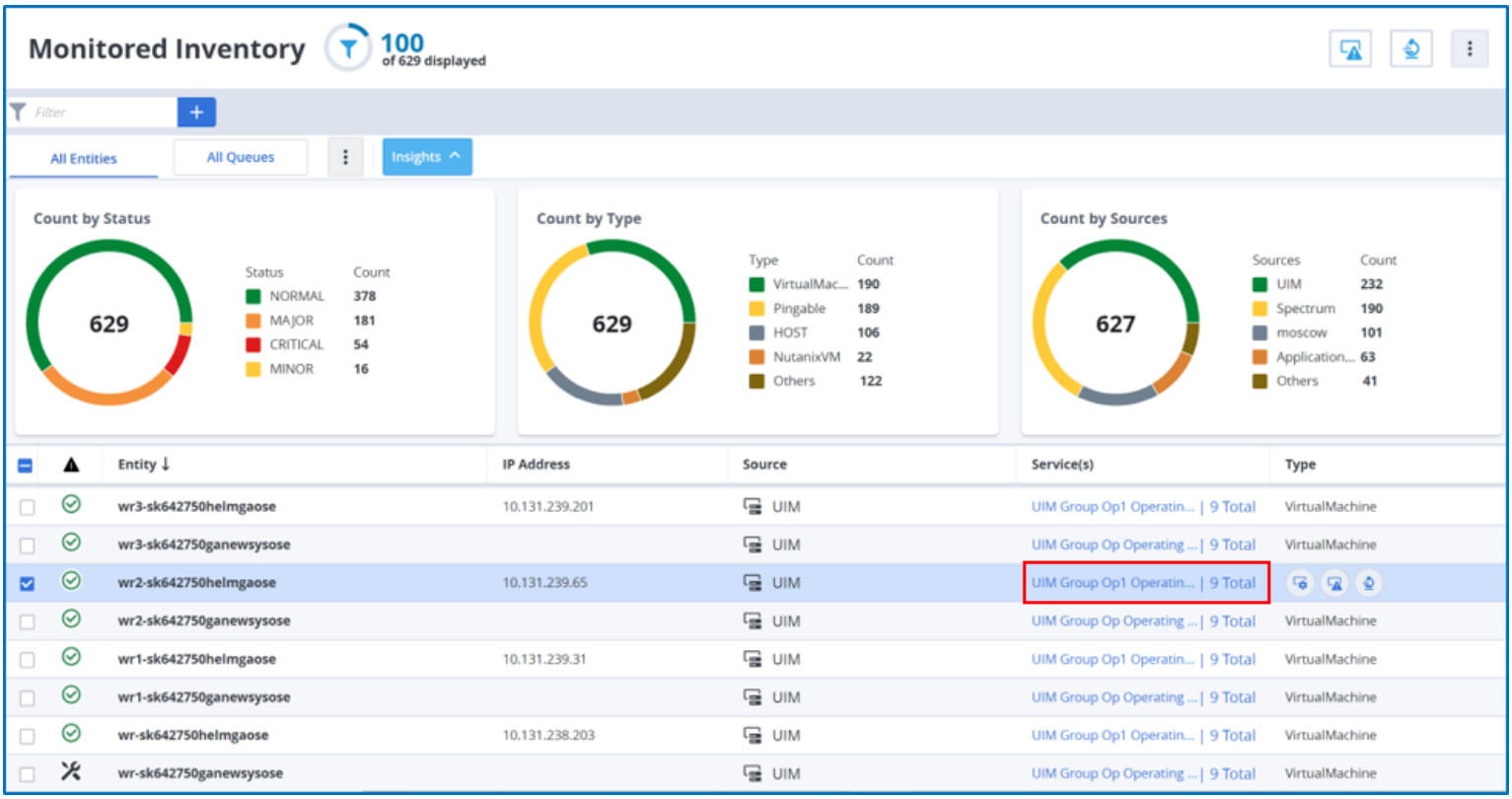
The Monitored Inventory table displays the following information:



Column	Description
Entity State	Specifies the entity state. You can sort the entity state by alarm severity in the following order: • Critical • Major • Minor • Informational • Normal
Entity	Indicates the device or application name.
IP Address	Displays the unique address of the device.
Source	Displays the product from which the entity is generated.
Services	Displays the service that is associated with the entity. Click Service which redirects you to the specific service on the Service Analytics page. You can also view the total number of services that the entity belongs to. Click the Count and search for the service using the filter option.
Type	Displays the type of entity. You can hover on the type name that launches Service Analytics, Alarm Analytics, Performance Analytics in the context of an entity.

In-Context Navigation to Services View

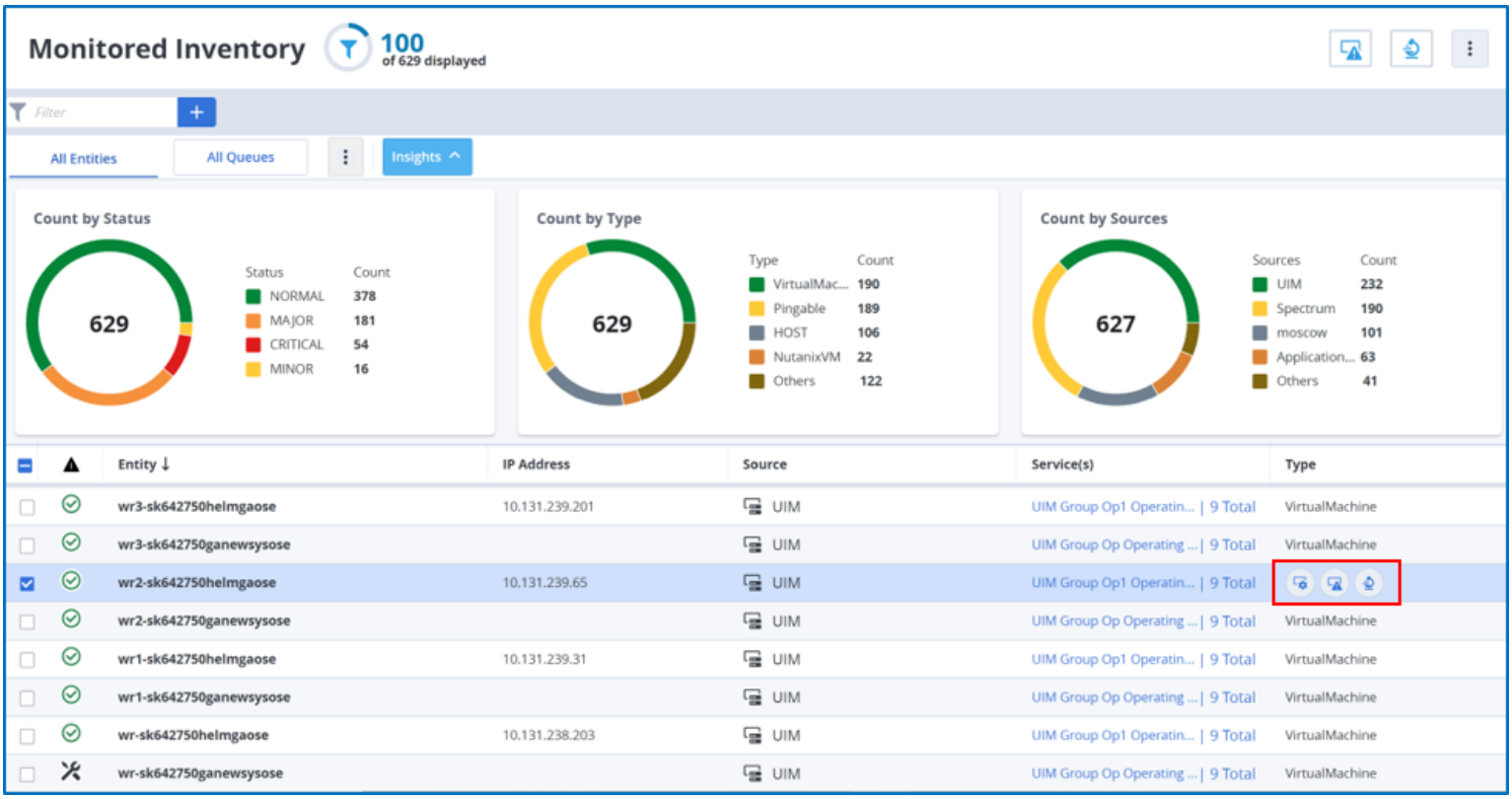
The Services column provides the total number of services that are associated with the selected entity.



Click the service link which redirects to the **Service Analytics** page with the entity filter applied.

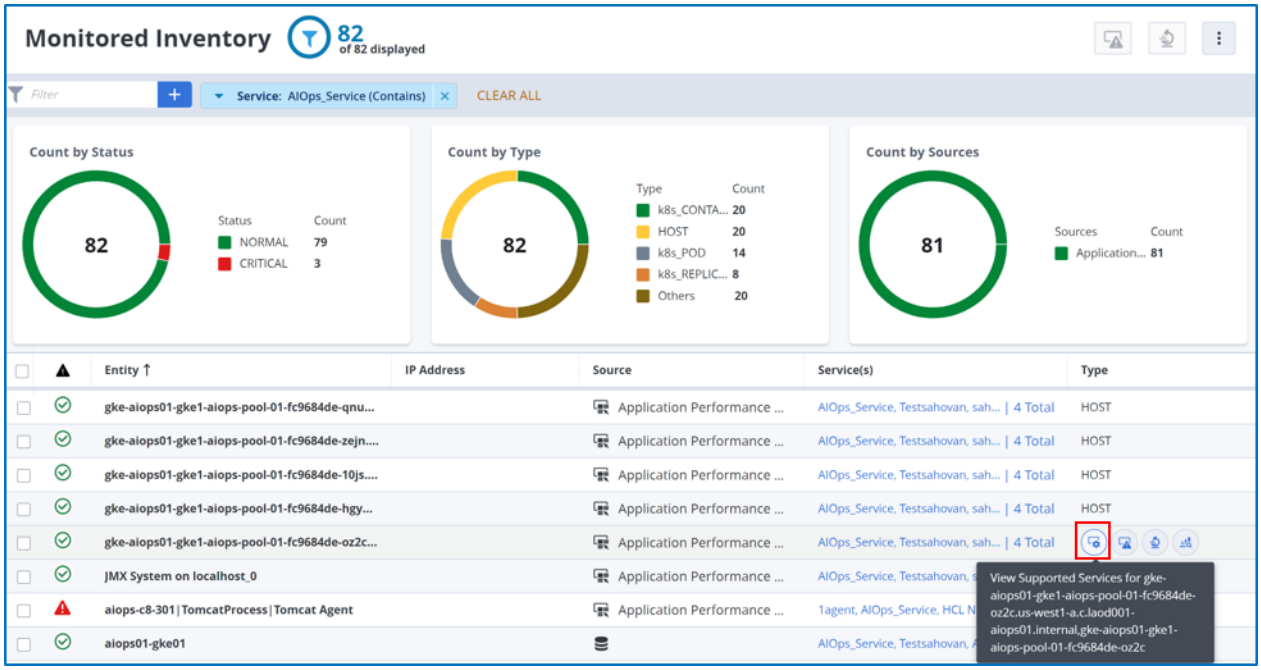
In-Context Navigation to DX Operational Intelligence Capabilities

You can navigate in-context to Service Analytics, Performance Analytics, Alarm Analytics, and Capacity Analytics. Click the respective capability icon to view the selected entity detail.

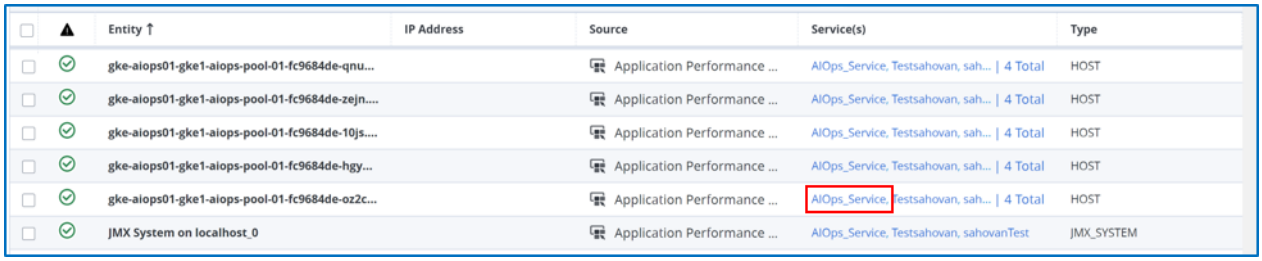


You can also navigate to Log Analytics from this page.

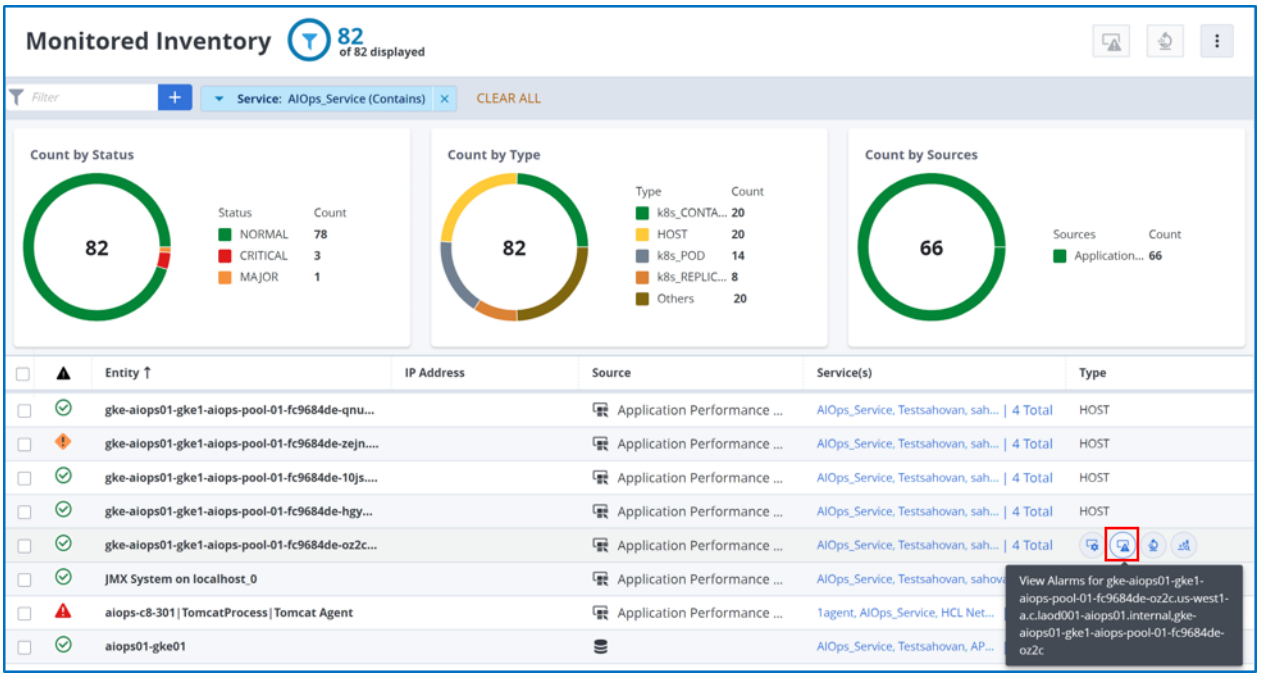
- **Service Analytics:** View the service and service details that are associated with an entity:
  To view the Service overview and its sub-services in the context of an entity, click the **Service** icon under the **Type** column to navigate to the **Services** page.



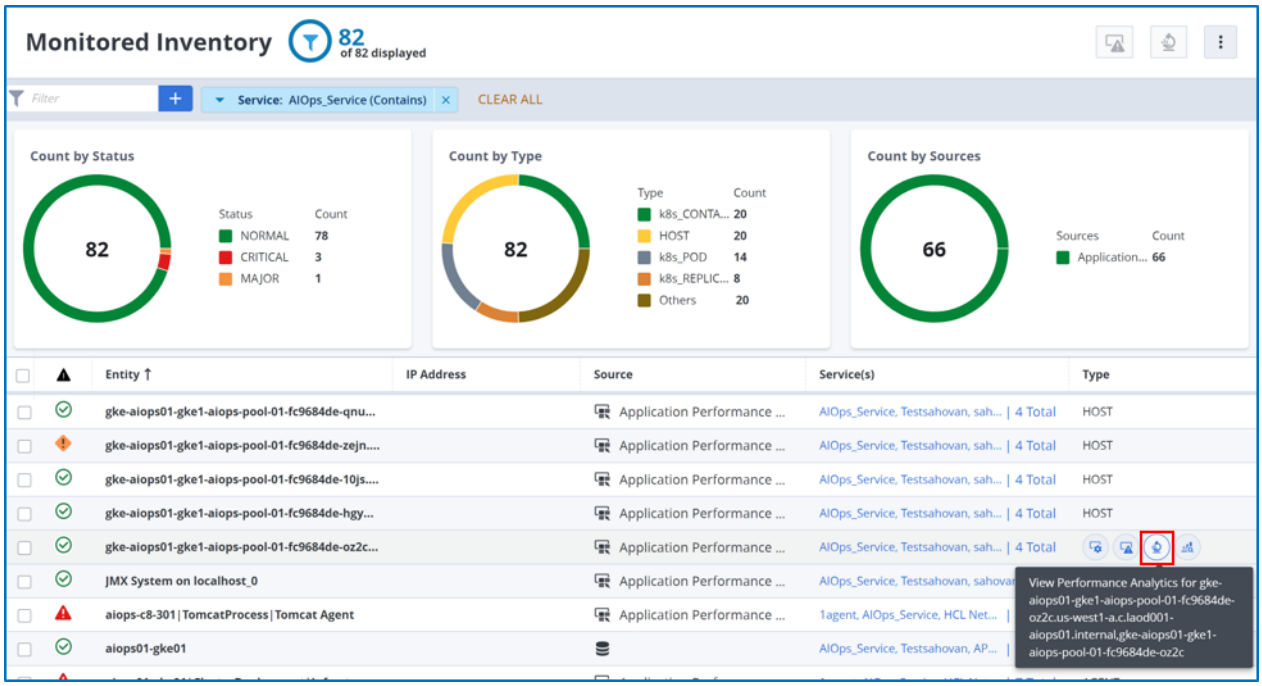
– To view service details in the context of an entity, click the **Service name** available under the Services column. You can also view the count of the services.



- **Alarm Analytics:** View the alarms for an entity:
  a. Click the **Alarm** icon under the **Type** column, which launches the **Alarm Analytics** page in the context of the entity. You can view the alarms of the entity and can perform the alarm actions based on your requirement.



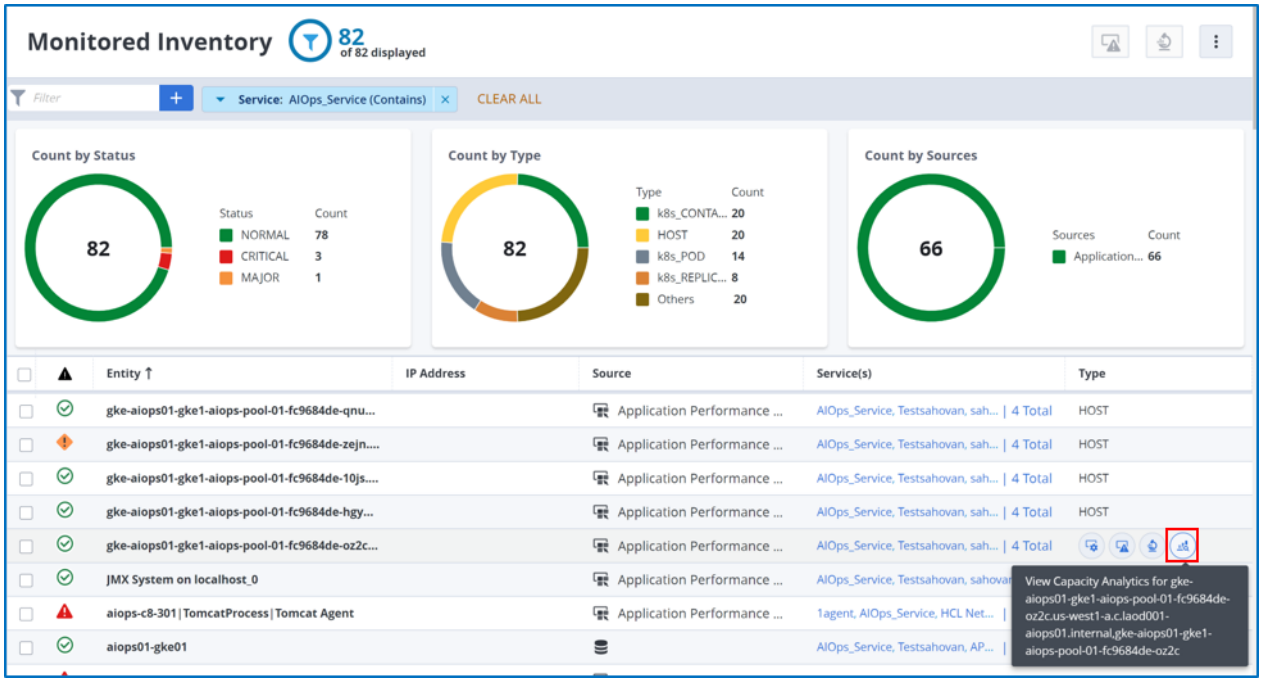
- **Performance Analytics:** View the metrics associated with the specific entity:
  a. Click the **Performance** icon under the **Type** column, which launches the **Performance Analytics** page in the context of the entity. You can select the metrics for that entity and can view the charts.



NOTE

When you click the **Performance Analytics** icon for an entity, the **Performance Analytics** panel is displayed on the right with the context maintained. The Performance Analytics data is displayed only for 6 Hours, 12 Hours, or 1 Day depending on your selection. You cannot select any other time period.

- **Capacity Analytics:** View the capacity for an entity:
  a. Click the **Capacity Analytics** icon under the **Type** column, which launches to the **Capacity Analytics** page in the context of the entity. You can view the capacity that is required for the resources such as CPU, memory, storage, and network for the operational continuity of the enterprise workloads.

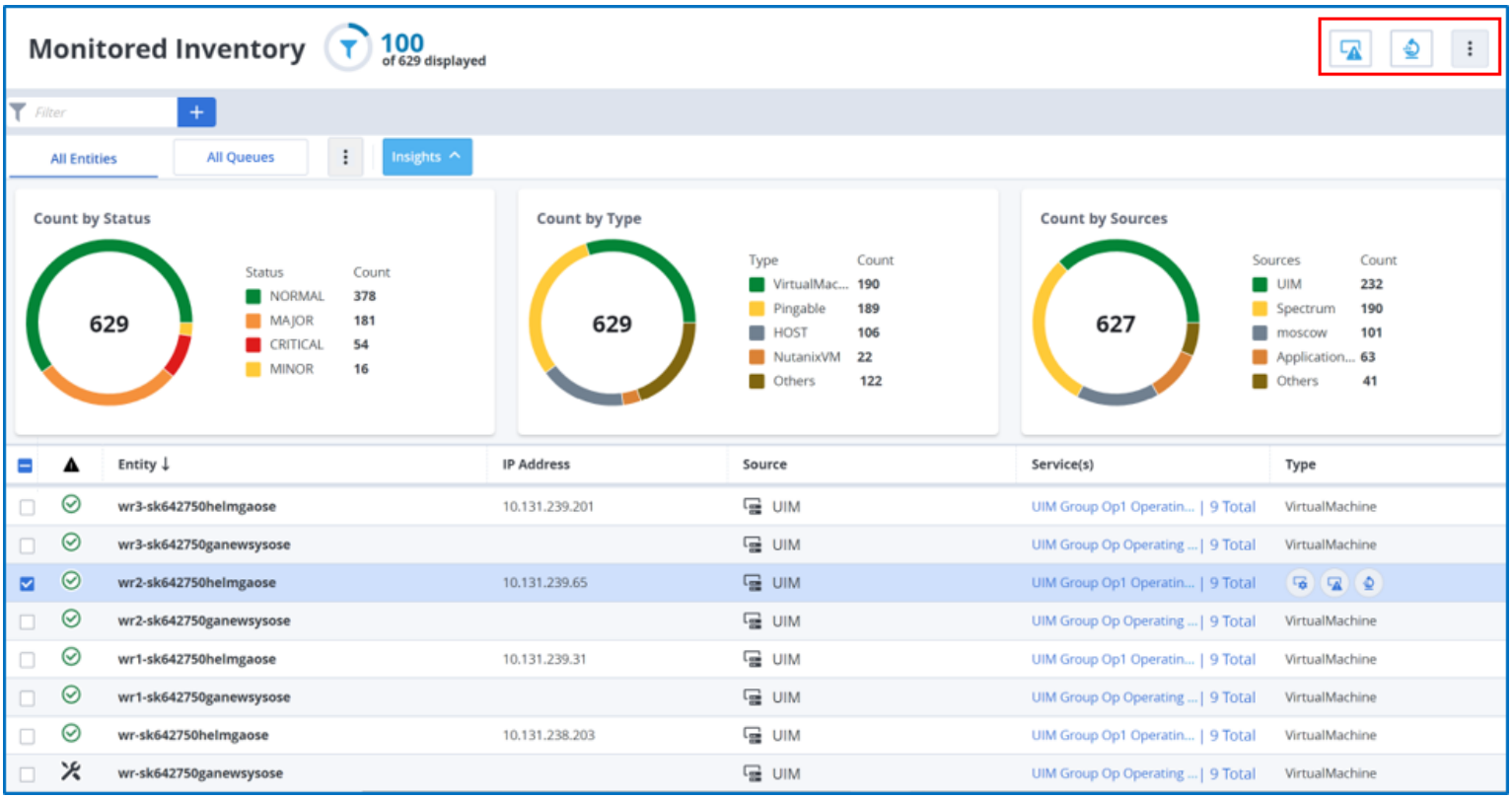


- **Log Analytics:** Click the **View Log Analytics** icon to navigate to DX OI - Logs.

NOTE
The **View Log Analytics** button is displayed for an entity or CI only when Log Analytics is one of the source products and the entity or CI is of the type **HOST**.

Additional Options

You can use the following options additionally:



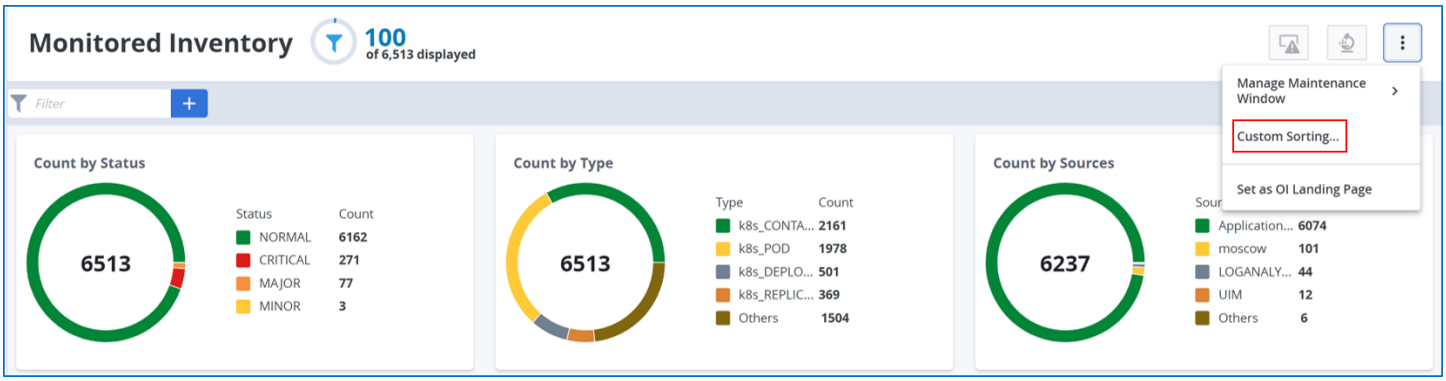
- **Three Dot Menu Option:** You can view the following options by clicking the

  icon.

  - **Manage Maintenance Window:** You can use this option to schedule a maintenance period for a required entity or multiple entities to perform any maintenance activity during this period. For more information, see the Manage Maintenance Window section.
  - **Custom Sorting:** The Custom Sorting option allows you to sort by the columns in ascending or descending order. For more information, see the Custom Sorting section.
  - **Set As OI Landing Page:** To make the **Monitored Inventory** view as your default landing page in DX Operational Intelligence, click **Set As OI Landing Page**.
- **Performance Analytics** : You can select up to 10 entities and click the icon to navigate to the **Performance Analytics** page in the context of the entities.
- **Alarm Analytics** : You can select up to 10 entities and click the icon to navigate to the **Alarm Analytics** page in the context of the entities.

Custom Sorting

The Custom Sorting option enables you to sort the columns in ascending or descending order. You can select **Sort by** and **Then by** based on the name of the columns.



- **Sort by:** Entity, Source, Type, and State
- **Then by:** IP Address, Source, Type, and State

Manage Maintenance Window

You can use the **Manage Maintenance Windows** option to schedule a maintenance period for a required entity or multiple entities in order to perform any maintenance activity during this period. You can apply a filter, select all entities on the screen and then create a maintenance window, add an entity to the maintenance window, and remove an entity from the maintenance window.

During this period, alarms are silenced and you would not receive any alarm notifications. However, you can still monitor all the entities of the devices. An alarm that raises within the maintenance time is tagged as in the Entity State column.

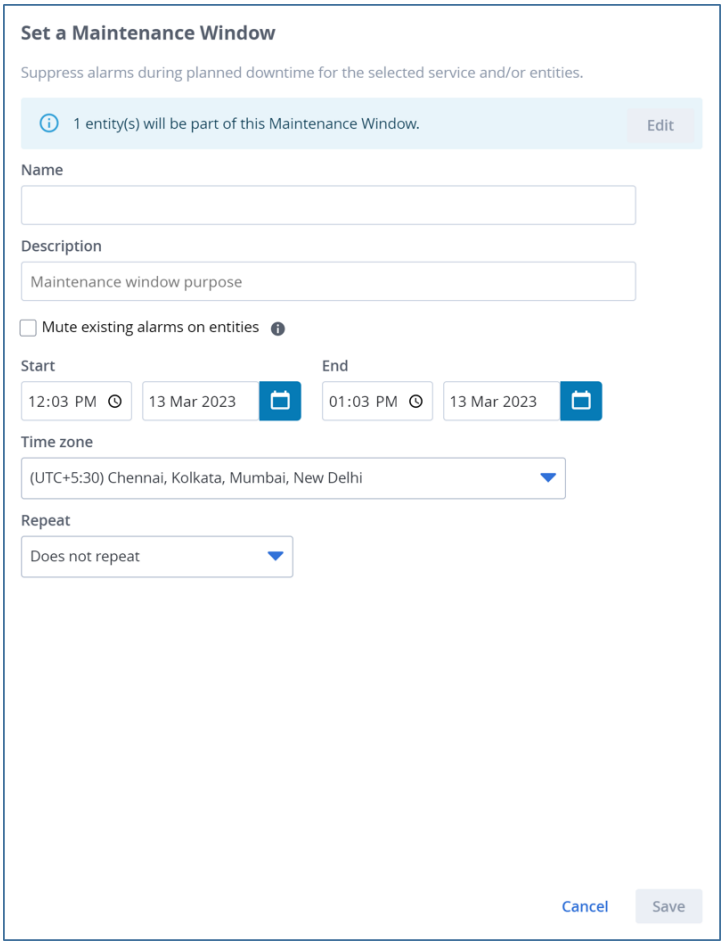
You can create a maintenance schedule for once, daily, weekly, or monthly time period. You are allowed to create multiple maintenance windows.

Follow the steps:

1. **Create a Maintenance Window:**
   a) Open the **Monitored Inventory** page.
   b) Click

   ,

   Manage Maintenance Window > Create.
   c) Enter the following details to schedule the maintenance window.



   d) **Name:** Enter a name for the schedule.
   e) **Description:** Enter a description for the schedule.
   f) **Mute Existing alarms on entities**: Select this option to mute the existing open alarms that were raised before or during an active maintenance period.

   When you select this option, all updates to the existing alarms which are part of this maintenance window, are muted during the maintenance window. They are restored to their original state when the maintenance window ends. For example, the alarm is restored to the open state if the alarm is not closed during the maintenance period.

   During the maintenance period, the muted alarms are greyed out, and a maintenance icon is also displayed for those alarms. If multiple maintenance windows are scheduled at the same time on the same entity, and if this option is selected in at least one of the schedules, then the alarms for all the overlapped schedules are muted.

If you end an active maintenance window or delete a window, the alarms are restored to their original state, that is, the alarms come out of the maintenance.

g) **Start** and **End**: Set the start and end times of the schedule. This field defines the duration of the schedule.
h) **Time zone:** Select the time zone for the schedule.
i) **Repeat:** Select the **Custom** option if you want a recurring maintenance schedule.
j) **Every:** Set the time period based on Days, Weeks, or Months.
k) **End:** Set the end time for the recurring maintenance window using the calendar. If you do not want to end the recurring maintenance window, select **Never**.
l) Click **Save**.

   Your maintenance window is created.

2. **Add Entity to Maintenance Window:**
   a) Select entities you want to add to the maintenance window on the **Monitory Inventory** page.
   b) Click

      ,

      Manage Maintenance Window > Add to.
   c) Select the maintenance window and click **Add**.

      The entity is added to the maintenance window.
3. **Remove Entity from Maintenance Window**:
   a) Select entities you want to remove from the maintenance window on the **Monitory Inventory** page.
   b) Click

      ,

      Manage Maintenance Window > Remove from.
   c) Select the maintenance window from which you want to remove the entity and click **Remove**.

      The entity is removed from the maintenance window.

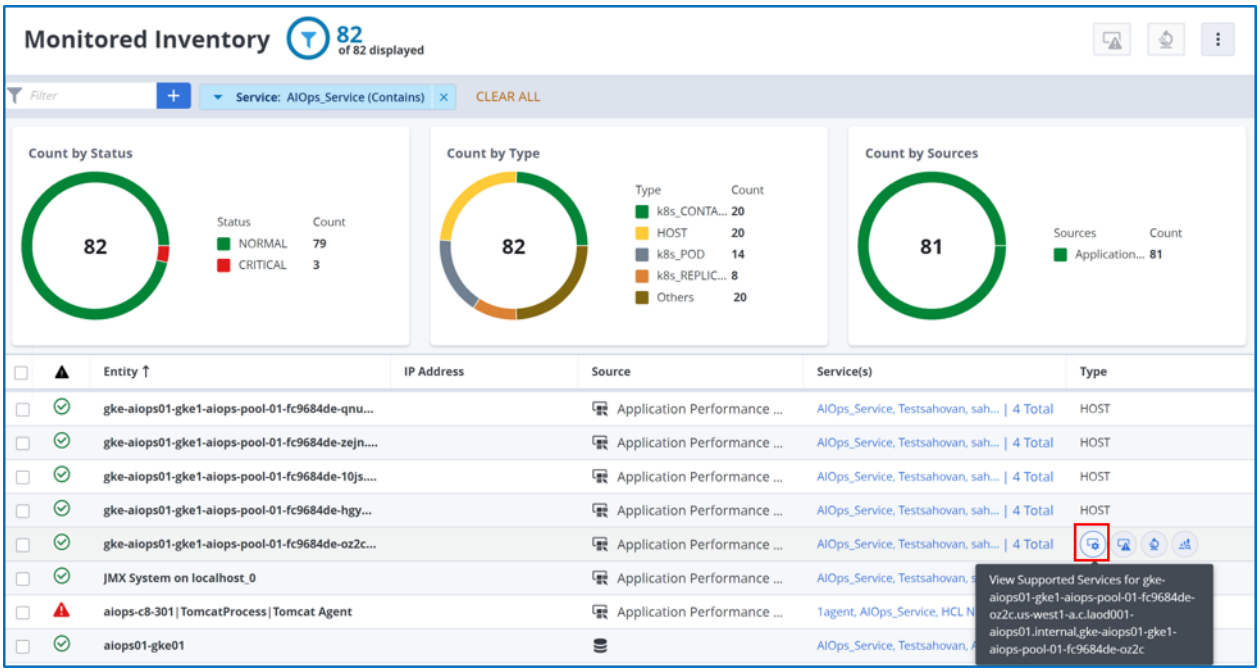
Monitored Inventory Workflow

As a tools administrator, I want a unified view of all the entities for my environment and see their details and interdependencies, such as services, generated alarms, metrics, and capacity.

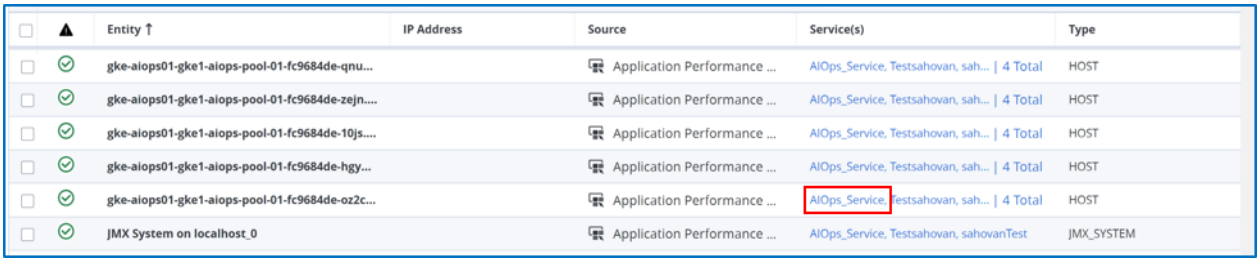
Follow these steps:

1. Log in to DX Operational Intelligence.
2. Click **Monitored Inventory** in the left navigation pane.

   The entities are populated for your environment.
3. Use the **filter** option to search for your entities and sort the columns using the **Custom Sorting** option.
4. View the entities and perform the following actions:
   a) View the service and service details that are associated with an entity:
      1. To view the Service overview and its sub-services in the context of an entity, click the **Service** icon under the **Type** column to navigate to the **Services** page.

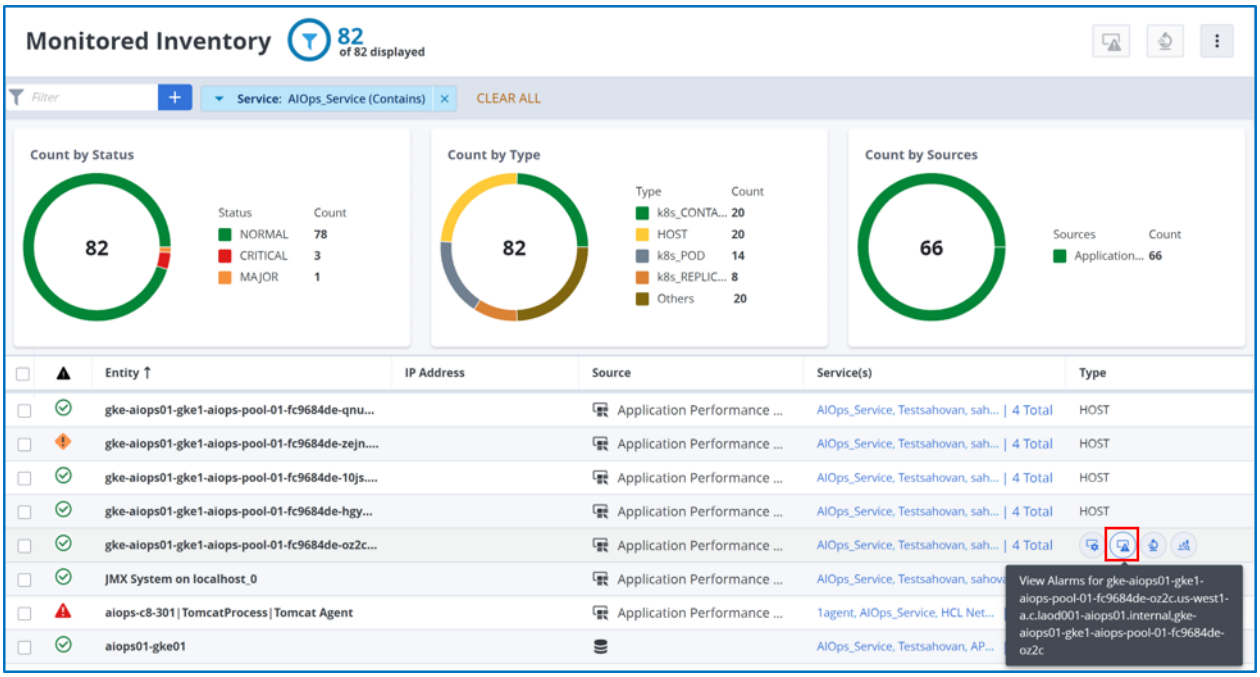


2. To view service details in the context of an entity, click the **Service name** available under the Services column. You can also view the count of the services.



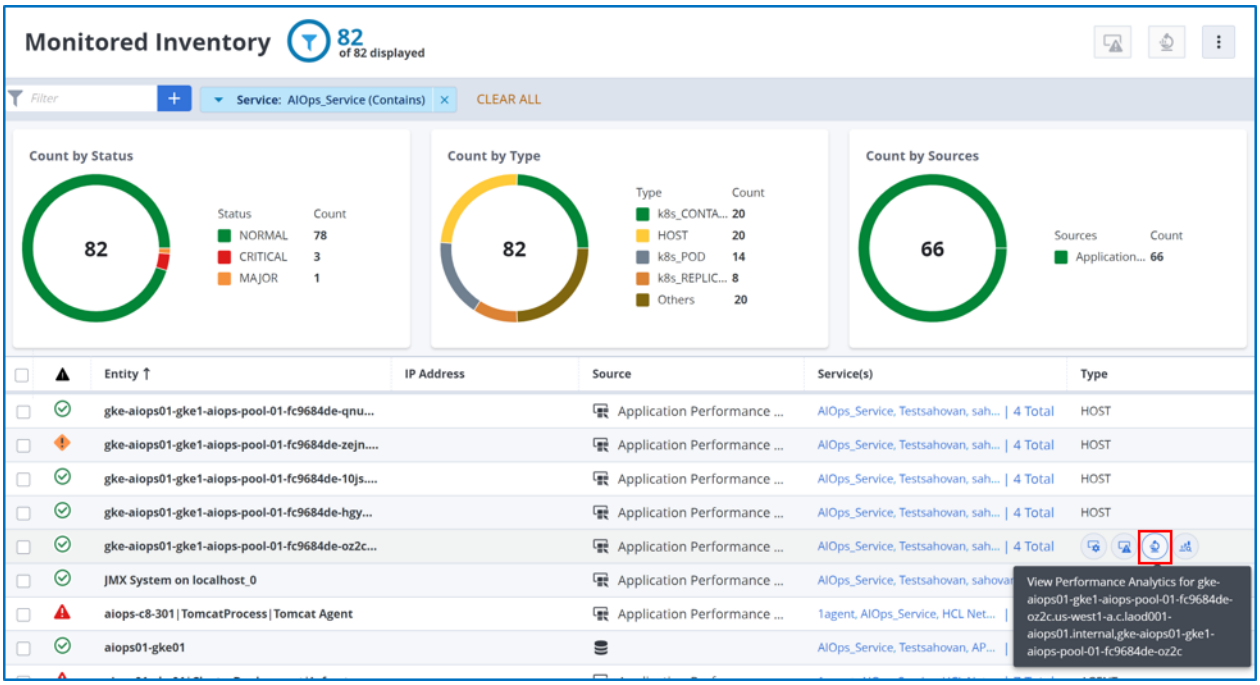
b) View the alarms for an entity:

1. Click the **Alarm** icon under the **Type** column, which launches the **Alarm Analytics** page in the context of the entity. You can view the alarms of the entity and can perform the alarm actions based on your requirement.



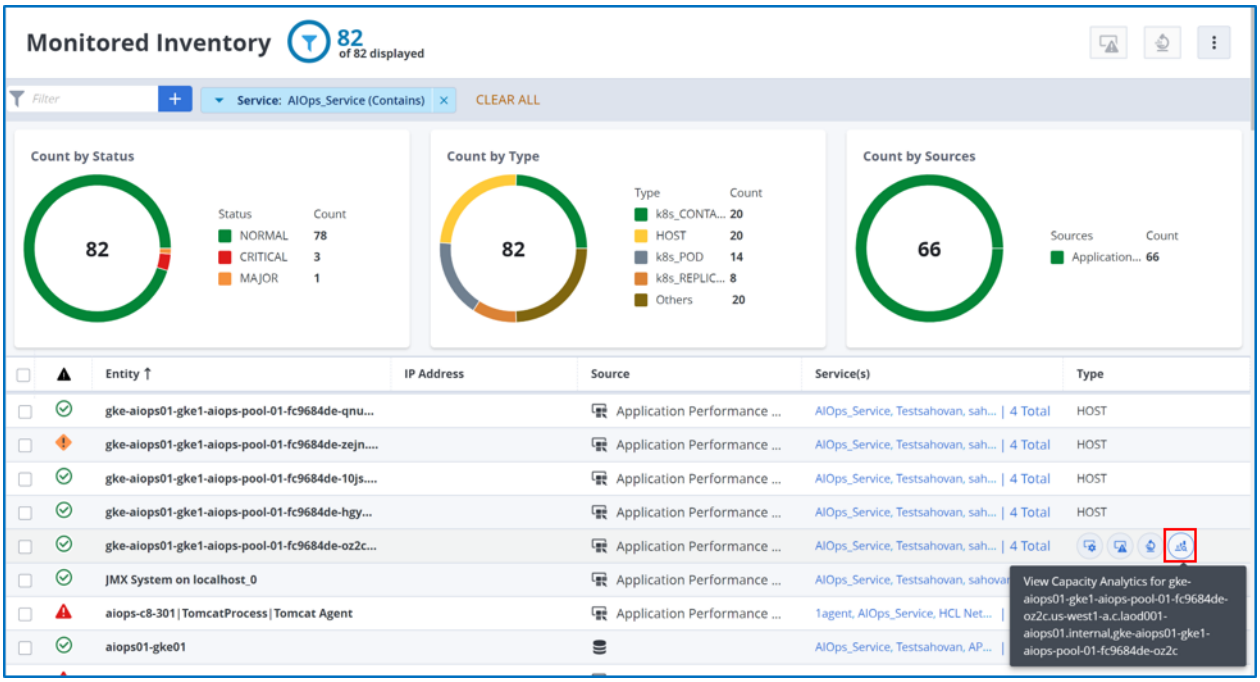
c) View the metrics associated with the specific entity:

1. Click the **Performance** icon under the **Type** column, which launches the **Performance Analytics** page in the context of the entity. You can select the metrics for that entity and can view the charts.



d) View the capacity for an entity:

1. Click the **Capacity Analytics** icon under the **Type** column, which launches to the **Capacity Analytics** page in the context of the entity. You can view the capacity that is required for the resources such as CPU, memory, storage, and network for the operational continuity of the enterprise workloads.



DX OI - Logs

An overview of the DX OI - logs capabilities, architecture, user profiles, and the supported log types and channels.

DX Operational Intelligence - Logs is a purpose-built analytics capability that collects analyses and visualizes the log data generated by your applications and IT infrastructure to gain operational insights.

This section provides the following information about Log Analytics:

- About DX OI - Logs
- Installation and Configuration for Log Ingestion
- Using Logs

About DX OI - Logs

Logs in DX Operational Intelligence provide a single pane of glass to perform analysis, visualization, and get insight into the logs across application workloads, infrastructure, and network.

The key capabilities of DX OI - Logs are:

- **Log Forwarding and Collection**
  There are five different channels for log forwarding and collection. These channels combine the usage of log forwarding agent such as Filebeat, Winlogbeat along with the Log Collector. The Log Collector helps the users in collecting and forwarding logs across workloads, infrastructure, and network.
  For more information about configuring and installing the log collectors and the agents, see the Configuring and Installing Log Collectors and Agents section.
- **Log Parsing**
  The Log Parser parses the collected logs based on log type, so that logs can be ingested into the right log index. Parsing for 16 log types is supported out-of-the-box. For other log types, logs are by default ingested into the generic index. Pre-parsed logs can be ingested using the Custom JSON API.
  For more information about custom log ingestion, see the Custom Support JSON API section.
- **Log Search and Filter**

Search and filter through your logs on the dashboard using available filters or build custom filters. You can also use DQL or Lucene to query through logs.

For more information, see the Search and Filter Logs Using Discover Option section.

- **Log Dashboarding**

  DX OI - Logs comes with packaged dashboards for the out-of-the-box supported log types. You can also create custom dashboards and visualizations.

  For more information, see the Log Dashboarding section.
- **Log Alarms**

  Log Alarms provide you the ability to monitor specific log patterns like errors, and specific error messages. You can define rules to generate log-based alarms.

  For more information about log alarms definitions, see the Log Alarm Configuration section.
- **Log Events Enrichment**

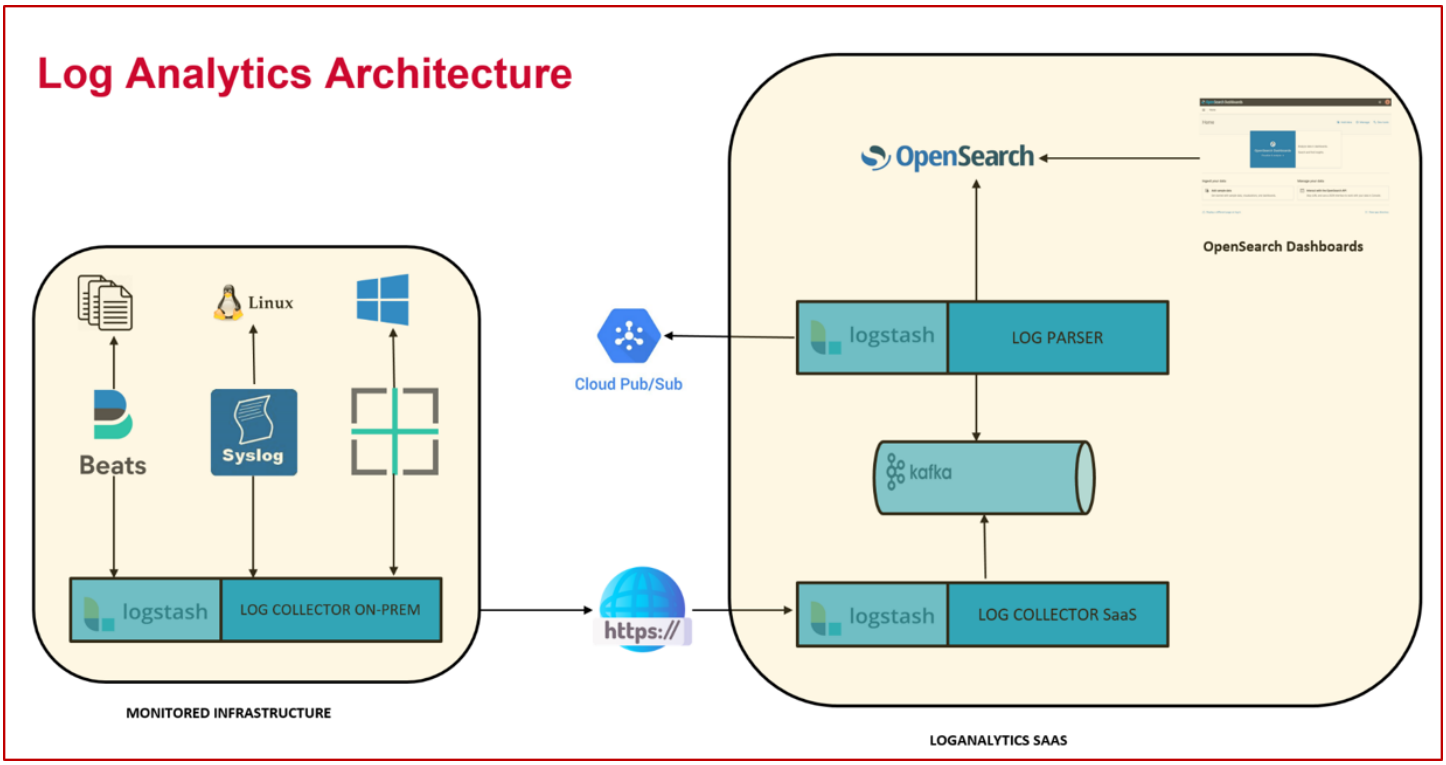
  The syslog log event is enriched with information about the entity, CI, or host from where the log is generated if the syslog log has information about the source or the host. You can view the logs for the host in Log Analytics, in the context of any alarms raised by other source products.
- **Log Ingestion Throttling**

  DX Operational Intelligence imposes caps on the log data that the Log Collectors can ingest per day for a tenant account. When the volume reaches the configured ingestion limit (hard limit) for your tenant account, the log ingestion is throttled.

  For more information, see the Log Ingestion Throttling section.

DX OI - Logs Architecture

The following diagram illustrates the DX OI - Logs architecture:



Components

DX OI - Logs requires the following DX Operational Intelligence components:

- **Log Collector**
  The Log Collector collects and aggregates logs and permits the separation of the software that generates messages, the system that stores the messages, and the software that reports and analyzes them. Each message is labeled with a facility code, indicating the software type generating the message, and is assigned a severity level. You can gain insights from data using analytics dashboards. The Log Collector package contains the files that are required to execute the Log Collector as a standalone component.

  NOTE
  For more information, see the Standalone Log Collector Installation section.

  Windows Event logs are also received through the syslog channel. You can use the open-source tool nxlog to send the event logs to DX Operational Intelligence through the syslog channel.

  NOTE
  For more information about configuration, see the Agent-less log collection methods section.
- **Filebeat**
  Filebeat is a lightweight log collection agent that reads any log files from the disk in near real time and sends the log data to the Log Collector.
- **Ingestion API**
  The Ingestion API ingests the data from the Log Collectors into DX Operational Intelligence.
- **Google Cloud Pub/Sub**
  The Google Cloud Pub/Sub messaging service ingests the GKE K8_Container logs directly into the DX Operational Intelligence. The log parser does not normalize or enrich the data before ingesting the logs in the data lake.
- **Kafka**
  Kafka processes the log data from the Log Collector and sends the log data to the Log Parser.
- **Log Parser**
  Log Parser receives log data from Kafka, parses the log data, extracts relevant fields, transforms the log data into the JSON format, and sends it to OpenSearch. For each supported log type, specific patterns are defined to parse and transform the data. This configuration is stored in the config files.
- **OpenSearch**
  OpenSearch is used as the data store and the analytics platform to store the log data. Log ingestion to OpenSearch is done by Log Parser. Each type of log data is stored as a separate document_type in the OpenSearch.
- **Log Dashboards Using OpenSearch**
  Log Analytics provides an integrated OpenSearch interface for dashboarding. Log Analytics is bundled with the pre-packaged dashboards for the out-of-the-box supported log types. You can also create custom dashboards and visualizations.

Who Can Use DX OI - Logs

The Application Developers, Administrators, and Support can use DX OI - Logs to get insight into the production workloads through the dashboards and alarm functionality that is provided for logs across applications, infrastructure, and network.

The different personas that can use DX OI - Logs are:

- **Administrators**
  The Administrators can monitor the entire landscape of the workloads that are deployed through a single-window dashboarding for all the logs that are generated across infrastructure, application, and network.
- **Developers**

The Developers can triage any issue in the monitored environment for applications by analyzing the logs using either the pre-packaged log dashboards or by creating the custom dashboards.

- **Support Engineer**
  The support engineers can configure alarms to look for specific error patterns in logs, and assign the alarms to the team members. They can even open a ticket in the context of that alarm in the integrated ticketing system.

Supported Log Ingestion Channels

DX Operational Intelligence supports multiple channels through which you can collect and ingest logs. The logs come in different formats, sizes, and from different sources. DX Operational Intelligence uses these channels to collect the diverse log data and unify at a single location in DX Operational Intelligence Data Lake.

- **Syslog**
  Collects the logs and events from the syslog enabled servers and network devices, and sends them over network through TCP and UDP ports using the native protocol. Log Stash (Log Collector) collects and sends the logs to the unified data lake using Ingest API Endpoint.
  Data Format: Raw
- **Filebeat**
  Collects logs from servers and Virtual Machines (VMs) on which you deploy Filebeat agents. The agent collects the logs directly from the files. Log collector collects the log files from each filebeat agent sends them to the unified data lake using Ingest API Endpoint.
  Data Format: Raw
- **Filebeat (sidecar log agent)**
  Collects logs from each pod of a cluster in Kubernetes on which you deploy the Filebeat agent. The cluster typically runs the containerized applications and other workloads. The logs of these applications are available in the various pods of a cluster. The file beat agent reads these logs and sends them to the unified data lake using Ingestion APIs through Log Collector.
  Data Format: Raw
- **Winlogbeat**
  Collects logs from Windows Machines on which you deploy Winlogbeat agent. The agent collects the logs directly from the files. Log collector collects the log files from each winlogbeat agent sends them to the unified data lake using Ingest API Endpoint.
  Data Format: Raw
- **Custom Tools**
  Ingests JSON formatted custom log data directly into the unified data lake using the Ingest API endpoint. DX Operational Intelligence directly ingests the normalized log into the OI data lake.
  Data Format: Normalized
- **Google Cloud Pub/Sub**
  Ingests the logs from Google Cloud Pub/Sub messaging service directly into the DX OI Environment. The Log Stash data pipeline parses the data.
  Data Format: Raw

Supported Log Types

DX Operational Intelligence supports multiple log types and collects these logs using both Agentless and Agent-Based collection methods.

Agent-Based Collection Method

In this method, you can deploy the agents and can run on the systems that generate the log data. These agents collect the logs as files and send them to Log Collector. DX Operational Intelligence supports 16 log types that you can collect using agents.

You can collect the log data for the following log types using Agent-Based collection method:

- Apache Access
- Apache Error
- Apache Kafka
- Docker
- Generic
- IIS
- Java Application Logs (Log4j)
- NGINX
- Oracle (Alert Logs and Audit Logs)
- Salt
- Spectrum
- SQL Server (Event Logs and Audit Logs)
- Syslog
- Tomcat Access
- Tomcat Catalina
- z/OS Syslog

> **NOTE**
> For more information about configuring log data collection, see the Agent-Based Log Data Collection section.

Agent-Less Collection Method

This method supports the collection of log data without installing any agent. The servers and devices send the generated log data over the network using native protocols such as TCP and UDP. Log Collector collects these logs before ingesting the collected log into the DX Operational Intelligence Environment.

Use Agentless Collection methods on systems where the Agent-Based collection method is not feasible.

You can collect the log data for the following log types using the Agentless collection method:

- Event logs (Event Logs, Event Logs - Oracle Audit (Windows), Microsoft Exchange, and Active Directory Server)
- Syslog (Syslog, Syslog - Oracle Audit (Linux), and Syslog - Docker)

> **NOTE**
> For more information about configuring log data collection, see the Agent-Less Log Data Collection section.

Installation and Configuration for Log Ingestion

You must install and configure the standalone Log Collectors and agents on various systems to collect and ingest the log data into DX Operational Intelligence. Agents collect logs from various sources and send them to the standalone Log Collectors. The standalone Log Collectors ingest the log data into DX Operational Intelligence using the Ingestion APIs.

This section provides the list of tasks that you must complete to install and configure the log collections and agents.

- Sizing Guidelines
- Verify System Requirements
- Download the Installer
- Install Standalone Log Collector
  - Standalone Log Collecter Installer - Virtual Machine
  - Standalone Log Collecter Installer - Docker
- Collect Logs
  - Agent-based Log Collection Method
  - Agent-less Log Collection Method
- Configure Google Cloud Pub/Sub to ingest logs from Google Kubernetes Clusters directly into the DX Operational Intelligencedata lake:
  - Configure Google Cloud Pub/Sub for Log Ingestion

Sizing Guidelines

The sizing process helps you determine the storage, CPU, and memory requirements for Filebeat and Log Collector for the log ingestion.

Default Configuration

The following table lists the default configuration for Filebeat and Log Collector.

Log Component	CPU Cores	RAM in GB	Storage in GB	Number of Log Events per Second	*Volume of Logs in KB per Second
Filebeat (Docker)	Min: 0.5 Max: 1	Min: 1 Max: 2	1 GB	~1000 events/sec	300 - 500 KB/Sec
**Filebeat (Standalone)	0.5 - 1	1 - 2 GB	1 G	~1000 events/sec	300 - 500 KB/Sec
Log Collector (docker)	Min: 2 Max: 4	Min: 2 Max: 4	N/A	~7000 events/sec	2000 - 3500 KB/Sec
**Log Collector (Standalone)	2 - 4	2 - 4	N/A	~7000 events/sec	2000 - 3500 KB/Sec

NOTE

- * Considering the event size of 300 to 500 bytes.
- ** Hardware requirements for the operating system are not accounted for in the mentioned metrics.

Configuration Recommendations

In addition to the horizontal scaling of Filebeat and Standalone Log Collector (SLC), the following are the recommendations to configure Filebeat and SLC for better performance.

- Filebeat
- Log Collector

Filebeat

Filebeat uses an internal queue to store events before publishing them. If no flush interval and no number of events to flush are configured, all events published to this queue will be directly consumed by the outputs. To enforce spooling in the queue, set the below configuration options.

- **MAX_EVENTS_IN_QUEUE:** The maximum number of events the queue can buffer. The default value is 4096.
- **MIN_EVENTS_BEFORE_FLUSH:** The minimum number of events required for publishing. If this value is set to 0, the output can start publishing events without additional waiting times. Otherwise, the output has to wait for more events to become available. The default value is 2048

The following configuration parameters are for the Log Collector output in the Filebeat.

- **PIPELINING:** Configures the number of batches to be sent asynchronously to Logstash while waiting for acknowledgment from Logstash. Output only becomes blocking once a number of pipelining batches have been written. Pipelining is disabled if a value of 0 is configured. The default value is 2.
- **WORKERS:** The number of workers per configured host publishing events to Logstash. The default value is 1
- **BULK_MAX_SIZE:** The maximum number of events to buffer internally during publishing. The default is 2048.
- **TTL_IN_SEC:** Time to live for a connection to Logstash after which the connection will be re-established. The default value is 30 seconds.

Log Collector

Log Collector provides the following configurable options for tuning pipeline performance.

- **LOGCOLLECTOR_WORKER_THREADS:** This setting determines how many threads to run for filter and output processing. The default value is 8.
- **LOGCOLLECTOR_BATCH_SIZE:** This setting defines the maximum number of events an individual worker thread collects before attempting to execute filters and outputs. Larger batch sizes are generally more efficient, but increase memory overhead. The default value is 1000.
- **LOGCOLLECTOR_BATCH_DELAY:** Pipeline batch delay is the maximum amount of time in milliseconds that Logstash waits for new messages after receiving an event in the current pipeline worker thread. After this time elapses, Logstash begins to execute filters and outputs. The default value is 3000 milliseconds

Sizing Examples

Use the following examples to help you determine the sizing requirements for log ingestion:

- Sizing Example for Filebeat
- Sizing Example for Log Collector

Sizing Example for Log Collector

Log Ingestion Per Day	<=10 GB	> 10 GB AND <= 100 GB	> 100 GB and <= 256 GB	> 256 GB and <= 512 GB
Number of Log Collectors	1	1	1	2
CPU Cores	0.5 - 1	1 - 2	2 - 4	2 - 4
RAM (GB)	2 - 4	2 - 4	4 - 8	4 - 8
LOGCOLLECTOR_WORKER_THREADS	4	8	8	8
LOGCOLLECTOR_BATCH_SIZE	500	500	1000	1000
LOGCOLLECTOR_BATCH_DELAY	3000	3000	3000	3000

Sizing Example for Filebeat

Log Ingestion Per Day	<= 100 MB	>100MB AND <= 1 GB	> 1 GB AND <= 10 GB	> 10 GB AND <= 100 GB
CPU Cores	0.05 - 0.1	0.05 - 0.1	0.2 - 0.5	1 - 2
Memory (GB)	0.125 - 0.25	0.125 - 0.25	1 - 2	4 - 6
Storage in GB	0.5	0.5	1	1
MAX_EVENTS_IN_QUEUE	256	256	1024	4096
MIN_EVENTS_BEFORE_FLUSH	128	128	512	2048
PIPELINING	0	0	1	2
WORKERS	1	1	2	2
BULK_MAX_SIZE	128	128	1024	4096

Verify System Requirements

Ensure that the following system requirements are met before proceeding with the Log Collector installation:

- Port 9600 is available for the Log Collector.
- Java 11 or higher version is required for the Log Collector.
- Systemd support is available on the host machine to enable registration of the Standalone Log Collector as a service.

  NOTE
  The Standalone Log Collection registration as a service fails if the host machine does not support systemd. You may have to manually start the Log Collector.

Ensure that the host or the VM machine on which you are installing the Log Collector meets the following hardware requirements:

RAM	**Minimum 2 GB**	
Supported Operating Systems	Red Hat Enterprise Linux CentOS	7.7 7.6 and 7.9

Download the Installer

You can download the installer from the **Settings** page.

Follow these steps:

1. Log in to DX Operational Intelligence.
2. Click **Settings** in the left navigation pane.
3. Click **Setup** in the **Datasources** tile.
   The application displays the **Downloads** page.
4. Download **Log Collector**.
   The **lc-onprem -oss-*.tar** is downloaded.

Components

The downloaded tar file includes the following folders:

- **FileBeat_Linux:** This folder includes the files that are required for installation on Linux.

  - filebeat-linux-x86_64.tar
  - installFileBeat.sh
  - startFileBeat.sh
  - stopFileBeat.sh
- **FileBeat_Windows:** This folder includes the files that are required for installation on Windows.
  - filebeat-windows-x86_64.tar
  - installFileBeat.batch
  - startFileBeat.batch
  - stopFileBeat.batch
- **OnPremLogCollector:** This folder includes the files that are required for Log Collector installation:
  - logcollector.tar
  - installLogCollector.sh
  - install_config
- **WinLogBeat:** This folder includes the files that are required for WinLogBeat installation:
  - winlogBeat-windows-x86_64.tar
  - installWinlogbeat.batch
  - startWinlogbeat.batch
  - stopWinlogbeat.batch

Install Standalone Log Collector

DX Operational Intelligence provides the Standalone Log Collector (SLC) installer as a package - **lc-onprem-<version>.tar.gz** . Using the SLC Installer, you can install the standalone Log Collector (log collector) on Linux systems.

You can install the Log Collector on one or more computers or VM machines based on the requirements of your organization. The SLC installer supports the following installation types:

- Interactive mode
- Silent mode

Perform the following tasks to install and manage Log Collector using the SLC Installer:

- Configure config.json File
- Run SLC Installer in Interactive Mode
- Run SLC Installer in Silent Mode
- Troubleshoot the Installation

Configure config.json File

You must configure the **config.json** file before you run the Standalone Log Collector installer in the Silent mode. The SLC installer uses the information that is provided in the config.json file while installing the standalone Log Collector. The config.json file contains the following information:

- **(Mandatory) Ingestion API URL:** The config.json file contains a default Ingestion API URL. You can change the API endpoint if necessary. The installer accesses the config.json file for API URL in both modes of installation.
- **(Optional) Proxy Servers Details:** Proxy servers provide an extra layer of security while sending the logs to DX Operational Intelligence over the Internet. You must provide the proxy connection details only when your organization has a proxy server. The Log Collector uses the proxy details to connect and route the logs to the proxy server, which in

turn, forwards the logs to DX Operational Intelligence for ingestion. The installer accesses the config.json file for Proxy details in the silent installation mode.

- **Default Installation Location:** By default, the SLC installer installs the log collector in the `/opt/brcm/logcollector` folder. You can change the installation location if necessary. The installer accesses the config.json file for the installation folder in both modes of installation.

Follow these steps:

1. Log in to DX Operational Intelligence.
2. Click **Settings** in the left navigation pane.
3. Click **Setup** in the **Datasources** tile.
   The application displays the Downloads page.
4. Download **Log Collector**.
5. Extract the **lc-onprem-<version>.tar** package from the Log Collector package on a host computer or a VM Machine.
6. Open Linux Command Prompt and untar the **lc-onprem-<version>.tar** package using the following command:
   ```
   tar -xzf lc-onprem-<version>.tar.gz
   ```
7. Open the config.json file from the **`<downloaded_loc/OnPremLogCollector/install_config/config.json>`** directory.
8. Provide the log ingestion API URL that the Log Collector uses to ingest the logs into DX Operational Intelligence:
   ```
   "dxsaas_logcollector_ingestion_api_url" : "https://logcollector.dxi-na1.saas.broadcom.com"
   ```
9. Provide the following proxy details:
   - **proxy_enabled**: Enter if the Ingestion API requests must be routed through the proxy server. **Default:** ‘false'.
     NOTE
     If proxy_enabled is set to **false**, the installer ignores the proxy connection credentials.
   - **proxy_hostname**: Enter the hostname of the proxy server.
   - **proxy_port**: Enter the port number of the proxy server.
   - **proxy_protocol**: Enter the proxy protocol: HTTP or HTTPS.
   - **proxy_Authentication**: Enter if proxy authentication is required. **Default:** ‘false'.
     NOTE
     If Proxy Authentication is set to **false**, the installer ignores the proxy authentication credentials.
   - **proxy_username** and **proxy_password**: Enter the proxy login credentials that the Log Collector uses to authenticate the proxy server.
   - **proxy_password_encrypted**: Encrypts the password when the value in the field is 'false'. After encryption, the value changes to 'true' indicating that the password is now encrypted.
     NOTE
     Do not make any changes to the **proxy_password_encrypted.**
10. (Optional) Specify the Log Collector installation directory. If the installation directory is not specified, the SLC installer installs the Log Collector in the **/opt/brcm/logcollector** folder.
    ```
    "logcollector_onprem_location" : "/opt/brcm/logcollector"
    ```
11. Save the config.json file.
    The configuration of config.json file is complete.

Run SLC Installer in Silent Mode

Using the SLC installer, you can install the Log Collector on one or more systems or VM machines in the silent mode with no user interaction. Before you begin, ensure that the following requirements are met:

- The system or the VM machine on which you are installing the Log Collector meets the specified system requirements.
- You have the required information in the config.json file for the installation.

Follow these steps:

1. Open the Linux Command Prompt.
2. Navigate to the **OnPremLogCollector** the directory from the command prompt.
   ```
   cd ./OnPremLogCollector
   ```
3. Execute one of the following commands to run the SLC installer:
   - If the config.json file is at the default location, run the following command:
     ```
     ./installaLogCollector.sh -s
     ```
   - If the config.json file is not in the default location, run the following command:
     ```
     ./installaLogCollector.sh –s <absolute-path>/config.json
     ```

   The SLC installer completes the Log Collector installation in silent mode.
4. Perform the following tasks to validate the Log Collector installation:
   a) Navigate to the **<SLC installation-location>/logstash/logs** directory from the command prompt.
   b) Execute the following Command:
      ```
      tail -f logstash-plain.log
      ```
      The Installer starts the Log Collector API endpoint and displays the following message:
      ```
      Successfully Started Log Stash API endpoint {:port=>9600}
      ```

Run SLC Installer in Interactive Mode

Using the SLC installer, you can install the Log Collector on one or more systems and VM machines in the interactive mode. When you run the installer in the interactive mode, the installer prompts you to provide the information during installation.

The installer accesses the config.json file to read the API URL and the default installation folder details. If this information is not available in the config.json file, the installer prompts for this information during installation.

Follow these steps:

1. Download the **lc-onprem-<version>.tar.gz package** on the system or the VM machine.
2. Open the Linux Command Prompt.
3. Untar the **lc-onprem-<version>.tar.gz** package using the following command:
   ```
   tar -xzf lc-onprem-<version>.tar.gz
   ```
4. Change the directory to the **OnPremLogCollector** folder using the cd command.
5. Run the following command to install the standalone Log Collector:
   ```
   ./installaLogCollector.sh
   ```
6. Perform one of the following actions when the installer prompts you to confirm the installation folder path:
   ```
   Enter Logcollector_Onprem Install Location (defaults to '/opt/brcm/logcollector' ):
   ```
   – Press Enter to install the Log Collector in the default folder.
   – Type the folder path in which you want to install the Log Collector.
7. Do one of the following when the installer prompts you to confirm the Ingestion API endpoint:
   ```
   Enter OI SaaS Ingestion API Endpoint (defaults to 'https://logs.dxi-na1.saas.broadcom.com' )
   ```
   You can select the URI based on the deployment and region of your tenant:
   – DX SaaS USA (Default):
     ```
     "https://logs.dxi-na1.saas.broadcom.com"
     ```
     Press Enter to use the default endpoint.
   – DX SaaS Europe:
     ```
     "https://logs.dxi-eu1.saas.broadcom.com"
     ```

8. Perform one of the following actions when installer prompts you to confirm the proxy server configuration:
   `"Do you wish to configure http proxy?(Yes/No)"`
   – Type **Yes** to enable the proxy server configuration.
   – Type **No** to disable the proxy server configuration.
9. If you enabled proxy server configuration, provide the following details when the installer prompts:
   – **PROXY_HOSTNAME**: Defines the hostname of the proxy server.
   – **PROXY_PORT_VAL**: Defines the port number of the proxy server.
   – **PROXY_PROTOCOL_VAL**: Specifies the proxy protocol – HTTP or HTTPS.
10. Perform one of the following actions when the installer prompts you to confirm the proxy server authentication:
    `"Do you wish to authenticate http proxy?(Yes/No)"`
    – Type **Yes** to enable the proxy server authentication.
    – Type **No** to disable the proxy server authentication.
11. If you enabled proxy server authentication, complete the following steps when Installer prompts:
    a. Provide the login credentials (proxy_username and proxy_passwod) to authenticate the proxy server access.
    b. Reenter the password when the installer prompts to confirm the password.

    NOTE
    Ensure you re-enter the correct password, when the installer prompts for confirm password. If the password and confirm password are mismatched, After 3 attempts the installer aborts the installation.

    The SLC installer completes the standalone Log Collector installation in interactive mode.
12. Perform the following tasks to validate the Log Collector installation:
    a. Navigate to the **<SLC installation-location>/logstash/logs** directory from the command prompt.
    b. Execute the following Command:
       ```
       tail -f logstash-plain.log
       ```

    The installer starts the Log Collector API endpoint and displays the following message:

    ```
    Successfully Started Log Stash API endpoint {:port=>9600}
    ```

Perform Health Check of Log Collector Installation

You can perform a health check on the standalone Log Collector to ensure that the installation is successful and is running as expected.

To perform a health check, complete one of the following steps:

- Execute the following command from the Command Prompt on Linux systems:
  ```
  curl -XGET '{machine-ip}:9600/?pretty'
  ```
  The application returns the following response:
  ```
  {
    "host" : "mn037518",
    "version" : "7.11.2",
    "http_address" : "10.17.164.109:9600",
    "id" : "cbbd4631-4025-4d06-a2fd-7181b4993c88",
    "name" : "mn037518",
    "ephemeral_id" : "7c5d6424-5bf3-4d7a-8729-e1fc4f20b8a4",

   "status" : "green",
    "snapshot" : false,
    "pipeline" : {
      "workers" : 32,
      "batch_size" : 500,
      "batch_delay" : 1000
    },
  ```

```
  "build_date" : "2021-03-06T04:40:05Z",
  "build_sha" : "5ea72dd819364370a8170ea90774578382e2fe42",
  "build_snapshot" : false
}
```

- Send the request from the Postman tool using the following endpoint:

  ```
  http://10.17.164.109:9600/?pretty
  ```

  The application returns the following response:

  ```
  {
      "host": "mn037518",
      "version": "7.11.2",
      "http_address": "10.17.164.109:9600",
      "id": "cbbd4631-4025-4d06-a2fd-7181b4993c88",
      "name": "mn037518",
      "ephemeral_id": "7c5d6424-5bf3-4d7a-8729-e1fc4f20b8a4",
      "status": "green",
      "snapshot": false,
      "pipeline": {
          "workers": 32,
          "batch_size": 500,
          "batch_delay": 1000
      },
      "build_date": "2021-03-06T04:40:05Z",
      "build_sha": "5ea72dd819364370a8170ea90774578382e2fe42",
      "build_snapshot": false
  }
  ```

Reinstall SLC Installer

You can reinstall the Log Collector on a host system using the SLC installer.

Follow these steps:

1. Navigate to the installation folder of the Log Collector.
2. Stop the Log Collector using the following command:

   ```
   ./stopLogcollector.sh
   ```

3. Run the SLC installer either in the Silent or Interactive mode.

Troubleshooting Standalone Log Collector Installation

This section describes some of the issues you might encounter during installation and suggests possible causes and solutions:

- Installation Aborted due to Unsupported Operating System
- SLC Installation Aborted due to Insufficient RAM
- Installation Aborted Due to Non-availability of Port 9600
- SLC Installation Aborted Due to Missing config.json File
- SLC Installation Aborted Due to Missing Configuration Parameter
- SLC Installation Aborted Due to Invalid Proxy Data
- Installation Aborted Due to Password Decryption Issue
- Installation Aborted Due to Missing $SLC_JAR File

Installation Aborted due to Unsupported Operating System

Condition

When I run the SLC installer in silent or interactive mode, the installer aborts the installation with the following message:

```
Unsupported OS, Unable to register the brcm_logcollector service
```

Solution

You can install log collectors using the SLC Installer only when the operating systems of host systems are:

1. Red Hat Enterprise Linux, Version - 7.7
2. CentOS, Versions- 7.6 and 7.9

SLC Installation Aborted due to Insufficient RAM

Symptom

When I run the SLC installer in silent mode, the installer aborts the log collector installation with the following message:

```
Insufficient System RAM Memory: $SLC_MAX_MEMORY, Aborting installation
```

Solution

Before installing the log collector on a host system using SLC installer, ensure that the host machine has minimum 2-GB RAM.

Installation Aborted Due to non-availability of Port 9600

Symptom

When I run the SLC installer in silent mode, the installer aborts the installation with the following message:

```
Already 9600 port is being used by other service, Required port is 9600, Aborting
 installation.
```

Solution

The port 9600 must be available for the log collector to communicate and ingest log data into the DX Operational Intelligence data lake.

SLC Installation Aborted Due to Missing config.json File

Symptom

When I run the SLC installer in silent mode, the installer aborts the installation with the following message:

```
Missing config file, Please retry with valid config file
```

Solution

The config.json file contains information such as the default installation folder of log collector, Ingestion API URL, and the proxy server settings. The SLC installer uses this information while installing the log collector in silent mode. You can perform one of the following tasks to re-run the SLC installer:

1. Ensure that the config.json file is available in `<installation_loc/OnPremLogCollector/install_config/config.json/Absolute Folder Path>` where the installer package is extracted.
2. Run the following command and provide the absolute file path of config.json file:

   ```
   ./installaLogCollector.sh -s <absolute-path>/config.json
   ```

3. Install the SLC log collector in interactive mode, and provide the relevant details when SLC installer prompts for information.

SLC Installation Aborted Due to Missing Configuration Parameter

Symptom

When I run the SLC installer in silent mode, the installer aborts the installation with the following message:

```
Invalid/Missing configuration parameter in the config file, aborting installation
```

Solution

The config.json file contains proxy server details. If the proxy authentication is set to 'true', the SLC installer reads the proxy details during installation. If any information about the proxy server is missing, the installer throws this error. Update the proxy details in the config.json file re-run the SLC log collector.

SLC Installation Aborted Due to Invalid Proxy Data

Symptom

When I install the log collector in silent mode, the SLC installer aborts the installation with any of the following error messages:

```
Invalid boolean values : $BOOLEAN_VAL. Value can be either true/false
Invalid Proxy host : $PROXY_HOSTNAME. Only allowed characters are : A-Z,a-z,0-9,-(hyphen),.(dot)
Invalid proxy port : $PROXY_PORT_VAL. Value can be only number.
Invalid proxy scheme : $PROXY_PROTOCOL_VAL. Value can be either http/https.
```

Solution

The config.json file contains proxy server settings. The SLC installer uses this information while installing the log collector. If any information about the proxy server is invalid, the installer aborts the installation with an appropriate error message. The error message also contains the valid data format and values for the proxy setting. Update the config.json file with the valid data and re-run the SLC installer.

Installation Aborted Due to Password Decryption Issue

Symptom

When I run the SLC installer in silent mode, the installer aborts the installation with the following message:

```
Unable to decrypt the Proxy Password, Aborting installation.
```

Solution

You may encounter this issue when the encrypted password in the Config.json file is tampered. To fix the password decryption issue, follow these steps:

1. Open the Config.json file.
2. Set the **proxy_password_encrypted** value to 'False'.
3. Manually enter the correct password in the **proxy_password** field and save the Config.json file.
4. Start the log collector installation using the SLC installer.

Installation Aborted Due to Missing $SLC_JAR File

Symptom

When I run the SLC installer in silent mode, the installer aborts the installation with the following message:

```
$SLC_JAR does not exist, Aborting installation.
```

Solution

The $SLC_JAR stores information about the password encryption. The SLC Installer uses the information to encrypt and decrypt the proxy server password.

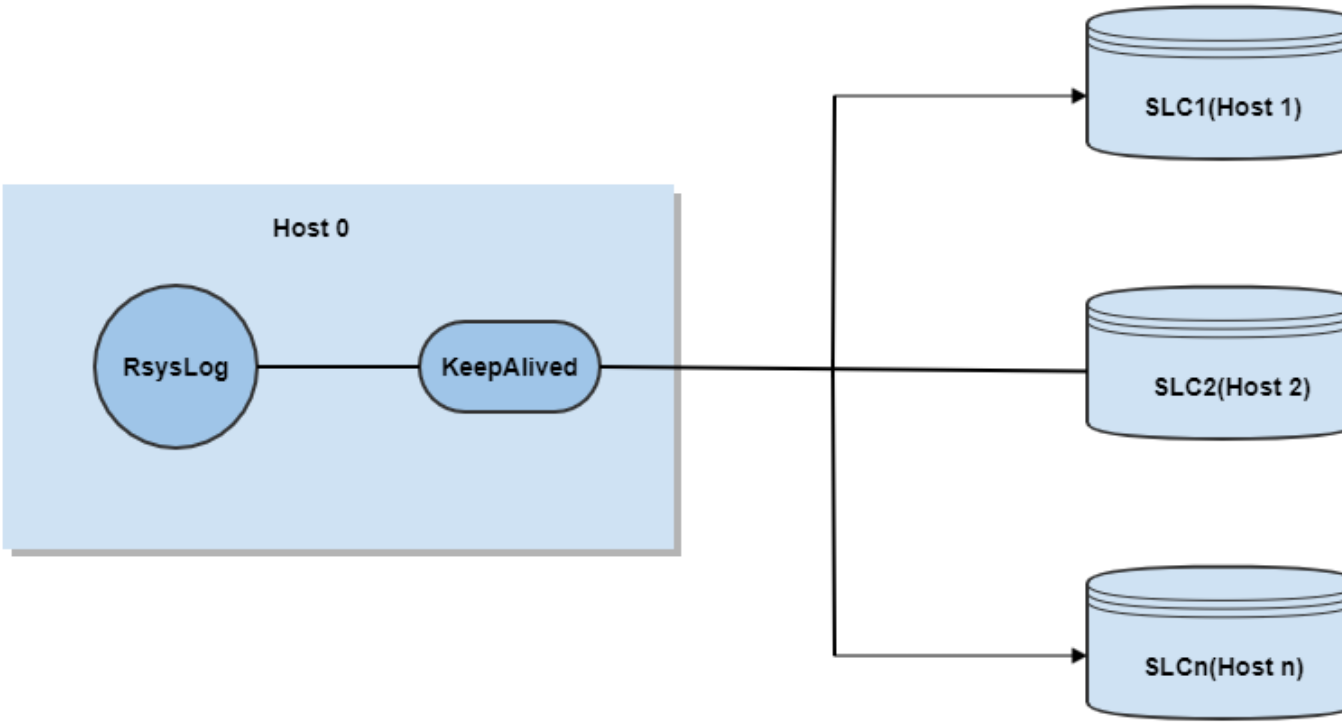
You can restore this jar file from the SLC installer package to enable password encryption.

Load Balancing of Rsyslog Logs

DX Operational Intelligence supports the installation of two or more standalone Log Collectors to manage the high volumes of logs from Rsyslog. This ensures the high availability of the Log Collectors by distributing the log traffic across a cluster of host machines using virtual and real IP Addresses.

DX Operational Intelligence uses the ipvsadm and Keepalived packages to implement the high availability of Log Collectors to load balance the huge volumes of Rsyslog log data.

- ipvsadm is used to set up the Virtual Server (IPVS) kernel module.
- Keepalived is used to implement the load balancing and high availability framework.



Prerequisites

Complete the following tasks before you configure load balancing for collecting logs from the Rsyslog host machine using the standalone Log Collectors:

- Install ipvsadm and keepalived packages on Rsyslog host machine using the following command:
  ```
  sudo yum -y install  ipvsadm keepalived
  ```
- Ensure you make the following tweaks:
  - Set the nis_enabled SELinux boolean to enable keepalived to call scripts that access the network.
    ```
    sudo setsebool -P nis_enabled=1
    ```
  - Execute the following command to enable IP forwarding and binding to a nonlocal IP address settings in the sysctl.conf:
    ```
    sudo vi /etc/sysctl.conf
    net.ipv4.ip_forward = 1
    net.ipv4.ip_nonlocal_bind = 1
    ```
- Install two or more standalone Log Collectors on different machines. Make a note of the IP Address and port for collecting the log data from the virtual IP Address:
  ```
  http.host is set to Machine IP address
  http.port is set to 9600 port
  ```

Configure keepalived.conf File

You must provide the virtual and real IP addresses and the ports in the keepalived.conf file to enable load balancing of log data. RSyslog sends the logs to a specified virtual IP address. Keepalived manages the Virtual IP traffic by distributing the log files to the IP addresses of Log Collectors using the IP Addresses(real) and port details of the host machines.

> **NOTE**
> Before you make the necessary changes, take a backup of the keepalived.conf file.

Follow these steps:

1. Add the virtual and two or more standalone Log Collector host machine (Real) IP Addresses in the keepalived.conf file.

   > **NOTE**
   > Ensure the virtual IP address is unique in your network.

```
global_defs {
   router_id LVS_DEVEL
   vrrp_skip_check_adv_addr
   vrrp_strict
   vrrp_garp_interval 0
   vrrp_gna_interval 0
}

vrrp_instance VI_2 {
    state MASTER
    interface ens160
    virtual_router_id 50
    priority 100
    advert_int 1
    authentication {
        auth_type PASS
        auth_pass 1111
    }
    virtual_ipaddress {
         10.131.239.200/23 ens160
    }
}
virtual_server 10.131.239.200 516 {
  delay_loop 5
  lb_algo rr
  lb_kind NAT
  ops
  protocol TCP

 real_server 10.131.239.154 6514 {
   weight 1
 #  TCP_CHECK {
 #         connect_timeout 500
 #         connect_port 516
 #       }
    HTTP_GET {
        url {
                path /_node/pipelines
           }
```

```
            connect_port 9600
            connect_timeout 3
            retry 3
            delay_before_retry 2
            }
      }

    real_server 10.131.238.118 6514 {
      weight 1
      HTTP_GET {
            url {
                    path /_node/pipelines
               }

            connect_port 9600
            connect_timeout 3
            retry 3
            delay_before_retry 2
            }
     }
```

2. Start keepalived by executing the following command:

   ```
   sudo systemctl start keepalived.service
   ```

3. Ensure the keepalived is up and running by executing the following command:

   ```
   sudo systemctl status keepalived.service
   ```

4. Execute the following command to verify the output:

   ```
   ipvsadm -ln
   ```

Deploy Standalone Log Collector - Docker

Complete the following steps to install the docker version of Log Collector:

Follow these steps:

1. Access the Log Collector image from the Docker Hub Repo: caapm/logcollector:latest
2. Create the lc-onprem deployment yaml file. Click here for the sample yaml file.
3. Update the following environment variables in the Log Collector deployment yaml file.
   - **SYSLOG_PORT:** Enter the port number to listen syslog data over TCP. Default: 6514.
   - **SYSLOG_UDP_PORT:** Enter the port number for syslog data over UDP. Default: 5140.
   - **LOGCOLLECTOR_BEATS_PORT:** Enter the port number to accept data from Filebeat. Default: 5044.
   - Enabling HTTP Proxy:
     - **PROXY_CONFIG:** Enter if you want to enable HTTP for proxy. **Values:** true or false. **Default:** false. If false, the following values are ignored:
       - **PROXY_HOSTNAME:** Enter the hostname of the proxy server.
       - **PROXY_PORT:** Enter the proxy server port value.
       - **PROXY_USERNAME:** Enter the username for the proxy server.
       - **PROXY_PASSWORD:** Enter the password for the proxy server.
   - **LOGCOLLECTOR_LOGGER_FILE_SIZE_IN_MB** (Defaults to 50 MB)
   - **LOGCOLLECTOR_BATCH_SIZE:** Enter the number of events or log messages to be processed in a single batch by a worker. Set this value to 1000, that is, b1000 (Default: b75).

NOTE
LOGCOLLECTOR_BATCH_SIZE must be tuned properly if the type of ingestion is custom and events/logs are in the JSON format. Depending on the average size of JSON object, this reduces or increases for the optimum performance. For non-custom and non-json type, use 1000.

- **LOGCOLLECTOR_WORKER_THREADS:** Enter the number of workers to be run. Should be 3-4 times the maximum number of cpu cores available per pod. If the maximum CPU cores requested is 4, then set this value to 16, that is, w16 (Default: w8).
- **LOGCOLLECTOR_BATCH_DELAY:** Enter the maximum time in milliseconds for a worker to wait before the required batch size of events that are available in the memory queue to pick and process. Set this value to 2000 (Default: 1000).
- **ENABLE_FILE_LOGGING:** Enable if you want to enable file logging. If true, logs are written to the file. If false, no logs are written to the file (Default: true)

4. Update the following ports to be exposed:
   - "6514:6514" - TCP
   - "5140:5140/udp" - UDP
   - "5044:5044" - TCP
5. Update the mounts directories:
   - **/logcollector_config:** Configuration folder of the Log Collector. This folder can be mounted to make any changes to the Log Collector configuration.

     A configuration mount directory is required to have the following permissions:

     - chmod -R 777 "$LOGCOLLECTOR_ONPREM_MOUNT"
     - chown -R 1010:1010 "$LOGCOLLECTOR_ONPREM_MOUNT"

     For example, if you are mounting a nfs folder such as /home/mount/logcollector to /logcollector_config, then /home/mount/logcollector folder should have the mentioned permissions.

     NOTE
     The Log Collector pod must be restarted for the changes to reflect in the config directory.
   - **/opt/caemm/logs** - Logs folder for the Log Collector. This folder can be mounted to store the logs of the Log Collector.

Agent-less Log Collection Methods

You can use Agent-less log collection methods to collect System logs (Windows Eventlog and Unix syslog) through TCP using Log Collector.

- Configure Syslog Using Rsyslog
- Configure Syslog Using syslog-ng (Unix and Linux)
- Configure Syslog Message Collection for Network Devices
- Post Installation Verification

You can refer to the following troubleshooting topics in case you encounter any issues while collecting the system logs.

- Troubleshooting Agent-less Log Collection
- Troubleshooting Errors in Rsyslog Log Message

Configure Syslog Using Rsyslog

To configure syslog for remote access with rsyslog, follow these steps:

1. On the source system, open the **/etc/rsyslog.conf** file for editing.
2. Uncomment the following lines:

```
# An on-disk queue is created for this action. If the remote host is
# down, messages are spooled to disk and sent when it is up again.

$WorkDirectory /var/lib/rsyslog # where to place spool files
$ActionQueueFileName fwdRule1 # unique name prefix for spool files
$ActionQueueMaxDiskSpace 1g # 1gb space limit (use as much as possible)
$ActionQueueSaveOnShutdown on # save messages to disk on shutdown
$ActionQueueType LinkedList # run asynchronously
$ActionResumeRetryCount -1 # infinite retries if host is down
```

3. Add the following lines in the file (template to parse data as JSON) for version 7 and above:

```
template(name="ls_json"
type="list"
option.json="on") {
constant(value="{")
constant(value="\"syslog_timestamp\":\"") property(name="timereported" dateFormat="rfc3339")
constant(value="\",\"syslog_pri\":\"") property(name="pri")
constant(value="\",\"syslog_ver\":\"1")
constant(value="\",\"tenant_id\":\"<Tenant UUID>")
constant(value="\",\"syslog_message\":\"") property(name="msg")
constant(value="\",\"host\":\"") property(name="hostname")
constant(value="\",\"syslog_severity\":\"") property(name="syslogseverity-text")
constant(value="\",\"syslog_facility\":\"") property(name="syslogfacility-text")
constant(value="\",\"syslog_severity_code\":\"") property(name="syslogseverity")
constant(value="\",\"syslog_facility_code\":\"") property(name="syslogfacility")
constant(value="\",\"syslog_program\":\"") property(name="programname")
constant(value="\",\"syslog_pid\":\"") property(name="procid")
constant(value="\",\"syslog_hostname\":\"") property(name="$myhostname")
constant(value="\",\"syslog_priority\":\"") property(name="syslogpriority")
constant(value="\"}\n")
}
For Connection with TCP Port 6514
*.*; @@<LOG ANALYTICS HOST>:6514;ls_json
For Connection with UDP Port 5140
*.*; @@<LOG ANALYTICS HOST>:5140;ls_json

Replace <Tenant UUID> with the unique UUID. For more information about how to find UUID, see the
 Troubleshooting section.
Replace <LOG ANALYTICS HOST> with the hostname or IP.
```

4. Add the following lines under the **GLOBAL DIRECTIVES** section of the rsyslog configuration:

```
$PreserveFQDN on
```

5. Add the following lines (template to parse data as JSON) for version 5 and below.

```
$template LogFormat,"{ \"syslog_timestamp\":\"%timereported:::date-rfc3339%\", \"syslog_message\":
\"%msg%\", \"syslog_severity\":\"%syslogseverity-text%\", \"syslog_facility\":\"%syslogfacility-text
%\", \"syslog_severity_code\":\"%syslogseverity%\", \"syslog_facility_code\":\"%syslogfacility%\" ,
\"syslog_program\":\"%programname%\", \"syslog_pid\":\"%procid%\", \"syslog_hostname\":\"%$myhostname%\",
 \"syslog_priority\":\"%syslogpriority%\", \"tenant_id\":\"<Tenant UUID>\" }"

*.*; @@<LOG ANALYTICS HOST>:6514;LogFormat

Replace <Tenant UUID> with the unique UUID. For more information about how to find UUID, see the
 Troubleshooting section.
In the OpenShift environment, for agentless log data ingestion, update the hostname to the connection
 parameters with port 6514.
 *.*; @@<logcollector-host-or-route-to-port-6514>;ls_json
```

6. (Optional) Configure Syslog to Send Logs Over SSL or TLS using rsyslog:
   a) Install the rsyslog-gnutls dependency (SSL/TLS dependencies).

      ```
      Environments?.DEB: apt-get install rsyslog-gnutls
      Environments?.RPM: yum install rsyslog-gnutls
      ```

   b) Add the following lines in the **/etc/rsyslog.conf** file after the lines mentioned in step 3.

      ```
      #####Added for TLS Support###
      $DefaultNetstreamDriverCAFile <CA Certificate Path>
      $DefaultNetstreamDriver gtls # use gtls netstream driver
      $ActionSendStreamDriverMode 1 # require TLS for the connection
      $ActionSendStreamDriverAuthMode anon # server is NOT authenticated
      #####END####
      ```

   c) Replace <CA Certificate Path> with the actual location where the CA Certificate is stored. Currently .pem and .crt certificates are supported.
   d) Restart the rsyslog service using command: ***service syslog restart***

   You can find the Rsyslog log messages at **/var/log/messages**. You can verify for any abnormalities using the following command: ***cat /var/log/messages | grep syslog*** . This command lists down all the messages of the rsyslog service.

Configure Syslog Using syslog-ng (Unix and Linux)

To configure syslog using syslog-ng, follow these steps:

Add the following configuration in the **'/etc/syslog-ng/syslog-ng.conf'** file:

```
source msg_source {
 file("/var/log/messages" follow-freq(1));
 };
 template LogAnalyticsTemplate {
 template ("{\"syslog_timestamp\": \"$ISODATE\", \"syslog_pid\": \"$PID\", \"syslog_facility\": \"$FACILITY
\", \"syslog_priority\": \"$LEVEL_NUM\", \"syslog_pri\": \"$PRI\", \"syslog_severity\": \"$LEVEL\",
 \"syslog_severity_code\": \"$LEVEL_NUM\", \"syslog_facility_code\": \"$FACILITY_NUM\", \"syslog_hostname
\": \"$HOST_FROM\", \"host\": \"$HOST\", \"syslog_program\": \"$PROGRAM\", \"tenant_id\":\"<TENANT_ID>\",
 \"syslog_message\": \"$MSG\"}\n" );
 template_escape(yes);
};
rewrite r_rewrite_subst{subst("\'", "`", value("$MSG"));};
destination send_json {
 tcp("<LOG_COLLECTOR_IP>" port(6514) template(LogAnalyticsTemplate));
```

```
};
log {
 source(msg_source);
 rewrite(r_rewrite_subst);
 destination(send_json);
};
Restart rsyslog-ng service by executing the "service syslog-ng restart" command.
Replace <LOG_COLLECTOR_IP> with the log collector route.
Replace port (6514) with the port (80).
```

Configure Syslog Message Collection for Network Devices

Network devices can send syslog messages. These messages must be sent to a Linux or Unix system with syslog daemon. For example, Rsyslog can be used to listen to syslog messages sent by devices and can forward it to Log Analytics.

To configure the Syslog Message collection for network devices, follow these steps:

1. Uncomment the following lines in the **/etc/rsyslog.conf** file to enable listening of syslog messages:

   ```
   # Provides UDP syslog reception
    $ModLoad imp
    $UDPServerRun 514
    # Provides TCP syslog reception
    $ModLoad imtcp
    $InputTCPServerRun 514
   ```

2. Run the following command to restart the Rsyslog service: ***service rsyslog restart***

Post Installation Verification

Perform the following steps after you have configured log forwarding to DX Operational Intelligence using agentless (syslog/eventlog) or UIM mechanism.

To verify installation, follow these steps:

1. Log in to DX Operational Intelligence.
2. Click and select **DX Dashboard**.

   The Dashboards Manage dashboards & folders page which is the landing page opens in a new tab.

   For a new user, the landing page displays the Out-of-the-box dashboards under the folders that are named as **General** and **Health Monitoring**. For an existing user, this page displays both the Out-of-the-box dashboards and the custom dashboards, if any.

   For an existing user, this page displays both the Out-of-the-box dashboards and the custom dashboards, if any.
3. Click the required dashboard in the list to open and view the details.
4. Click **Load Saved Dashboards** and search **LA: Log Analytics Dashboard.**

   If the installation and configuration were successful, you see log data in the dashboard. You can open Dashboards for individual log types from the left panel of the Log Analytics Dashboard.

Troubleshooting Agent-less Log Collection

1. Verify if the correct CA AXA Tenant ID is configured in Nxlog (For Windows Event) or Rsyslog (Linux Syslog Messages). For more information, see Agent-less Log Collection Methods.
2. Navigate to the USER_INSTALL_DIR/logcollector/logstash-2.3.4/dictionary directory on the system where Log Collector is deployed.
3. Verify if the CA AXA Tenant ID is available in the whitelist file of tenant (tenants.yml). Manually add the ID if missing. For example, 3C987D4A-3FF8-E3D8-8F5F-074CC89FCB8F: "true".
4. Restart Log Collector by executing the stopLogCollector.sh and startLogCollector.sh scripts present in the $DIST_HOME/bin directory.

Troubleshooting Errors in Rsyslog Log Message

Condition

If you encounter any similar errors in the log messages of syslog:

```
could not load module '/usr/lib/rsyslog/lmnsd_gtls.so', rsyslog error -2078 [try http://www.rsyslog.com/
e/2068 ]

or

could not load module '/usr64/lib/rsyslog/lmnsd_gtls.so', rsyslog error -2078 [try http://www.rsyslog.com/
e/2068 ]
```

Remedy

1. Look for the actual module at the given path using the following command:

   ```
   ls -la /usr/lib/rsyslog/lmnsd_gtls.so or ls -la /usr64/lib/rsyslog/lmnsd_gtls.so
   ```

2. If not found, then reinstall rsyslog-gnutls.
3. Also, verify if the user has privileges to install and use rsyslog and is able to read the certificates.

Agent-based Log Collection Methods

To collect logs using agent-based log collection methods, you must deploy or install and configure the agents on host machines. A host can be a Windows or Linux-based server and VM Machine, or Kubernetes pods.

You can collect and send logs for the following log types using agent-based collection methods:

- Apache Access
- Apache Error
- Docker
- Generic
- IIS
- Java Application Logs (Log4j)
- NGINX Oracle (Alert Logs and Audit Logs)
- Spectrum SQL Server (Event Logs and Audit Logs)
- Syslog
- Tomcat Access
- Tomcat Catalina
- z/OS Syslog

This section provides information to install and configure agents:

- Install and Manage Filebeat Agent
- Install and Manage Winlogbeat agent

Install and Manage Winlogbeat Agent

Winlogbeat Agent collects logs from windows based hosts. This section provides information to install and manage the Winlogbeat agents:

- Install Winlogbeat
- Start and Stop Winlogbeat Agent

Install Winlogbeat

You can collect logs from Windows-based hosts by installing Winlogbeat on the host systems. Winlogbeat sends the raw logs to the Log Collector for ingestion into the DX Operational Intelligence.

> **NOTE**
> Ensure that you run the batch file with Administrator access for a successful installation.

Follow these steps:

1. Download the **lc-onprem-*.tar.gz** package from the **Settings > Datasources** tile in DX Operational Intelligence.
2. Extract the tar file and navigate to the **\WinlogBeat** folder.
3. Execute the **installWinlogbeat.bat** file to install Winlogbeat.
4. Provide the following details when prompted:
   - **Winlogbeat Install Location**: Provide the path of the Winlogbeat installation folder.
   - **Enter OI SaaS Tenant ID**: Type the tenant ID for which you are collecting the logs.
   - **LogCollector_Onprem HostName**: Provide the IP address or the host name on which the log collector is installed.

     > **NOTE**
     > **Note:** Command line parameter values should not contain leading and trailing spaces.
5. Verify if the Winlogbeat Installation is successful by validating the Winlogbeat logs in the following path:

   ```
   <Installation-Location>\winlogbeat\winlogbeat-windows-x86_64\logs\
   ```

   The Winlogbeat Installation is complete.
6. (Optional) Change the logging level to **debug** in the winlogbeat.yml file for collecting detailed logs.

   Default: Default: 'info' is the default logging level that collects the basic logs.

   ```
   logging.level: debug
   ```

Start and Stop Winlogbeat Agent

You can start and stop the Winlogbeat agent by executing the following commands from the Command prompt:

- To start the Winlogbeat agent, execute:

  ```
  .\StartWinlogbeat.bat
  ```
- To stop the Winlogbeat agent, execute:

  ```
  .\StopWinlogbeat.bat
  ```

> **NOTE**
> When you change any configuration for Winlogbeat, you must restart the Winlogbeat agent to update the winlogbeat.yml file.

Install and Manage Filebeat Agent

Filebeat collects logs from Windows-based and Linux-based servers, VMs and Kubernetes pods, and parses them based on the predefined patterns. **Filebeat** then sends these logs to the Log Collector. Filebeat requires the hostname and port of the Log Collector as the input. You can find the details of the log paths to collect the data from the *filebeat.yml* configuration file.

Filebeat supports only the following patterns:

- Apache access
- Apache error
- Docker
- IIS logs
- Tomcat Catalina
- Tomcat Access
- ZOS syslog
- Log4j
- Oracle Alert
- Kafka
- Salt
- SQL Server

Install Filebeat on Linux

You must install Filebeat on the Linux systems after installing the Log Collector.

> **NOTE**
> Ensure that you run the script file with Administrator access for a successful installation.

Follow these steps:

1. Download **Log Collector** from the **Settings > Datasources** tile in DX Operational Intelligence.
2. Untar the lc-onprem-oss-*.tar.gz file:

   ```
   tar -xvf lc-onprem-oss-*.tar.gz
   ```

3. Change the path to the **FileBeat_Linux** folder:

   ```
   cd FileBeat_Linux/
   ```

4. Navigate to the installation folder.
5. Execute the installFileBeat.sh file to install Filebeat for Linux.

   ```
   ./installFileBeat.sh
   ```

6. Provide the following details when prompted:
   - **Filebeat_Onprem Install Location:** Provide the path of the Filebeat installation folder.
   - **Enter Log File Location:** Provide the log file path that Filebeat accesses to read the logs.
   - **Enter OI SaaS Tenant ID**: Type the tenant ID for which you are collecting the logs.
   - **LogCollector_Onprem HostName:** Provide the IP address or the hostname on which the Log Collector is installed.

   The Filebeat installation is complete.
7. Verify if the Filebeat installation is successful by validating the Filebeat logs in the following path:

   ```
   <installation folder>/filebeat/filebeat-linux-x86_64/nohup.out
   ```

8. (Optional) Change the logging level to **debug** in the filebeat.yml file to collect detailed logs.

   Default: **info** is the default logging level that collects the basic logs.

```
logging.level: debug
```

Install Filebeat on Windows

After installing the Log Collector, you can install Filebeat on Windows systems based on your requirements.

> **NOTE**
> Ensure that you run the batch file with Administrator access for a successful installation.

Prerequisites:

- Ensure that you remove the filebeat folder from the following path before you install Filebeat on Windows:
  `C:\ProgramData`

Follow these steps:

1. Download the **lc-onprem-*.tar.gz** package from the **Settings > Datasources** tile in DX Operational Intelligence.
2. Extract the lc-onprem-*.tar.gz file.
3. Navigate to the **\FileBeat_Linux** folder.
4. Navigate to the **\FileBeat_Linux** that has installFileBeat.bat.
5. Execute the **installFileBeat.bat** file to install the Filebeat on Windows.
6. Provide the following details when prompted:
   - **Filebeat_Onprem Install Location:** Provide the path of the Filebeat installation folder.
   - **Enter Log File Location:** Provide the log file path that Filebeat accesses to read the logs.
   - **Enter OI SaaS Tenant ID**: Type the tenant ID for which you are collecting the logs.
   - **LogCollector_Onprem HostName:** Provide the IP address or the hostname on which the Log Collector is installed.

     > **NOTE**
     > **Note:** Command line parameter values should not contain leading and trailing spaces.

   After providing the details, the Filebeat installation success message appears if the provided values are valid.
7. Verify if the Filebeat Installation is successful by validating the Filebeat logs in the following path:

```
<installation-path>\filebeat\filebeat-windows-x86_64\logs\filebeat
```

8. (Optional) Change the logging level to **debug** in the filebeat.yml file for collecting detailed logs.

   Default: 'info' is the default logging level that collects the basic logs.

```
logging.level: debug
```

Deploy Filebeat - Docker

Complete the following steps to deploy the docker version of Filebeat.

Follow these steps:

1. Access the Filebeat image from the Docker Hub Repo: caapm/filebeat:22.3.1.1
2. Create the Filebeat deployment yaml file. Click here for the sample yaml file.
3. Update the following environment variables in the Filebeat yaml file.

- **TENANT_ID:** Enter the tenant ID in the capital case. For example, "3E289377-1DBD-402D-8A09-55D5FEF983D7".
- **LOGSTASH_HOST_NAME:** Enter the hostname/IP address of the system where the Log Collector is installed.
- **LOGSTASH_BEATS_PORT:** Enter the port number of the Log Collector which accepts the beats data (Default: 5044).
- **FILEBEAT_IP:** (Optional) Enter the custom IP address of Filebeat which is to be sent along with the data.
- **FILEBEAT_HOSTNAME:** (Optional) Enter the custom hostname for Filebeat. If this value is not set, the container name becomes the hostname.
- **CONTAINER_NAME:** This name is set as the deployment name.
- **FILEBEAT_LOGGER_MAXSIZE_IN_MB:** Enter the maximum amount of storage in MB which can be used by logs that are written by Filebeat. Once the maximum limit is reached, Filebeat rotates the log files.

4. Update the mounts directories.
   - **Configuration Mount (Mandatory):** /filebeat_config - The configuration folder of Filebeat which also contains the registry information of files being read by Filebeat. The filebeat agent reads the logs from the mounted folder. A configuration mount directory is required to have the following permissions:
     - chmod -R 777 "$FILEBEAT_CONF_DIR"
     - chown -R 1010:1010 "$FILEBEAT_CONF_DIR"

     For example, if you are mounting a nfs folder **/home/mount/filebeat** to **/filebeat_config**, then the **/home/mount/filebeat** folder should have the mentioned permissions.

     NOTE
     If the configuration directory is not mounted, Filebeat reads all the logs files in its configured path from the beginning every time it restarts.
   - **Filebeat Logs (Written by Filebeat - Optional):**
     - **/opt/caemm/logs** - Log folder of Filebeat. This folder can be mounted to store the logs written by Filebeat.
   - **Log Location Mounts (Read by Filebeat):**
     - Filebeat supports many log types and each type has their own mount folder as described in the following table. For example, for Syslog type, the folder to be mounted is /home/default/filebeat_logs/syslog. For Kafka type, the folder to be mounted is /home/default/filebeat_logs/kafka.

Log Type	Mount Directory
Generic	/home/default/filebeat_logs/generic
Syslog	/home/default/filebeat_logs/syslog
Log4j	/home/default/filebeat_logs/log4j
Apache Access	/home/default/filebeat_logs/apache_access
Apache Error	/home/default/filebeat_logs/apache_error
Tomcat Access	/home/default/filebeat_logs/tomcat_access
Tomcat Catalina(Error)	/home/default/filebeat_logs/tomcat
Kafka	/home/default/filebeat_logs/kafka
Docker	/home/default/filebeat_logs/docker
Salt	/home/default/filebeat_logs/salt
SQL Server	/home/default/filebeat_logs/sqlserver
Oracle	/home/default/filebeat_logs/oracle
ADA	/home/default/filebeat_logs/ada
Windows Event	/home/default/filebeat_logs/windows_event
Nginx Access	/home/default/filebeat_logs/nginx

Log Type	Mount Directory
IIS Server	/home/default/filebeat_logs/iis
Ntevl	/home/default/filebeat_logs/ntevl
ZOS Syslog	/home/default/filebeat_logs/zos

NOTE
You can either create multiple PVs for supporting multiple log types or you can use the same PV with different subPath's to configure multiple log types.

Start and Stop Filebeat
When you change any configuration for Filebeat, you must restart the Filebeat agent to update the filebeat.yml file.

Execute the following commands in the command prompt:

- On Windows systems,
  - To start the Filebeat agent,
    ```
    ./startFileBeat.bat
    ```
  - To stop the Filebeat agent,
    ```
    ./stopFileBeat.bat
    ```
- On Linux systems,
  - To start the Filebeat agent,
    ```
    ./startFileBeat.sh
    ```
  - To stop the Filebeat agent,
    ```
    ./stopFileBeat.sh
    ```

Deploy Filebeat as Sidecar

DX Operational Intelligence supports the log collection from the Google Kubernetes Engine clusters by deploying the Filebeat agent (Filebeat) as the sidecar. You can deploy the Filebeat agent alongside any existing container in Kubernetes pods.

The cluster typically runs the containerized applications and other workloads. The logs of these applications are available in the various pods of a cluster. The application supports log accessibility for the Filebeat agent by mounting the logs directory of the application container with the Filebeat container. You can use multiple volume types such as persistent, emptyDir, while mounting the logs directory.

A Filebeat agent deployed as a sidecar directly reads these logs for a pod and sends them to the DX Operational Intelligence data lake using the Log Collector.

NOTE
You must deploy multiple Log Collector instances to load balance the collection of logs from Filebeats across all the pods.

You must update the Filebeat deployment and configuration details in a yaml-based deployment file. You can download a sample deployment file from here.

Follow these steps:

1. Access Filebeat Image from the Docker Hub Repo: `caapm/filebeat:latest`
2. Complete the following steps in the deployment file:
   a) Provide the information for the following variables:
      - **TENANT_ID:** Provide the DX Operational Intelligence tenant ID.
      - **LOGSTASH_HOST_NAME:** Provide the Hostname or the IP Address of the system on which the Log Collector is installed.
      - **LOGSTASH_BEATS_PORT:** Provide the port number of the Log Collector that accepts the logs from Filebeat. Default: 5044
      - **(Optional) FILEBEAT_IP:** Provide the Custom IP Address of Filebeat to be sent along with logs.
      - **(Optional) FILEBEAT_HOSTNAME:** Provide the custom hostname for Filebeat. If the hostname is not provided, the container name is used as hostname.
      - **CONTAINER_NAME:** Provide the container name. The container name is set as the deployment name.
      - **FILEBEAT_LOGGER_MAXSIZE_IN_MB:** Define the maximum storage limit in MB that Filebeat can use to write the logs. When the maximum limit is reached, Filebeat rotates the log files.

   b) Mount the configuration directory (/filebeat_config) of Filebeat that contains the registry information of files. The filebeat agent reads the logs from the mounted folder.

      NOTE
      If the configuration directory is not mounted, Filebeat reads all the logs files in the configured path from the beginning when you restart Filebeat.

   c) Ensure that the Configuration Mount directory has the following permissions:
      - chmod -R 777 "$FILEBEAT_CONF_DIR"
      - chown -R 1010:1010 "$FILEBEAT_CONF_DIR"

      Example: If you are mounting an NFS folder like **/home/mount/filebeat** to **/filebeat_config**, then **/home/mount/filebeat** folder must have these permissions.

   d) (Optional) Mount the log folder **/opt/caemm/logs** of the Filebeat to store the logs written by Filebeat.

   e) Mount log location folder for each log type that you want the Filebeat to read.

      Filebeat supports many log types and each log type has a mount folder.

      Examples:

      The **Syslog** type: The mounting folder is **/home/default/filebeat_logs/syslog**

      The **Kafka** type: The mounting folder is **/home/default/filebeat_logs/kafka**

      You can either create multiple volumes for supporting multiple log types or use the same volume with different sub paths to configure multiple log types. By default Filebeat supports only the reading log files in the specified folder. If you want the log files to be read from the sub folders, then update the same in filebeat.yml at **/filebeat_config/config/filebeat.yml**

Example: If a syslog application writes logs into /root/appname/logs/*, then mounting it to /home/default/filebeat_logs/syslog/* works. But if the same application creates a sub directory to write the log files, then configure the logs path in the filebeat.yml (/filebeat_config/conf/filebeat.yml) config file:

Local Mount Dir	Target Directory in filebeat	Logs type	Configuring log path in filebeat.yml
/root/appname/logs/*	/home/default/filebeat_logs/syslog/*	Logs written directly in the mount path	/home/default/filebeat_logs/syslog**/***
/root/appname/logs/*	/home/default/filebeat_logs/syslog/*	Logs written in files under the /subdir directory of the mount path	/home/default/filebeat_logs/syslog/**subdir/***
/root/appname/logs/*	/home/default/filebeat_logs/syslog/*	Logs are written in more than one sub directory of mount path	/home/default/filebeat_logs/syslog**/*/***

This filebeat deployment and configuration is complete.

Authenticate Log Ingestion

You must authenticate the log ingestion with the user token when the ingestion endpoint is APM Services Gateway. You can generate the user token on the **Settings > Tokens** page. For more information, see the Tokens section.

Provide the user token as follows:

- **Standalone SLC:** Provide the user token as the authorization token at the time of the installation. For the Silent installation mode, add the user token in the config.json file.
- **Docker SLC:** Provide the user token as the value for the **LOGCOLLECTOR_REQUEST_AUTHORIZATION** environment variable.
- **Custom Logs Ingestion via Direct API:** Provide the user token as the Bearer Token in the HTTP Header.

Configure Google Cloud Pub/Sub for Log Ingestion

DX Operational Intelligence supports Ingestion of logs from Google Kubernetes Clusters directly into the DX Operational Intelligence data lake using Google Cloud Pub/Sub messaging service (Pub/Sub).

> **NOTE**
> Currently, DX Operational Intelligence supports the stderr and stdout of resource type k8s_container and ingests as generic log events.

The Log Parser does not parse or enrich the data before ingesting them into the data lake.

As a Tenant Administrator, you must ensure that the following requirements are in place to configure the Google Cloud Pub/Sub messaging service for the log ingestion:

- The Google Cloud Platform Service Account with the JSON private key file
- A Service Account with permissions to pull messages from the subscription
- One or more topic names
- The dedicated subscription name for DX Operational Intelligence

You must complete the following tasks to configure Google Cloud Pub/Sub for Log Ingestion:

1. Log into the Google Cloud Console and create the topics and the dedicated subscription for DX Operational Intelligence.

   For more information on how to create and manage topics and subscription, see Google Cloud Pub/Sub Documentation.

2. Create Service Account to generate the JSON Key file.

   The JSON Key file is required for LogParser and the Pub/Sub Configuration.

   For more information on creating and managing the service accounts, see Google Cloud Pub/Sub Documentation.

3. Configure the Google Cloud Pub/Sub as a log sink and map the topics to export logs from StackDriver to Pub/Sub.

   Define the appropriate filters to separate different logs to appropriate topics and then map them to relevant tenants in DX Operational Intelligence.

   For more information on how to configure the Google Console pub/sub as a sink, see Google Cloud Pub/Sub documentation.

4. Define the payload for log exports in the log sink.

   Before you define the payload requirements, ensure that the following information is available:

   - Tenant namespace name is available under the JSON key "resource.labels.namespace_name"
   - namespace_name is of the format **<tenant_name_space>-<project_identifier>**
   - namespace_name is same as configured in the Google Kubernetes Engine cluster.
   - Tenant name space is always followed by hyphen(-) and then project_identifier (project_identifier as suffix, after the last hyphen)
   - project_identifier format can be anything and of any format.

   Sample Payload Format


```
{
  "insertId": "uizewwf6hoah7",
  "jsonPayload": {
    "message": "Attach failed with error failed to execute command. Container
 \"99c71338c66642e0de07e39b51857e32ca31aecde224fe25193b857668e029a2\" not found",
    "pid": "2050005"
  },
  "labels": {
    "compute.googleapis.com/resource_name": "gke-gke1-pool-n1-64-f-7849c8c1-r2b9",
    "k8s-pod/app": "caagent",
    "k8s-pod/controller-revision-hash": "5cbcffdbcf",
    "k8s-pod/pod-template-generation": "2"
  },
  "logName": "projects/saas-dev-us-cstack-gke1/logs/containerinfo",
  "receiveTimestamp": "2020-11-17T22:42:12.02820461Z",
  "resource": {
    "labels": {
      "cluster_name": "gke1",
      "container_name": "containerinfo",
      "location": "us-east4",
      "namespace_name": "broadcom-aiops",
      "pod_name": "app-container-monitor-sk4jw",
      "project_id": "saas-dev-us-cstack-gke1"
    },
    "type": "k8s_container"
  },
```

```
    "severity": "INFO",
    "sourceLocation": {
      "file": "autoattach.go",
      "line": "199"
    },
    "timestamp": "2020-11-17T22:42:11.914728Z"
  }
```

Configure Custom Configuration Files for Log Collector

Log Collector by default provides configuration files with the default settings and without any parsing rules. For the ingestion of custom logs, you can use the existing Log Collector with log type specific configurations, which might vary based on the log type. Some examples of custom configurations are creating JSON formatted data from unstructured data or dropping data based on some logical conditions. For any custom log parsing, you can enhance the Log Collector to include code to parse the data. This can be done either in Log Collector or in the Filebeat using processors.

Enable Custom Configuration - Docker

To enable custom configuration for Log Collector, create two directories named **conf** and **patterns** and mount both these directories at **/home/default/custom** of the Log Collector. All the custom conf files (example filename.conf) must be placed in the **conf** directory. These files extensions shall be .conf. Also, the supporting pattern files must be placed in the **patterns** directory. The following snippet is an example of the Config Map Mounts:

```
- mountPath: /home/default/custom/conf/custom.conf
  name: cm-data
  subPath: config
- mountPath: /home/default/custom/patterns/custom-patterns
  name: cm-data
  subPath: patterns
```

While starting, the Log Collector checks for the custom config files in the **/home/default/custom/conf** directory. If the config files are present, the Log Collector copies the files into the actual Log Collector config location. If the files are not present, the Log Collector does not copy any files.

For any custom log parsing,

- **Log Collector:** If parsing in Log Collector, ensure that the following requirements are met:
  - **Log Collector:**
    - The custom data being ingested should contain the **logformat, logtype (custom index name),** and **timestamp** fields before the data is ingested into the Log Collector.
    - The timestamp should be in UTC format.
    - All the grok pattern references in the custom conf files should point to the **/home/default/custom/patterns** directory.
  - **Filebeat:** If the custom logs data is being ingested from Filebeat, add the required fields under the fields section in the Filebeat configuration as shown:

```
- type: log

  enabled: true
  paths:
    - ${FILEBEAT_LOGS_DIR}/custom/*

  tags: [ "Custom Logs" ]
  fields:
    logtype: "<name of your logtype>"
```

```
    log format: "custom"
  fields_under_root: true
  clean_removed: false
  exclude_files: ['.gz$','.zip$','.tar$','.bz2$','.tgz$','.lz$']
```

- **Filebeat:** If parsing in Filebeat, you can filter the data using one of the following methods:
  - Filtering at the input level
  - Filtering using processors

    NOTE
    For more information, see the Filter Data Using Filebeat section.

Enable Custom Configuration - Standalone Log Collector Installer

To enable custom configuration, write the custom code (Logstash configuration code) in the Log Collector.

For custom log parsing, ensure that the following requirements are met:

- **Log Collector:** If parsing in Log Collector, ensure that the following requirements are met:
  - **Log Collector:**
    - The custom data being ingested should contain the **logformat, logtype,** and **timestamp** fields before the data is ingested into the Log Collector.
    - The timestamp is in epochmillis, yyyy-MM-dd'T'HH:mm:ss.SSSX, or yyyy-MM-dd'T'HH:mm:ss.nXXXXX format. For more information on the formats, see the **Mapping Format** documentation.

      NOTE
      strict_date_optional_time_nanos is also supported.
    - The mandatory fields are provided with the values:
      - timestamp in the epoch_millis format.
      - tenant_id : <cohort_ID>
      - logformat : "custom"
      - logtype: "<name of your logtype>
    - All the grok pattern references and other changes in the custom conf files should be placed at the proper path as mentioned in the custom conf file. The file path is **<installDir>/conf**.
  - **Filebeat:** If the custom logs data is being ingested from Filebeat, add the required fields under the fields section in the Filebeat configuration as shown:

```
- type: log

  enabled: true
  paths:
    - ${FILEBEAT_LOGS_DIR}/custom/*

  tags: [ "Custom Logs" ]
  fields:
    logtype: "<name of your logtype>"
    logformat: "custom"
  fields_under_root: true
  clean_removed: false
  exclude_files: ['.gz$','.zip$','.tar$','.bz2$','.tgz$','.lz$']
```

- **Filebeat:** If parsing in Filebeat, you can filter the data using one of the following methods:
  - Filtering at the input level
  - Filtering using processors

NOTE
For more information, see the Filter Data Using Filebeat section.

Follow these steps:

1. Write the logic in the new file with the extension .conf. For more information about how to write the filtering logic, see the **Logstash** documentation. You can write one or more files.
2. Place all the custom conf files (<filename1.conf>, optionally <filename2.conf>) in the **<installDir>/conf** directory. And, place all the supporting pattern files in the patterns directory.
3. Restart the Log Collector startup script that is available in the bin (**/bin/startLogcollector.sh**) directory for the following changes:
   – Log Collector:
   ```
   Replace
    ${LOGCOLLECTOR_LOGSTASH_HOME}/bin/logstash -f ${LOGCOLLECTOR_LOGSTASH_HOME}/conf/logcollector.conf -w8
    -b1000 -u3000 &> /dev/null &

   to

   ${LOGCOLLECTOR_LOGSTASH_HOME}/bin/logstash -f ${LOGCOLLECTOR_LOGSTASH_HOME}/conf -w8 -b1000 -u3000 &> /
   dev/null & -/bin/startLogcollector.sh
   ```
   – Log Collector Service:
   ```
   Replace
    ${LOGCOLLECTOR_LOGSTASH_HOME}/bin/logstash -f ${LOGCOLLECTOR_LOGSTASH_HOME}/conf/
   logcollectorservice.conf -w8 -b1000 -u3000 &> /dev/null &

   to

   ${LOGCOLLECTOR_LOGSTASH_HOME}/bin/logstash -f ${LOGCOLLECTOR_LOGSTASH_HOME}/conf -w8 -b1000 -u3000 &> /
   dev/null & -/bin/startLogcollectorService.sh
   ```

Support Other Customizations

To support other customizations, write the custom code to write any logic and copy the files in the paths mentioned earlier. The following example illustrates how to filter the logs.

Follow these steps:

1. Write the custom code to drop all events containing the word info in the message field:
2. Place this conf file in the **<LOGCOLLECTOR_LOGSTASH_HOME>/conf** directory.
3. Restart the Log Collector after making the required changes.

Filter Data Using Filebeat

Filebeat provides the following options to filter the data it reads or before sending the data to Log Analytics:

- Filtering at the input level
- Filtering using the processors

Filtering at Input Level

The input section in the **filebeat.yml** file specifies how Filebeat locates and processes the input data. To configure Filebeat, specify the inputs in the **filebeat.inputs** section as shown:

```
filebeat.inputs:
```

```
- type: log

  paths:

    - /var/kafka/logs

    - /var/kafka/*.log
```

This configuration defines a single input with a single path. You can configure each input to include or exclude specific lines or files. This allows you to specify different filtering criteria for each input. To do this, use the include_lines, exclude_lines, and exclude_files options.

- **exclude_lines:** The exclude_lines filter specifies a list of regular expressions. Any line that matches the regular expression in the list is dropped. The following example configures Filebeat to drop any lines that start with ERR.

```
filebeat.inputs:

- type: log

  paths:

    - /var/kafka/logs

    - /var/kafka/*.log

  exclude_lines: ['^ERR’]
```

- **include_lines:** The include_lines filter specifies a list of regular expressions. Any line that matches the regular expression in the list will be read. The following example configures Filebeat to export any lines that start with DBG or WARN.

```
filebeat.inputs:

- type: log

  paths:

    - /var/kafka/logs

    - /var/kafka/*.log

include_lines: ['^DBG’, '^WARN']
```

- **exclude_files:** The exclude_files filter specifies a list of regular expressions. Any file that matches the regular expression in the list will be ignored. The following example configures Filebeat to ignore all the files that have a zip extension.

```
filebeat.inputs:

- type: log

  paths:

    - /var/kafka/logs

    - /var/kafka/*.log
```

```
    exclude_files: ['\.zip$']
```

Filtering Using Processors

In the filtering using the processors' method, define processors in the Filebeat configuration to process events before they are sent to the Log Collector. The processors are applied to all events that are processed by the Filebeat. The **drop_event** processor drops the entire event if the associated condition is fulfilled. The basic syntax of the drop event processor is as shown:

```
processors:

  - drop_event:

      when:

        condition
```

The drop event processor supports multiple filtering conditions. Here are a few conditions:

- **equals:** This condition compares if the field in the event has a particular value. The following condition drops all the events if the response code of the HTTP response code is 201.

```
processors:

  - drop_event:

      when:

        equals:

           code: 201
```

- **contains:** This condition checks if the given string or an array of strings is part of a field. The following condition drops all events if **Exception** is part of the status field.

```
processors:

  - drop_event:

      when:

        contains:

           status: "Exception"
```

- **regexp:** This condition checks the field against a regular expression. The following condition drops all events if the hostname starts with **test**.

```
processors:

  - drop_event:

      when:

         regexp:

           host: "^test.*"
```

NOTE
For more information about the conditions, see the Filebeat documentation.

Set up Custom Log Ingestion

DX Operational Intelligence supports ingestion of custom log data directly in to the unified data lake using Custom JSON Support API.

Prerequisites:

Before you process the custom JSON ingestion, you must share the log type with the DX Operational Intelligence Support team. The Support team enables the log type for smooth Custom JSON ingestion.

Custom JSON Support API

The Custom JSON Support API enables you to send the pre-parsed custom logs directly in to the DX Operational Intelligence environment. These logs are not parsed or enriched further as they are pre-parsed using the custom tools in the tenant environment.

Resource URI

```
https://<oi_host>:<oi_port>/la/v2/api/ingestion/logs
```

Method

```
POST
```

HTTP Headers

- Authorization: Bearer {token}
  For more information on generating the token, see the Token Management section.
- Content Type: Application/JSON

Request Payload Syntax

```
{
"tenant_id" : <"TENANT ID">,
"logtype" : <"LOG TYPE">,
"logformat": "custom",
"envtype": <"Environment Type">,
"c_double_fieldName": 100.4567,
"c_long_fieldName": 113123412312,
"c_timestamp_fieldName" : <"TimeStamp">,
"c_text_fieldName": <"">,
"resposneCode": <400>,
"customer_id" :<"Customer ID",
"class" : <"Class Name">
"timestamp" :<time stamp>
}
```

Table 6: Mandatory Parameters

Parameter	Description
tenant_id	Sends the tenant ID or the cohort ID of the tenant.
logotype	Sends the logs of the specified log type to the DX Operational Intelligence data lake.
timestamp	Specifies the log event timestamp in the epoch millis format.
logformat	Sends the log format as "custom".

Few examples of optional parameters:

Table 7: Optional or Normalized Parameters

Parameter	Description
ip	The IP address of the system on which log is generated.
host	The host name of the system where the log is generated.
container_id	Container ID or Pod ID where the log is generated.
tag	List of tag names associated with the log.
container_name	Service name or Service name where the log is generated.
file	Location of the file

Guidelines for Payload Format

While sending the input payload to DX Operational Intelligence, ensure that the following conditions are met:

- The input payload must contain the mandatory information.
- Do not include reserved fields in the input payload.
- Maximum number of unique fields in an index per log type is 150.
- Maximum payload size, that is one json document size must not exceed 250 KB.
- Maximum content Length of a single request must not exceed 10 MB, If the length is more than 10 MB, DX Operational Intelligence rejects the request with 413 response code.
- Normalize the fields by mapping the field names in the input payload with the field names specified in the table, when both fields refer to the same data. Example: Use "host" instead of "hostname", similarly use "ip" instead of "ip_address".
- Do not include fields with empty values in the input payload.
- Supported timestamp format is epoch_millis.
- log format must always be "custom".
- By default, the text, numeric, and decimal fields are indexed as keyword, integer, and float fields in elastic search. You can override the data type by prefixing the field with "c_<data_type>_ ". If you want to index the field as long, prefix the filed with c_long_<field_name>. Example: c_long_executiontime, c_long_hitcount.

  NOTE
  You can override the data type of a field only when you ingest custom data for the first time. For subsequent ingestions, you cannot override the data type. The application discards the log document when there is a mismatch in data type of a field.

Response Syntax

```
{
"code": 200,
```

```
}
```

Error Handling

- When there is a data congestion and stress on the receiving endpoint, the API returns 429 response code.

  ```
  {
  "code": 429
  }
  ```

  Solution: You must wait and then retry with exponential back-off and jitter strategy.
- When the maximum content length of a single HTTP request is greater than 10 MB, the API returns 413 response code.

  ```
  {
  "code": 429
  }
  ```

  Solution: Ensure that the content length does not exceed 10 MB.

Sample Request-Response

Sample URI

```
http://adminui.10.00.0.nip.io/la/v2/api/ingestion/logs
```

Sample Payload

```
{
"tenant_id" : "FC073035-9D3E-4103-B68E-0B2EC2DF8E7B",
"logtype" : "log4j_custom",
"logformat": "custom",
"envtype": "dev",
"c_double_fieldName": 100.4567,
"c_long_fieldName": 113123412312,
"c_timestamp_fieldName" : "1600961070000",
"c_text_fieldName": "some sample text",
"resposneCode": 400,
"customer_id" :"cust_123",
"class" : "org.apache.commons.SampleClass"
"timestamp" :1600961070000
}
```

Sample Response

```
{
    "code": 200
}
```

Custom Logs Parsing Rules

Log parsing refers to the process of converting the unstructured log data to a structured JSON format. By default, DX OI - Logs includes the log data parsing rules for log types such as log4j, Kafka, and Oracle. These parsing rules are defined in the Log Parser component of DX OI - Logs which applies them to the incoming data, pulls out attributes (**key:value** pairs) from the unstructured log data, and stores them in the indices as a JSON document. However, if any new log types or new log formats for the supported log types are ingested, the existing parsing rules may not be able to parse the logs resulting in parsing failures.

> **NOTE**
> For more information about the log types, see the Agent-based Log Collection Methods section.

For new log types or log formats, you can create log parsing rules that convert the unstructured log data to structured JSON to easily parse, store, search, and create visualizations and dashboards. A parsing rule basically defines the new log type and the corresponding patterns that specify the rules to parse the log lines for the given log type.

> **NOTE**
> Every log event that is sent to **DX OI - Logs** must have the **timestamp** field. If the timestamp field is not present, then DX OI - Logs adds the timestamp field with the current time in UTC.

Create Parsing Rule

Before you create a parsing rule, understand the following terminology:

- **Pattern Definition:** A pattern definition is a regular expression that specifies a search pattern. DX OI - Logs includes in-built definitions that you can reuse to define parsing rules for new log types. You can also define your own pattern definitions to parse non-standard log patterns. The following table lists sample pattern definitions:

Pattern Definition Name	Value
HOSTNAME	\b(?:[0-9A-Za-z][0-9A-Za-z-]{0,62})(?:\.(?:[0-9A-Za-z][0-9A-Za-z-]{0,62}))*(\.?|\b)
USERNAME	[a-zA-Z0-9._-]+
USER	%{USERNAME}

> **NOTE**
> For more information, see the **Built-in Definitions Reference** section on this page.

- **Pattern:** A pattern is a simple parsing rule that is created by combining the pattern definitions which break down the log line into different fields that are specified in the pattern. A pattern is specified using Grok, an industry standard for parsing log messages. Any incoming log is checked against the built-in parsing rules, and if possible, the associated Grok pattern is applied to the log.

  Sample Log:

  ```
  2016-07-11T23:56:42.000+00:00 INFO [MyApp.com.Transaction.Manager]:Starting transaction for session
   -464410bf-37bf-475a-afc0-498e0199f008
  ```

  Sample Pattern:

  ```
  "%{TIMESTAMP_ISO8601:timestamp} %{LOGLEVEL:log-level} \[%{DATA:class}\]:%{GREEDYDATA:message}"
  ```

Follow these steps:

1. Log in to DX Operational Intelligence.
2. Open the **Settings** page.
3. Click **Define** in the **Define Log Parsing Rules** tile.

   The **Log Parsing Rules** page is displayed. This page displays the parsing rules for MongoDB, MySQL, and PostgreSQL out-of-the-box and also any parsing rules if created for custom logs. These out-of-the-box rules are disabled by default. You must redefine them as required based on the log format in your specific environment and must enable them to be applied to the log data.
4. Click **+ Create New Rule**.
5. Provide the following information:
   - **Name:** Enter a name for the rule.
   - **Log Type:** Enter the log type. Only alphanumeric characters, hyphens, and underscores are allowed.
   - **Field to Parse:** Enter the field name to parse. That is, enter the field that has the unstructured data.

     > **NOTE**
     > Do not use the field name that is entered here as the variable name in the pattern.
   - **Description:** Enter a description for the rule.
   - **Rule is Active:** Enable this option to activate this rule.

NOTE
If you disable a parsing rule, logs will be dropped.

6. Add the pattern for the parsing rule. You can provide the pattern using the built-in pattern definitions or you can add your own pattern definitions.
   – **Built-in Pattern Definitions:**
     1. Enter the sample JSON payload to evaluate it against the patterns defined in this parsing rule. For example,

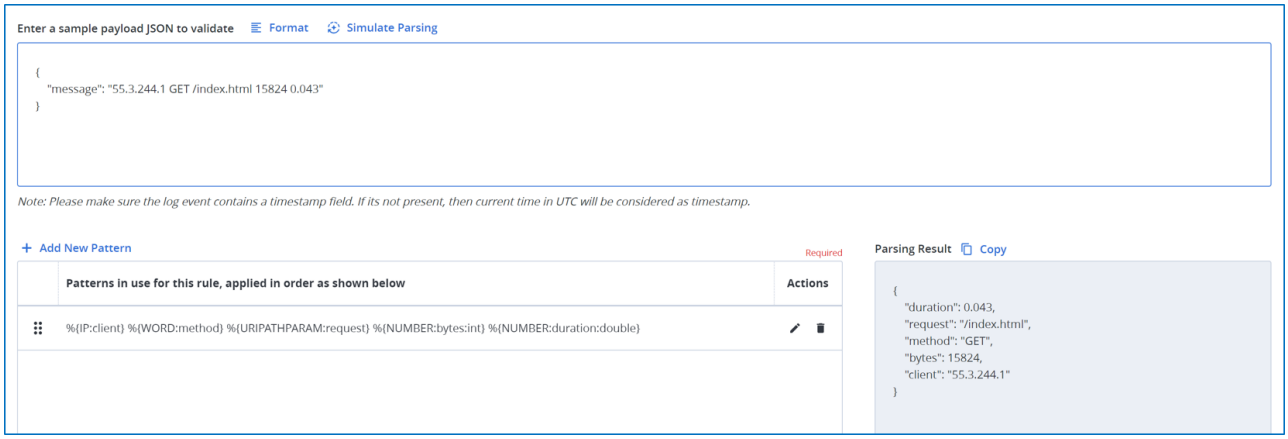
```
{
    "message": "55.3.244.1 GET /index.html 15824 0.043"
}
```

NOTE
In this example, the **Field to Parse** is **message**.

     2. Enter the pattern using the built-in definitions.
        a. Click **Show built in definitions** to display the **Pattern Definition Library**, which lists the pattern definitions that you can reuse for reference.
        b. Enter the pattern. For example, enter,

```
%{IP:client} %{WORD:method} %{URIPATHPARAM:request} %{NUMBER:bytes:int} %{NUMBER:duration:double}
```

        c. Click **Simulate Parsing** to display the **Parsing Result** as shown:

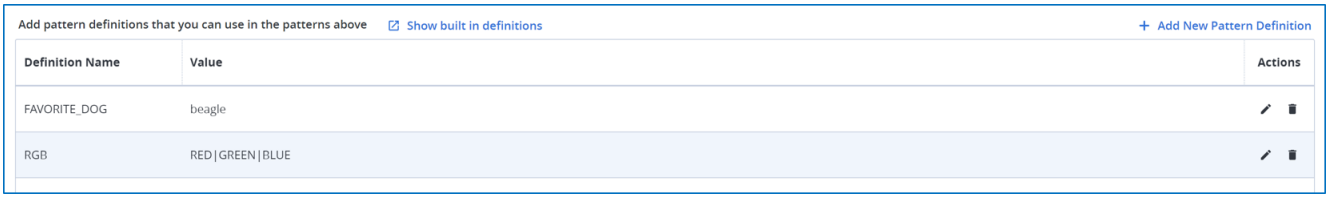


        d. Save the pattern definition.
     3. Click **+ Add New Pattern** and repeat the steps to add more patterns.
   – **Custom Pattern Definitions:**
     1. Enter the sample JSON payload to evaluate it against the patterns defined in this parsing rule. In the following example, the **Field to Parse** is message.

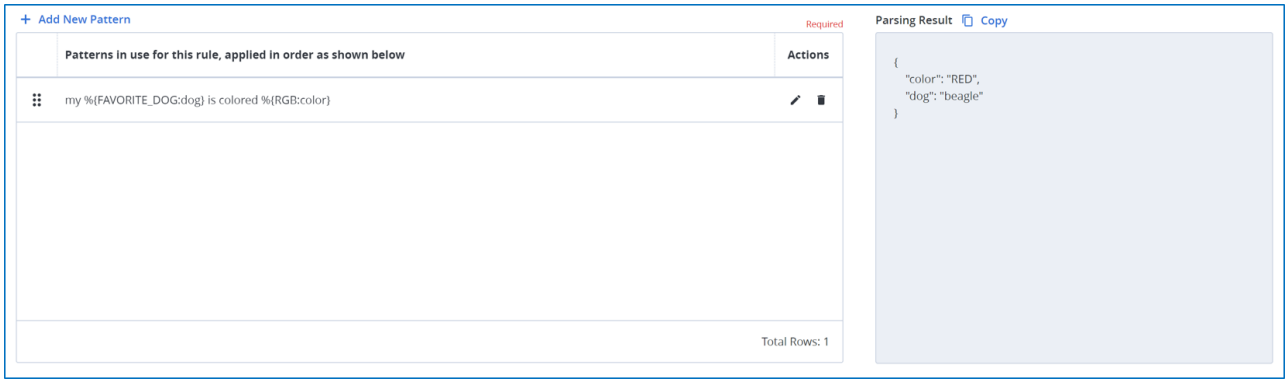
```
{
    "message": "my beagle is colored RED"
}
```

     2. Add the Definition Pattern: Before you use the custom definitions in the pattern, you must add the custom definitions.
        1. Click **+ Add New Pattern Definition**.
        2. Provide the following information:
           - **Definition Name:** Enter a name for the definition.
           - **Value:** Enter the value for the definition.
        3. Click the **Save** icon.
        4. Click **+ Add New Pattern Definition** and repeat the steps to add more definitions.

        The following image illustrates two sample definitions to be used in the pattern:



3. Add the Pattern.
   a. Click **+ Add New Pattern.**
   b. Enter the pattern using the definitions that you created earlier. For example, enter the pattern as,
      `my %{FAVORITE_DOG:dog} is colored %{RGB:color}`
   c. Click **Simulate Parsing** to display the **Parsing Result** as shown:



   d. Save the pattern.
   e. Click **+ Add New Pattern** and repeat the steps to add more patterns.
7. Click **Create** to save the parsing rule.
   The rule creation takes a few seconds. The created parsing rule is added to the **Log Parsing Rules** page. The Log Parsing Rule page displays the following information for each rule: Name, Description, Log Type (Index), Field to Parse, Created by, Last Updated, Status, and Action (Edit and Delete)

Use the Parsing Rule

After you create the parsing rule, you can use the parsing rule in the configuration in the following ways depending on how you ingest the logs.

- **Filebeat Ingestion:** If you ingest logs into DX OI - Logs using Filebeat, add the **__parsingRule** field to the Filebeat configuration.
  a. Add the **__parsingRule** field to the Filebeat configuration and update the value with the name of the parsing rule as shown in the sample image:

```
###################Custom Logs#####################################

- type: log

  enabled: true
  paths:
    - ${FILEBEAT_LOGS_DIR}/mysql/*

  tags: [ "Custom Logs" ]
  fields:
    logtype: "mysql"
    logformat: "custom"
    __parsingRule: "mysqlparsingrule2"
  fields_under_root: true
  clean_removed: false
  exclude_files: ['.gz$','.zip$','.tar$','.bz2$','.tgz$','.lz$']

##################Generic Logs#######################################
```

NOTE
Ensure that this field value and the **Name** field value on the **Create Log Parsing Rule** page match.

b. Ingest the data using Filebeat.

NOTE
Before you ingest the data, ensure that the parsing rule is enabled. If the rule is disabled, logs ingested using the rule are dropped and are not ingested into DX OI - Logs.

c. Discover and visualize the data in the dashboard.

- **Third-party Agent or Tool Ingestion:** If you ingest logs into DX OI - Logs using a third-party agent or tool, ensure that the log event payload sent to DX OI - Logs has the following mandatory attributes:

Parameter	Description
tenant_id	Tenant ID or the cohort ID of the tenant.
logtype	Type of the log events.
logformat	Sends the log format as "custom".
__parsingRule	Name of the parsing rule

Sample Payload:

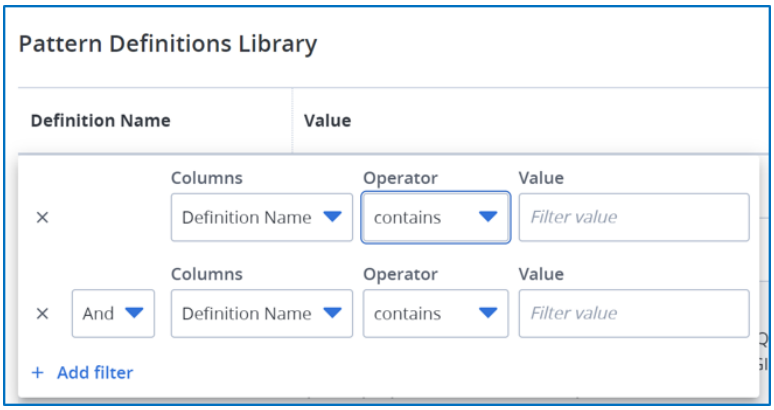
```
{
    "tenant_id": "{{TENANT_ID_UPPERCASE}}",
    "logtype": "MySQL",
    "logformat": "custom",
    "__parsingRule": "my_sql_parsing_rule",
    "timestamp": "{{Log Event Timestamp}}",
    "message": "2015-10-26 19:35:15 12955 [Warning] Shutting down plugin 'InnoDB"
}
```

Built-in Definitions Reference

The Built-in pattern definitions are out-of-the-box definitions that you can reuse in the pattern. Click **Show built in definitions** in the pattern definition section to display the **Pattern Definitions Library** list.

NOTE

You can sort and also filter the list both by the definition name and value. Hover over the name or value to display the sort and filter options.



The following table lists the in-built definitions:

Definition Name	Value
BACULA_CAPACITY	%{INT}{1,3}(,%{INT}{3})*
BACULA_DEVICE	%{USER}
BACULA_DEVICEPATH	%{UNIXPATH}
BACULA_HOST	[a-zA-Z0-9-]+
BACULA_JOB	%{USER}
BACULA_LOG_ALL_RECORDS_PRUNED	All records pruned from Volume \\\"%{BACULA_VOLUME:volume}\\\"; marking it \\\"Purged\\\"
BACULA_LOG_BEGIN_PRUNE_FILES	Begin pruning files.
BACULA_LOG_BEGIN_PRUNE_JOBS	Begin pruning Jobs older than %{INT} month %{INT} days.
BACULA_LOG_CANCELLING	Canceling duplicate JobId=%{INT}
BACULA_LOG_CLIENT_RBJ	shell command: run ClientRunBeforeJob \\\"%{GREEDYDATA:runjob}\\\"
BACULA_LOG_DIFF_FS	\\s+%{UNIXPATH} is a different filesystem. Will not descend from %{UNIXPATH} into it.
BACULA_LOG_DUPLICATE	Fatal error: JobId %{INT:duplicate} already running. Duplicate job is not allowed.
BACULA_LOG_END_VOLUME	End of medium on Volume \\\"%{BACULA_VOLUME:volume}\\\" Bytes= %{BACULA_CAPACITY} Blocks=%{BACULA_CAPACITY} at %{MONTHDAY}-%{MONTH}-%{YEAR} %{HOUR}:%{MINUTE}.
BACULA_LOG_ENDPRUNE	End auto prune.
BACULA_LOG_FATAL_CONN	Fatal error: bsock.c:133 Unable to connect to (Client: %{BACULA_HOST:client}|Storage daemon) on %{HOSTNAME}:%{POSINT}. ERR=(?<berror>%{GREEDYDATA})
BACULA_LOG_JOB	(Error:)?Bacula %{BACULA_HOST} %{BACULA_VERSION} \\(%{BACULA_VERSION}\\):
BACULA_LOG_JOBEND	Job write elapsed time = %{DATA:elapsed}, Transfer rate = %{NUMBER} (K|M|G)? Bytes/second
BACULA_LOG_MARKCANCEL	JobId %{INT}, Job %{BACULA_JOB:job} marked to be canceled.
BACULA_LOG_MAX_CAPACITY	User-defined maximum volume capacity %{BACULA_CAPACITY} exceeded on device \\\"%{BACULA_DEVICE:device}\\\" \\(%{BACULA_DEVICEPATH}\\)
BACULA_LOG_MAXSTART	Fatal error: Job canceled because max start delay time exceeded.
BACULA_LOG_NEW_LABEL	Labeled new Volume \\\"%{BACULA_VOLUME:volume}\\\" on device \\\"%{BACULA_DEVICE:device}\\\" \\(%{BACULA_DEVICEPATH}\\).

Definition Name	Value
BACULA_LOG_NEW_MOUNT	New volume \\\"%{BACULA_VOLUME:volume}\\\" mounted on device \\\"%{BACULA_DEVICE:device}\\\" \\(%{BACULA_DEVICEPATH}\\) at %{MONTHDAY}-%{MONTH}-%{YEAR} %{HOUR}:%{MINUTE}.
BACULA_LOG_NEW_VOLUME	Created new Volume \\\"%{BACULA_VOLUME:volume}\\\" in catalog.
BACULA_LOG_NO_AUTH	Fatal error: Unable to authenticate with File daemon at %{HOSTNAME}. Possible causes:
BACULA_LOG_NO_CONNECT	Warning: bsock.c:127 Could not connect to (Client: %{BACULA_HOST:client}|Storage daemon) on %{HOSTNAME}:%{POSINT}. ERR=(?<berror>%{GREEDYDATA})
BACULA_LOG_NOJOBS	There are no more Jobs associated with Volume \\\"%{BACULA_VOLUME:volume}\\\". Marking it purged.
BACULA_LOG_NOJOBSTAT	Fatal error: No Job status returned from FD.
BACULA_LOG_NOOPEN	\\s+Cannot open %{DATA}: ERR=%{GREEDYDATA:berror}
BACULA_LOG_NOOPENDIR	\\s+Could not open directory %{DATA}: ERR=%{GREEDYDATA:berror}
BACULA_LOG_NOPRIOR	No prior Full backup Job record found.
BACULA_LOG_NOPRUNE_FILES	No Files found to prune.
BACULA_LOG_NOPRUNE_JOBS	No Jobs found to prune.
BACULA_LOG_NOSTAT	\\s+Could not stat %{DATA}: ERR=%{GREEDYDATA:berror}
BACULA_LOG_NOSUIT	No prior or suitable full backup found in the catalog. Doing FULL backup.
BACULA_LOG_PRUNED_FILES	Pruned Files from %{INT} Jobs* for client %{BACULA_HOST:client} from catalog.
BACULA_LOG_PRUNED_JOBS	Pruned %{INT} Jobs* for client %{BACULA_HOST:client} from catalog.
BACULA_LOG_READYAPPEND	Ready to append to end of Volume \\\"%{BACULA_VOLUME:volume}\\\" size=%{INT}
BACULA_LOG_STARTJOB	Start Backup JobId %{INT}, Job=%{BACULA_JOB:job}
BACULA_LOG_STARTRESTORE	Start Restore Job %{BACULA_JOB:job}
BACULA_LOG_USEDEVICE	Using Device \\\"%{BACULA_DEVICE:device}\\\"
BACULA_LOG_VOLUME_PREVWRITTEN	Volume \\\"%{BACULA_VOLUME:volume}\\\" previously written, moving to end of data.
BACULA_LOG_VSS	(Generate)?VSS (Writer)?
BACULA_LOG_WROTE_LABEL	Wrote label to prelabeled Volume \\\"%{BACULA_VOLUME:volume}\\\" on device \\\"%{BACULA_DEVICE}\\\" \\(%{BACULA_DEVICEPATH}\\)

Definition Name	Value
BACULA_LOGLINE	%{BACULA_TIMESTAMP:bts} %{BACULA_HOST:hostname} JobId %{INT:jobid}: (%{BACULA_LOG_MAX_CAPACITY}| %{BACULA_LOG_END_VOLUME}|%{BACULA_LOG_NEW_VOLUME}| %{BACULA_LOG_NEW_LABEL}|%{BACULA_LOG_WROTE_LABEL}| %{BACULA_LOG_NEW_MOUNT}|%{BACULA_LOG_NOOPEN}| %{BACULA_LOG_NOOPENDIR}|%{BACULA_LOG_NOSTAT}| %{BACULA_LOG_NOJOBS}|%{BACULA_LOG_ALL_RECORDS_PRUNED}| %{BACULA_LOG_BEGIN_PRUNE_JOBS}|%{BACULA_LOG_BEGIN_PRUNE_FILES}| %{BACULA_LOG_PRUNED_JOBS}|%{BACULA_LOG_PRUNED_FILES}| %{BACULA_LOG_ENDPRUNE}|%{BACULA_LOG_STARTJOB}| %{BACULA_LOG_STARTRESTORE}|%{BACULA_LOG_USEDEVICE}| %{BACULA_LOG_DIFF_FS}|%{BACULA_LOG_JOBEND}| %{BACULA_LOG_NOPRUNE_JOBS}|%{BACULA_LOG_NOPRUNE_FILES}| %{BACULA_LOG_VOLUME_PREVWRITTEN}|%{BACULA_LOG_READYAPPEND}| %{BACULA_LOG_CANCELLING}|%{BACULA_LOG_MARKCANCEL}| %{BACULA_LOG_CLIENT_RBJ}|%{BACULA_LOG_VSS}| %{BACULA_LOG_MAXSTART}|%{BACULA_LOG_DUPLICATE}| %{BACULA_LOG_NOJOBSTAT}|%{BACULA_LOG_FATAL_CONN}| %{BACULA_LOG_NO_CONNECT}|%{BACULA_LOG_NO_AUTH}| %{BACULA_LOG_NOSUIT}|%{BACULA_LOG_JOB}|%{BACULA_LOG_NOPRIOR})
BACULA_TIMESTAMP	%{MONTHDAY}-%{MONTH} %{HOUR}:%{MINUTE}
BACULA_VERSION	%{USER}
BACULA_VOLUME	%{USER}
BASE10NUM	(?<![0-9.+-])(?>[+-]?(?:(?:[0-9]+(?:\\.[0-9]+)?)|(?:\\.[0-9]+)))
BASE16FLOAT	\\b(?<![0-9A-Fa-f.])(?:[+-]?(?:0x)?(?:(?:[0-9A-Fa-f]+(?:\\.[0-9A-Fa-f]*)?)|(?:\\.[0-9A-Fa-f]+)))\\b
BASE16NUM	(?<![0-9A-Fa-f])(?:[+-]?(?:0x)?(?:[0-9A-Fa-f]+))
BIND9	%{BIND9_TIMESTAMP:timestamp} queries: %{LOGLEVEL:loglevel}: client %{IP:clientip}#%{POSINT:clientport} \\(%{GREEDYDATA:query}\\): query: %{GREEDYDATA:query} IN %{GREEDYDATA:querytype} \\(%{IP:dns}\\)
BIND9_TIMESTAMP	%{MONTHDAY}[-]%{MONTH}[-]%{YEAR} %{TIME}
BRO_CONN	%{NUMBER:ts}\\t%{NOTSPACE:uid}\\t%{IP:orig_h}\\t%{INT:orig_p}\\t%{IP:resp_h}\\t%{INT:resp_p}\\t%{WORD:proto}\\t%{GREEDYDATA:service}\\t%{NUMBER:duration}\\t%{NUMBER:orig_bytes}\\t%{NUMBER:resp_bytes}\\t%{GREEDYDATA:conn_state}\\t%{GREEDYDATA:local_orig}\\t%{GREEDYDATA:missed_bytes}\\t%{GREEDYDATA:history}\\t%{GREEDYDATA:orig_pkts}\\t%{GREEDYDATA:orig_ip_bytes}\\t%{GREEDYDATA:resp_pkts}\\t%{GREEDYDATA:resp_ip_bytes}\\t%{GREEDYDATA:tunnel_parents}
BRO_DNS	%{NUMBER:ts}\\t%{NOTSPACE:uid}\\t%{IP:orig_h}\\t%{INT:orig_p}\\t%{IP:resp_h}\\t%{INT:resp_p}\\t%{WORD:proto}\\t%{INT:trans_id}\\t%{GREEDYDATA:query}\\t%{GREEDYDATA:qclass}\\t%{GREEDYDATA:qclass_name}\\t%{GREEDYDATA:qtype}\\t%{GREEDYDATA:qtype_name}\\t%{GREEDYDATA:rcode}\\t%{GREEDYDATA:rcode_name}\\t%{GREEDYDATA:AA}\\t%{GREEDYDATA:TC}\\t%{GREEDYDATA:RD}\\t%{GREEDYDATA:RA}\\t%{GREEDYDATA:Z}\\t%{GREEDYDATA:answers}\\t%{GREEDYDATA:TTLs}\\t%{GREEDYDATA:rejected}

Definition Name	Value
BRO_FILES	%{NUMBER:ts}\\t%{NOTSPACE:fuid}\\t%{IP:tx_hosts}\\t%{IP:rx_hosts}\\t%{NOTSPACE:conn_uids}\\t%{GREEDYDATA:source}\\t%{GREEDYDATA:depth}\\t%{GREEDYDATA:analyzers}\\t%{GREEDYDATA:mime_type}\\t%{GREEDYDATA:filename}\\t%{GREEDYDATA:duration}\\t%{GREEDYDATA:local_orig}\\t%{GREEDYDATA:is_orig}\\t%{GREEDYDATA:seen_bytes}\\t%{GREEDYDATA:total_bytes}\\t%{GREEDYDATA:missing_bytes}\\t%{GREEDYDATA:overflow_bytes}\\t%{GREEDYDATA:timedout}\\t%{GREEDYDATA:parent_fuid}\\t%{GREEDYDATA:md5}\\t%{GREEDYDATA:sha1}\\t%{GREEDYDATA:sha256}\\t%{GREEDYDATA:extracted}
BRO_HTTP	%{NUMBER:ts}\\t%{NOTSPACE:uid}\\t%{IP:orig_h}\\t%{INT:orig_p}\\t%{IP:resp_h}\\t%{INT:resp_p}\\t%{INT:trans_depth}\\t%{GREEDYDATA:method}\\t%{GREEDYDATA:domain}\\t%{GREEDYDATA:uri}\\t%{GREEDYDATA:referrer}\\t%{GREEDYDATA:user_agent}\\t%{NUMBER:request_body_len}\\t%{NUMBER:response_body_len}\\t%{GREEDYDATA:status_code}\\t%{GREEDYDATA:status_msg}\\t%{GREEDYDATA:info_code}\\t%{GREEDYDATA:info_msg}\\t%{GREEDYDATA:filename}\\t%{GREEDYDATA:bro_tags}\\t%{GREEDYDATA:username}\\t%{GREEDYDATA:password}\\t%{GREEDYDATA:proxied}\\t%{GREEDYDATA:orig_fuids}\\t%{GREEDYDATA:orig_mime_types}\\t%{GREEDYDATA:resp_fuids}\\t%{GREEDYDATA:resp_mime_types}
CATALINA_DATESTAMP	%{MONTH} %{MONTHDAY}, 20%{YEAR} %{HOUR}:?%{MINUTE}(?::?%{SECOND}) (?:AM|PM)
CATALINALOG	%{CATALINA_DATESTAMP:timestamp} %{JAVACLASS:class} %{JAVALOGMESSAGE:logmessage}
CISCO_ACTION	Built|Teardown|Deny|Denied|denied|requested|permitted|denied by ACL|discarded|est-allowed|Dropping|created|deleted
CISCO_DIRECTION	Inbound|inbound|Outbound|outbound
CISCO_INTERVAL	first hit|%{INT}-second interval
CISCO_REASON	Duplicate TCP SYN|Failed to locate egress interface|Invalid transport field|No matching connection|DNS Response|DNS Query|(?:%{WORD}\\s*)*
CISCO_TAGGED_SYSLOG	^<%{POSINT:syslog_pri}>%{CISCOTIMESTAMP:timestamp} (%{SYSLOGHOST:sysloghost})? ?: %%{CISCOTAG:ciscotag}:
CISCO_XLATE_TYPE	static|dynamic
CISCOFW104001	\\((?:Primary|Secondary)\\) Switching to ACTIVE - %{GREEDYDATA:switch_reason}
CISCOFW104002	\\((?:Primary|Secondary)\\) Switching to STANDBY - %{GREEDYDATA:switch_reason}
CISCOFW104003	\\((?:Primary|Secondary)\\) Switching to FAILED\\.
CISCOFW104004	\\((?:Primary|Secondary)\\) Switching to OK\\.
CISCOFW105003	\\((?:Primary|Secondary)\\) Monitoring on [li]nterface %{GREEDYDATA:interface_name} waiting
CISCOFW105004	\\((?:Primary|Secondary)\\) Monitoring on [li]nterface %{GREEDYDATA:interface_name} normal
CISCOFW105005	\\((?:Primary|Secondary)\\) Lost Failover communications with mate on [li]nterface %{GREEDYDATA:interface_name}
CISCOFW105008	\\((?:Primary|Secondary)\\) Testing [li]nterface %{GREEDYDATA:interface_name}
CISCOFW105009	\\((?:Primary|Secondary)\\) Testing on [li]nterface %{GREEDYDATA:interface_name} (?:Passed|Failed)

Definition Name	Value
CISCOFW106001	%{CISCO_DIRECTION:direction} %{WORD:protocol} connection %{CISCO_ACTION:action} from %{IP:src_ip}/%{INT:src_port} to %{IP:dst_ip}/%{INT:dst_port} flags %{GREEDYDATA:tcp_flags} on interface %{GREEDYDATA:interface}
CISCOFW106006_106007_106010	%{CISCO_ACTION:action} %{CISCO_DIRECTION:direction} %{WORD:protocol} (?:from|src) %{IP:src_ip}/%{INT:src_port}(\\(%{DATA:src_fwuser}\\))? (?:to|dst) %{IP:dst_ip}/%{INT:dst_port}(\\(%{DATA:dst_fwuser}\\))? (?:on interface %{DATA:interface}|due to %{CISCO_REASON:reason})
CISCOFW106014	%{CISCO_ACTION:action} %{CISCO_DIRECTION:direction} %{WORD:protocol} src %{DATA:src_interface}:%{IP:src_ip}(\\(%{DATA:src_fwuser}\\))? dst %{DATA:dst_interface}:%{IP:dst_ip}(\\(%{DATA:dst_fwuser}\\))? \\(type %{INT:icmp_type}, code %{INT:icmp_code}\\)
CISCOFW106015	%{CISCO_ACTION:action} %{WORD:protocol} \\(%{DATA:policy_id}\\) from %{IP:src_ip}/%{INT:src_port} to %{IP:dst_ip}/%{INT:dst_port} flags %{DATA:tcp_flags} on interface %{GREEDYDATA:interface}
CISCOFW106021	%{CISCO_ACTION:action} %{WORD:protocol} reverse path check from %{IP:src_ip} to %{IP:dst_ip} on interface %{GREEDYDATA:interface}
CISCOFW106023	%{CISCO_ACTION:action}(protocol)? %{WORD:protocol} src %{DATA:src_interface}:%{DATA:src_ip}(/%{INT:src_port})?(\\(%{DATA:src_fwuser}\\))? dst %{DATA:dst_interface}:%{DATA:dst_ip}(/%{INT:dst_port})?(\\(%{DATA:dst_fwuser}\\))?(\\(type %{INT:icmp_type}, code %{INT:icmp_code}\\))? by access-group \"?%{DATA:policy_id}\"? \\[%{DATA:hashcode1}, %{DATA:hashcode2}\\]
CISCOFW106100	access-list %{NOTSPACE:policy_id} %{CISCO_ACTION:action} %{WORD:protocol} %{DATA:src_interface}/%{IP:src_ip}\\(%{INT:src_port}\\)(\\(%{DATA:src_fwuser}\\))? -> %{DATA:dst_interface}/%{IP:dst_ip}\\(%{INT:dst_port}\\)(\\(%{DATA:src_fwuser}\\))? hit-cnt %{INT:hit_count} %{CISCO_INTERVAL:interval} \\[%{DATA:hashcode1}, %{DATA:hashcode2}\\]
CISCOFW106100_2_3	access-list %{NOTSPACE:policy_id} %{CISCO_ACTION:action} %{WORD:protocol} for user '%{DATA:src_fwuser}' %{DATA:src_interface}/%{IP:src_ip}\\(%{INT:src_port}\\) -> %{DATA:dst_interface}/%{IP:dst_ip}\\(%{INT:dst_port}\\) hit-cnt %{INT:hit_count} %{CISCO_INTERVAL:interval} \\[%{DATA:hashcode1}, %{DATA:hashcode2}\\]
CISCOFW110002	%{CISCO_REASON:reason} for %{WORD:protocol} from %{DATA:src_interface}:%{IP:src_ip}/%{INT:src_port} to %{IP:dst_ip}/%{INT:dst_port}
CISCOFW302010	%{INT:connection_count} in use, %{INT:connection_count_max} most used
CISCOFW302013_302014_302015_302016	%{CISCO_ACTION:action}(?: %{CISCO_DIRECTION:direction})? %{WORD:protocol} connection %{INT:connection_id} for %{DATA:src_interface}:%{IP:src_ip}/%{INT:src_port}(\\(%{IP:src_mapped_ip}/%{INT:src_mapped_port}\\))?(\\(%{DATA:src_fwuser}\\))? to %{DATA:dst_interface}:%{IP:dst_ip}/%{INT:dst_port}(\\(%{IP:dst_mapped_ip}/%{INT:dst_mapped_port}\\))?(\\(%{DATA:dst_fwuser}\\))?(duration %{TIME:duration} bytes %{INT:bytes})?(?: %{CISCO_REASON:reason})?(\\(%{DATA:user}\\))?
CISCOFW302020_302021	%{CISCO_ACTION:action}(?: %{CISCO_DIRECTION:direction})? %{WORD:protocol} connection for faddr %{IP:dst_ip}/%{INT:icmp_seq_num}(?:\\(%{DATA:fwuser}\\))? gaddr %{IP:src_xlated_ip}/%{INT:icmp_code_xlated} laddr %{IP:src_ip}/%{INT:icmp_code}(\\(%{DATA:user}\\))?
CISCOFW304001	%{IP:src_ip}(\\(%{DATA:src_fwuser}\\))? Accessed URL %{IP:dst_ip}:%{GREEDYDATA:dst_url}

Definition Name	Value
CISCOFW305011	%{CISCO_ACTION:action} %{CISCO_XLATE_TYPE:xlate_type} %{WORD:protocol} translation from %{DATA:src_interface}:%{IP:src_ip}(/%{INT:src_port})?(\\(%{DATA:src_fwuser}\\))? to %{DATA:src_xlated_interface}:%{IP:src_xlated_ip}/%{DATA:src_xlated_port}
CISCOFW313001_313004_313008	%{CISCO_ACTION:action} %{WORD:protocol} type=%{INT:icmp_type}, code=%{INT:icmp_code} from %{IP:src_ip} on interface %{DATA:interface}(to %{IP:dst_ip})?
CISCOFW313005	%{CISCO_REASON:reason} for %{WORD:protocol} error message: %{WORD:err_protocol} src %{DATA:err_src_interface}:%{IP:err_src_ip}(\\(%{DATA:err_src_fwuser}\\))? dst %{DATA:err_dst_interface}:%{IP:err_dst_ip}(\\(%{DATA:err_dst_fwuser}\\))? \\(type %{INT:err_icmp_type}, code %{INT:err_icmp_code}\\) on %{DATA:interface} interface\\. Original IP payload: %{WORD:protocol} src %{IP:orig_src_ip}/%{INT:orig_src_port}(\\(%{DATA:orig_src_fwuser}\\))? dst %{IP:orig_dst_ip}/%{INT:orig_dst_port}(\\(%{DATA:orig_dst_fwuser}\\))?
CISCOFW321001	Resource '%{WORD:resource_name}' limit of %{POSINT:resource_limit} reached for system
CISCOFW402117	%{WORD:protocol}: Received a non-IPSec packet \\(protocol= %{WORD:orig_protocol}\\) from %{IP:src_ip} to %{IP:dst_ip}
CISCOFW402119	%{WORD:protocol}: Received an %{WORD:orig_protocol} packet \\(SPI= %{DATA:spi}, sequence number= %{DATA:seq_num}\\) from %{IP:src_ip} \\(user= %{DATA:user}\\) to %{IP:dst_ip} that failed anti-replay checking
CISCOFW419001	%{CISCO_ACTION:action} %{WORD:protocol} packet from %{DATA:src_interface}:%{IP:src_ip}/%{INT:src_port} to %{DATA:dst_interface}:%{IP:dst_ip}/%{INT:dst_port}, reason: %{GREEDYDATA:reason}
CISCOFW419002	%{CISCO_REASON:reason} from %{DATA:src_interface}:%{IP:src_ip}/%{INT:src_port} to %{DATA:dst_interface}:%{IP:dst_ip}/%{INT:dst_port} with different initial sequence number
CISCOFW500004	%{CISCO_REASON:reason} for protocol=%{WORD:protocol}, from %{IP:src_ip}/%{INT:src_port} to %{IP:dst_ip}/%{INT:dst_port}
CISCOFW602303_602304	%{WORD:protocol}: An %{CISCO_DIRECTION:direction} %{GREEDYDATA:tunnel_type} SA \\(SPI= %{DATA:spi}\\) between %{IP:src_ip} and %{IP:dst_ip} \\(user= %{DATA:user}\\) has been %{CISCO_ACTION:action}
CISCOFW710001_710002_710003_710005_710006	%{WORD:protocol} (?:request|access) %{CISCO_ACTION:action} from %{IP:src_ip}/%{INT:src_port} to %{DATA:dst_interface}:%{IP:dst_ip}/%{INT:dst_port}
CISCOFW713172	Group = %{GREEDYDATA:group}, IP = %{IP:src_ip}, Automatic NAT Detection Status:\\s+Remote end\\s*%{DATA:is_remote_natted}\\s*behind a NAT device\\s+This\\s+end\\s*%{DATA:is_local_natted}\\s*behind a NAT device
CISCOFW733100	\\[\\s*%{DATA:drop_type}\\s*\\] drop %{DATA:drop_rate_id} exceeded. Current burst rate is %{INT:drop_rate_current_burst} per second, max configured rate is %{INT:drop_rate_max_burst}; Current average rate is %{INT:drop_rate_current_avg} per second, max configured rate is %{INT:drop_rate_max_avg}; Cumulative total count is %{INT:drop_total_count}
CISCOMAC	(?:(?:[A-Fa-f0-9]{4}\\.){2}[A-Fa-f0-9]{4})
CISCOTAG	[A-Z0-9]+-%{INT}-(?:[A-Z0-9_]+)
CISCOTIMESTAMP	%{MONTH} +%{MONTHDAY}(?: %{YEAR})? %{TIME}

Definition Name	Value
CLOUDFRONT_ACCESS_LOG	(?<timestamp>%{YEAR}-%{MONTHNUM}-%{MONTHDAY}\\t%{TIME})\\t%{WORD:x_edge_location}\\t(?:%{NUMBER:sc_bytes:int}|-)\\t%{IPORHOST:clientip}\\t%{WORD:cs_method}\\t%{HOSTNAME:cs_host}\\t%{NOTSPACE:cs_uri_stem}\\t%{NUMBER:sc_status:int}\\t%{GREEDYDATA:referrer}\\t%{GREEDYDATA:agent}\\t%{GREEDYDATA:cs_uri_query}\\t%{GREEDYDATA:cookies}\\t%{WORD:x_edge_result_type}\\t%{NOTSPACE:x_edge_request_id}\\t%{HOSTNAME:x_host_header}\\t%{URIPROTO:cs_protocol}\\t%{INT:cs_bytes:int}\\t%{GREEDYDATA:time_taken:float}\\t%{GREEDYDATA:x_forwarded_for}\\t%{GREEDYDATA:ssl_protocol}\\t%{GREEDYDATA:ssl_cipher}\\t%{GREEDYDATA:x_edge_response_result_type}
COMBINEDAPACHELOG	%{COMMONAPACHELOG} %{QS:referrer} %{QS:agent}
COMMONAPACHELOG	%{IPORHOST:clientip} %{HTTPDUSER:ident} %{USER:auth} \ \[%{HTTPDATE:timestamp}\\] \"(?:%{WORD:verb} %{NOTSPACE:request}(?: HTTP/%{NUMBER:httpversion})?|%{DATA:rawrequest})\" %{NUMBER:response} (?:%{NUMBER:bytes}|-)
COMMONMAC	(?:(?:[A-Fa-f0-9]{2}:){5}[A-Fa-f0-9]{2})
CRON_ACTION	[A-Z]+
CRONLOG	%{SYSLOGBASE} \\(%{USER:user}\\) %{CRON_ACTION:action} \\(%{DATA:message}\\)
DATA	.*?
DATE	%{DATE_US}|%{DATE_EU}
DATE_EU	%{MONTHDAY}[./-]%{MONTHNUM}[./-]%{YEAR}
DATE_US	%{MONTHNUM}[/-]%{MONTHDAY}[/-]%{YEAR}
DATESTAMP	%{DATE}[-]%{TIME}
DATESTAMP_EVENTLOG	%{YEAR}%{MONTHNUM2}%{MONTHDAY}%{HOUR}%{MINUTE}%{SECOND}
DATESTAMP_OTHER	%{DAY} %{MONTH} %{MONTHDAY} %{TIME} %{TZ} %{YEAR}
DATESTAMP_RFC2822	%{DAY}, %{MONTHDAY} %{MONTH} %{YEAR} %{TIME} %{ISO8601_TIMEZONE}
DATESTAMP_RFC822	%{DAY} %{MONTH} %{MONTHDAY} %{YEAR} %{TIME} %{TZ}
DAY	(?:Mon(?:day)?|Tue(?:sday)?|Wed(?:nesday)?|Thu(?:rsday)?|Fri(?:day)?|Sat(?:urday)?|Sun(?:day)?)
ELB_ACCESS_LOG	%{TIMESTAMP_ISO8601:timestamp} %{NOTSPACE:elb} %{IP:clientip}:%{INT:clientport:int} (?:(%{IP:backendip}:?:%{INT:backendport:int})|-) %{NUMBER:request_processing_time:float} %{NUMBER:backend_processing_time:float} %{NUMBER:response_processing_time:float} %{INT:response:int} %{INT:backend_response:int} %{INT:received_bytes:int} %{INT:bytes:int} \"%{ELB_REQUEST_LINE}\"
ELB_REQUEST_LINE	(?:%{WORD:verb} %{ELB_URI:request}(?: HTTP/%{NUMBER:httpversion})?|%{DATA:rawrequest})
ELB_URI	%{URIPROTO:proto}://(?:%{USER}(?::[^@]*)?@)?(?:%{URIHOST:urihost})?(?:%{ELB_URIPATHPARAM})?
ELB_URIPATHPARAM	%{URIPATH:path}(?:%{URIPARAM:params})?
EMAILADDRESS	%{EMAILLOCALPART}@%{HOSTNAME}
EMAILLOCALPART	[a-zA-Z][a-zA-Z0-9_.+-=:]+

Definition Name	Value
EXIM_DATE	%{YEAR:exim_year}-%{MONTHNUM:exim_month}-%{MONTHDAY:exim_day} %{TIME:exim_time}
EXIM_EXCLUDE_TERMS	(Message is frozen|(Start|End) queue run| Warning: | retry time not reached | no (IP address|host name) found for (IP address|host) | unexpected disconnection while reading SMTP command | no immediate delivery: |another process is handling this message)
EXIM_FLAGS	(<=|[-=>*]>|[*]{2}|==)
EXIM_HEADER_ID	(id=%{NOTSPACE:exim_header_id})
EXIM_INTERFACE	(I=\\[%{IP:exim_interface}\\](:%{NUMBER:exim_interface_port}))
EXIM_MSG_SIZE	(S=%{NUMBER:exim_msg_size})
EXIM_MSGID	[0-9A-Za-z]{6}-[0-9A-Za-z]{6}-[0-9A-Za-z]{2}
EXIM_PID	\\[%{POSINT}\\]
EXIM_PROTOCOL	(P=%{NOTSPACE:protocol})
EXIM_QT	((\\d+y)?(\\d+w)?(\\d+d)?(\\d+h)?(\\d+m)?(\\d+s)?)
EXIM_REMOTE_HOST	(H=(%{NOTSPACE:remote_hostname})?(\\(%{NOTSPACE:remote_heloname}\\))?\ \[%{IP:remote_host}\\])
EXIM_SUBJECT	(T=%{QS:exim_subject})
GREEDYDATA	.*
HAPROXYCAPTUREDREQUESTHEADERS	%{DATA:captured_request_headers}
HAPROXYCAPTUREDRESPONSEHEADERS	%{DATA:captured_response_headers}
HAPROXYDATE	%{MONTHDAY:haproxy_monthday}/%{MONTH:haproxy_month}/%{YEAR:haproxy_year }:%{HAPROXYTIME:haproxy_time}.%{INT:haproxy_milliseconds}
HAPROXYHTTP	(?:%{SYSLOGTIMESTAMP:syslog_timestamp}| %{TIMESTAMP_ISO8601:timestamp8601}) %{IPORHOST:syslog_server} %{SYSLOGPROG}: %{HAPROXYHTTPBASE}
HAPROXYHTTPBASE	%{IP:client_ip}:%{INT:client_port} \\[%{HAPROXYDATE:accept_date}\ \] %{NOTSPACE:frontend_name} %{NOTSPACE:backend_name}/ %{NOTSPACE:server_name} %{INT:time_request}/%{INT:time_queue}/ %{INT:time_backend_connect}/%{INT:time_backend_response}/ %{NOTSPACE:time_duration} %{INT:http_status_code} %{NOTSPACE:bytes_read} %{DATA:captured_request_cookie} %{DATA:captured_response_cookie} %{NOTSPACE:termination_state} %{INT:actconn}/%{INT:feconn}/ %{INT:beconn}/%{INT:srvconn}/%{NOTSPACE:retries} %{INT:srv_queue}/ %{INT:backend_queue} (\\{%{HAPROXYCAPTUREDREQUESTHEADERS}\\})? ( )?(\\{%{HAPROXYCAPTUREDRESPONSEHEADERS}\\})?( )?\"(<BADREQ>| (%{WORD:http_verb} (%{URIPROTO:http_proto}://)?(?:%{USER:http_user}(?::[^@]*)? @)?(?:%{URIHOST:http_host})?(?:%{URIPATHPARAM:http_request})?(HTTP/ %{NUMBER:http_version})?))?\"

Definition Name	Value
HAPROXYTCP	(?:%{SYSLOGTIMESTAMP:syslog_timestamp}| %{TIMESTAMP_ISO8601:timestamp8601}) %{IPORHOST:syslog_server} %{SYSLOGPROG}: %{IP:client_ip}:%{INT:client_port} \ \[%{HAPROXYDATE:accept_date}\\] %{NOTSPACE:frontend_name} %{NOTSPACE:backend_name}/%{NOTSPACE:server_name} %{INT:time_queue}/%{INT:time_backend_connect}/%{NOTSPACE:time_duration} %{NOTSPACE:bytes_read} %{NOTSPACE:termination_state} %{INT:actconn}/ %{INT:feconn}/%{INT:beconn}/%{INT:srvconn}/%{NOTSPACE:retries} %{INT:srv_queue}/ %{INT:backend_queue}
HAPROXYTIME	(?!<[0-9])%{HOUR:haproxy_hour}:%{MINUTE:haproxy_minute}(?::%{SECOND:haproxy_ second})(?![0-9])
HOSTNAME	\\b(?:[0-9A-Za-z][0-9A-Za-z-]{0,62})(?:\\.(?:[0-9A-Za-z][0-9A-Za-z-]{0,62}))*(\\.?|\\b)
HOSTPORT	%{IPORHOST}:%{POSINT}
HOUR	(?:2[0123]|[01]?[0-9])
HTTPD_ERRORLOG	%{HTTPD20_ERRORLOG}|%{HTTPD24_ERRORLOG}
HTTPD20_ERRORLOG	\\[%{HTTPDERROR_DATE:timestamp}\\] \\[%{LOGLEVEL:loglevel}\\] (?:\\[client %{IPORHOST:clientip}\\]){0,1}%{GREEDYDATA:errormsg}
HTTPD24_ERRORLOG	\\[%{HTTPDERROR_DATE:timestamp}\\] \\[(%{WORD:module})?: %{LOGLEVEL:loglevel}\\] \\[pid %{POSINT:pid}(:tid %{NUMBER:tid})?\\](\ \(%{POSINT:proxy_errorcode}\\)%{DATA:proxy_errormessage}:)?(\\[client %{IPORHOST:client}:%{POSINT:clientport}\\])?(%{DATA:errorcode}:)? %{GREEDYDATA:message}
HTTPDATE	%{MONTHDAY}/%{MONTH}/%{YEAR}:%{TIME} %{INT}
HTTPDERROR_DATE	%{DAY} %{MONTH} %{MONTHDAY} %{TIME} %{YEAR}
HTTPDUSER	%{EMAILADDRESS}|%{USER}
INT	(?:[+-]?(?:[0-9]+))
IP	(?:%{IPV6}|%{IPV4})
IPORHOST	(?:%{IP}|%{HOSTNAME})
IPV4	(?<![0-9])(?:(?:[0-1]?[0-9]{1,2}|2[0-4][0-9]|25[0-5])[.](?:[0-1]?[0-9]{1,2}|2[0-4][0-9]|25[0-5])[.](?:[0-1]?[0-9]{1,2}|2[0-4][0-9]|25[0-5])[.](?:[0-1]?[0-9]{1,2}|2[0-4][0-9]|25[0-5]))(?![0-9])
IPV6	((([0-9A-Fa-f]{1,4}:){7}([0-9A-Fa-f]{1,4}|:))|(([0-9A-Fa-f]{1,4}:){6}(:[0-9A-Fa-f]{1,4}|((25[0-5]| 2[0-4]\\d|1\\d\\d|[1-9]?\\d)(\\.(25[0-5]|2[0-4]\\d|1\\d\\d|[1-9]?\\d)){3})|:))|(([0-9A-Fa-f]{1,4}:){5 }(((:[0-9A-Fa-f]{1,4}){1,2})|:((25[0-5]|2[0-4]\\d|1\\d\\d|[1-9]?\\d)(\\.(25[0-5]|2[0-4]\\d|1\\d\\d|[1 -9]?\\d)){3})|:))|(([0-9A-Fa-f]{1,4}:){4}(((:[0-9A-Fa-f]{1,4}){1,3})|((:[0-9A-Fa-f]{1,4})?:((25[0-5]|2[0-4]\\d|1\\d\\d|[1-9]?\\d)(\\.(25[0-5]|2[0-4]\\d|1\\d\\d|[1-9]?\\d)){3}))|:))|(([0-9A-Fa-f]{1,4}:) {3}(((:[0-9A-Fa-f]{1,4}){1,4})|((:[0-9A-Fa-f]{1,4}){0,2}:((25[0-5]|2[0-4]\\d|1\\d\\d|[1-9]?\\d)(\\.(25[0-5]|2[0-4]\\d|1\\d\\d|[1-9]?\\d)){3}))|:))|(([0-9A-Fa-f]{1,4}:){2}(((:[0-9A-Fa-f]{1,4}){1,5})|((:[0-9A-Fa-f]{1,4}){0,3}:((25[0-5]|2[0-4]\\d|1\\d\\d|[1-9]?\\d)(\\.(25[0-5]|2[0-4]\\d|1\\d\\d|[1-9]? \\d)){3}))|:))|(([0-9A-Fa-f]{1,4}:){1}(((:[0-9A-Fa-f]{1,4}){1,6})|((:[0-9A-Fa-f]{1,4}){0,4}:((25[0- 5]|2[0-4]\\d|1\\d\\d|[1-9]?\\d)(\\.(25[0-5]|2[0-4]\\d|1\\d\\d|[1-9]?\\d)){3}))|:))|(:(((:[0-9A-Fa-f]{1 ,4}){1,7})|((:[0-9A-Fa-f]{1,4}){0,5}:((25[0-5]|2[0-4]\\d|1\\d\\d|[1-9]?\\d)(\\.(25[0-5]|2[0-4]\\d|1\ \d\\d|[1-9]?\\d)){3}))|:)))(%.+)?
ISO8601_HOUR	(?:2[0123]|[01][0-9])
ISO8601_SECOND	(?:%{SECOND}|60)

Definition Name	Value
ISO8601_TIMEZONE	(?:Z|[+-]%{HOUR}(?::?%{MINUTE}))
JAVACLASS	(?:[a-zA-Z$_][a-zA-Z$_0-9]*\\.)*[a-zA-Z$_][a-zA-Z$_0-9]*
JAVAFILE	(?:[A-Za-z0-9_. -]+)
JAVALOGMESSAGE	(.*)
JAVAMETHOD	(?:(<(?:cl)?init>)|[a-zA-Z$_][a-zA-Z$_0-9]*)
JAVASTACKTRACEPART	at %{JAVACLASS:class}\\.%{WORD:method}\\(%{JAVAFILE:file}:%{NUMBER:line}\\)
JAVATHREAD	(?:[A-Z]{2}-Processor[\\d]+)
LOGLEVEL	([Aa]lert|ALERT|[Tt]race|TRACE|[Dd]ebug|DEBUG|[Nn]otice|NOTICE|[Ii]nfo|INFO|[Ww]arn?(?:ing)?|WARN?(?:ING)?|[Ee]rr?(?:or)?|ERR?(?:OR)?|[Cc]rit?(?:ical)?|CRIT?(?:ICAL)?|[Ff]atal|FATAL|[Ss]evere|SEVERE|EMERG(?:ENCY)?|[Ee]merg(?:ency)?)
MAC	(?:%{CISCOMAC}|%{WINDOWSMAC}|%{COMMONMAC})
MAVEN_VERSION	(?:(\\d+)\\.)?(?:(\\d+)\\.)?(*|\\d+)(?:[.-](RELEASE|SNAPSHOT))?
MCOLLECTIVE	., \\[%{TIMESTAMP_ISO8601:timestamp} #%{POSINT:pid}\\]%{SPACE}%{LOGLEVEL:event_level}
MCOLLECTIVEAUDIT	%{TIMESTAMP_ISO8601:timestamp}:
MINUTE	(?:[0-5][0-9])
MONGO_LOG	%{SYSLOGTIMESTAMP:timestamp} \\[%{WORD:component}\\] %{GREEDYDATA:message}
MONGO_QUERY	\\{ (?<={).*(?= } ntoreturn:) \\}
MONGO_SLOWQUERY	%{WORD} %{MONGO_WORDDASH:database}\\.%{MONGO_WORDDASH:collection} %{WORD}: %{MONGO_QUERY:query} %{WORD}:%{NONNEGINT:ntoreturn} %{WORD}:%{NONNEGINT:ntoskip} %{WORD}:%{NONNEGINT:nscanned}.*nreturned: %{NONNEGINT:nreturned}..+ (?<duration>[0-9]+)ms
MONGO_WORDDASH	\\b[\\w-]+\\b
MONGO3_COMPONENT	%{WORD}|-
MONGO3_LOG	%{TIMESTAMP_ISO8601:timestamp} %{MONGO3_SEVERITY:severity} %{MONGO3_COMPONENT:component}%{SPACE}(?:\\[%{DATA:context}\\])? %{GREEDYDATA:message}
MONGO3_SEVERITY	\\w
MONTH	\\b(?:[Jj]an(?:uary|uar)?|[Ff]eb(?:ruary|ruar)?|[Mm](?:a|ä)?r(?:ch|z)?|[Aa]pr(?:il)?|[Mm]a(?:y|i)?|[Jj]un(?:e|i)?|[Jj]ul(?:y|i)?|[Aa]ug(?:ust)?|[Ss]ep(?:tember)?|[Oo](?:c|k)?t(?:ober)?|[Nn]ov(?:ember)?|[Dd]e(?:c|z)(?:ember)?)\\b
MONTHDAY	(?:(?:0[1-9])|(?:[12][0-9])|(?:3[01])|[1-9])
MONTHNUM	(?:0?[1-9]|1[0-2])
MONTHNUM2	(?:0[1-9]|1[0-2])
NAGIOS_CURRENT_HOST_STATE	%{NAGIOS_TYPE_CURRENT_HOST_STATE:nagios_type}: %{DATA:nagios_hostname};%{DATA:nagios_state};%{DATA:nagios_statetype}; %{DATA:nagios_statecode};%{GREEDYDATA:nagios_message}
NAGIOS_CURRENT_SERVICE_STATE	%{NAGIOS_TYPE_CURRENT_SERVICE_STATE:nagios_type}: %{DATA:nagios_hostname};%{DATA:nagios_service};%{DATA:nagios_state}; %{DATA:nagios_statetype};%{DATA:nagios_statecode}; %{GREEDYDATA:nagios_message}

Definition Name	Value
NAGIOS_EC_DISABLE_HOST_CHECK	DISABLE_HOST_CHECK
NAGIOS_EC_DISABLE_HOST_NOTIFICATIONS	DISABLE_HOST_NOTIFICATIONS
NAGIOS_EC_DISABLE_HOST_SVC_NOTIFICATIONS	DISABLE_HOST_SVC_NOTIFICATIONS
NAGIOS_EC_DISABLE_SVC_CHECK	DISABLE_SVC_CHECK
NAGIOS_EC_DISABLE_SVC_NOTIFICATIONS	DISABLE_SVC_NOTIFICATIONS
NAGIOS_EC_ENABLE_HOST_CHECK	ENABLE_HOST_CHECK
NAGIOS_EC_ENABLE_HOST_NOTIFICATIONS	ENABLE_HOST_NOTIFICATIONS
NAGIOS_EC_ENABLE_HOST_SVC_NOTIFICATIONS	ENABLE_HOST_SVC_NOTIFICATIONS
NAGIOS_EC_ENABLE_SVC_CHECK	ENABLE_SVC_CHECK
NAGIOS_EC_ENABLE_SVC_NOTIFICATIONS	ENABLE_SVC_NOTIFICATIONS
NAGIOS_EC_LINE_DISABLE_HOST_CHECK	%{NAGIOS_TYPE_EXTERNAL_COMMAND:nagios_type}: %{NAGIOS_EC_DISABLE_HOST_CHECK:nagios_command}; %{DATA:nagios_hostname}
NAGIOS_EC_LINE_DISABLE_HOST_NOTIFICATIONS	%{NAGIOS_TYPE_EXTERNAL_COMMAND:nagios_type}: %{NAGIOS_EC_DISABLE_HOST_NOTIFICATIONS:nagios_command}; %{GREEDYDATA:nagios_hostname}
NAGIOS_EC_LINE_DISABLE_HOST_SVC_NOTIFICATIONS	%{NAGIOS_TYPE_EXTERNAL_COMMAND:nagios_type}: %{NAGIOS_EC_DISABLE_HOST_SVC_NOTIFICATIONS:nagios_command}; %{GREEDYDATA:nagios_hostname}
NAGIOS_EC_LINE_DISABLE_SVC_CHECK	%{NAGIOS_TYPE_EXTERNAL_COMMAND:nagios_type}: %{NAGIOS_EC_DISABLE_SVC_CHECK:nagios_command}; %{DATA:nagios_hostname};%{DATA:nagios_service}
NAGIOS_EC_LINE_DISABLE_SVC_NOTIFICATIONS	%{NAGIOS_TYPE_EXTERNAL_COMMAND:nagios_type}: %{NAGIOS_EC_DISABLE_SVC_NOTIFICATIONS:nagios_command}; %{DATA:nagios_hostname};%{GREEDYDATA:nagios_service}
NAGIOS_EC_LINE_ENABLE_HOST_CHECK	%{NAGIOS_TYPE_EXTERNAL_COMMAND:nagios_type}: %{NAGIOS_EC_ENABLE_HOST_CHECK:nagios_command}; %{DATA:nagios_hostname}
NAGIOS_EC_LINE_ENABLE_HOST_NOTIFICATIONS	%{NAGIOS_TYPE_EXTERNAL_COMMAND:nagios_type}: %{NAGIOS_EC_ENABLE_HOST_NOTIFICATIONS:nagios_command}; %{GREEDYDATA:nagios_hostname}
NAGIOS_EC_LINE_ENABLE_HOST_SVC_NOTIFICATIONS	%{NAGIOS_TYPE_EXTERNAL_COMMAND:nagios_type}: %{NAGIOS_EC_ENABLE_HOST_SVC_NOTIFICATIONS:nagios_command}; %{GREEDYDATA:nagios_hostname}
NAGIOS_EC_LINE_ENABLE_SVC_CHECK	%{NAGIOS_TYPE_EXTERNAL_COMMAND:nagios_type}: %{NAGIOS_EC_ENABLE_SVC_CHECK:nagios_command}; %{DATA:nagios_hostname};%{DATA:nagios_service}
NAGIOS_EC_LINE_ENABLE_SVC_NOTIFICATIONS	%{NAGIOS_TYPE_EXTERNAL_COMMAND:nagios_type}: %{NAGIOS_EC_ENABLE_SVC_NOTIFICATIONS:nagios_command}; %{DATA:nagios_hostname};%{GREEDYDATA:nagios_service}

Definition Name	Value
NAGIOS_EC_LINE_PROCESS_HOST_CHECK_RESULT	%{NAGIOS_TYPE_EXTERNAL_COMMAND:nagios_type}: %{NAGIOS_EC_PROCESS_HOST_CHECK_RESULT:nagios_command}; %{DATA:nagios_hostname};%{DATA:nagios_state}; %{GREEDYDATA:nagios_check_result}
NAGIOS_EC_LINE_PROCESS_SERVICE_CHECK_RESULT	%{NAGIOS_TYPE_EXTERNAL_COMMAND:nagios_type}: %{NAGIOS_EC_PROCESS_SERVICE_CHECK_RESULT:nagios_command}; %{DATA:nagios_hostname};%{DATA:nagios_service};%{DATA:nagios_state}; %{GREEDYDATA:nagios_check_result}
NAGIOS_EC_LINE_SCHEDULE_HOST_DOWNTIME	%{NAGIOS_TYPE_EXTERNAL_COMMAND:nagios_type}: %{NAGIOS_EC_SCHEDULE_HOST_DOWNTIME:nagios_command}; %{DATA:nagios_hostname};%{NUMBER:nagios_start_time}; %{NUMBER:nagios_end_time};%{NUMBER:nagios_fixed}; %{NUMBER:nagios_trigger_id};%{NUMBER:nagios_duration};%{DATA:author}; %{DATA:comment}
NAGIOS_EC_PROCESS_HOST_CHECK_RESULT	PROCESS_HOST_CHECK_RESULT
NAGIOS_EC_PROCESS_SERVICE_CHECK_RESULT	PROCESS_SERVICE_CHECK_RESULT
NAGIOS_EC_SCHEDULE_HOST_DOWNTIME	SCHEDULE_HOST_DOWNTIME
NAGIOS_EC_SCHEDULE_SERVICE_DOWNTIME	SCHEDULE_SERVICE_DOWNTIME
NAGIOS_HOST_ALERT	%{NAGIOS_TYPE_HOST_ALERT:nagios_type}: %{DATA:nagios_hostname}; %{DATA:nagios_state};%{DATA:nagios_statelevel};%{NUMBER:nagios_attempt}; %{GREEDYDATA:nagios_message}
NAGIOS_HOST_DOWNTIME_ALERT	%{NAGIOS_TYPE_HOST_DOWNTIME_ALERT:nagios_type}: %{DATA:nagios_hostname};%{DATA:nagios_state};%{GREEDYDATA:nagios_comment}
NAGIOS_HOST_EVENT_HANDLER	%{NAGIOS_TYPE_HOST_EVENT_HANDLER:nagios_type}: %{DATA:nagios_hostname};%{DATA:nagios_state};%{DATA:nagios_statelevel}; %{DATA:nagios_event_handler_name}
NAGIOS_HOST_FLAPPING_ALERT	%{NAGIOS_TYPE_HOST_FLAPPING_ALERT:nagios_type}: %{DATA:nagios_hostname};%{DATA:nagios_state};%{GREEDYDATA:nagios_message}
NAGIOS_HOST_NOTIFICATION	%{NAGIOS_TYPE_HOST_NOTIFICATION:nagios_type}: %{DATA:nagios_notifyname}; %{DATA:nagios_hostname};%{DATA:nagios_state};%{DATA:nagios_contact}; %{GREEDYDATA:nagios_message}
NAGIOS_PASSIVE_HOST_CHECK	%{NAGIOS_TYPE_PASSIVE_HOST_CHECK:nagios_type}: %{DATA:nagios_hostname};%{DATA:nagios_state};%{GREEDYDATA:nagios_comment}
NAGIOS_PASSIVE_SERVICE_CHECK	%{NAGIOS_TYPE_PASSIVE_SERVICE_CHECK:nagios_type}: %{DATA:nagios_hostname};%{DATA:nagios_service};%{DATA:nagios_state}; %{GREEDYDATA:nagios_comment}
NAGIOS_SERVICE_ALERT	%{NAGIOS_TYPE_SERVICE_ALERT:nagios_type}: %{DATA:nagios_hostname}; %{DATA:nagios_service};%{DATA:nagios_state};%{DATA:nagios_statelevel}; %{NUMBER:nagios_attempt};%{GREEDYDATA:nagios_message}
NAGIOS_SERVICE_DOWNTIME_ALERT	%{NAGIOS_TYPE_SERVICE_DOWNTIME_ALERT:nagios_type}: %{DATA:nagios_hostname};%{DATA:nagios_service};%{DATA:nagios_state}; %{GREEDYDATA:nagios_comment}
NAGIOS_SERVICE_EVENT_HANDLER	%{NAGIOS_TYPE_SERVICE_EVENT_HANDLER:nagios_type}: %{DATA:nagios_hostname};%{DATA:nagios_service};%{DATA:nagios_state}; %{DATA:nagios_statelevel};%{DATA:nagios_event_handler_name}

Definition Name	Value
NAGIOS_SERVICE_FLAPPING_ALERT	%{NAGIOS_TYPE_SERVICE_FLAPPING_ALERT:nagios_type}: %{DATA:nagios_hostname};%{DATA:nagios_service};%{DATA:nagios_state}; %{GREEDYDATA:nagios_message}
NAGIOS_SERVICE_NOTIFICATION	%{NAGIOS_TYPE_SERVICE_NOTIFICATION:nagios_type}: %{DATA:nagios_notifyname};%{DATA:nagios_hostname};%{DATA:nagios_service}; %{DATA:nagios_state};%{DATA:nagios_contact};%{GREEDYDATA:nagios_message}
NAGIOS_TIMEPERIOD_TRANSITION	%{NAGIOS_TYPE_TIMEPERIOD_TRANSITION:nagios_type}: %{DATA:nagios_service};%{DATA:nagios_unknown1};%{DATA:nagios_unknown2}
NAGIOS_TYPE_CURRENT_HOST_STATE	CURRENT HOST STATE
NAGIOS_TYPE_CURRENT_SERVICE_STATE	CURRENT SERVICE STATE
NAGIOS_TYPE_EXTERNAL_COMMAND	EXTERNAL COMMAND
NAGIOS_TYPE_HOST_ALERT	HOST ALERT
NAGIOS_TYPE_HOST_DOWNTIME_ALERT	HOST DOWNTIME ALERT
NAGIOS_TYPE_HOST_EVENT_HANDLER	HOST EVENT HANDLER
NAGIOS_TYPE_HOST_FLAPPING_ALERT	HOST FLAPPING ALERT
NAGIOS_TYPE_HOST_NOTIFICATION	HOST NOTIFICATION
NAGIOS_TYPE_PASSIVE_HOST_CHECK	PASSIVE HOST CHECK
NAGIOS_TYPE_PASSIVE_SERVICE_CHECK	PASSIVE SERVICE CHECK
NAGIOS_TYPE_SERVICE_ALERT	SERVICE ALERT
NAGIOS_TYPE_SERVICE_DOWNTIME_ALERT	SERVICE DOWNTIME ALERT
NAGIOS_TYPE_SERVICE_EVENT_HANDLER	SERVICE EVENT HANDLER
NAGIOS_TYPE_SERVICE_FLAPPING_ALERT	SERVICE FLAPPING ALERT
NAGIOS_TYPE_SERVICE_NOTIFICATION	SERVICE NOTIFICATION
NAGIOS_TYPE_TIMEPERIOD_TRANSITION	TIMEPERIOD TRANSITION
NAGIOS_WARNING	Warning:%{SPACE}%{GREEDYDATA:nagios_message}

Definition Name	Value
NAGIOSLOGLINE	%{NAGIOSTIME} (?:%{NAGIOS_WARNING}| %{NAGIOS_CURRENT_SERVICE_STATE}|%{NAGIOS_CURRENT_HOST_STATE}| %{NAGIOS_SERVICE_NOTIFICATION}|%{NAGIOS_HOST_NOTIFICATION}| %{NAGIOS_SERVICE_ALERT}|%{NAGIOS_HOST_ALERT}| %{NAGIOS_SERVICE_FLAPPING_ALERT}|%{NAGIOS_HOST_FLAPPING_ALERT}| %{NAGIOS_SERVICE_DOWNTIME_ALERT}| %{NAGIOS_HOST_DOWNTIME_ALERT}|%{NAGIOS_PASSIVE_SERVICE_CHECK}| %{NAGIOS_PASSIVE_HOST_CHECK}|%{NAGIOS_SERVICE_EVENT_HANDLER}| %{NAGIOS_HOST_EVENT_HANDLER}|%{NAGIOS_TIMEPERIOD_TRANSITION}| %{NAGIOS_EC_LINE_DISABLE_SVC_CHECK}| %{NAGIOS_EC_LINE_ENABLE_SVC_CHECK}| %{NAGIOS_EC_LINE_DISABLE_HOST_CHECK}| %{NAGIOS_EC_LINE_ENABLE_HOST_CHECK}| %{NAGIOS_EC_LINE_PROCESS_HOST_CHECK_RESULT}| %{NAGIOS_EC_LINE_PROCESS_SERVICE_CHECK_RESULT}| %{NAGIOS_EC_LINE_SCHEDULE_HOST_DOWNTIME}| %{NAGIOS_EC_LINE_DISABLE_HOST_SVC_NOTIFICATIONS}| %{NAGIOS_EC_LINE_ENABLE_HOST_SVC_NOTIFICATIONS}| %{NAGIOS_EC_LINE_DISABLE_HOST_NOTIFICATIONS}| %{NAGIOS_EC_LINE_ENABLE_HOST_NOTIFICATIONS}| %{NAGIOS_EC_LINE_DISABLE_SVC_NOTIFICATIONS}| %{NAGIOS_EC_LINE_ENABLE_SVC_NOTIFICATIONS})
NAGIOSTIME	\\[%{NUMBER:nagios_epoch}\\]
NETSCREENSESSIONLOG	%{SYSLOGTIMESTAMP:date} %{IPORHOST:device} %{IPORHOST}: NetScreen device_id=%{WORD:device_id}%{DATA}: start_time=%{QUOTEDSTRING:start_time} duration=%{INT:duration} policy_id=%{INT:policy_id} service=%{DATA:service} proto= %{INT:proto} src zone=%{WORD:src_zone} dst zone=%{WORD:dst_zone} action= %{WORD:action} sent=%{INT:sent} rcvd=%{INT:rcvd} src=%{IPORHOST:src_ip} dst=%{IPORHOST:dst_ip} src_port=%{INT:src_port} dst_port=%{INT:dst_port} src-xlated ip=%{IPORHOST:src_xlated_ip} port=%{INT:src_xlated_port} dst-xlated ip= %{IPORHOST:dst_xlated_ip} port=%{INT:dst_xlated_port} session_id=%{INT:session_id} reason=%{GREEDYDATA:reason}
NONNEGINT	\\b(?:[0-9]+)\\b
NOTSPACE	\\S+
NUMBER	(?:%{BASE10NUM})
PATH	(?:%{UNIXPATH}|%{WINPATH})
POSINT	\\b(?:[1-9][0-9]*)\\b
POSTGRESQL	%{DATESTAMP:timestamp} %{TZ} %{DATA:user_id} %{GREEDYDATA:connection_id} %{POSINT:pid}
PROG	[\\x21-\\x5a\\x5c\\x5e-\\x7e]+
QS	%{QUOTEDSTRING}
QUOTEDSTRING	(?>(?<!\\\\)(?>\"(?>\\\\.|[^\\\\\"]+)+\"|\"\"|(?>'(?>\\\\.|[^\\\\']+)+')|"|(?>`(?>\\\\.|[^\\\\`]+)+`)|``))
RAILS3	%{RAILS3HEAD}(?:%{RPROCESSING})?(?<context>(?:%{DATA}\\n)*)(?:%{RAILS3FOOT})?
RAILS3FOOT	Completed %{NUMBER:response}%{DATA} in %{NUMBER:totalms}ms %{RAILS3PROFILE}%{GREEDYDATA}

Definition Name	Value
RAILS3HEAD	(?m)Started %{WORD:verb} \"%{URIPATHPARAM:request}\" for %{IPORHOST:clientip} at (?<timestamp>%{YEAR}-%{MONTHNUM}-%{MONTHDAY} %{HOUR}:%{MINUTE}: %{SECOND} %{ISO8601_TIMEZONE})
RAILS3PROFILE	(?:\\(Views: %{NUMBER:viewms}ms \\\| ActiveRecord: %{NUMBER:activerecordms}ms|\ \(ActiveRecord: %{NUMBER:activerecordms}ms)?
RCONTROLLER	(?<controller>[^#]+)#(?<action>\\w+)
REDISLOG	\\[%{POSINT:pid}\\] %{REDISTIMESTAMP:timestamp} *
REDISMONLOG	%{NUMBER:timestamp} \\[%{INT:database} %{IP:client}:%{NUMBER:port}\\] \"%{WORD:command}\"\\s?%{GREEDYDATA:params}
REDISTIMESTAMP	%{MONTHDAY} %{MONTH} %{TIME}
RPROCESSING	\\W*Processing by %{RCONTROLLER} as (?<format>\\S+)(?:\\W*Parameters: {%{DATA:params}}\\W*)?
RT_FLOW_EVENT	(RT_FLOW_SESSION_CREATE|RT_FLOW_SESSION_CLOSE|RT_FLOW_SESSION_DENY)
RT_FLOW1	%{RT_FLOW_EVENT:event}: %{GREEDYDATA:close-reason}: %{IP:src-ip}/%{INT:src-port}->%{IP:dst-ip}/%{INT:dst-port} %{DATA:service} %{IP:nat-src-ip}/%{INT:nat-src-port}->%{IP:nat-dst-ip}/%{INT:nat-dst-port} %{DATA:src-nat-rule-name} %{DATA:dst-nat-rule-name} %{INT:protocol-id} %{DATA:policy-name} %{DATA:from-zone} %{DATA:to-zone} %{INT:session-id} \\d+\\(%{DATA:sent}\\) \\d+\\(%{DATA:received}\\) %{INT:elapsed-time} .*
RT_FLOW2	%{RT_FLOW_EVENT:event}: session created %{IP:src-ip}/%{INT:src-port}->%{IP:dst-ip}/%{INT:dst-port} %{DATA:service} %{IP:nat-src-ip}/%{INT:nat-src-port}->%{IP:nat-dst-ip}/%{INT:nat-dst-port} %{DATA:src-nat-rule-name} %{DATA:dst-nat-rule-name} %{INT:protocol-id} %{DATA:policy-name} %{DATA:from-zone} %{DATA:to-zone} %{INT:session-id} .*
RT_FLOW3	%{RT_FLOW_EVENT:event}: session denied %{IP:src-ip}/%{INT:src-port}->%{IP:dst-ip}/%{INT:dst-port} %{DATA:service} %{INT:protocol-id}\\(\\d\\) %{DATA:policy-name} %{DATA:from-zone} %{DATA:to-zone} .*
RUBY_LOGGER	[DFEWI], \\[%{TIMESTAMP_ISO8601:timestamp} #%{POSINT:pid}\\] * %{RUBY_LOGLEVEL:loglevel} -- +%{DATA:progname}: %{GREEDYDATA:message}
RUBY_LOGLEVEL	(?:DEBUG|FATAL|ERROR|WARN|INFO)
RUUID	\\h{32}
S3_ACCESS_LOG	%{WORD:owner} %{NOTSPACE:bucket} \\[%{HTTPDATE:timestamp}\\] %{IP:clientip} %{NOTSPACE:requester} %{NOTSPACE:request_id} %{NOTSPACE:operation} %{NOTSPACE:key} (?:\"%{S3_REQUEST_LINE}\"|-) (?:%{INT:response:int}|-) (?:-| %{NOTSPACE:error_code}) (?:%{INT:bytes:int}|-) (?:%{INT:object_size:int}|-) (?: %{INT:request_time_ms:int}|-) (?:%{INT:turnaround_time_ms:int}|-) (?:%{QS:referrer}|-) (?:\"?%{QS:agent}\"?|-) (?:-|%{NOTSPACE:version_id})
S3_REQUEST_LINE	(?:%{WORD:verb} %{NOTSPACE:request}(?: HTTP/%{NUMBER:httpversion})?| %{DATA:rawrequest})
SECOND	(?:(?:[0-5]?[0-9]|60)(?:[:.,][0-9]+)?)

Definition Name	Value
SFW2	((%{SYSLOGTIMESTAMP})|(%{TIMESTAMP_ISO8601}))\\s*%{HOSTNAME}\\s*kernel\\S+\\s*%{NAGIOSTIME}\\s*SFW2\\-INext\\-%{NOTSPACE:nf_action}\\s*IN=%{USERNAME:nf_in_interface}.*OUT=((\\s*%{USERNAME:nf_out_interface})|(\\s*))MAC=((%{COMMONMAC:nf_dst_mac}:%{COMMONMAC:nf_src_mac})|(\\s*)).*SRC=%{IP:nf_src_ip}\\s*DST=%{IP:nf_dst_ip}.*PROTO=%{WORD:nf_protocol}((.*SPT=%{INT:nf_src_port}.*DPT=%{INT:nf_dst_port}.*)|())
SHOREWALL	(%{SYSLOGTIMESTAMP:timestamp}) (%{WORD:nf_host}) kernel:.*Shorewall:(%{WORD:nf_action1})?: (%{WORD:nf_action2})?.*IN=(%{USERNAME:nf_in_interface})?.*(OUT= *MAC=(%{COMMONMAC:nf_dst_mac}):(%{COMMONMAC:nf_src_mac})?|OUT= %{USERNAME:nf_out_interface}).*SRC=(%{IPV4:nf_src_ip}).*DST=(%{IPV4:nf_dst_ip}).*LEN=(%{ *TOS=(%{WORD:nf_tos}).?*PREC=(%{WORD:nf_prec}).? *TTL=(%{INT:nf_ttl}).?*ID=(%{INT:nf_id}).?*PROTO=(%{WORD:nf_protocol}).? *SPT=(%{INT:nf_src_port}?.*DPT=%{INT:nf_dst_port}?.*)
SPACE	\\s*
SQUID3	%{NUMBER:timestamp}\\s+%{NUMBER:duration}\\s%{IP:client_address}\\s%{WORD:cache_result}/%{POSINT:status_code}\\s%{NUMBER:bytes}\\s%{WORD:request_method}\\s%{NOTSPACE:url}\\s(%{NOTSPACE:user}|-)\\s%{WORD:hierarchy_code}/%{IPORHOST:server}\\s%{NOTSPACE:content_type}
SYSLOG5424BASE	%{SYSLOG5424PRI}%{NONNEGINT:syslog5424_ver} +(?: %{TIMESTAMP_ISO8601:syslog5424_ts}|-) +(?:%{IPORHOST:syslog5424_host}|-) +(-|%{SYSLOG5424PRINTASCII:syslog5424_app}) +(-| %{SYSLOG5424PRINTASCII:syslog5424_proc}) +(-| %{SYSLOG5424PRINTASCII:syslog5424_msgid}) +(?: %{SYSLOG5424SD:syslog5424_sd}|-|)
SYSLOG5424LINE	%{SYSLOG5424BASE} +%{GREEDYDATA:syslog5424_msg}
SYSLOG5424PRI	<%{NONNEGINT:syslog5424_pri}>
SYSLOG5424PRINTASCII	[!-~]+
SYSLOG5424SD	\\[%{DATA}\\]+
SYSLOGBASE	%{SYSLOGTIMESTAMP:timestamp} (?:%{SYSLOGFACILITY})? %{SYSLOGHOST:logsource} %{SYSLOGPROG}:
SYSLOGBASE2	(?:%{SYSLOGTIMESTAMP:timestamp}|%{TIMESTAMP_ISO8601:timestamp8601}) (?: %{SYSLOGFACILITY})?%{SYSLOGHOST:logsource}+(?: %{SYSLOGPROG}:|)
SYSLOGFACILITY	<%{NONNEGINT:facility}.%{NONNEGINT:priority}>
SYSLOGHOST	%{IPORHOST}
SYSLOGLINE	%{SYSLOGBASE2} %{GREEDYDATA:message}
SYSLOGPAMSESSION	%{SYSLOGBASE} (?=%{GREEDYDATA:message})%{WORD:pam_module}\\(%{DATA:pam_caller}\\): session %{WORD:pam_session_state} for user %{USERNAME:username}(?: by %{GREEDYDATA:pam_by})?
SYSLOGPROG	%{PROG:program}(?:\\[%{POSINT:pid}\\])?
SYSLOGTIMESTAMP	%{MONTH} +%{MONTHDAY} %{TIME}
TIME	(?!<[0-9])%{HOUR}:%{MINUTE}(?::%{SECOND})(?![0-9])
TIMESTAMP_ISO8601	%{YEAR}-%{MONTHNUM}-%{MONTHDAY}[T]%{ISO8601_HOUR}:?%{MINUTE}(?::?%{SECOND})?%{ISO8601_TIMEZONE}?

Definition Name	Value
TOMCAT_DATESTAMP	20%{YEAR}-%{MONTHNUM}-%{MONTHDAY} %{HOUR}:?%{MINUTE}(?::?%{SECOND}) %{ISO8601_TIMEZONE}
TOMCATLOG	%{TOMCAT_DATESTAMP:timestamp} \\| %{LOGLEVEL:level} \\| %{JAVACLASS:class} - %{JAVALOGMESSAGE:logmessage}
TTY	(?:/dev/(pts|tty([pq])?)(\\w+)?/?(?:[0-9]+))
TZ	(?:[APMCE][SD]T|UTC)
UNIXPATH	(/([\\w_%!$@:.,+~-]+|\\\\.)*)+
URI	%{URIPROTO}://(?:%{USER}(?::[^@]*)?@)?(?:%{URIHOST})?(?:%{URIPATHPARAM})?
URIHOST	%{IPORHOST}(?::%{POSINT:port})?
URIPARAM	\\?[A-Za-z0-9$.+!*'|(){},~@#%&/=:;_?\\-\\[\\]<>]*
URIPATH	(?:/[A-Za-z0-9$.+!*'(){},~:;=@#%&_\\-]*)+
URIPATHPARAM	%{URIPATH}(?:%{URIPARAM})?
URIPROTO	[A-Za-z]([A-Za-z0-9+\\-.]+)+
URN	urn:[0-9A-Za-z][0-9A-Za-z-]{0,31}:(?:%[0-9a-fA-F]{2}|[0-9A-Za-z()+,.:=@;$_!*'/?#-])+
USER	%{USERNAME}
USERNAME	[a-zA-Z0-9._-]+
UUID	[A-Fa-f0-9]{8}-(?:[A-Fa-f0-9]{4}-){3}[A-Fa-f0-9]{12}
WINDOWSMAC	(?:(?:[A-Fa-f0-9]{2}-){5}[A-Fa-f0-9]{2})
WINPATH	(?>[A-Za-z]+:|\\\\)(?:\\\\[^\\\\?*]*)+
WORD	\\b\\w+\\b
YEAR	(?>\\d\\d){1,2}

Using DX OI - Logs

Information and procedures to use DX OI - Logs.

This section provides the following information:

- Authentication and Authorization
- Access DX OI - Logs
- DX OI - Logs User Interface
- Log Alarm Configuration
- Log Ingestion Throttling
- DX OI - Logs APIs
- Troubleshoot DX OI - Logs

Authentication and Authorization

The following table lists the privileges for Log Analytics:

Role	Write Access	Read Access
Tenant Administrator	Yes	Yes
Power User	Yes	Yes
User	No	Yes
Custom Role User	No	Yes

These permissions for the roles cannot be changed or configured explicitly in Log Analytics. A Tenant Administrator can add the users to the tenant and provide them access through **User Management** on the **Settings** page. For more information, see the User Management section.

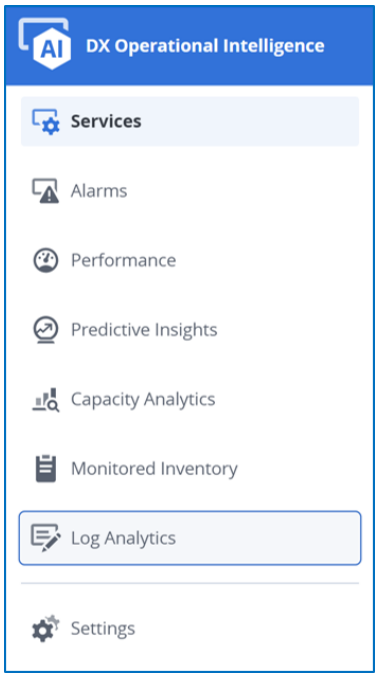
> **NOTE**
> Custom roles are not supported. However, a user with a custom role has the privileges of the role **User**.

Access DX OI - Logs

You can access DX OI - Logs from the left navigation pane of DX Operational Intelligence.

Follow these steps:

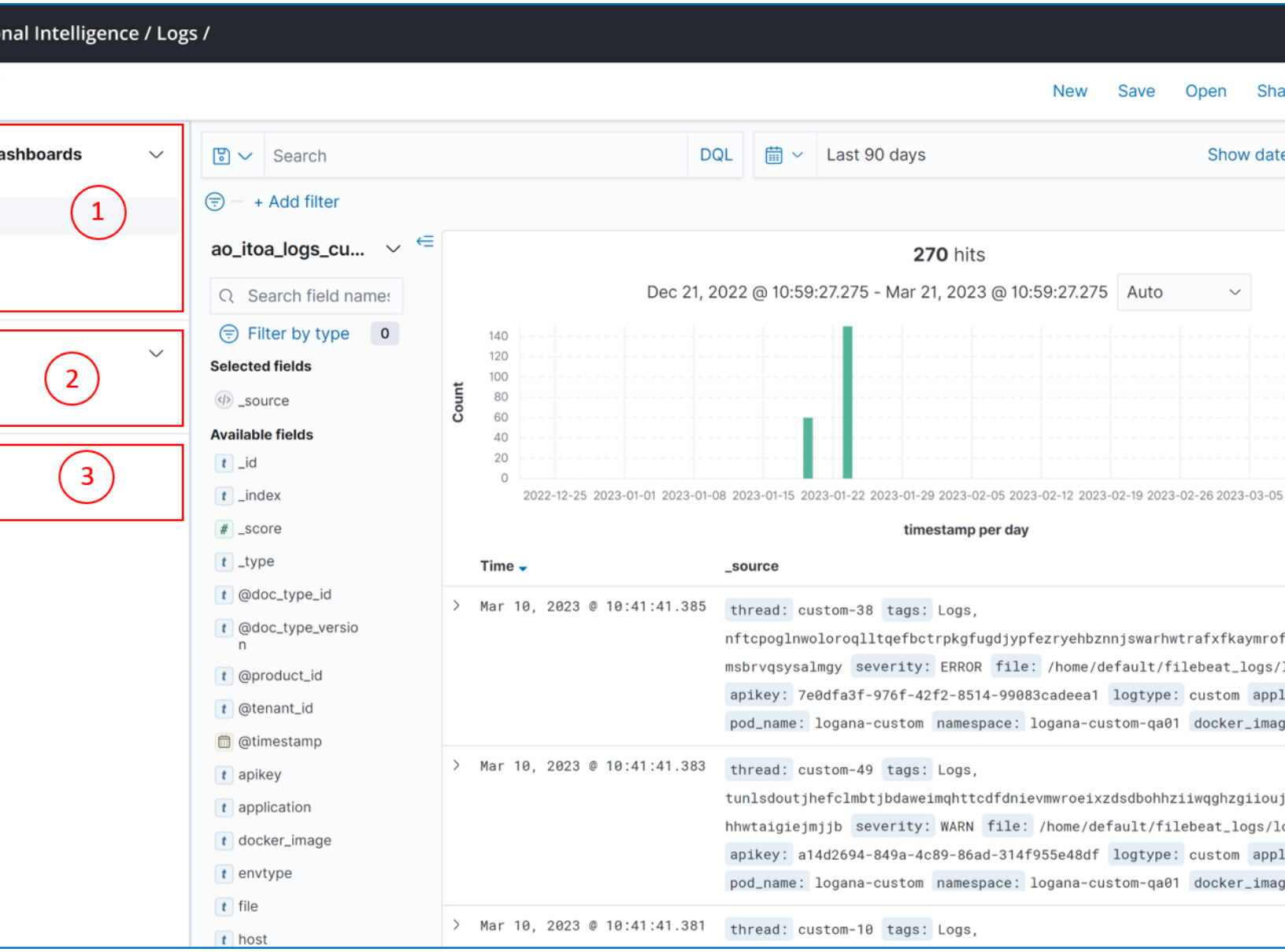
1. Log into DX Operational Intelligence.
2. Click Log Analytics in the left navigation pane as shown:



The **DX Operational Intelligence / Logs** page is launched in a new tab.

DX OI - Logs User Interface

The **DX Operational Intelligence / Logs** page is divided into the following sections:



- **OpenSearch Dashboards (1):** This section provides quick links to pages where you can search and filter data using queries, view the OOTB dashboards for Log Analytics, also create custom dashboards.
- **Management (2):** This section provides the link to stack management where you can manage your dashboards and saved objects.
- **Undock Navigation (3):** This section provides the link to enable and disable docking of the navigation pane.

Dashboards

This section provides links to the following capabilities:

- **Discover:** On this page, you can search and filter data using queries.
- **Dashboard:** On this page, you can view the out-of-the-box dashboards for Log Analytics. You can also create a custom dashboard.
- **Visualize:** On this page, you can visualize data.

Discover

The Discover option in Log Dashboards allows you to interactively explore the log data by querying and filtering the raw log documents. You can search, create filter criteria, and display the log documents in a tabular format. You can create searches and fetch the precise data that you want to analyze and gain insights.

While working with searches using the Discover option, you can perform the following tasks:

- Select, explore, and present large amounts of data in a visualization.
- Customize and save your discover search for the future reference.
- Use the saved searches in dashboards and visualization

You can perform the following actions while creating a discovery search filter:

- Select Log Index Pattern and Fields
- Set Time Filters
- Add Filters to fields
- Filter by Any Value
- Open
- Save Discover Search
- Share Discover Search
- Inspect Search
- Visualize Fields

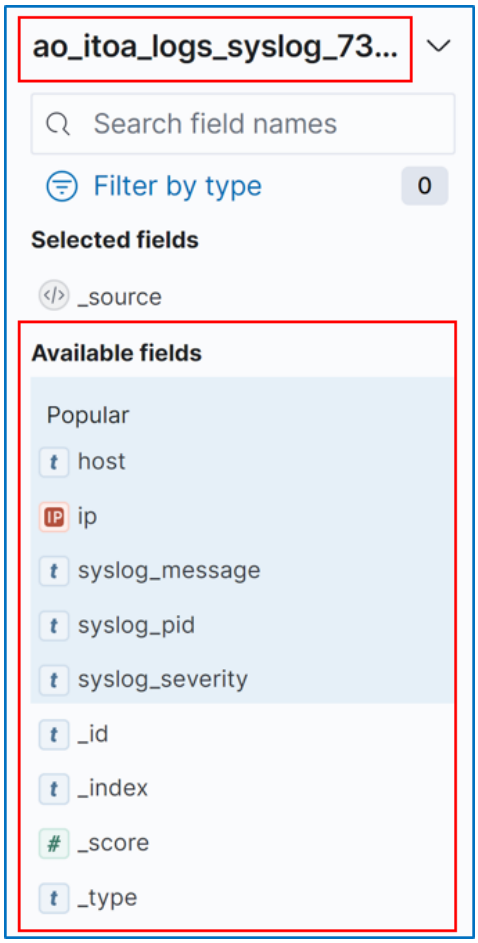
Select Log Pattern and Fields

Searches on the Index patterns point to a specific log data (OpenSearch index) that you want to access, analyze, and gain insights. For example, an index pattern can point to log data generated from the Apache access logs.

After you select the index pattern, you can view the available fields in the selected index pattern.

Follow these steps:

1. Select the log index pattern from the drop-down on which you want to work.
   The available fields of the selected index are displayed in the **Available Fields** section.



2. Hover over the field and click **+**.

   The application adds the selected fields to the **Selected Fields** section. Also, the data for the selected field is displayed in the Discover search table of documents.

Set Time Filters

Time Filters on dashboards, visualizations, and searches enable you to restrict and display the data specific to a selected time range. If the selected time range fetches data, the search in the Discover option displays a histogram at the top of the table of log documents. The histogram displays the distribution of events over time as per the specified filter criteria.

By default, the time range is set to last 15 Minutes. The search provides multiple options to filter based on the time range.

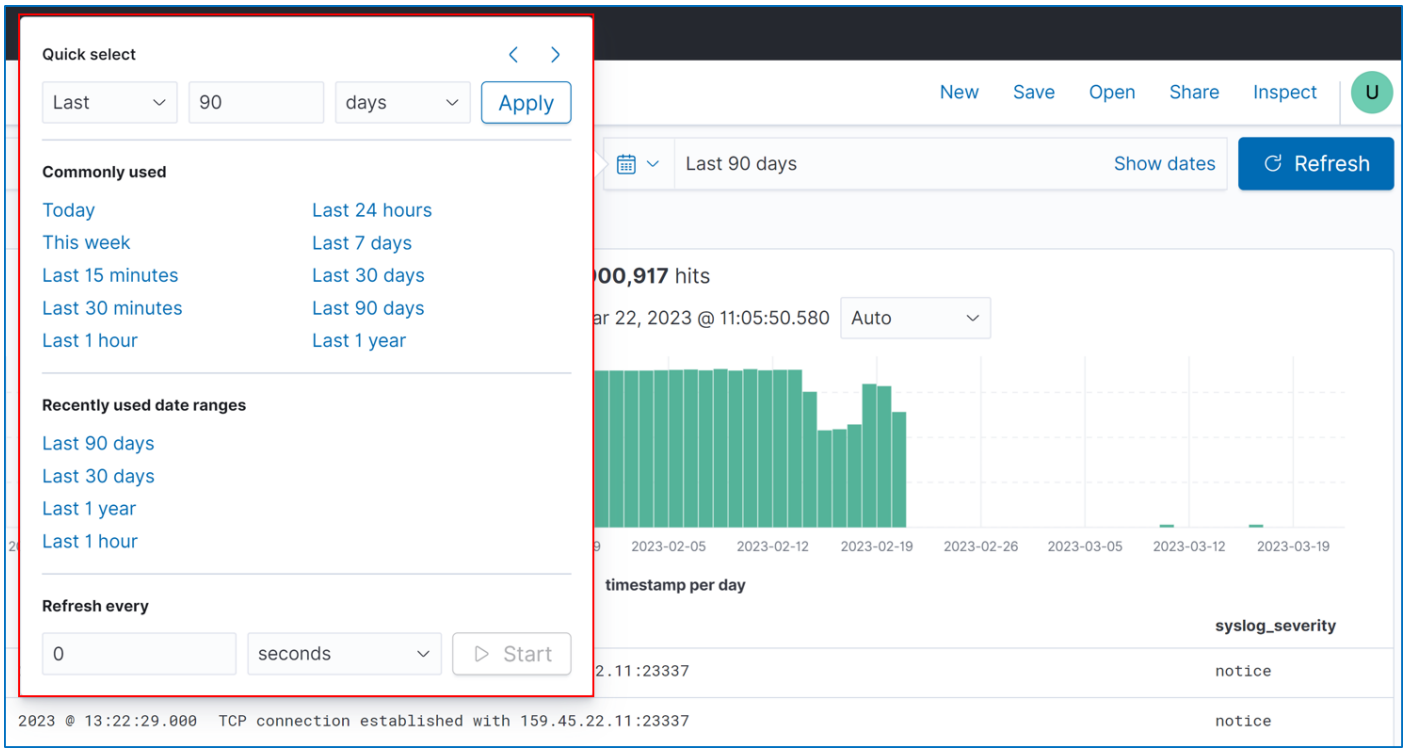
- You can define a time filter for the last 15 Minutes to the last 15 Months or Years and so on.
- You can also set the time for the discover search to refresh periodically.

To set a time filter, follow these steps:

1. Click the calendar icon

   ( ) .

   The Time Filter pop-up appears. You can see the commonly used and the recently used date ranges in this window.



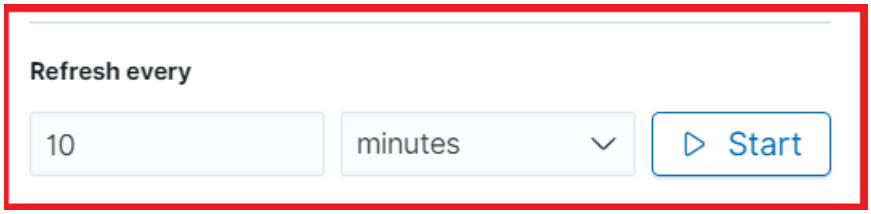
2. Complete the following actions in the **Quick Select** section:
   a) Select relative time range (Last or Next) from the drop-down.
   - The **Last** option allows you to step back in time, relative to now.
   - The **Next** option allows you to step forward in time, relative to now.
3. Select the time and the units in the respective fields.
   For example, if you select the time range as Last, 15, and Minutes in the respective fields, the search fetches the data from now till the last 15 minutes.
4. Use < > to view previous and next-time windows.
5. Click **Apply**.
   The application applies the selected time filter.

Auto Refresh Discover Search Visualization

Search on the Discover page does not automatically refresh. If you want to periodically refresh a discover search, you can specify the refresh interval in the Time filter pop-up.

Follow these steps:

1. In the **Refresh every** section of the Time Filters pop-up, mention the refresh time interval.



2. Select the time units from the drop-down.
3. Click **Start.**

The application refreshes the search and the log documents table as per the defined refresh interval.

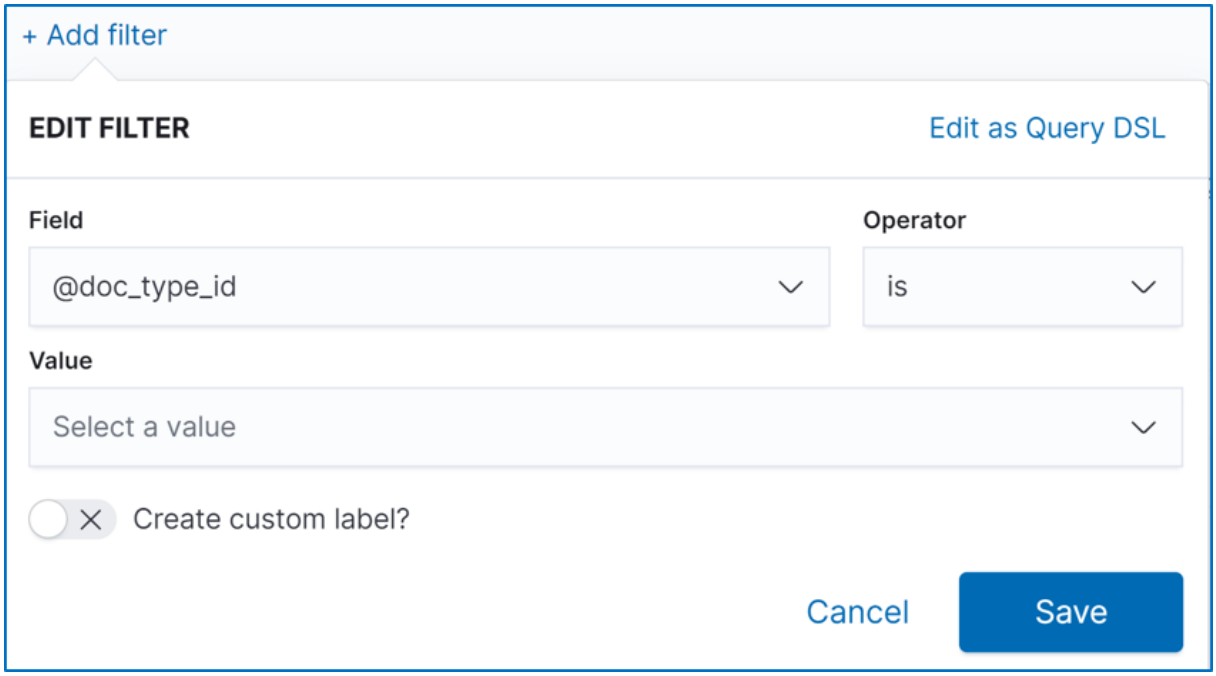
For example, if you specify the refresh option as 10 minutes, the Discover Search refreshes every 10 minutes.

Add Filters to Fields

Filters on Discover Searches enable you to get different combinations and views of your data. You can use a combination of fields, operators, and values to define one or more filters. The search displays only those log documents that meet the defined filter criteria for a field.

Follow these steps:

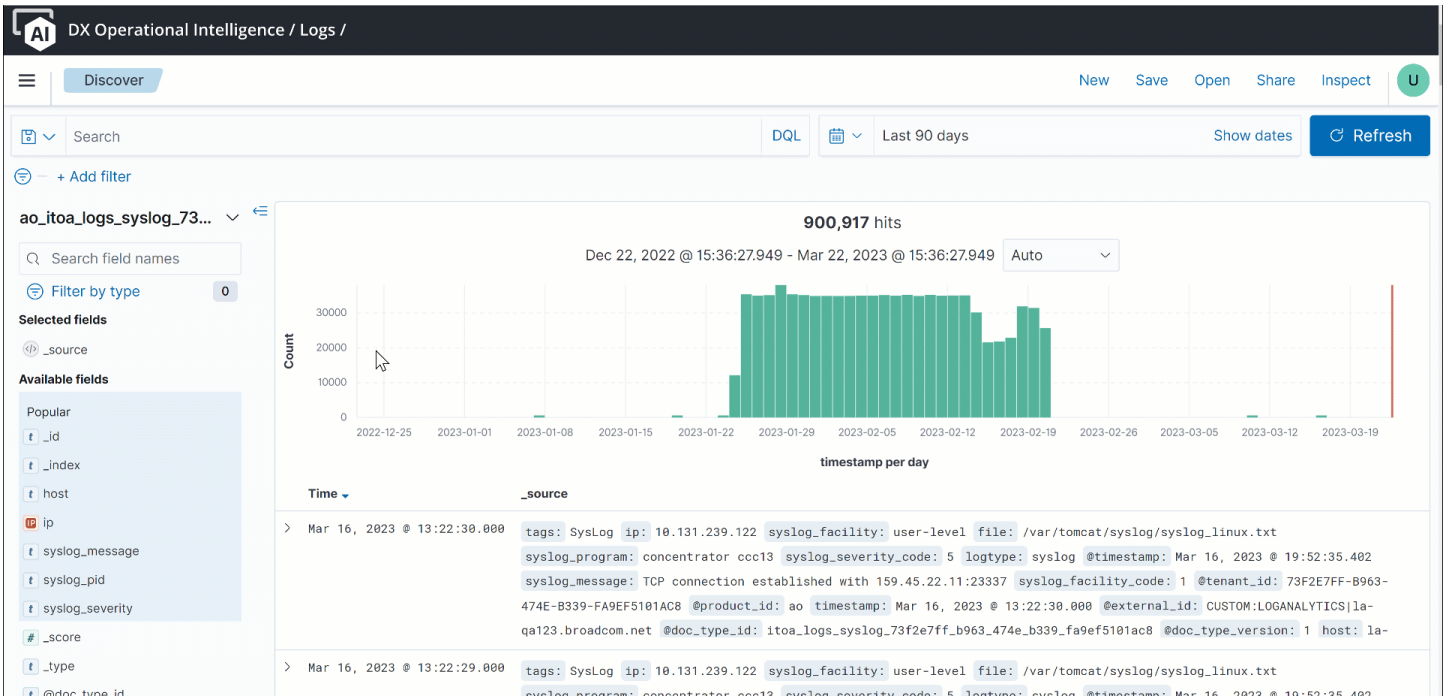
1. Click **+ Add filter**.



2. Select the field, operator, and the value combination in the respective fields to filter the data.
3. (Optional) Enable **Create Custom Label** and provide a name to your filter.
4. Click **Save** to save the filter.

Add Filters to Available Fields

Alternatively, you can also apply filters in the **Available Field** section.



Follow these steps:

1. Click the field in the Available Fields section.
   The field displays the top 5 values that are associated with the field.
2. Hover over the value on which you want to add filter criteria and perform one of the following actions:
   - Click **+** to show only those documents that have the selected value.
   - Click **-** to exclude the documents that have the selected value.

The filter on the specific field and the value is added.

Filter Operators

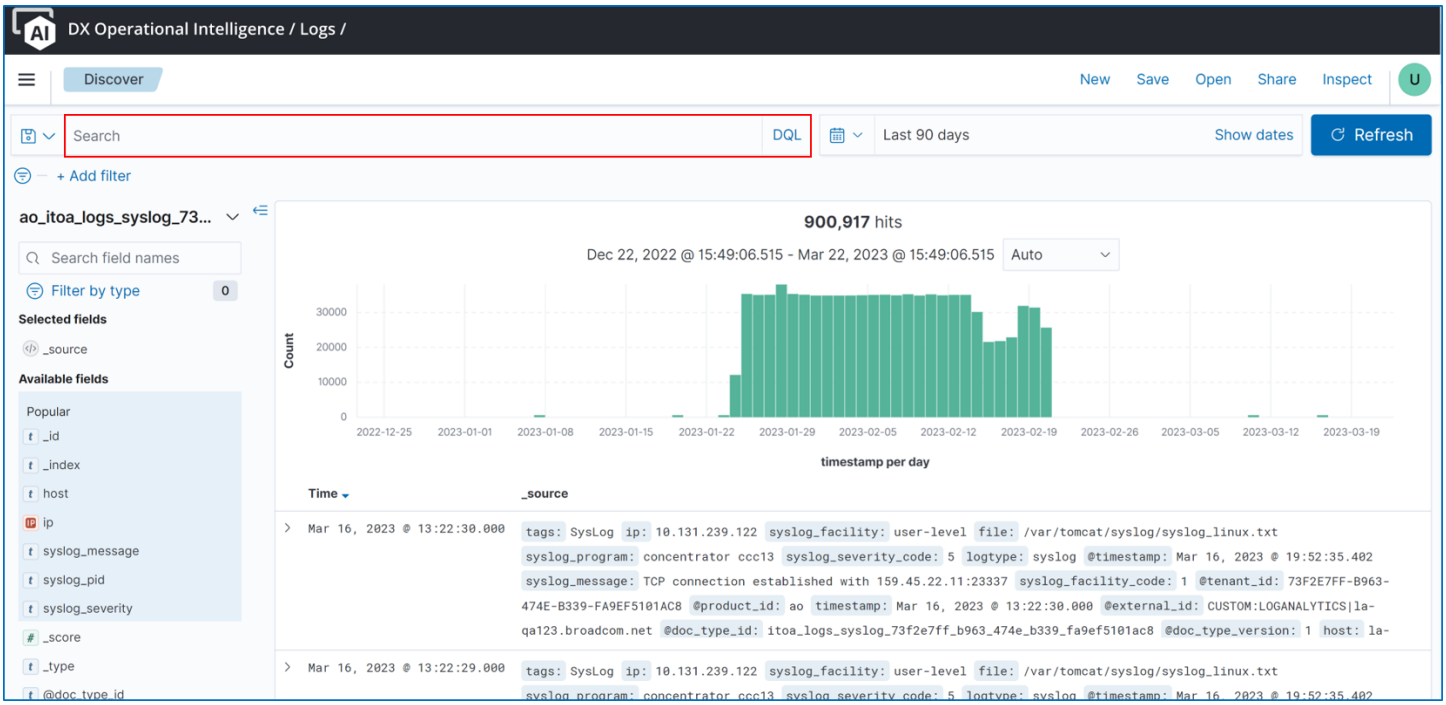
Using Filter operators, you can create both positive and negative filters. Positive filters include the data in the search results, while the negative filters exclude the data from the search results.

- Is – Shows results that contain the specified value in the selected field.
- Is not – Shows results that does not contain the specified value in the selected field.
- Is one of Shows results that match with one of the specified values for the selected field.
- Is not one of – Shows results that do not match with the specified values for the selected field.
- Exists – Shows results where the selected field is not blank.
- Does not exist – Shows results where the selected field is blank.

Filter by Any Value

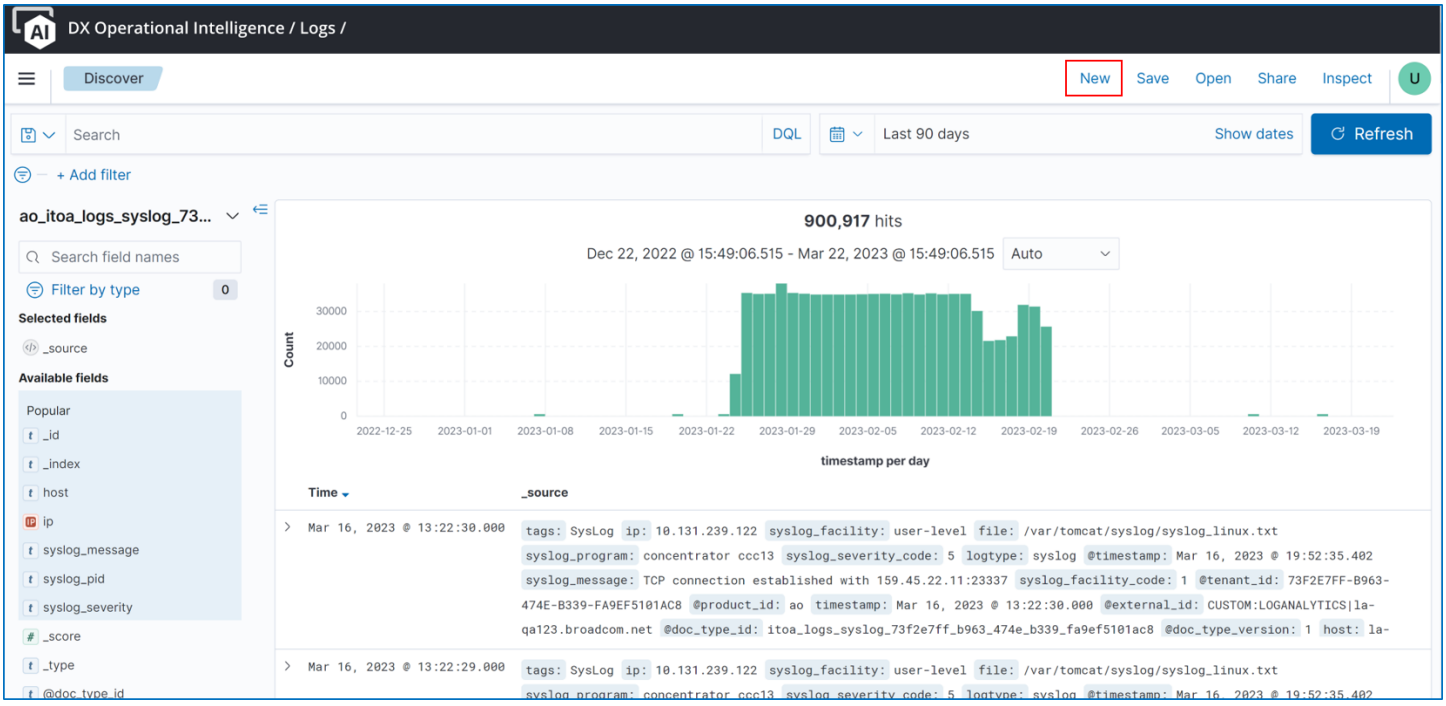
The Discover Search provides a universal search feature that enables you to filter the table of documents by any value or text. Type any text or value in the Search field, the application fetches the table of documents that has the selected text or value.

- To perform such universal search, by default the Discover Search uses the Lucene Search Engine.
- To build more complex queries, select OpenSearch Dashboards Query Language (DQL).



Reset Search Filters

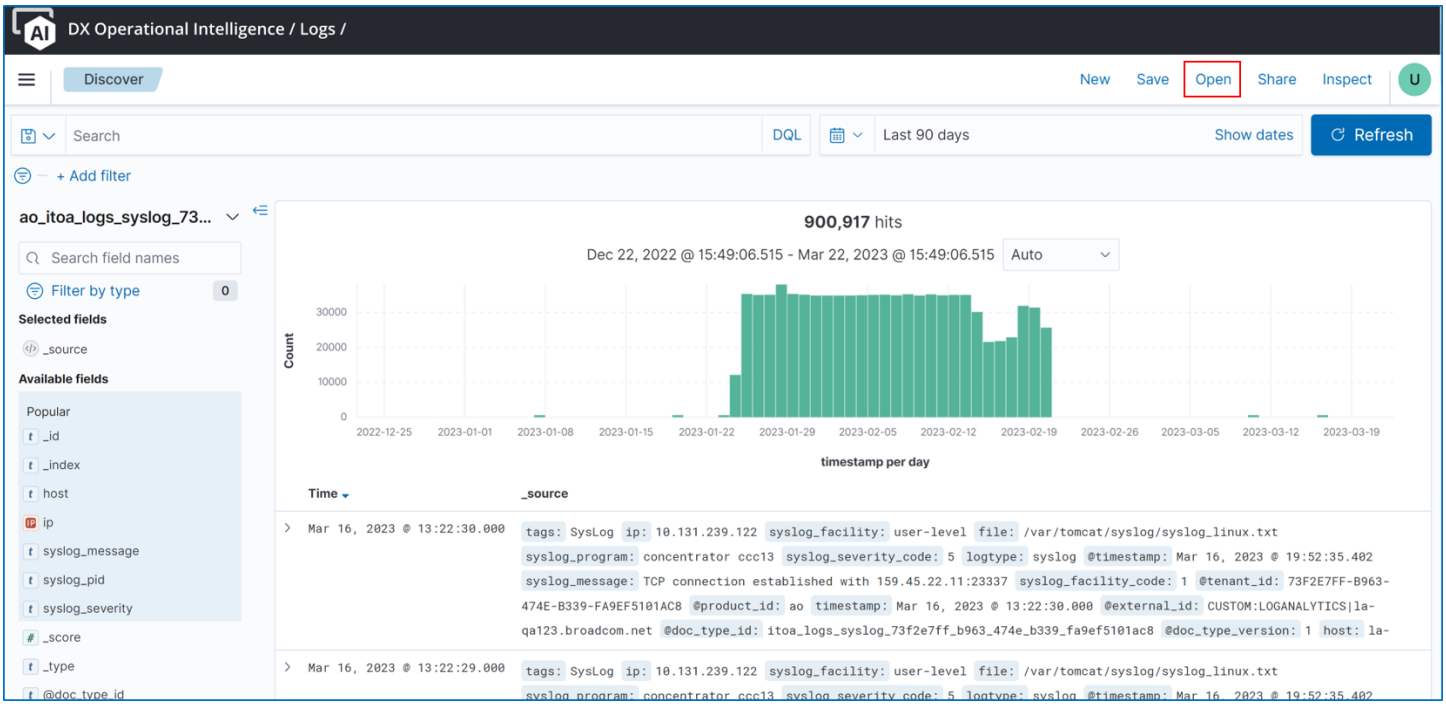
Using the New option, you can reset the Search. On clicking the new, you can remove the filters and selected fields from a search.



Open Saved Search

You can open the saved searches, visualizations, or the dashboards using the Open option.

To open any saved object, click **Open**.

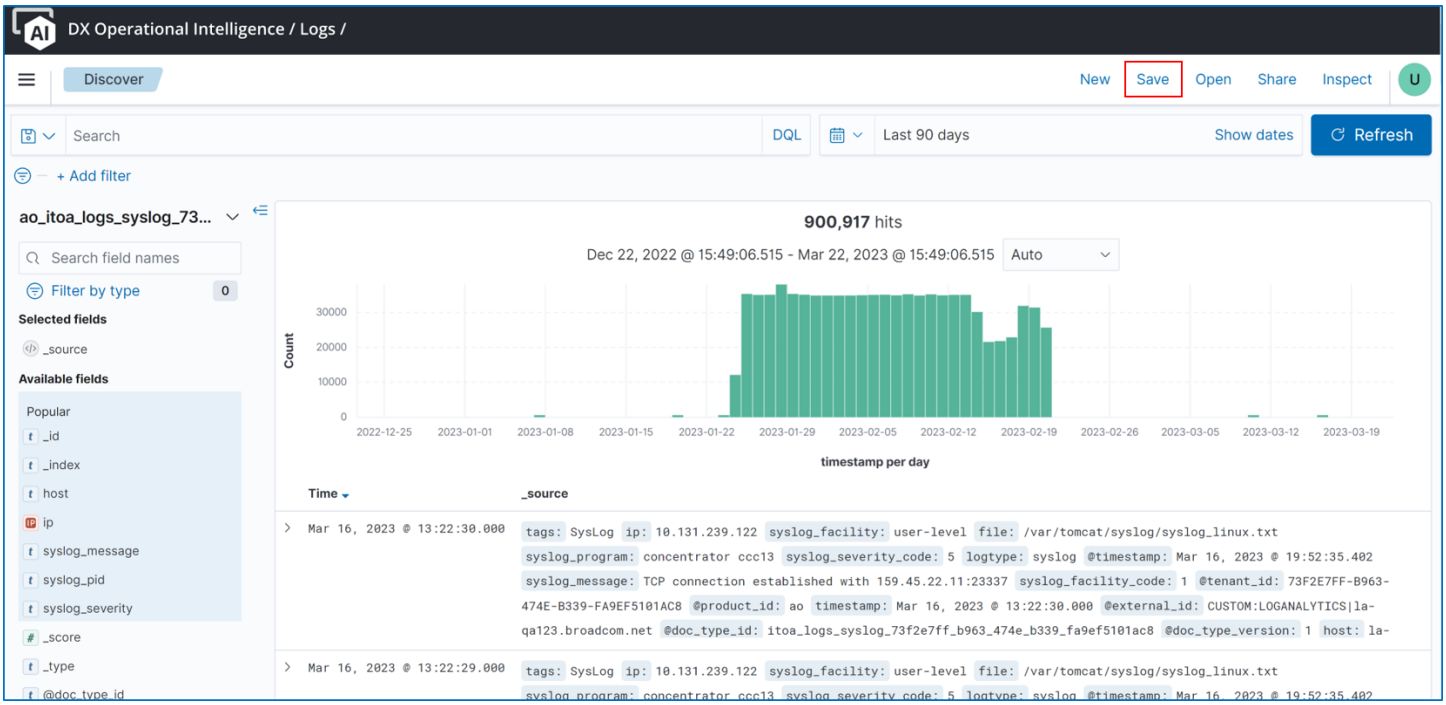


The Open Searches pop-up displays both pre-packaged and custom saved searches. You can select the saved search that you wanted to access from the Open Searches Pop-up.

Save the Discover Search

You can save your searches for the future reference and use. Also, you can access the saved searches in Dashboard and Visualization pages.

To save a search, click **Save** and provide a title for the saved search.

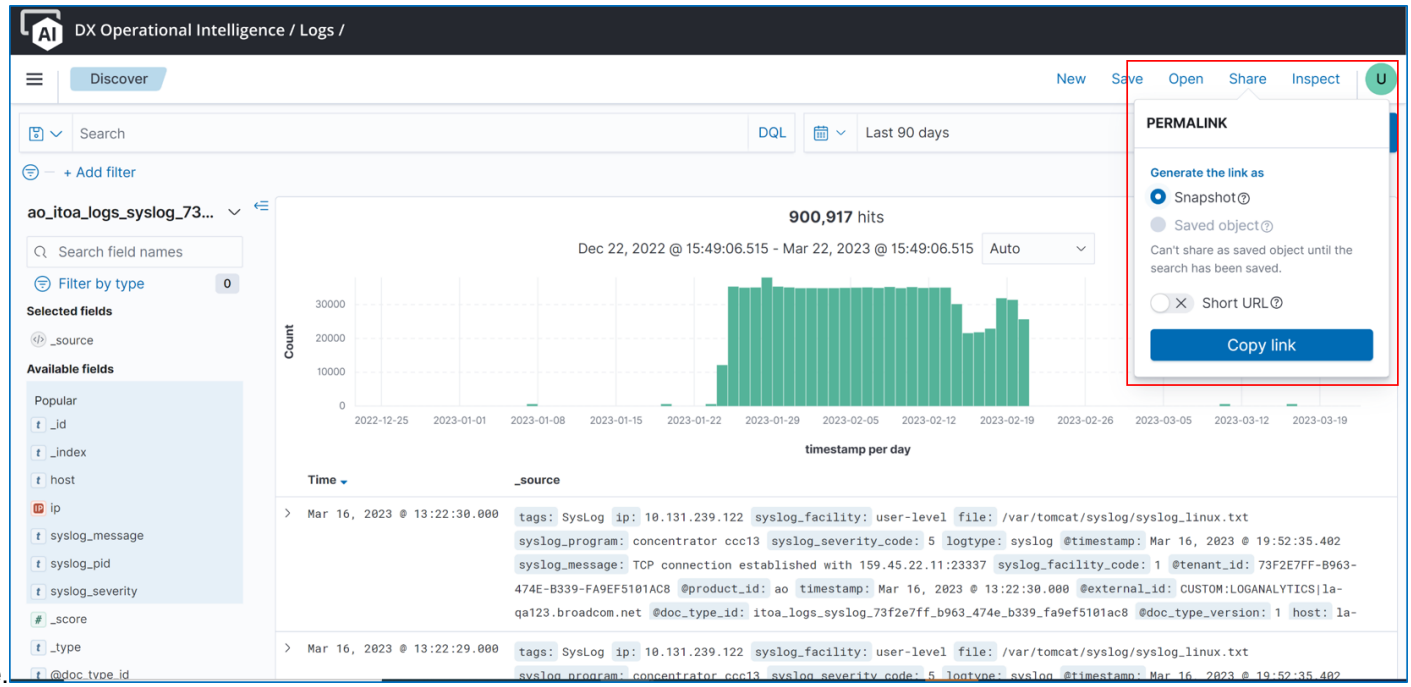


Share the Discover Search

You can share your findings from the search, with the concerned personnel.

To share the search results, follow these steps:

1. Click



Share.

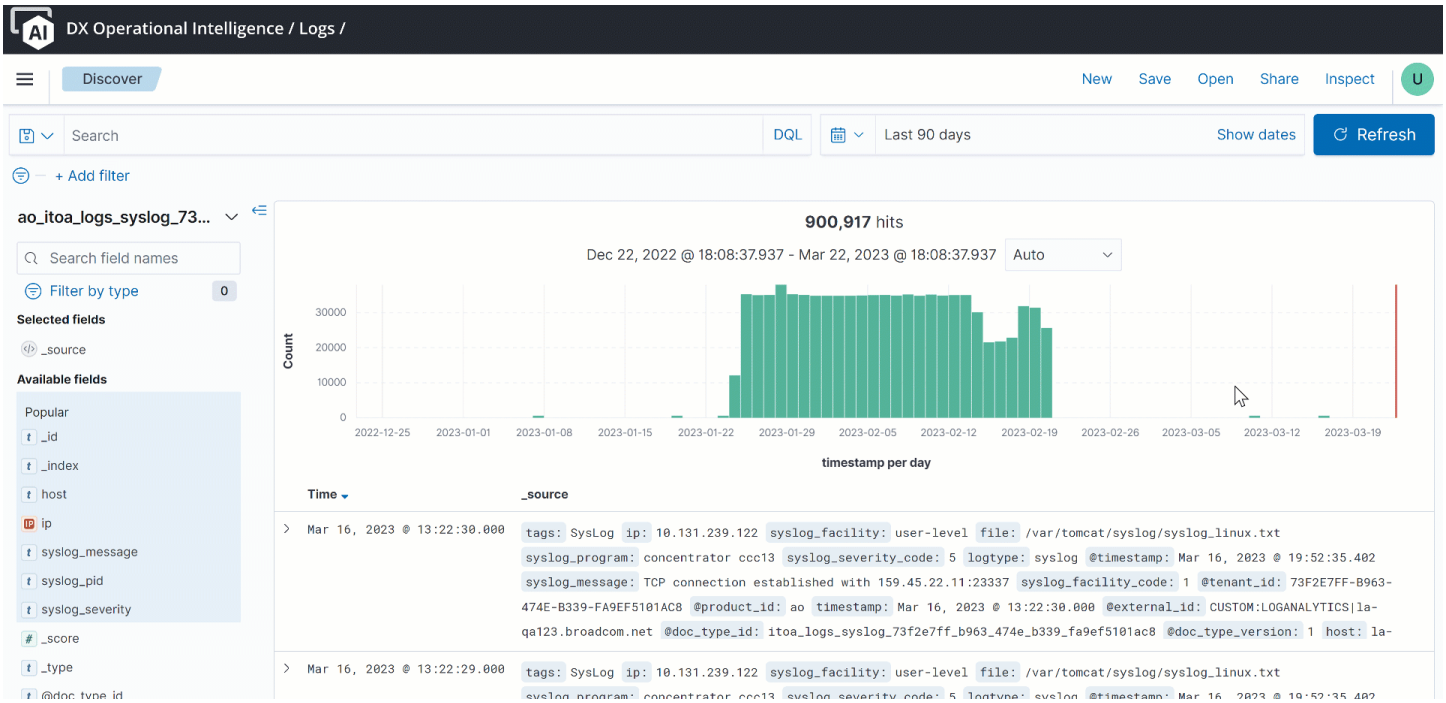
2. In the PERMALINK drop-down that appears, select one of the following options:
   - Click the Snapshot and copy the link to share data available in the search currently. Enable the Short URL toggle and then copy the link to share the short link for the snapshot option.
   - Click the Saved object option and copy to share the recent data available in your search.
3. Share the copied link with others using communication tools.

> **NOTE**
> The users with whom the link is shared can access the search link only the users have the access to DX Operational Intelligence.

Inspect Search

The Inspect option enables you to review the actions that the search performs to get the desired results. You can quickly get a snapshot of the various attributes that the search uses to filter the data.

Based on these statistics, you can modify the query to get the required results. To inspect the data, click **Inspect**. The application displays the View Requests pop-up with the following tabs:



- **Statistics**
  This tab displays the statistics as per the selected index and the defined search criteria. You can view the following information:
  - Hits: Number of query documents.
  - Hits (total)
  - Index pattern: Index pattern that is connected to the OpenSearch indices.
  - Index pattern ID: ID of the index pattern.
  - Query time: Time taken to process the query.
  - Request timestamp
- **Request**
  This tab displays the auto-generated query DSL (Domain-Specific Language) based on the selected index and the defined search criteria in the search. The query DSL contains the code that the application uses to fetch the log documents for the defined search.

  > **NOTE**
  > DX Operational Intelligence supports monitoring and auto-generation of alarms for your searches using the alarm definitions. Use the auto-generated queries in the Request tab to create these alarm definitions, if you want to automate monitoring and alarm generation for any search.

  You must have appropriate privileges to create the alarm definitions.
- **Response**
  This tab displays the query response in the JSON format for the search and filter query that you have configured using the Discover option.

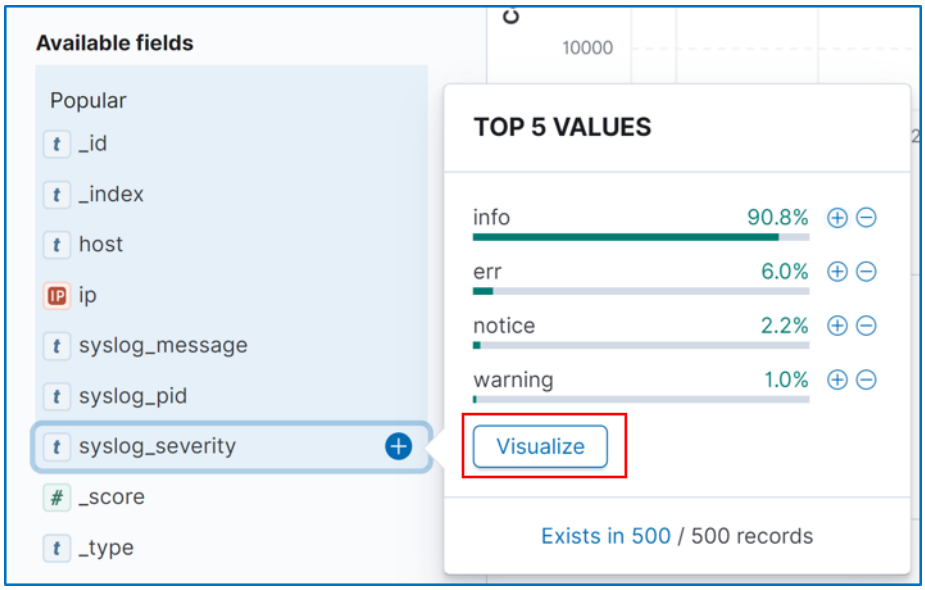
Visualize Fields

The Searches support visualization of data at the field level, when you can the aggregate the data in the field.

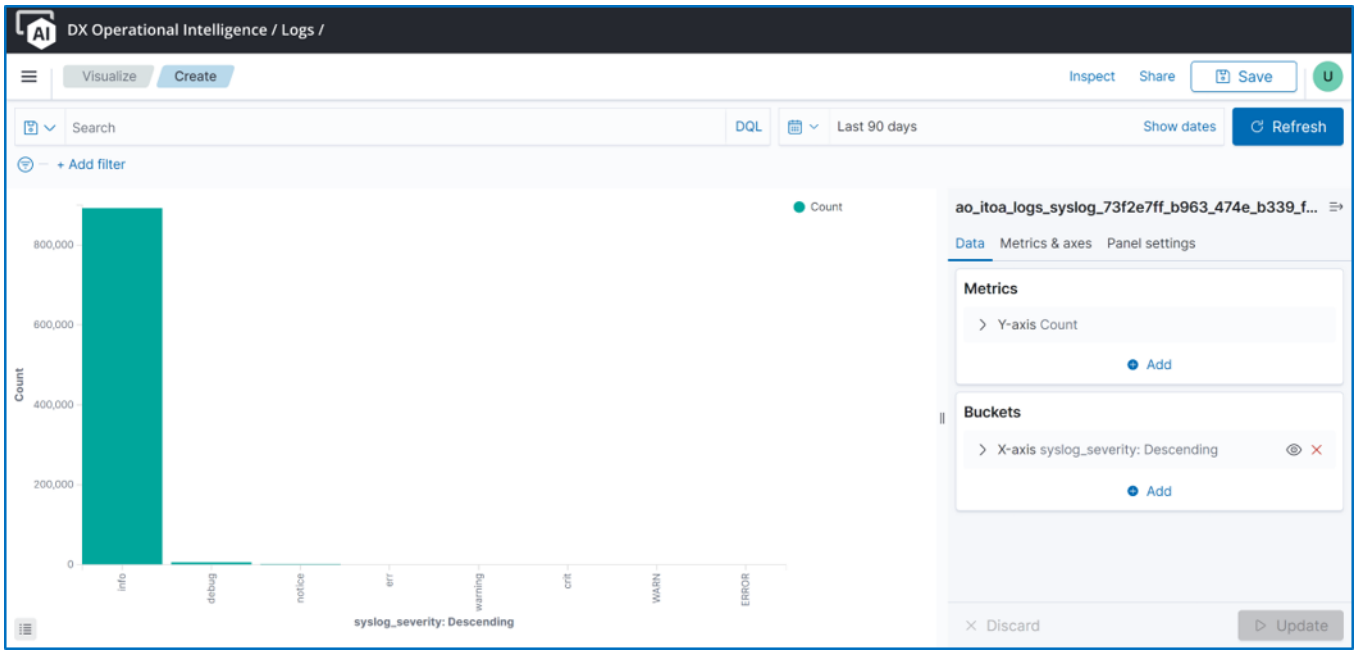
You can define the criteria for visualization using Bucket and Metric Aggregation.

Follow these steps:

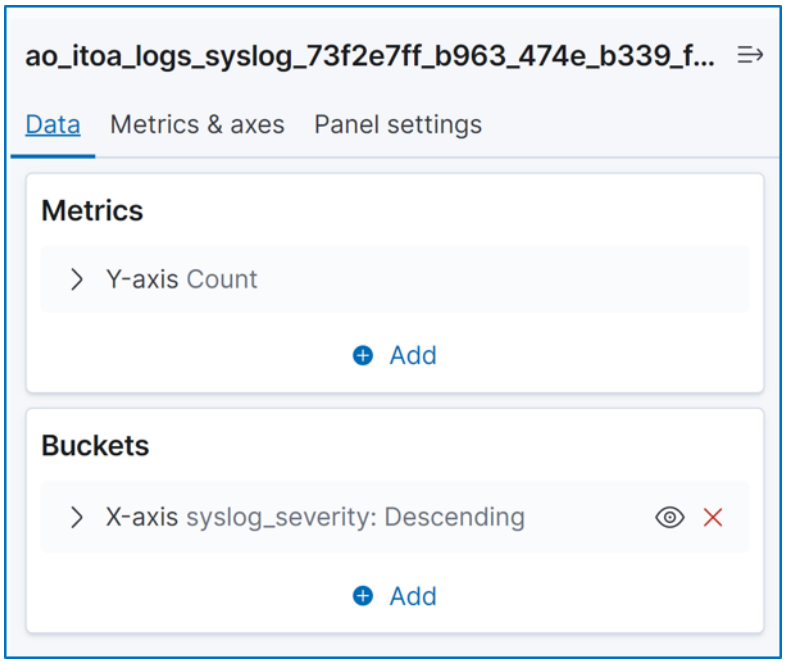
1. Select the field that you want to visualize in the Available Field section (or Selected Fields section) and click **Visualization**.



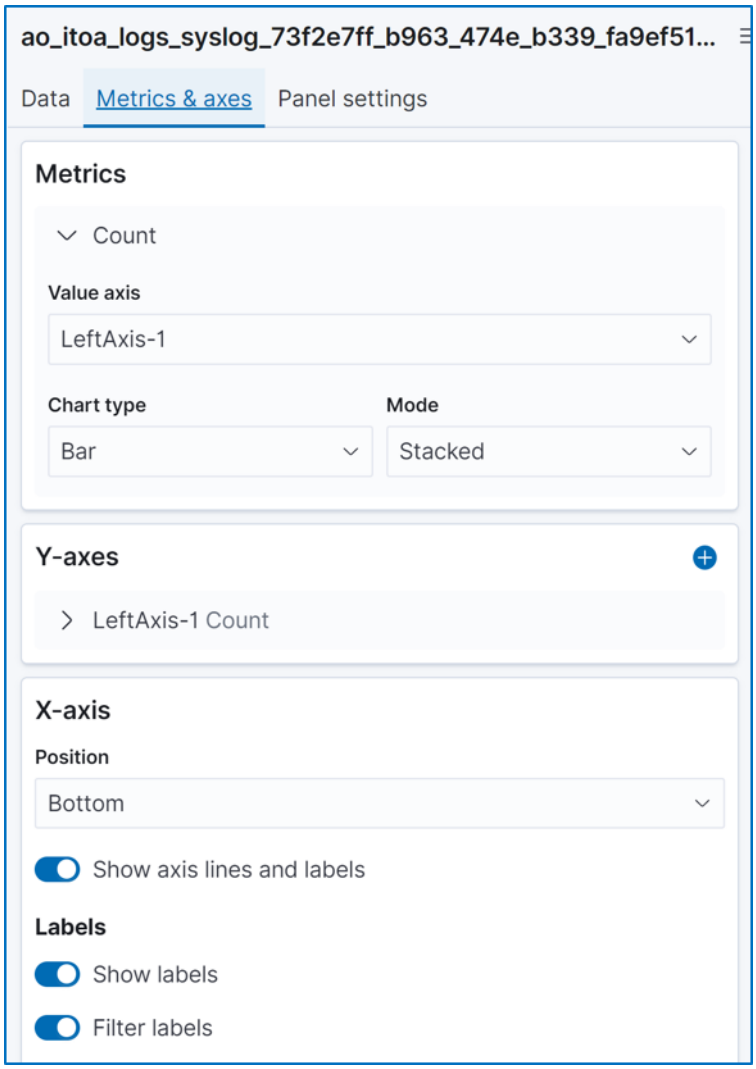
The application displays the visualization as per the defined filter criteria.



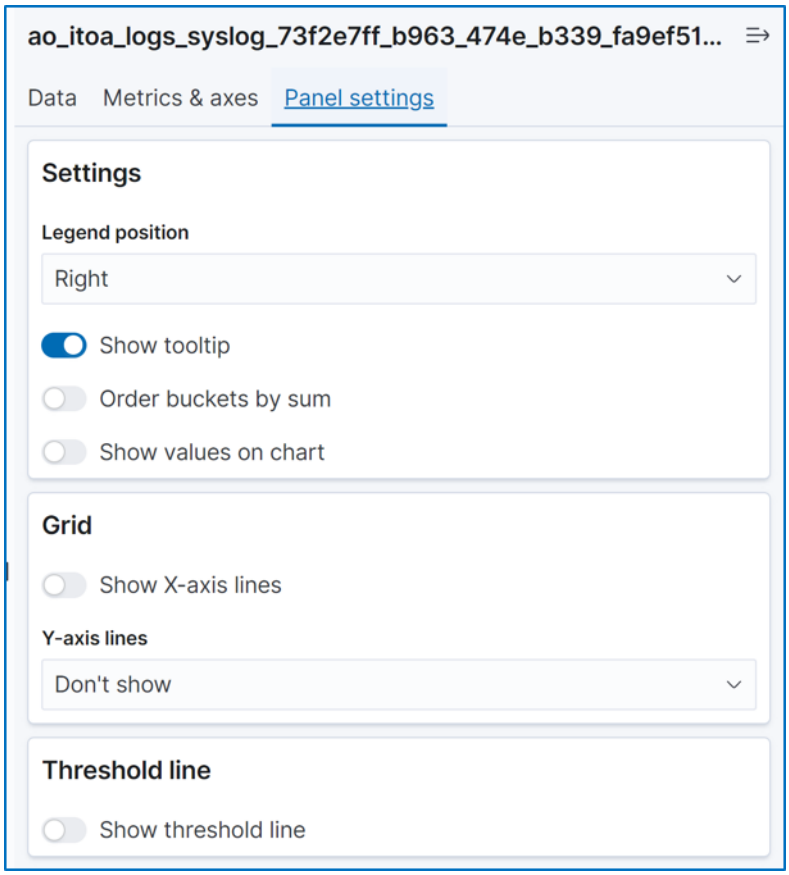
2. Add **Metric** and **Bucket** filters in the **Data** tab.



3. Update the metrics and axes in the **Metrics** and **Axes** tab.



4. Modify the panel settings if any in the **Panel Settings** tab.



5. Click **Refresh**.
6. Click **Save**.

The application saves the visualization. You can use the saved visualization in dashboards.

Dashboards

The **Dashboard** page provides links to the out-of-the-box Log Analytics dashboards and also the option to create custom dashboards.

- Out-of-the-box Dashboards
- Create Custom Dashboards

Pre-Packaged Dashboards

DX Operational Intelligence provides a set of pre-packaged dashboards. These pre-packaged dashboards contain a collection of visualizations and searches that you can access to get started quickly with Log Dashboards.

Each pre-packaged dashboard is created using an out-of-the-box index file and the log type.

The pre-packaged Dashboards are:

- LA - Java Application Logs Overview
- LA MS Exchange Logs Overview
- LA - NGINX Logs Overview
- LA - Oracle Logs Overview
- LA - SQL Server Logs Overview
- LA - Syslog Overview
- LA - Tomcat Logs Overview
- LA - Windows Eventlog Overview
- LA - ZOS Syslogs Overview

LA - Java Application Logs Overview

Index: ao_itoa_logs_log4j_*

Log Type: log4j

This dashboard contains the following visualizations:

Visualization Name	Description
Events Over Time	Displays the number of events for a particular log type. (Count, Log Type, Timestamp, and Events Spike)
Events Over Time By Host	Displays number of events by the host. (Count, Host, Timestamp, and Events Spike for Hosts)
Top Eventlog Categories	Displays top event log categories and the count. (Count and Category)
Top Eventlog Sources with Max Events	Displays top eventlog sources with the maximum number of events and the count. (Count and Source Name)
Top Used Webpages	Displays webpages that are frequently used and the number of times the page was used. (Count and Web Page)
Top Web Servers with Max Requests	Displays top web servers with the maximum number of requests and the count of the requests. (Count and Host)
Top Webpages with Slow Response Time	Displays top webpages with the maximum response time. (Response Time and Web Page)
Webserver Client IP Geo Map	Displays the total number of events and the exact latitudinal and longitudinal coordinates where the events occurred (Center).
All Logs	Displays raw messages for all the log events. (Displaying Time, host, clientip, logtype, message, syslog_message, severity, request, Response Code)

LA MS Exchange Logs Overview

Index: ao_itoa_logs_eventlog_*

Log Type: eventlogs AND sourcename: msexchange*

This dashboard contains the following visualizations:

Visualization Name	Description
Events Over Time	Displays the number of events by the host. (Count, Host, and Timestamp)
Event Distribution by Categories	Displays event distribution by category and also the count. (Count, Host, and Timestamp)
Event Distribution by Source Name	Displays event distribution by source name and the count. (Count, Host, and Timestamp)
Log Levels	Displays count of each log level. (Log level and Count)
All Microsoft Exchange Logs	Displays raw messages for all the events.

LA - NGINX Logs Overview

Index: ao_itoa_logs_nginx_access

Log Type: nginx_access

This dashboard contains the following visualizations:

Visualization Name	Description
Server Response Time	Displays number of requests and the response time. (Number of Requests, Average Response Time, Max Response Time, Min Response Time)
Throughput Over Time	Displays sum of bytes represented as throughput during the time interval (Sum of Bytes)
Top URIs Causing 404 Responses	Displays top requests with a maximum number of 404 responses. (Count and web page)
Top Webpages with Slow Response Time	Displays top webpages with the maximum response time. (Max web page)
Top Used Webpages	Displays top webpages with the maximum number of requests. (Count and web page)
Client IPs with Max Requests	Displays top client IPs with the maximum number of requests (Count and Users)
Top Used HTTP Methods	Displays top HTTP methods with the maximum number of requests (Count and Method)
Requests Over Time	Displays count of events that are received over time (Count and Response Code)
HTTP Error Response Codes	Displays distributions of error response codes by the log type and response code. (Count and Response Code)
NGINX: Top Servers with Max Requests	Displays top host with the maximum number of requests (Count and Host)
NGINX: Client IP Geo Map	Displays position by Longitude and Latitude on geoIP for number of records (Count and GeoIP)
Log Analytics: All NGINX Logs	Displays raw logs for NGINX logs.

LA - Oracle Logs Overview

Index: ao_itoa_logs_oracle_*

Log Type: oracle (Oracle Alert Logs), oracle_audit (Oracle Audit Logs)

The Oracle Logs are categorized into, Alert Logs and Audit Logs. This dashboard contains the following visualizations:

Visualization Name	Description
Alerts Over Time	Displays number of Oracle alerts generated by hosts. (Count, Host, Timestamp, and Event Spikes)
ORA Occurred	Displays errors (ORA codes) encountered and the count. (Code and Count)
Account Lockout	Displays hosts and the number of times the account got locked. (Count, Filter, and Host)
Deadlock Detected	Displays hosts where deadlocks were detected and the count. (Host and Count)
Internal Error Occurred	Displays hosts that encountered the highest number of internal errors. (Count and Host)
Audit Logs Over Time	Displays number of Oracle audit logs by host. (Count, Host, Timestamp)
Logon Failed Over Time	Displays number of times the login failed. (Count and Timestamp)
Logon/Logoff Trends	Displays log in and logout trend.
Status Trend	Displays number of times the database has started or has run.
All Oracle Logs	Displays raw messages for all the Oracle events.

LA - SQL Server Logs Overview

Index: ao_itoa_logs_sqlserver_*

Log Type: SQL server logs AND sourcename: mssqlserver (audit log)

This dashboard contains the following visualizations:

Visualization Name	Description
Events Over Time	Displays the number of events for each host. (Count, Host, and Timestamp)
Top Servers with Max Errors	Displays servers with maximum errors and the count of errors. (Host and Count)
Top Processes with Max Errors	Displays processes with maximum errors and the count of errors. (Process and Count)
Errors Over Time	Displays the number of errors by the host. (Count, Host, and Timestamp)
Start/Shutdown Events	Displays the number of times the hosts have started or have shut down. (Host, Filter, and Count)
Logon Failed by Client IPs	Displays the number of times logins by the client IPs have failed. (Count, Client IP, and Timestamp)
Queries with Longer Time	Displays queries that took a long time to process.

Visualization Name	Description
Top Logged-on Active IP Addresses	Displays top client IPs that have logged in and are active and the count. (Count and Client IP)
Transaction Activity	Displays the number of times the transaction was rolled forward or rolled back.
Restore Activity	Displays the number of times an action was restored or completed. (Count, Activity, and Timestamp)
All SQL Server Logs	Displays raw messages for all the events.

LA - Syslog Overview

This dashboard contains the following visualizations: (Operating systems message logs UNIX and LINUX, physical servers, and network devices)

Log Type: syslog

Visualization Name	Description
Events Over Time	Displays number of events by host. (Count, Host, Timestamp, and Event Spikes)
Failed Logon/Connection Attempts	Displays number of failed logons or failed connection attempts. (Count, Filter, and Timestamp)
Severity Distribution	Displays distribution of severity and the count. (Severity and Count)
Programs and Severity Distribution	Displays program and level of information that has been written to the log file for each program. (Count, Severity, and Program)
Facility Distribution by Host	Displays distribution of facilities by host. (Count, Host, and Facility)
Top Host with Max Events	Displays top host with the maximum number of events. (Count and Host)
Priority Distribution	Displays Priority and Count.
SUDO/CROND Jobs by Host	Displays top SUDO or CROND jobs by host and the count. (Count, Host, and Filter)
Top Facilities with Max Events	Displays top facilities and the number of events for each facility. (Count and Facility)
All Syslog Logs	Displays raw messages for all the events.

LA - Tomcat Logs Overview

Index: ao_itoa_logs_tomcat_*

Log Type: tomcat_access and tomcat (Catalina Logs)

This dashboard contains the following visualizations:

Visualization Name	Description
Server Response Time	Displays the total number of requests that are made to the server, average response time, maximum response time, and minimum response time for the selected time period.
Requests Over Time	Displays the number of requests that are made to the server during the selected time period. (Count, Timestamp, and Event Spikes)
Top Webpages with Slow Response Time	Displays webpages with the maximum response time. (Max Response time and Web Page)

Visualization Name	Description
Top Clients Causing 4xx Errors	Displays IP addresses of the clients that encountered 4xx errors and the count of errors. (Client IP and Count)
Top URIs Causing 404 Responses	Displays top ten links that caused 404 responses in a web page. (Count and Web Page)
HTTP Status Codes	Displays response codes and the count. (Response Code and Count)
Top Used Webpages	Displays webpages that are frequently used and the number of times the page was used. (Count and Web Page)
Top Used HTTP Methods	Displays top HTTP methods used. (Count and Method)
Warning/Severity by Host	Number of warning or severity messages by host. (Count, Host, and Warn/Severity)
Client IPs with Max Requests	Displays top client IPs with the maximum number of requests and the number of requests. (Count and Client IP)
Client IP Geo Map	Displays number of requests and the latitudinal and longitudinal coordinates from where the requests were made. (Count and Center)
Exceptions Over Time	Displays exceptions by the host. (Count, Host, and Timestamp)
Throughput Over Time	Displays sum of bytes downloaded per minute. (Sum of Bytes and Timestamp per Minute)
Top Servers with Max Errors	Displays servers with the maximum errors and the count of errors. (Host and Count)
All Tomcat Logs	Displays raw messages for all the events.

LA - Windows Eventlog Overview

Index: ao_itoa_logs_eventlog_*

Log Type: eventlogs

This dashboard contains the following visualizations:

Visualization Name	Description
Events Over Time	Displays the number of events by the host and the timestamp. (Count, Host, and Timestamp)
Logon Success vs Failure	Displays the number of times the log-in event succeeded or failed and the timestamp. (Count, Filter, and Timestamp)
Errors/Warnings Over Time	Displays the number of errors or warnings that are thrown and the timestamp. (Count, Filter, and Timestamp)
Top Source Generating Most Events	Displays top sources generating most events and the number of events that are generated by each source. (Count, Host, and Source)
Severity Distribution	Displays severity distribution and the count. (Severity and Count)
Computer Account Management	Displays account activities such as accounts created, deleted, and modified by the host. (Count, Host, and Event)

Visualization Name	Description
Top Users with Most Events	Displays top users with most events and the number of events for each user. (Count and User)
User Account Management	Displays user account-related activities such as user-created, deleted, and password change by the host. (Count, Host, and Event)
Top Categories	Displays event categories by host. (Count, Host, and Category)
System Restarts	Displays the number of times the systems restarted by the host.
User Account Usage	Displays usage such as login and logout success or failure. (Count, Host, and Filter)
Changes to Administrative Groups	Displays administrative events by the host. (Count and Host)
Top Host with Max Events	Displays top host with the maximum number of events and the number of events. (Count and Host)
System Management	Displays system events such as application crashes, software and service installations, system or service failures by the host. (Count, Host, and Filters)
Event Distribution by Types	Displays event types by the host (Count, Host, and Event Types)
Recent Policy Changes	Displays policy changes-related to events by the host.
Windows Update Errors and Firewall Events	Displays Windows update errors and firewall events by the host.
All Windows Eventlogs	Displays raw messages for all the events.

LA - ZOS Syslogs Overview

Index: ao_itoa_logs_zos_syslog

Log Type: zos_syslog

This dashboard contains the following visualizations:

Visualization Name	Description
Events Over Time	Displays the number of events that are received over time categorized by the host. (Count, Timestamp, and Host)
Message ID Distribution	Displays the distribution of messages that are categorized by message ID. (Count and Message ID)
Top IP with Max Events	Displays top IPs receiving the maximum number of events. (Count and IP)
Top Host Generating Max Events	Displays top hosts with the maximum number of events generation. (Count and Host)
Messages by Host Name	Displays count of messages by the host that is categorized by message ID. (Count, Host, and Messages)
Job Distribution by Host	Displays count of jobs that are distributed by the host and categorized by the job ID. (Count, Host, and Job ID)
All ZOS Syslogs	Displays raw logs for all the events.

Create Custom Dashboards

DX Operational Intelligence enables you to create your own dashboards using visualizations and searches. The custom visualizations and searches allow you to focus on the data that is crucial to you and help you understand your data better.

Follow these steps:

1. Log in to DX Operational Intelligence and click Log Analytics in the left navigation pane.
   The Log Analytics page opens in a new tab.
2. Click the **Dashboard** link under **OpenSearch Dashboards**.
3. Click

.
4. Perform one of the following actions:
   - Click the **Add an Existing** link to add an existing visualization panel. Select the panels from the **Add Panels** pop-up.
   - Click

to create a new visualization for the dashboard.
5. Define search and filter criteria to refine the data in your dashboard after adding the searches and visualizations.
6. Click **Options** and enable the following toggles:
   - **Use margins between panels** to add margins between the panels in the dashboard.
   - **Show panel Titles** to show panel titles.
7. Click **Save** and provide a name for the dashboard for future access.
   For more information, see the OpenSearch Documentation.

Management

This section provides the link to stack management where you can manage your dashboards and saved objects.

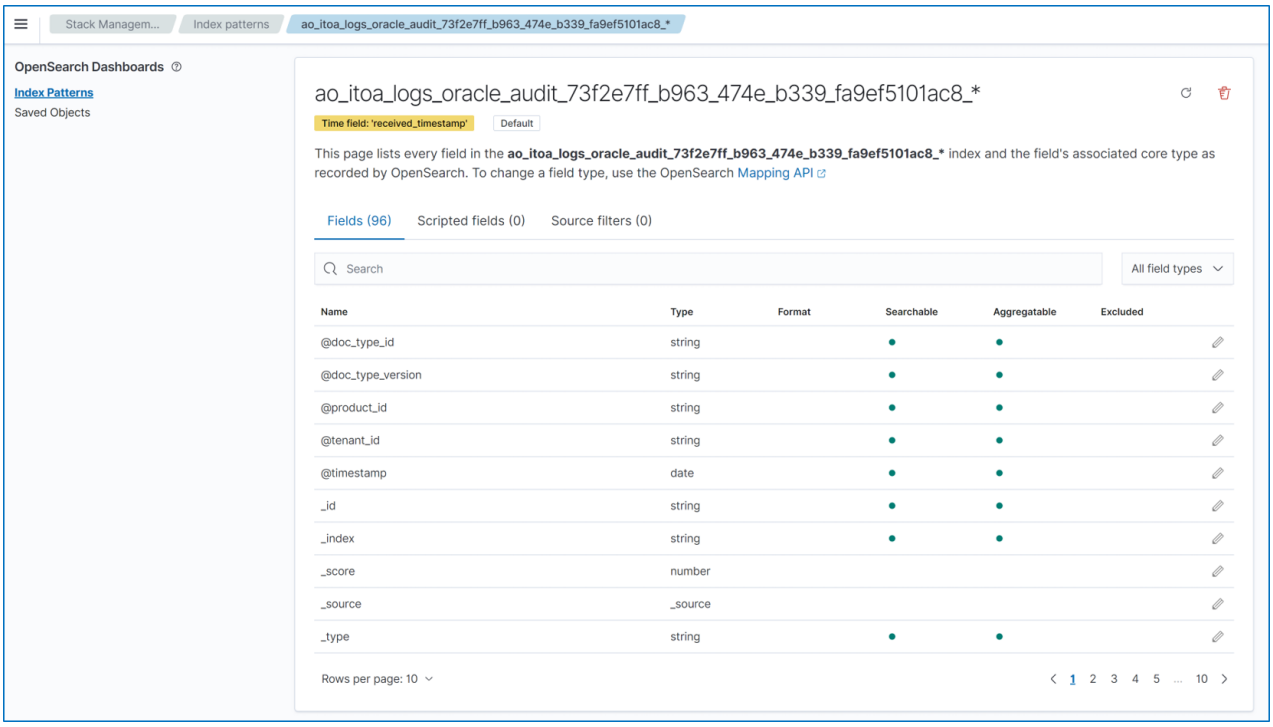
Stack Management

The Stack Management provides links to:

- Index Patterns
- Saved Objects

Index Patterns

The Index Patterns page is divided into the following tabs:



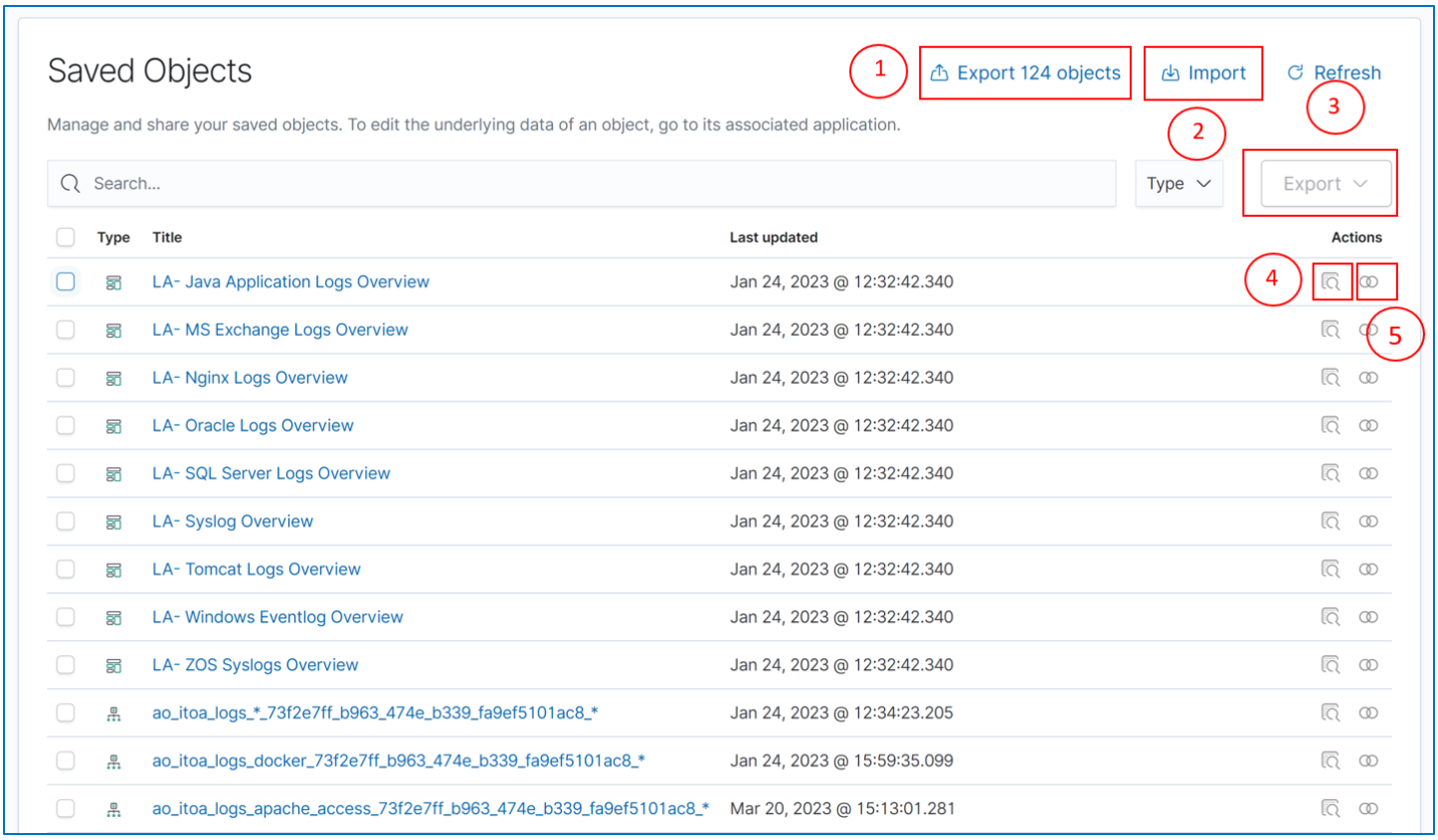
NOTE
After Broadcom onboards a new custom log type, the fields for the newly onboarded log type are visible only after the log type is refreshed. To refresh, search for the custom index and then click the **Refresh field list** icon to refresh the log type.

- **Fields:** This page lists all the index patterns that you can use to retrieve data from OpenSearch. Click any index pattern to view all the fields in the index and the field's associated core type as recorded by OpenSearch.
- **Scripted Fields:** On this page, you can add a scripted field. You can use scripted fields in visualizations and can display them in your documents. However, you cannot search scripted fields.
- **Source Filters:** On this page, you can add the source filters. Source filters can be used to exclude one or more fields when fetching the document source. This happens when viewing a document in the Discover app, or with a table displaying results from a saved search in the Dashboard app. Each row is built using the source of a single document, and if you have documents with large or unimportant fields you may benefit from filtering those out at this lower level.

  NOTE
  Note that multi-fields appear incorrectly as matches in the table. These filters only apply to fields in the original source document, so matching multi-fields are not actually being filtered.

Saved Objects

On the Saved Objects page, you can manage and share your saved objects.

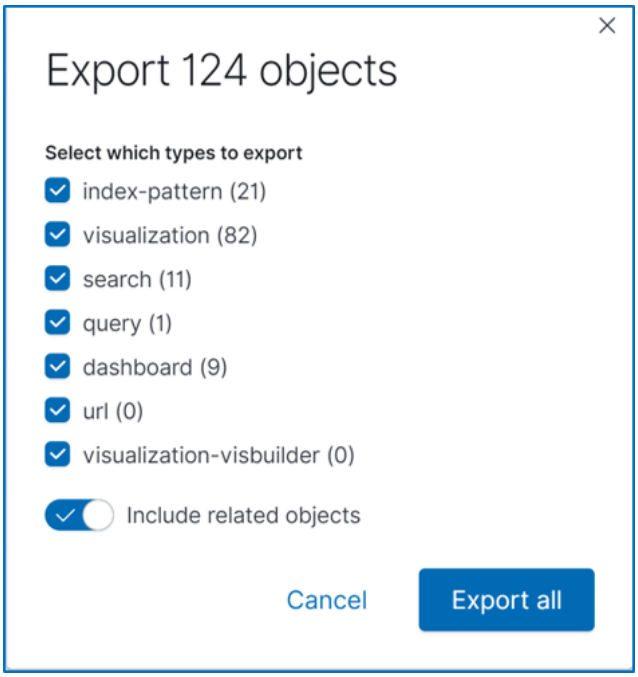


You can,

- Export the saved objects (1)
- Import the saved objects (2)
- Export the dashboard (3)
- Inspect the saved object (4)
- View the relationship this saved object has to other saved objects (5)

Export the Saved Objects

Click the **Export objects** link and select the type of objects to export. You can export all the objects or exported only the selected type of objects. You can also include the related objects.

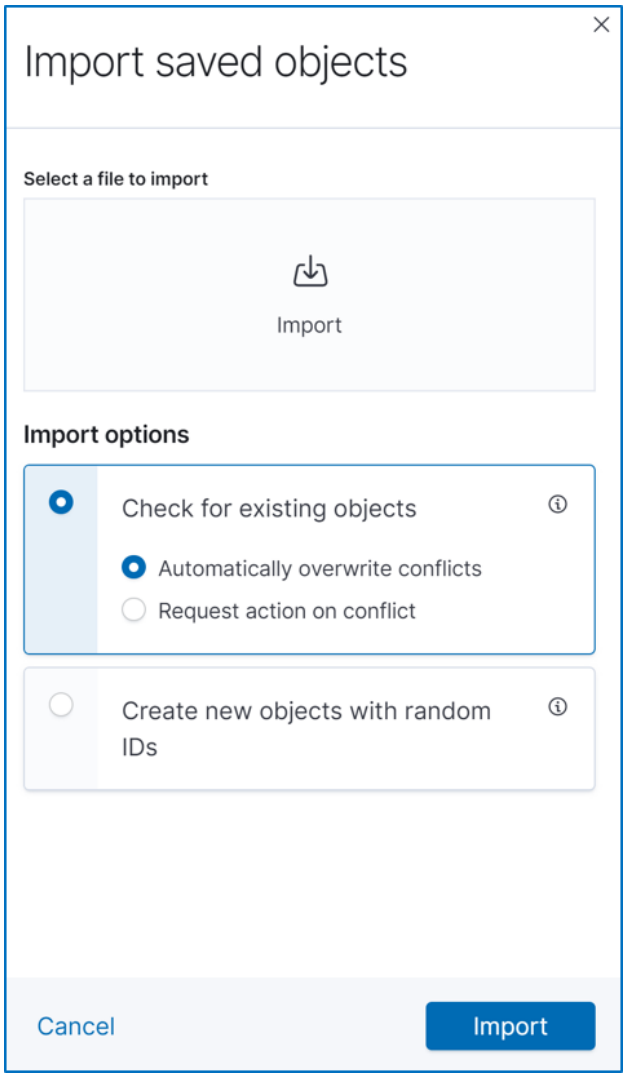


Import the Saved Objects

You can import the saved objects.

Follow these steps:

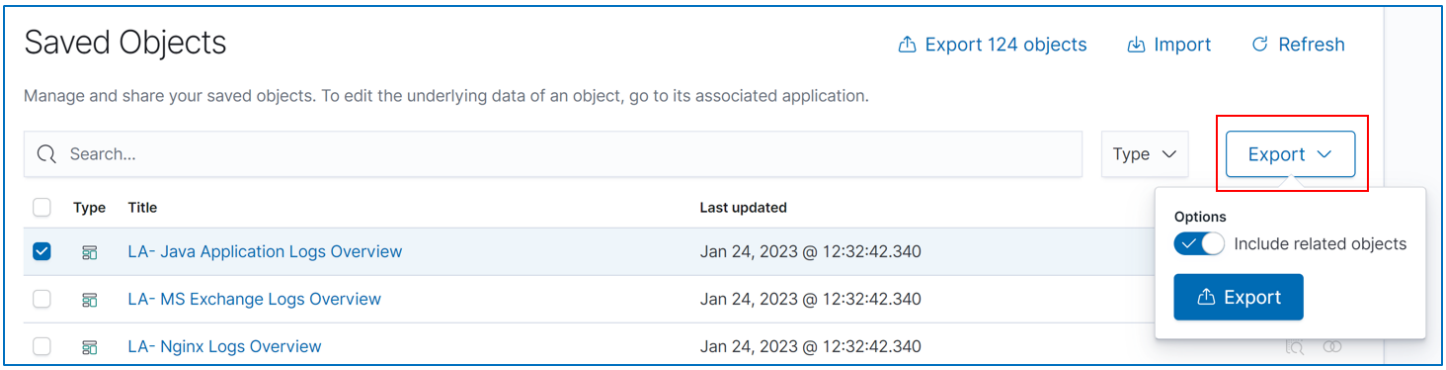
1. Open the **Saved Objects** page.
2. Click **Import**.



3. Select the file to import.
4. Select the import options as required:
   – **Check for existing objects:** Check if objects were previously copied or imported.
     - Automatically overwrite conflicts
     - Request action on conflict.
   – **Create new objects with random IDs:** Use this option to create one or more copies of the object.
5. Click **Import**.

Export the Dashboard

On the Saved Objects page, select the dashboard and click **Export**. You can include the related objects too.

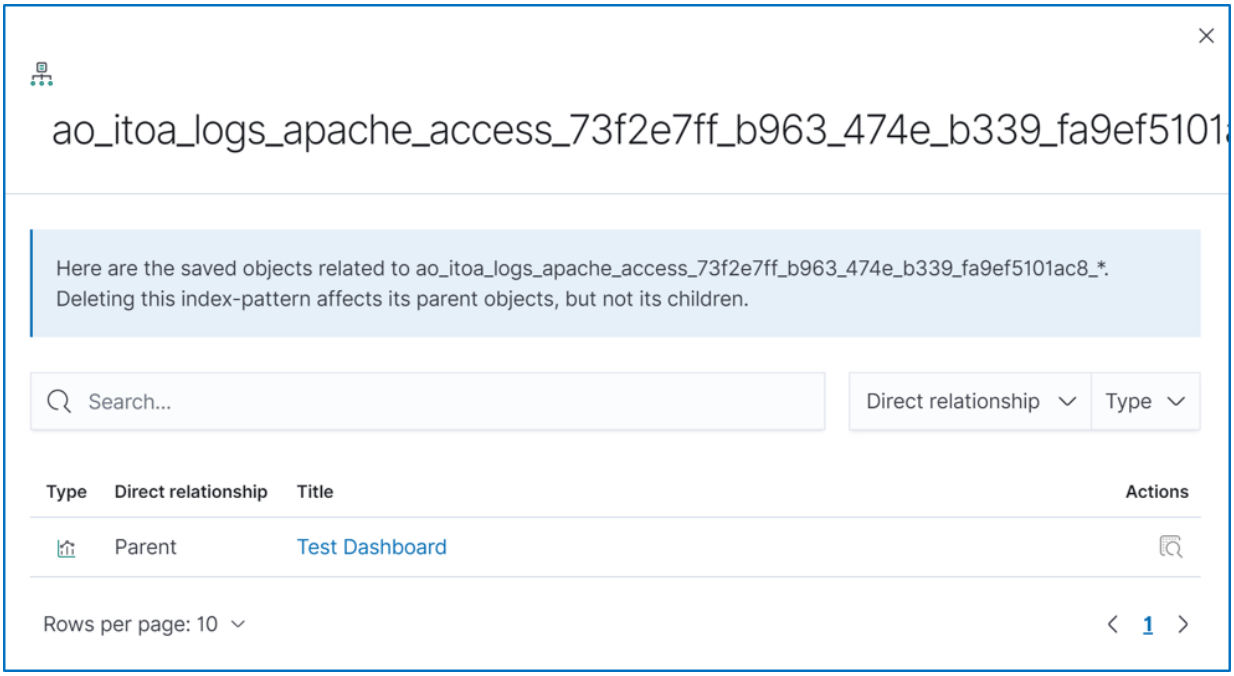


Inspect the Saved Object

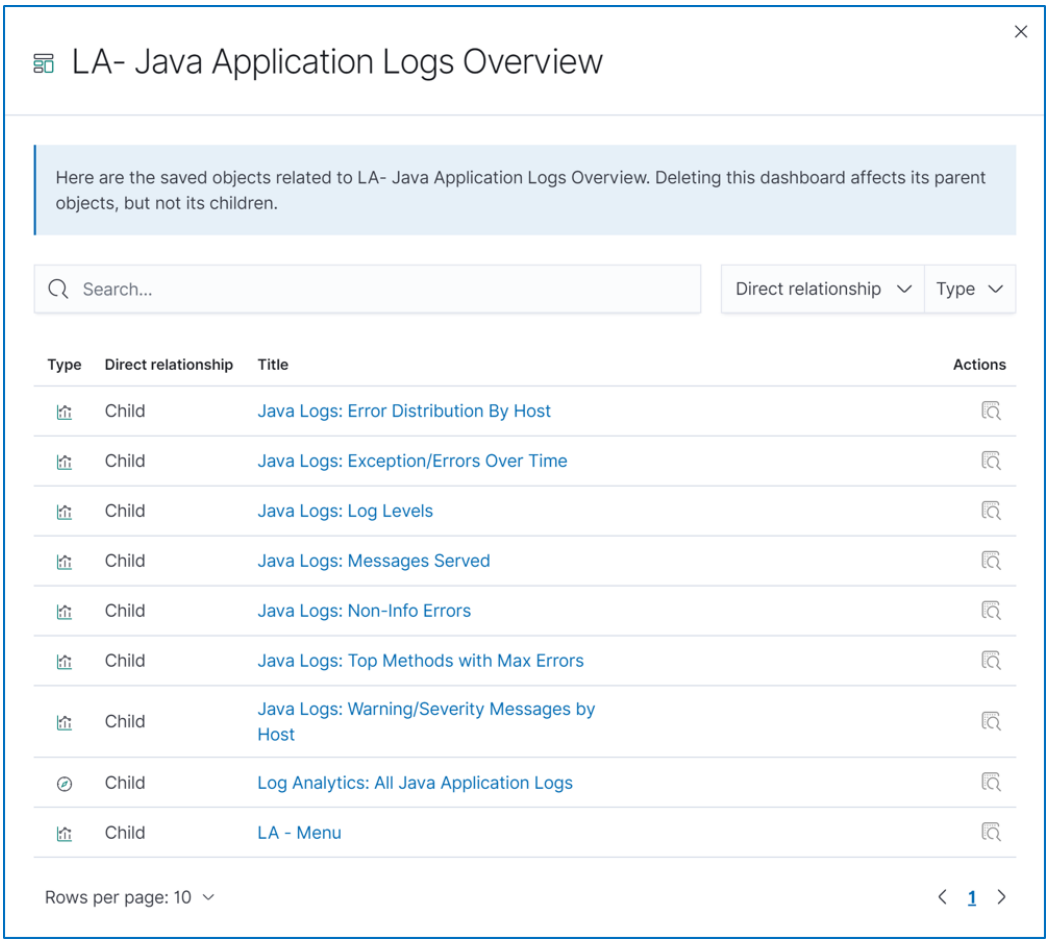
You can inspect the saved object using the icon that is displayed under the Actions column. If you click this icon for a dashboard, the **Edit Dashboard** page is displayed. You can also delete the dashboard from here. If you click this icon for an index pattern, the index patterns for that index are displayed.

View Saved Objects Relationships

When you click this icon for a dashboard, all the related saved objects are displayed. If you delete this dashboard, all the parent objects are affected. No children are affected.



If you click this icon for an index pattern, all the saved objects that are related to this index are displayed.



Log Alarm Configuration

DX Operational Intelligence provides a User Interface for configuring the log pattern-based alarms that continuously look for the defined log pattern and triggers an alarm and notification, based on the configured conditions.

This section contains the topics that help you configure alarm definition:

- Log Alarm Overview
- Configure Alarm Definition
- Manage Alarm Definitions
- View Alarms in Alarm Analytics

Log Alarms Overview

DX Operational Intelligence supports the configuration of log alarm definitions. The alarm definitions contain the criteria for generating alarms when events are found in the logs. Log alarms enable the system administrators and the support personnel to respond to events quickly.

As a tenant administrator, you can configure the alarm definition and provide the threshold conditions for generating alarms.

Configuring alarm definitions help you to effectively reduce the mean time to triage and resolve issues. While configuring the log alarms definitions,

- Determine and specify the events that must generate the alarms.
- Configure the threshold conditions for the alarm to trigger.
- Define the severity of the event.
- Provide the log attributes that enable the support personnel and the administrators to triage and resolve issues quickly.

Configure Log Alarm Definition

You can configure an alarm definition to generate alarms as per the required criteria. An alarm definition is a job that runs the query on the indexes at a defined schedule and generates an alarm (notification) if the criteria to generate one matches in the query result. The results of these queries are then used as input for one or more thresholds. A threshold is a condition to raise an alarm.

An alarm definition runs the query and generates alarm notifications if the definition criteria match.

The Alarm definition consists of the following components:

- **Query**
  The alarm definition uses Query Domain Specific Language (DSL) for querying. The query helps in selecting the log data or the result set on which the threshold conditions will be applied. You can manually define a query or use an auto-generated query from your search using the **Discover** option in the dashboards.
- **Query Results**
  On executing the query, the alarm definition generates the query results. The query results contain the following information:
  – The data that you can use to define threshold conditions for the alarm generation.
  – The log attributes that you can use in the alarm message. The log attributes contain information that is useful for triaging and resolving a particular issue.
- **Threshold Conditions**
  The threshold conditions define the criteria for the alarm generation. The threshold conditions use the data from the query results to define the criteria and the severity of an alarm. The alarm definition generates the alarms periodically as per the defined threshold conditions.

Complete the following tasks to configure an alarm definition:

- Create Alarm Definition
  – Use Dashboards Generated Queries in Alarm Definition
- Define Alarm Thresholds
- Configure Alarm Message

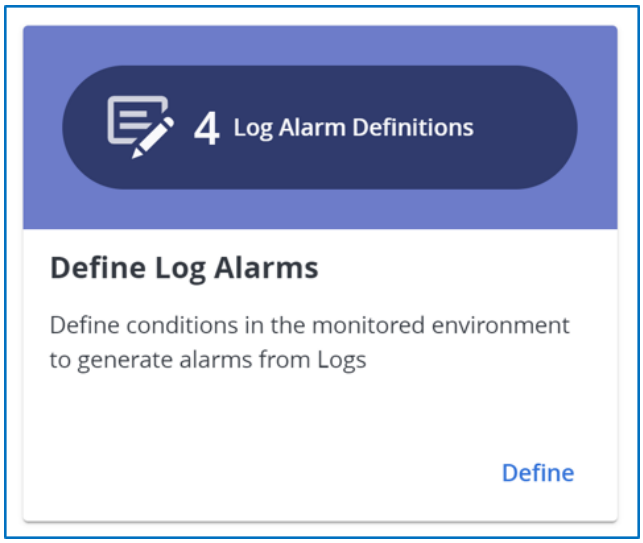
Create Alarm Definition
You can create alarm definitions to monitor the log data of the selected log indices.

NOTE

- The **Log Alarm Definitions** tile is available only if Log Analytics is enabled for the tenant.
- You can create or save the alarm definition only after the query is run successfully, the added threshold conditions are tested, and all the mandatory fields are entered without any validation errors. If you create the definition without adding the threshold conditions, the alarm definition is created but the status of the definition remains as Inactive.
- By default, you can add only a maximum of 50 alarm definitions for a tenant. To add more definitions, contact **Broadcom Support**.
- You can use variables to customize the alarm definition. For more information, see the **Alarm Definition Variables** section on this page.

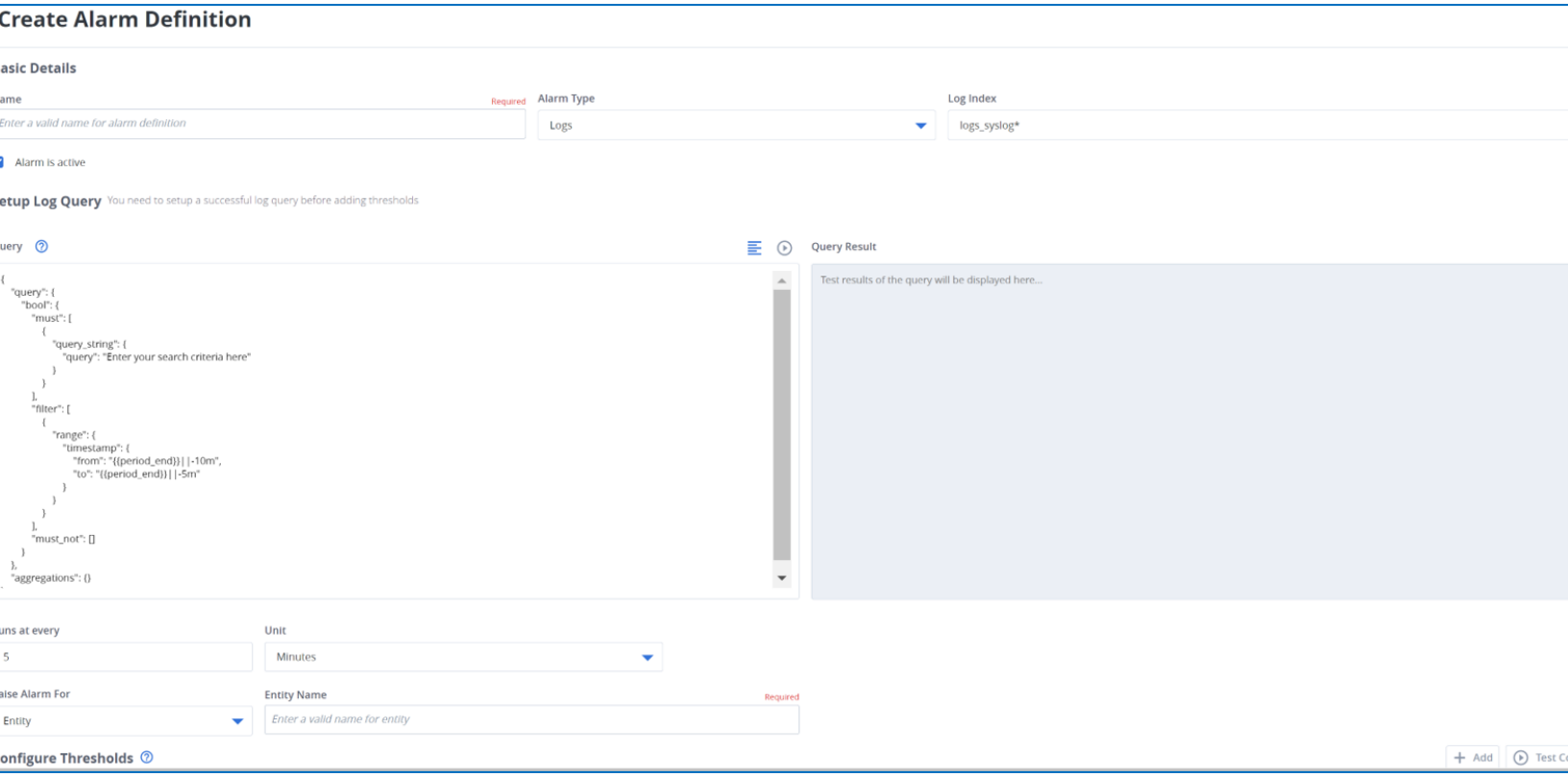
Follow these steps:

1. Log in to DX Operational Intelligence.
2. Click **Settings** in the left navigation pane.
3. Click **Define** in the **Define Log Alarms** tile.

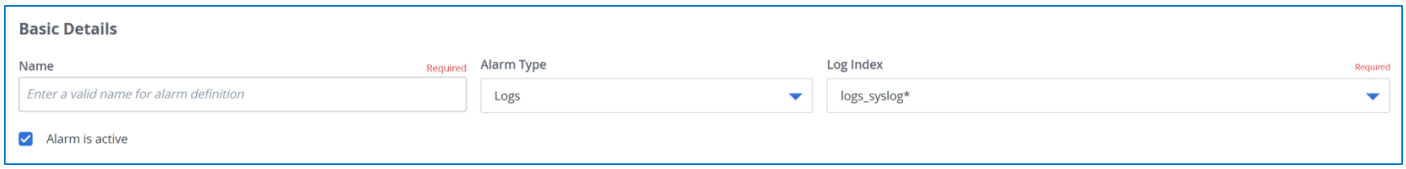


The **Log Alarm Definitions** page is displayed.

4. Click **+ Alarm Definition**.
   The application displays the **Create Alarm Definition** page.



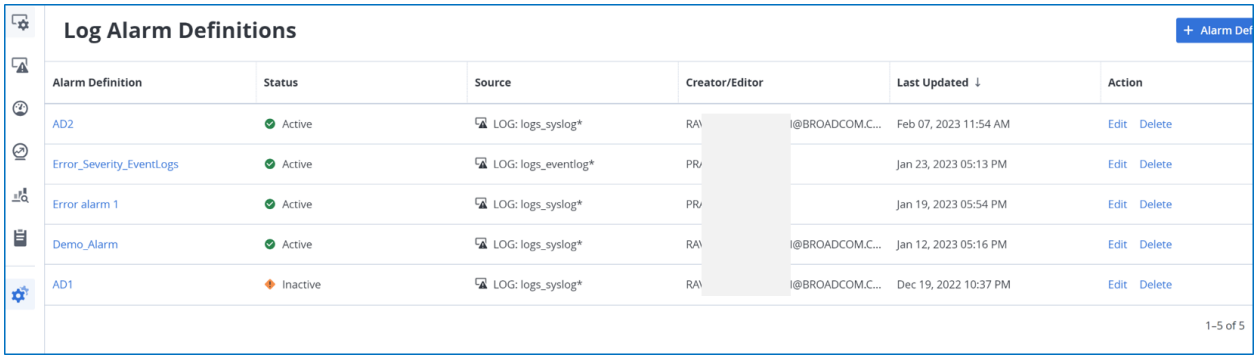
5. Complete the following details in the **Basic Details** section:



– **Name:** Enter a unique alarm definition name in the **Name** field. The name can contain alphanumeric characters and an underscore.
– **Alarm Type:** Select **Log.**
– **Log Index:** Select the log index to monitor.
– **Alarm is Active:** Select this checkbox. The alarm definition is active only when this check box is selected.

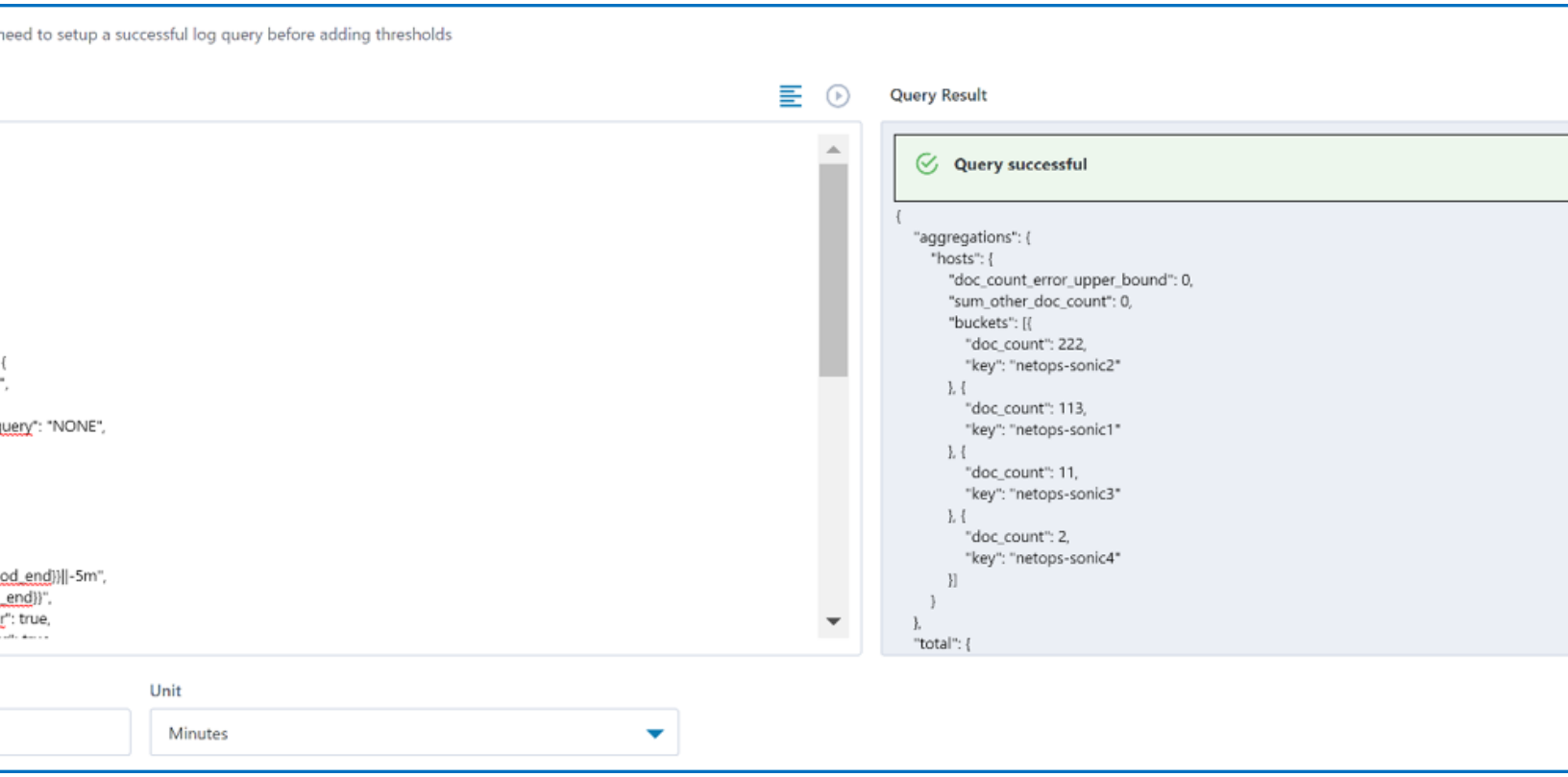
NOTE

The following image illustrates an inactive alarm log definition:



6. Define the query in the **Setup Log Query** section.
   a. Define the query using Query DSL. By default, a standard query template is populated. You can modify the template to create your own queries. Alternatively, you can also use the auto-generated queries.

NOTE



For more information about defining queries, see the following links:

- Boolean Queries
- Term-level Queries
- Full-text Queries
- Query String Syntax
- Range Queries

b. Click

to execute the query results in the **Query Results** section.

NOTE

If the results do not match your requirements, modify the query.

c. Specify the interval or frequency of the execution using the **Runs at every** and **Units** field.

d. Select the option for **Raise Alarms for** and also enter the entity name. The entity name is available in the **Entity Details** section of the **Monitored Inventory** page.

7. Click **Create**.

The application creates the alarm definition but the status is **Inactive**.

Alarm Definition Variables

The following table lists the variables that you can use to customize the alarm definition:

Variable	Data Type	Description
ctx.monitor	Object	Includes `ctx.monitor.name`, `ctx.monitor.type`, `ctx.monitor.enabled`, `ctx.monitor.enabled_time`, `ctx.monitor.schedule`, `ctx.monitor.inputs`, `triggers` and `ctx.monitor.last_update_time`.
ctx.monitor.user	Object	Includes information about the user who created the alarm definition. Includes `ctx.monitor.user.backend_roles` and `ctx.monitor.user.roles`, which are arrays that contain the backend roles and roles that are assigned to the user.
ctx.monitor.inputs	Array	An array that contains the indexes and definition that is used to create the alarm definition.
ctx.monitor.inputs.search.indices	Array	An array that contains the indexes the alarm definition observes.
ctx.results	Array	An array with one element. For example, `ctx.results[0]`. Contains the query results. This variable is empty if the alarm definition was unable to retrieve results. For more information, see `ctx.error`.
ctx.last_update_time	Milliseconds	Displays UNIX epoch time of when the alarm definition was last updated.
ctx.periodStart	String	Displays the UNIX timestamp for the start period during which the alarm was triggered. For example, if an alarm definition runs every 10 minutes, a period might begin at 11:40 and end at 11:50.
ctx.periodEnd	String	Displays the end period during which the alarm was triggered.
ctx.error	String	Displays the error message if the alarm definition was unable to retrieve results or was unable to evaluate the threshold due to a compile error or null pointer exception. Displays Null otherwise.
ctx.alert	Object	Displays the current, active alarm (if it exists). Includes `ctx.alert.id`, `ctx.alert.version`, and `ctx.alert.isAcknowledged`. Displays Null if no alarm is active. Available only with query-level alarm definitions.

Use OpenSearch Dashboards Generated Queries in Alarm Definition

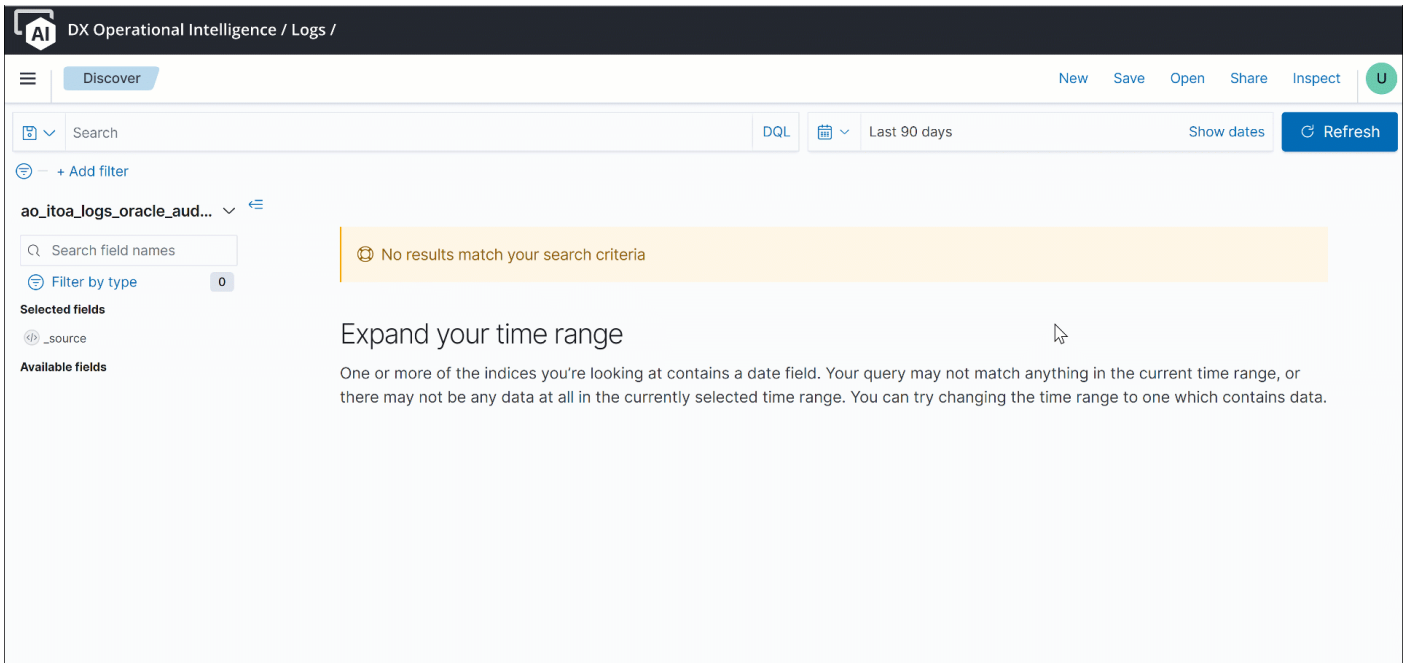
The searches that are created on the **Discover** page of Log Analytics facilitate manual monitoring of the log data using the defined filter criteria. The **Inspect** option of the search generates a query using the filter criteria for the search.

You can use this query to create the alarm definitions and obviate the manual monitoring by automatically generating the alarms.

Follow these steps:

1. Click **Log Analytics** in DX Operational Intelligence.
   The **Discover** page is displayed.

2. Copy the query of the saved search.



   a) Click **Open** and select the saved search.
   b) Click **Inspect** and then click the **Request** tab.
   c) Select and copy the query that is displayed.

3. Navigate to the **Log Alarm Definition** page.

4. Paste the query in the Query section and make the necessary changes.

Define Alarm Threshold Conditions

Alarm threshold conditions help you to determine when to generate an alarm for a particular alarm definition. At every scheduled interval, the alarm definition executes the query and returns the query results.

Using the query results, you can configure the following types of threshold conditions as per your requirements:

- **Simple:** Using the Simple option, you can configure one or more threshold conditions that are based on the hit count of the query results.
- **Advanced:** Using the Advanced option, you can create one or more threshold conditions for an alarm definition using the **Painless** script. The Painless script is the default OpenSearch scripting language and has a syntax similar to Groovy. The threshold script generates a boolean value which is based on the data in the query results. If the generated boolean value is **true**, then the application triggers the alarm generation.
  You can reference the data from the query results while defining the alarm condition script using the **ctx.results[0]** expression.

Follow these steps:

1. Open the alarm definition for which you want to define the threshold conditions.

   NOTE
   You cannot edit the alarm type after you define and save the alarm definition. If you want to use a different alarm type, you must create an alarm definition with the same threshold conditions.

2. Select **Entity** as the value for **Raise Alarm For** in the **Setup Log Query** section. Also, provide the **Entity Name** to which you want to map the alarm generated based on the query and the defined threshold conditions.
   An entity is an element or a Configuration Item that you can associate with a service when creating a service in Services Analytics. The entity name is available in the **Entity Details** section on the **Monitored Inventory** page.

NOTE
You can create a maximum of three simple or advanced threshold conditions for an alarm definition when the alarm type is Entity. DX Operational Intelligence raises a separate alarm for each threshold condition that occurs.

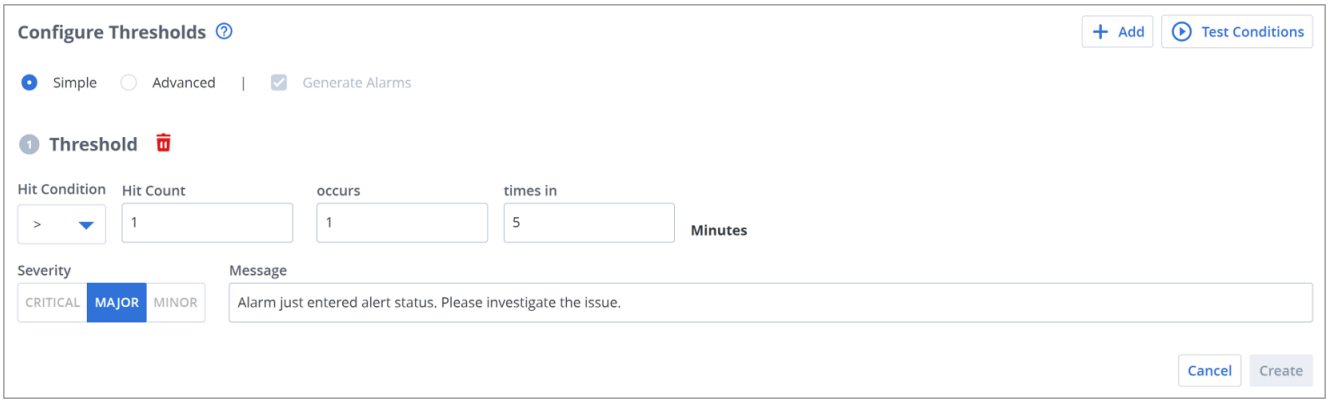
3. Click **+ Add** in the **Configure Thresholds** section.

The application displays the **Configure Thresholds** section with **Simple** selected by default.

4. Select one of the following options.

NOTE
The application also selects the **Generate Alarms** checkbox.

– **Simple:** Complete the following steps to configure the threshold conditions:
  a. Select **Simple**.



  b. **Hit Condition:** Select the relevant operator that you want to use while defining the threshold condition in the Hit Count field. The available operators are '>', '>=' and '==''.
  c. **Hit Count:** Specify the hit count number in the Hit Count field.
  d. **Occurs** and **Times in:** Specify the number of times the specified hit count condition must occur in the **Occurs** and **Times in** fields.

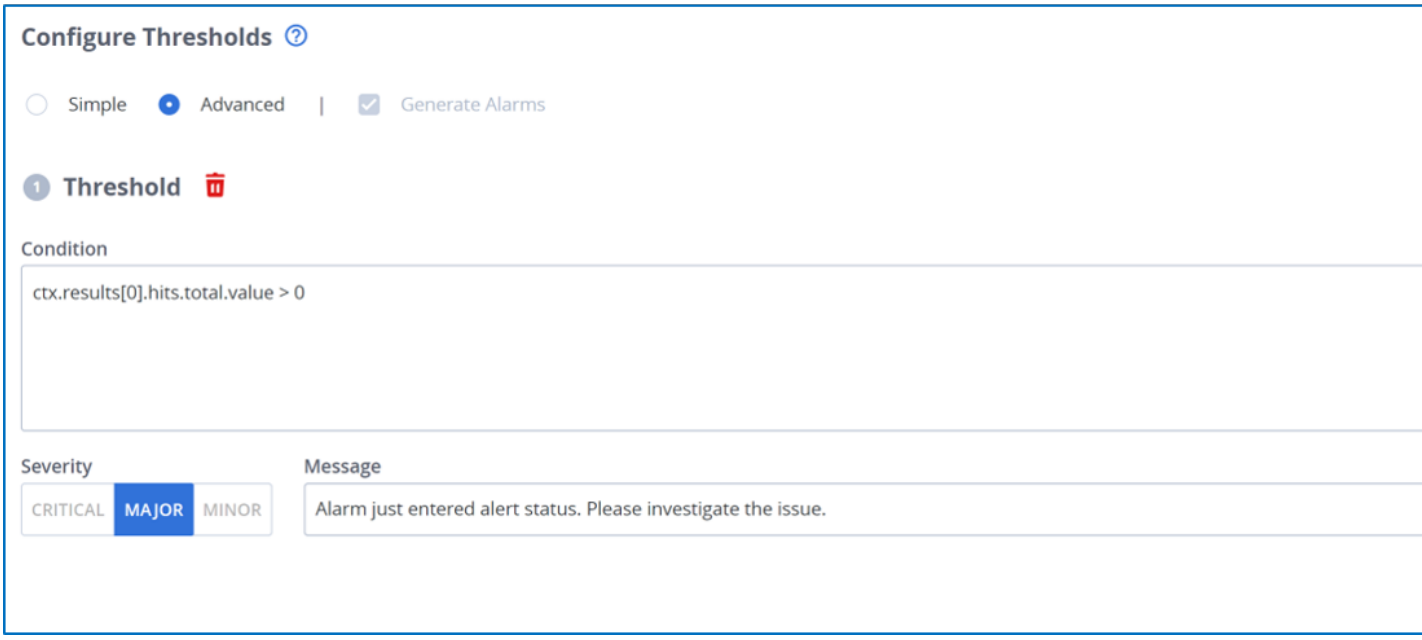
NOTE
While providing the values for the **Occurs** and **Times in** fields, ensure:

- The value in the **Times in** field is in multiple of the query execution frequency that you specify in the **Runs at every** field.
- The value in the **Occurs** field is less than or equal to the **Times in'** per **'Runs at every'** fields (Occurs <= (times in/runs at every)).

DX Operational Intelligence generates the alarm or event when the specified threshold condition is met.

  e. **Severity:** Select the severity of the alarm for the threshold condition that you defined:
    - Critical
    - Major
    - Minor
  f. **Message:** Configure the format of the alarm message for the notification. For more information on including the log attributes in the alarm message, see the Configure Alarm Message section.

– **Advanced Alarm Definition:** Complete the following steps to configure the threshold conditions:
  a. Select **Advanced** and click **+ Add** to display the **Condition** text box.



b. Define the threshold condition in the painless script using the following expression:

```
ctx.results[0]
```

For example, your script can reference:

```
ctx.results[0].hits.total.value or ctx.results[0].hits.hits[i]._source.error_code.
```

c. Specify the number of times the specified condition must occur in the **Occurs** and **Times in** fields.

> **NOTE**
> Ensure that the specified value in the **Times in** field is in multiples of the query execution frequency that you specified in the **Runs at every** field.

DX Operational Intelligence generates the alarm when the specified threshold condition is met.

d. **Severity:** Select one of the following options, to define the severity of the alarm for each threshold condition that you defined.
   - Critical
   - Major
   - Minor

e. **Message:** Configure the format of the alarm message for the notification. For more information on including the log attributes in the alarm message, see the Configure Alarm Message section.

5. Click **Test Conditions** to validate the conditions.

> **NOTE**
>
> If the validation fails, you must revalidate the threshold conditions after making the necessary changes. DX Operational Intelligence does not allow saving an alarm definition with errors.

6. Click **Create / Save**.

DX Operational Intelligence creates the alarm definition with the threshold conditions. You can access the alarms that are generated based on the alarm definition configuration in Alarm Analytics.

Threshold Variables

The following table lists the variables that you can use to define the alarm threshold conditions:

Variable	Data Type	Description
ctx.trigger.id	String	Indicates the threshold ID.
ctx.trigger.name	String	Indicates the threshold name.
ctx.trigger.severity	String	Indicates the severity of the threshold.
ctx.trigger.condition	Object	Contains the Painless script that is used when creating the alarm definition.
ctx.trigger.condition.script.source	String	Indicates the source.
ctx.trigger.condition.script.lang	String	Indicates the language used to define the script that is used to define the threshold. The language must be painless.
ctx.trigger.actions	Array	An array with one element that contains information about the action the alarm definition must trigger.

Sample Alarm Definition Queries and Trigger Scripts

This section provides few sample queries and triggers scripts for Alarm Definition:

Sample 1: Query with filter condition on one or more fields

Get the count of all the documents in past one hour where host name is "ibndev123" and exception is "Nullpointer" and OS is not "its":

```
"query" : {
                    "size" : 0,
                    "query" : {
                      "bool" : {
                        "must_not": [
                                   {
                                     "term": {
                                       "os": "ios"
                                                 }
                                      } ],
                        "filter" : [
                          {
                            "range" : {
                              "timestamp" : {
                                "from" : "{{period_end}}||-1h",
                                "to" : "{{period_end}}",
                                "include_lower" : true,
                                "include_upper" : true,
                                "format" : "epoch_millis",
                                "boost" : 1.0
                              }
                            }
                          },
                          {
                            "match_phrase" : {
                              "exception" : {
                                "query" : "Nullpointer",
                                "slop" : 0,
```

```
                                "zero_terms_query" : "NONE",
                                "boost" : 1.0
                              }
                            }
                          },
                        {
                            "term" : {
                              "host" : "ibndev123"
                            }
                          }
                        ],
                        "adjust_pure_negative" : true,
                        "boost" : 1.0
                      }
                    }
```

Threshold Conditions:

Raise a Severity 1 (Critical) alert, if the total number of documents is more than 1000:

```
ctx.results[0].hits.total.value > 1000
```

Raise a Severity 2 (Major) alert, if the total number of documents is less than 1000:

```
ctx.results[0].hits.total.value < 1000
```

Sample query 2: Query with no filter criteria on a field but aggregation on a field.

Query to find out an average RAM usage in one hour. The aggregation can be the metric aggregation (like avg, sum, min, max) or bucket aggregation. The following example shows the metric aggregation:

```
{
    "size": 10,
    "query": {
        "bool": {
            "filter": [
                {
                    "range": {
                        "timestamp": {
                            "from": "{{period_end}}||-1h",
                            "to": "{{period_end}}",
                            "include_lower": true,
                            "include_upper": true,
                            "format": "epoch_millis",
                            "boost": 1
                        }
                    }
                }
            ],
            "adjust_pure_negative": true,
            "boost": 1
        }
    },
    "aggregations": {
        "avg_cpu": {
            "avg": {
                "field": "machine.ram"
            }
```

```
        }
    }
}
```

Threshold Conditions

Raise an alarm when the CPU usage is more than 90:

```
(ctx.results[0].aggregations.avg_cpu.value > 90) {
  return true;
}
```

Sample query 3: Query with filter condition and bucket aggregation

Query to search for all occurrences of "Exception" keyword and find the top five method names in last one hour. If the occurrences of one of the method names is greater than 500, then trigger an alert:

```
{
    "aggs": {
      "methodnames": {
        "terms": {
          "field": "method",
          "order": {
            "_count": "desc"
          },
          "size": 5
        }
      }
    },
    "size": 0,
    "query": {
      "bool": {
        "must": [],
        "filter": [
          {
            "match_all": {}
          },
          {
            "match_all": {}
          },
          {
            "match_phrase": {
              "message": "*exception*"
            }
          },
          {
            "range": {
              "@timestamp": {
                "from" : "||-1h",
                              "to" : "",
                "format": "strict_date_optional_time"
              }
            }
          }
        ],
        "should": [],
```

```
        "must_not": []
      }
    }
  }
```

Threshold Conditions

Raise an alarm when the documents count is more than 90:

```
for (int i = 0; i < ctx.results[0].aggregations.methodnames.buckets.length; i++) {
  if (ctx.results[0].aggregations.methodnames.buckets[i].doc_count>= "500") {
   return true;
    }
}

  return false;
}
```

Sample query 4: Query with filter condition and bucket aggregation

Query to search for all occurrences of "Exception" keyword and find the top five method names in last one hour. If the occurrences of one of the method names is greater than 500, then trigger an alert:

```
{
    "size": 10,
    "query": {
        "bool": {
            "filter": [
                {
                    "range": {
                        "timestamp": {
                            "from": "{{period_end}}||-1h",
                            "to": "{{period_end}}",
                            "include_lower": true,
                            "include_upper": true,
                            "format": "epoch_millis",
                            "boost": 1
                        }
                    }
                }
            ],
            "adjust_pure_negative": true,
            "boost": 1
        }
    },
    "aggregations": {
        "when": {
            "avg": {
                "field": "machine.ram"
            }
        }
    }
}
2:
```

```
{
    "size": 0,
    "query": {
        "bool": {
            "filter": [
                {
                    "range": {
                        "visit_date": {
                            "gte": "now-1d/d",
                            "lt": "now/d"
                        }
                    }
                }
            ],
            "must_not": [
                {
                    "term": {
                        "is_user_seen": {
                            "value": "true"
                        }
                    }
                }
            ]
        }
    },
    "aggregations": {
        "tenants": {
            "terms": {
                "field": "device_id",
                "size": 100
            },
            "aggs": {
                "uid_count": {
                    "cardinality": {
                        "field": "user_id",
                        "precision_threshold": 40000
                    }
                }
            }
        }
    }
}
3:
 "query": {
          "query": {
            "match_all": {
              "boost": 1
            }
          }
        }
```

Threshold Conditions

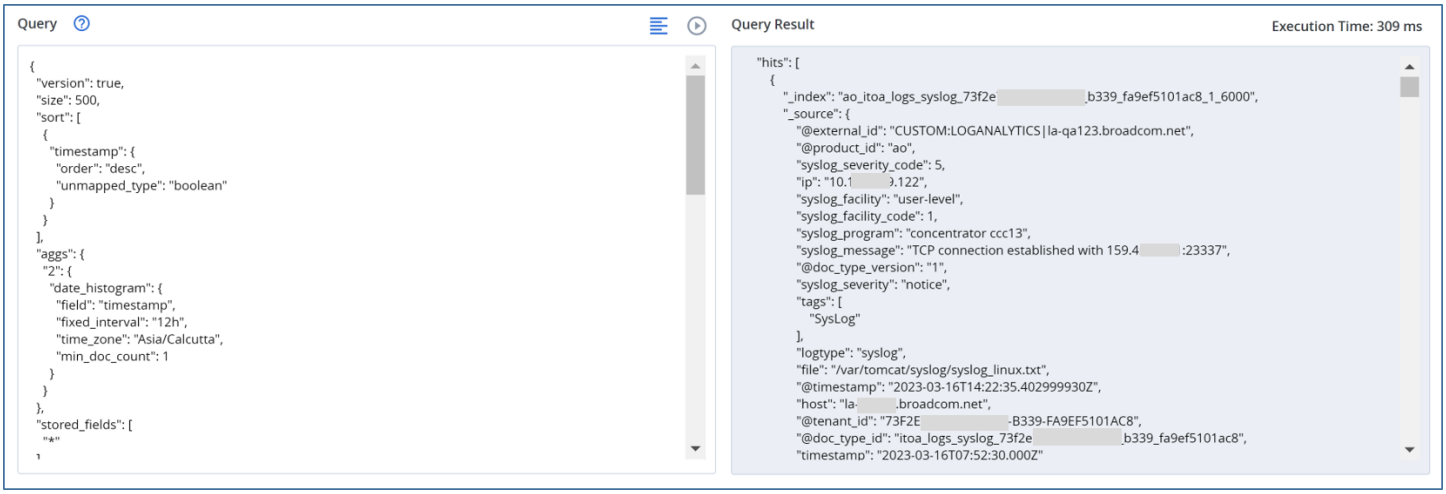
Raise an alarm when the CPU value is more than 90:

```
(ctx.results[0].aggregations.avg_cpu.value > 90) {
  return true;
}
```

Configure Alarm Message

You can configure the alarm message by including the log attributes in the message. A few examples of log attributes are the host, IP address, message, and so on. The configured alarm message is displayed as part of the generated alarm on the Alarm Analytics page of DX Operational Intelligence. The administrators and the support personnel use the log attributes to triage and resolve the issues quickly.

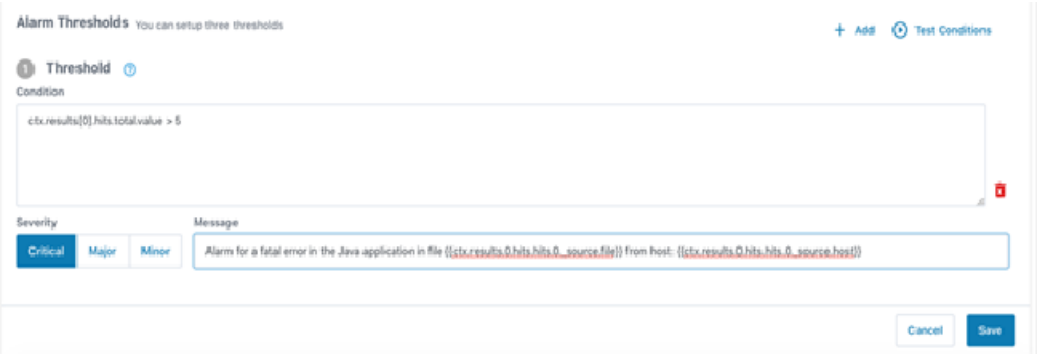
To return the log attributes in the Query Result, execute the query in the alarm definition. The following screenshot displays the configured query for the log alarm and the query result on execution.



To include the log attributes in the message, use the following variable format in the Message configuration text area:

```
{{ctx.results.0.hits.hits.0._source.<attribute name> }}
```

These variables can be combined along with the static text in the message to make the alarm message both meaningful and useful. The following example in the screenshot depicts the same.



Few more examples for accessing other fields/attributes in the Alarm Message are as follows:

severity: `{{ctx.results.0.hits.hits.0._source.severity}}`

ip: `{{ctx.results.0.hits.hits.0._source.ip}}`

container_name: `{{ctx.results.0.hits.hits.0._source.container_name }}`

```
container_id: {{ctx.results.0.hits.hits.0._source.container_id}}
```

You can access any log attribute for a log type in the alarm message.

> **NOTE**
> You must ensure that the attribute name is the same as what you see in the query result.

Manage Log Alarm Definitions

After you create alarm definitions, you can perform the following actions on the alarm definition:

- View and Edit Alarm Definition
- Delete Alarm Definition

View and Edit Log Alarm Definition

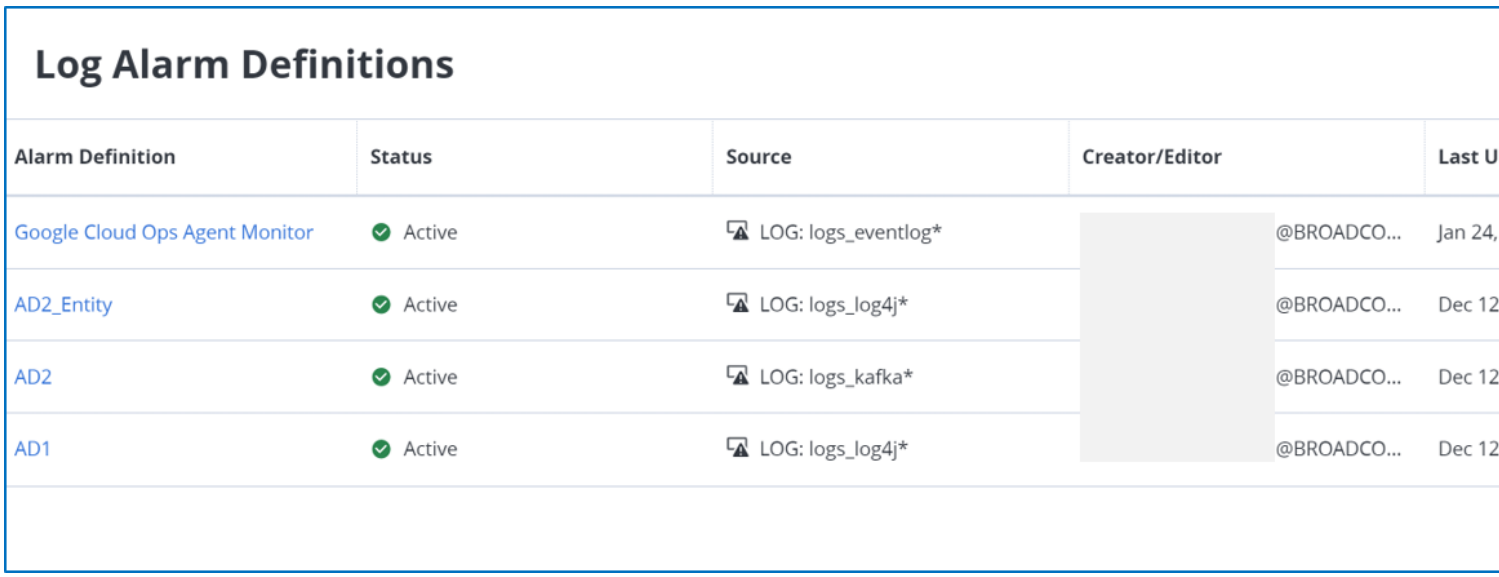
You can edit the log alarm definition as per your requirements. However, you cannot edit the alarm type after you define and save the alarm definition. If you want to use a different alarm type, then you must create an alarm definition with the same threshold conditions and select the relevant alarm type.

> **NOTE**
> After you make any changes to the query, test the query to validate the changes. The changes can be saved only if the validation is successful.

Follow these steps:

1. Log in to DX Operational Intelligence and navigate to the **Settings** Page.
2. Click **Define** in the **Define Log Alarms** tile.

   The application displays all the log alarm definitions.



3. (View the Log Alarm Definition) Click the alarm definition name that you want to view,

   The application displays the View <Alarm Definition Name> page with alarm definition details.
4. (Edit the Log Alarm Definition) Click **Edit** to edit the log alarm definition. Alternatively, you can click **Edit** under **Actions** on the **Log Alarm Definitions** page.
5. Make the necessary changes and click **Save**.

   The application updates the log alarm definition with the changes.

Delete Alarm Definition

You can delete the alarm definition based on your requirements.

Follow these steps:

1. Navigate to the Alarm Definitions page.
2. Click **Delete** in the Actions column.

3. Click **Yes** to confirm the deletion.
   The application deletes the alarm definition.

Access and View Log Alarms in Alarm Analytics

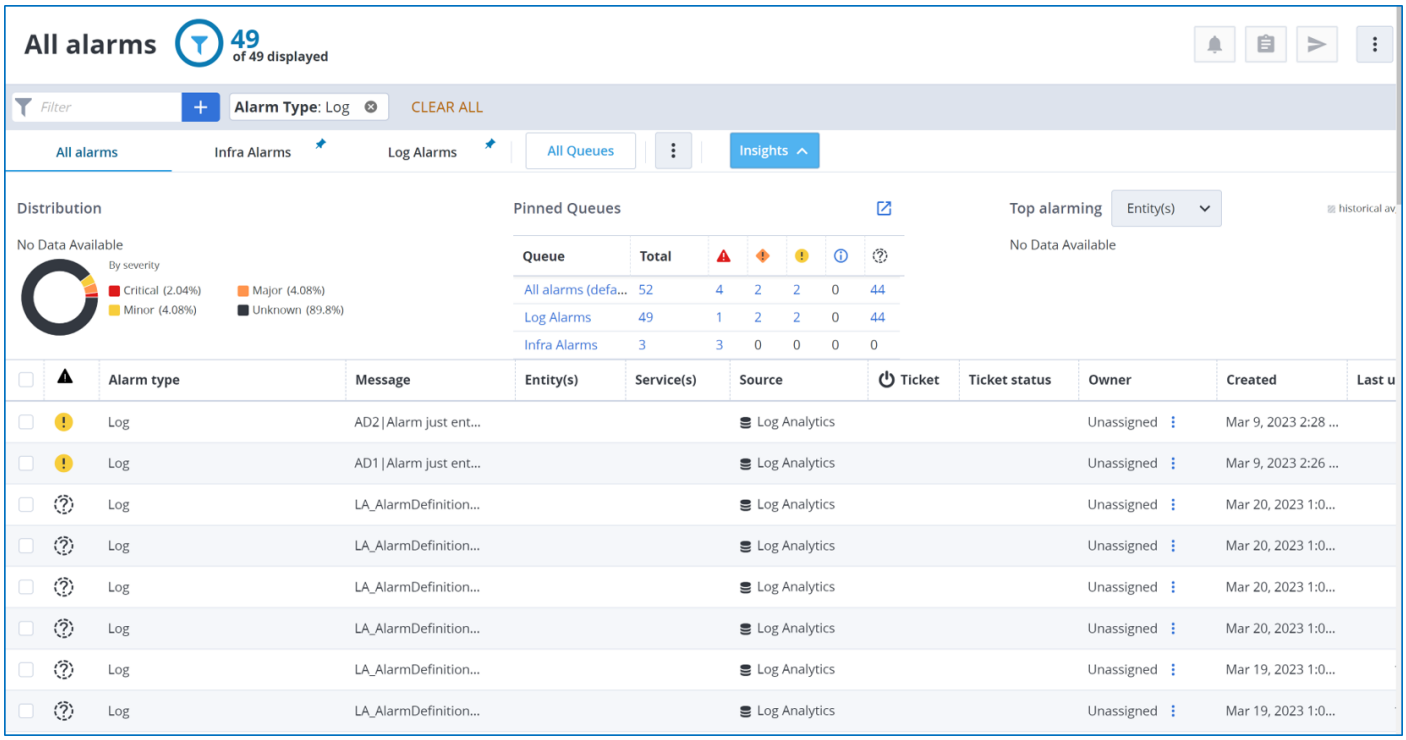
When an alarm definition generates an alarm, you can access and view the alarm details in Alarm Analytics and take necessary actions.

Follow these steps:

1. Log in to DX Operational Intelligence.
2. Select **Alarms** in the left navigation pane.
   The Service Alarms page opens.

   NOTE
   Ensure you are in the **All Alarms** view.
3. Create the filter criteria. In the filter, ensure to select the Alarm Type as **Log**.

   The application displays the log alarms.



Log-Based Triaging for Alarms and Monitored Inventory

DX Operational Intelligence can also help you get more context on the alarms and observe the logs in the context of the registered devices or hosts. This is done when **Logs for OI** (Log Analytics) is enabled for the tenant and the tenant is actively ingesting the logs for those hosts using the standard log ingestion components shipped or distributed by DX Operational Intelligence.

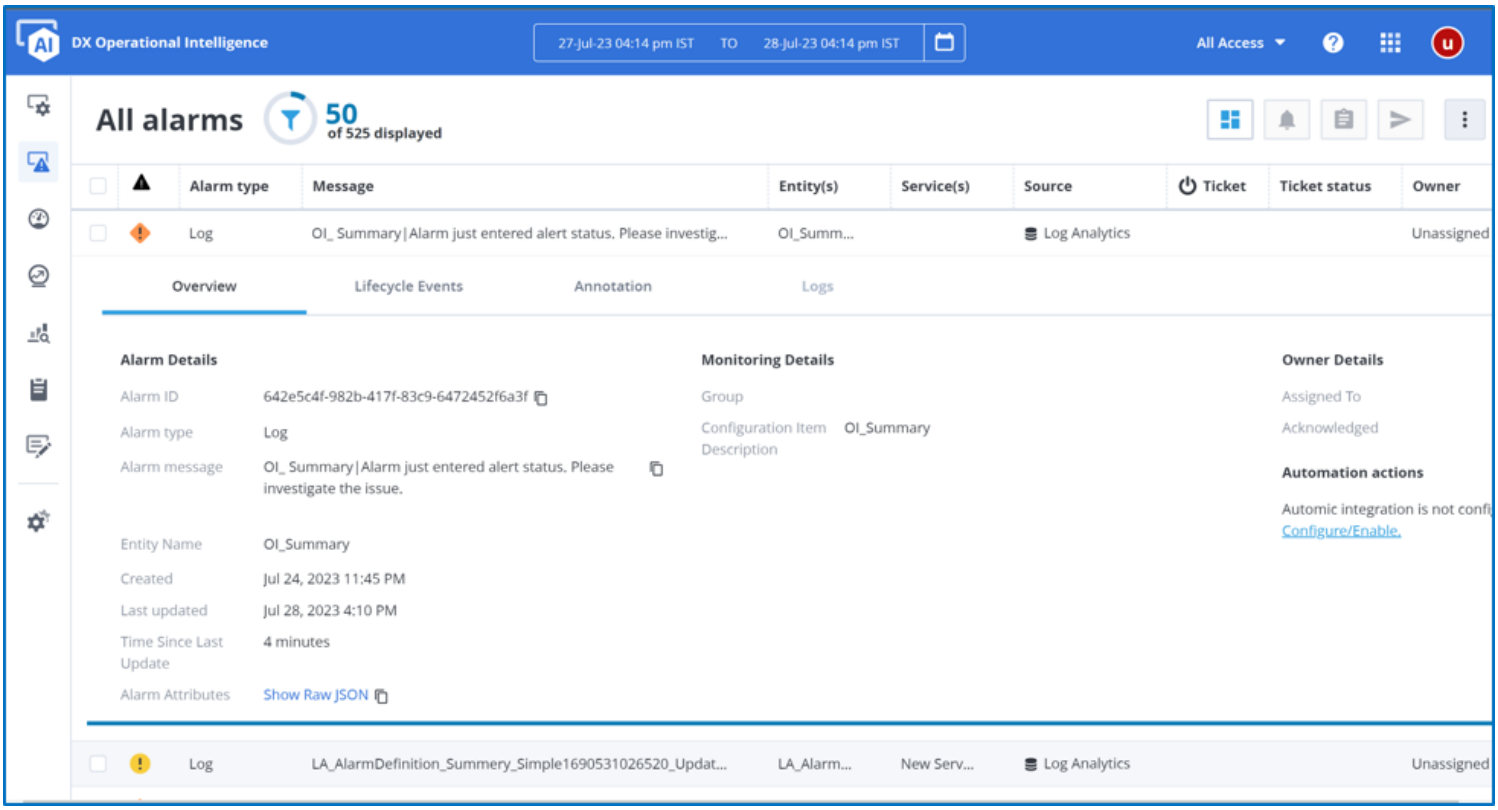
> **NOTE**
> **Prerequisite:** Ensure that the hostname is being sent by the log collection agents (Filebeat, Winlogbeat). The hostname is a Fully Qualified Domain Name (FQDN) for the co-relation to work correctly.

All Alarms

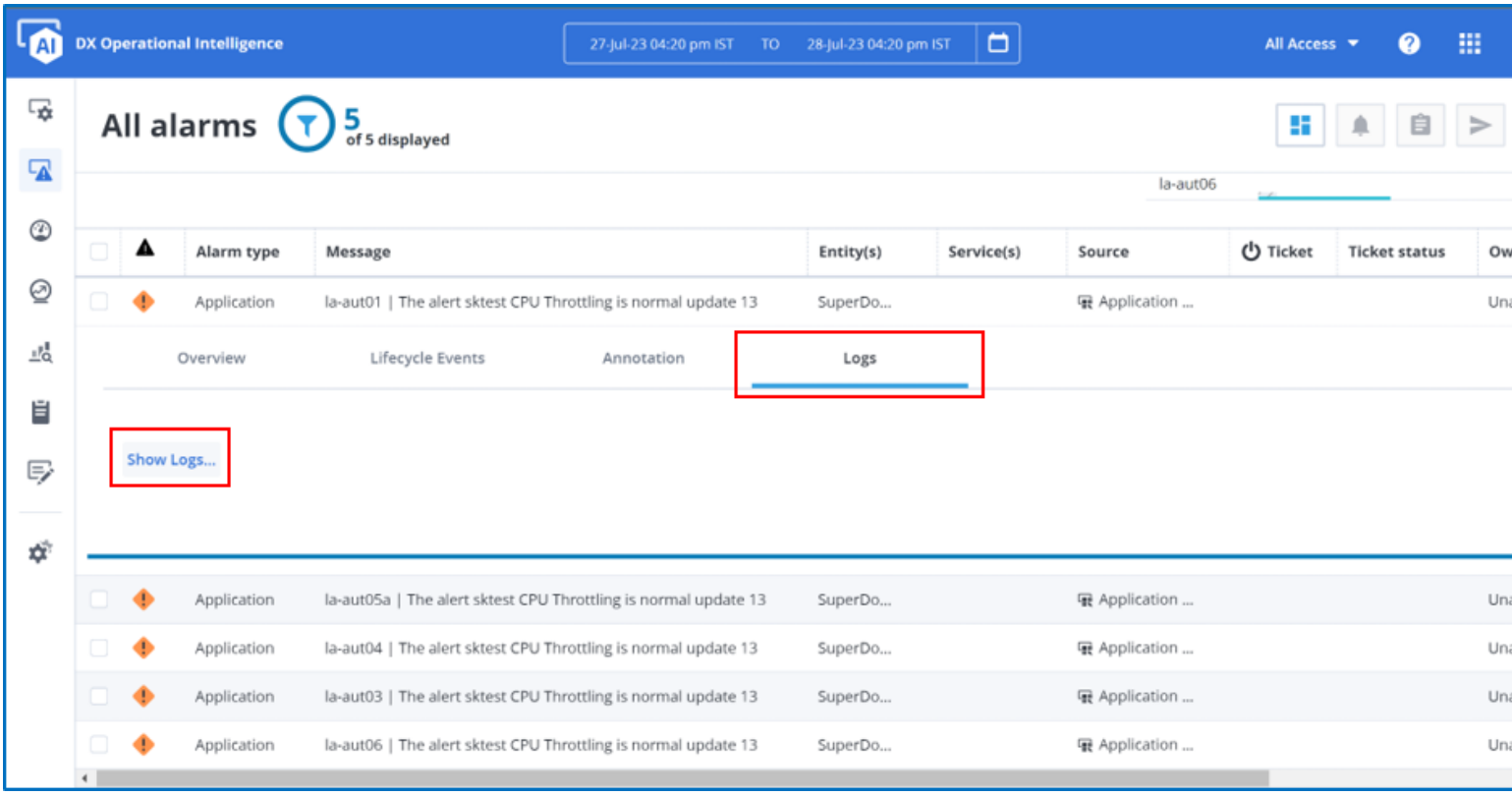
When an alarm is raised in DX Operational Intelligence by a source product for a particular host, you can launch the logs in the context of alarms in the OpenSearch dashboards for all the log types from the All Alarms page. This helps to narrow down the possible root cause of the raised alarm. The context is set by displaying the logs from 15 minutes prior to the creation time of the alarm and the opening of the dashboards for logs. You can subsequently change the filter criteria in the dashboard for further troubleshooting.

The **Logs** tab for a specific alarm on the **All Alarms** page is enabled only if DX OI Logs is enabled and the host (with FQDN) gets correlated with the other source products in DX Operational Intelligence.

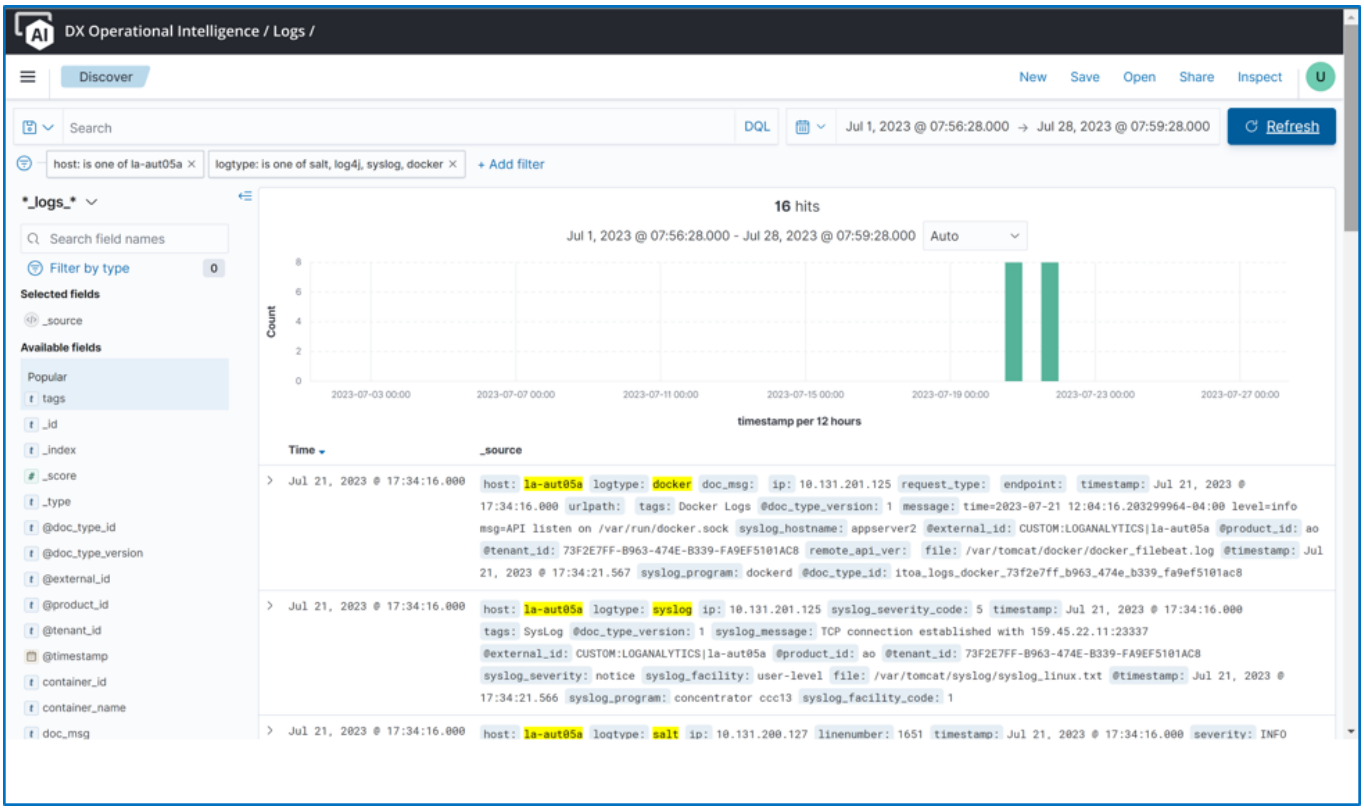
The following image illustrates the Logs tab being disabled when there is no correlated entity for a host:



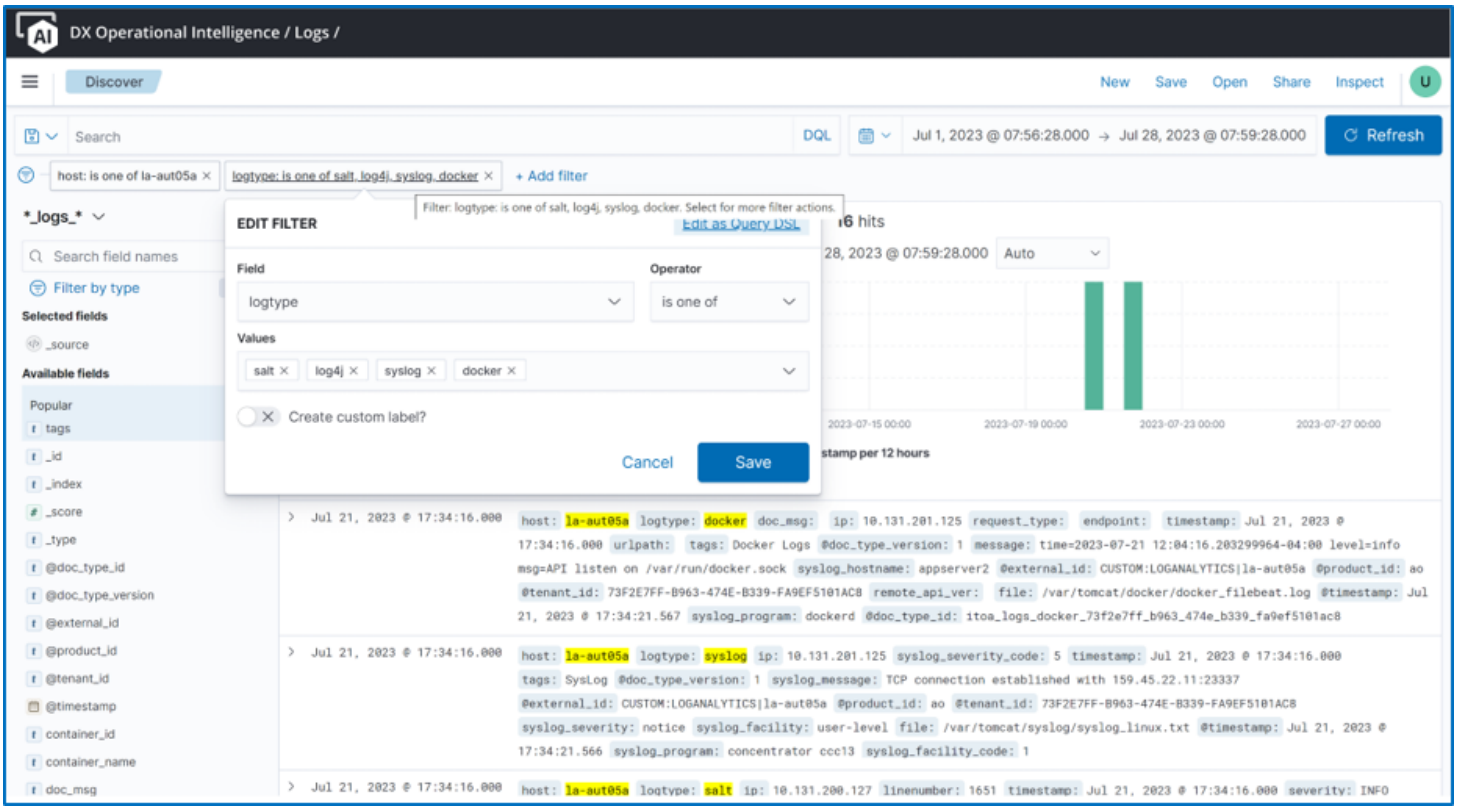
The following image illustrates when there is a correlated entity for a host.



Click **Show Logs** to navigate to the **DX OI - Logs** page to visualize the logs for the host from when the alarm was created. The following image illustrates the Logs dashboard when launched with the alarms context and pre-filled filters:



You can subsequently change the filter criteria in the dashboard for further troubleshooting.

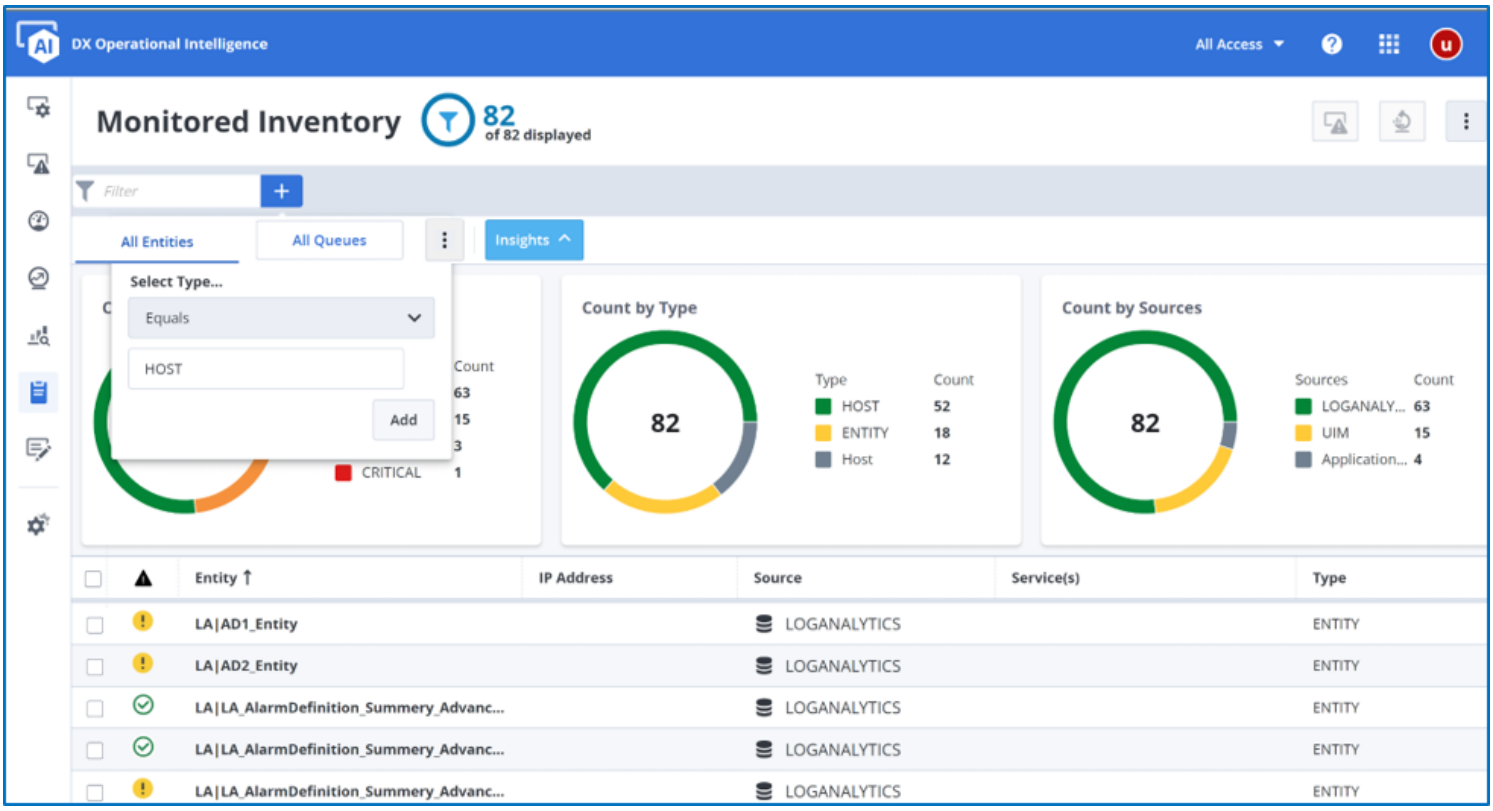


NOTE

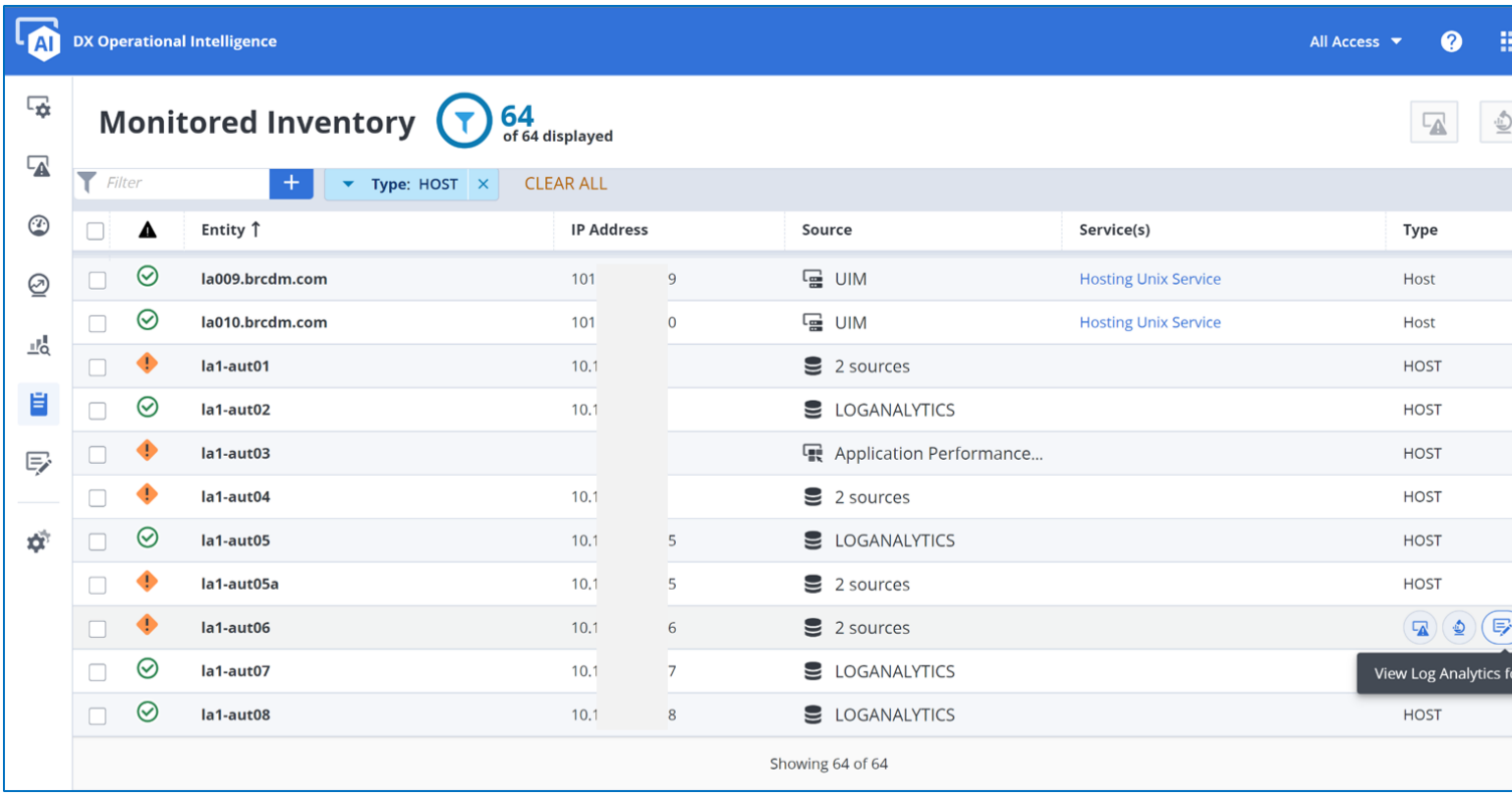
- You can launch the logs only for alarms coming from the source products such as Spectrum, UIM, and so on but not for log-based alarms.
- If the host has multiple log types, then the **ao_itao_logs_*_<*>** index pattern is used to query all the log types by default.

Monitored Inventory

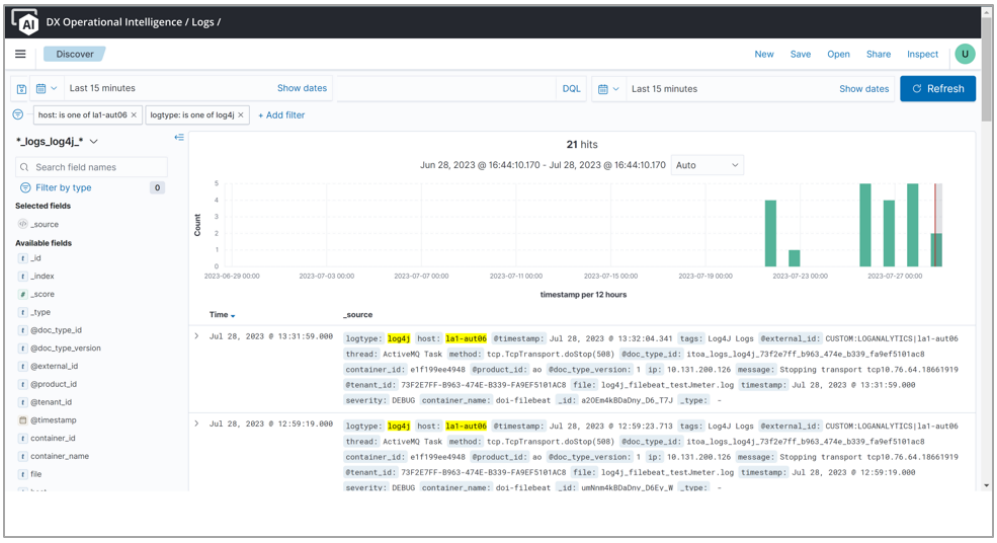
You can view logs for an entity or CI from the **Monitored Inventory** page, when **Log Analytics** is one of the source products and the entity or CI is of the type **HOST**.



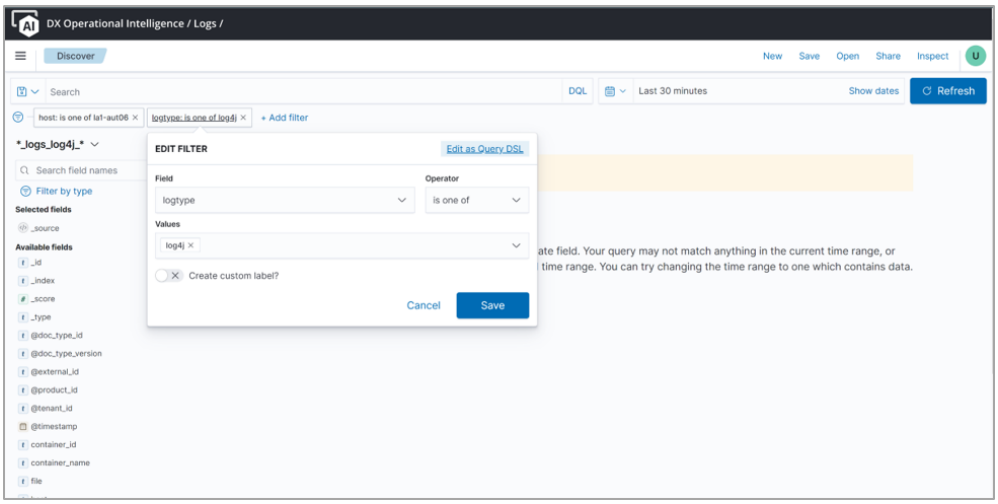
You can also launch logs in the context of monitored inventory for all the log types which help in checking the logs that were being generated for the last 15 minutes for a given hostname. If Log Analytics is present for an entity, the **View Log Analytics** icon is displayed under the **Type** column. Hover over the entity for this icon to be displayed. This icon is not available if the entity is not of the type **HOST** or Log Analytics is not in the source product list.



Click the **View Log Analytics** icon to navigate to the **DX OI - Logs** dashboard to check the logs for the last 15 minutes for a given hostname. You can view logs for all the log types and multiple inventory items.



You can subsequently change the filter criteria in the dashboard for further troubleshooting.



NOTE

- If the host has multiple log types, then the **ao_itao_logs_*_<*>** index pattern is used to query all the log types.
- Log-based triaging is not supported for custom logs ingested for the host. That is, the custom logs are not shown either in the context of alarms or monitored inventory.
- Logs ingested for the host must have FQDN for them to participate in the host-based co-relation (log enrichment) and thus have the logs shown in the alarm and monitored inventory context.

Log Ingestion Throttling

Throttling is generally a defensive measure for shared services to protect themselves from excessive use whether intended or unintended to maintain the service availability. Log Ingestion throttling helps you to manage your storage costs by predefining the log volume that you want to ingest. Using log ingestion throttling, you can throttle the ingestion of log data when the volume reaches the configured ingestion limit (hard limit) threshold for your tenant account.

DX Operational Intelligence imposes caps on the log ingestion data that the log collectors can ingest per day for a tenant account. The caps are defined as per the communication between the tenant and Broadcom considering your current consumption and requirements.

DX Operational Intelligence enables you to define the following caps on the log data that you can ingest into the DX Operational Intelligence data lake:

Soft Limit	A soft limit allows a particular tenant to exceed the limits for a short period or allows additional log volume. The cap for soft limit is the maximum log volume (in GB) that the log collectors can ingest per day for a tenant. When the soft limit is exceeded, a soft limit breached alarm is raised in DX Operational Intelligence but the ingestion is not stopped.
Hard Limit	A hard limit forces the immediate dropping of log messages. The cap for hard limit is 15% more than the defined soft limit log volume. When the hard limit is exceeded, a hard limit breached alarm is raised and the ingestion stops.

This section provides the following information:

- How Does the Log Ingestion Throttling Work
- Enable Notifications for Throttling Alarms
- View Throttling Alarms

How does the Log Ingestion Throttling Works

After the soft and hard limits are configured, DX Operational Intelligence starts monitoring the volume of log data that the log collectors are ingesting for your tenant account.

- When the log ingestion volumes are within the soft limit, DX Operational Intelligence does not impact the log ingestion process. The application resets the counter at the end of the day (12:00 AM) and restarts the log ingestion process for the next day (12:01 AM).
- When the log ingestion volume reaches the 80% mark, DX Operational Intelligence raises the first alarm in DX Operational Intelligence and sends a notification to the concerned personnel.
- When the ingestion volume breaches the soft limit, the application updates the original alarm and sends a notification to the concerned personnel.
- When the ingestion volume breaches the hard limit, the application raises a second alarm and sends a notification to the concerned personnel. The application throttles the log ingestion and restarts the log Ingestion process the next day (12:01 AM).

  NOTE
  When the application aborts the log ingestion upon exceeding the hard limit, you can still access the OpenSearch dashboards and search and filter the logs. You can also configure the log-based alarms for the ingested logs.

Enable Notifications for Throttling Alarms

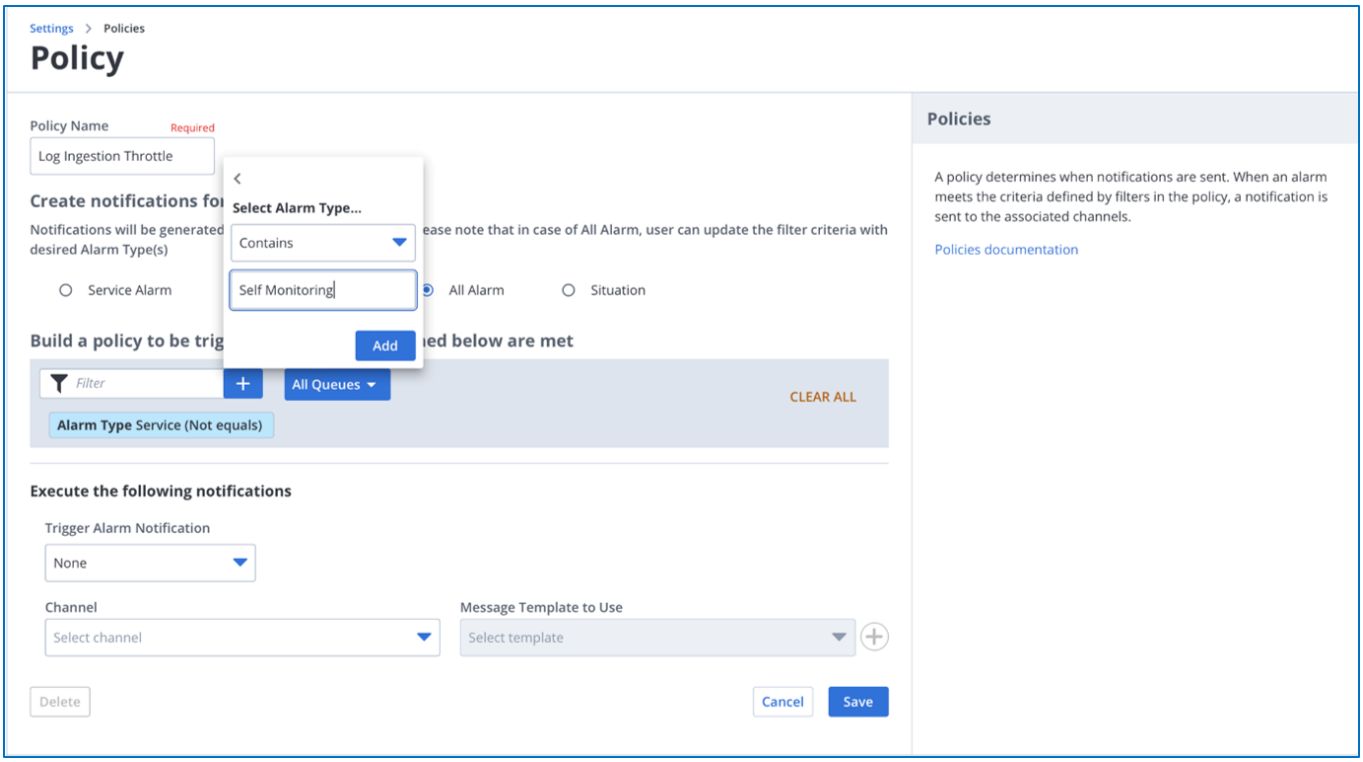
As a tenant administrator, you must enable email notifications for log ingestion throttling alarms by configuring the notification channel and a policy. DX Operational Intelligence sends a notification alert to the concerned personnel when the log ingestion throttle alarms are raised.

Follow these steps:

1. Create or update an existing Email notification channel with relevant email IDs.

   For more information, see the Configure Email Notification Channel section.

2. Create a policy for All Alarms and define the following policy trigger rule:
   a) Select **Alarm type** as **All Alarms**.
   b) Select the filter attribute as **Alarm Type**.
   c) Select the filter operator as **Contains**.
   d) Enter the value as **Self Monitoring**.



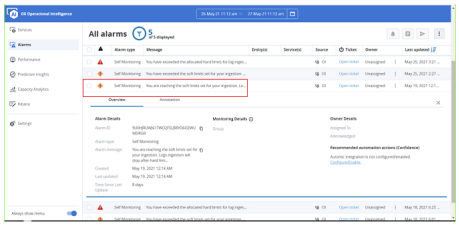
   e) Click **Add**.
   f) Select the channel and the message template to use.
   g) Click **Save**.

   The application creates the policy with a trigger rule to generate the log ingestion throttling alarms.

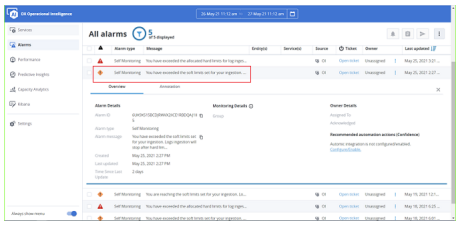
View Throttling Alarms

DX Operational Intelligence raises two alarms when the log ingestion volume reaches the following log ingestion limits:

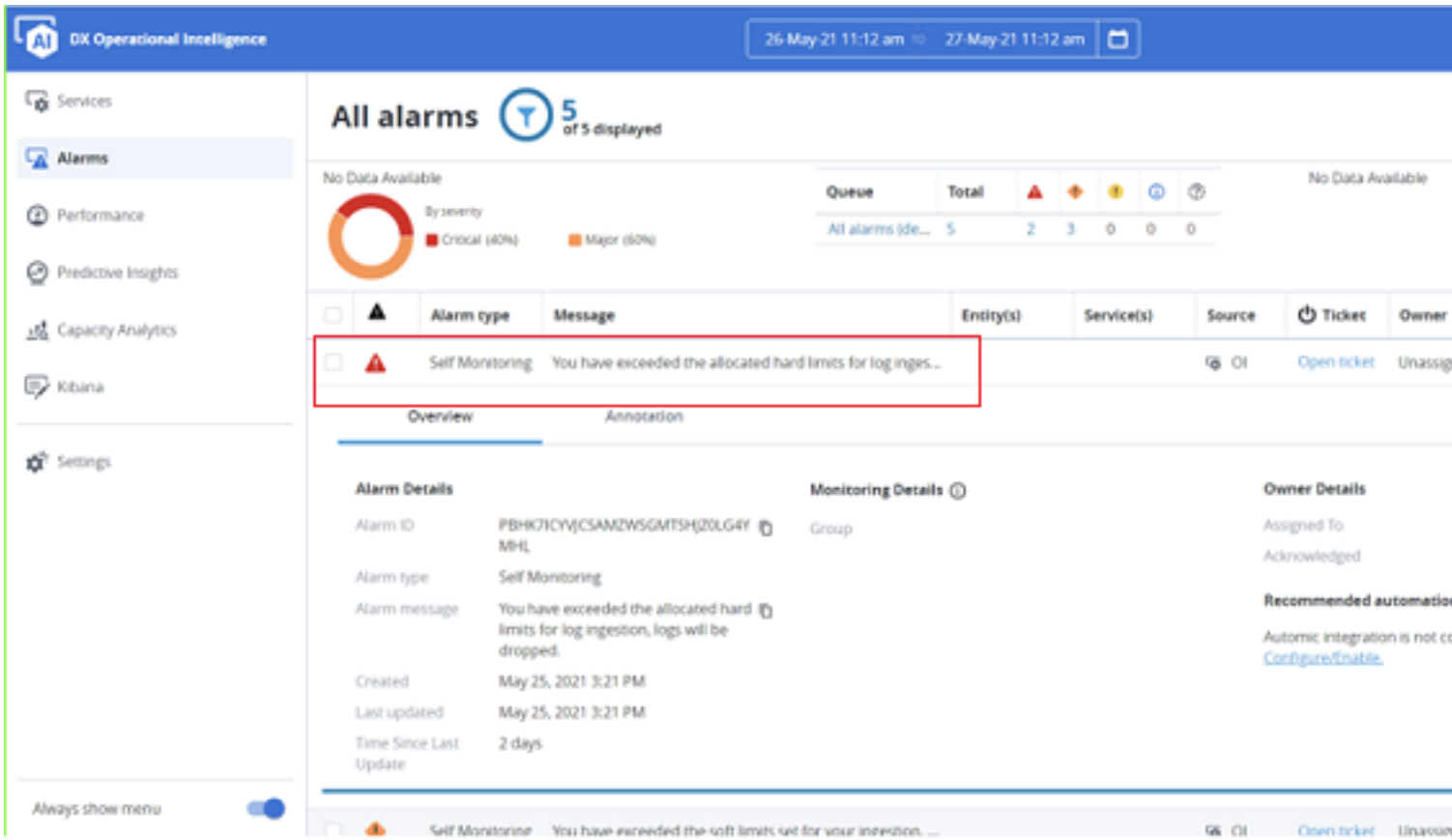
- The log ingestion volume reaches 80% of the defined soft limit.



- The log ingestion volume breaches the defined soft limit.



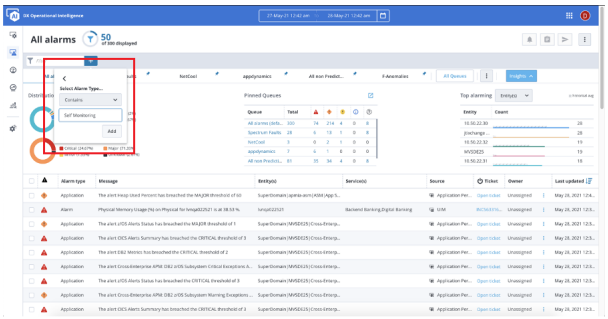
- The log ingestion volume exceeds the defined soft limit by 15% (hard limit).



Follow these steps:

1. Click **Alarm Analytics** in the left navigation menu.
2. Click

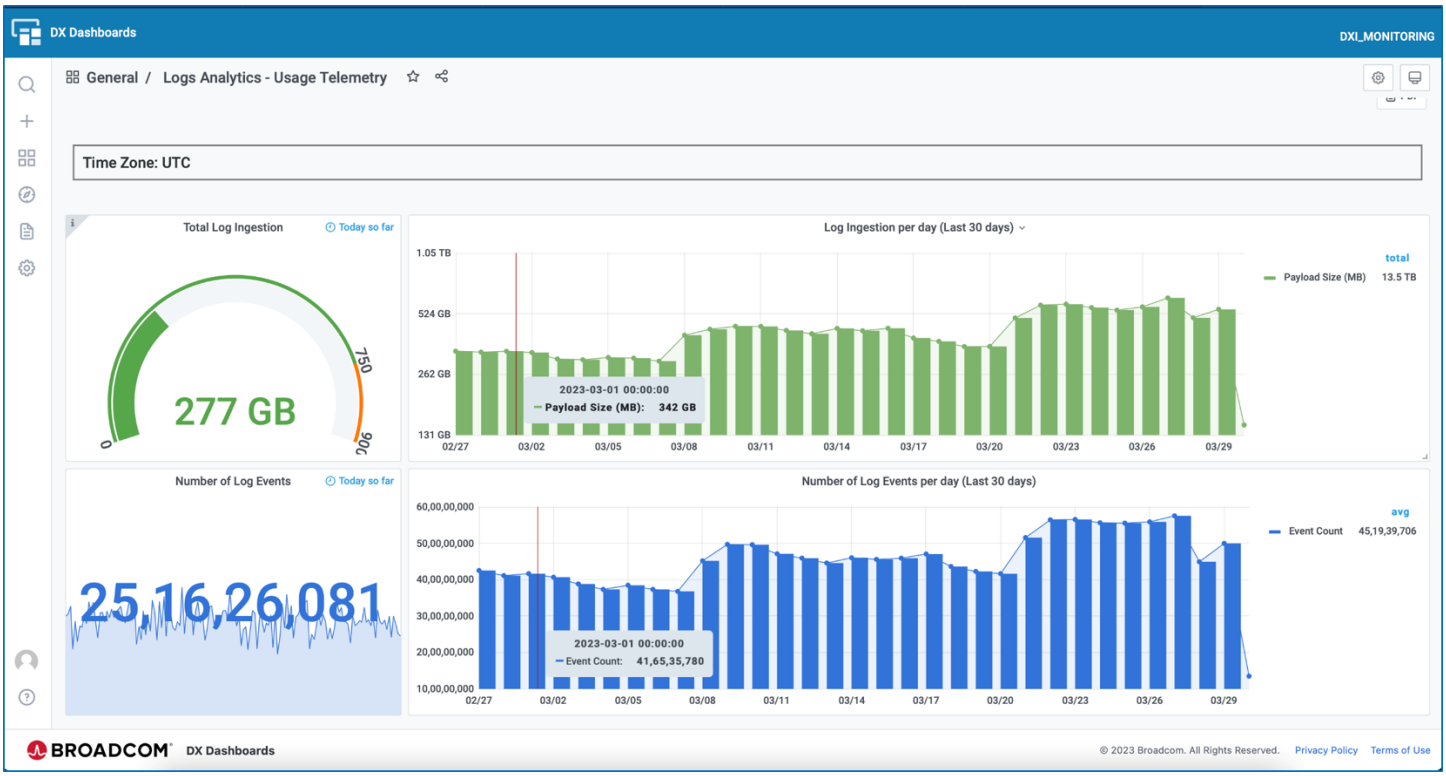
   and select **All Alarms** from the drop-down.
3. Define the following filter criteria:
   a) Select the filter attribute as **Alarm Type**.
   b) Select the filter operator as **Contains**.
   c) Enter the value as **Self Monitoring**.

   The application lists the alarms related to log ingestion throttling.

Logs Analytics - Usage Telemetry Dashboard

The **Logs Analytics - Usage Telemetry** dashboard is available in DX Dashboards. This dashboard includes the following visualizations:



NOTE
The Stat and Gauge panels display as zero if no data is available.

Visualization Name	Description
Time Zone	Displays the time zone.
Total Log Ingestion (Today so far)	Displays the total log ingestion volume of that day. This visualization displays the following limits: • **Soft Limit:** A soft limit allows a particular tenant to exceed the limits for a short period or allows additional log volume. The cap for soft limit is the maximum log volume (in GB) that the Log Collectors can ingest per day for a tenant. When the soft limit is exceeded, a soft limit breached alarm is raised in DX Operational Intelligence but the ingestion is not stopped. • **Hard Limit:** A hard limit forces the immediate dropping of log messages. The cap for hard limit is 15% more than the defined soft limit log volume. When the hard limit is exceeded, a hard limit breached alarm is raised and the ingestion stops.
Log Ingestion Per Day (Last 30 days)	Displays the total log ingestion volume per day for the last 30 days.
Number of Log Events (Today so far)	Displays the total number of events of that day.
Number of Log Events Per Day (Last 30 days)	Displays the number of events per day for the last 30 days.

DX OI - Logs APIs

This section describes the following APIs:

- Purge or Delete Logs
- Log Availability Status APIs
- Log Archival and Retrieval
- Log Events APIs

Purge or Delete Logs

DX Operational Intelligence provides the purge APIs to perform selective purging of log data based on a search criteria and the date range.

Using Purge APIs, you can process the following requests:

- Get the log documents count based on the search criteria, date range and log type.
- Submit a purge request to delete the log documents based on the search criteria, date range and log type.
- Get status of a purge request using task ID.
- Get active purge requests task list.

The Purge APIs are:

- Get Count API
- Delete API
- Get Task Status
- Get Active Status

Supported Time Stamp Formats

Purge APIs supports the following time formats:

- UTC time format in epoch_miils.
- strict_date_optional_time in yyyy-MM-dd'T'HH:mm:ss.SSSX format.

Get Count API

The Get Count API returns the number of log documents based on the specified search criteria, date range, and the log type from the OpenSearch cluster.

Resource URI

```
https://<oi_host>:<oi_port>/oi/v2/api/la/purge/_count?q=*&timefrom=<Time From>&timeto=<Time To>&logtype=<Log
 Type>&iscustom=<Boolean>
```

Table 8: Query Parameters

Parameter	Description
q	Requests the log documents to be filtered based on the specified query in the Lucene syntax.
timefrom and timeto	Requests the log documents count with the specified date range. Ensure that you use the same timestamp format that is updated for the log type.
Log type	Requests the log documents of the specified log type. For example, syslog, kafka, generic, apache_access,log4j. In the case of custom logs, give the custom name of the index and set **iscustom** to **true**.

Parameter	Description
(Optional) iscustom	Sends a boolean value that indicates whether the log type is custom. Default: false

Method

```
GET
```

HTTP Headers

- Authorization: Bearer {token}
  For more information on generating the token, see Authentication and Authorization of APIs.

Response Syntax

```
{
"code": <http success of failure code>,
"message": <"success or failure message">,
"response": { "count": <"number of log documents count"> }
}
```

Sample Request-Response

Sample URI

```
http://adminui.10.00.0.nip.io/oi/v2/api/la/purge/_count?
q=*&timefrom=1624441503&timeto=1624442503&logtype=custom&iscustom=true
```

Sample Response

```
{
"code": 200,
    "message": "OK",
    "response": {"count" : "89000"}
}
```

Delete API

The Delete API sends request to purge the log documents from the OpenSearch cluster based on the log type, search criteria, and the date range. DX Operational Intelligence processes the deletion of the log documents if the log documents count configured is within the maximum limit.

When you invoke a purge request, using the Delete API, the DX Operational Intelligence creates a task ID for the purge request.

DX Operational Intelligence also ensures that the number of requests the API sends in an hour is within the maximum purge requests limit configured in the tenant. The application uses the **last_execution_time** and **count** attributes to keep a count on number of purge requests being processed in an hour.

- **Max Purge Doc Limit:** Contains the maximum number of log documents count that the Delete API purges in a single request for a tenant. **Default Value:** 100000
- **Maximum Purge Request:** Contains the maximum number of purge requests that the Delete API processes within an hour for a particular tenant. **Default Value**: 5

> **NOTE**
> The “count” and the "last_execution_time" is reset every hour.

Resource URI

```
https://<oi_host>:<oi_port>/oi/v2/api/la/purge/_delete
```

Method

```
POST
```

HTTP Headers

- Authorization: Bearer {token}
  For more information on generating the token, see Authentication and Authorization of APIs.
- Content Type: Application/JSON

Request Payload Syntax

```
{
"q": <"query">,
"timefrom": <"time stamp>",
"timeto": <"time stamp>",
"logtype": "<log type>",
"iscustom": "<boolean>"
}
```

Table 9: Parameters

Parameter	Description
Q	Requests the log documents based on the specified search criteria.
timefrom and timeto	Requests the log documents with in the specified date range. Ensure that you use the same time stamp format that is updated for log type.
Logtype	Requests the log documents of the specified log type.
(Optional) iscustom	Sends a boolean value that Indicates if the log type is custom. Default: false

Response Syntax

```
{
    "code": 200,
    "message": "Deletion Initiated. Use the /_task/<taskId> for tracking the status of the task",
    "response": {
        "taskId" : "<taskID>"
    }
}
```

Sample Request-Response

Sample URI

```
http://adminui.10.00.0.nip.io/ oi/v2/api/la/purge/_delete
```

Sample Payload

```
{
"q": "*",
"timefrom": "1624441503",
```

```
"timeto": "1624441503",
"logtype": "custom",
"iscustom": "true"
}
```

Sample Response

```
{
    "code": 200,
    "message": "Deletion Initiated. Use the /_task/<taskId> for tracking the status of the task",
    "response": {
        "taskId" : "yJoftkukQSCLGS9tigeFqw:202536406"
    }
}
```

Get Task Status API

The Get Start Status API returns the status of a specified task. DX Operational Intelligencereturns a task ID for each purge request sent using the Delete API. You can get the status of the purge request by passing the task ID in the Get Task Status API.

Resource URI

```
https://<doi-adminui>/oi/v2/api/la/purge/_status/{taskId}
```

Method

```
GET
```

HTTP Headers

- Authorization: Bearer {token}
  For more information on tokens, see the Token Management section.
- Content Type: Application/JSON

Request Payload Syntax

```
{taskId}
```

Table 10: Parameters

Parameter	Description
taskid	Requests the status of the specified task ID.

Response Syntax

```
{
    "code": 200,
    "message": "OK",
    "response": {
        "completed": <"Task Status">,
        "deleted" : <number of deleted log documents>
    }
}
```

Sample Request-Response

Sample URI

```
https://doi-adminui.10.00.0.nip.io/oi/v2/api/la/purge/_status/4edx51W9Qt-c1gsJeoTzdA:21895555
```

Sample Response

```
{
    "code": 200,
    "message": "OK",
    "response": {
        "completed": true",
        "deleted" : 89000
    }
}
```

Log Events APIs

DX Operational Intelligence enables you to fetch the log events data using Log Event APIs. Using these APIs you can fetch, the log events(alarms), raw logs, alarm definitions, and the query defined for alarm definitions:

- Log Events Query API
- Event Matching Raw Logs API
- Alarm Definition API
- Alarm Definition Get Query API

Log Event Query API

Log Event API fetches the list of matching log events(Alarms) and the associated alarm definition details for a specified time period. You can also use the API to return the total number of events count for the specified period. The events can be from one or more log alarm definitions for a given tenant. You can pass the tenant information using Gateway token specific to the tenant.

DX Operational Intelligence uses the Log Analytics Meta Data Index for the tenant information for processing this request.

Resource URI

```
http://<apmservices-endpoint>/oi/v3/oipublic/la/events?period_start=<Period Start Date and
 Time>>&period_end=<Period End Date and Time> &size=<Size>
```

Table 11: Query Parameters

Required /Optional	Parameter	Description	Default Value
Required	period_start	Specifies the start date and time of the period from when you want to fetch the matching log events. You must specify the time stamp using UTC standard and in the epoch_mills format: yyyy-MM-dd'T'HH:mm:ss	NA
Required	period_end	Specifies the end date and time of the period you want to fetch the matching log events. You must specify the time stamp using UTC standard and in the epoch_mills format: yyyy-MM-dd'T'HH:mm:ss	NA
Optional	size	Specifies the maximum number of records that must be returned Maximum Limit = 10000	1000

Required /Optional	Parameter	Description	Default Value
Optional	countOnly	Fetches the total number of log events count for the specified period when this parameter is set to 'true'. If you set the 'countOnly' parameter to true, the API does not fetch the list of log event records, but only the total number of log events count.	false
Required	fetchAfter	Fetches the log events records after reaching specified size limit. FetchAfter is not used for the first request. the parameter is required only for the subsequent pagination.	

Method

```
GET
```

HTTP Headers

- APM Gateway {token}
  For more information on generating the token, see APM Tenant Token.
- Content Type: Application/JSON

Response Syntax

```
{
    "code": 200,
    "response": {
        "fetchAfter": "1620212827223-124",
        "hits": [
            {
            "hostname": "<Host Name>",
            "count": <Count>,
            "alarm_definition": "<Alarm Definition>",
            "period_start": "<Period Start Date in epoch mills format>",
            "period_end": "<Period End Date in epoch mills format>",
            "severity": "<Alarm Severity>"
            },
        {
            "hostname": "<Host Name>",
            "count": <Count>,
            "alarm_definition": "<Alarm Definition>",
            "period_start": "<Period Start Date in epoch mills format>",
            "period_end": "<Period End Date in epoch mills format>",
            "severity": "<Alarm Severity>"
            },
        ]
    }
}
```

Response Interpretation

If there are no events for the specified period:

```
{
    "code": 200,
    "response": {
```

```
        "fetchAfter": "",
        "hits": []
    }
}
```

If a mandatory Input parameter is missed while sending the API request:

```
{
    "code": 400,
    "message": "Invalid.missing input parameters PeriodStart/getPeriodEnd"
}
```

Sample Request-Response

Sample URI

```
http://<apmservices-endpoint>/oi/v3/oipublic/la/events?
period_start=2021-10-01T01:00:01&period_end=2022-01-05T19:00:01&size=2000
```

Sample Response

```
{
    "code": 200,
    "response": {
        "fetchAfter": "1641292800249-1641283872",
        "hits": [
            {
                "hostname": "la-qa03",
                "count": 6,
                "alarm_definition": "LA3",
                "period_start": "2022-01-04T10:35:00.249Z",
                "period_end": "2022-01-04T10:40:00.249Z",
                "severity": "MAJOR"
            },
            {
                "hostname": "la-qa04",
                "count": 6,
                "alarm_definition": "LA3",
                "period_start": "2022-01-04T10:35:00.249Z",
                "period_end": "2022-01-04T10:40:00.249Z",
                "severity": "MAJOR"
            }
        ]
    }
}
```

Event Matching Raw Logs API

You can fetch the raw logs from the log analytics clusters using Event Matching Raw Logs API. The raw logs that are fetched are for the specified alarm definition and the host. You can pass the tenant information using Gateway token specific to the tenant.

DX Operational Intelligence uses the Log Analytics Meta Data Index for the tenant information for processing this request.

Resource URI

```
http://<apmservices-endpoint>/oipublic/la/events/raw_logs?period_start=<Period Start time
 stamp>&period_end=<Period End Time Stamp>&alarm_definition=<Alarm Definition>&hostname=<Host
 Name>&custom=<Boolean>
```

Table 12: Query Parameters

Required /Optional	Parameter	Description	Default Value
Required	period_start	Specifies the start date and time of the period from when you want to fetch the raw logs. You must specify the time stamp using UTC standard in the epoch_mills format: yyyy-MM-dd'T'HH:mm:ss	
Required	period_end	Specifies the end date and time of the period you want to fetch the raw logs. You must specify the time stamp using UTC standard in the epoch_mills format: yyyy-MM-dd'T'HH:mm:ss	
Required	alarm_definition	Specifies the alarm definition. The API fetches raw logs for events that occurred based on the specified alarm definition.	1000
Required	host	Provides the device name for which you want to fetch the raw logs.	
(Optional)	log type	Specifies the log type of the raw logs. DX Operational Intelligence considers syslog as the log type when no log type is received in the request.	Syslog
(Optional)	custom	Sends a boolean value that Indicates whether the log type is 'custom'.	False

Method

```
GET
```

HTTP Headers

- APM Gateway {token}
  For more information on generating the token, see APM Tenant Token.
- Content Type: Application/JSON

Response Syntax

```
{
    "took": 3,
    "timed_out": false,
    "_shards": {
        "total": 1,
        "successful": 1,
        "skipped": 0,
        "failed": 0
    },
    "hits": {
        "total": {
            "value": 450,
            "relation": "eq"
        },
        "max_score": 1.0,
        "hits": [
```

```
{
    "_index": "<Index Name>",
    "_type": "_doc",
    "_id": "<ID>",
    "_score": 1.0,
    "_source": {
        "@timestamp": "<Timestamp>",
        "@tenant_id": "<Tenant ID>",
        "@doc_type_id": "<Doc Type ID>",
        "host": "<Host Name>",
        "syslog_severity_code": "<Severity Code>",
        "syslog_facility": "authpriv",
        "syslog_severity": "info",
        "syslog_timestamp": "<Timestamp>",
        "timestamp": "<Timestamp>",
        "syslog_priority": "6",
        "@product_id": "ao",
        "syslog_pid": "8905",
        "logtype": "<Log Type>",
        "origin_timestamp": "<Timestamp>",
        "syslog_hostname": "lvnqa025320",
        "received_timestamp": "<Timestamp>",
        "@doc_type_version": "1",
        "syslog_program": "sshd",
        "syslog_message": "pam_unix(sshd:session): session opened for user root by (uid=0)",
        "syslog_facility_code": "10",
        "syslog_pri": "86"
    }
},
{
    "_index": "<Index Name>",
    "_type": "_doc",
    "_id": "<ID>",
    "_score": 1.0,
    "_source": {
        "@timestamp": "<Timestamp>",
        "@tenant_id": "<Tenant ID>",
        "@doc_type_id": "<Doc Type ID>",
        "host": "<Host Name>",
        "syslog_severity_code": "<Severity Code>",
        "syslog_facility": "daemon",
        "syslog_severity": "notice",
        "syslog_timestamp": "<Timestamp>",
        "timestamp": "<Timestamp>",
        "syslog_priority": "5",
        "@product_id": "ao",
        "syslog_pid": "6261",
        "logtype": "<Log Type>",
        "origin_timestamp": "<Timestamp>",
        "syslog_hostname": "lvnqa025320",
        "received_timestamp": "<Timestamp>",
        "@doc_type_version": "1",
        "syslog_program": "dbus",
```

```
                    "syslog_message": "[system] Activating service name='org.freedesktop.problems' (using
 servicehelper)",
                    "syslog_facility_code": "3",
                    "syslog_pri": "29"
                }
            }
        ]
    }
}
```


Response Interpretation

- If a mandatory input parameter is missed or is invalid, the API returns the following response:


```
{
    "code": 400,
    "message": "Invalid/missing input parameter(s)"
}
```


- If the period start and end values are invalid, the API returns the following responses:


```
{
    "code": 400,
    "message": "Invalid input period_start parameter"
}
{
    "code": 400,
    "message": "Invalid input period_end parameter"
}
```


- The API returns the following response when there are no matching logs:


```
{
    "took": 1,
    "timed_out": false,
    "_shards": {
        "total": 1,
        "successful": 1,
        "skipped": 0,
        "failed": 0
    },
    "hits": {
        "total": {
            "value": 0,
            "relation": "eq"
        },
        "max_score": null,
        "hits": []
```

Sample Request-Response

Sample URI

```
http://<apmservices-endpoint>/oipublic/la/events/raw_logs?
period_start=2021-11-14T13:36:24&period_end=2021-12-14T13:36:24&alarm_definition=AD1&hostname=la-
qa01&custom=false
```

Sample Response


```
{
    "took": 3,
    "timed_out": false,
    "_shards": {
        "total": 1,
        "successful": 1,
        "skipped": 0,
        "failed": 0
    },
    "hits": {
        "total": {
            "value": 450,
            "relation": "eq"
        },
        "max_score": 1.0,
        "hits": [
            {
                "_index": "ao_itoa_logs_syslog_184174a0_090c_41fc_9e10_322ec144ef9f_1_1",
                "_type": "_doc",
                "_id": "3MI2uX0BNZ1ZOUqWqz9G",
                "_score": 1.0,
                "_source": {
                    "@timestamp": "2021-12-14T13:50:40.523Z",
                    "@tenant_id": "184174A0-090C-41FC-9E10-322EC144EF9F",
                    "@doc_type_id": "itoa_logs_syslog_184174a0_090c_41fc_9e10_322ec144ef9f",
                    "host": "la-qa01",
                    "syslog_severity_code": "6",
                    "syslog_facility": "authpriv",
                    "syslog_severity": "info",
                    "syslog_timestamp": "2021-12-14T13:36:24.336Z",
                    "timestamp": "2021-12-14T13:36:24.336Z",
                    "syslog_priority": "6",
                    "@product_id": "ao",
                    "syslog_pid": "8905",
                    "logtype": "syslog",
                    "origin_timestamp": "2021-12-14T13:36:24.336Z",
                    "syslog_hostname": "lvnqa025320",
                    "received_timestamp": "2021-12-14T13:50:40.523Z",
                    "@doc_type_version": "1",
                    "syslog_program": "sshd",
                    "syslog_message": "pam_unix(sshd:session): session opened for user root by
 (uid=0)",
                    "syslog_facility_code": "10",
                    "syslog_pri": "86"
                }
```

```
            },
            {
                "_index": "ao_itoa_logs_syslog_184174a0_090c_41fc_9e10_322ec144ef9f_1_1",
                "_type": "_doc",
                "_id": "oMI2uX0BNZ1ZOUqWs0nU",
                "_score": 1.0,
                "_source": {
                    "@timestamp": "2021-12-14T13:50:40.527Z",
                    "@tenant_id": "184174A0-090C-41FC-9E10-322EC144EF9F",
                    "@doc_type_id": "itoa_logs_syslog_184174a0_090c_41fc_9e10_322ec144ef9f",
                    "host": "la-qa01",
                    "syslog_severity_code": "5",
                    "syslog_facility": "daemon",
                    "syslog_severity": "notice",
                    "syslog_timestamp": "2021-12-14T13:36:25.112Z",
                    "timestamp": "2021-12-14T13:36:25.112Z",
                    "syslog_priority": "5",
                    "@product_id": "ao",
                    "syslog_pid": "6261",
                    "logtype": "syslog",
                    "origin_timestamp": "2021-12-14T13:36:25.112Z",
                    "syslog_hostname": "lvnqa025320",
                    "received_timestamp": "2021-12-14T13:50:40.527Z",
                    "@doc_type_version": "1",
                    "syslog_program": "dbus",
                    "syslog_message": "[system] Activating service
 name='org.freedesktop.problems' (using servicehelper)",
                    "syslog_facility_code": "3",
                    "syslog_pri": "29"
                }
            }
        ]
    }
}
```


Alarm Definitions API

The Alarm Definition API returns the list of alarm definitions for the specified log type for a given tenant. You can pass the tenant information using Gateway token specific to the tenant.

Resource URI

```
http://<apmservices-endpoint>/oi/v3/oipublic/la/alarms/alarm_defs?logtype=<Log Type>&custom=fallse
```

Table 13: Query Parameters

Required /Optional	Parameter	Description	Default Value
(Optional)	log type	Specifies the log type for the alarm definition. DX Operational Intelligence considers syslog as the log type when no log type is received in the request.	Syslog
(Optional)	custom	Sends a boolean value that Indicates whether the log type is 'custom'.	False

Method

```
GET
```

HTTP Headers

- APM Gateway {token}
  For more information on generating the token, see APM Tenant Token.
- Content Type: Application/JSON

Response Syntax

```
{
    "code": 200,
    "response": {
        "definitions":
        [
            {
                        "name": "<Alarm Definition 1>",
                        "severity": "major",
                        "enabled": true
            },
            {
                        "name": "<Alarm Definition n>",
                        "severity": "critical",
                        "enabled": false
            }
        ]
    }
}
```

Response Interpretation

- The API returns the following response, when there are no alarm definitions:

```
{
    "code": 200,
    "response": {
        "definitions": []
    }
}
```

Sample Request-Response

Sample URI

```
http://<apmservices-endpoint>/oi/v3/oipublic/la/alarms/alarm_defs?logtype=oracle_alert&custom=true
```

Sample Response

```
{
    "code": 200,
    "response": {
        "definitions":
        [
            {
                        "name": "definition1",
```

```
                            "severity": "major",
                            "enabled": true
                },
                {
                            "name": "definition2",
                            "severity": "critical",
                            "enabled": false
                }
            ]
        }
    }
```

Alarm Definition Get Query API

The Alarm Definition Get Query API returns the defined query of the specified alarm definition for a given tenant. You can pass the tenant information using Gateway token specific to the tenant.

Resource URI

```
http://<apmservices-endpoint>/oi/v3/oipublic/la/alarms/alarm_defs/<Alarm Definition>
```

Table 14: Query Parameters

Required /Optional	Parameter	Description	Default Value
Optional	log type	Specifies the log type for the alarm definition. DX Operational Intelligence considers syslog as the log type when no log type is received in the request.	Syslog
Optional	custom	Sends a boolean value that Indicates whether the log type is 'custom'.	False

Method

```
GET
```

HTTP Headers

- APM Gateway {token}
  For more information on generating the token, see APM Tenant Token.
- Content Type: Application/JSON

Response Syntax

```
{
    "code": 200,
    "response": {
        "query": {
        "bool": {
            "must": [
                {
                    "query_string": {
                        "query": "syslog_message: \"Error\"",
                        "analyze_wildcard": true
                    }
                }
```

```
            ],
            "filter": [
                {
                    "range": {
                        "timestamp": {
                            "gte": "2021-10-21T06:15:57",
                            "lte": "2021-10-21T06:30:57",
                            "format": "strict_date_optional_time"
                        }
                    }
                }
            ],
            "should": [],
            "must_not": []
        }
    }

    }
}
```

Response Interpretation

- The API returns the following response, when there is no matching alarm definition:

```
{
    "code": 400,
    "message": "Could not find alarm definition - test"
}
```

Sample Request-Response

Sample URI

```
http://<apmservices-endpoint>/oi/v3/oipublic/la/alarms/alarm_defs?logtype=oracle_alert&custom=true
```

Sample Response

```
{
    "code": 200,
    "response": {
        "query": {
            "bool": {
                "must": [
                    {
                        "query_string": {
                            "query": "*",
                            "fields": [],
                            "type": "best_fields",
                            "default_operator": "or",
                            "max_determinized_states": 10000,
                            "enable_position_increments": true,
                            "fuzziness": "AUTO",
                            "fuzzy_prefix_length": 0,
                            "fuzzy_max_expansions": 50,
                            "phrase_slop": 0,
                            "escape": false,
```

```
                                "auto_generate_synonyms_phrase_query": true,
                                "fuzzy_transpositions": true,
                                "boost": 1.0
                            }
                        }
                    ],
                    "filter": [
                        {
                            "range": {
                                "timestamp": {
                                    "from": "2021-11-11T09:00:01.946Z",
                                    "to": "2021-11-16T09:00:01.946Z",
                                    "include_lower": true,
                                    "include_upper": true,
                                    "boost": 1.0
                                }
                            }
                        }
                    ],
                    "adjust_pure_negative": true,
                    "boost": 1.0
                }
            },
            "aggregations": {}
        }
    }
```

Troubleshoot DX OI - Logs

Fields for Newly Onboarded Log Type Not Visible

Symptom: After Broadcom onboarded the custom log type, the fields for the newly onboarded log type are not visible in DX OI - Logs.

Solution: Refresh the index.

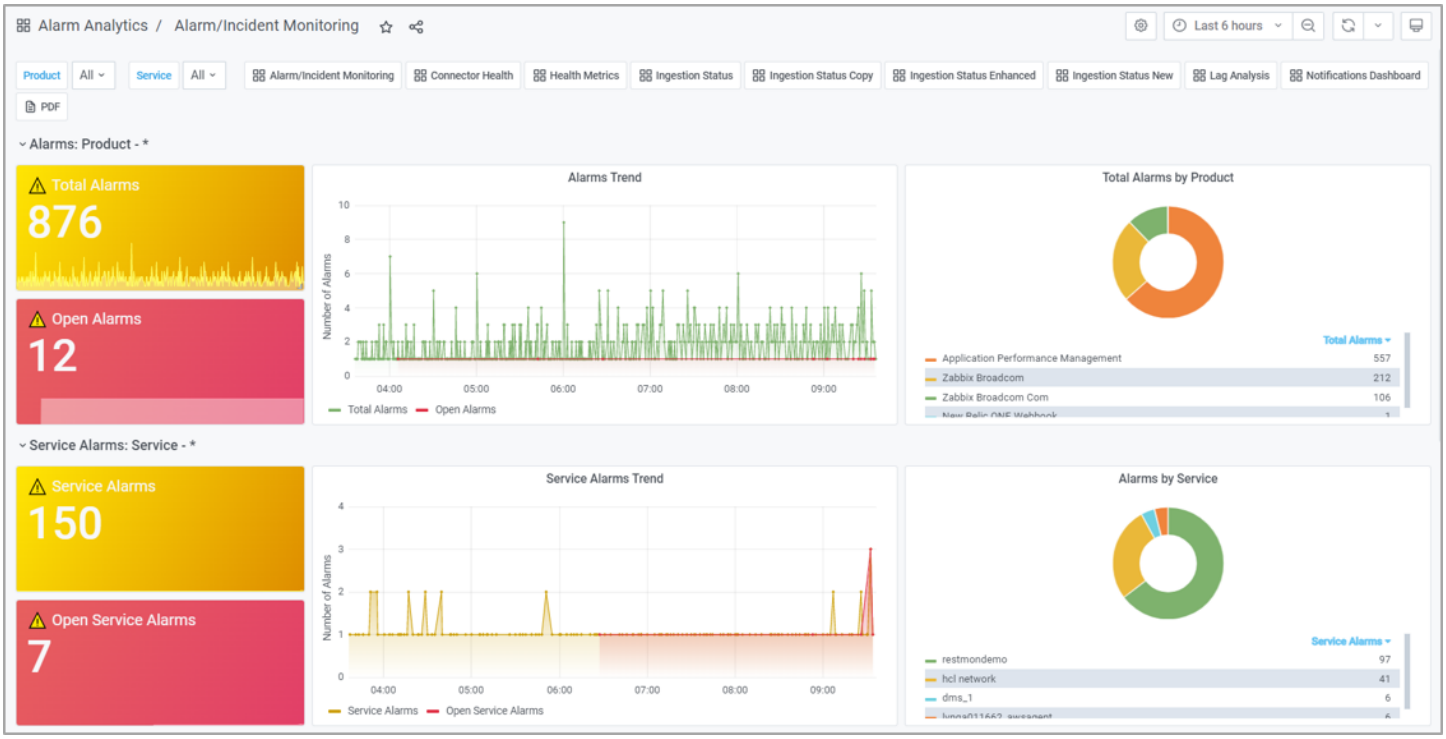
Follow these steps:

1. Log in to DX Operational Intelligence as a Tenant Administrator or a Power User.
2. Click **Log Analytics** in the left navigation pane.
3. Click **Management > Stack Management** in the left navigation pane.
4. Click **Index Patterns**.
5. Search for the custom index that you want to refresh.
6. Click the **Refresh field list** icon that is displayed on the top-right corner.

DX Dashboards

DX Dashboards is a visualization platform that is developed to enable you to search, view, and interact with the stored data. Using DX Dashboards, you can create comprehensive business reports to visualize real-time analytics. Each *DX Dashboard* is a collection of panels that are arranged in a grid pattern. Each panel in the dashboard interacts with the data from the data source and provides the visualization of your data.

The following image illustrates the Alarm/Incident Monitoring dashboard:



DX Dashboards includes the following benefits:

- Enables you to visualize the inventory, health, alarms, metrics, and logs
- Supports multi-tenancy
- Supports graph annotations
- Enables you to drill down to different layers
- Provides slicing and dicing of the AIOps data lake

NOTE
For more information, see the DX Dashboards documentation.

Reference

- DX Operational Intelligence APIs
- DX Platform APIs

DX Operational Intelligence APIs

This section lists the following APIs:

- Service Analytics APIs
- Topology Processor APIs
- Situation Alarm Action APIs
- Situation Clustering Dimensions APIs
- DX Operational Intelligence Query API

Authentication and Authorization of APIs

Access the REST APIs

The REST APIs connect to the DX Operational Intelligence Server to read and retrieve the data. To restrict rogue requests, the server processes the REST API requests after authenticating them. Before you invoke the REST API calls in your code, pass the authentication details and receive the authorization token, which you must include in the header of the subsequent API calls. The EMM Security Server handles the authentication and authorization of the REST APIs.

To access the REST APIs, perform the following operations:

- Authenticate the REST APIs
- Pass the Authorization Details

Authenticate the REST APIs

The following steps describe the REST API authentication process:

1. The resource owner or the user provides the client with their username and password.
2. The client requests an access token from the EMM Security Server's token endpoint by including the credentials that are received from the resource owner. When making the request, the client authenticates with the EMM Security Server.
3. The EMM Security Server authenticates the user and issues an access token. If the authentication fails, then the EMM Security Server returns an error response.

Use the **Authentication** API for authenticating to the REST APIs. This API takes the user credentials as the input parameters and returns the access token and the refresh token. Pass the credentials as the HTTP POST request to the EMM Security Server. Ensure that you also include the **Content Type** as **application/x-www-form-urlencoded** in the POST call. The post request must be sent to the following URL where the EMM Security Server service is available:

```
ess/security/v1/token
```

The following table lists the input parameters of the Authentication APIs:

Parameter	Description	Parameter Type	Required
grant_type	Indicates whether you want to enter the password or the refresh token to receive the access token. The possible values are: • Password • Refresh_Token	form	Yes
username	The username of the authenticating user.	form	Yes, if grant_type=PASSWORD
password	The password of the user. This field is applicable only if the grant_type is set to Password.	form	Yes, if grant_type=PASSWORD
refresh_token	The token that is used to issue a new access token. This field is applicable only if the grant_type is set to REFRESH_TOKEN.	form	Yes, if grant_type=REFRESH_TOKEN
exttoken_requested	Indicates if you want to use the token that an external authentication module issues.	form	No
Authorization	The base64-encoded value of the tenant name.	header	Yes
client_txn_id	The unique identifier to track the transactions.	query	No

The following code snippet provides an example for HTTP POST requests for user authentication:

```
POST /ess/security/v1/token HTTP/1.1
Host: <host_name>:8080
Authorization: Basic REVGQVVMVE9SRw==
Accept-Language: es
Cache-Control: no-cache
Content-Type: application/x-www-form-urlencoded
grant_type=PASSWORD&username=jsmith&password=Wc1205

<host_name> is the FQDN route name of the OpenShift environment.
```

The following code snippet provides an authentication response example:

```
{
"userRefID": 1,
"tkn": "a3bdf820-1eaf-47f4-acd6-7caca7fe17d2",
"tt": "Bearer",
"v": 1800,
"rtkn": "22d27b41-c392-41b6-8785-0503b398b4e7",
"lc": "es"
}
```

The following table explains the response parameters:

Parameter	Description
userRefID	The unique identifier to track the request.
tkn	The access token that must be used for authorization.
tt	The token type. Possible values are: • Basic • Bearer
v	The period for which the token is valid, if it is not used. The default value for which the token is valid if not in use is 1800 seconds.
rtkn	The refresh token that is used to issue a new access token.
lc	The locale that is used for the session.

Pass the Authorization Details

The access token that you have obtained as the result of the Authentication API must be passed as the **Authorization** parameter in the header of API calls. This token indicates that the user is already authenticated. The token therefore eliminates the need for user credentials for successive authentication attempts. Before you pass the token, append the tenant scope to the token, convert this appended string to the base64-encoded format. Now, include this token in the header. The EMM Security Server verifies the incoming request that is based on the token and tenant scope. The EMM Security Server then grants access to the protected resources.

Perform the following steps to pass the authorization details in the API calls:

1. Construct the string in the following format by using the token and tenant scope:
   {"tkn":"<token>","<tenant_scope_argument>":"<value>"}

   NOTE

   The token that is issued for the user must have scope on the tenants that are specified in *<tenant_scope_argument>*.

   The following table lists the possible arguments for tenant scope:

Argument	Description

t	Specifies that the authorization is for the tenant for which the request is being made. The following code provides an example for the string using this argument:
	`{`
	`“tkn”:”a3bdf820-1eaf-47f4-acd6-7caca7fe17d2”,`
	`“t”:”<cohortid_of_tenant>”`
	`}`

all	Specifies that the authorization for all the tenants in the user's scope for which the request is being made. The following code provides an example for the string using this argument:
	`{`
	`“tkn”:”a3bdf820-1eaf-47f4-acd6-7caca7fe17d2”,`
	`“all”:true`
	}
	Note: By default, this value is set to false.

2. Convert the string to the base64-encoded format. For example, you can use **http://www.base64encode.org/** to perform this conversion.

3. Include the base64-encoded string in the header of DX Operational Intelligence APIs. Refer to the **Authorization** line in the following code:

```
GET /ess/security/v1/me HTTP/1.1
Host: <host_name>:8080
Accept: application/json
Authorization: Bearer eyJ0a24iOiI4Njc5YzU3Yy02ZWYyLTQ2OTUtOTc3NC0yYjkwNDQwNDQ3ZGYiLCJhbGwiOnRydWV9
Cache-Control: no-cache
<host_name> is the FQDN route name of the OpenShift environment.
```

You can now use these authorization details to access the protected DX Operational Intelligence APIs. The DX Operational Intelligence API requests must be posted at **/mdo/v2/aoanalytics**.

DX Operational Intelligence Query API

The Data Retrieval API queries and retrieves events and alarms data from DX Operational Intelligence. You can query the data using the permanent user token that you can generate on the Tokens page. This section provides the following information:

> **NOTE**
> This API can return a maximum of 10k records for a query.

Prerequisite

Before you query the data, generate a user token on the Tokens page. For more information, see the Token Management section.

URI Pattern For Alarm Search

You can select the URI based on the deployment and region of your tenant:

- DX SaaS - USA:
  `https://doi.dxi-na1.saas.broadcom.com/oi/v2/aoanalytics/alarms/alarms_all/_search`
- DX SaaS - EU:
  `https://doi.dxi-eu1.saas.broadcom.com/oi/v2/aoanalytics/alarms/alarms_all/_search`

```
https://<doi-admin_route>/oi/v2/aoanalytics/alarms/alarms_all/_search
For example, http://doi-adminui.10.17.105.010.nip.io/oi/v2/aoanalytics/alarms/alarms_all/_search
```

URI Pattern For Events Search

You can select the URI based on the deployment and region of your tenant:

- DX SaaS - USA:
  `https://doi.dxi-na1.saas.broadcom.com/oi/v2/aoanalytics/events/events_change_custom/_search`
- DX SaaS - EU:
  `https://doi.dxi-eu1.saas.broadcom.com/oi/v2/aoanalytics/events/events_change_custom/_search`

```
https://<doi-admin_route>/oi/v2/aoanalytics/events/events_change_custom/_search
For example, http://doi-adminui.10.17.105.010.nip.io/oi/v2/aoanalytics/events/events_change_custom/_search
```

HTTP Method

POST

HTTP Headers

- **Content-Type:** application/JSON
- **Authorization:** bearer <Generated_DX_User_Token>

Request Parameters

The parameter names are case-sensitive.

Parameter	Description	Example
q	Phrase or query to search the data (using the Apache Lucene query syntax) **Default Value:** None • For more information, see the following documentation: – Query String – Date Range • For information about the supported fields for search, see the following sections: – Supported Alarm Fields for Search – Supported Event Fields for Search	For example, enter “product:UIM” to find all the UIM alarms. You can also provide the Timerange in the query to filter based on any timestamp field instead of using timeFrom and timeTo values. For example, timestamp: [2021-08-30T11:00:17+0000 TO 2021-08-30T11:50:17+0000]
timefrom	Start time of the search in the ISO8601 format. This uses the last updated time of the alarm (timestamp field). **Default Value:** None	2021-07-03T07:16:23Z
timeto	End time of the search in the ISO8601 format. This uses the last updated time of the alarm (timestamp field). **Default Value:** None	2021-07-03T07:16:23Z
from	The starting from the index of the hits to return. **Default Value:** 0	
size	The number of hits to return. **Default Value:** 10	

Supported Alarm fields for Search

The following table lists the alarm fields for raw alarms:

Field Name	Format	Mandatory/Optional	Description
alarm_unique_id	String	Mandatory	Unique_id that is used to manage the lifecycle of the alarm. The same unique id should be sent when the status of the alarm changes (from NEW → UPDATED → CLOSED). However, a new unique id should be used once an alarm has been closed and a new alarm is generated for a similar condition. For example, threshold crossing on a metric.
product	String	Mandatory	Source Product that generated the alarm. For example, NewRelic.

Field Name	Format	Mandatory/Optional	Description
message	String	Mandatory	The message of the alarm.
status	String	Mandatory	The status of the alarm. Valid values are NEW, UPDATED, and CLOSED.
severity	String	Mandatory	The severity of the alarm. Valid values are critical, major, minor, and information
timestamp	String	Mandatory	Last updated time of the alarm in ISO 8601 date string. This will be the same as startTime for NEW alarms. "format": "yyyy-MM-dd'T'HH:mm:ssZ"
startTime	String	Mandatory	Creation time of the alarm in ISO 8601 date string. "format": "yyyy-MM-dd'T'HH:mm:ssZ"
closedTime	String	Optional	Timestamp of alarm closure.
ci_unique_id	String	Mandatory	Unique id for the entity on which alarm is raised.
summary	String	Optional	Short summary of the alarm.
host	String	Optional	The hostname of the device/ entity on which the alarm was generated.
alarmType	String	Optional	Type of the alarm. For example, Application, Infrastructure.
product_version	String	Optional	The version of the source product.
metric_name	String	Optional	Name of the metric if the alarm was generated based on the metric threshold crossing.
metric_type	String	Optional	Type of the metric if the alarm was generated based on metric threshold crossing.
configuration_item_type	String	Optional	Entity type on which alarm is raised.
configuration_item	String	Optional	Entity name on which alarm is raised.
tags	Array of Strings	Optional	Adhoc string values.
metric_unique_id	String	Optional	Metric Store (NASS) unique_id. This is mandatory for an alarm to metric drill-down in DX Operational Intelligence.
external_ids	Array of Strings	Optional	Unique ID for the entity on which alarm is raised from the Topology Store (TAS).
alarmURL	String	Optional	URL link to the source product that generated the alarm.

Field Name	Format	Mandatory/Optional	Description
group	Array of Strings	Optional	An array of group names from the source product.
group_id	Array of Strings	Optional	An array of group IDs from the source product.
services_impacted	Array of Strings	Optional	List of OI services impacted by this alarm.
troubleShooterName	String	Optional	Name of the user to whom the alarm is assigned.
troubleTicket	String	Optional	Ticket/Incident ID from the ticketing systems (for example, ServiceNow) if any ticket is assigned for this alarm.
troubleTicketUrl	String	Optional	Link to the ticketing systems.
acknowledged	String	Optional	Set to true if the alarm has been acknowledged.
maintenance	String	Optional	Set to true if this alarm is in an active maintenance window.
annotation	String	Optional	User-defined annotation text on the alarm.
correlated_external_ids	Array of Strings	Optional	Listed of compacted external_ids in TAS if the external_ids have been compacted by Topology Processor.
custom_1, custom_2 etc up to custom_10	String	Optional	Custom string fields.
custom_num_1 custom_num_2	Double	Optional	Custom number fields.

Supported Change Event Fields for Search

The following table lists the change event fields:

Field Name	Format	Mandatory/Optional	Description
product	String	Mandatory	Source Product that generated the event. For example, NewRelic.
message	String	Mandatory	The message of the event
event_unique_id	String	Mandatory	Unique_id for the event.
timestamp	String	Mandatory	Occurrence time of the event in ISO 8601 date string. "format": "yyyy-MM-dd'T'HH:mm:ssZ"
ci_unique_id	String	Optional	Unique id for the entity on which event is raised.
status	String	Optional	The status of the event.

Field Name	Format	Mandatory/Optional	Description
severity	String	Optional	The severity of the event. Valid values are critical, major, minor, and information.
summary	String	Optional	Short summary of the event.
host	String	Optional	The hostname of the device/ entity on which the alarm was generated.
change_type	String	Optional	Type of change event.
product_version	String	Optional	The version of the source product.
configuration_item_type	String	Optional	Entity type on which event is raised.
configuration_item	String	Optional	Entity name on which event is raised.
tags	Array of Strings	Optional	Adhoc string values.
custom_1, custom_2 etc up to custom_10	String	Optional	Custom string fields.
custom_num_1 custom_num_2	Double	Optional	Custom number fields.

Service Analytics APIs

This section describes the Service Analytics APIs. You can access these APIs using the DX Operational Intelligence endpoint and apmgateway endpoint as follows:

- DX Operational Intelligence admin endpoint: <oihost>:<oiport>/oi/v2/oipublic/serviceanalytics/*
- APM Gateway endpoint: <apmgateway>:<port>/oipublic/serviceanalytics/*

The following table describes the Read and Write APIs with the authorization type that can be used with the DX Operational Intelligence endpoint and apmgateway endpoint:

Endpoint Name	Endpoint URI	Authorization Token	User Access To APIs
APM Gateway endpoint	<apmgateway>:<port>/oipublic/serviceanalytics/*	APM Tenant Token	Read APIs
APM Gateway endpoint	<apmgateway>:<port>/oipublic/serviceanalytics/*	User token with all (scope on tenants)	Read APIs and Write APIs
DX Operational Intelligence admin endpoint	<oihost>:<oiport>/oi/v2/oipublic/serviceanalytics/*	User token with all (scope on tenants) and Cohort id	Read APIs and Write API
DX Operational Intelligence admin endpoint:	<oihost>:<oiport>/oi/v2/oipublic/serviceanalytics/*	APM Tenant Token	Read APIs

NOTE
You can find the apmgateway endpoint on the Connectors Parameters page.

The following video explains the Service Analytics APIs:

Write APIs

You can access the following write APIs through the apmgateway endpoint and DX Operational Intelligence endpoint using a user token.

- CREATE Service
- UPDATE Service
- DELETE Services and Service Association

CREATE Service

You can create a service or service hierarchy using *one* of the following APIs:

- **Save API**: /oi/v2/sa/save. For more information, see Save API.
- **Store API:** /oi/v2/sa/store. For more information, see Store API.

NOTE

- Using the Save API, you can update the service by sending the entire service hierarchy including the vertex to be updated.
- Using the Store API, you can update the input vertex by sending the entire vertex payload.

Save API

The Save API enables you to create or update a standalone service or the complete service hierarchy. You can update the service or the service hierarchy using the flag ***replace=true***. This flag replaces the existing service definition with the new service definition. Use this API to perform the following:

- Create a service
- Add a service to the existing service
- Remove service from hierarchy
- Alter the associations between the services (add edges or remove edges)

NOTE

- Ensure you always pass the complete hierarchy. This is because the API checks the difference between the existing hierarchy and the new hierarchy request.
- Edges that are not passed in the payload are considered deleted.
- Vertices (services) that are not passed in the payload are removed from the service hierarchy. However, these services are not deleted from the inventory.
- It is not recommended to update the state of a service using the Save API with replace option.

API Parameters

- Resource URI: You can select the URI based on the deployment and region of your tenant:
  - DX SaaS - USA:
    `https://apmgw.dxi-na1.saas.broadcom.com/oi/v2/sa/save?replace=true`
  - DX SaaS - EU:
    `https://apmgw.dxi-eu1.saas.broadcom.com/oi/v2/sa/save?replace=true`

    NOTE
    For more information on the endpoints, see Service Analytics APIs
- **Method:** POST
- **HTTP Headers:**

  - **Content-Type:** application/JSON
  - **Authorization:** user token that can be generated from the Token Management page.
- **Syntax:**

```
{
  "vertices": [
        {
            "attributes": {
                "type": "<servicetype>",
                "name": "<service name1>",
                "state": "<status of the service>",
                "metrics": [],
                "serviceContent": [
                    {
                        "query": [
                            {
                                "attributeName": "<attribute name>",
                                "attributeValue": "<attribute value>"
                            }
                        ]
                    }
                ],
                "tags": [],
                "location": "",
                "description": "",
                "customProperties": []
            },
            "externalId": "<service name1>"
        },
        {
            "attributes": {
                "type": "<service type>",
                "name": "<service name2>",
                "state": "<status>",
                "serviceContent": [],
                "tags": [],
                "location": "<location>",
                "description": "",
                "customProperties": []
            },
            "externalId": "<service name2>"
        }
    ],
    "edges": [
        {
            "targetExternalId": "<servce name 1>",
            "sourceExternalId": "<service name 2>",
            "attributes": {
                "health_weight": 0.25,
                "risk_weight": 0.25,
                "semantic": "AggregateOf"
            }
        },
```

```
    ]
}
```

- **Response:** 200 OK

```
{ "message": [
   "Service successfully created"
   ],
   "status": "Success"
 }
```

Example:

- **Resource:** https://apmgw.dxi-na1.saas.broadcom.com/oi/v2/sa/save?replace=true
- **Payload:**

```
{
    "vertices": [
        {
            "attributes": {
                "type": "saService",
                "name": "VM1",
                "state": "ACTIVE",
                "metrics": [],
                "serviceContent": [
                    {
                        "query": [
                            {
                                "attributeName": "type",
                                "attributeValue": "VM1"
                            }
                        ]
                    }
                ],
                "tags": [],
                "location": "",
                "description": "",
                "customProperties": []
            },
            "externalId": "VM1"
        },
        {
            "attributes": {
                "type": "saService",
                "name": "All Device Service",
                "state": "ACTIVE",
                "serviceContent": [],
                "tags": [],
                "location": "CANADA",
                "description": "",
                "customProperties": []
            },
            "externalId": "All Device Service"
        }
    ],
```

```
        "edges": [
            {
                "targetExternalId": "VM1",
                "sourceExternalId": "All Device Service",
                "attributes": {
                    "health_weight": 0.25,
                    "risk_weight": 0.25,
                    "semantic": "AggregateOf"
                }
            }
        ]
    }
```

- **Response:**

```
{ "message": [
 "Service successfully created"
],
  "status": "Success"
}
```

Store API

The Store API is a sophisticated version of Save API. You can create a standalone service or service hierarchy and alter a single standalone service or service hierarchy, or services within the hierarchy.

The flag ***replace=true,*** replaces the existing definition with the new one. The vertices and edges (associations) of a service can be in any hierarchy when you want to modify a service. The API validates the hierarchy before it saves. If the corresponding service is not present, then the service association is not created.

You can use the flag auto_weight=true, to set the edge weights automatically.

NOTE

- Store API has a cap of 1000 entities (vertices or edges combined) at a time. If the service hierarchy has more than 1000 entities (vertices or edges combined), you must split the request.
- By using the Store API, you can create the services (vertices) first and then the service associations (edges).

API Parameters

- **Resource URI:** You can select the URI based on the deployment and region of your tenant:
  - DX SaaS - USA:
    `https://apmgw.dxi-na1.saas.broadcom.com/oi/v2/sa/store`
  - DX SaaS - EU:
    `https://apmgw.dxi-eu1.saas.broadcom.com/oi/v2/sa/store`

    NOTE
    For more information on the endpoints, see Service Analytics APIs
- **Method:** POST
- **HTTP Headers:**
  - **Content-Type:** application/JSON
  - **Authorization:** user token that can be generated from the Token Management page.
- **Syntax:**

```
{
  "vertices": [
    {
```

```
    "attributes": {
      "name": "<service name1>",
      "root_service": [
        "<root service1>"
      ],
      "maintenance": true
    }
  },
  {
    "attributes": {
      "name": "<service name2>",
      "root_service": [
        "<root service1>"
      ],
      "serviceContent": [
        {
          "query": [
            {
              "attributeName": "<attribute name>",
              "attributeValue": "<attribute value>"
            }
          ]
        }
      ]
    }
  },
  {
    "attributes": {
      "name": "<service name3>",
      "state": "<status>",
      "root_service": [
        "<root service1>"
      ],
      "serviceContent": [
        {
          "query": [
            {
              "attributeName": "<attribute name>",
              "attributeValue": "<attribute type>"
            }
          ]
        }
      ],
      "tags": [
        "Booking",
        "Production"
      ],
      "location": "<location of service>"
    }
  },
  {
    "attributes": {
      "name": "<service name 4>",
```

```
        "state": "<status>",
        "root_service": [
          "<root service 3>"
        ],
        "serviceContent": [
          {
            "query": [
              {
                "attributeName": "<attribute name>",
                "attributeValue": "<attribute type>"
              }
            ]
          }
        ],
        "tags": [
          "Booking",
          "Production"
        ],
        "location": "<location>"
      }
    },
    {
      "attributes": {
        "name": "<service name5>",
        "root_service": [
          "<root service 3>"
        ],
        "serviceContent": [
          {
            "query": [
              {
                "attributeName": "<attribute name>",
                "attributeValue": "<attribute type>"
              }
            ]
          }
        ],
        "location": ""
      }
    }
  ],
  "edges": [
    {
      "targetExternalId": "<service 2>",
      "attributes": {
        "semantic": "AggregateOf",
        "health_weight": <risk weight>,
        "risk_weight": <risk weight>
      },
      "sourceExternalId": "<service 1>"
    },
    {
      "targetExternalId": "<service 3>",
```

```
      "attributes": {
        "semantic": "AggregateOf",
        "health_weight": <risk weight>,
        "risk_weight": <risk weight>
      },
      "sourceExternalId": "<service 1>"
    },
    {
      "targetExternalId": "<service 3>",
      "attributes": {
        "semantic": "AggregateOf",
        "health_weight": <risk weight>,
        "risk_weight": <risk weight>
      },
      "sourceExternalId": "<service 3>"
    },
    {
      "targetExternalId": "<service 4>",
      "attributes": {
        "semantic": "AggregateOf",
        "health_weight": <risk weight>,
        "risk_weight": <risk weight>
      },
      "sourceExternalId": "<service 3>"
    }
  ]
}
```

Vertex Payload Sample:

```
{
    "vertices": [
        {
            "attributes": {
                "type": "saService",
                "name": "00_NameService01",
                "state": "ACTIVE",
                "serviceContent": [
                    {
                        "query": [
                            {
                                "attributeName": "name",
                                "attributeValue": "tas-scfld-n194",
                                "operator": "IN"
                            }
                        ]
                    }
                ],
                "tags": [
                    "perf"
                ],
                "location": "INDIA",
                "geopoint": {
                    "lat": 17.38,
```

```
                "lon": 78.48
            },
            "numeric_code": "numeric_code",
            "description": "Performance Test Applications",
            "customProperties": [
                {
                    "name": "environment",
                    "value": "perf"
                }
            ],
            "externalId": "00_NameService01"
        }
    },
    {
        "attributes": {
            "type": "saService",
            "name": "00_NameService02",
            "state": "ACTIVE",
            "serviceContent": [
                {
                    "query": [
                        {
                            "attributeName": "name",
                            "attributeValue": "tas-scfld-n195",
                            "operator": "IN"
                        }
                    ]
                }
            ],
            "tags": [
                "perf"
            ],
            "location": "INDIA",
            "geopoint": {
                "lat": 17.38,
                "lon": 78.48
            },
            "numeric_code": "numeric_code",
            "description": "Performance Test Applications",
            "customProperties": [
                {
                    "name": "environment",
                    "value": "perf"
                }
            ],
            "externalId": "00_NameService02"
        }
    }

],
"edges": [

]
```

```
}
```

Edge Payload Sample:

```
{
    "vertices": [

    ],
    "edges": [
        {
            "targetExternalId": "00_NameService02",
            "attributes": {
                "semantic": "AggregateOf",
                "health_weight": 0.7,
                "risk_weight": 0.7
            },
            "sourceExternalId": "00_NameService01"
        },
        {
            "targetExternalId": "00_NameService03",
            "attributes": {
                "semantic": "AggregateOf",
                "health_weight": 0.3,
                "risk_weight": 0.3
            },
            "sourceExternalId": "00_NameService01"
        }
    ]
}
```

Auto Weight Payload Sample: Providing the auto_weight=true, overrides the input weights. **Endpoint:** https://apmgw.dxi-na1.saas.broadcom.com/oi/v2/sa/store?replace=true&auto_weight=true

```
{
    "vertices": [

    ],
    "edges": [
        {
            "targetExternalId": "00_NameService02",
            "attributes": {
                "semantic": "AggregateOf",
                "health_weight": 0.7,
                "risk_weight": 0.7
            },
            "sourceExternalId": "00_NameService01"
        },
        {
            "targetExternalId": "00_NameService03",
            "attributes": {
                "semantic": "AggregateOf",
                "health_weight": 0.3,
                "risk_weight": 0.3
            },
            "sourceExternalId": "00_NameService01"
        }
```

```
    ]
}
```

- **Response:** 200 OK

Example:

- **Resource URL:** https://apmgw.dxi-na1.saas.broadcom.com/oi/v2/sa/store?replace=true&auto_weight=true
- **Payload:**

```
{
  "vertices": [
    {
      "attributes": {
        "name": "BB1",
        "root_service": [
          "BB1"
        ],
        "maintenance": true
      }
    },
    {
      "attributes": {
        "name": "BB2",
        "root_service": [
          "BB1"
        ],
        "serviceContent": [
          {
            "query": [
              {
                "attributeName": "uim.groups",
                "attributeValue": "Operating Systems|Windows"
              }
            ]
          }
        ]
      }
    },
    {
      "attributes": {
        "name": "BB3",
        "state": "ACTIVE",
        "root_service": [
          "BB1"
        ],
        "serviceContent": [
          {
            "query": [
              {
                "attributeName": "uim.groups",
                "attributeValue": "Operating Systems|Windows"
              }
            ]
```

```
          }
        ],
        "tags": [
          "Booking",
          "Production"
        ],
        "location": "India,test1"
      }
    },
    {
      "attributes": {
        "name": "BB9",
        "state": "ACTIVE",
        "root_service": [
          "BB3"
        ],
        "serviceContent": [
          {
            "query": [
              {
                "attributeName": "uim.groups",
                "attributeValue": "Operating Systems|Windows"
              }
            ]
          }
        ],
        "tags": [
          "Booking",
          "Production"
        ],
        "location": "India"
      }
    },
    {
      "attributes": {
        "name": "BB8",
        "root_service": [
          "BB9"
        ],
        "serviceContent": [
          {
            "query": [
              {
                "attributeName": "uim.groups",
                "attributeValue": "Operating Systems|Windows"
              }
            ]
          }
        ],
        "location": ""
      }
    }
  ],
```

```
  "edges": [
    {
      "targetExternalId": "BB2",
      "attributes": {
        "semantic": "AggregateOf",
        "health_weight": 0.2,
        "risk_weight": 0.5
      },
      "sourceExternalId": "BB1"
    },
    {
      "targetExternalId": "BB3",
      "attributes": {
        "semantic": "AggregateOf",
        "health_weight": 0.5,
        "risk_weight": 0.5
      },
      "sourceExternalId": "BB1"
    },
    {
      "targetExternalId": "BB9",
      "attributes": {
        "semantic": "AggregateOf",
        "health_weight": 0.5,
        "risk_weight": 0.5
      },
      "sourceExternalId": "BB3"
    },
    {
      "targetExternalId": "BB8",
      "attributes": {
        "semantic": "AggregateOf",
        "health_weight": 0.5,
        "risk_weight": 0.5
      },
      "sourceExternalId": "BB9"
    }
  ]
}
```

Response:

```
{ "message": [
  "Service/s successfully created.",
   "No of edges created: 4"
],
  "status": "Success"
}
```

UPDATE Service

UPDATE Service or Service Hierarchy

You can use the Save API and Store API to update the service.

NOTE

- Using the Save API, you can update the service by sending the entire service hierarchy including the vertex to be updated.
- Using the Store API, you can update the input vertex by sending the entire vertex payload.

Update Service Attributes

The Update API allows you to perform the following:

- Add, update and delete tags to the selected service
- Add and Update service attributes

API Parameters

- **Resource URI:** You can select the URI based on the deployment and region of your tenant:
  - DX SaaS - USA:
    `https://apmgw.dxi-na1.saas.broadcom.com/oi/v2/sa/update/{service_name}`
  - DX SaaS - EU:
    `https://apmgw.dxi-eu1.saas.broadcom.com/oi/v2/sa/update/{service_name}`
- **Method:** POST
- **HTTP Headers:**
  - **Content-Type:** application/JSON
  - **Authorization:** user token that can be generated from the Token Management page.
- **Response:** 200 OK

```
 {

{
    "message":
      ["service message"
      ],
    "status": "<Success/Failed>"
}

}
```

- **Syntax of Update Service Content:**

```
            {
    "serviceContent": [{
        "query": [{
            "attributeName": "<type>",
            "attributeValue": "<value>"
        },
        {
            "attributeName": "<type1>",
            "attributeValue": "<value1>"
        }]
    }]
}
```

- **Response:**

```
{
    "serviceContent": [{
```

```
        "query": [{
            "attributeName": "type",
            "attributeValue": "vm"
        },
        {
            "attributeName": "environment",
            "attributeValue": "prod"
        }]
    }]
}
```

- **Multi Attributes Update Syntax**

```
{
    "tags":["<tag>"],
    "customProperties":[{"name":"<inputname>", "value":"<inputvalue>"}, {"name":"<inputname>",
 "value":"<inputvalue>"}],
 "state":"<state of service>"
}
```

- **Example for Multi-Attribute Update:**

```
{
    "tags":["hello"],
    "customProperties":[{"name":"country", "value":"India"}, {"name":"city", "value":"Hyderabad"}],
    "state":"ACTIVE"
}
```

Example:

- **Resource URI:** https://apmgw.dxi-na1.saas.broadcom.com/oi/v2/sa/update/BB?merge=true>
- **Response:**

```
{
  {
    "message": ["Service successfully updated"],
    "status": "Success"
```

- **Invalid Response**

```
{
    "message": ["Service name - Serv101 doesn't exist"],
    "status": "Failed"
}
```

DELETE Services and Service Association

DELETE Services

This API deletes a service. Use flag rescursive=true to recursively delete the service and its underlying sub-services.

API Parameters

- **Resource URI:** You can select the URI based on the deployment and region of your tenant:
  - DX SaaS - USA:
    `https://apmgw.dxi-na1.saas.broadcom.com/oi/v2/sa/service/status`
  - DX SaaS - EU:
    `https://apmgw.dxi-eu1.saas.broadcom.com/oi/v2/sa/service/status`
- **Method:** POST
- **HTTP Headers:**
  - **Content-Type:** application/JSON
  - **Authorization:** user token that can be generated from the Token Management page.
- **Syntax:**

```
{
 "name": "<service_name>",
 "action":"delete"
}
```

- **Response:** 202 Accepted

```
{"message":["<body of message>"],"status":"<status>"}
```

Example

Resource URI: https://apmgw.dxi-na1.saas.broadcom.com/oi/v2/sa/service/status

Payload:

```
{
  "name":"A1",
  "action":"delete"
}
```

Response:

```
{"message":["Service successfully marked for deletion"],"status":"Accepted"}
```

DELETE Service Association

This API validates and deletes the edges between the services. Inputs the list of edges with source and target service names that need to be deleted.

API Parameters

- **Resource URI:** You can select the URI based on the deployment and region of your tenant:
  - DX SaaS - USA:
    `https://apmgw.dxi-na1.saas.broadcom.com/oi/v2/sa/deleteEdges?auto_weight=true`
  - DX SaaS - EU:
    `https://apmgw.dxi-eu1.saas.broadcom.com/oi/v2/sa/deleteEdges?auto_weight=true`
- **Method:** POST
- **HTTP Headers:**
  - **Content-Type:** application/JSON
  - **Authorization:** user token that can be generated from the Token Management page.
- **Syntax:**

```
[
  {
    "targetExternalId":"<sourceservicename>",
    "sourceExternalId":"<targetservicename>"
  }
]
```

- **Response:** 200 OK

```
{
  "message": [
      "No. of edges deleted: <count of edges deleted>"
   ],
   "status": "<status>"
}
```

Example

- **Resource URI:**https://apmgw.dxi-na1.saas.broadcom.com/oi/v2/sa/deleteEdges?auto_weight=true
- **Payload:**

```
[
  {
     "targetExternalId":"BB2",
     "sourceExternalId":"BB1"
  }
  {
     "targetExternalId":"BB3",
     "sourceExternalId":"BB1"
  }
]
```

Response:

```
{
   "FailedEdges": [{"targetExternalid":"BB3"
,
"sourceExternalId"
:
"BB1"}],

   "message": [
        "No. of edges deleted:1", "No. of edges failed: 1"],

    "status": "Success
 }
```

Read APIs

You can access the following Read APIs through apmgateway endpoint and DX Operational Intelligence admin UI endpoint using APM tenant token.

- GET Service Hierarchy
- Retrieve a Service
- Retrieve All Services

GET Service Hierarchy

Retrieve Service Hierarchy

This API fetches the service hierarchy, including attributes and sub-services of a specific service.

API Parameters

- **Resource URI:** You can select the URI based on the deployment and region of your tenant:
  - DX SaaS - USA:
    `https://apmgw.dxi-na1.saas.broadcom.com/oi/v2/sa/service/details/hierarchy`
  - DX SaaS - EU:
    `https://apmgw.dxi-eu1.saas.broadcom.com/oi/v2/sa/service/details/hierarchy`
- **Method:** GET
- **HTTP Headers:**
  - **Content-Type:** application/JSON
  - **Authorization:** user token that can be generated from the Token Management page.
- **Request:**
  `https://apmgw.dxi-na1.saas.broadcom.com/oi/v2/sa/service/details/hierarchy/{service_name}`
- **Response:** 200 OK

```
{
   "vertices": [
   {
     "externalId": "<external id>",
           "attributes": {
               "type": "<service type>",
               "name": "<service name>",
               "state": "<service status>",
               "root_service": [
                   "<root service name>"
               ],
               "serviceContent": [],
               "tags": [],
               "createTimestamp": <service created timestamp>,
               "timestamp": <timestamp>,
               "maintenance": true,
               "situationsIncludeChildServices": <true/false>,
               "customProperties": [],
               "metrics": [
                   {
                       "type": "<type of service>",
                       "sourceName": "<source name>",
                       "attributeName": "<attribute name:status>",
                       "id": "<id>",
                       "firstSeen": <firstseen>
                       "extension_Ids": <external id>
                       "prediction_health|LOWER|UPPER"
                       ]
                   },

               ],
```

```
                "customMetrics": []
            },
            "endTime": <end time>,
            "id": <id>,
            "startTime": <start time>,
            "associated_graph": {
                "vertices": [],
                "edges": []
            }
    }
```

Example

GET Request:https://apmgw.dxi-na1.saas.broadcom.com/oi/v2/sa/service/details/hierarchy/BB

Response:


```
{
    "vertices": [
        {
            "externalId": "SA:1F:dc4e3cdc-e027-4ddf-839a-503275434c86",
            "attributes": {
                "type": "saService",
                "name": "BB",
                "state": "ACTIVE",
                "root_service": [
                    "BB"
                ],
                "serviceContent": [],
                "tags": [],
                "createTimestamp": 1619691854422,
                "timestamp": 1619691854422,
                "maintenance": true,
                "situationsIncludeChildServices": false,
                "customProperties": [],
                "metrics": [
                    {
                        "type": "service_risk",
                        "sourceName": "OI|SA|SERVICES|ALL",
                        "attributeName": "dc4e3cdc-e3275434c86:service_risk",
                        "id": "zZB-AE-D-NoMZXt",
                        "firstSeen": 1619691380000
                    },
                    {
                        "type": "prediction_health",
                        "sourceName": "OI|SA|SERVICES|ALL",
                        "attributeName": "dc44ddf-839a-503275434c86:service_health",
                        "id": "1ZB-AAB-D-7fp2Fu",
                        "firstSeen": 1619691380000,
                        "extension_Ids": [
                            "prediction_health|LOWER|UPPER"
                        ]
                    },
                    {
```

```
                        "type": "service_availability",
                        "sourceName": "OI|SA|SERVICES|ALL",
                        "attributeName": "dc4e3cdc-e3275434c86:service_availability",
                        "id": "yZB-AAB-D-HMsxFt",
                        "firstSeen": 1619691380000
                    },
                    {
                        "type": "service_health",
                        "sourceName": "OI|SA|SERVICES|ALL",
                        "attributeName": "dc44ddf-839a-503275434c86:service_health",
                        "id": "0ZB-AAB-D-r9Xu.u",
                        "firstSeen": 1619691380000
                    }
                ],
                "customMetrics": []
            },
            "endTime": 1622283854427,
            "id": 2122,
            "startTime": 1619691854427,
            "associated_graph": {
                "vertices": [],
                "edges": []
            }
        }
    ],
    "edges": [],
    "version": "eyJ2ZXk5ODMxfNpzNywidCI6MTYxOTcxNDI5OTgzMX0sInN0YXR1c1ZlcnNpb24iOm51bGx9",
    "nextOffset": -1,
    "totalVertices": 1,
    "totalEdges": 0
}
```


Get Service Hierarchy for Set of Services

This API lets you retrieve the service hierarchy (parent or children) based on the relationType. The response is key-value pair, where Key is a service ID, and the value is the service name. The relationType can be parent or children.

API Parameters

- **Resource URI:** You can select the URI based on the deployment and region of your tenant:
  - DX SaaS - USA:
    `https://apmgw.dxi-na1.saas.broadcom.com/oi/v2/sa/service/details/hierarchy`
  - DX SaaS - EU:
    `https://apmgw.dxi-eu1.saas.broadcom.com/oi/v2/sa/service/details/hierarchy`
- **Method:** POST
- **HTTP Headers:**
  - **Content-Type:** application/JSON
  - **Authorization:** user token that can be generated from the Token Management page.
- **Syntax:**

```
{

"services": [
"Service Names"
```

```
    ],
    "relationType": "children/parents",
    "state": "ALL"
    }
```

- **Response:** 200 OK

```
    {
      "Servicename": {
            "serviceid1": "children/parentservicename1",
            "serviceid2": "children/parentservicename2",
            "serviceid3": "children/parentservicename3",
           }
    }
```

Example**Resource URI:** https://apmgw.dxi-na1.saas.broadcom.com/oi/v2/sa/service/details/hierarchy

Payload:

```
{
"services"
: [
"A1"
],
"relationType"
:
"children"
,
"state"
:
"ALL"
}
```

Response:

```
{
  "A1": {
        "SA:1A4D509E-A2F5450AEDFF:dd79b1f5-6d3af572e419": "A1",
        "SA:1A4D509E-A2F5450AEDFF:70e1d82d-9b6d20271d6c": "A4",
        "SA:1A4D509E-A2F5450AEDFF:11c34751-394a897887a0": "A13",
        "SA:1A4D509E-A2F5450AEDFF:578e3a9f-c54b74c69ae6": "A15",
        "SA:1A4D509E-A2F5450AEDFF:bc9fefc6-2b1ced5a6eef": "A9",
        "SA:1A4D509E-A2F5450AEDFF:318eda58-abe89296c12f": "A11",
        "SA:1A4D509E-A2F5450AEDFF:b2960054-20d35344fe0c": "A5"
       }
}
```

Retrieve a Service

The Services API fetches the following:

- Service information of the selected service using the parameter ***service_name***
- Service information of the selected service and its underlying sub services using the flag ***subservices=true***

API Parameters

- **Resource URI:** You can select the URI based on the deployment and region of your tenant:

  - DX SaaS - USA:
    `https://apmgw.dxi-na1.saas.broadcom.com/oi/v2/sa/services/{service_name}`
  - DX SaaS - EU:
    `https://apmgw.dxi-eu1.saas.broadcom.com/oi/v2/sa/services/{service_name}`
- **Method:** GET
- **HTTP Headers:**
  - **Content-Type:** application/JSON
  - **Authorization:** user token that can be generated from the Token Management page.
- **Request:**
  `https://apmgw.dxi-na1.saas.broadcom.com/oi/v2/sa/services/{service_name}/?subservices=true`
- **Response:** 200 OK

```
{
   "vertices": [
   {
     "externalId": "<external id>",
           "attributes": {
               "type": "<service type>",
               "name": "<service name>",
               "state": "<service status>",
               "root_service": [
                   "<root service name>"
               ],
               "serviceContent": [],
               "tags": [],
               "createTimestamp": <service created timestamp>,
               "timestamp": <timestamp>,
               "maintenance": true,
               "situationsIncludeChildServices": <true/false>,
               "customProperties": [],
               "metrics": [
                   {
                       "type": "<type of service>",
                       "sourceName": "<source name>",
                       "attributeName": "<attribute name:status>",
                       "id": "<id>",
                       "firstSeen": <firstseen>
                       "extension_Ids": <external id>
                       "prediction_health|LOWER|UPPER"
                       ]
                   },

               ],
               "customMetrics": []
           },
           "endTime": <end time>,
           "id": <id>,
           "startTime": <start time>,
           "associated_graph": {
               "vertices": [],
               "edges": []
```

```
                }
    }
```

Example

GET Request: https://apmgw.dxi-na1.saas.broadcom.com/oi/v2/sa/services/{service_name}/BB?subservices=true

Response:


```
{
    "vertices": [
        {
            "externalId": "SA:1F:dc4e3cdc-e027-4ddf-839a-503275434c86",
            "attributes": {
                "type": "saService",
                "name": "BB",
                "state": "ACTIVE",
                "root_service": [
                    "BB"
                ],
                "serviceContent": [],
                "tags": [],
                "createTimestamp": 1619691854422,
                "timestamp": 1619691854422,
                "maintenance": true,
                "situationsIncludeChildServices": false,
                "customProperties": [],
                "metrics": [
                    {
                        "type": "service_risk",
                        "sourceName": "OI|SA|SERVICES|ALL",
                        "attributeName": "dc4e3cdc-e3275434c86:service_risk",
                        "id": "zZB-AE-D-NoMZXt",
                        "firstSeen": 1619691380000
                    },
                    {
                        "type": "prediction_health",
                        "sourceName": "OI|SA|SERVICES|ALL",
                        "attributeName": "dc44ddf-839a-503275434c86:service_health",
                        "id": "1ZB-AAB-D-7fp2Fu",
                        "firstSeen": 1619691380000,
                        "extension_Ids": [
                            "prediction_health|LOWER|UPPER"
                        ]
                    },
                    {
                        "type": "service_availability",
                        "sourceName": "OI|SA|SERVICES|ALL",
                        "attributeName": "dc4e3cdc-e3275434c86:service_availability",
                        "id": "yZB-AAB-D-HMsxFt",
                        "firstSeen": 1619691380000
                    },
                    {
                        "type": "service_health",
```

```
                        "sourceName": "OI|SA|SERVICES|ALL",
                        "attributeName": "dc44ddf-839a-503275434c86:service_health",
                        "id": "0ZB-AAB-D-r9Xu.u",
                        "firstSeen": 1619691380000
                    }
                ],
                "customMetrics": []
            },
            "endTime": 1622283854427,
            "id": 2122,
            "startTime": 1619691854427,
            "associated_graph": {
                "vertices": [],
                "edges": []
            }
        }
    ],
    "edges": [],
    "version": "eyJ2ZXk5ODMxfSwiZWRnZVZlcnNpzNywidCI6MTYxOTcxNDI5OTgzMX0sInN0YXR1c1ZlcnNpb24iOm51bGx9",
    "nextOffset": -1,
    "totalVertices": 1,
    "totalEdges": 0
}
```


Retrieve All Services

This API fetches all the services. It fetches a paginated service list. You can input the page_num & page_size to get the output. The max page_size is 1000.

API Parameters

- **Resource URI:** You can select the URI based on the deployment and region of your tenant:
  - DX SaaS - USA:
    `https://apmgw.dxi-na1.saas.broadcom.com/oi/v2/servicerepo/allservices`
  - DX SaaS - EU:
    `https://apmgw.dxi-eu1.saas.broadcom.com/oi/v2/servicerepo/allservices`
- **Method:** GET
- **HTTP Headers:**
  - **Content-Type:** application/JSON
  - **Authorization:** user token that can be generated from the Token Management page.
- **Request:**
  `https://apmgw.dxi-na1.saas.broadcom.com/oi/v2/servicerepo/allservices?page_num=1&page_size=100`
- **Response:** 200 OK


```
{
    "services": [
        {
            "name": "<servicename>",
            "status": "<status>",
            "rootServices": [
                "<rootservice>"
```

```
                ],
                "id": "<serviceid>"
            },

            {
                "name": "<servicename>",
                "status": "<status>",
                "rootServices": [
                    "<rootservice>"
                ],
                "id": "<serviceid>"
            }
        ],
        "totalCnt": <service count>
    }
```

Example

Request URI: https://apmgw.dxi-na1.saas.broadcom.com/oi/v2/servicerepo/allservices?page_num=1&page_size=100

Response:

```
{
    "services": [
        {
            "name": "a1",
            "status": "ACTIVE",
            "rootServices": [
                "grp1"
            ],
            "id": "SA:A347F578BBCB:d14ce9ba-113a-4e5b-acb4-7da2bf8dc011"
        },

        {
            "name": "PK19",
            "status": "ACTIVE",
            "rootServices": [
                "PK16"
            ],
            "id": "SA:A347F578BBCB:fa9f069d-6a46-4de0-8480-410248960aeb"
        }
    ],
    "totalCnt": 10
}
```

Service Templates APIs

DX Operational Intelligence provides the following APIs for service templates:

- Create a Service Template API
- Update Service Template API
- List All Available Templates
- Get Service Template Definition by Template Name
- Create a Service Template by Source Name
- Apply Service Template
- Delete Custom Service Template by Name
- Delete All Custom Service Templates
- Get All Custom Service Templates

Vertex Attributes

The following lists the attributes:

Attribute	Mandatory	Description
name	Yes	Name of the service template. In TAS, the name has an externalId as SATemplate:*<name>* The name is limited to 255 characters.
state	No	State of the service. Can be ACTIVE, INACTIVE, DEPLOYED, or DRAFT. Defaulted to DRAFT if not given.
serviceContent	Yes	The actual TAS query that fetches the underlying CIs and their relationship.
tags	No	Associate any tags with the service (limited to 50 tags per service).
location	No	Location of the service.
geopoint	No	
numeric_code	No	
description	No	A short write-up describing the purpose of the service created.
customProperties	No	Add any custom property apart from the ones available. Limited to 20 (Property key limited to 50 characters)
source	No	Either OI or Custom.
scanType	No	Either Automatic or Manual. The default value is Manual.
serviceModelingType	No	Either individual or Aggregate. The default value is Aggregate.
vertexAttributeNames	No	The Vertex attribute name which needs to make part of service tags.
addTemplateAsParentService	No	If true, then one service is created with a template parent name and all services are children.
serviceNamePattern	No	Default is the name of the attributeValue.

Create a Service Template

You can create a service template using the following API information:

Name	Description
Resource URI	`https://doi.dxi-na1.saas.broadcom.com/oi/v2/template/create` `http or https://<doi.adminui>/oi/v2/template/create`
Method	POST
HTTP Headers	`Content-Type:application/json`
Authorization	`Bearer {bearerToken/userToken}`

Name	Description

Request	{
	"vertices": [
	{
	"attributes": {
	"type": "GROUPS",
	"name": "Spectrum Group1",
	"state": "ACTIVE",
	"maintenance": false,
	"situationsIncludeChildServices": false,
	"serviceContent": [
	{
	"query": [
	{
	"attributeName":
	"k8s_service_labels_app.kubernetes.io/instance",
	"attributeValue": "rating-server-
	gcp-prd-gke-gasse1a",
	"operator": "IN"
	}
	]
	}
	],
	"tags": [
	"Channel",
	"Data",
	"Service"
	],
	"location": "INDIA",
	"description": "This is health Application",
	"customProperties": [
	{
	"name": "Name",
	"value": "App"
	},
	{
	"name": "Type",
	"value": "banking"
	},
	{
	"name": "Subtype",
	"value": "Dev"
	}
	],
	"customMetrics": [
	{
	"displayName": "cpu",
	"attributeName": "CPU:Processor Count",
	"sourceName": "SuperDomain|
	SpringBootTestServer|Java|Nowhere-Engine-One",
	"id": null
	}
	],
	"source": "CUSTOM",

	"scanType": "MANUAL", "addTemplateAsParentService": "true", "serviceModelingType": "Individual" } }

Update Service Template

You can update a service template using the following API information:

Name	Description
Resource URI	`https://doi.dxi-na1.saas.broadcom.com/oi/v2/template/update` `http or https://<doi.adminui>/oi/v2/template/update`
Method	POST
HTTP Headers	`Content-Type:application/json`
Authorization	`Bearer {bearerToken/userToken}`

Name	Description

Request	{
	"vertices": [
	{
	"attributes": {
	"type": "GROUPS",
	"name": "Spectrum Group1",
	"state": "ACTIVE",
	"maintenance": false,
	"situationsIncludeChildServices": false,
	"serviceContent": [
	{
	"query": [
	{
	"attributeName": "k8s_service_labels_app.kubernetes.io/instance",
	"attributeValue": "rating-server-gcp-prd-gke-gasse1a",
	"operator": "IN"
	}
	]
	}
	],
	"tags": [
	"Channel",
	"Data",
	"Service"
	],
	"location": "INDIA",
	"description": "This is health Application",
	"customProperties": [
	{
	"name": "Name",
	"value": "App"
	},
	{
	"name": "Type",
	"value": "banking"
	},
	{
	"name": "Subtype",
	"value": "Dev"
	}
	],
	"customMetrics": [
	{
	"displayName": "cpu",
	"attributeName": "CPU:Processor Count",
	"sourceName": "SuperDomain|SpringBootTestServer|Java|Nowhere-Engine-One",
	"id": null
	}
	]
	}

	}] }

List All Available Templates

You can get a list of all the service templates that are available in the drop-down for service creation using the following API information:

Name	Description
Resource URI	`https://doi.dxi-na1.saas.broadcom.com/oi/v2/template/allAvailable` `http or https://<doi.adminui>/oi/v2/template/allAvailable`
Method	GET
HTTP Headers	`Content-Type:application/json`
Authorization	`Bearer {bearerToken/userToken}`

Name	Description

Response	[
	{
	"name": "SpectrumGroup",
	"id": "SATemplate:SpectrumGroup",
	"type": "GROUPS",
	"origin": "OI",
	"description": "This is health Application"
	},
	{
	"name": "EachSpectrumGroup",
	"id": "SATemplate:EachSpectrumGroup",
	"type": "GROUPS",
	"origin": "OI",
	"description": "Using this template you can create service where template will be a parent service and each the spectrum groups will ba a a child service"
	},
	{
	"name": "UIM Group Test one two three",
	"id": "SATemplate:UIM Group",
	"type": "GROUPS",
	"origin": "OI",
	"description": "This template wwill create Service using UIM groups"
	},
	{
	"name": "Rating Server Test gassela",
	"id": "SATemplate:Rating Server Test gassela",
	"type": "saService",
	"origin": null,
	"description": "rating"
	},
	{
	"name": "Spectrum Group",
	"id": "SATemplate:Spectrum Group",
	"type": "GROUPS",
	"origin": null,
	"description": "service to create from using this template"
	}
	]

Get Service Template Definition by Template Name

You can get the service template definition by template name using the following information:

Name	Description
Resource URI	`https://doi.dxi-na1.saas.broadcom.com/oi/v2/template/byName/<Source>/<Template_Name>` `http or https://<doi.adminui>/oi/v2/template/byName/<Source>/<Template_Name>`
Method	GET
HTTP Headers	`Content-Type:application/json`
Authorization	`Bearer {bearerToken/userToken}`

Name	Description

Sample Response	{
	"vertices": [
	{
	"externalId": "SATemplate:SpectrumGroup",
	"attributes": {
	"type": "GROUPS",
	"name": "SpectrumGroup",
	"state": "ACTIVE",
	"root_service": [
	
	],
	"serviceContent": [
	{
	"query": [
	{
	"attributeName": "spectrum.groups",
	"attributeValue": "Global-TOR-Switches.*",
	"operator": "MATCHES"
	}
	],
	"options": {
	"followTransactions": false
	}
	}
	],
	"tags": [
	"Channel",
	"Service",
	"Data"
	],
	"location": "INDIA",
	"description": "This is health Application",
	"maintenance": false,
	"situationsIncludeChildServices": false,
	"restrictSLOOnMaintenance": false,
	"customProperties": [
	{
	"name": "Name",
	"value": "App"
	},
	{
	"name": "Type",
	"value": "banking"
	},
	{
	"name": "Subtype",
	"value": "Dev"
	}
	],
	"metrics": [
	
	],

	"customMetrics": [{ "displayName": "cpu", "sourceName": "SuperDomain|SpringBootTestServer|Java|Nowhere-Engine-One",

Create a Service Template by Source Name

You can create a service template by source name using the following information:

Name	Description
Resource URI	`https://doi.dxi-na1.saas.broadcom.com/oi/v2/template/byName/<Source>` `http or https://<doi.adminui>/oi/v2/template/byName/<Source>`
Method	POST
HTTP Headers	`Content-Type:application/json`
Authorization	`Bearer {bearerToken/userToken}`

Sample Request	`{`
	`  "names": [`
	`  "ASM Monitor",`
	`  "Spectrum Group"`
	`  ]`
	`}`

Name	Description

Sample Response	{
	"vertices": [
	{
	"externalId": "SATemplate:ASM Monitor",
	"attributes": {
	"type": "GROUPS",
	"name": "ASM Monitor",
	"state": "ACTIVE",
	"root_service": [],
	"serviceContent": [
	{
	"query": [
	{
	"attributeName": "monitor",
	"attributeValue": "Service Analytics",
	"operator": "IN"
	}
	],
	"options": {
	"followTransactions": false
	}
	}
	],
	"tags": [
	"Axa Application service"
	],
	"location": "INDIA",
	"description": "This template wwill create Service from Axa Apps",
	"maintenance": false,
	"situationsIncludeChildServices": false,
	"restrictSLOOnMaintenance": false,
	"customProperties": [],
	"metrics": [],
	"customMetrics": [],
	"scanType": "MANUAL",
	"vertexAttributeNames": [
	"resourceType"
	],
	"serviceModelingType": "Individual",
	"addTemplateAsParentService": "true",
	"source": "OI",
	"serviceNamePattern": "Monitor-${attributeValue}"
	}
	}
	]
	}

Apply Service Template

You can use this API to validate and preview the changes that are made to the service template before you save and update the information:

Name	Description
Resource URI	`https://doi.dxi-na1.saas.broadcom.com/oi/v2/template/apply` `http or https://<doi.adminui>/oi/v2/template/apply`
Method	POST
HTTP Headers	`Content-Type:application/json`
Authorization	`Bearer {bearerToken/userToken}`

Name	Description

Request Payload	{
	"vertices": [
	{
	"externalId": "SATemplate:UIM_API3",
	"attributes": {
	"type": "GROUPS",
	"name": "UIM_API3",
	"state": "ACTIVE",
	"maintenance": false,
	"situationsIncludeChildServices": false,
	"serviceContent": [
	{
	"query": [
	{
	"attributeName": "spectrum.groups",
	"attributeValue": ".*",
	"operator": "MATCHES"
	}
	]
	}
	],
	"tags": [
	"UIM Service Template",
	"UIM Service work"
	],
	"location": "INDIA",
	"description": "This is health Application",
	"customProperties": [
	{
	"name": "Name",
	"value": "App"
	},
	{
	"name": "Type",
	"value": "banking"
	},
	{
	"name": "Subtype",
	"value": "Dev"
	}
	],
	"customMetrics": [
	{
	"displayName": "cpu",
	"attributeName": "CPU:Processor Count",
	"sourceName": "SuperDomain|
	SpringBootTestServer|Java|Nowhere-Engine-One",
	"id": null
	}
	],
	"source": "CUSTOM",
	"scanType": "AUTOMATIC",
	"addTemplateAsParentService": "false",
	"serviceModelingType": "Aggregate"
	}
	}
	]
	}

Name	Description

Response	{
	"vertices": [
	{
	"externalId": "UIM_API3",
	"attributes": {
	"name": "UIM_API3",
	"root_service": [
	
	],
	"serviceContent": [
	{
	"query": [
	{
	"attributeName": "spectrum.groups",
	"attributeValue": ".*",
	"operator": "MATCHES"
	}
	],
	"options": {
	"followTransactions": false
	}
	}
	],
	"tags": [
	"UIM Service Template",
	"UIM Service work"
	],
	"location": "INDIA",
	"description": "This is health Application",
	"maintenance": false,
	"situationsIncludeChildServices": false,
	"restrictSLOOnMaintenance": false,
	"customProperties": [
	{
	"name": "Name",
	"value": "App"
	},
	{
	"name": "Type",
	"value": "banking"
	},
	{
	"name": "Subtype",
	"value": "Dev"
	}
	],
	"metrics": [
	
	],
	"customMetrics": [
	{
	"displayName": "cpu",
	"sourceName": "SuperDomain|
	SpringBootTestServer|Java|Nowhere-Engine-One",
	"attributeName": "CPU:Processor Count",
	"id": null
	}
	]

Delete Custom Service Template by Name

You can delete the custom service template by name using the following API information:

Name	Description
Resource URI	`https://doi.dxi-na1.saas.broadcom.com/oi/v2/template/deleteCustomTemplate\<Template_Name>` `http or https://<doi.adminui>/oi/v2/template/deleteCustomTemplate\<Template_Name>`
Method	DELETE
HTTP Headers	`Content-Type:application/json`
Authorization	`Bearer {bearerToken/userToken}`

Delete All Custom Service Templates

You can delete all the custom service templates at once using the following API information:

Name	Description
Resource URI	`https://doi.dxi-na1.saas.broadcom.com/oi/v2/template/deleteCustomTemplates` `http or https://<doi.adminui>/oi/v2/template/deleteCustomTemplates`
Method	DELETE
HTTP Headers	`Content-Type:application/json`
Authorization	`Bearer {bearerToken/userToken}`

Request	`{`
	`"names": [`
	`"<name of service template>"`
	`]`
	`}`

Get All Custom Service Templates

You can get all the service templates created by a user or tenant using the following information:

Name	Description
Resource URI	`https://doi.dxi-na1.saas.broadcom.com/oi/v2/template/allCustom` `http or https://<doi.adminui>/oi/v2/template/allCustom`
Method	GET
HTTP Headers	`Content-Type:application/json`
Authorization	`Bearer {bearerToken/userToken}`

Name	Description

Sample Request	{
	
	"attributes"
	: [
	
	"name"
	
	]
	}

Name	Description

Sample Response	{
	"vertices": [
	{
	"id": 21418,
	"externalId": "SATemplate:Rating Server Test gassela",
	"startTime": 1680677666380,
	"endTime": 1683269666380,
	"attributes": {
	"name": "Rating Server Test gassela"
	}
	},
	{
	"id": 27437,
	"externalId": "SATemplate:UIM_API4111",
	"startTime": 1680615235198,
	"endTime": 1683208985658,
	"attributes": {
	"name": "UIM_API4111"
	}
	},
	{
	"id": 27435,
	"externalId": "SATemplate:UIM_API41",
	"startTime": 1680605511958,
	"endTime": 1683197511958,
	"attributes": {
	"name": "UIM_API41"
	}
	},
	{
	"id": 27429,
	"externalId": "SATemplate:UIM_API",
	"startTime": 1680521958451,
	"endTime": 1683113958451,
	"attributes": {
	"name": "UIM_API"
	}
	},
	{
	"id": 27424,
	"externalId": "SATemplate:Spectrum Group",
	"startTime": 1678796317805,
	"endTime": 1681388317805,
	"attributes": {
	"name": "Spectrum Group"
	}
	},
	{
	"id": 27432,
	"externalId": "SATemplate:UIM_API2",
	"startTime": 1680527541773,
	"endTime": 1683119541773,
	"attributes": {
	"name": "UIM_API2"
	}
	},

Get Services for an Inventory

This API returns the service details for an inventory including the hierarchy (either parent or child services) along with service metrics.

Resource URI

You can select the URI based on the deployment and region of your tenant:

- DX SaaS - USA:
  `https://apmgw.dxi-na1.saas.broadcom.com/oi/v2/servicerepo/inventory/services/hierarchy`
- DX SaaS - EU:
  `https://apmgw.dxi-eu1.saas.broadcom.com/oi/v2/servicerepo/inventory/services/hierarchy`

Method

`POST`

HTTP Headers

- **Content-Type:** application/JSON
- **Authorization:** user token that can be generated from the Token Management page.

Attributes

Attribute	Mandatory	Type	Description
externalId	Yes	String	Inventory external ids. The values must be the same as
relationType	No	String	Possible values are [***parent, child and both]*** Default: **b**
projectionFilter	No	Array	List of the service attributes in the response. By default, Possible values: 1. "hierarchy", 2. "availability", 3. "health", 4. "name", 5. "maintenance", 6. "risk" 7. "situationCount" 8. "rollupFilteredAlarmCount" 9. "status"

Request Payload

```
{
"externalId": "<externalId-1>",
"relationType": "parent",
"projectionFilter":
   [
      "availability",
      "health",
      "name",
      "risk"
   ]
}
```

Response

Success Code: 200 OK


```
{
  "vertices": [
    {
      "externalId": "SA:33C09258-0116-4F25-BACF-2B4C19C6568F:2f7b87d0-299a-4080-b304-5193dfc38be6",
      "attributes": {
        "name": "AT&T",
        "risk": 0,
        "state": "ACTIVE",
        "health": 100,
        "alarms": 0
      },
      "child": [
        {
          "externalId": "SA:33C09258-0116-4F25-BACF-2B4C19C6568F:6a347554-a920-4cbf-bbd6-c25b8839d8d4",
          "attributes": {
            "name": "AT&T Northeast",
            "state": "ACTIVE",
            "health": 100,
            "risk": 0,
            "alarms": 0
          },
          "child": [
            {
              "externalId": "SA:33C09258-0116-4F25-BACF-2B4C19C6568F:22c4bb89-5d92-49b0-a0b3-bde562d53c8b",
              "attributes": {
                "name": "ATnT Newyark",
                "state": "ACTIVE",
                "health": 100,
                "risk": 2,
                "alarms": 1
              }
            },
            {
              "externalId": "SA:33C09258-0116-4F25-BACF-2B4C19C6568F:1212132-24232-424234-93e2-0fde8998f116",
              "attributes": {
                "name": "ATnT NEW England",
                "state": "ACTIVE",
                "health": 100,
                "risk": 0,
                "alarms": 0
              },
              "child": [
                {
                  "externalId": "SA:33C09258-0116-4F25-BACF-2B4C19C6568F:67f762da-1acf-448a-
b235-2579c0ec48c0",
                  "attributes": {
                    "name": "Site A Circuit X",
                    "state": "ACTIVE",
                    "health": 100,
                    "risk": 0,
```

```
                    "alarms": 0
                  }
                },
                {
                  "externalId": "SA:33C09258-0116-4F25-
BACF-2B4C19C6568F:436b79f6-661e-4e00-93e2-0fde8998f116",
                  "attributes": {
                    "name": "Site A Circuit Y",
                    "state": "ACTIVE",
                    "health": 100,
                    "risk": 0,
                    "alarms": 0
                  },
                  "child": [
                    {
                      "externalId": "SA:33C09258-0116-4F25-BACF-2B4C19C6568F:1d01a5ec-06e3-41e5-
ba9a-01cedfbc8d3f",
                      "attributes": {
                        "name": "CE Port",
                        "state": "ACTIVE",
                        "health": 100,
                        "risk": 0,
                        "alarms": 0
                      }
                    },
                    {
                      "externalId": "SA:33C09258-0116-4F25-BACF-2B4C19C6568F:4cd311be-d82f-47e6-
a645-7d0e5af6b304",
                      "attributes": {
                        "name": "Metro Ethernet",
                        "state": "ACTIVE",
                        "health": 100,
                        "risk": 0,
                        "alarms": 0
                      }
                    },
                    {
                      "externalId": "SA:33C09258-0116-4F25-
BACF-2B4C19C6568F:39ea60f5-1768-4314-9e07-86cd4da2e667",
                      "attributes": {
                        "name": "Backbone",
                        "state": "ACTIVE",
                        "health": 100,
                        "risk": 0,
                        "alarms": 0
                      }
                    }
                  ]
                },
                {
                  "externalId": "SA:33C09258-0116-4F25-BACF-2B4C19C6568F:67f762da-1a3cf-448a-
b2335-2579c0ec6567",
                  "attributes": {
```

```
                    "name": "Site B Circuit Z",
                    "state": "ACTIVE",
                    "health": 100,
                    "risk": 0,
                    "alarms": 0
                  }
                },
                {
                  "externalId": "SA:33C09258-0116-4F25-BACF-2B4C19C6568F:67f762da-1a3cf-448a-
b2335-2579c0ec6567",
                  "attributes": {
                    "name": "Site C Circuit Q",
                    "state": "ACTIVE",
                    "health": 100,
                    "risk": 0,
                    "alarms": 0
                  }
                }
              ]
            }
          ]
        },
        {
          "externalId": "SA:33C09258-0116-4F25-BACF-2B4C19C6568F:0fde8998f116-661e-4e00-93e2-0fde8998f116",
          "attributes": {
            "name": "A Southeast",
            "state": "ACTIVE",
            "health": 100,
            "risk": 0,
            "alarms": 0
          }
        }
      ]
    }
  ]
}
```


Error Codes:

- **500**: internal server error
- **403:** unauthorized

Example

If an inventory(externalId-1) is part of the "Backbone" service and the user calls the API request as below, then the API response contains all the parent services which belong to the direct hierarchy.

Request

```
{
   "externalId": "<externalId-1>",
   "relationType": "parent",
   "projectionFilter":
      [
```

```
        "state",
        "health",
        "name",
        "risk"
      ]
}
```


Response


```
{
  "vertices": [
    {
      "externalId": "SA:33C09258-0116-4F25-BACF-2B4C19C6568F:2f7b87d0-299a-4080-b304-5193dfc38be6",
      "attributes": {
        "name": "AT&T",
        "risk": 0,
        "state": "ACTIVE",
        "health": 100,
        "alarms": 0
      },
      "child": [
        {
          "externalId": "SA:33C09258-0116-4F25-BACF-2B4C19C6568F:6a347554-a920-4cbf-bbd6-c25b8839d8d4",
          "attributes": {
            "name": "AT&T Northeast",
            "state": "ACTIVE",
            "health": 100,
            "risk": 0,
            "alarms": 0
          },
          "child": [
            {
              "externalId": "SA:33C09258-0116-4F25-BACF-2B4C19C6568F:1212132-24232-424234-93e2-0fde8998f116",
              "attributes": {
                "name": "ATnT NEW England",
                "state": "ACTIVE",
                "health": 100,
                "risk": 0,
                "alarms": 0
              },
              "child": [
                {
                  "externalId": "SA:33C09258-0116-4F25-
BACF-2B4C19C6568F:436b79f6-661e-4e00-93e2-0fde8998f116",
                  "attributes": {
                    "name": "Site A Circuit Y",
                    "state": "ACTIVE",
                    "health": 100,
                    "risk": 0,
                    "alarms": 0
                  },
                  "child": [
                    {
```

```
                      "externalId": "SA:33C09258-0116-4F25-
BACF-2B4C19C6568F:39ea60f5-1768-4314-9e07-86cd4da2e667",
                      "attributes": {
                        "name": "Backbone",
                        "state": "ACTIVE",
                        "health": 100,
                        "risk": 0,
                        "alarms": 0
                      }
                    }
                  ]
                }
              ]
            }
          ]
        }
      ]
    }
  ]
}
```


ServiceRepo APIs

The ServiceRepo APIs maintain tenant-wise services, its parent hierarchy, and metric values, and refresh the services every 10 minutes or when there is a service change. The following are the list of ServiceRepo APIs:

- GET Service Filter Options
- POST Services
- POST Group for a Tenant
- Add Attributes to Service Filter

GET Service Filter Options

This API returns the service with the filters applied.

Resource URI

You can select the URI based on the deployment and region of your tenant:

- DX SaaS - USA:
  `https://apmgw.dxi-na1.saas.broadcom.com/oi/v2/servicerepo/filters`
- DX SaaS - EU:
  `https://apmgw.dxi-eu1.saas.broadcom.com/oi/v2/servicerepo/filters`

Method

`GET`

HTTP Headers

- **Content-Type:** application/JSON
- **Authorization:** user token that can be generated from the Token Management page.

Request

```
https://apmgw.dxi-eu1.saas.broadcom.com/oi/v2/servicerepo/filters
```

Response

```
{
  "filters": [
    {
      "isCustom": <true/false>,
      "field": "<fieldname>",
      "fieldDescription": "<fielddesc>",
      "hideMoreOptions": <true/false>,
      "values": []
    },
    {
      "isCustom": <true/false>,
      "field": "<fieldname>",
      "fieldDescription": "<fielddesc>",
      "hideMoreOptions": <true/false>,
      "values": [
        "
      ]
    },

  ]
  "groupFields": []
}
```

Example

Resource URI

```
https://apmgw.dxi-na1.saas.broadcom.com/oi/v2/servicerepo/filters
```

Response

```
{
  "filters": [
    {
      "isCustom": false,
      "field": "name",
      "fieldDescription": "Service Name",
      "hideMoreOptions": false,
      "values": []
    },
    {
      "isCustom": false,
      "field": "risk",
      "fieldDescription": "Risk",
```

```
      "hideMoreOptions": true,
      "values": [
        "Severe",
        "High",
        "Moderate",
        "Slight",
        "None",
        "Unknown"
      ]
    },
    {
      "isCustom": false,
      "field": "health",
      "fieldDescription": "Health",
      "hideMoreOptions": true,
      "values": [
        "Bad",
        "Average",
        "Good",
        "Unknown"
      ]
    },
    {
      "isCustom": false,
      "field": "availability",
      "fieldDescription": "Availability",
      "hideMoreOptions": true,
      "values": [
        "Down",
        "Maintenance",
        "Good",
        "Unknown"
      ]
    },
    {
      "isCustom": false,
      "field": "maintenance",
      "fieldDescription": "Maintenance",
      "hideMoreOptions": true,
      "values": [
        "False",
        "True"
      ]
    },
    {
      "isCustom": false,
      "field": "location",
      "fieldDescription": "Location",
      "hideMoreOptions": false,
      "values": [
        "CANADA",
        "GREENLAND",
        "RUSSIAN FEDERATION",
```

```
      "UNITED STATES"
    ]
  },
  {
    "isCustom": false,
    "field": "tags",
    "fieldDescription": "Tags associated with the service",
    "hideMoreOptions": false,
    "values": [
      "child",
      "parent",
      "test",
      "test-2",
      "v"
    ]
  },
  {
    "isCustom": false,
    "field": "separator",
    "fieldDescription": "CustomProperties",
    "hideMoreOptions": false,
    "values": []
  },
  {
    "isCustom": true,
    "field": "asd",
    "fieldDescription": "asd",
    "hideMoreOptions": false,
    "values": [
      "asdas"
    ]
  },
  {
    "isCustom": true,
    "field": "custom1",
    "fieldDescription": "custom1",
    "hideMoreOptions": false,
    "values": [
      "child",
      "parent"
    ]
  },
  {
    "isCustom": true,
    "field": "custom2",
    "fieldDescription": "custom2",
    "hideMoreOptions": false,
    "values": [
      "test",
      "test"
    ]
  },
  {
```

```
      "isCustom": true,
      "field": "fsdfsdf",
      "fieldDescription": "fsdfsdf",
      "hideMoreOptions": false,
      "values": [
        "sdfsdf"
      ]
    },
    {
      "isCustom": true,
      "field": "property-1",
      "fieldDescription": "property-1",
      "hideMoreOptions": false,
      "values": [
        "value-1"
      ]
    },
    {
      "isCustom": true,
      "field": "property-2",
      "fieldDescription": "property-2",
      "hideMoreOptions": false,
      "values": [
        "value-2"
      ]
    }
  ],
  "groupFields": []
}
```


POST Services

This API returns the service with filters, sorted and paginated list of services.

Resource URI

You can select the URI based on the deployment and region of your tenant:

- DX SaaS - USA:
  `https://apmgw.dxi-na1.saas.broadcom.com/oi/v2/servicerepo/services`
- DX SaaS - EU:
  `https://apmgw.dxi-eu1.saas.broadcom.com/oi/v2/servicerepo/services`

Method

`POST`

HTTP Headers

- **Content-Type:** application/JSON
- **Authorization:** user token that can be generated from the Token Management page.

Request

```
https://apmgw.dxi-eu1.saas.broadcom.com/oi/v2/servicerepo/services
```

Payload Syntax

```
{
  "time": 1601942403254,
  "filters": [
    {
      "fieldDescription": "<fielddescription>",
      "field": "<fieldname>",
      "value": "<>",
      "condition": "",
      "isCustom":
    },
    {
      "fieldDescription": "<fielddescription>",
      "field": "<fieldname>",
      "value": "",
      "condition": "",
      "isCustom":
    },
    {
      "fieldDescription": "<fielddescription>",
      "field": "<fieldname>",
      "value": "",
      "condition": "",
      "isCustom":
    }
  ],
  "sortField": "",
  "sortOrder": "",
  "groupServices": ,
  "pageNum": ,
  "pageSize":
}
```

Response

```
{
    "services": [
        {
            "name": "<servicename>",
            "status": "<statusofservice>",
            "alarmCount": ,
            "alarmFirstTs": ,
            "parents": [
            {
                "name": "<servicename>",
                "status": "<statusofservice>",
                "alarmCount": ,
                "alarmFirstTs": ,
                "isAvailabilityConfigured": <true/false>,
                "id": "<id>"
```

```
            },
            {
                "name": "<servicename>",
                "status": <statusofservice>",
                "alarmCount": ,
                "alarmFirstTs": ,
                "parents": [
                {
                    "name"
      "status": "ACTIVE",
                    "alarmCount": 0,
                    "alarmFirstTs": 0,
                    "isAvailabilityConfigured": <true/false>,
                    "id": "<id>"
                }
                ],
                "isAvailabilityConfigured": <true/false>,
                "id": "<id>"
            }
            ],
            "isAvailabilityConfigured": <true/false>,
            "location": "",
            "maintenance": <true/false>,
            "risk": 0,
            "health": 100,
            "tags": [],
            "trend": {
                "alarms": "",
                "prediction": "",
                "health": "",
                "risk": "",
                "availability": ""
            },
            "serviceApplicableMetrics": {
                "prediction": [
                    "fx-AAB-L-7jItNq"
                ],
                "health": [
                    "dx-AE-L-0AX0uo",
                    "ex-AAB-L-fPOZno"
                ]
            },
            "customProperties": {},
            "id": ""
        },
    .........
    ],
    "filteredCnt": <count>,
    "totalCnt": <count>
}
```

Example

Resource URI

```
https://apmgw.dxi-na1.saas.broadcom.com/oi/v2/servicerepo/services
```

Payload


```
{
  "time": 1601942403254,
  "filters": [
    {
      "fieldDescription": "Risk",
      "field": "risk",
      "value": "Severe",
      "condition": "equals",
      "isCustom": false
    },
    {
      "fieldDescription": "Risk",
      "field": "risk",
      "value": "Moderate",
      "condition": "equals",
      "isCustom": false
    },
    {
      "fieldDescription": "Name",
      "field": "name",
      "value": "AAA",
      "condition": "not_contains",
      "isCustom": false
    }
  ],
  "sortField": "health",
  "sortOrder": "ascending",
  "groupServices": false,
  "pageNum": 1,
  "pageSize": 5
}
```


Response


```
{
    "services": [
        {
            "name": "BRT_web_app_svc",
            "status": "ACTIVE",
            "alarmCount": 0,
            "alarmFirstTs": 0,
            "parents": [
            {
                "name": "New Hier",
                "status": "ACTIVE",
                "alarmCount": 0,
                "alarmFirstTs": 0,
```

```
                "isAvailabilityConfigured": false,
                "id": "SA:65CC1F1F-4594-4238-A0DC-BA39285E39DC:a108027d-18dc-4f6f-aaf9-fc52967bbfb3"
            },
            {
                "name": "WebApp",
                "status": "ACTIVE",
                "alarmCount": 0,
                "alarmFirstTs": 0,
                "parents": [
                {
                    "name": "Other Services",
                    "status": "ACTIVE",
                    "alarmCount": 0,
                    "alarmFirstTs": 0,
                    "isAvailabilityConfigured": false,
                    "id": "SA:65CC1F1F-4594-4238-A0DC-BA39285E39DC:751d6545-804c-4141-a2a9-7b56859f9501"
                }
                ],
                "isAvailabilityConfigured": false,
                "id": "SA:65CC1F1F-4594-4238-A0DC-BA39285E39DC:d1916914-662b-442c-9244-1c716e304d7c"
            }
            ],
            "isAvailabilityConfigured": false,
            "location": "",
            "maintenance": false,
            "risk": 0,
            "health": 100,
            "tags": [],
            "trend": {
                "alarms": "up",
                "prediction": "up",
                "health": "up",
                "risk": "down",
                "availability": "up"
            },
            "serviceApplicableMetrics": {
                "prediction": [
                    "fx-AAB-L-7jItNq"
                ],
                "health": [
                    "dx-AE-L-0AX0uo",
                    "ex-AAB-L-fPOZno"
                ]
            },
            "customProperties": {},
            "id": "SA:65CC1F1F-4594-4238-A0DC-BA39285E39DC:d61b8dff-99ce-4414-b57f-5a2d3c49749b"
        },
    .........
    ],
    "filteredCnt": 20,
    "totalCnt": 150
}
```

POST Group for a Tenant

This API returns the service group of a service that is associated with a tenant. The group field attribute must be as follows:

- risk
- health
- availability

Resource URI

You can select the URI based on the deployment and region of your tenant:

- DX SaaS - USA:
  `https://apmgw.dxi-na1.saas.broadcom.com/oi/v2/servicerepo/group?tenant_id=`
- DX SaaS - EU:
  `https://apmgw.dxi-eu1.saas.broadcom.com/oi/v2/servicerepo/group?tenant_id=`

Method

```
POST
```

HTTP Headers

- **Content-Type:** application/JSON
- **Authorization:** user token that can be generated from the Token Management page.

Request

```
https://apmgw.dxi-eu1.saas.broadcom.com/oi/v2/servicerepo/group?tenant_id=
```

Payload Syntax

```
{
  "time": 1601942403254,
  "filters": [
    {
      "fieldDescription": "Name",
      "field": "name",
      "value": "AAA",
      "condition": "not_contains",
      "isCustom": false
    }
  ],
  "groupServices": false,
  "groupFields": [
    "risk",
    "health",
    "availability"
  ]
}
```

Response

```
    {
```

```
"useAvailabilityStatus": true,
"metrics": {
  "health": [
    {
      "name": "Bad",
      "displayName": "Levels > 0.0< 75.0 ",
      "value":
    },
    {
      "name": "Average",
      "displayName": "Levels > 76.0 < 95.0 ",
      "value": 0
    },
    {
      "name": "Good",
      "displayName": "Levels > 96.0 < 100.0 ",
      "value": 170
    },
    {
      "name": "Unknown",
      "displayName": "Health still not computed as a it is a new service",
      "value": 0
    }
  ],
  "risk": [
    {
      "name": "Severe",
      "displayName": "Level = 4",
      "value": 0
    },
    {
      "name": "High",
      "displayName": "Level = 3",
      "value": 0
    },
    {
      "name": "Moderate",
      "displayName": "Level = 2",
      "value": 10
    },
    {
      "name": "Slight",
      "displayName": "Level = 1",
      "value": 1
    },
    {
      "name": "None",
      "displayName": "Levels = 0",
      "value": 159
    },
    {
      "name": "Unknown",
      "displayName": "Risk Level still not computed as a it is a new service",
```

```
        "value": 0
      }
    ],
    "availability": [
      {
        "name": "Down",
        "displayName": "Levels > ",
        "value": 3
      },
      {
        "name": "Maintenance",
        "displayName": "Levels > ",
        "value": 0
      },
      {
        "name": "Good",
        "displayName": "Good > ",
        "value": 0
      },
      {
        "name": "Unknown",
        "displayName": "Availability still not computed as it is a new service",
        "value": 167
      }
    ]
  }
}
```


Example

Resource URI

```
https://apmgw.dxi-na1.saas.broadcom.com/oi/v2/servicerepo/services
```

Payload


```
{
  "time": 1601942403254,
  "filters": [
    {
      "fieldDescription": "Name",
      "field": "name",
      "value": "AAA",
      "condition": "not_contains",
      "isCustom": false
    }
  ],
  "groupServices": false,
  "groupFields": [
    "risk",
    "health",
```

```
    "availability"
  ]
}
```


Response


```
{
  "useAvailabilityStatus": true,
  "metrics": {
    "health": [
      {
        "name": "Bad",
        "displayName": "Levels > 0.0< 75.0 ",
        "value":
      },
      {
        "name": "Average",
        "displayName": "Levels > 76.0 < 95.0 ",
        "value": 0
      },
      {
        "name": "Good",
        "displayName": "Levels > 96.0 < 100.0 ",
        "value": 170
      },
      {
        "name": "Unknown",
        "displayName": "Health still not computed as a it is a new service",
        "value": 0
      }
    ],
    "risk": [
      {
        "name": "Severe",
        "displayName": "Level = 4",
        "value": 0
      },
      {
        "name": "High",
        "displayName": "Level = 3",
        "value": 0
      },
      {
        "name": "Moderate",
        "displayName": "Level = 2",
        "value": 10
      },
      {
        "name": "Slight",
        "displayName": "Level = 1",
        "value": 1
      },
      {
```

```
        "name": "None",
        "displayName": "Levels = 0",
        "value": 159
      },
      {
        "name": "Unknown",
        "displayName": "Risk Level still not computed as a it is a new service",
        "value": 0
      }
    ],
    "availability": [
      {
        "name": "Down",
        "displayName": "Levels > ",
        "value": 3
      },
      {
        "name": "Maintenance",
        "displayName": "Levels > ",
        "value": 0
      },
      {
        "name": "Good",
        "displayName": "Good > ",
        "value": 0
      },
      {
        "name": "Unknown",
        "displayName": "Availability still not computed as it is a new service",
        "value": 167
      }
    ]
  }
}
```


Add Attributes to Service Filter

This API allows you to add the following attributes/fields to the filter payload for a service.

- health
- risk
- availability

Resource URI

You can select the URI based on the deployment and region of your tenant:

- DX SaaS - USA:
  `https://apmgw.dxi-na1.saas.broadcom.com/oi/v2/servicerepo/services`
- DX SaaS - EU:
  `https://apmgw.dxi-eu1.saas.broadcom.com/oi/v2/servicerepo/services`

Method

```
POST
```

HTTP Headers

- **Content-Type:** application/JSON
- **Authorization:** user token that can be generated from the Token Management page.

Request

```
https://apmgw.dxi-eu1.saas.broadcom.com/oi/v2/servicerepo/services
```

Payload Syntax

```
                "time": 1599158925523,
                "filters": [
    .
    .
    .],
    "groupServices": true,
    "sortField": "health",// "selectedKPI": 'KPI Selection' possible values are health, risk and availability
    "sortOrder": "asc",
    "pageNum": 1,
    "pageSize": 5
}
```

Response

```
[
    {
        "parentIds": [
            "SA:01AA27AA-1A9A-4957-A66A-2856DB5A2DCE:bb5ee57b-c630-4445-ae53-55c682393b40",
            "SA:01AA27AA-1A9A-4957-A66A-2856DB5A2DCE:0bf3d929-00a8-4740-b471-cc6d525c7260",
            "SA:01AA27AA-1A9A-4957-A66A-2856DB5A2DCE:6c5618c3-ba8d-42c1-9a96-11853888321e"
        ],
        "alarms": 0,
        "childAlarms": 0,
        "users": 0,
        "name": "12 Aug",
        "tags": [],
        "description": "",
        "customProperties": [
            {
                "name": "dasd",
                "value": "asdas"
            }
        ],
        "revenue": 0,
        "risk": 0,
        "sentiment": null,
        "isRoot": false,
        "isAvailabilityConfigured": false,
        "availability": null,
        "availability_lastday": null,
```

```
        "upChildren": 0,
        "health": 100,
        "id": "SA:01AA27AA-1A9A-4957-A66A-2856DB5A2DCE:be179e41-5c1a-459c-a224-2605485a3342",
        "children": [
            "SA:01AA27AA-1A9A-4957-A66A-2856DB5A2DCE:a260e979-4f94-4393-b587-0ca439a9a827",
            "SA:01AA27AA-1A9A-4957-A66A-2856DB5A2DCE:0cdbb0d2-9f05-4131-ab65-dc9de15ffcdd",
            "SA:01AA27AA-1A9A-4957-A66A-2856DB5A2DCE:7340efb6-d010-4532-bc43-3fe4d8a04041"
        ],
        "conversion": 0,
        "retention": 0,
        "status": "ACTIVE",
        "prediction": null,
        "trend": {
            "revenue": "up",
            "alarms": "up",
            "prediction": "up",
            "health": "up",
            "risk": "up",
            "availability": "up",
            "users": "up",
            "retention": "up"
        },
        "maintenance": false,
        "firstRawAlarmDate": null,
        "location": "",
        "indexId": "SA:01AA27AA-1A9A-4957-A66A-2856DB5A2DCE:be179e41-5c1a-459c-a224-2605485a3342:",
        "allChildAlarms": 0
    },
    {
```


Example

Resource URI

```
https://apmgw.dxi-na1.saas.broadcom.com/oi/v2/servicerepo/services
```

Payload


```
                "time": 1599158925523,
                "filters": [
    .
    .
    .],
    "groupServices": true,
    "sortField": "health",// "selectedKPI": 'KPI Selection' possible values are health, risk and availability
    "sortOrder": "asc",
    "pageNum": 1,
    "pageSize": 5
```


Response


```
[
    {
```

```
        "parentIds": [
            "SA:01AA27AA-1A9A-4957-A66A-2856DB5A2DCE:bb5ee57b-c630-4445-ae53-55c682393b40",
            "SA:01AA27AA-1A9A-4957-A66A-2856DB5A2DCE:0bf3d929-00a8-4740-b471-cc6d525c7260",
            "SA:01AA27AA-1A9A-4957-A66A-2856DB5A2DCE:6c5618c3-ba8d-42c1-9a96-11853888321e"
        ],
        "alarms": 0,
        "childAlarms": 0,
        "users": 0,
        "name": "12 Aug",
        "tags": [],
        "description": "",
        "customProperties": [
            {
                "name": "dasd",
                "value": "asdas"
            }
        ],
        "revenue": 0,
        "risk": 0,
        "sentiment": null,
        "isRoot": false,
        "isAvailabilityConfigured": false,
        "availability": null,
        "availability_lastday": null,
        "upChildren": 0,
        "health": 100,
        "id": "SA:01AA27AA-1A9A-4957-A66A-2856DB5A2DCE:be179e41-5c1a-459c-a224-2605485a3342",
        "children": [
            "SA:01AA27AA-1A9A-4957-A66A-2856DB5A2DCE:a260e979-4f94-4393-b587-0ca439a9a827",
            "SA:01AA27AA-1A9A-4957-A66A-2856DB5A2DCE:0cdbb0d2-9f05-4131-ab65-dc9de15ffcdd",
            "SA:01AA27AA-1A9A-4957-A66A-2856DB5A2DCE:7340efb6-d010-4532-bc43-3fe4d8a04041"
        ],
        "conversion": 0,
        "retention": 0,
        "status": "ACTIVE",
        "prediction": null,
        "trend": {
            "revenue": "up",
            "alarms": "up",
            "prediction": "up",
            "health": "up",
            "risk": "up",
            "availability": "up",
            "users": "up",
            "retention": "up"
        },
        "maintenance": false,
        "firstRawAlarmDate": null,
        "location": "",
        "indexId": "SA:01AA27AA-1A9A-4957-A66A-2856DB5A2DCE:be179e41-5c1a-459c-a224-2605485a3342:",
        "allChildAlarms": 0
    },
    {
```

Situation Alarm Action APIs

DX Operational intelligence provides list of action APIs that the automated scripts can call to perform actions on situation alarms. Automation engineers can run these automated scripts manually, through UI or through ITSM policies to perform actions on the situations.

Using Situation action APIs in your automation scripts, you can perform the following actions on situation alarms:

- Acknowledge or unacknowledge a situation
- Assign or unassign (dissociate) a user for a situation
- Annotate Situation with useful notes and comments
- Close or Resolve a Situation
- Create Ticket in ServiceNow for unresolved issues
- Retrieve the situation alarms based on the severity and status that are associated with an alarm cluster

The Situation Alarm APIs are:

- Acknowledge API
- Unacknowledge API
- Assign API
- Unassign API
- Annotation API
- Close API
- Close Ticket API
- Cluster Alarms API
- Cluster Host API

Acknowledge API

You can perform the acknowledgment action on a situation alarm by calling the Acknowledge API in your automation scripts. When you send an API request with relevant payload details using the Acknowledge API, DX Operational Intelligence updates the acknowledgment status for the given situational alarm as acknowledged.

Resource URI

You can select the URI based on the deployment and region of your tenant:

- DX SaaS - USA:
  `https://doi.dxi-na1.saas.broadcom.com/oi/v2/api/alarmactions/ack`
- DX SaaS - EU:
  `https://doi.dxi-eu1.saas.broadcom.com/oi/v2/api/alarmactions/ack`

```
https://<oi_host>:<oi_port>/oi/v2/api/alarmactions/ack
```

Method

```
POST
```

HTTP Headers

- Authorization: Bearer {token}
  For more information on tokens, see the Token Management section.

For more information on tokens, see the Token Management section.

- Content Type: Application/JSON

Request Payload Syntax

```
{
"alarmDetails":[
{
"alarmId":"<Alarm ID>",
"alarmType":"situation"
}
],
"actionDetails":{
"actionType":"acknowledge"
}
}
```

Table 15: Parameters

Parameter	Description
alarmId	Sends the situation (alarm) ID on which you want to perform the Acknowledgement action.
Alarmtype	Sends the alarm type as “Situation”. DX Operational Intelligenceverifies the alarm type as “Situation” before updating the Acknowledgement status.
actionType	Sends the action type as “acknowledge”. DX Operational Intelligence updates the acknowledgement status of the alarm as acknowledged when the alarm type is “Situation”.

Response Syntax

```
{
"actionStatus": [
{
"alarmId": "<Alarm ID>",
"status": "Success",
"updatedFields": {
"acknowledged": "true",
"acknowledgedBy": "<User ID>",
"lastUpdateTime": "<Last Updated Time>"
},
"ignore": false
}
]
}
```

Sample Request-Response

Sample URI

```
https://doi.dxi-na1.saas.broadcom.com/oi/v2/api/alarmactions/ack
```

Sample Payload

```
{
"alarmDetails":[
{
"alarmId":"1751330",
"alarmType":"situation"
}
],
"actionDetails":{
"actionType":"acknowledge"
}
}
```

Sample Response

```
"alarmId":"1751330",
"alarmType":"situation"
}
],
"actionDetails":{
"actionType":"acknowledge"
}
}    {
"actionStatus": [
{
"alarmId": "1751330",
"status": "Success",
"updatedFields": {
"acknowledged": "true",
"acknowledgedBy": "TADMIN1",
"lastUpdateTime": "1625230120274"
},
"ignore": false
}
]
}
```

Unacknowledge API

You can unacknowledge a situation alarm by calling the Unacknowledge API in your automation scripts. When you send a request with relevant payload details using the Unacknowledge API, DX Operational Intelligence updates the acknowledgement status for the given situational alarm as unacknowledged.

Resource URI

You can select the URI based on the deployment and region of your tenant:

- DX SaaS - USA:
  `https://doi.dxi-na1.saas.broadcom.com/oi/v2/api/alarmactions/ack`
- DX SaaS - EU:
  `https://doi.dxi-eu1.saas.broadcom.com/oi/v2/api/alarmactions/ack`

```
https://<oi_host>:<oi_port>/oi/v2/api/alarmactions/ack
```

Method

```
POST
```

HTTP Headers

- Authorization: Bearer {token}
  For more information on tokens, see the Token Management section.
  For more information on tokens, see the Token Management section.
- Content Type: Application/JSON

Request Payload Syntax

```
{
"alarmDetails":[
{
"alarmId":<"Alarm ID”>,
"alarmType":"situation”
}
],
"actionDetails":{
"actionType":" unAcknowledge"
}
}
```

Table 16: Parameters

Parameter	Description
alarmId	Sends the alarm ID on which you want to perform the unacknowledgement action.
Alarmtype	Sends the alarm type as **Situation**. DX Operational Intelligence verifies the alarm type as **Situation** before updating the acknowledgement status.
actionType	Sends the action type as **unacknowledge**. DX Operational Intelligence updates the acknowledgement status of a situation alarm as unacknowledged, when the alarm type is **Situation**.

Response Syntax

```
{
"actionStatus": [
{
"alarmId": <"Alarm ID">,
"status": "Success",
"updatedFields": {
"acknowledged": "false",
"acknowledgedBy": "<User ID>",
"lastUpdateTime": "<Time Stamp"
},
"ignore": false
}
]
}
```

Sample Request-Response

Sample URI

```
https://doi.dxi-na1.saas.broadcom.com/oi/v2/api/alarmactions/ack
```

Sample Payload

```
{
"alarmDetails":[
{"alarmId":"1751330",
"alarmType":"situation"}],
"actionDetails":{"actionType":"unAcknowledge"}
}
}
```

Sample Response

```
{
"actionStatus": [
{
"alarmId": "1751330",
"status": "Success",
"updatedFields": {
"acknowledged": "false",
"acknowledgedBy": "TADMIN1",
"lastUpdateTime": "1625230223159"
},
"ignore": false
}
]
}
```

Assign API

You can assign a situation alarm to a user by calling the Assign API in your automation scripts. When you send a request with relevant payload details using the Assign API, DX Operational Intelligence assigns the alarm to the specified user for resolution.

Resource URI

You can select the URI based on the deployment and region of your tenant:

- DX SaaS - USA:
  `https://doi.dxi-na1.saas.broadcom.com/oi/v2/api/alarmactions/assign`
- DX SaaS - EU:
  `https://doi.dxi-eu1.saas.broadcom.com/oi/v2/api/alarmactions/assign`

```
https://<oi_host>:<oi_port>/oi/v2/api/alarmactions/assign
```

Method

```
POST
```

HTTP Headers

- Authorization: Bearer {token}
  For more information on tokens, see the Token Management section.

For more information on tokens, see the Token Management section.

- Content Type: Application/JSON

Request Payload Syntax

```
{
"alarmDetails":[
{"alarmId":<"Alarm ID">,
"alarmType":"situation"}],
"actionDetails":{
"actionType":"assignment",
"target":[<"User ID">]
}
}
```

Table 17: Parameters

Parameter	Description
alarmId	Sends the Situation alarm ID to which you want to assign a user.
Alarmtype	Sends the alarm type as **situation**. DX Operational Intelligence verifies the alarm type as **situation** before assigning the user ID.
actionType	Sends the action type as **assignment**. DX Operational Intelligence assigns the alarm to the specified user ID, when the alarm ID with alarm type is **situation**.
target	Sends the target user ID. DX Operational Intelligence assigns the alarm to the target user ID for resolution.

Response Syntax

```
{
"actionStatus": [
{
"alarmId": <"Alarm ID">,
"status": "Success",
"updatedFields": {
"troubleShooter": <"Target User ID>",
"assignedBy": <"User ID">,
"lastUpdateTime": <"Time stamp">
},
"ignore": false
}
]
}
```

Sample Request-Response

Sample URI

```
https://doi.dxi-na1.saas.broadcom.com/oi/v2/api/alarmactions/assign
```

Sample Payload

```
{"alarmDetails":[
{"alarmId":"1751330",
```

```
"alarmType":"situation"}],
"actionDetails":{
"actionType":"assignment",
"target":["TADMIN1"]
}
}
```

Sample Response

```
{
"actionStatus": [
{
"alarmId": "1751330",
"status": "Success",
"updatedFields": {
"troubleShooter": "TADMIN1",
"assignedBy": "TADMIN1",
"lastUpdateTime": "1625230591753"
},
"ignore": false
}
]
}
```

Unassign API

You can dissociate or remove a user from a situation alarm by calling the Unassign API in your automation scripts. When you send a request with relevant payload details using the Unassign API, DX Operational Intelligence dissociates the assigned user from an alarm.

Resource URI

You can select the URI based on the deployment and region of your tenant:

- DX SaaS - USA:
  `https://doi.dxi-na1.saas.broadcom.com/oi/v2/api/alarmactions/assign`
- DX SaaS - EU:
  `https://doi.dxi-eu1.saas.broadcom.com/oi/v2/api/alarmactions/assign`

```
https://<oi_host>:<oi_port>/oi/v2/api/alarmactions/assign
```

Method

```
POST
```

HTTP Headers

- Authorization: Bearer {token}
  For more information on tokens, see the Token Management section.
  For more information on tokens, see the Token Management section.
- Content Type: Application/JSON

Request Payload Syntax

```
{
"alarmDetails":[
```

```
{"alarmId":”<Alarm ID>",
"alarmType":"situation"}],
"actionDetails":{
"actionType":"unAssignment",
}
}
```

Table 18: Parameters

Parameter	Description
alarmId	Sends an alarm ID for which you want to dissociate or remove the assigned user.
Alarmtype	Sends the alarm type as **situation**. DX Operational Intelligence verifies the alarm type as **situation** before dissociating the user.
actionType	Sends the action type as **unAssignment**. DX Operational Intelligencedissociates the user from the alarm when the alarm type is **situation**.

Response Syntax

```
{
"actionStatus": [
{
"alarmId": <"Alarm ID">,
"status": "Success",
"updatedFields": {
"troubleShooter": "",
"assignedBy": "<User ID>",
"lastUpdateTime": "time stamp"
},
"ignore": false
}
]
}
```

Sample Request-Response

Sample URI

```
https://doi.dxi-na1.saas.broadcom.com/oi/v2/api/alarmactions/assign
```

Sample Payload

```
{
"alarmDetails":[
{"alarmId":"1751330",
"alarmType":"situation"}],
"actionDetails":{"actionType":"unAssignment"}
}
```

Sample Response

```
{
"actionStatus": [
{
```

```
"alarmId": "1751330",
"status": "Success",
"updatedFields": {
"troubleShooter": "",
"assignedBy": "TADMIN1",
"lastUpdateTime": "1625230683075"
},
"ignore": false
}
]
}
```

Annotation API

You can annotate a situation cluster by calling the Annotation API in your automation scripts. When you send a request with relevant payload details using the Annotation API, DX Operational Intelligenceannotates the situation cluster.

Resource URI

You can select the URI based on the deployment and region of your tenant:

- DX SaaS - USA:
  `https://doi.dxi-na1.saas.broadcom.com/oi/v2/api/alarms/annotations`
- DX SaaS - EU:
  `https://doi.dxi-eu1.saas.broadcom.com/oi/v2/api/alarms/annotations`

```
https://<oi_host>:<oi_port>/oi/v2/api/alarms/annotations
```

Method

```
POST
```

HTTP Headers

- Authorization: Bearer {token}
  For more information on tokens, see the Token Management section.
  For more information on tokens, see the Token Management section.
- Content Type: Application/JSON

Request Payload Syntax

```
[{
"annotation":"<Annotation>",
"alarmType":"SituationCluster",
"alarm_unique_id":<"Alarm ID>",
"ticketID":<"Ticket ID">
"createdBy":<"User ID">}
]
```

Request Payload Syntax if the Ticket ID is Blank

```
[{
"annotation":"<Annotation>",
"alarmType":"SituationCluster",
```

```
"alarm_unique_id":<"Alarm ID>",
"createdBy":<"User ID">}
]
```

Table 19: Parameters

Parameter	Description
Alarmtype	Sends the alarm type as **SituationCluster**. DX Operational Intelligence verifies the alarm type as **SituationCluster** before updating the alarm with annotation.
alarm_unique_id	Sends the alarm ID for which you want to add the annotation.
ticketID (Optional)	Sends the ticket ID of the alarm in which DX Operational Intelligence must add the annotation. If the ticket ID is blank, DX Operational Intelligence does not update annotation in the associated ticket.
createdBy	Sends the user ID who annotates the alarm.

Response Syntax

```
{
"IngestionSuccessfulAnnotations": [
{
"annotation": "This is a sample annotation",
"alarmType": "SituationCluster",
"startTime": <"Time Stamp">,
"timestamp": <"Time Stamp">,
"createdBy": <"User ID">,
"updatedBy": null,
"alarm_unique_id": <"Alarm ID">,
"annotation_id": <"Annotation ID ">
}
]
}
```

Sample Request-Response

Sample URI

```
https://doi.dxi-na1.saas.broadcom.com/oi/v2/api/alarms/annotations
```

Sample Payload

```
[
{
"annotation":"test",
"alarmType":"SituationCluster",
"alarm_unique_id":"1751330",
"createdBy":" tadmin1"
}
]
```

Sample Payload if the Ticket ID is Blank

```
[
{"annotation":"test3",
```

```
"alarmType":"SituationCluster",
"alarm_unique_id":"2117188",
"ticketID":"",
"createdBy":"tadmin1 tadmin1"}
]
```

Sample Response

```
{
"IngestionSuccessfulAnnotations": [
{
"annotation": "test",
"alarmType": "SituationCluster",
"startTime": "1625231531000",
"timestamp": "1625231531000",
"createdBy": " tadmin1",
"updatedBy": null,
"alarm_unique_id": "1751330",
"annotation_id": "ff71552c-6d0a-49c2-9c49-a09fa87d0d2c"
}
]
} "ignore": false
}
]
}
```

Close API

You can update the status of a situation alarm as closed by calling the Close API in your automation scripts. When you send a request with relevant payload details using the Close API, DX Operational Intelligence closes the situation alarm.

Resource URI

You can select the URI based on the deployment and region of your tenant:

- DX SaaS - USA:
  `https://doi.dxi-na1.saas.broadcom.com/oi/v2/api/alarmactions/clear`
- DX SaaS - EU:
  `https://doi.dxi-eu1.saas.broadcom.com/oi/v2/api/alarmactions/clear`

```
https://<oi_host>:<oi_port>/oi/v2/api/alarmactions/clear
```

Method

```
POST
```

HTTP Headers

- Authorization: Bearer {token}
  For more information on tokens, see the Token Management section.
  For more information on tokens, see the Token Management section.
- Content Type: Application/JSON

Request Payload Syntax

```
{
```

```
"alarmDetails":[
{
"alarmId":<"Alarm ID”>,
"alarmType":"situation”
}
],
"actionDetails":{
"actionType":"clear"
}
}
```

Table 20: Parameters

Parameter	Description
alarmId	Sends the Situation alarm ID that you want to close.
alarmType	Sends the alarm type as **situation**. DX Operational Intelligence verifies the alarm type as **Situation** before closing the ticket.
actionType	Sends the action type as **clear**. DX Operational Intelligence updates the status of the situation alarm ID as closed, when the alarm type is **situation**.

Response Syntax

```
{
"actionStatus": [
{
"alarmId": <"Alarm ID">,
"status": "Success",
"updatedFields": {
"status": "closed",
"clearedBy": <"User ID">,
"lastUpdateTime": <"Time Stamp">
},
"ignore": false
}
]
}
```

Sample Request-Response

Sample URI

```
https://doi.dxi-na1.saas.broadcom.com/oi/v2/api/alarmactions/clear
```

Sample Payload

```
{"alarmDetails":[
{"alarmId":"1751606",
"alarmType":"situation"}],
"actionDetails":{"actionType":"clear"
}
}
```

Sample Response

```
{
"actionStatus": [
{
"alarmId": "1751606",
"status": "Success",
"updatedFields": {
"status": "closed",
"clearedBy": "TADMIN1",
"lastUpdateTime": "1625231914234"
},
"ignore": false
}
]
}
```

Create Ticket API

You can create a ticket in ServiceNow for a situation alarm by calling the Create API from the automated script. When you send a request with relevant payload details using the Create Ticket API, DX Operational Intelligence creates the ticket in ServiceNow for the specified situation alarm.

Resource URI

You can select the URI based on the deployment and region of your tenant:

- DX SaaS - USA:
  `https://doi.dxi-na1.saas.broadcom.com/oi/v2/api/alarmactions/ticket`
- DX SaaS - EU:
  `https://doi.dxi-eu1.saas.broadcom.com/oi/v2/api/alarmactions/ticket`

```
https://<oi_host>:<oi_port>/oi/v2/api/alarmactions/ticket
```

Method

```
POST
```

HTTP Headers

- Authorization: Bearer {token}
  For more information on tokens, see the Token Management section.
  For more information on tokens, see the Token Management section.
- Content Type: Application/JSON

Request Payload Syntax

```
{
"alarmDetails":[
{
"alarmId":<"Alarm ID”>,
"alarmType":"situation”
}
],
"actionDetails":{
```

```
"actionType":"ticket"
}
}
```

Table 21: Parameters

Parameter	Description
alarmId	Sends the Situation alarm ID for which you want to create a ticket.
alarmType	Sends the alarm type as **Situation**. DX Operational Intelligenceverifies the alarm type of the alarm as **Situation** before creating the ticket.
actionType	Sends the action type as **ticket**. DX Operational Intelligence creates the ticket for the alarm, when alarm type is **Situation**.

Response Syntax

```
"actionStatus": [
{
"alarmId": <"Alarm ID">,
"status": "Success",
"ticketID": <"Ticket ID>",
"troubleTicketUrl": <"URL of the ticket">,
"ticketLoggedBy": <"Valid Email ID of the user">,
"lastUpdateTime": "Time Stamp"
},
"ignore": false
}
]
}
```

Sample Request-Response

Sample URI

```
https://doi.dxi-na1.saas.broadcom.com/oi/v2/api/alarmactions/ticket
```

Sample Payload

```
{
"alarmDetails":[
{"alarmId":"1751606",
"alarmType":"situation"}],
"actionDetails":{"actionType":"ticket"
}
}
```

Sample Response

```
{
"actionStatus": [
{
"alarmId": "17262011",
"status": "Success",
"updatedFields": {
"ticketID": "INC64711470",
```

```
"troubleTicketUrl": "https://XYZ.service-now.com:443/nav_to.do?uri=incident.do?
sys_id=9a5228eedbc1b458dc1dd37a4896190a",
"ticketLoggedBy": "PERFTESTUSER2@DXI.COM",
"lastUpdateTime": "1625232575157"
},
"ignore": false
}
]
}
```

Cluster Alarms API

You can search and retrieve the list of situation alarms based on a search criteria by calling the Cluster Alarms API in your automation scripts. You can define the search criteria using the severity and alarm status filters. When you send a request with relevant payload details using the Cluster Alarms API, DX Operational Intelligence fetches the situation alarms for the given alarm cluster based on the defined search criteria.

> **NOTE**
> The Cluster Alarm API can retrieve maximum of 1000 alarms (newest) in response.

Resource URI

You can select the URI based on the deployment and region of your tenant:

- DX SaaS - USA:
  `https://doi.dxi-na1.saas.broadcom.com/oi/v2/api/situationAction/clusterAlarms/_search`
- DX SaaS - EU:
  `https://doi.dxi-eu1.saas.broadcom.com/oi/v2/api/situationAction/clusterAlarms/_search`

```
https://<oi_host>:<oi_port>/oi/v2/api/situationAction/clusterAlarms/_search
```

Method

```
POST
```

HTTP Headers

- Authorization: Bearer {token}
  For more information on tokens, see the Token Management section.
  For more information on tokens, see the Token Management section.
- Content Type: Application/JSON

Request Payload Syntax

```
{
"customFilter":{"and":{"expressions":[{"or":{"expressions":
[{"fieldDescription":"Severity","field":"severity","value":<"Severity">,"condition":"equals"}]}},{"or":
{"expressions":[{"fieldDescription":"Status","field":"status","value":<”Status">,"condition":"equals"}]}}]}},
"clusterAlarmId":<"cluster alarm ID">
}
```

Table 22: Parameters

Parameter	Description
Custom Filter	Defines the filter criteria to retrieve the alarms based on severity and status of the alarms.
clusterAlarmId	Sends the alarm cluster ID for which you want the DX Operational Intelligence to search and retrieve the associated alarms based on the defined filter criteria.

Response Syntax

```
{
"alarmCount": <Alarm count>,
"alarms": [
{
"metricName": <Metric Name>,
"csId": <csID>
"timeOrigin": <"Time STamp">,
"metricValue": "0",
"deviceId": <"Device ID">,
"deviceName": <"Device Name">,
"ciId": <"csID",
"alarmDescription": <"Alarm Description">,
"productVersion": <"Product Version">,
"problem": true,
"docTypeId": <"Doc Type ID">,
"ciUniqueId": <ciUniqueID">,
"host": <"Host">,
"alarmId": <"Alarm ID">,
"actionsSupported": {
"acknowledge": true,
"unAcknowledge": true,
"visible": false,
"ticket": true,
"unAssignment": true,
"assignment": true,
"clear": false,
"invisible": false
},
"isRootCause": false,
"severity": "critical",
"productId": "ao",
"sourceProduct": "Application Performance Management",
"vertexId": <"Vertex ID">,
"alarmURL": <"Alarm URL">,
"message": <"Message">,
"external_ids": [
<"External IDs">
],
"apm_alarm_unique_id": <"Alarm Unique ID">,
"exists_metadata": false,
"alarmType": "Application",
"tenantId": <"Tenant ID",
```

```
"docTypeVersion": "1",
"maintenance": "false",
"entity": <"Entity">,
"status": "NEW",
"lastUpdateTime": <"Time Stamp">
}
]
}
```


Sample Request-Response

Sample URI

```
https://doi.dxi-na1.saas.broadcom.com/oi/v2/api/situationAction/clusterAlarms/_search
```

Sample Payload


```
{
"customFilter":{"and":{"expressions":[{"or":{"expressions":
[{"fieldDescription":"Severity","field":"severity","value":"Critical","condition":"equals"}]}},
{"or":{"expressions":
[{"fieldDescription":"Status","field":"status","value":"New","condition":"equals"}]}}]}},"clusterAlarmId":"113540"}
```


Sample Response


```
{
"alarmCount": 498,
"alarms": [
{
"metricName": "SuperDomain|OI-Hosted-KafkaMonitor|OI-Hosted--Process|OI-Hosted-KafkaMonitorAgent|
Kafka|kafka2.OI-host.svc.cluster_9092|ConsumerGroups|Incident_bd751956-be76-4a76-9953-500454e93643|
Topics|tenantConfigTopic|5:Current Offset",
"csId": "SuperDomain|OI-Hosted-KafkaMonitor|OI-Hosted-Process|OIHosted-KafkaMonitorAgent|Kafka|
kafka2.OI-Host.svc.cluster.9092|ConsumerGroups|Incident_bd751956-be76-4a76-9953-500454e93643|
Topics|tenantConfigTopic|5",
"timeOrigin": "1623213030000",
"metricValue": "0",
"deviceId": "APM$$host$$OI-Hosted-KafkaMonitor",
"deviceName": "OI-Hosted-KafkaMonitor",
"ciId": "APM$$host$$OI-Hosted-KafkaMonitor$$agent$$SuperDomain|OI-Hosted-KafkaMonitor|OI-Hosted-
Process|OIHosted-KafkaMonitorAgent$$Kafka$$kafka2.OI-host.svc.cluster.local_9092$$ConsumerGroups
$$Incident_bd751956-be76-4a76-9953-500454e93643$$Topics$$tenantConfigTopic$$5$$Current Offset$
$null",
"alarmDescription": "Triggered when Kafka Server connection is up and Offset becomes zero, that
 means the consumer group stopped reading messages.",
"productVersion": "21.4.0.32(Build 990032)",
"problem": true,
"docTypeId": "itoa_alarms_apm",
"ciUniqueId": "SuperDomain|OI-Hosted-KafkaMonitor|OI-Hosted-Process|OI-Hosted-KafkaMonitorAgent|
Kafka|kafka2.OI-host.svc.cluster.9092|ConsumerGroups|Incident_bd751956-be76-4a76-9953-500454e93643|
Topics|tenantConfigTopic|5",
"host": "OI-Hosted-KafkaMonitor",
"alarmId": "587661.app.10.00.00.00.nip.io-f5b31f2d-13de-4267-929a-bd569e7c111b-OI-Hosted-
KafkaMonitor",
"actionsSupported": {
```

```
"acknowledge": true,
"unAcknowledge": true,
"visible": false,
"ticket": true,
"unAssignment": true,
"assignment": true,
"clear": false,
"invisible": false
},
"isRootCause": false,
"severity": "critical",
"productId": "ao",
"sourceProduct": "Application Performance Management",
"vertexId": "587661.app.10.00.00.00.nip.io_3272269",
"alarmURL": "https:\/\/apmservices-gateway-ao-apm.apps.aiops-OIHosted01-lb.lvn.domain.net\/link\/
apm_alarm_link?t_id=257&alert=SuperDomain%3ASaaS%3AKafka+Consumer+Offset",
"message": "The alert Kafka Consumer Offset has breached the CRITICAL threshold of 0",
"external_ids": [
"APM_INFRASTRUCTURE:AGENT:OI-Hosted-KafkaMonitor|OI-Hosted-Process|OI-Hosted-KafkaMonitorAgent"
],
"apm_alarm_unique_id": "587661.app.10.00.00.00.nip.io|SuperDomain|OI-Hosted-KafkaMonitor|OI-
Hosted-Process|OI-Hosted-KafkaMonitorAgent|Kafka|kafka2.OI-host.svc.cluster.9092|ConsumerGroups|
Incident_bd751956-be76-4a76-9953-500454e93643|Topics|tenantConfigTopic|5:Current Offset|
SuperDomain|Kafka Consumer Offset",
"exists_metadata": false,
"alarmType": "Application",
"tenantId": "0AF98315-6518-48AB-ACB8-337BEAE78B0D",
"docTypeVersion": "1",
"maintenance": "false",
"entity": "SuperDomain|OI-Hosted-KafkaMonitor|OI-Hosted-Process|OI-Hosted-KafkaMonitorAgent|Kafka|
kafka2.OI-host.svc.cluster.9092|ConsumerGroups|Incident_bd751956-be76-4a76-9953-500454e93643|
Topics|tenantConfigTopic|5",
"status": "NEW",
"lastUpdateTime": "1623213030000"
}
]
}
```


Cluster Host API

You can search and fetch the list of entities (hosts) of an alarm cluster that are not closed, by calling the Cluster Host API in your automation scripts. You can define the search criteria using the alarm status filters. When you send a request with relevant payload details using the Cluster Host API, DX Operational Intelligence fetches the entities that are not closed for a given alarm cluster.

Resource URI

You can select the URI based on the deployment and region of your tenant:

- DX SaaS - USA:
  `https://doi.dxi-na1.saas.broadcom.com/oi/v2/api/situationAction/clusterHosts/_search`
- DX SaaS - EU:
  `https://doi.dxi-eu1.saas.broadcom.com/oi/v2/api/situationAction/clusterHosts/_search`

```
https://<oi_host>:<oi_port>/oi/v2/api/situationAction/clusterHosts/_search
```

Method

```
POST
```

HTTP Headers

- Authorization: Bearer {token}
  For more information on tokens, see the Token Management section.
  For more information on tokens, see the Token Management section.
- Content Type: Application/JSON

Request Payload Syntax

```
{"customFilter":{"and":{"expressions":[{"or":{"expressions":
[{"fieldDescription":"isClosed","field":"isClosed","value":"false","condition":"equals"}]}}]}},"clusterAlarmId":<"Clust
 Alarm ID">}
```

Table 23: Parameters

Parameter	Description
Custom Filter	Defines the filter criteria to fetch the entities associated with the specified cluster alarm that are in the **open** state.
clusterAlarmId	Sends the cluster alarm ID for which you want to search and retrieve the associated alarms that are not closed.

Response Syntax

```
{
"totalRecords": <count of alarms that are in open state>,
"hosts": [ <"Host">
]
}
```

Sample Request-Response

Sample URI

```
https://doi.dxi-na1.saas.broadcom.com/oi/v2/api/situationAction/clusterHosts/_search
```

Sample Payload

```
{"customFilter":{"and":{"expressions":[{"or":{"expressions":
[{"fieldDescription":"isClosed","field":"isClosed","value":"false","condition":"equals"}]}}]}},"clusterAlarmId":"113
```

Sample Response

```
{
"totalRecords": 498,
"hosts": [ "SuperDomain|OI-Hosted-KafkaMonitor|OI-Hosted-Process|OI-Host-KafkaMonitorAgent|Kafka|
kafka3.oi-host.svc.cluster.9092|ConsumerGroups|Incident_c04b23e5-0b33-4fa0-a791-82ba39af9cde|
Topics|tenantConfigTopic|7"
]
}
```

Situation Clustering Dimensions APIs

The Situation Clustering APIs allow the tenant administrator to configure the situations clustering, and root cause analysis functionality. These configurations take precedence over the default configuration values when the configurations are specified by the user. The following parameters are supported for situation clustering APIs.

Key	Default Value	Min/Max Value	Data Type	Description
historicalweight	0.5	0.0/1.0		Define the Weightage of historical data in the template similarity calculation.
serviceweight	0.5	0.0/1.0		Define the Weightage of services in the template similarity calculation.
ciSignificance	1.0	0.0/1.0		Define the Weightage of CIs in clustering calculation for APM alarms.
temporalSignificance	1.0	0.0/1.0		Define the Weightage of temporal in the clustering calculation.
textSignificance	1.0	0.0/1.0		Define the Weightage of text in clustering calculation.
enableServiceClustering	TRUE			Signifies whether service similarity is considered for clustering or not.
is_consider_absolute_service	FALSE			If true, service similarity is considered without service the weightage.
percentile	0.95	0.0/1.0		Removes RCA candidates with low score edges.
damping_factor	1	1/-		
rootcause_count	3	1/-		Maximum number of root causes.
skip_hist_priors	FALSE			Signifies whether historical data is considered for RCA or not.
score_threshold	0.5	0.0/1.0		Score Limit to consider a template as an RCA candidate.
noise_threshold	0.8	0.0/1.0		Entropy value to identify the noise.

Key	Default Value	Min/Max Value	Data Type	Description
skip_alarm_types	anamoly,informative,predictive		list	The specified type does not participate in clustering.
enable_historical_flow	TRUE			Enables historical flow for similarity calculation.
excludeNonServiceImpactedTemplates	FALSE			Excludes the non-services impacted templates during clustering.
alarm_updates_consider_period	1000	1/1825		Specifies the age of the alarm (in terms of the number of days) to be considered for situation clustering. If the alarm's age is greater than configured age, they are not considered for situation clustering.
enable_single_alarm_association	False			Enables Single Alarm association with a situation when the value is set to 'True'. DX Operational Intelligence creates one situation when a new alarm is raised. After the Situation is stable, the application does not create situations when there are updates to the alarm.
additional_cluster_group_by			list	Enables you to add custom clustering criteria to determine the alarm similarity for clustering. For more information, see the **Custom Clustering Criteria** section on this page.

Configure Attributes API

This API allows you to configure the attributes for situation clustering.

API Parameters

- You can select the URI based on the deployment and region of your tenant:
  - DX SaaS - USA:
    `https://doi.dxi-na1.saas.broadcom.com/oi/v2/api/clusteralarms/tenantConfigAttributes`
  - DX SaaS - EU:

```
https://doi.dxi-eu1.saas.broadcom.com/oi/v2/api/clusteralarms/tenantConfigAttributes
```

- **Endpoint:** https://<oi_host>:<oi_port>/oi/v2/api/clusteralarms/tenantConfigAttributes
- **Method:** POST
- **HTTP Headers:**
  - **Content-Type:** application/JSON
  - **Authorization:** bearer token

    NOTE
    For more information on tokens, see the Token Management section.
- **Syntax:**

```
{
    "tenantId": "DF1D62B7-DB3D-4100-8738-9409A02845A0",
    "configAttrList": [
       {
            "key": "serviceWeight",
            "value": 0.7,
            "type": "double"
        },
         {
            "key": "is_consider_absolute_service",
            "value": false,
            "type": "boolean"
        },
        {
            "key": "temporalSignificance",
            "value": 0.0,
            "type": "double"
        }
         {
            "key": "additional_cluster_group_by",
            "value": [
                {
                    "type": "MATCH_BY_PERCENTAGE",
                    "fields": [
                        "product","severity"
                    ]
             }
            ],
            "type": "list"
    ]
}
```

- **Response:** 200 OK

```
{
   "status": "Success"
 }
```

Usage Example

Resource URL: https://doi.dxi-na1.saas.broadcom.com/oi/v2/api/clusteralarms/tenantConfigAttributes

Payload:

```
{
    "tenantId": "DF1D62B7-DB3D-4100-8738-9409A02845A0",
```

```
"configAttrList": [
    {
        "key": "historicalWeight",
        "value": 0.0,
        "type": "double"
    },
    {
        "key": "serviceWeight",
        "value": 0.7,
        "type": "double"
    },
    {
        "key": "ciSignificance",
        "value": 1.0,
        "type": "double"
    },
    {
        "key": "temporalSignificance",
        "value": 0.0,
        "type": "double"
    },
    {
        "key": "textSignificance",
        "value": 1.0,
        "type": "double"
    },
    {
        "key": "enabledServiceClustering",
        "value": true,
        "type": "boolean"
    },
    {
        "key": "is_consider_absolute_service",
        "value": false,
        "type": "boolean"
    },
    {
        "key": "percentile",
        "value": 0.95,
        "type": "double"
    },
    {
        "key": "damping_factor",
        "value": 0.85,
        "type": "double"
    },
    {
        "key": "rootcause_count",
        "value": 3,
        "type": "double"
    },
    {
        "key": "skip_hist_priors",
```

```
        "value": false,
        "type": "boolean"
    },
    {
        "key": "score_threshold",
        "value": 0.5,
        "type": "double"
    },
    {
        "key": "noise_threshold",
        "value": 0.8,
        "type": "double"
    },
    {
        "key": "skip_alarm_types",
        "value": [
            "anamoly",
            "informative",
            "predictive"
        ],
        "type": "list"
    },
    {
        "key": "enable_historical_flow",
        "value": true,
        "type": "boolean"
    },
   {
        "key": "excludeNonServiceImpactedTemplates",
        "value": false,
        "type": "boolean"
    },
    {
        "key": "alarm_updates_consider_period",
        "value": 1000,
        "type": "integer"
    },
    {
        "key": "enable_single_alarm_association",
        "value": true,
        "type": "boolean"
    }
     {
        "key": "additional_cluster_group_by",
        "value": [
            {
                "type": "MATCH_BY_PERCENTAGE",
                "fields": [
                    "product","severity"
                ]
            }
        ],
        "type": "list"
```

```
    ]
}
```

Response:

```
{
  "status": "Success"
}
```

Update Existing Values

This API allows you to update the existing value of situation clustering.

API Parameters

- You can select the URI based on the deployment and region of your tenant:
  - DX SaaS - USA:
    `https://doi.dxi-na1.saas.broadcom.com/oi/v2/api/clusteralarms/tenantConfigAttributes`
  - DX SaaS - EU:
    `https://doi.dxi-eu1.saas.broadcom.com/oi/v2/api/clusteralarms/tenantConfigAttributes`
- **Endpoint:** https://<oi_host>:<oi_port>/oi/v2/api/clusteralarms/tenantConfigAttributes
- **Method:** PUT
- **HTTP Headers:**
  - **Content-Type:** application/JSON
  - **Authorization:** bearer token

    NOTE
    For more information on tokens, see Token Management section.
- **Body:**

```
{
    "tenantId": "<tenantid>",
    "configAttrList": [
        {
            "key": "<keyvalue>",
            "value": <defaultvalue/user-specified value>,
            "type": "<type of the key>"
        },
        .
        .
        .
        {
            "key": "temporalSignificance",
            "value": 0.0,
            "type": "double"
        },
    ]
}
```

- **Response:** 200 OK

```
{"Status":"Success"}
```

Usage Example

Resource URL: https://doi.dxi-na1.saas.broadcom.com/oi/v2/api/clusteralarms/tenantConfigAttributes

Body:

```
{
    "tenantId": "DF1D62B7-DB3D-4100-8738-9409A02845A0",
    "configAttrList": [
       {
            "key": "serviceWeight",
            "value": 0.7,
            "type": "double"
        },
         {
            "key": "is_consider_absolute_service",
            "value": false,
            "type": "boolean"
        },
        {
            "key": "temporalSignificance",
            "value": 0.0,
            "type": "double"
        }
         {
            "key": "additional_cluster_group_by",
            "value": [
                {
                    "type": "MATCH_BY_PERCENTAGE",
                    "fields": [
                        "product","severity"
                    ]
                }
            ],
            "type": "list"
    ]
}
```

Response:

```
{"Status":"Success"}
```

Error Response:

```
{
"Error":"Error in attribute with key: enable_historical_flowz,
type: boolean, value: true",
"Reason":"Invalid key - enable_historical_flowz"
}
```

Custom Clustering Criteria

This API enables you to add custom clustering criteria.

NOTE

- You can cluster the alarms by entity, message, and service out-of-the-box. You cannot delete the default criteria.
- You can add only a maximum of five additional clustering criteria.
- A custom clustering criteria can be deleted only if it is not associated with any situation definition.

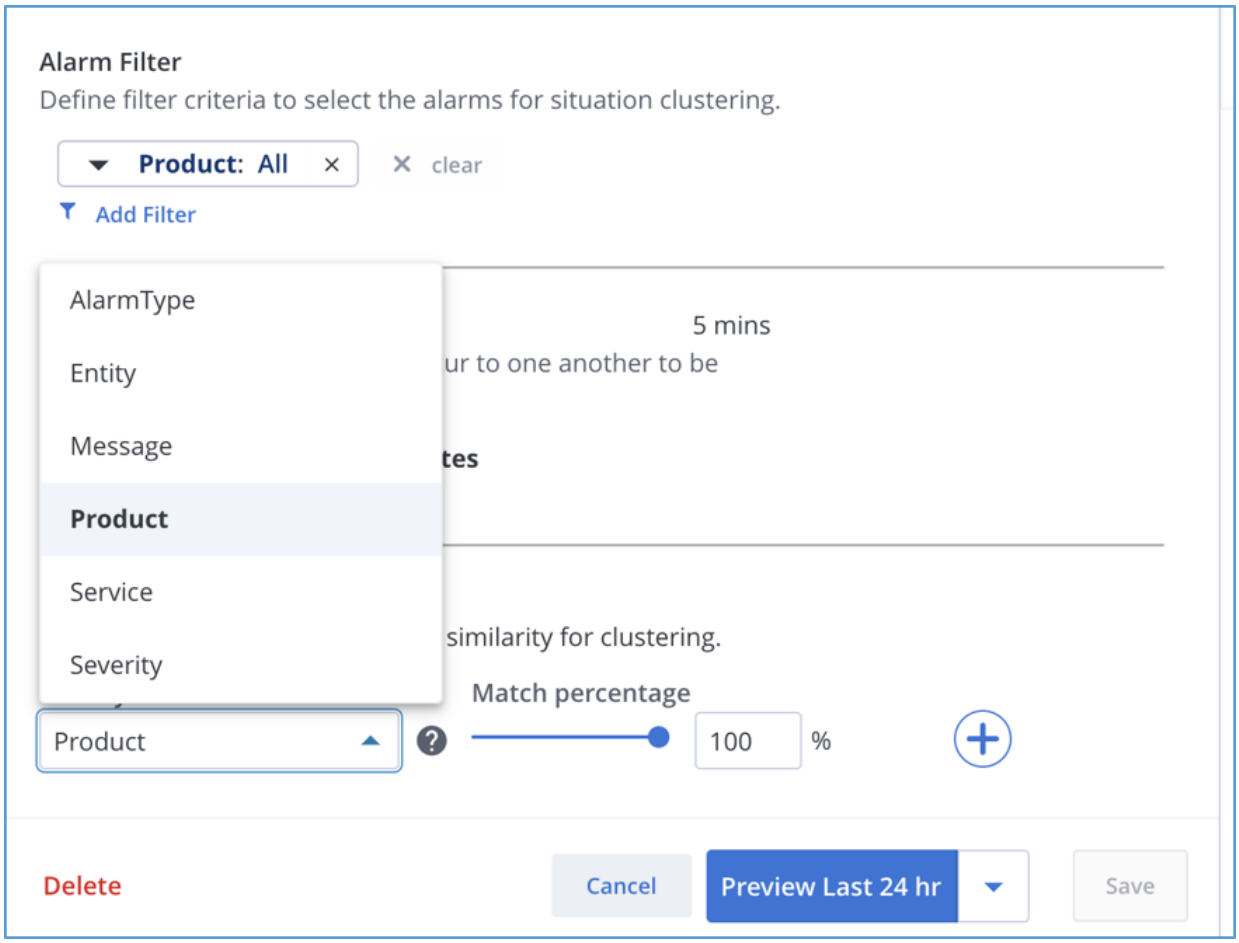
API Parameters

- You can select the URI based on the deployment and region of your tenant:
  - DX SaaS - USA:
    `https://doi.dxi-na1.saas.broadcom.com/oi/v3/api/clusteralarms/tenantConfigAttributes`
  - DX SaaS - EU:
    `https://doi.dxi-eu1.saas.broadcom.com/oi/v3/api/clusteralarms/tenantConfigAttributes`
- **Method:** PUT
- **HTTP Headers:**
  - **Content-Type:** application/JSON
  - **Authorization:** bearer token

    NOTE
    For more information on tokens, see Token Management section.
- **Sample Body:**
  The following sample body illustrates the payload for the custom criteria to be added:

```
{
"tenantId": "BB40FB73-3AEC-4B75-B96C-91DB89SW2",
"configAttrList": [
{
"key": "additional_cluster_group_by",
"value": [
{
"type": "MATCH_BY_PERCENTAGE",
"fields": [
            "product",
            "severity",
            "alarmType",
            "severity"
          ]
}
          ],
"type": "list"
}
                  ]
}
```

  The following image illustrates the custom clustering criteria that was added using this API:



- **Sample Body for Custom Clustering Criteria Deletion:**
  The following sample body illustrates the payload to delete the added custom criteria:

```
{
"tenantId": "BB40FB73-3AEC-4B75-B96C-91DB89SW2",
"configAttrList": [
{
"key": "additional_cluster_group_by",
"value": [
{
"type": "MATCH_BY_PERCENTAGE",
"fields": [
            "product",
            "severity",
                    ]
}
          ],
"type": "list"
}
                  ]
}
```

- **Response:** 200 OK

```
{"Status":"Success"}
```

Topology Processor APIs

The topology processor APIs provide the following details about the topology compaction when the compaction rules are modified:

- The number of impacted vertexes to a compaction rule.
- Current topology compaction rules
- Validation on changes without impacting TAS

- Get Correlation Information
- Get Tenant Correlation Rules
- Dry-Run Correlation Rules API
- Update Correlation Rules API

Get Correlation Information

This API retrieves the current correlation information.

Resource URI

You can select the URI based on the deployment and region of your tenant:

- DX SaaS - USA:
  `https://apmgw.dxi-na1.saas.broadcom.com/oi/v2/tp/correlation`
- DX SaaS - EU:
  `https://apmgw.dxi-eu1.saas.broadcom.com/oi/v2/tp/correlation`

Method

```
GET
```

HTTP Headers

- Authorization: user token that can be generated from the Token Management page.
- Content Type: Application/JSON

Request Syntax

```
https://apmgw.dxi-na1.saas.broadcom.com/oi/v2/tp/correlation
```

Response Syntax

```
{
    "errorMessage": null,
    "numberOfCorrelatedVertices": <count of correlated vertices>,
    "numberOfAcnVertices": <count of acn vertices>,
    "correlatedVertices": [
        {
            "acnExternalId": "<acn id>",
            "correlatedVertexExternalIds": [
                "<correlatedVertexExternalIds1>",
                "<correlatedVertexExternalIds2>"
            ],
            "correlatedByAttrs": [
                "<ip_address>",
```

```
                "<mac_address>"
            ]
        },
        {
           "acnExternalId": "<acn id>",
            "correlatedVertexExternalIds": [
                "<correlatedVertexExternalIds1>",
                "<correlatedVertexExternalIds2>"
            ],
            "correlatedByAttrs": [
                "<ip_address>",

            ]
        },
        {
             "acnExternalId": "<acn id>",
            "correlatedVertexExternalIds": [
                "<correlatedVertexExternalIds1>",
                "<correlatedVertexExternalIds2>"
            ],
            "correlatedByAttrs": [
                "<ip_address>",
                "<mac_address>"
            ]
        }
    ]
}
```

Example

Request

```
https://apmgw.dxi-na1.saas.broadcom.com/oi/v2/tp/correlation
```

Response

```
200 OK
{
    "errorMessage": null,
    "numberOfCorrelatedVertices": 6,
    "numberOfAcnVertices": 3,
    "correlatedVertices": [
        {
            "acnExternalId": "ACN:V-Spectrum-3ff16f45-c8ef-472e-32773a4625b9",
            "correlatedVertexExternalIds": [
                "NETWORK_SPECTRUM:acn-spectrum-qa04_0x101265",
                "NETWORK_VNA:872128e6-b21f-0dd3181cce62_14219"
            ],
            "correlatedByAttrs": [
                "dx_ip_address",
                "dx_mac_address"
            ]
        },
        {
```

```
            "acnExternalId": "ACN:V-Spectrum-502b36a5-9683-478d-98e7a4c9cf90",
            "correlatedVertexExternalIds": [
                "NETWORK_SPECTRUM:acn-spectrum-qa04_0x1011ed",
                "NETWORK_VNA:872128e6-b21f-0dd3181cce62_19547"
            ],
            "correlatedByAttrs": [
                "dx_ip_address"
            ]
        },
        {
            "acnExternalId": "ACN:V-Spectrum-c275da91-92eb-495b-c5ad5e8f57cc",
            "correlatedVertexExternalIds": [
                "NETWORK_SPECTRUM:acn-spectrum-qa04_0x1011e5",
                "NETWORK_VNA:872128e6-b21f-0dd3181cce62_6450"
            ],
            "correlatedByAttrs": [
                "dx_ip_address",
                "dx_mac_address"
            ]
        }
    ]
}
```


Get Tenant Correlation Rules

This API retrieves the tenant-level correlation rules (normalized attributes mapping rules and correlation attributes).

Resource URI

You can select the URI based on the deployment and region of your tenant:

- DX SaaS - USA:
  `https://apmgw.dxi-na1.saas.broadcom.com/oi/v2/tp/correlation/rules`
- DX SaaS - EU:
  `https://apmgw.dxi-eu1.saas.broadcom.com/oi/v2/tp/correlation/rules`

Parameters

`effective=true/false`

Method

`GET`

HTTP Headers

- Authorization: user token that can be generated from the Token Management page.
- Content Type: Application/JSON

Request Syntax

`https://apmgw.dxi-na1.saas.broadcom.com/oi/v2/tp/correlation/rules`

Response Syntax

```
{
   "    ""errorMessage":" null",
   "    ""customAttrMappingRules":" "{
      "        ""XYZ":" "[
         "            "{
            "                ""defaultRule":" "{
               "                    ""toAttrNames":" "[
                  "                        ""<ip_address>""                    "
               ],
               "                    ""toValueType":" ""<valuttype>",
               "                    ""fromAttrNames":" "[
                  "                        ""<primaryIp>""                    "
               ],
               "                    ""excludedValues":" "[
                  "                        ""<ipaddress>",
                  "                        ""::1""                    "
               ],
               "                    ""useFirstValue":" <true/false>",
               "                    ""fromDataValidatorType":" ""<IP>",
               "                    ""joinFromValues":" <true/false>",
               "                    ""joinDelimiter":" """,
               "                    ""valid":" <true/false>

                "
            },
            "                ""override":" <true/false>",
            "                ""valid":" <true/false>",
            "                ""toAttrNames":" "[
               "                    ""<ip_address>""                "
            ]"            "
         },
         "            "{
            "                ""defaultRule":" "{
               "                    ""toAttrNames":" "[
                  "                        ""<hostname>""                    "
               ],
               "                    ""toValueType":" ""<valuttype>",
               "                    ""fromAttrNames":" "[
                  "                        ""<dnsName>",
                  "                        ""<host>""                    "
               ],
               "                    ""excludedValues":" "[
                  "                        """"                    "
               ],
               "                    ""useFirstValue":" <true/false>",
               "                    ""toValueFormatterType":" ""<LOWER_CASE>",
               "                    ""joinFromValues":" <true/false>",
               "                    ""joinDelimiter":" """,
               "                    ""valid":" <true/false>

                "
            },
```

```
            "                ""conditionRules":" "[
               "                    "{
                  "                        ""orderPosition":" <positioncount>",
                  "                        ""conditionExp":" "{
                     "                           ""matchAttrName":" ""<attributetype>",
                     "                           ""matchAttrValue":" ""<attributevalue>",
                     "                           ""ignoreCase":" <true/false>",
                     "                           ""regexMatch":" <true/false>

                        "
                  },
                  "                        ""rule":" "{
                     "                           ""toAttrNames":" "[
                        "                              ""<hostname>""                            "
                     ],
                     "                           ""toValueType":" ""<valuetype>",
                     "                           ""fromAttrNames":" "[
                        "                              ""<dnsName>",
                        "                              ""<vmName>""                            "
                     ],
                     "                           ""excludedValues":" "[
                        "                              """                            "
                     ],
                     "                           ""useFirstValue":" <true/false>",
                     "                           ""toValueFormatterType":" ""<LOWER_CASE>",
                     "                           ""joinFromValues":" <true/false>",
                     "                           ""joinDelimiter":" """,
                     "                           ""valid":" <true/false>

                        "
                  },
                  "                        ""valid":" <true/false>

                     "
               }"                "
            ],
            "                ""override":" <true/false>",
            "                ""valid":" <true/false>",
            "                ""toAttrNames":" "[
               "                    ""<hostname>""                "
            ]"            "
         }"        "
      ]"    "
   },
   "    ""effectiveAttrMappingRules":" null",
   "    ""correlationAttributes":" "[
      "        ""<ip_address>",
      "        ""<mac_address>""    "
   ]
}
```

Example Request

```
https://apmgw.dxi-na1.saas.broadcom.com/oi/v2/tp/correlation/rules
```

Example Response

```
200 OK
{
   "    ""errorMessage":" null",
   "    ""customAttrMappingRules":" "{
      "        ""XYZ":" "[
         "            "{
            "                ""defaultRule":" "{
               "                    ""toAttrNames":" "[
                  "                        ""dx_ip_address""                    "
               ],
               "                    ""toValueType":" ""SCALAR",
               "                    ""fromAttrNames":" "[
                  "                        ""primaryIp""                    "
               ],
               "                    ""excludedValues":" "[
                  "                        ""127.0.0.1",
                  "                        ""::1""                    "
               ],
               "                    ""useFirstValue":" true",
               "                    ""fromDataValidatorType":" ""IP",
               "                    ""joinFromValues":" false",
               "                    ""joinDelimiter":" """,
               "                    ""valid":" true

                "
            },
            "                ""override":" false",
            "                ""valid":" true",
            "                ""toAttrNames":" "[
               "                    ""dx_ip_address""                "
            ]"            "
         },
         "            "{
            "                ""defaultRule":" "{
               "                    ""toAttrNames":" "[
                  "                        ""dx_hostname""                    "
               ],
               "                    ""toValueType":" ""SCALAR",
               "                    ""fromAttrNames":" "[
                  "                        ""dnsName",
                  "                        ""host""                    "
               ],
               "                    ""excludedValues":" "[
                  "                        """"                    "
               ],
               "                    ""useFirstValue":" true",
               "                    ""toValueFormatterType":" ""LOWER_CASE",
               "                    ""joinFromValues":" false",
```

```
            "                    ""joinDelimiter":" """,
            "                    ""valid":" true

             "
         },
         "                ""conditionRules":" "[
            "                    "{
               "                       ""orderPosition": 1000,
               "                       ""conditionExp":" "{
                  "                          ""matchAttrName":" ""type",
                  "                          ""matchAttrValue":" ""vSwitch",
                  "                          ""ignoreCase":" false",
                  "                          ""regexMatch":" false

                     "
               },
               "                       ""rule":" "{
                  "                          ""toAttrNames":" "[
                     "                             ""dx_hostname""                            "
                  ],
                  "                          ""toValueType":" ""SCALAR",
                  "                          ""fromAttrNames":" "[
                     "                             ""dnsName",
                     "                             ""vmName""                            "
                  ],
                  "                          ""excludedValues":" "[
                     "                             """"                            "
                  ],
                  "                          ""useFirstValue":" true",
                  "                          ""toValueFormatterType":" ""LOWER_CASE",
                  "                          ""joinFromValues":" false",
                  "                          ""joinDelimiter":" """,
                  "                          ""valid":" true

                     "
               },
               "                       ""valid":" true

                 "
            }"                "
         ],
         "                ""override":" false",
         "                ""valid":" true",
         "                ""toAttrNames":" "[
            "                    ""dx_hostname""                "
         ]"            "
      }"        "
   ]"    "
},
"    ""effectiveAttrMappingRules":" null",
"    ""correlationAttributes":" "[
   "        ""dx_ip_address",
   "        ""dx_mac_address""    "
```

```
    ]
}
```

Dry-Run Correlation Rules API

This API allows you to view the correlation results of current rules or given rules without applying the rules. This API helps the customer or field engineer to add, test, or debug the custom correlation rules.

Resource URL

You can select the URI based on the deployment and region of your tenant:

- DX SaaS - USA:
  `https://apmgw.dxi-na1.saas.broadcom.com/oi/v2/tp/correlation/rules/dryrun`
- DX SaaS - EU:
  `https://apmgw.dxi-eu1.saas.broadcom.com/oi/v2/tp/correlation/rules/dryrun`

Method

`POST`

HTTP Headers

- Authorization: user token that can be generated from the Token Management page.
- Content Type: Application/JSON

Request URL Syntax

`https://apmgw.dxi-na1.saas.broadcom.com/oi/v2/tp/correlation/rules/dryrun`

Request Payload Syntax

> **NOTE**
> If attributeMappingRules or correlationAttributes is not present in the body, current attributes mapping rules for this tenant is used for the dry-run.

```
{
      "correlationAttributes": ["<normalized attribute>"],
      "attributeMappingRules": [
         {
           "products": [ "xyz"],
           "rules": [
             {
               "toAttrName": ["<normalized attribute>"],
               "toValueType": "<valuetype>",
               "fromAttrNames": ["<IP>"],
               "fromDataValidator": "<IP>",
               "excludedValues": [ "<ipaddress>",  "::1"],
               "excludedAttrValueNames": null
             }
          ]
        }
      ]
    }
```

Response Syntax

```
200 OK
{
    "errorMessage": null,
    "numberOfCorrelatedVertices": <number of vertices>,
    "numberOfAcnVertices": <number of acn vertices>,
    "correlatedVertices": [
        {
            "acnExternalId": null,
            "correlatedVertexExternalIds": [
                "<externalid1>",
                "<externalid2>"
            ],
            "correlatedByAttrs": [
                "<ip_address>"
            ]
        }
    ]
}
```

Example

Request URL

```
https://apmgw.dxi-na1.saas.broadcom.com/oi/v2/tp/correlation/rules/dryrun
```

Request Payload

```
{
      "correlationAttributes": ["normalized attribute"],
      "attributeMappingRules": [
         {
           "products": [ "xyz"],
           "rules": [
             {
               "toAttrName": ["normalized attribute"],
               "toValueType": "scalar",
```

```
                "fromAttrNames": ["ip"],
                "fromDataValidator": "IP",
                "excludedValues": [ "127.0.0.1",  "::1"],
                "excludedAttrValueNames": null
              }
          ]
        }
      ]
    }
```

Response

```
200 OK
{

    "errorMessage": null,

    "numberOfCorrelatedVertices": 2,

    "numberOfAcnVertices": 1,

    "correlatedVertices": [

        {

            "acnExternalId": null,

            "correlatedVertexExternalIds": [

                "CUSTOM:productA-1000",

                "CUSTOM:productB-2000"

            ],

            "correlatedByAttrs": [

                "dx_ip_address"

            ]

        }

    ]

}
```

Update Correlation Rules API

This API allows you to update the correlation results of current rules or given rules without applying the rules.

Resource URI

You can select the URI based on the deployment and region of your tenant:

- DX SaaS - USA:
  `https://apmgw.dxi-na1.saas.broadcom.com/oi/v2/tp/correlation/rules/update`
- DX SaaS - EU:
  `https://apmgw.dxi-eu1.saas.broadcom.com/oi/v2/tp/correlation/rules/update`

Method

`POST`

HTTP Headers

- Authorization: user token that can be generated from the Token Management page.
- Content Type: Application/JSON

Request URL Syntax

`https://apmgw.dxi-na1.saas.broadcom.com/oi/v2/tp/correlation/rules/update`

Request Payload Syntax

`hhttps://apmgw.dxi-na1.saas.broadcom.com/oi/v2/tp/correlation/rules/update`

Response Syntax

`200 OK`

```
{
 "correlationAttributes": [
     "<correlationattribute>"
 ],
 "attributeMappingRules": [
     {
         "products": [
             "<productname>"
         ],
         "rules": [
             {
                 "toAttrName": [
                     "<attributename>"
                 ],
                 "toValueType": "<valuetype>",
                 "fromAttrNames": [
                     "<fromattributename>"
                 ],
                 "fromDataValidator": "<datavalidator value>",
                 "excludedValues": [
                     "<excludedvalue1>",
                     "<excludedvalue2>"
                 ],
                 "excludedAttrValueNames": null
             }
         ]
```

```
        }
    ],
    "associationRules": [
        {
            "sourceTasVertices": {
                "attributeFilters": [
                    {
                        "attributeName": "<attributename>",
                        "attributeValues": [
                            "<attributevalue>"
                        ],
                        "operator": "<value>"
                    },
                    {
                        "attributeName": "<attributename>",
                        "attributeValues": [
                            "<attributevalue>"
                        ],
                        "operator": "<value>"
                    }
                ],
                "layer": "<layername>",
                "matchingAttributeNames": [
                    "<matchingattributename>",
                    "<matchingattributename>"
                ]
            },
            "targetTasVertices": {
                "attributeFilters": [
                    {
                        "attributeName": "<attributename>",
                        "attributeValues": [
                            "<attributevalue>"
                        ],
                        "operator": "<value>"
                    },
                    {
                        "attributeName": "<attributename>",
                        "attributeValues": [
                            "<attributevalue>"
                        ],
                        "operator": "<value>"
                    }
                ],
                "layer": "<layername>",
                "matchingAttributeNames": [
                    "<matchingattributename>",
                    "<matchingattributename>"
                ]
            },
            "semantic": "contains"
        }
    ]
```

```
}
```

Example

Request URI

```
https://apmgw.dxi-na1.saas.broadcom.com/oi/v2/tp/correlation/rules/update
```

Response

```
200 OK
{
    "correlationAttributes": [
        "dx_ip_address"
    ],
    "attributeMappingRules": [
        {
            "products": [
                "Broadcom"
            ],
            "rules": [
                {
                    "toAttrName": [
                        "dx_ip_address"
                    ],
                    "toValueType": "scalar",
                    "fromAttrNames": [
                        "ip"
                    ],
                    "fromDataValidator": "IP",
                    "excludedValues": [
                        "127.0.0.1",
                        "::1"
                    ],
                    "excludedAttrValueNames": null
                }
            ]
        }
    ],
    "associationRules": [
        {
            "sourceTasVertices": {
                "attributeFilters": [
                    {
                        "attributeName": "product",
                        "attributeValues": [
                            "productA"
                        ],
                        "operator": "IN"
                    },
                    {
                        "attributeName": "type",
                        "attributeValues": [
```

```
                        "device"
                    ],
                    "operator": "IN"
                }
            ],
            "layer": "CUSTOM",
            "matchingAttributeNames": [
                "origin",
                "device_ip"
            ]
        },
        "targetTasVertices": {
            "attributeFilters": [
                {
                    "attributeName": "product",
                    "attributeValues": [
                        "productB"
                    ],
                    "operator": "IN"
                },
                {
                    "attributeName": "type",
                    "attributeValues": [
                        "interface"
                    ],
                    "operator": "IN"
                }
            ],
            "layer": "CUSTOM",
            "matchingAttributeNames": [
                "source",
                "parent_ip"
            ]
        },
        "semantic": "contains"
    }
  ]
}
```

Update Specific Fields API

The Update Specific Fields API enables you to add or update fields in the raw alarms or all alarms using the APM Gateway or NGINX endpoint. You can find the endpoint information on the **Connector Parameters** page.

Update the Fields Using APM Gateway Endpoint

In the APM Gateway Endpoint method, send the request using POSTMAN or other REST clients. In the request, update the tenant token, index name, header, and body.

Follow these steps:

1. Enter the URL for the APM Gateway endpoint. You can find this information on the Connector Parameters page.
2. Select the HTTP method as POST.

3. Update the tenant token.
4. Update the index_name to **itoa_update_alarms_all**.
5. Update the header and body. The following sample payload illustrates how to update the troubleTicketUrl field:

```
{
   "documents":[
      {
         "header":{
            "product_id":"ao",
            "tenant_id":"abcd123-def356-234gasf",
            "doc_type_id":"itoa_update_alarms_all",
            "doc_type_version":"1"
         },
         "body":[
            {
               "alarm_unique_id":"BCDA67-5976",
               "doc_type_id":"itoa_alarms_uim",
               "doc_type_version":"1",
               "product":"UIM",
               "username":"TestUser"
               "update_fields":[
                  {
                     "name":"severity",
                     "value":"critical"
                  },
                  {
                     "name":"message",
                     "value":"sample updated message"
                  }
               ]
            }
         ]
      }
   ]
}
```

Update the Fields Using NGINX Endpoint

In the NGINX Endpoint method, send the request using POSTMAN or other REST clients. In the request, update only the index name, header, and body.

Follow these steps:

1. Enter the URL for the NGINX endpoint. You can find this information on the Connector Parameters page.
2. Select the HTTP method as POST.
3. Update the tenant token.
4. Update the index_name to **itoa_update_alarms_all**.
5. Update the header and body. The following sample payload illustrates how to update the troubleTicketUrl field:

```
{
   "documents":[
      {
         "header":{
            "product_id":"ao",
            "tenant_id":"abcd123-def356-234gasf",
```

```
                "doc_type_id":"itoa_update_alarms_all",
                "doc_type_version":"1"
            },
            "body":[
                {
                    "alarm_unique_id":"BCDA67-5976",
                    "doc_type_id":"itoa_alarms_uim",
                    "doc_type_version":"1",
                    "product":"UIM",
                    "username":"TestUser"
                    "update_fields":[
                        {
                            "name":"severity",
                            "value":"critical"
                        },
                        {
                            "name":"message",
                            "value":"sample updated message"
                        }
                    ]
                }
            ]
        }
    ]
}
```


DX SaaS APIs

This section provides information about the different APIs that are a key part of DX Platform.

APIs are a key part of DX Platform, that enables you to easily manage the power of various services available as part of the AIOps solution.

Available APIs

The following table lists all currently available API Categories for DX Platform:

API	Description
DX Platform Catalog APIs	Use the Platform Catalog APIs to manage the DX Platform services including access control, service configurations, creating and managing tenancy units among others.
Global Maintenance APIs	Use the Global Maintenance APIs to perform maintenance activities on entities (devices, CIs, groups, or services) using scheduled maintenance mode.
RESTMon APIs	Use the RESTMon APIs to create, retrieve, update, or delete access to the RESTMon Profile and Schema resources.

DX Platform Catalog APIs

Describes the DX Platform Catalog APIs to manage DX Platform services including access control, service configurations, creating and managing tenancy units among others.

Use the Platform Catalog APIs to manage DX Platform services, including access control, service configurations, creating and managing tenancy units, among others.

Tenant Onboarding

API Operation	Description
Retrieve Tenant Cohort ID	Retrieves the Tenant Cohort ID

Identity Management

API Operation	Description
Authenticate User	Authenticates a user
Retrieve Auth Config	Retrieves Authentication Configuration

Channel Management

Category	API Operation	Description
Channels	Create Email Channel	Creates an email channel.
	Delete Email Channel	Deletes the email channel.
	Delete ITSM Channel	Deletes the ITSM channel.
	List Channels With Filters Applied	Lists channels with specific filters applied.
	List Existing Channels	Lists existing channels.
Mail Server	Delete Configured Mail Server	Deletes the mail server configuration.
	Retrieve Mail Server	Retrieves the mail server information.
	Save Mail Server Configuration	Saves the mail server configuration.
	Test Mail Server Configuration	Tests the mail server configuration.
Message Template	Delete Message Template	Deletes the message template.
	List Message Templates	Lists the message template.
	Retrieve Linked Template Names	Retrieves linked template names.
	Retrieve Template Details	Retrieves the template details.
Policy	Create Policy	Creates a policy.
	Delete Policy	Deletes a policy.
	List All Policies	Lists all policies.
	List Specific Policy Details	Lists specific policy details.
	Update Policy by Linking Channel	Updates a policy by linking the channel.

Product Usage Collector API

This section describes how to use the Product Usage Collector (PUC) API.

- Create a Telemetry PUC User
- API Usage

Create a Telemetry PUC User

Before you start using the PUC API, create a user with scope limited to PUC retrieval. Also, generate a user token to access the AIOps PLA Telemetry information.

NOTE
The following steps are for a basic auth-enabled tenant.

Follow these steps:

1. Ensure that the **esdplatelemetry_onpreminfo.properties** file is populated. For more information, see **Configure Telemetry** in the **Post-Installation Tasks** sections.
2. Log in as a Tenant Administrator for the tenant that is configured in the **dxplatelemetry/configcommon/esdplatelemetry_onpreminfo.properties** file. See the value for the **dxitenantid** key. For more information, see **Configure Telemetry** in the **Post-Installation Tasks (Cluster Administrator)**, **Post-Installation Tasks (Cluster Reader)**, and **Post-Installation Tasks (Namespace Administrator)** sections.
3. Create a role named **telemetry_puc**:
   a. Navigate to the **Settings > Roles** page.
   b. Click the **+ New Role** button.
   c. Provide the following information:
      a. **Name:** Enter **telemetry_puc** as the name.
      b. **Description:** Enter the description.
      c. **Active:** Select the **Active** checkbox.
      d. **Accesses:** Select the following access: **DX PLATFORM > All Features > Telemetry > Product Usage Collector** checkbox.
   d. Click the **Create** button.
4. Create a user named **telemetry_puc**.
   a. Navigate to the **Settings > Users** page.
   b. Click the **+ Add user** button to create a user that will be referenced as the telemetry user.
   c. Provide the following information:
      a. **First Name and Last Name**: Enter the first and last name. For example, enter the First Name as Telemetry and Last Name as PUC.
      b. **Email:** Enter the email address. You can enter your personal email too.

         NOTE
         If you are unable to save the User Details due to the email address being in use, save the details using a fictitious email address. After the details are saved, edit the email to the desired entry.
      c. **Role:** Select the **telemetry_puc** role that you had created earlier.
      d. **User name:** Enter **telemetry_puc** as the username.
   d. Click the **Save** button.
5. Update the password for the **telemetry_puc** user.
   a. Open the **telemetry_puc** user.
   b. Click the **Change password** button.
   c. Enter the new password.
   d. Click the **Save** button.
6. Log out of the tenant.
7. Log in to the tenant again as the **telemetry_puc** user.
8. Generate the user token.
   a. Navigate to the **Settings > Tokens** page.
   b. Click the **+ New Token** button.
   c. Provide the following information:
      a. **Name:** Enter **telemetry_puc_token01** as the name.
      b. **Type:** Select the **user** radio button.
      c. Click the **Generate** button.
   d. Copy the generated token to the clipboard and store it for future use.

This token will be referenced as the API Key within PUC for accessing the data.

API Usage

The following table provides the information that is required to use this API:

Name	Description
Endpoint	`https or http://<dxi-adminui.route>/spp/v5/telemetry/puc?start_date={start_date_epoc_millis}&end_date={end_date_epoc_millis}`
Method	GET
Query Params	The following parameters are used for a range to fetch the usage information: • **start_date:** Pass the start date in epoch milliseconds. • **end_date:** Pass the end date in epoch milliseconds.
Authorization	The endpoint is accessible by a Tenant Administrator bearer token authorization. However, we recommend creating a user with a role having the **Telemetry > PUC** access in the target tenant to limit the telemetry information that is retrieved. For more information, see the Create a Telemetry PUC User section on this page.

Authorization: Header	`Bearer <token>`
	<token> is the user token that was created earlier.
	Example of the authorization header with API Key:
	`Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJFUzI1NiJ9.eyJzdWIiOiJURVNUX1RFTkFOVEBURVNULkNPTSIsInR u9BunAtMkIT2b3MNoqnTZYdFl244FTRGYfTVEtQEkdBtG8Y7wkA`

Curl Example	milliseconds 01/01/2023 at midnight (12:00:00 AM) = 1672531200000 milliseconds 03/31/2023 at midnight (11:59:59 PM) = 1680307199000 `$ curl -H "Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJFUzI1NiJ9.eyJzdWIiOiJURVNUX1RFTkFOVEBURVNULkNPTSIsInR u9BunAtMkIT2b3MNoqnTZYdFl244FTRGYfTVEtQEkdBtG8Y7wkA" http://adminui.11.239.228.00.nip.io/spp/v5/telemetry/puc?start_date=1672531200000&end_date=1680307199000`

Add or Update the CI Attributes

Use the Add or Update CI Attributes API to add the CI attributes to your message templates, policies, tickets, and channels (Slack, and Webhook). You can also use this API to update the attributes list. Examples of CI attributes are AWS_RDS_Maintenance, AWS_Resource_Group, AWS_Type, and so on.

NOTE

- If the attributes are not registered, the CI attributes will not be available for selection.
- You can register only a maximum of 50 CI attributes.
- The CI attributes are supported only for All Alarms.
- If the CI attribute does not have any value, then the default value is displayed for that attribute. If the default value is not defined, then the incident field value is blank or is not populated.
- You can get the list of the CI attributes from the **Entity Details** section on the **Monitored Inventory** page in DX Operational Intelligence. For more information about how to get the attributes, see the Monitored Inventory section.

Before you add or update the CI attributes, use the following information to check if any attributes are already added.

Name	Description
URL	`https://doi.dxi-na1.saas.broadcom.com/oi/v3/api/notifications/configAttributes?tenantId=<cohort_ID>` The cohort ID is available on the **Settings > Connector Parameters** page.
Method	GET
HTTP Header: Content-Type	`application/json`
HTTP Headers: Authorization	`Bearer <BASE64(JSON(AuthZRequestHdrBean))>`
Curl Command:	`curl --location --request GET 'https://doi.dxi-na1.saas.broadcom.com/oi/v3/api/notifications/configAttributes?tenantId=<cohort_ID>'`
Sample Response	`200 OK` The CI attributes are displayed if they were added.

Use the following information to add or update the CI attributes:

NOTE
If you add or update the attributes, you must send the entire payload in the request and not only the added or updated attributes.

Name	Description
Method	PUT
URL	`https://doi.dxi-na1.saas.broadcom.com/oi/v3/api/notifications/configAttributes?tenantId=<cohort_ID>` The cohort ID is available on the **Settings > Connector Parameters** page.
HTTP Header: Content-Type	`application/json`
HTTP Headers: Authorization	`Bearer <BASE64(JSON(AuthZRequestHdrBean))>`
Curl Command:	`curl --location --request PUT 'https://doi.dxi-na1.saas.broadcom.com/oi/v3/api/notifications/configAttributes?tenantId=<cohort_ID>'`

Name	Description
Sample Request	In the following sample code, fieldDescription is the label that is displayed on the UI. • The Field value is case-sensitive. Ensure that the value for field matches the TAS attribute. • Every new PUT request will override the existing request data. [{ "fieldDescription":"AWS_RDS_Maintenance", "field":"AWS_RDS_Maintenance", "Values":[ ], "fieldType":"String" }, { "fieldDescription":"AWS_RDS_Port", "field":"AWS_RDS_Port", "Values":[ ], "fieldType":"String" }, { "fieldDescription":"AWS_RDS_Storage-Full", "field":"AWS_RDS_Storage-Full", "Values":[ ], "fieldType":"String" }, { "fieldDescription":"AWS_Region", "field":"AWS_Region", "Values":[ ], "fieldType":"String" }, { "fieldDescription":"AWS_Resource_Group", "field":"AWS_Resource_Group", "Values":[ ], "fieldType":"String" }, { "fieldDescription":"AWS_RDS_Storage-Full", "field":"AWS_RDS_Storage-Full", "values":[ ], "fieldType":"String"
	 }, { "fieldDescription":"AWS_tag.CostCenter", "field":"AWS_tag.CostCenter", "values":[

Name	Description
Sample Response	`200 OK` If the request fails, rerun the PUT call.

Authenticate User

Using the Authenticate API, you can authenticate a user.

Name	Description
Method	POST
URL	`https://axa.dxi-na1.saas.broadcom.com/ess/security/v1/token` `http or https://<dxi-adminui.route>/ess/security/v1/token`
Header	`Authorization : Bearer <BASE64(JSON(AuthZRequestHdrBean))>`

Sample Request	`grant_type:`
	`PASSWORD`
	
	`username`
	`xyz@mailinator.com`
	
	`password:`
	`<*********>`

Create Email Channel

Using the Create Email Channel API, you can create the email channel.

Name	Description
Method	PUT
URL	`https://axa.dxi-na1.saas.broadcom.com/ess/notify/v1/channels` `http or https://<dxi-adminui.route>/ess/notify/v1/channels`
Header	`Bearer <BASE64(JSON(AuthZRequestHdrBean))>`

Name	Description

Sample Request	{
	"id":0,
	"orgId":"DXI-APMQA",
	"keyLevel":1,
	"name":"TestEmailChannel",
	"protocol":"SMTP",
	"subProtocol":"SMTP",
	"status":false,
	"author":"user01",
	"filterNames":[
	
	],
	"smtpItsmConfigs":{
	"templateName":"NotifyAlarmTemplate",
	"templateLocale":"en_US",
	"addresses":[
	"user011@broadcom.com"
	]
	},
	"maskRecipients":false
	}

Sample Response	{
	"newContentVersion" : 0
	}

Create Policy

Use the Create Policy API to create a policy without a channel or message template associated.

Name	Description
Method	PUT
URL	https://axa.dxi-na1.saas.broadcom.com/ess/notify/v1/filters http or https://<dxi-adminui.route>/ess/notify/v1/filters
Header	Bearer <BASE64(JSON(AuthZRequestHdrBean))>

Name	Description

Sample Request	• **Service Alarm**
	{
	"orgId":"DXI-APMQA",
	"id":15112,
	"keyLevel":1,
	"name":"Service_Alarm_Policy",
	"channelNames":[
	
	],
	"description":"description",
	"filterExpression":{
	"not":false,
	"op":"AND",
	"serviceAlarmAction":true,
	"alarmCategory":"service",
	"filterExpressions":[
	{
	"not":false,
	"op":"OR",
	"serviceAlarmAction":true,
	"alarmCategory":"service",
	"filterCriteria":[
	{
	"attr":"alarmType",
	"attrDesc":"Alarm Type",
	"op":"EQ",
	"filterValue":"Service",
	"not":false
	}
	]
	},
	{
	"not":false,
	"op":"OR",
	"serviceAlarmAction":true,
	"alarmCategory":"service",
	"filterCriteria":[
	{
	"attr":"state",
	"attrDesc":"Alarm Status",
	"op":"EQ",
	"filterValue":"Active",
	"not":false
	}
	]
	}
	]
	},
	"jobInfo":{
	"configId":"",
	"jobsPayload":[
	""
	]

	}, "linkedEntities":[ ], "author":"user01"

Name	Description

Sample Response	`{`
	`"newContentVersion":1`
	`}`

Delete Configured Mail Server

Using the Delete Configured Mail Server API, you can delete the configured mail server.

Name	Description
Method	DELETE

URL	`http://axa.dxi-na1.saas.broadcom.com/ess/notify/v1/providers/{provider_name}/{orgid}`
	Example:
	`http://axa.dxi-na1.saas.broadcom.com/ess/notify/v1/providers/SystemProvider/TEST1_TENANT`
	`http://<dxi-adminui.route>/ess/notify/v1/providers/{provider_name}/{orgid}`
	Example:
	`http://<dxi-adminui.route>/ess/notify/v1/providers/SystemProvider/TEST1_TENANT`

Header	`Bearer <BASE64(JSON(AuthZRequestHdrBean))>provider_name` `orgid`

Sample Response	`{`
	`"deletedCount" : 1`
	`}`

Delete Email Channel

Using the Delete Email Channel API, you can delete the Email Channel.

Name	Description
Method	DELETE
URL	`https://axa.dxi-na1.saas.broadcom.com/ess/notify/v1/channels/{orgid}/{channel_id}` `http or https://<dxi-adminui.route>/ess/notify/v1/channels/{orgid}/{channel_id}`

Header	`Bearer <BASE64(JSON(AuthZRequestHdrBean))>`
	`orgid`
	`channel_id`

Name	Description

Sample Response	`{`
	`"deletedCount" : 1`
	`}`

Delete ITSM Channel

Using the Delete ITSM Channel API, you can delete the ITSM Channel.

Name	Description
Method	DELETE
URL	`https://axa.dxi-na1.saas.broadcom.com/ess/notify/v1/channels/{orgid}/{channel_id}` `http or https://<dxi-adminui.route>/ess/notify/v1/channels/{orgid}/{channel_id}`

Header	`Bearer <BASE64(JSON(AuthZRequestHdrBean))>`
	
	`orgid`
	
	`channel_id`

Sample Response	`{`
	`"deletedCount" : 1`
	`}`

Delete Message Template

Using the Delete Message Template API, you can delete the message template.

Name	Description
Method	DELETE
URL	`https://axa.dxi-na1.saas.broadcom.com/ess/notify/v1/templates/{template_name}/{locale}/{orgid}` `http or https://<dxi-adminui.route>/ess/notify/v1/templates/{template_name}/{locale}/{orgid}`
Header	`Bearer <BASE64(JSON(AuthZRequestHdrBean))>`

Sample Response	`{`
	`"deletedCount" : 1`
	`}`

Delete Policy

Using the Delete Policy API, you can delete a policy.

Name	Description
Method	DELETE
URL	`https://axa.dxi-na1.saas.broadcom.com/ess/notify/v1/filters/{orgid}/{filtername}` `http or https://<dxi-adminui.route>/ess/notify/v1/filters/{orgid}/{filtername}`
Header	`Bearer <BASE64(JSON(AuthZRequestHdrBean))>`

Sample Response	`{`
	`"deletedCount" : 1`
	`}`

List All Policies

Using this API, you can list all the policy details.

Name	Description
Method	GET
URL	`https://axa.dxi-na1.saas.broadcom.com/ess/notify/v1/filters/{orgid}` `http or https://<dxi-adminui.route>/ess/notify/v1/filters/{orgid}`

Header	`Bearer <BASE64(JSON(AuthZRequestHdrBean))>`
	
	`orgid`

Name	Description

Sample Response	{
	"filters":[
	{
	"id":11326,
	"orgId":"DXI-APMQA",
	"keyLevel":1,
	"name":"1",
	"description":"description",
	"contentVersion":0,
	"filterExpression":{
	"not":false,
	"op":"AND",
	"filterExpressions":[
	{
	"not":false,
	"op":"AND",
	"filterCriteria":[
	{
	"attr":"alarmType",
	"op":"NE",
	"filterValue":"Service",
	"not":false
	}
	],
	"serviceAlarmAction":false,
	"alarmCategory":"rawAlarm"
	},
	{
	"not":false,
	"op":"OR",
	"filterCriteria":[
	{
	"attr":"management_module",
	"op":"EQ",
	"filterValue":"ddddd",
	"not":false
	}
	],
	"serviceAlarmAction":false,
	"alarmCategory":"rawAlarm"
	}
	],
	"serviceAlarmAction":false,
	"alarmCategory":"rawAlarm"
	},
	"channelNames":[
	
	],
	"created":"2022-01-07T05:03:01.359+0000",
	"lastUpdated":"2022-01-07T05:03:01.359+0000",
	"alarmViews":[
	
	],

	"isTimeBasedPolicy":0, "linkedEntities":[ ], "filterFlag":0

List Channels With Filters Applied

Using this API, you can list channels with filters applied.

Name	Description
Method	GET
URL	`https://axa.dxi-na1.saas.broadcom.com/ess/notify/v1/channels/{orgid}` `http or https://<dxi-adminui.route>/ess/notify/v1/channels/{orgid}`

Header	`Bearer <BASE64(JSON(AuthZRequestHdrBean))>`
	
	`orgid`

Name	Description

Sample Response	{
	"filters":[
	{
	"id":11326,
	"orgId":"DXI-APMQA",
	"keyLevel":1,
	"name":"1",
	"description":"description",
	"contentVersion":0,
	"filterExpression":{
	"not":false,
	"op":"AND",
	"filterExpressions":[
	{
	"not":false,
	"op":"AND",
	"filterCriteria":[
	{
	"attr":"alarmType",
	"op":"NE",
	"filterValue":"Service",
	"not":false
	}
	],
	"serviceAlarmAction":false,
	"alarmCategory":"rawAlarm"
	},
	{
	"not":false,
	"op":"OR",
	"filterCriteria":[
	{
	"attr":"management_module",
	"op":"EQ",
	"filterValue":"ddddd",
	"not":false
	}
	],
	"serviceAlarmAction":false,
	"alarmCategory":"rawAlarm"
	}
	],
	"serviceAlarmAction":false,
	"alarmCategory":"rawAlarm"
	},
	"channelNames":[
	
	],
	"created":"2022-01-07T05:03:01.359+0000",
	"lastUpdated":"2022-01-07T05:03:01.359+0000",
	"alarmViews":[
	
	],

	"isTimeBasedPolicy":0, "linkedEntities":[ ], "filterFlag":0

List Existing Channels

Using this API, you can list all the existing channels.

Name	Description
Method	GET
URL	`https://axa.dxi-na1.saas.broadcom.com/ess/notify/v1/channels/{orgid}` `http or https://<dxi-adminui.route>/ess/notify/v1/channels/{orgid}`
Header	`Bearer <BASE64(JSON(AuthZRequestHdrBean))>orgid`

Name	Description

Sample Response	{
	"channels":[
	{
	"id":91,
	"orgId":"DXI-APMQA",
	"keyLevel":1,
	"name":"0829",
	"protocol":"SMTP",
	"subProtocol":"SMTP",
	"status":false,
	"author":"user01",
	"contentVersion":11,
	"filterNames":[
	
	],
	"smtpItsmConfigs":{
	"channelId":91,
	"templateName":"NotifyAlarmTemplate",
	"templateLocale":"en_US",
	"templateContentVersion":0,
	"addresses":[
	"user01@broadcom.com"
	]
	},
	"created":"2019-08-29T05:48:43.161+0000",
	"lastUpdated":"2019-11-22T05:09:16.660+0000",
	"maskRecipients":false,
	
	"proxyEnabled":false,
	"proxyId":0,
	"linkedEntities":[
	
	],
	"enableProxy":false
	},
	{
	"id":661,
	"orgId":"DXI-APMQA",
	"keyLevel":1,
	"name":"1-kir1",
	"protocol":"SMTP",
	"subProtocol":"SMTP",
	"status":false,
	"author":"user01",
	"contentVersion":96,
	"filterNames":[
	
	],
	"smtpItsmConfigs":{
	"channelId":661,
	"templateName":"KirEmailTempl",
	"templateLocale":"en_US",
	"templateContentVersion":0,
	"addresses":[
	"user02@broadcom.com"
	]
	},
	"created":"2020-03-11T06:37:37.392+0000",

List Message Templates

Using this API, you can list the message template.

Name	Description
Method	GET
URL	`https://axa.dxi-na1.saas.broadcom.com/ess/notify/v1/templates` `http or https://<dxi-adminui.route>/ess/ess/notify/v1/templates`
Header	`Bearer <BASE64(JSON(AuthZRequestHdrBean))>`

Name	Description

Sample Response	{
	"templates":[
	{
	"orgId":"DXI-APMQA",
	"keyLevel":1,
	"templateName":"7799TEST",
	"locale":"en_US",
	"defaultTemplate":"false",
	"contentVersion":0,
	"subjectTemplate":"Message: ${message}",
	"senderNameTemplate":"DX SaaS Team",
	"sender":"dxi-na1@saas.broadcom.com",
	"replyTo":"no-reply-dxi-na1@saas.broadcom.com",
	"imageUsage":0,
	
	"templateText":"JHt0aW1lc3RhbXB9Ck1lc3NhZ2U6ICR7bWVzc2FnZX0KTWV0cmljIE5hbWU6ICR7
	"channelNames":[
	
	],
	"created":"2020-10-19T06:44:42.168+0000",
	"lastUpdated":"2020-10-19T06:44:42.168+0000",
	"linkedEntities":[
	
	]
	},
	{
	"orgId":"DXI-APMQA",
	"keyLevel":1,
	"templateName":"template 2",
	"locale":"en_US",
	"defaultTemplate":"false",
	"contentVersion":10,
	"subjectTemplate":"Invalid Message Test ",
	"senderNameTemplate":"DX SaaS Team",
	"sender":"dxi-na1@saas.broadcom.com",
	"replyTo":"no-reply-dxi-na1@saas.broadcom.com",
	"imageUsage":0,
	
	"templateText":"ZGttPT1sZGpza2ZsaEZMaGZubEZIREZKTEhmbGtmamRza2YKClRlc3Q=",
	"channelNames":[
	"slackchannel"
	],
	"created":"2021-11-16T12:20:02.175+0000",
	"lastUpdated":"2021-11-16T12:20:02.175+0000",
	"linkedEntities":[
	{
	"channelName":"slackchannel",
	"messageTemplateName":null,
	"filterName":null
	}
	]
	}

List Specific Policy Details

Using this API, you can list the specific policy details.

Name	Description
Method	GET
URL	`https://axa.dxi-na1.saas.broadcom.com/ess/notify/v1/filters/{orgid}/{filtername}` `http or https://<dxi-adminui.route>/ess/notify/v1/filters/{orgid}/{filtername}`

Header	`Bearer <BASE64(JSON(AuthZRequestHdrBean))>`
	
	`orgid`
	
	`filtername`

Name	Description

Sample Response	{
	"notifyFilter":{
	"id":15118,
	"orgId":"DXI-APMQA",
	"keyLevel":1,
	"name":"Situation_Policy",
	"description":"description",
	"contentVersion":2,
	"filterExpression":{
	"not":false,
	"op":"AND",
	"filterExpressions":[
	{
	"not":false,
	"op":"OR",
	"filterCriteria":[
	{
	"attr":"alarmType",
	"op":"EQ",
	"filterValue":"SituationCluster",
	"not":false
	}
	],
	"serviceAlarmAction":false,
	"alarmCategory":"SituationCluster"
	},
	{
	"not":false,
	"op":"OR",
	"filterCriteria":[
	{
	"attr":"isOrphan",
	"op":"EQ",
	"filterValue":"false",
	"not":false
	}
	],
	"serviceAlarmAction":false,
	"alarmCategory":"SituationCluster"
	},
	{
	"not":false,
	"op":"OR",
	"filterCriteria":[
	{
	"attr":"clusteringType",
	"op":"EQ",
	"filterValue":"Custom",
	"not":false
	},
	{
	"attr":"clusteringType",
	"op":"EQ",
	"filterValue":"Spectrum",
	"not":false
	}
	],
	"serviceAlarmAction":false,

Retrieve Auth Config

Using this API, you can retrieve the authentication configuration.

Name	Description
Request Method	GET
Request URL	`https://axa.dxi-na1.saas.broadcom.com/ess/security/v1/authconfig/<cohortID>` `http or https://<dxi-adminui.route>/ess/security/v1/authconfig/<cohortID>`
Header	`Authorization : Bearer <BASE64(JSON(AuthZRequestHdrBean))>`

Sample Response	`{`
	`  "cohort": "<cohortID>",`
	`  "moduleName": "BASIC_AUTH",`
	`  "url": "/ess/login?orgid=<cohortID>"`
	`}`

Retrieve Licensed SKUs

You can retrieve the licensed SKUs using the following APIs:

- Trigger Recalculation of plaTenantsXmlfile Content
- Download plaTenantsXmlfile Content

Trigger Recalculation of plaTenantsXmlfile Content

Use this API to recalculate the PLA tenants.

Name	Description
Resource URI	`https://doi.dxi-na1.saas.broadcom.com/spp/v5/telemetry/pla/saas/platenantsxml/recalculate` `http or https://<doi.route>/spp/v5/telemetry/pla/saas/platenantsxml/recalculate`
Method	POST
HTTP Headers	`Content-Type: text/plain`
Authorization	`Bearer {bearerToken/userToken}`
Sample Curl Command	`curl --location --request POST 'dxi-adminui.broadcom.net/spp/v5/telemetry/pla/saas/platenantsxml/recalculate' \--header 'Authorization: Bearer eyJ0a24iOiI5YTEyMDhmYy01MTliLTQzYjYtYTNhOS02Y2M4NzBmZGJhNTciLCJhbGwiOnRydWV9Cg==`

Download plaTenantsXmlfile Content

Use this API to download the XML file that has the calculated PLA Tenants.

Name	Description
Resource URI	`https://doi.dxi-na1.saas.broadcom.com/oi/v2/api/maintenance/service-schedules-history` `http or https://<doi.route>/oi/v2/api/maintenance/service-schedules-history`
Method	POST
HTTP Headers	`Content-Type: text/plain`
Authorization	`Bearer`
Sample Curl Command	`curl --location 'https://dxi-adminui.broadcom.net/spp/v5/telemetry/pla/saas/platenantsxml' \--header 'Content-Type: text/plain' \--header 'Authorization: Bearer eyJ0a24iOiJhMmQyYWNlOC0wNmQyLTRlNDAtOWM3ZC05MTIwNGExMGZlNzEiLCJhbGwiOnRydWV9Cg==`

Name	Description
Sample Response	"platenants": [{ "tenantid": "FLAG_TEST", "cohortid": "34E0B7A9-9327-4168-99A7-16EAA9BB95E9", "siteid": "123456", "domainname": "FLAG_TEST_domain", "plaenabled": "true", "chargebackid": "", "sku": "DXSANU990", "subscriptions": null, "rootElement": "platenant", "cohortidAsSet": ["34E0B7A9-9327-4168-99A7-16EAA9BB95E9"], "skuAsSet": ["DXSANU990"] }, { "tenantid": "FLAG_TEST1", "cohortid": "B4D6BA1D-DE5C-46AA-9BB7-6F574A49D213", "siteid": "123456", "domainname": "FLAG_TEST1_domain", "plaenabled": "true", "chargebackid": "", "sku": "DXSANU990", "subscriptions": null, "rootElement": "platenant", "cohortidAsSet": ["B4D6BA1D-DE5C-46AA-9BB7-6F574A49D213"], "skuAsSet": ["DXSANU990"] }, { "tenantid": "DEFECT_TEST", "cohortid": "47C5AB19-4D74-4656-9C5E-7F92F31CBFA9", "siteid": "123456", "domainname": "DEFECT_TEST_domain", "plaenabled": "true", "chargebackid": "", "sku": "DXSANU990", "subscriptions": null, "rootElement": "platenant", "cohortidAsSet": ["47C5AB19-4D74-4656-9C5E-7F92F31CBFA9"], "skuAsSet": ["DXSANU990"] }, {
	"tenantid": "DP_UI_AUTOMATION", "cohortid": "2304AA1D-75F3-4FC1-8D7A-0154DEE4F465", "siteid": "123456", "domainname": "DP_UI_AUTOMATION_domain", "plaenabled": "true",

Retrieve Linked Template Names

Using this API, you can retrieve the linked template names.

Name	Description
Request Method	GET

Request URL	`https://axa.dxi-na1.saas.broadcom.com/ess/notify/v1/templates`
	`Example URL: https://axa.dxi-na1.saas.broadcom.com/ess/notify/v1/templates?orgid=DXI-APMQA`
	`http or https://<dxi-adminui.route>/ess/notify/v1/templates`
	`Example URL: http or https://<dxi-adminui.route>/ess/notify/v1/templates?orgid=DXI-APMQA`

Header	`Bearer <BASE64(JSON(AuthZRequestHdrBean))>`

Name	Description

Sample Response	{
	
	"templates":[
	{
	"orgId":"DXI-APMQA",
	"keyLevel":1,
	"templateName":"DefaultTicketingManagementTemplate",
	"locale":"en_US",
	"defaultTemplate":"false",
	"contentVersion":0,
	"subjectTemplate":"",
	"senderNameTemplate":"",
	"sender":"",
	"replyTo":"",
	"imageUsage":0,
	
	"templateText":"QWxhcm0gSUQ6ICR7YWxhcm1fdW5pcXVlX2lkfQ0KTWVzc2FnZTogJHttZXNzYWdl
	0cmljX25hbWV9DQpPSSBBbGFybTogJERPSV9BZG1pbl9VSV9VUkwvZGlnaXRhbC1vaS9hbGFybXMtYW5h
	VfaWR9JmZyb21UaW1lPSR7c3RhcnRUaW1lfSZ2aWV3VHlwZT1hbGxBbGFybXMNCkFsYXJtIHR5cGU6ICR
	RldmljZSBJUDogJHtpcH0NClByb2R1Y3QgRGV0YWlsczogJHtwcm9kdWN0X2RldGFpbHN9DQpQcm9kdWN
	Byb2R1Y3RfdmVyc2lvbn0NCkFsYXJtVVJMOiAke2FsYXJtVVJMfQ==",
	"channelNames":[
	"ITSM1"
	],
	"created":"2019-09-25T14:36:45.934+0000",
	"lastUpdated":"2019-09-25T14:36:45.934+0000"
	},
	{
	"orgId":"DXI-APMQA",
	"keyLevel":1,
	"templateName":
	"NotifyAlarmTemplate",
	"locale":"en_US",
	"defaultTemplate":"true",
	"contentVersion":0,
	"subjectTemplate":"Alarm from Digital Experience Insights
	from CA",
	"senderNameTemplate":"CA Digital Experience Insights Team",
	
	"sender":"sender@ca.com",
	"replyTo":"donotreply@ca.com",
	"imageUsage":0,
	
	"templateText":"Q0EgVGVjaG5vbG9naWVzCk5FVyBBTEFSTQoKQ0EgRGlnaXRhbCBFeHBlcmllbmNl
	XB9Cgoke21lc3NhZ2V9CgpNZXRyaWMgTmFtZTogJHttZXRyaWNfbmFtZX0KClNldmVyaXR5OiAke3Nldm
	5cGV9Ckhvc3Q6ICR7aG9zdH0KSVA6ICR7aXB9CkFsYXJtIElEOiAke2FsYXJtX3VuaXF1ZV9pZH0KClBy
	XJzaW9uOiAke3Byb2R1Y3RfdmVyc2lvbn0K",
	"channelNames":[
	"Mail1"
	],
	"created":"2019-09-25T14:36:45.916+0000",
	"lastUpdated":"2019-09-25T14:36:45.916+0000"
	}
	]
	}

Retrieve Mail Server

Using this API, you can retrieve the mail server configuration.

Name	Description
Request Method	GET

Request URL	`https://axa.dxi-na1.saas.broadcom.com/ess/notify/v1/{provider_name}`
	`Example - https://axa.dxi-na1.saas.broadcom.com/ess/notify/v1/providers/SystemProvider/DXI-APMQA`
	`http or https://<dxi-adminui.route>/ess/notify/v1/{provider_name}`
	`Example - http or https://<dxi-adminui.route>/ess/notify/v1/providers/SystemProvider/DXI-APMQA`

Header	`Bearer <BASE64(JSON(AuthZRequestHdrBean))>provide_name`

Sample Response	`{`
	`  "orgId": "DXI-APMQA",`
	`  "keyLevel": 1,`
	`  "name": "SystemProvider",`
	`  "protocol": "SMTP",`
	`  "url": "smtp://<smtpserver:port>",`
	`  "account": "",`
	`  "credential": "",`
	`  "options": "",`
	`  "created": "2019-09-25T14:35:34.348+0000",`
	`  "lastUpdated": "2019-10-16T20:42:10.925+0000"`
	`}`

Retrieve Template Details

Using this API, you can retrieve all the template details.

Name	Description
Request Method	GET

Request URL	`https://axa.dxi-na1.saas.broadcom.com/ess/notify/v1/templates/{template_name}/{locale}/{orgid}`
	`Example - https://axa.dxi-na1.saas.broadcom.com/ess/notify/v1/templates/DefaultTicketingManagementTemplate/en_US/DXI-APMQA`
	`http or https://<dxi-adminui.route>/ess/notify/v1/templates/{template_name}/{locale}/{orgid}`
	`Example - http or https://<dxi-adminui.route>/ess/notify/v1/templates/DefaultTicketingManagementTemplate/en_US/DXI-APMQA`

Name	Description

Header	`Bearer <BASE64(JSON(AuthZRequestHdrBean))>`
	
	`template_name`
	
	`orgid`
	
	`locale`

Sample Response	{
	"orgId" : "DXI-APMQA",
	"keyLevel" : 1,
	"templateName" : "DefaultTicketingManagementTemplate",
	"locale" : "en_US",
	"defaultTemplate" : "false",
	"contentVersion" : 0,
	"subjectTemplate" : "",
	"senderNameTemplate" : "",
	"sender" : "",
	"replyTo" : "",
	"imageUsage" : 0,
	"templateText" :
	"QWxhcm0gSUQ6ICR7YWxhcm1fdW5pcXVlX2lkfQ0KTWVzc2FnZTogJHttZXNzYWdlfQ0KTWV
	0cmljIE5hbWU6ICR7bWV0cmljX25hbWV9DQpPSSBBbGFybTogJERPSV9BZG1pbl9VSV9VUkwvZGlnaXRh
	GFybXMtYW5hbHl0aWNzP2FsYXJtSWQ9JHthbGFybV91bmlxdWVfaWR9JmZyb21UaW1lPSR7c3RhcnRUaW
	3VHlwZT1hbGxBbGFybXMNCkFsYXJtIHR5cGU6ICR7YWxhcm1fdHlwZX0NCkhvc3Q6ICR7aG9zdH0NCkRl
	ogJHtpcH0NClByb2R1Y3QgRGV0YWlsczogJHtwcm9kdWN0X2RldGFpbHN9DQpQcm9kdWN0OiAke3Byb2R
	cm9kdWN0IElEOiAke3Byb2R1Y3RfdmVyc2lvbn0NCkFsYXJtVVJMOiAke2FsYXJtVVJMfQ==",
	"channelNames" : [ ],
	"created" : "2019-09-25T14:36:45.934+0000",
	"lastUpdated" : "2019-09-25T14:36:45.934+0000"
	}

Retrieve Tenant Cohort ID

You can retrieve a Tenant Cohort ID in one of the following ways:

- Using the Google Chrome Developer tools
- Using the following curl command:

```
curl https://https://axa.dxi-na1.saas.broadcom.com/ess/security/v1/authconfig/<TENANT_ID>
curl https://<dxi-adminui.route>/ess/security/v1/authconfig/<TENANT_ID>
```

Sample:


```
{
  "cohort" : "<cohortID>",
  "moduleName" : "BASIC_AUTH",
  "url" : "/ess/login?orgid=<cohortID>",
  "params" : "{\"issuer\":\"urn:<urn>\",\"entityId\":\"DXI_DXI-APMQA\",\"skewTime\":500,\"autoCreateUser
\":true,\"attrMap\":{\"e\":\"email\",\les\"},\"cookieDomains\":[\".ca.com\"]}"
```

Save Mail Server Configuration

Using this API, you can save the mail server configuration.

Name	Description
Method	PUT
URL	`https://axa.dxi-na1.saas.broadcom.com/ess/notify/v1/providers` `http or https://<dxi-adminui.route>/ess/notify/v1/providers`
Header	`Bearer <BASE64(JSON(AuthZRequestHdrBean))>`

Sample Request	`{orgId: "DXI-APMQA", url: "smtp://<smtpserver:port>", keyLevel: 1,…}`
	`account: ""`
	`credential: ""`
	`keyLevel: 1`
	`name: "SystemProvider"`
	`options: ""`
	`orgId: "DXI-APMQA"`
	`protocol: "SMTP"`
	`url: "smtp://<smptserver:port>"`

Sample Response	`{`
	`  "tenantId": "string",`
	`  "url": "string",`
	`  "authType": "string",`
	`  "account": "string",`
	`  "credential": "string",`
	`  "recipients": [`
	`  "string"`
	`  ],`
	`  "useDefaultConfigs": "boolean",`
	`  "product": "string",`
	`  "templateName": "string",`
	`  "maskRecipients": "boolean"`
	`}`

Test Mail Server Configuration

Using this API, you can retrieve the test mail server configuration.

Name	Description
Method	PUT
URL	`https://axa.dxi-na1.saas.broadcom.com/ess/notify/v1/testProvider` `http or https://<dxi-adminui.route>/ess/notify/v1/testProvider`
Header	`Bearer <BASE64(JSON(AuthZRequestHdrBean))>`

Name	Description

Sample Request	`orgId: "DXI-APMQA", url: "smtp://<smtpserver:port>", keyLevel: 1,…}`
	`account: ""`
	`credential: ""`
	`keyLevel: 1`
	`name: "SystemProvider"`
	`options: ""`
	`orgId: "DXI-APMQA"`
	`product: "DX SaaS"`
	`protocol: "SMTP"`
	`recipients: ["xyz@mailinator.com"]`
	`templateName: "SMTPConfigValidateTemplate"`
	`url: "smtp://<smtpserver:port>"`
	`useDefaultConfigs: false`

Sample Response	`{`
	`"msg" : "Successfully sent email to provided recipients"`
	`}`

Update Possible Values Mapping of Incident Fields

Use this API to update the possible values mapping of the incident fields.

Retrieve Existing Possible Values Mapping

Use the following information to retrieve the existing possible values mapping:

- **Resource URI:** http://<doi_admin_ui_url>/oi/v3/api/notifications/nimMapping
- **Method:** GET
- **HTTP Headers:**
  - **Content-Type:** application/JSON
  - **Authorization:** User token that can be generated on the Token Management page.
- **Sample Request:**

```
GET : http://doi-adminui.67029057.10.005.2.12.nip.io/oi/v3/api/notifications/nimMapping
```

- **Sample Response:**

  NOTE
  For the complete payload, click here to download.

```
{
    "name": "incident",
    "mappedTo": "incident",
    "description": "CA NIM incident",
    "mappings": [
        {
            "name": "affectedCIID",
            "mappedTo": "cmdb_ci",
            "description": "CI on which Incident is created",
            "dataType": "string",
            "default": "",
            "custom": false,
            "modificationTime": 1633656193860
        },
```

```
            {
                "name": "id",
                "mappedTo": "sys_id",
                "description": "Unique Identifier of the issue generated by the system",
                "dataType": "string",
                "default": "",
                "custom": false,
                "modificationTime": 1633656193860
            },
            {
                "name": "assigneeUserID",
                "mappedTo": "assigned_to",
                "description": "User to which the incident is assigned",
                "dataType": "string",
                "default": "",
                "custom": false,
                "modificationTime": 1633656193860
            },
            {
                "name": "status",
                "mapped": "state",
                "description": "Status of the Incident",
                "dataType": "string",
                "default": "",
                "custom": false,
                "possibleValues": [
                      "New=1",
                      "Active=2",
                      "Awaiting Problem=3",
                      "Awaiting User Info=4",
                      "Awaiting Evidence=5",
                      "Resolved=6",
                      "Closed=7"
        ],
    },..
```

Update the Mappings

Use the following information to update the mappings:

> **NOTE**
>
> **Limitation:** You can update only one possible value at a time.
>
> **Best Practice:** Edit the body in a text editor and use that body in the PUT call.

- **Resource URI:** http://<doi_admin_ui_url>/oi/v3/api/notifications/nimMapping
- **Method:** PUT
- **HTTP Headers:**
  - **Content-Type:** application/JSON
  - **Authorization:** User token that can be generated on the Token Management page.
- **Sample Request:**

```
PUT : http://doi-adminui.67029057.10.005.2.12.nip.io/oi/v3/api/notifications/nimMapping
```

- **Sample Body:** For example, add **"Cancelled=8"** to the **possibleValues** section.

```
{
   "name": "status",
   "mapped": "state",
   "description": "Status of the Incident",
   "dataType": "string",
   "default": "",
   "custom": false,
   "possibleValues": [
      "New=1",
      "Active=2",
      "Awaiting Problem=3",
      "Awaiting User Info=4",
      "Awaiting Evidence=5",
      "Resolved=6",
      "Closed=7"
      "Cancelled=8"
            ],
}
```

Update Policy by Linking Channel and Message Template

Using this API, you can update the policy by linking a channel and message template.

Name	Description
Method	PUT
URL	`https://axa.dxi-na1.saas.broadcom.com/ess/notify/v1/filters` `http or https://<dxi-adminui.route>/ess/notify/v1/filters`
Header	`Bearer <BASE64(JSON(AuthZRequestHdrBean))>`

Name	Description

Sample Request	• **Service Alarm**
	{
	"orgId":"DXI-APMQA",
	"id":15113,
	"keyLevel":1,
	"name":"Service_Alarm_Policy_Channel",
	"channelNames":[
	"Test_Email_Channel"
	],
	"description":"description",
	"filterExpression":{
	"not":false,
	"op":"AND",
	"serviceAlarmAction":true,
	"alarmCategory":"service",
	"filterExpressions":[
	{
	"not":false,
	"op":"OR",
	"serviceAlarmAction":true,
	"alarmCategory":"service",
	"filterCriteria":[
	{
	"attr":"alarmType",
	"attrDesc":"Alarm Type",
	"op":"EQ",
	"filterValue":"Service",
	"not":false
	}
	]
	},
	{
	"not":false,
	"op":"OR",
	"serviceAlarmAction":true,
	"alarmCategory":"service",
	"filterCriteria":[
	{
	"attr":"state",
	"attrDesc":"Alarm Status",
	"op":"EQ",
	"filterValue":"Active",
	"not":false
	}
	]
	}
	]
	},
	"jobInfo":{
	"configId":"",
	"jobsPayload":[
	""
	]

	}, "linkedEntities":[{ "channelName":"Test_Email_Channel", "messageTemplateName":"ServiceAlarmNotificationTemplate"

Name	Description

Sample Response	`{`
	`"newContentVersion" : 1`
	`}`

Global Maintenance APIs

To perform maintenance activities on entities (devices, CIs, groups or services) you must regularly schedule maintenance mode using APIs. During this time, alarms are silenced and you will not receive any alarm notifications. However, you can still monitor the status and health of these devices. An alarm that raises within the maintenance time is tagged in the Alarm Analytics page. You can also view the alarms that are in maintenance mode by enabling the **Show Maintenance alarms** option in all **Alarms View.**

> **NOTE**
> The alarms are **Active** only when they are updated after the maintenance time window. Use the following APIs for the global maintenance window.

Device Lookup

The device lookup API provides a way to retrieve information about a device based on the search criteria. You can use this device information to create a maintenance schedule.

Name	Description
Resource URI	`https://doi.dxi-na1.saas.broadcom.com/oi/v2/api/maintenance/inventory/_search` `http or https://<doi.route>/oi/v2/api/maintenance/inventory/_search`
Method	POST
HTTP Headers	`Content-Type:application/json`
Authorization	Bearer {token}

Name	Description

Payload	`{`
	`"searchText":"<wildcard search>",`
	`"size":"<number of devices to lookup>",`
	`"source":"<source product>"`
	`}`
	Parameters:
	• **searchText:** Allows you to perform wild-card search as follows:
	— **UIM:** host,ip,origin,deviceType
	— **capm:** host,ip,type,product
	— **spectrum:** host, ip, modelName, modeltypeName, modelHandle
	— **custom:** host, ip, product
	• **size:** Number of devices to lookup (maximum size: 1000, and minimum size: 1)
	• **source:** Values can be comma-separated, to query across different source products. Possible values:
	• UIM
	• SPECTRUM
	• CAPM
	• CUSTOM
	Response:
	`{`
	`"inventory": [`
	`{"product": "<source product>",`
	`"ip": "<ip address>",`
	`"name": "<hostname>",`
	`"host": "<hostname>",`
	`"id": "",`
	`"ci_unique_id": "<ci unique id>",`
	`"source": "<source product>",`
	`"type": "DEVICE"`
	`}],`
	`"count": <devices count>`
	`}`

Sample Payload	`{`
	`"searchText":"",`
	`"size":"1",`
	`"source":"CUSTOM,UIM"`
	`}`

Name	Description

Sample Response	{
	"inventory": [
	{
	"product": "CA UIM",
	"ip": "<ip_adddress>",
	"name": "<name>",
	"host": "<host>",
	"id": "",
	"ci_unique_id": "cc3099eda1709d71d45a51e046ac7520",
	"source": "CA UIM",
	"type": "DEVICE"
	}
	]
	},

Service Lookup

The Service Lookup API provides a way to retrieve information about a service based on the search criteria. You can use this service information to create a maintenance schedule. If the API call is returned as false, then the device lookup API is invoked. For more information, see the Device Lookup API.

Name	Description
Resource URI	`https://doi.dxi-na1.saas.broadcom.com/oi/v2/api/maintenance/inventory/_search?services=true` `http or https://<doi.route>/oi/v2/api/maintenance/inventory/_search?services=true`
Method	POST
HTTP Headers	`Content-Type:application/json`
Authorization	Bearer {token}

Payload	`{`
	`"searchText":"<wildcard search>",`
	`"size":"<number of services to lookup>",`
	`}`
	searchText: Allows you to perform a wild card search on name, state, and type.
	Required: True
	size: Number of services to lookup (maximum size: 1000, and minimum size: 1)
	Required: False

Name	Description

Response	`{`
	`  "inventory": [`
	`  {`
	`  "name": "<hostname>",`
	`  "id": "",`
	`  "state": "ACTIVE",`
	`  "type": "SERVICE"`
	`  },`
	`  {`
	`  "name": "<hostname>",`
	`  "id": "",`
	`  "state": "ACTIVE",`
	`  "type": "SERVICE"`
	`  }`
	`  ],`
	`  "count": <services count>`
	`}`

Sample Payload	`{`
	` "searchText":"",`
	` "size":"2"`
	`}`

Sample Response	`{`
	`  "inventory": [`
	`  {`
	`  "name": "Automation_App",`
	`  "id": "Automation_App",`
	`  "state": "ACTIVE",`
	`  "type": "saService"`
	`  },`
	`  {`
	`  "name": "Automation_Boston_All",`
	`  "id": "Automation_Boston_All",`
	`  "state": "ACTIVE",`
	`  "type": "saService"`
	`  }`
	`  ],`
	`  "count": 2`
	`}`

Create a Schedule

This API is used to create a maintenance schedule. You can create a schedule once, daily, weekly, monthly, and yearly.

NOTE
You cannot create a recurring maintenance schedule to occur on the same day within the same schedule. For example, if you have created a maintenance schedule for a service to run from 2 PM to 3 PM, you cannot create within the same schedule a service to run from 5 PM to 6 PM. You need to create a new maintenance schedule for such same day requirements.

Resource URI

```
https://doi.dxi-na1.saas.broadcom.com/oi/v2/api/maintenance
```

```
http or https://<doi.route>/oi/v2/api/maintenance
```

Method: POST

HTTP Headers

```
Content-Type:application/json
```

Authorization

```
Bearer {bearerToken/userToken}
```

Payload Syntax

Use the following payload for this API:

```
{"name":"name of the maintenance schedule",
 "schedule":
    {
      "startTime":"<schedule start time>",
      "duration": "<duration of maintenance time>",
      "recurrencePattern": "<pattern to be used for schedule>",
      "recurrencePeriod": "<repeat of schedule>",
      "recurrenceDaysOfTheWeek": "<weekdays to run schedule>",
      "recurrenceInstance": "<day to run schedule>",
      "recurrenceDayOfTheMonth": "<month to run schedule>",
      "endTime":"<end time for the maintenance schedule>",
      "timeZone":"<timezone for the schedule>"
    },
    "reason":"<Reason for maintenance>",
    "description":"<description of maintenance>",
    "members":[
      {
        "id":"<member id>",
        "name":"<Device/agent/interface/service/group name>",
        "type":"<ci type>",
        "parent":"",
        "ciUniqueId":"<Configurationitem ID>",
        "source":"<source product>"
      },
  ]
"excludedMembers": [
   {
      "id":"<excluded member ID>",
      "type":"<ci type>",
      "name":"<excluded member name>"

   }
   {
      "id":"<excluded member ID>",
      "type":"<ci type>",
      "name":"<excluded member name>"
```

```
      }
    ]

  }
```

Parameters:

The following parameters are mandatory:

Name	Description	Required
name	Specify the name of the maintenance schedule. The name must be unique for each tenant.	True
reason	Specify the maintenance reason.	False
description	Specify the maintenance description.	False

Schedule Parameters: Specify the schedule parameters.

Required: False

Parameters	Description	Required	Notes
startTime	Specify the start time for the maintenance schedule.	True	Start or trigger time of the schedule. Input: "YYYY-MM-DD HH:MM:SS" format. (24 hours)
duration	Specify the duration period of the maintenance schedule.	True	Value in minutes.
endTime	Specify the end time for the maintenance schedule.	True	The end time of the schedule. If you want to run the schedule forever, do not enter any value. Input: "YYYY-MM-DD HH:MM:SS" format. (24 hours)
timeZone	Specify the time zone for the schedule in the format shown in these examples: • America/Chicago • America/Indiana/Knox • Asia/Kolkata • Atlantic/Reykjavik • Europe/Lisbon • Europe/Zurich • Pacific/Honolulu	True	
recurrencePattern	Specify the pattern to be used for the schedule.	True	**Possible values:** (1,2,3,4,5) • **1** - Only once schedule • **2** - Daily schedule • **3** - Weekly schedule • **4** - Monthly schedule • **5** - Yearly schedule

Parameters	Description	Required	Notes
recurrencePeriod	Specify when to repeat the schedule again.	**recurrencePattern 2,3,4**	• **recurrencePattern = 2 (Daily).** Repeat after every n days. The upper limit can be 365 days. Possible values: 1 - 365 • **recurrencePattern = 3 (Weekly).** Repeat after every n weeks. The upper limit can be 52. Possible values: 1 - 52 • **recurrencePattern = 4 (Monthly).** Repeat after every n months. The upper limit can be 12. Possible values: 1 - 12
recurrenceDaysOfTheWeek	Specify the days of the week to run the schedule.	**recurrencePattern 3 (Weekly)** **recurrencePattern 4 (Monthly)**	**Possible values: 1 - 7 (**Sunday to Saturday). Where 1=Sunday, 2=Monday, and 7=Saturday. For example, • — For **recurrencePattern 3 (Weekly),** you can provide comma-separated values, such as 1,2,3. If you want to run the job on Sunday, Monday, and Tuesday. — For **recurrencePattern 4 (Monthly),** you can provide only one value.
recurrenceDayOfTheMonth	Specify the day of the month to run the schedule.	**recurrencePattern 4 (Monthly)**	**Possible values:** 1 - 31
recurrenceInstance	Specify the day of the month to run the schedule on.	**recurrencePattern 4 (Monthly)**	For example, the third Tuesday of a month. **Possible values:** (1,2,3,4,5) corresponding to (first, second, third, fourth, last) The day goes in the field **recurrenceDaysOfTheWeek**

Members Parameters: Specify the member-related information.

Required: False

Parameters	Description	Required	Notes
id	Specify the member id.	True	Start or triggered time of the schedule. Input: "YYYY-MM-DD HH:MM:SS" format. (24 hr)
name	Provide the values for the following parameters: • Device agent name • interface name • service-name • group name	True	
type	Specify the CI type.	True	Following are the types: • Device • Group • Service • Agent (can contain a list of applications/ device) • Interface • Application

Parameters	Description	Required	Notes
ciUniqueId	Specify the Configuration item ID.		
Source	Specify the source product.	True	**Possible values:** • APM • UIM • SPECTRUM • CAPM • CUSTOM • NFA • ADA

excludedMembers Parameters: Specify the excluded member's information.

Required: False

Parameters	Description	Required	Notes
id	Specify the excluded member id.	True	For Example: "CA"
type	Specify the CI type.	True	Following are the CI types: • Device • Group • Service • Agent (can contain a list of applications/ devices) • Interface • Application Example: "SERVICE"
name	Specify the excluded member name.	True	Example: "CA" or "OI"

Examples to Create Schedules

The following section describes examples to create schedules once, daily, weekly, monthly, and yearly:

Sample URI:

```
https://doi.dxi-na1.saas.broadcom.com/oi/v2/api/maintenance

http or https://<doi.route>/oi/v2/api/maintenance
```

Once

Create a schedule to run only once on September 30, 2019, from 6 PM to 8 PM for UIM device (host = <host> and cs_id = 1).

```
{
  "name":"Sample 2023-02-08 IST @ Once Only",
  "schedule":{
  "startTime":"2023-02-08 15:27:00.0",
  "duration":60,
  "recurrencePattern":1,
  "endTime":"2023-02-08 16:27:00.0",
  "timeZone":"Asia/Kolkata" },
  "reason":"Thanks Giving Day",
  "description":"Gifts",
```

```
  "members":
  [
    {
       "id":"1",
       "name":"<host>",
       "type":"DEVICE",
       "source":"UIM"
     }
 ]
```

Response:

```
Successfully Created Maintenance Schedule : 32daccb0-a718-4878-8441-af91e9f372ec
```

Daily (Forever)

Create a schedule to run Daily for infinity/forever, starting from September 30, 2019, from 6 PM to 8 PM for UIM device (host = <host> and cs_id = 1).

```
{
  "name":"Sample 2019-09-30 IST @ Daily",
  "schedule":
  {
   "startTime":"2019-09-30 18:00:00",
   "duration":60,
   "recurrencePattern":2,
   "recurrencePeriod":1
   "timeZone":"Asia/Kolkata"
  },
    "reason":"Thanks Giving Day",
    "description":"Gifts","members":
    [
      {
        "id":"1",
        "name":"<host>",
        "type":"DEVICE",
        "source":"UIM"
      }
   ]
}
```

Response:

```
Successfully Created Maintenance Schedule : 32daccb0-a718-4878-8441-af91e9f372ec
```

Daily (Forever)

Create a schedule to run Daily (Based on Start and End Time), starting from September 30, 2019, from 6 PM to 8 PM and ending on October 30, 2019, for UIM device (host = <host>and cs_id = 1).

```
{
  "name":"Sample 2019-09-30 IST @ Daily",
  "schedule":
   {
     "startTime":"2019-09-30 18:00:00",
     "duration":60,
```

```
      "recurrencePattern":2,
      "recurrencePeriod":1
      "timeZone":"Asia/Kolkata"
      "endTime": "2020-10-30 20:31:00"
     },
      "reason":"Thanks Giving Day",
     "description":"Gifts",
     "members":
      [
       {
        "id":"1",
        "name":"<host>",
        "type":"DEVICE",
        "source":"UIM"
       }
    ]
}
```

Response:

Note: To repeat schedule every alternate day, use **recurrencePeriod**": 2.

```
Successfully Created Maintenance Schedule : 32daccb0-a718-4878-8441-af91e9f372ec
```

Weekly

Create a schedule to run weekly every Sunday, Monday, and Tuesday starting from September 30, 2019, from 6 PM to 8 PM for UIM device (host = <host> and cs_id = 1).

```
{
"name":"Weekly Maintenance Schedule",
"schedule":
{

"startTime":"2019-09-30 18:00:00",
"duration":120,
 "recurrencePattern":3,
"recurrenceDaysOfTheWeek": "1,2,3",
"timeZone":"Asia/Kolkata"
},
"reason":"Thanks Giving Day",
"description":"Gifts",
"members":
[
 {
"id":"1",
"name":"<host>",
 "type":"DEVICE",
 "source":"UIM"
  }
  ]
}
```

Note: To repeat schedule every alternate day, use **recurrencePeriod**: 2.

Response:

```
Successfully Created Maintenance Schedule : 32daccb0-a718-4878-8441-af91e9f372ec
```

Monthly (Every Third Day of the Month)

Create a schedule to run monthly every third day of the month starting from September 30, 2019, from 6 PM to 8 PM for UIM device (host = <host> and cs_id = 1).

```
{
"name":"Monthly Maintenance Schedule",
"schedule":
{
  "startTime":"2019-09-30 18:00:00",
  "duration":120,     "recurrencePattern":4,
  "recurrenceDaysOfTheMonth": "3",
  "timeZone":"Asia/Kolkata"
},
"reason":"Thanks Giving Day",
"description":
"Gifts","members":
[
{
   "id":"1",
   "name":"<host>",
   "type":"DEVICE",
  "source":"UIM"
}
 ]
}
```

Response:

```
Successfully Created Maintenance Schedule : 32daccb0-a718-4878-8441-af91e9f372ec
```

Monthly (Forever on Fourth Tuesday)

Create a schedule to run monthly every fourth Tuesday of the month starting from September 30, 2019, from 6 PM to 8 PM for UIM device (host = <host> and cs_id = 1).

```
{
"name":"Monthly Maintenance Schedule",
"schedule":
{
"startTime":"2019-09-30 18:00:00",
"duration":120,
"recurrencePattern":4,
"recurrenceDaysOfTheMonth": "3",
 "timeZone":"Asia/Kolkata"
},
"reason":"Thanks Giving Day",
"description":"Gifts",
"members":
[
 {
  "id":"1",
  "name":"<host>",
```

```
 "type":"DEVICE",
 "source":"UIM"
 }
 ]
}
```

Response:

```
Successfully Created Maintenance Schedule : 32daccb0-a718-4878-8441-af91e9f372ec
```

Yearly

Create a schedule to run every year on Christmas starting from 2 PM to 10 PM for UIM device (host = <host> and cs_id = 1).

```
{
"name":"Yearly Maintenance Schedule",
"schedule":
{
"startTime":"2019-12-25 14:00:00",
 "duration":480,
"recurrencePattern":5,
"timeZone":"Asia/Kolkata"},
"reason":"Christmas",
"description":"Gifts",
"members":
[
{
 "id":"1",
  "name":"<host>",
 "type":"DEVICE",
 "source":"UIM"
 }
 ]
}
```

Response:

```
Successfully Created Maintenance Schedule : 32daccb0-a718-4878-8441-af91e9f372ec
```

Retrieve Schedules for a Tenant

This API retrieves all schedules for a tenant based on the status type. The schedule status types are as follows:

- Completed
- Active
- Scheduled
- Deleted
- All: Use this type to get all schedules that are not deleted.

NOTE

- If you do not provide any status type parameter, by default, the status type is considered as SCHEDULED.
- If you do not provide any value for **days**, the default value is set as 15 days. The minimum value is 15 days, and the maximum value is 180 days.

Name	Description
Resource UR	`https://doi.dxi-na1.saas.broadcom.com/oi/v2/api/maintenance?status=<status type>` `http or https://<doi.route>/oi/v2/api/maintenance?status=<status type>`
Method	GET
HTTP Headers	`Content-Type:application/json`
Authorization	`Bearer {token}`

Response	`{`
	`"count": 1,`
	`"schedules": [`
	`{`
	`"scheduleId": "<scheduleid>",`
	`"name": "<name of the schedule>",`
	`"startTime": "<schedule start time>",`
	`"duration": <duration of maintenance time>,`
	`"description": "<description of maintenance>",`
	`"reason": "<reason for maintenance>",`
	`"recurrencePattern": <pattern to be used for schedule>,`
	`"recurrencePeriod": <repeat of schedule>,`
	`"endTime": "<end time for the maintenance schedule>",`
	`"timeZone": "<timezone for the schedule>",`
	`"status": "<status type>"`
	`},`
	`}`

Sample Request	`https://doi.dxi-na1.saas.broadcom.com/oi/v2/api/maintenance?status=scheduled&days=180`
	`http or https://<doi.route>/oi/v2/api/maintenance?status=scheduled&days=180`
	You can retrieve the schedule for **status** or **days** specifically if required, or you can combine status and days as shown in the above example.

Name	Description

Sample Response	{
	"count": 2,
	"schedules": [
	{
	"scheduleId": "5434f314-eaf9-4119-91c1-4e2a89e6927a",
	"name": "test 2019-09-16 22:531:00111",
	"startTime": "2019-09-19 22:53:00.0",
	"duration": 60,
	"description": "Gifts",
	"reason": "Thanks Giving Day",
	"recurrencePattern": 1,
	"recurrencePeriod": 1,
	"endTime": "2019-10-30 20:31:00.0",
	"timeZone": "IST",
	"status": "SCHEDULED"
	},
	{
	"scheduleId": "35b84a73-129e-440c-afbd-a0e71688cde5",
	"name": "Sample 2019-09-30 IST @ Dail11y",
	"startTime": "2019-09-30 22:53:00.0",
	"duration": 60,
	"description": "Gifts",
	"reason": "Thanks Giving Day",
	"recurrencePattern": 2,
	"recurrencePeriod": 1,
	"endTime": "2020-10-30 20:31:00.0",
	"timeZone": "IST",
	"status": "SCHEDULED"
	},
	]
	}

Retrieve Information for a Schedule

This API retrieves information for a schedule by providing a schedule id.

Name	Description
Resource URI	`https://doi.dxi-na1.saas.broadcom.com/oi/v2/api/maintenance/{schedule id}` `http or https://<doi.route>/oi/v2/api/maintenance/{schedule id}`
Method	GET
HTTP Headers	`Content-Type:application/json`
Authorization	`Bearer {token}`

Name	Description

Response	`{`
	`"scheduleId": "<scheduleid>",`
	`"name": "<name of the schedule>",`
	`"startTime": "<schedule start time>",`
	`"duration": <duration of maintenance time>,`
	`"description": "<description of maintenance>",`
	`"reason": "<reason for maintenance>",`
	`"recurrencePattern": <pattern to be used for schedule>,`
	`"recurrencePeriod": <repeat of schedule>,`
	`"endTime": "<end time for the maintenance schedule>",`
	`"timeZone": "<timezone for the schedule>",`
	`"members":[`
	`{`
	`"id": "<member id>",`
	`"type": "<ci type>",`
	`"name": "<Device/agent/interface/service/group name>",`
	`"source": "<source product>"`
	`}`
	`]`
	`"excludedMembers": [`
	`{`
	`"id":"<excluded member ID>",`
	`"type":"<ci type>",`
	`"name":"<excluded member name>"`
	
	`}`
	`{`
	`"id":"<excluded member ID>",`
	`"type":"<ci type>",`
	`"name":"<excluded member name>"`
	
	`}`
	`]`
	`}`
	Note: For the description of the parameters, see the Parameters section in the Create a Schedule section.

Sample Request	`https://doi.dxi-na1.saas.broadcom.com/oi/v2/api/maintenance/{schedule id}` `http or https://<doi.route>/oi/v2/api/maintenance/{schedule id}`

Name	Description

Sample Response	{
	"scheduleId": "a89788fb-a9cc-4a92-9752-fed4d8a48d08",
	"name": "MyAp110111111tel1111-112",
	"startTime": "2020-03-16 20:02:30.0",
	"duration": 60,
	"description": "Gifts",
	"reason": "Thanks Giving Day",
	"recurrencePattern": 1,
	"endTime": "2021-03-31 12:00:00.0",
	"timeZone": "IST",
	"status": "SCHEDULED",
	"members": [
	{
	"id": "UIMS_VMware",
	"type": "SERVICE",
	"name": "UIMS_VMware",
	"parent": "Y",
	"source": "OI"
	}
	],
	"excludedMembers": [
	{
	"id":"CA",
	"type":"SERVICE",
	"name":"CA"
	},
	{
	"id":"OI",
	"type":"SERVICE",
	"name":"OI"
	}
	]
	}

Retrieve Information for Bulk Schedules

This API retrieves information for the specified schedules.

Name	Description
Resource URI	https://doi.dxi-na1.saas.broadcom.com/oi/v2/api/maintenance/get http or https://<doi.route>/oi/v2/api/maintenance/get
Method	POST
HTTP Headers	Content-Type:application/json
Authorization	Bearer {token}
Payload	{**"scheduleIds"**:["<Schedule id1>", "<Schedule id2>"]}

Name	Description

Response	[
	{
	"scheduleId": "<scheduleid>",
	 "name": "<name of the schedule>",
	 "startTime": "<schedule start time>",
	 "duration": <duration of maintenance time>,
	 "description": "<description of maintenance>",
	 "reason": "<reason for maintenance>",
	 "recurrencePattern": <pattern to be used for schedule>,
	 "recurrencePeriod": <repeat of schedule>,
	"timeZone": "<timezone for the schedule>",
	 "members": [
	  {
	  "id": "<member id>",
	  "type": "<ci type>",
	  "name": "<Device/agent/interface/service/group name>",
	  "source": "<source product>"
	 }
	]
	  }

Sample URI	https://doi.dxi-na1.saas.broadcom.com/oi/v2/api/maintenance/get http or https://<doi.route>/oi/v2/api/maintenance/get
Sample Payload	{"**scheduleIds**": ["2000025a-ef77-4977-a559-a03fb3559fc3","bdf5f9e1-5959-4deb-b176-5096097ec634"]}

Name	Description

Sample Response	[
	{
	"scheduleId": "2000025a-ef77-4977-a559-a03fb3559fc3"
	"name": "Sample 2019-09-30 IST @ Once Only",
	"startTime": "2019-10-16 14:50:00.0",
	"duration": 60,
	"description": "Gifts",
	"reason": "Thanks Giving Day",
	"recurrencePattern": 1,
	"timeZone": "IST",
	"members": [
	{
	"id": "1",
	"type": "DEVICE",
	"name": "d1",
	"source": "UIM"
	}
	]
	},
	{
	"scheduleId": "bdf5f9e1-5959-4deb-b176-5096097ec634",
	"name": "Sample IST @ Once Only",
	"startTime": "2019-10-16 14:55:00.0",
	"duration": 60,
	"description": "Gifts",
	"reason": "Thanks Giving Day",
	"recurrencePattern": 1,
	"timeZone": "IST",
	"members": [
	{
	"id": "1",
	"type": "DEVICE",
	"name": "d1",
	"source": "UIM"
	}
	]
	}
	]

Delete Schedules

You can delete a particular schedule or all schedules. Use the following APIs to delete a schedule or all schedules.

Delete a Schedule

This API deletes a particular schedule.

> **NOTE**
> You cannot delete an active schedule by default. To delete an active schedule, use the query param **forceStop=true**.

Syntax:

```
https://doi.dxi-na1.saas.broadcom.com/oi/v2/api/maintenance/{schedule_id}?forceStop=true

http or https://<doi.route>/oi/v2/api/maintenance/{schedule_id}?forceStop=true
```

Name	Description
Resource URI	`https://doi.dxi-na1.saas.broadcom.com/oi/v2/api/maintenance/{schedule_id}` `http or https://<doi.route>/oi/v2/api/maintenance/{schedule_id}`
Method	DELETE
HTTP Headers	`Content-Type:application/json`
Authorization	`Bearer {token}`
Response	`Successfully deleted Maintenance Schedule with scheduleId: <schedule id>`
Sample Request	`https://doi.dxi-na1.saas.broadcom.com/oi/v2/api/maintenance/2daccb0-a718-4878-8441-af91e9f372eb` `http or https://<doi.route>/oi/v2/api/maintenance/2daccb0-a718-4878-8441-af91e9f372eb`
Sample Response	`Successfully deleted Maintenance Schedule with scheduleId: 2daccb0-a718-4878-8441-af91e9f372eb`

Delete Specified Schedules

This API deletes all the specified schedules (bulk delete).

Name	Description
Resource URI	`https://doi.dxi-na1.saas.broadcom.com/oi/v2/api/maintenance/delete` `http or https://<doi.route>/oi/v2/api/maintenance/delete`
Method	POST
HTTP Headers	`Content-Type:application/json`
Authorization	`Bearer {token}`
Payload	`{"scheduleIds":[ "<schedule id1>", "<schedule id2>"]}`

Response	`Success response :-`
	`{`
	`"successCount": <number of success count>,`
	`"failedCount": <number of failed count>,`
	`"failedScheduleIds": []`
	`}`

Sample URI	`https://doi.dxi-na1.saas.broadcom.com/oi/v2/api/maintenance/delete` `http or https://<doi.route>/oi/v2/api/maintenance/delete`

Name	Description
Sample Payload	{"**scheduleIds**": ["2000025a-ef77-4977-a559-a03fb3559fc3", "bdf5f9e1-5959-4deb-b176-5096097ec634"] }

Sample Response	Success response :-
	{
	"successCount": 2,
	"failedCount": 0,
	"failedScheduleIds": []
	}

Edit a Schedule

This API enables you to edit a schedule by providing the schedule id.

Name	Description
Resource URI	https://doi.dxi-na1.saas.broadcom.com/oi/v2/api/maintenance/{schedule_id} http or https://<doi.route>/oi/v2/api/maintenance/{schedule_id}
Method	PUT
HTTP Headers	Content-Type:application/json
Authorization	Bearer {token}
Request	{ "scheduleName": "<update to schedulename>"}
Response	Successfully Updated Maintenance Schedule : {schedule id}
Sample URI	https://doi.dxi-na1.saas.broadcom.com/oi/v2/api/maintenance/32daccb0-a718-4878-8441-af91e9f372eb http or https://<doi.route>/oi/v2/api/maintenance/32daccb0-a718-4878-8441-af91e9f372eb
Sample Request	{ "scheduleName": "sample 2019-09-30 IST @ 1 @ Yearly11111"}
Sample Response	Successfully Updated Maintenance Schedule : 1c05d0db-3ad3-4c3a-98cd-51b108643cb4

Stop a Schedule

This API stops the specified schedule.

Name	Description
Resource URI	`https://doi.dxi-na1.saas.broadcom.com/oi/v2/api/maintenance/{schedule_id}/stop` `http or https://<doi.route>/oi/v2/api/maintenance/{schedule_id}/stop`
Method	PUT
HTTP Headers	`Content-Type:application/json`
Authorization	`Bearer {token}`
Request	`https://doi.dxi-na1.saas.broadcom.com/oi/v2/api/maintenance/{schedule_id}/stop` `http or https://<doi.route>/oi/v2/api/maintenance/{schedule_id}/stop`
Response	`For inactive schedule: Maintenance Schedule with schedule id: {schedule id} is currently not active. For active schedule: Successfully stopped Maintenance Schedule with scheduleId: {schedule id}`
Sample Request	`https://doi.dxi-na1.saas.broadcom.com/oi/v2/api/maintenance/32daccb0-a718-4878-8441-af91e9f372eb/stop` `http or https://<doi.route>/oi/v2/api/maintenance/32daccb0-a718-4878-8441-af91e9f372eb/stop`
Sample Response	`For inactive schedule: Maintenance Schedule with schedule id: 32cbd160-6ddf-434c-835b-98b96ac7ef9f is currently not active. For active schedule: Successfully stopped Maintenance Schedule with scheduleId: 32cbd160-6ddf-434c-835b-98b96ac7ef9f`

Retrieve the Maintenance Windows for the Specified Schedules

You can use this API to retrieve the list of maintenance windows for the provided schedules. You can only retrieve the list of active schedules, that is, you cannot see the list of completed or deleted schedules.

NOTE

- If you do not provide any value for **days**, the default value is set as 15 days. The minimum value is 15 days, and the maximum value is 180 days.
- The Time zone specified here refers to the Client's Time zone

Name	Description
Resource URI	`https://doi.dxi-na1.saas.broadcom.com/oi/v2/api/maintenance/windows` `http or https://<doi.route>/oi/v2/api/maintenance/windows`
Method	POST
HTTP Headers	`Content-Type:application/json`
Authorization	`Bearer {token}`

Name	Description
Sample Request 1	`{"scheduleIds":["78404599-5901-4f4d-b680-923bb24784e6"],"days":"5","timeZone":"IST"}` The startTime specified refers to the duration or maintenance window of the schedules.

Sample Response 1	`{`
	`  "errors": [],`
	`  "windows": [`
	`  {`
	`  "scheduleId": "78404599-5901-4f4d-b680-923bb24784e6",`
	`  "startTime": [`
	`  "2019-12-24 22:30:00.0",`
	`  "2019-12-25 22:30:00.0",`
	`  "2019-12-26 22:30:00.0",`
	`  "2019-12-27 22:30:00.0"`
	`  ]`
	`  }`
	`  ]`
	`}`

Sample Request 2	`{"scheduleIds":["78404599-5901-4f4d-b680-923bb24784e7"],"days":"20","timeZone":"IST"}` The startTime specified refers to the duration or maintenance window of the schedules.

Name	Description

Sample Response 2	{
	"errors": [],
	"windows": [
	{
	"scheduleId": "78404599-5901-4f4d-b680-923bb24784e7",
	"startTime": [
	"2019-12-24 22:30:00.0",
	"2019-12-25 22:30:00.0",
	"2019-12-26 22:30:00.0",
	"2019-12-27 22:30:00.0",
	"2019-12-24 22:30:00.0",
	"2019-12-25 22:30:00.0",
	"2019-12-26 22:30:00.0",
	"2019-12-27 22:30:00.0",
	"2019-12-24 22:30:00.0",
	"2019-12-25 22:30:00.0",
	"2019-12-26 22:30:00.0",
	"2019-12-27 22:30:00.0",
	"2019-12-24 22:30:00.0",
	"2019-12-25 22:30:00.0",
	"2019-12-26 22:30:00.0",
	"2019-12-27 22:30:00.0",
	"2019-12-24 22:30:00.0",
	"2019-12-25 22:30:00.0",
	"2019-12-26 22:30:00.0",
	"2019-12-27 22:30:00.0"
	]
	}
	]
	}

Sample Request 3	{"scheduleIds":["b834d16b-6ca9-4721-8342-b82bd2d20f00"],"days":"180", "timeZone": "IST"}

Name	Description
Sample Response 3	`{ "member": [ { "id": "2569", "type": "DEVICE", "name": "ILFCDEMOUNFIT", "source": "UIM" } ], “excludedMembers”: [ { "id":"CA", "type":"SERVICE", "name":"CA" } ] "fromTime": 1577553182000, "toTime": 1581663284000 }`

Retrieve Maintenance Window Schedules for a Service

You can use this API to get the list of all the maintenance window schedules for a particular service in the last one year.

Name	Description
Resource URI	`https://doi.dxi-na1.saas.broadcom.com/oi/v2/api/maintenance/service-schedules-history` `http or https://<doi.route>/oi/v2/api/maintenance/service-schedules-history`
Method	POST
HTTP Headers	`Content-Type:application/json`

Name	Description

Sample Request	{
	"member": [
	{
	"id": "Automation_App",
	"name": "Automation_App",
	"type": "SERVICE",
	"source": "OI"
	},
	{
	"id": "Automation_Infra",
	"name": "Automation_Infra",
	"type": "SERVICE",
	"source": "OI"
	},
	{
	"id": "AutomationGalaxy",
	"name": "AutomationGalaxy",
	"type": "SERVICE",
	"source": "OI"
	}
	],
	"fromTime": 1651343400000,
	"toTime": 1681803432059,
	"status": "ALL",
	"pageNumber": 1,
	"pageSize": 999
	}
	The end time should not be greater than the current time.

Sample Response	[
	{
	"serviceName": "Automation_Infra",
	"windows": [
	{
	"scheduleName": "Test Demo Infr serv MW 2",
	"restrictSLOOnMaintenance": false,
	"timeZone": "Asia/Kolkata",
	"status": "SCHEDULED",
	"windowStartTime": 1681730100109,
	"windowEndTime": 1681731900086,
	"startTime": "2023-04-17T11:15:00.109+0000",
	"endTime": "2023-04-17T11:45:00.086+0000"
	}
	]
	}
	]

Retrieve Schedules for a Configuration Item (CI)

You can use this API to retrieve the list of schedules for the following Configuration Item(s):

- Agent
- Application
- Service
- Device
- Group
- Interface

Ensure that you specify the number of **days** and the **time zone** for the maintenance window. For more information on the member inputs, see the Create a Schedule section.

NOTE

- If you do not provide any value for **days**, the default value is set as 15 days. The minimum value is 15 days, and the maximum value is 180 days.
- The Time zone specified here refers to the Time zone in which the schedule was created.

Name	Description
Resource URI	`https://doi.dxi-na1.saas.broadcom.com/oi/v2/api/maintenance/getSchedulesForCI` `http or https://<doi.route>/oi/v2/api/maintenance/getSchedulesForCI`
Method	POST
HTTP Headers	`Content-Type:application/json`
Authorization	`Bearer {token}`

Sample Request 1

The sample request for the **types** of **CIs** are as follows:

```
{
   "members": [
        {
            "id": "NYORK",
            "type": "SERVICE",
            "name": "NYORK",
            "source": "UIM"
        },
            {
            "id": "agenttomcat",
            "type": "AGENT",
            "name": "agenttomcat",
            "source": "APM"
        },
        {
            "id": "agentnotepad",
            "type": "APPLICATION",
            "name": "notepad",
            "source": "APM"
        },
        {
            "id": "311",
            "type": "DEVICE",
            "name": "vm111029-uimmain",
```

```
            "source": "UIM"
        }
    ]
}
```

NOTE
The *endTime* refers to the end of the maintenance schedule for the given Configuration Item(s).

Sample Response


```
[
    {
        "memberId": "NYORK",
        "schedules": [
            {
                "scheduleId": "12ac21a-f0c2-4294-8121-1eae36dc78182",
                "name": "20200102 - Schedule114",
                "startTime": "2020-01-17T01:40:00.000+0000",
                "endTime": "2020-01-20T09:00:00.000+0000",
                "timeZone": "IST"
            }
        ],
        "memberId": "agenttomcat",
        "schedules": [
            {
                "scheduleId": "2271adba-f212-4111-8c71-1eae36dc619a",
                "name": "20200102 - Schedule124",
                "startTime": "2020-01-10T15:40:00.000+0000",
                "endTime": "2020-03-12T09:00:00.000+0000",
                "timeZone": "IST"
            }
        ],
        "memberId": "agentnotepad",
        "schedules": [
            {
                "scheduleId": "45acadba-f022-4114-8231-1eae36df1189",
                "name": "20200102 - Schedule234",
                "startTime": "2020-01-08T15:40:00.000+0000",
                "endTime": "2020-02-09T09:00:00.000+0000",
                "timeZone": "IST"
            }
        ],
        "memberId": "311",
        "schedules": [
            {
                "scheduleId": "62acadba-f0c2-4294-8c71-1eae36dc91a4",
                "name": "20200102 - Schedule104",
                "startTime": "2020-01-07T15:40:00.000+0000",
                "endTime": "2020-01-10T09:00:00.000+0000",
                "timeZone": "IST"
            }
        ]

    }
]
```

Sample Request 2

The sample request for **multiple CIs** are as follows:

```
{
   "members": [
{
            "id": "1022",
            "type": "DEVICE",
            "name": "bwka-uim3",
            "source": "UIM"
        },
{
            "id": "9291",
            "type": "DEVICE",
            "name": "iksd-2kwin",
            "source": "UIM"
        },
{
            "id": "4101",
            "type": "DEVICE",
            "name": "knmn-773z",
            "source": "UIM"
        },
{
            "id": "3989",
            "type": "DEVICE",
            "name": "lkh-dns2020",
            "source": "UIM"
        },
        {
            "id": "2115",
            "type": "DEVICE",
            "name": "hb299551iis",
            "source": "UIM"
        }
    ]
}
```

NOTE
The *endTime* refers to the end of the maintenance schedule for the given Configuration Item(s).

Sample Response

```
[
    {
        "memberId": "1022",
        "schedules": [
            {
                "scheduleId": "12ac21a-f0c2-4294-8121-1eae36dc78182",
                "name": "20200102 - Schedule115",
                "startTime": "2020-01-18T01:40:00.000+0000",
                "endTime": "2020-01-20T09:00:00.000+0000",
                "timeZone": "IST"
            }
        ],
```

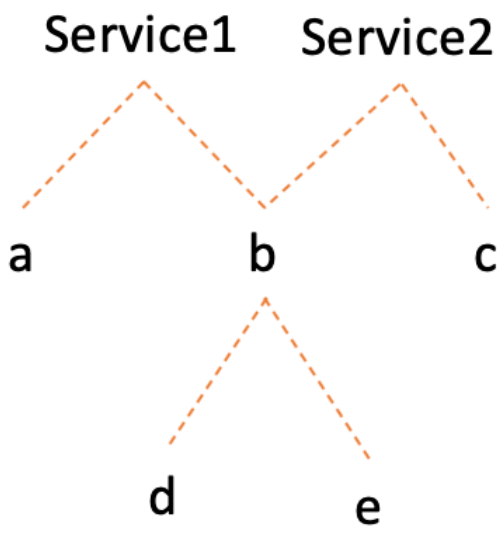
```
        "memberId": "9291",
        "schedules": [
            {
                "scheduleId": "2271adba-f212-4111-8c71-1eae36dc619a",
                "name": "20200102 - Schedule125",
                "startTime": "2020-01-15T15:40:00.000+0000",
                "endTime": "2020-03-17T09:00:00.000+0000",
                "timeZone": "IST"
            }
        ],
        "memberId": "4101",
        "schedules": [
            {
                "scheduleId": "45acadba-f022-4114-8231-1eae36df1189",
                "name": "20200102 - Schedule235",
                "startTime": "2020-01-11T15:40:00.000+0000",
                "endTime": "2020-02-13T09:00:00.000+0000",
                "timeZone": "IST"
            }
        ],
        "memberId": "3989",
        "schedules": [
            {
                "scheduleId": "62acadba-f0c2-4294-8c71-1eae36dc91a4",
                "name": "40200102 - Schedule105",
                "startTime": "2020-01-09T15:40:00.000+0000",
                "endTime": "2020-01-10T09:00:00.000+0000",
                "timeZone": "IST"
            }
        ],
"memberId": "2115",
        "schedules": [
            {
                "scheduleId": "62acadba-f0c2-4294-8c71-1eae36dc91a4",
                "name": "40200102 - Schedule105",
                "startTime": "2020-01-07T15:40:00.000+0000",
                "endTime": "2020-01-10T09:00:00.000+0000",
                "timeZone": "IST"
            }
        ]

    }
]
```


Get Overlap Sub Services

Use this API to retrieve the overlapping sub-services when you create a schedule. For example, when you create a schedule at Service1, it automatically includes its children and based on your selection may include overlapping children from Service2. Service1 has "a" and "b" as child services, "b" has further "d" and "e" as its sub-service. "b" is a child for "Service2" as well. When you create a schedule, you are pointed to the fact that "b", "d", and "e" belong to other services.

The following image illustrates this example.



Name	Description
Resource URI	`https://doi.dxi-na1.saas.broadcom.com/oi/v2/api/maintenance/services/overlap` `http or https://<doi.route>/oi/v2/api/maintenance/services/overlap`
Method	GET
HTTP Header	`Content-Type:application/json`
Sample Payload	`{"services": ["service1"]}`
Sample Response	The response points out that "b", "d", and "e" sub-services are part of other services. `{"b","d","e"}`

Validate Maintenance Schedule Name

This API validates the maintenance schedule based on the schedule name.

Name	Description
Resource URI	`https://doi.dxi-na1.saas.broadcom.com/oi/v2/api/maintenance/validateScheduleName?scheduleName={any name}` `http or https://<doi.route>/oi/v2/api/maintenance/validateScheduleName?scheduleName={any name}`
Method	GET

Name	Description
HTTP Headers	`Content-Type:application/json`
Response	True or False

Usage Data (Telemetry)

Telemetry is a foundational element of the Enterprise Software Portfolio License Agreement (PLA) model. The initial requirement of the Telemetry effort is to collect and report product-specific usage daily in support of the new consumption model. It is mandatory for a customer under Enterprise Software PLA to enable telemetry and share the usage data. This article describes how to enable telemetry and route the usage data to Usage Reporting Portal. For more information, see the Usage Reporting Portal section.

Data Collected By Telemetry

Telemetry collects two types of details for each PLA customer:

- **Customer data:** This data identifies the customer, its site through the site ID, and an optional Charge back ID to identify the division or group to be charged for usage.
- **Usage data:** This is the actual usage data based on the consumption, which is collected and shared. You must enable the upload of the usage data. For more information about the usage data that is collected, see the respective product documentation.

> **NOTE**
> Telemetry does not collect any personally identifiable information (PII) or sensitive information. For additional information about how your information is collected and used, read our privacy statement.

Frequency of Data Collection

By default, telemetry collects and stores the data daily at 12.00 a.m. If the scheduler is not active at 12.00 a.m., the data is collected only in the next day run. The data is collected only once per day.

Data Collected

The following tables display the Telemetry metrics that are calculated in DX Operational Intelligence and sent to the **Usage Reporting Portal**:

Device Count

Metric	Metric Key
APM Monitored Device Count	apm.monitoreddevice.count
UIM Monitored Device Count	uim.monitoreddevice.count
NetOps Monitored Device Count	netops.monitoreddevice.count
VNA Monitored Device Count	vna.monitoreddevice.count
Third-party Monitored Device Count	thirdparty.monitoreddevice.count
Unknown Monitored Source Device Count	unknownsource.monitoreddevice.count

PLA Device Count

Metric	Metric Key
APM Monitored PLA Device Count	apm.monitoreddevice.pladevice.count
UIM Monitored PLA Device Count	uim.monitoreddevice.pladevice.count
NetOps Monitored PLA Device Count	netops.monitoreddevice.pladevice.count

Metric	Metric Key
VNA Monitored PLA Device Count	vna.monitoreddevice.pladevice.count
Third-party Monitored PLA Device Count	thirdparty.monitoreddevice.pladevice.count
Unknown Monitored PLA Source Device Count	unknownsource.monitoreddevice.pladevice.count

How License Metrics are Calculated

The DX Operational Intelligence metrics are calculated for all the tenants in an instance. The metrics from different data sources are calculated for DX Operational Intelligence. The following table shows the metric calculation :

Product	Multiplier for 1 Entity	Mapping
DX APM	0.5	2 device counts are considered as 1 device apm.monitoreddevice.count * 0.5 = apm.monitoreddevice.pladevice.count
DX NETOPS	0.25	4 device counts are considered as 1 device netops.monitoreddevice.count * 0.25 = uim.monitoreddevice.pladevice
UIM	0.5	2 device count are considered as 1 device uim.monitoreddevice.count *0.5 = netops.monitoreddevice.pladevice.count
VNA	0.1	10 vna device counts are considered as 1 device vna.monitoreddevice.count *0.1 = vna.monitoreddevice.pladevice.count
THIRD-PARTY Devices	1.0	1 device count are considered as 1 device thirdparty.monitoreddevice.count *1 = thirdparty.monitoreddevice.pladevice.count
THIRD-PARTY Agents	0.25	4 agent counts are considered as 1 device thirdparty.monitoredagent.count *0.25 = thirdparty.monitoreddevice.pladevice.count
UNKNOWN Devices	1	1 device count is considered as 1 device unknownsource.monitoreddevice.count * 1 = unknownsource.monitoreddevice.pladevice.coun

Documentation Legal Notice

This Documentation, which includes embedded help systems and electronically distributed materials, (hereinafter referred to as the “Documentation”) is for your informational purposes only and is subject to change or withdrawal by Broadcom at any time. This Documentation is proprietary information of Broadcom and may not be copied, transferred, reproduced, disclosed, modified or duplicated, in whole or in part, without the prior written consent of Broadcom.

If you are a licensed user of the software product(s) addressed in the Documentation, you may print or otherwise make available a reasonable number of copies of the Documentation for internal use by you and your employees in connection with that software, provided that all Broadcom copyright notices and legends are affixed to each reproduced copy.

The right to print or otherwise make available copies of the Documentation is limited to the period during which the applicable license for such software remains in full force and effect. Should the license terminate for any reason, it is your responsibility to certify in writing to Broadcom that all copies and partial copies of the Documentation have been returned to Broadcom or destroyed.

TO THE EXTENT PERMITTED BY APPLICABLE LAW, BROADCOM PROVIDES THIS DOCUMENTATION “AS IS” WITHOUT WARRANTY OF ANY KIND, INCLUDING WITHOUT LIMITATION, ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NONINFRINGEMENT. IN NO EVENT WILL BROADCOM BE LIABLE TO YOU OR ANY THIRD PARTY FOR ANY LOSS OR DAMAGE, DIRECT OR INDIRECT, FROM THE USE OF THIS DOCUMENTATION, INCLUDING WITHOUT LIMITATION, LOST PROFITS, LOST INVESTMENT, BUSINESS INTERRUPTION, GOODWILL, OR LOST DATA, EVEN IF BROADCOM IS EXPRESSLY ADVISED IN ADVANCE OF THE POSSIBILITY OF SUCH LOSS OR DAMAGE.

The use of any software product referenced in the Documentation is governed by the applicable license agreement and such license agreement is not modified in any way by the terms of this notice.

The manufacturer of this Documentation is Broadcom Inc.

Provided with “Restricted Rights.” Use, duplication or disclosure by the United States Government is subject to the restrictions set forth in FAR Sections 12.212, 52.227-14, and 52.227-19(c)(1) - (2) and DFARS Section 252.227-7014(b)(3), as applicable, or their successors.

Copyright © 2005–2023 Broadcom. All Rights Reserved. The term “Broadcom” refers to Broadcom Inc. and/or its subsidiaries. All trademarks, trade names, service marks, and logos referenced herein belong to their respective companies.

BROADCOM®
SOFTWARE