



ATOM User & Admin Guide

version 11.8

Table of Contents

Table of Contents	2
Getting Started with ATOM	9
Intended Audience	9
References	9
ATOM Solution Overview	9
General Concepts	12
RBAC	12
Model Driven User Interface	12
Multi Tenancy	12
Viewing the Dashboard	12
Resource Management	13
Device Management	14
Credential Management	15
Credential Sets	15
Credential Profile	18
Credential Maps	19
Device Onboarding	20
Device Views	22
Device Explorer	23
Device Actions	24
Jobs & Subscriptions	24
VTY Sessions	26
Default Jobs	27
Device Summary	27
Configuration Management	28
Configuration Archive	28
Configuration Change Log	31
Configuration Change Management - Create/Update/Delete	32

Device Inventory (SNMP)	33
Adding Unmanaged devices	34
Adding Dummy devices	35
Network Topology	35
Network Connections	35
Network Topology	36
Resource Pools	36
Locations	39
Location Types	40
IPAM	40
IP Address Pool Group	40
IP Address Pools	41
VLAN Groups	43
Configuration Compliance	45
Policies	45
Use Cases	47
Scenario 1: IP Domain Name	50
Configure Policy	50
Configure Rules	51
Action Details	56
Scenario 2: NTP Server configuration check	59
Condition1	60
Condition2	61
Scenario 3: Interface configuration check	64
Scenario 4: Enforce VTY Session Timeouts	68
Scenario 5: Enforce OSPF Router Id as Loopback0	72
Condition1	74
Condition2	78
Scenario 6: BGP TTL Hop-count	80
Condition1	81

Condition2	84
YANG Compliance	85
Policy creation with Xpath Expressions	85
Scenario 7: IP Domain Name	86
Scenario 8: IP Name-server check	88
Scenario 9 : NTP server Check	89
Scenario 10 : Interface Check with rule_variable	91
Scenario 11 : VRF Check with rule_variable	94
How to derive the X-path expressions	97
Policy creation with XML Template Payload	97
Scenario 12 : IP Domain name check	98
Scenario 13 : IP Name Server check	100
Scenario 14 : Interface check	103
Scenario 15 : VRF check	105
How to derive the XML Template payload	109
Report	114
Remediation :	125
Dashboard	132
Appendix	136
Collections in ATOM	139
Jobs	139
Collection Job	141
Configuration Job	141
Diagnostics Job	143
Discovery Job	145
Inventory Job	149
Maintenance Job	149
Device Inventory (SNMP)	152
Monitoring	153
Import SNMP MIBs	153

SNMP	156
Telemetry	159
Configure Telemetry Collection	160
SNMP Traps	164
Syslogs	168
Charts & Templates	171
Chart Creation	174
Create an Advanced Custom Chart	176
Alert, Actions & Closed Loop Automation	183
Define Alert Rules	184
Define Alert Filter	187
Attach Remediation Actions	189
Actions	192
Slack	192
Email	194
Workflow	196
RPC	197
Alerts Summary	198
Alert Dashboard	201
Inventory	202
Software Image Management	202
Image Upgrade Workflow	
Image Repository	
Software Version Compliance	
Network Automation	206
Network Workflow & Low Code Automation	206
Uploading Workflow Package	207
Workflow Lifecycle Management	208
Workflow Variables	213

Network Service Automation	217
Transactional control at the Service level	220
Service Approvals	224
Configuration Drift (Network Services)	226
Service Compliance	230
Resolving Service Violations	231
Agents	233
Administration	233
Tasks & Events	233
Events	234
TraceLogs	234
System	236
Rule Engine	237
Rule	238
Licensing & Entitlements	241
ATOM in Dedicated Mode	241
ATOM in Multi-Tenant or Shared Mode	242
Uploading a License	242
License Summary	242
General Settings	243
URL Management	243
Alert Monitoring	244
Chart-setting	244
Device Management	244
Service now	245
Service Management	245
TSDB	245
SNMP v2 Configurations	246
SMTP Configurations	246
Notification	247

Python Remote Debug	247
Developer Options	247
System Maintenance	247
View Actions	247
Request Sanitization	247
Password Profile	248
Primary container load limit	248
Look and Feel	248
Event Summary	248
Notifications	249
Message Brokers	251
AMQP Listeners	253
System Manager	256
FQDN Agent Settings	260
Plugins and Extensions	265
Packages	265
Package Explorer	265
SNMP	266
Device Support	269
Managing Tenants	269
Overview of Multi Tenancy	269
Root Tenant	270
Top Level Tenants	270
Simple Multi Tenancy	271
Hierarchical Multi Tenancy	271
Sub Tenant	271
System users	272
Tenant Users	273
Concept Of Visibility And Usability	275
Shared With Variations	276

Creating Tenants	279
Onboarding Tenants through Keycloak scripts	279
User Management	286
Roles	287
Creating Authentication Mode Priority	288
Managing Users	291
Assigning Role to the User	300
Creating SNMPv3 users	300
Subscribing to Events	301
Workflow-User-Level-Authorization	302
Configuring NACM	303
Adding User Groups	305
Creating Rule lists	306
Integrating ATOM with Central Authentication Systems	317
Managing Active Directory Users()	317
Managing OpenLDAP Users	319
Managing TACACS Users	321
Customizing the Dashboard using DSL	323
Writing DSL Queries	324
Customizing the Dashboard	326
DSL Assignment	328
Tag Management	330
UI Customizations	330
Troubleshoot	337
Device Comm and Inv	338
Ping	338
SNMP	338
Config Parser	339
File Server	340
About	340

Getting Started with ATOM

Intended Audience

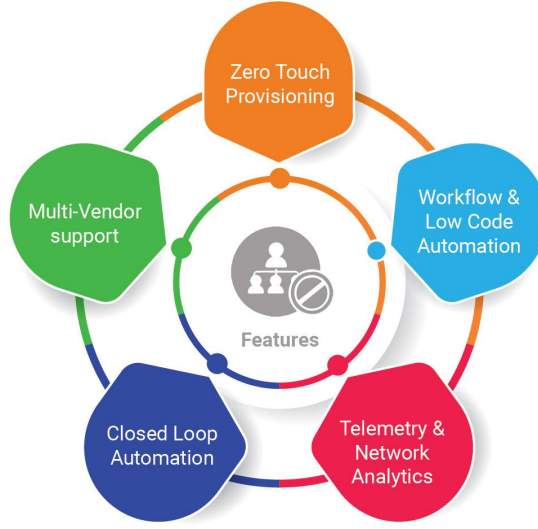
This document is intended for Network Administrators & Operators that are using ATOM to perform Network management, configuration management, services automation and MOPs.

References

1. ATOM Deployment Guide - All aspects of ATOM Deployment including sizing and deployment process
2. ATOM User Guide - Master **[This Document]**
3. ATOM User Guide - Remote Agent Deployment Guide
4. ATOM User Guide - Performance Management & Alerting
5. ATOM User Guide - Network Configuration Compliance, Reporting & Remediation
6. ATOM API Guide - Discusses all external interfaces and integration flows
7. ATOM Platform Guide - Discusses Service model, Device model and Workflow development

ATOM Solution Overview

FEATURES



Following sections provide a brief overview of ATOM Features.

Configuration Management

ATOM provides Configuration management capabilities for a wide variety of devices. This includes configuration archival, scheduling, trigger driven configuration sync, configuration diff etc.,

Topology

ATOM provides topology discovery through CDP & LLDP. Topology can be displayed hierarchically using Resource Pools (Device Groups). Topology overlays Alarms and Performance information.

Collection & Reporting

ATOM supports collection of network operational and performance data through various protocols like SNMP, SNMP Trap, Syslog & Telemetry. Such information can be visualized in ATOM as reports or can be rendered on Grafana as Charts. Admin guide discusses Report customization in further detail.

Network Automation

ATOM provides Model driven Network automation for stateful services. Stateful services involve a Service model (YANG) and some business logic. Service model development is covered in ATOM Platform guide. Admin guide discusses how to deploy & operate a service.

Workflow & Low Code Automation

ATOM provides an intuitive graphical designer to design, deploy and execute simple or complicated network operations and procedures. It allows the administrator to configure pre-checks, post-checks and approval flow. Workflow creation flows will be covered in the ATOM Platform Guide. Admin guide discusses how to deploy & operate.

Telemetry & Network Analytics

In today's economy, data is the new oil. Anuta's ATOM helps organizations collect a massive amount of network data from thousands of devices and generate detailed in-depth insights that will help them deliver innovative applications and solutions to their customers. ATOM can collect network data from a variety of sources including model-driven telemetry, SNMP and Syslog. The diverse data format of each source is normalized to provide a single consistent view to the administrator. Grafana is packaged as part of ATOM to view historical data, observe patterns and predict future trends. Organizations can integrate their Big Data and AI platform with ATOM to generate business insights from the network element configuration and operational state.

Procedure to Create Native Telemetry Collection

- Create a new Telemetry Collection
 - Provide the name of collection
 - Choose Junos as platform
 - Select the transport as UDP which we will auto select the encoding as compact GPB with Dial Out Mode
- To configure resource filtering on device, select the filtering tab and choose the sensor name in dropdown & add regex pattern to configure
 - Select ALL option, if we have same resource filter across sensors
- Once the telemetry collection is provisioned, users can't edit the entry.
 - Subscription is not required in this case.

Closed Loop Automation

Anuta ATOM allows administrators to define a baseline behavior of their network and remediation actions to be initiated on any violation of this behavior. ATOM collects a large amount of network data from multi-vendor infrastructure using Google Protobufs and stores it in a time series database. ATOM correlation engine constantly monitors and compares the collected data with the baseline behavior to detect any deviations. On any violation, the pre-defined remediation action is triggered thereby always maintaining network consistency.

The solution simplifies troubleshooting by providing the context of the entire network. Customers can define KPI metrics and corrective actions to automate SLA compliance.

Multi-Vendor support

Anuta ATOM has the most comprehensive vendor support. It supports thousands of devices spanning across 45+ vendors and automates all the use-cases including Data Center Automation, InterCloud,

Micro-Segmentation, Security as a Service, LBaaS, Campus/Access, Branch/WAN, IP/MPLS Edge, Virtual CPE, and NFV.

General Concepts

RBAC

Various ATOM Features and Levels of Access (Read, Create, Delete) are customized through RBAC. This is described in further Detail in [User Management](#).

In case you are not able to access certain Feature or Policy / Data please contact your System Administrator.

Model Driven User Interface

Various ATOM Features are Model Driven or Driven by the Dynamic Pluggable Artifacts. Some of the following fall into this category:

1. Device Packages
2. Service Packages
3. Workflow
4. Reports

In case you do not find certain functionality expected in ATOM, please contact support@anutanetworks.com or your System Administrator.

Multi Tenancy

ATOM supports Multi-Tenancy across organizations and Sub-Tenancy within an Organization. This allows to vertically slice Any Data / Policies as per the business requirements of the Customer. Multi-Tenancy including Sharing, Wild Card usage to share across multiple Sub-tenants, Users within a Sub-Tenant and more details are discussed in [ATOM Multi Tenancy & Sub-Tenancy](#)

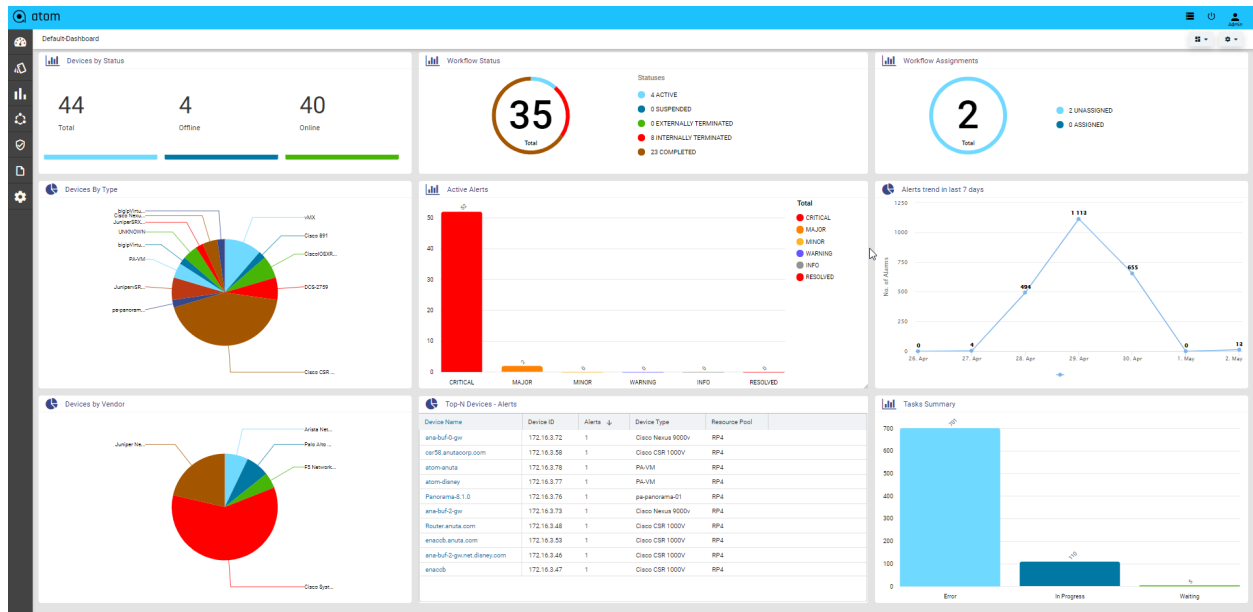
In case you are not able to access certain Feature or Policy / Data please contact your System Administrator.

Viewing the Dashboard

Dashboard provides a simple, integrated, comprehensive view of the data associated with the resources managed by ATOM. Information about the devices, services, service approvals are available “at-a-glance” for the administrator.

Starting from the 7.x release, Dashboard, the landing page of ATOM, is organized into dashlets. A dashlet is an individual component that can be added to or removed from a dashboard. Each dashlet is a reusable unit of functionality, providing a summary of the feature or the function supported by ATOM and is rendered as a result of the custom queries written in DSL.

You can customize the look of the Dashboard, by adding the dashlets of your choice, and dragging and dropping (the extreme right corner of the dashlet) to the desired location on the dashboard.



Each dashlet contains the summary or the overview of the feature or the functionality supported by ATOM.

For example, the dashlet “Device” displays the summary of devices managed by ATOM.

Some of the statistics that can be of interest in this dashlet could be as follows:

- Total number of devices
- Number of online devices
- Number of offline devices

These statistics can be gathered by ATOM and displayed in the corresponding dashlet depending on the DSL query written for each of them. You can save the layout containing the dashlets of your choice and set in a particular order.

Resource Management

ATOM Resource management involves device credential management, device onboarding through discovery or manual import, configuration archival, topology discovery & visualization, resource pools (device grouping), IP Address Management etc.,

Following table provides a quick summary of the activities that can be Resource Management activities.

If you want to..	Navigate to ...
------------------	-----------------

Credential Sets, Credential Maps and Devices	Resource Manager > Devices
Device Discovery	Resource Manager > Devices > Discovery
Visualize Topology	Resource Manager > Network > Topology
Create & Visualize Logical & Hierarchical Network Device Groups/Resource Pools	Resource Manager > Network > Resource Pools
Create physical locations	Resource Manager > Locations

Device Management

Device Management involves [onboarding of devices](#) and working with Device inventory, Configuration, Monitoring & Alerts. Devices can be added Manually, through an API or [Automated Discovery](#) using CDP/LLDP.

All Device Mgmt activities can be performed from Device Explorer & Grid View. Following are the three main views for a Device.

- Grid View - Grid layout of all Devices & and action on a device(s)
- Tree View - Device Group based tree view of devices that provides a much easier way to toggle between devices and inspect various device characteristics.
- Topology View - Devices can be visualized in a Topology view
- Device Detail View - On Clicking a Device from Tree View or Grid View a detailed view of the device is presented. This is same as the view when a device is selected from the Tree view

Grid, Tree view & Topology Views can be toggled using the view selector button available at the bottom right hand side corner of the page.

The screenshot displays the ATOM Network Monitoring interface. On the left is a sidebar with a tree view of devices, each represented by a colored dot and an IP address. The main area shows a table of alerts under the 'Alerts' tab. The table has columns for Alert Name, Device/NodeName, Device Name, Device FQDN, Resource, Severity, Ack, Status, and Message. Several alerts are listed, all with a 'Critical' severity and 'Open' status. A red box highlights a button in the bottom right corner of the table area.

Alert Name	Device/NodeName	Device Name	Device FQDN	Resource	Severity	Ack	Status	Message
InterfaceIngressPacketDrops	172.16.18.176	eor-cbb-2-gw.net.disney.com		/controller/devices/device=172.16.18.176/int...	Critical	⊗	Open	Interface GigabitEthernet
InterfaceError_Notification...	172.16.18.176	eor-cbb-2-gw.net.disney.com		/controller/devices/device=172.16.18.176/int...	Critical	⊗	Open	The input error rate of Gic
HC_Ingress_Interface_Utiliz...	172.16.18.176	eor-cbb-2-gw.net.disney.com		/controller/devices/device=172.16.18.176/int...	Critical	⊗	Open	The HC Ingress Utilisation
CPU_Utilization_Cisco	172.16.18.176	eor-cbb-2-gw.net.disney.com		/controller/devices/device=172.16.18.176	Critical	⊗	Open	CPU usage reached to 92
Memory_Utilization	172.16.5.83	test		/controller/devices/device=172.16.5.83	Critical	⊗	Open	Device 172.16.5.83 is usi
> [] DEVICE_DIAGNOSTI...	172.16.1.145			/controller/devices/device=172.16.1.145	Critical	⊗	Open	Device 172.16.1.145 Diag
[] DEVICE_AVAILABILITY	172.16.1.145			/controller/devices/device=172.16.1.145	Critical	⊗	Open	The Status of the Device
[] DEVICE_AVAILABILITY	172.16.3.37	BigIP_Active.com		/controller/devices/device=172.16.3.37	Critical	⊗	Open	The Status of the Device
[] DEVICE_DIAGNOSTICS	172.16.5.95	172.16.5.95		/controller/devices/device=172.16.5.95	Critical	⊗	Open	Device 172.16.5.95 Diag

Credential Management

ATOM provides multiple functions like Provisioning, Inventory Collection etc. Function like Provisioning can be various ways - Payload (CLI vs YANG or Other) over a Given Transport (SSH, Telnet, HTTP(S), etc.,). For example, based on the use case ATOM Workflow Engine can use various Payload + Transport mechanisms to perform Provisioning actions. ATOM helps accomplish this using:

- Credential Sets - Define the Transport/Connectivity & Authentication to the devices
- Credential Profile - Maps Credential Sets to various functions in ATOM

This addresses various scenarios, some as follows:

- Reuse of same SNMP Credentials across the entire Network, while retaining Device/Vendor Specific Transport for Provisioning.
- Inventory Collection Via SNMP for a Given Vendor/Device vs Telemetry for another

Credential Sets

Following section provides guidance on how to configure device credentials in ATOM.

1. Navigate to **Resource Manager > Devices > Grid View(Icon) > Credential Sets**
2. Create/Edit a Credential Set
 - **Name:** Enter a string that will be used to identify the Credential Set
 - **Description:** Enter a description w.r.t the created Credential Set(Optional)

SNMP Transport credentials:

Select Transport type as “SNMP” can view below option

- **SNMP version:** Select the version of SNMP that should be used for device communication
- **SNMP Read Community String:** Enter the string that is used by the device to authenticate ATOM before it can retrieve the configuration from the device
- **SNMP Write Community String:** Enter the string that is used by the device to authenticate ATOM while writing configuration to the device
- **Timeout:** Enter the time taken for the response from the device in seconds.
- **Number Of Retries:** Enter the number of times the SNMP request is sent when a timeout occurs.

The screenshot shows the 'Create Credential Set' form in the ATOM web interface. The form is titled 'Create Credential Set' and has a sidebar with navigation icons. The main form area contains the following fields and options:

- Name:** A text input field with the value 'SNMP_credential_set'.
- Description:** A text input field with the value 'Description of the credential set'.
- Transport Type:** A dropdown menu with the value 'SNMP'.
- SNMP Version:** A section with three radio buttons: 'SNMPV1', 'SNMPV2C' (selected), and 'SNMPV3'.
- SNMP Read Community STR:** A text input field with the value '*****'.
- SNMP Write Community STR:** A text input field with the value 'SNMP Write Community STR'.
- Time Out:** A text input field with the value '30'.
- Number Of Retries:** A text input field with the value '2'.

CLI Device(SSH/TELNET) Transport Credentials:

Select Transport type as “SSH/TELNET”

- **User name:** Enter a string that should be used to login to the device
- **Password:** Enter a string that used be a password for logging into the device
- **Enable Password:** Enter a password to enter into the privilege exec mode of the device.
- **Mgmt-VRF-Name:** Enter the name of the management VRF configured on the device. This will be used by ATOM to retrieve the audit logs from the device.

- **Port Number:** Enter the number of the port on the device that should be used for communication with ATOM
- **Command Execution Wait Time:** Enter the number (in millisecs) that ATOM should wait for the consecutive commands to be executed on the device. Enter any number between 10 to 30000.
- **CLI Configure Command TimeOut:** Enter the time (in seconds) that ATOM should wait for the command line prompt on the device to appear. Enter any between 1 to 1200.
- **Max Connections:** Enter the number of max connections that can be opened for a given device at any time.

The screenshot shows the 'Create Credential Set' form in the ATOM application. The form is titled 'Create Credential Set' and has a sidebar with various icons. The main form area contains the following fields:

- Name:** ssh-cli
- Description:** Description of the credential set
- Transport Type:** SSH
- User Name:** admin
- Password:** Password for logging into devices
- Enable Password:** Enable password used to enter into Privileged EXEC Mode on devices
- Management VRF Name:** Management VRF Name
- Port Number:** 22
- Command Execution Wait Time:** 150

API Device Transport Credential:

Select Transport type as “HTTP_HTTPS / GRPC”

- **User name:** Enter a string that should be used to login to the device
- **Password:** Enter a string that used be a password for logging into the device
- **Port Number:** Enter the number of the port on the device that should be used for communication with ATOM.
- **Max Connections:** Enter the number of max connections that can be opened for a given device at any time.

The screenshot shows the 'Create Credential Set' form in the ATOM web interface. The form is titled 'Create Credential Set' and has a sidebar with navigation icons. The main form area contains the following fields:

- Name:** A text input field containing 'https-netconf'.
- Description:** A text input field containing 'Description of the credential set'.
- Transport Type:** A dropdown menu with 'HTTP/HTTPS' selected.
- Commit-Confirm-Timeout:** A text input field containing '30'.
- User Name:** A text input field containing 'admin'.
- Password:** A password input field with a toggle icon.
- Port Number:** A text input field containing '830'.
- Max Connections:** A text input field containing '4'.

GRPC Transport credential:

The screenshot shows the 'Create Credential Set' form in the ATOM web interface, configured for a GRPC transport type. The form is titled 'Create Credential Set' and has a sidebar with navigation icons. The main form area contains the following fields:

- Name:** A text input field containing 'grpc'.
- Description:** A text input field containing 'Elastic+123'.
- Transport Type:** A dropdown menu with 'GRPC' selected.
- User Name:** A text input field containing 'admin'.
- Password:** A password input field with a toggle icon.
- Port Number:** A text input field containing '22'.
- Max Connections:** A text input field containing '4'.

Credential Profile

By default, ATOM has the following out of the box functions:

- Config Provisioning
- SNMP
- Telemetry
- HTTP provisioning
- NETCONF provisioning

Navigate to **Resource Manager > Devices > Grid View(Icon) > Credential Profile**

- Here, provide the name of credential profile, description and add the transport credentials by choosing the appropriate functions.

Create Credential-Profile

Entities

- Credential-Profile - Netconf-Https
 - Transport-Credentials - 1
 - Transport-Credentials - 2
 - Transport-Credentials - 3

Transport Credentials

netconf-https

Description

Description of the credential-profile

Description

- Below is the snapshot to attach the credential set with function.

Entities

- Credential-Profile - DistributedSWTelemetry
 - Transport-Credentials - HWInventory
 - Transport-Credentials - PerfStats
 - Transport-Credentials - Provisioning_Compliance

Id

transport credentials key

PerfStats

Function

Credential function name

TELEMETRY COLLECTION

Credential Set

credential set name

TelemetryPerformance

Credential profile payload in XML:

```
<credential-profile>
  <transport-credentials>
    <id>HWInventory</id>
    <function>SNMP COLLECTION</function>
  </transport-credentials>
  <credential-set>SNMPInventory</credential-set>
  <transport-credentials>
    <id>PerfStats</id>
    <function>TELEMETRY COLLECTION</function>
  </transport-credentials>
  <credential-set>TelemetryPerformance</credential-set>
  <transport-credentials>
    <id>Provisioning_Compliance</id>
    <function>NETCONF PROVISIONING</function>
  </transport-credentials>
  <credential-set>NetconfProvisioning</credential-set>
  <description>We will use SNMP to collect device inventory, Telemetry to get performance stats, Config Backup & Compliance with Raw CLI and Provisioning via Netconf</description>
  <name>DistributedSWTelemetry</name>
</credential-profile>
```

Credential Maps

Credential Map allows users to map multiple Credentials Profiles to an IP-Address range. This addresses the following use cases:

- Device Discovery - When ATOM needs to Perform Discovery using SNMP Sweep or CDP/LLDP. Since devices are yet to be onboarded, explicit assignment is not available.
- Credential profile is mandatory when onboarding a device.

When ATOM needs credentials for a device and explicit Device to Credential Profile is not available, ATOM will cycle through the IP Address range and use the first credential profile that works. The successful Credential Profile is mapped to the device. This process is repeated

whenever ATOM is unsuccessful communicating with the device using the current assigned credential profile.

To create a Credential Map:

1. Navigate to **Resource Manager > Devices > Grid View(Icon) > Credential Maps**
2. Create/Edit **Create Credential Map**:
 - **Name**: Enter a name for the Credential Map
 - **Start-IP-address**: Enter an IP address in the range from which ATOM starts the sweep for locating the devices.
 - **End-IP-address**: Enter an IP address in the range beyond which ATOM will not continue the sweep for locating the devices.

Note: The Start and the End IP address are the range of IP addresses of the devices.

- **Credential Profile**: Select one or more Credential Profiles shown.

The screenshot shows the 'Create Credential Map' window in the ATOM application. The interface has a blue header bar with the ATOM logo and navigation icons. The main content area is white and contains the following fields:

- Name**: A text input field with the value 'map'. Below it, a small note says 'Allows AlphaNumerics, hyphen and underscore characters only. Max length is 36'.
- IP Range**: A text input field with the value '172.16.3.30-172.16.3.41'.
- Credential-Profile**: A section with a sub-header 'Select one or more Credential-Set(s) to associate with range of IP Address specified'. Below this is a table with 9 rows. The first row is selected, and the 'profile1' row is checked.

Credential-Profile	Description	Owner
<input type="checkbox"/> BIGIP_API	BIGIP_API	system
<input type="checkbox"/> PAN_API_443	PAN_API_443	system
<input type="checkbox"/> SSH_SNMP	SSH and SNMP Profile	system
<input type="checkbox"/> WMTCCredentialProfile	WMTCCredentialProfile	system
<input type="checkbox"/> csd_profile	csd_profile	system
<input type="checkbox"/> evpn_3vpn_profile	evpn_3vpn_profile	system
<input type="checkbox"/> netconf_profile	netconf_profile	system
<input checked="" type="checkbox"/> profile1	profile1	system
<input type="checkbox"/> ssh_snmp_bgp	ssh_snmp_bgp	system

Device Onboarding

Devices can be onboarded into ATOM using an API, Manually through User Interface of Discovery using CDP/LLDP.

Discovering Devices:

Devices discovery is covered in section - [Device Discovery](#)

Adding Device Manually:

We may have scenarios where device discovery is not viable. Some reasons below:

- Lack of support for Layer 2 discovery support on the device
- Operational/Administrative reason to not use LLDP/CDP
- SNMP Sweep discovery is not suitable - IP Address Range are not well defined, contiguous or some other reasons

Before you begin, it's mandatory to define [Credential Sets](#) & [Credential Profiles](#).

To Add/Edit a Device:

1. Navigate to **Resource Manager** > [Devices \(Grid View\)](#)
2. Add - Select **Add** action
 - a. **IP address:** Enter the IP address of the device
 - b. **Credential Profile:** Select the Credential Profile of the device
 - c. **Driver name:** Driver can be selected for API devices.
 - d. **Latitude & Longitude:** is a measurement on a globe or map of location north or south of the Equator on devices
3. Modify - Select Device & Select **Edit** action
4. Delete - Select one/more device(s) and Select Delete Action

atom > Devices-List

Create Device

Id

Management Mode

MANAGED UNMANAGED DUMMY

Name

Fqdn-Name

Mgmt-Ip-Address

Credential-Profile

Device Type

UNKNOWN

Description

Driver-Name

Hardware-Throughput

Value should be in MB/s

Upon device addition, ATOM will perform the following:

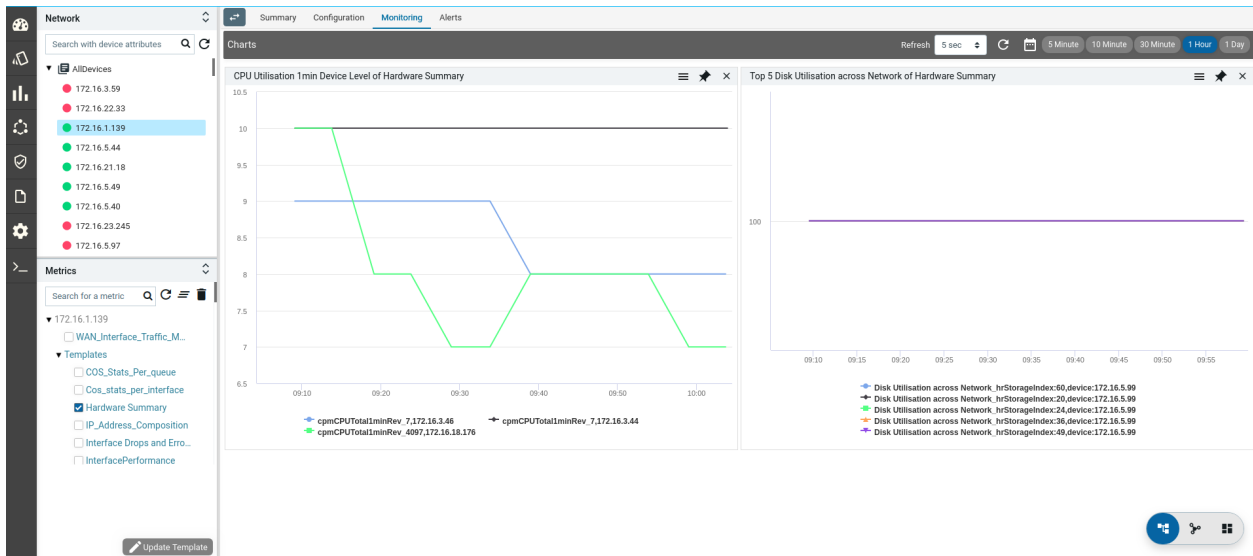
Added Devices are shown in Devices grid and Device status will be shown in **Green** if device is SNMP reachable and ATOM is able to work with the device successfully.

Devices								
Credential Sets Credential Profiles Credential Maps Discovery Discovered Devices								
32 Of 32 Search								
<input type="checkbox"/> Id ↑	Status	Mgmt-Ip-Address	Fqdn-Name	Name	Device Type	Credential Set	Last-Extended-Inv-Time	Os-Version
<input type="checkbox"/> 172.16.18.176	●	172.16.18.176		eor-cbb-2-gw.net.disney.com	CiscoIOSXrv9000		06/09/21, 5:00:35 PM	6.1.4
<input type="checkbox"/> 172.16.3.33	●	172.16.3.33		eorwdw-200cel1-gw.net.disney.com	Cisco CSR 1000V		06/17/21, 4:02:32 PM	15.6(1)S
<input type="checkbox"/> 172.16.3.34	●	172.16.3.34		ana-cd-1-gw.anutanetworks.com	Cisco CSR 1000V		06/15/21, 2:25:58 PM	15.6(1)S
<input type="checkbox"/> 172.16.3.36	●	172.16.3.36		test_enacbb.anutacorp.com	Cisco CSR 1000V		06/15/21, 2:25:43 PM	15.6(1)S
<input type="checkbox"/> 172.16.3.40	●	172.16.3.40		wnacrp-dtss-0-gw.anutanetworks.com	Cisco CSR 1000V		06/15/21, 2:25:25 PM	15.6(1)S
<input type="checkbox"/> 172.16.3.42	●	172.16.3.42		n7-cbb-0-gw.anutanetworks.com	Cisco CSR 1000V		05/20/21, 5:14:19 PM	15.6(1)S
<input type="checkbox"/> 172.16.3.43	●	172.16.3.43		wnacrp-dtss-0-gw.net.disney.com	Cisco CSR 1000V			
<input type="checkbox"/> 172.16.3.44	●	172.16.3.44		wbucbb-burbank0-gw.net.disney.com	Cisco CSR 1000V		06/09/21, 4:59:35 PM	15.6(1)S
<input type="checkbox"/> 172.16.3.45	●	172.16.3.45		ana-buf-1-gw.anutanetworks.com	Cisco CSR 1000V		06/08/21, 5:17:40 PM	15.6(1)S
<input type="checkbox"/> 172.16.3.46	●	172.16.3.46		wbucbb-burbank1-gw.anutanetworks.com	Cisco CSR 1000V		06/11/21, 3:57:53 PM	15.6(1)S
<input type="checkbox"/> 172.16.3.48	●	172.16.3.48		Router.anutanetworks.com	Cisco CSR 1000V		05/20/21, 2:53:55 PM	15.6(1)S
<input type="checkbox"/> 172.16.3.51	●	172.16.3.51			UNKNOWN			
<input type="checkbox"/> 172.16.4.166	●	172.16.4.166		vmX-4.99.anutanetworks.com	VMX		06/10/21, 3:00:42 PM	19.4R3.1
<input type="checkbox"/> 172.16.4.167	●	172.16.4.167		mx6.anutanetworks.com	VMX		06/14/21, 6:34:34 PM	19.4R3.1
<input type="checkbox"/> 172.16.4.168	●	172.16.4.168		mx7.anutanetworks.com	VMX		06/09/21, 7:08:03 PM	19.4R3.1
<input type="checkbox"/> 172.16.4.169	●	172.16.4.169		mx8.anutanetworks.com	VMX		06/10/21, 7:16:21 PM	19.4R3.1

Device Views

ATOM has 3 views for the devices - Tree (Device Explorer), Topology and Grid.

1. Tree View:



2. Topology View



3. Grid View:

atom > Devices-List									
Devices Credential Sets Credential Profiles Credential Maps Discovery									
1 - 50 Of 181 Page 1 Of 4 Search									
<input type="checkbox"/>	Id	us	Mgmt-Ip-Address	Fqdn-Name	Name	Device Type	Credential Set	Last-Extended-try-Time	Os-Version
<input type="checkbox"/>	10.1.1.4		10.1.1.4		ciscoAircT5508K9	ciscoAircT5508K9	SSH_1		
<input type="checkbox"/>	10.1.1.5		10.1.1.5		cat385024P-3	cat385024P	SSH_1		
<input type="checkbox"/>	10.1.1.6		10.1.1.6		cat385024P-1	cat385024P	SSH_1		
<input type="checkbox"/>	10.1.15.19		10.1.15.19		PaloAlto	PA-5060	SSH		
<input type="checkbox"/>	10.1.15.21		10.1.15.21		Clisco Virtual ASA	Clisco Virtual ASA	SSH		
<input type="checkbox"/>	10.1.15.23		10.1.15.23		ciscoASA5510	ciscoASA5510	SSH		
<input type="checkbox"/>	10.1.15.25		10.1.15.25		Clisco Nexus 7004	Clisco Nexus 7004	SSH		
<input type="checkbox"/>	10.1.15.27		10.1.15.27		Clisco Nexus 5010 Switch	Clisco Nexus 5010 Switch	SSH		
<input type="checkbox"/>	10.1.15.29		10.1.15.29		Clisco Nexus 3064 Switch	Clisco Nexus 3064 Switch	SSH		
<input type="checkbox"/>	10.1.15.31		10.1.15.31		Clisco ASR 9006	Clisco ASR 9006	SSH		
<input type="checkbox"/>	10.1.15.33		10.1.15.33		BIG-IP LTM VE	BIG-IP LTM VE	SSH		
<input type="checkbox"/>	10.1.15.34		10.1.15.34		APIC	APIC	SSH		
<input type="checkbox"/>	10.1.2.1		10.1.2.1		CliscoSR4331	CliscoSR4331	SSH_1		
<input type="checkbox"/>	10.1.2.2		10.1.2.2		ciscoAircT2504K9	ciscoAircT2504K9	SSH_1		
<input type="checkbox"/>	10.1.2.3		10.1.2.3		Catalyst4500X-16	Catalyst4500X-16	SSH_1		
<input type="checkbox"/>	10.1.2.4		10.1.2.4		Clisco Catalyst3560	Clisco Catalyst 3560E-12SD	SSH_1		
<input type="checkbox"/>	10.1.2.5		10.1.2.5		cat385024P-4	cat385024P	SSH_1		
<input type="checkbox"/>	10.10.2.3					INFOBLOX			
<input type="checkbox"/>	172.16.1.138		172.16.1.138		Router	Clisco 871		11/21/19, 4:03:40 PM	12.4(15)T10
<input type="checkbox"/>	172.16.1.139		172.16.1.139		eorwdw-aaahq1-vpn1-gw.net.disney.com	Clisco 891		03/03/20, 2:53:34 PM	15.5(1)T
<input type="checkbox"/>	172.16.1.145		172.16.1.145			UNKNOWN	telnet		
<input type="checkbox"/>	172.16.1.146		172.16.1.146		Router146.anutacorp.com	Clisco 2951	telnet	07/26/19, 3:52:06 PM	15.3(BLD_T_BASE_3_OLYMPUS_201302140413)
<input type="checkbox"/>	172.16.1.150		172.16.1.150		CE1-ISR.anutaqa.com	Clisco3945SP250	SSH	07/26/19, 3:52:25 PM	15.1(1)T2
<input type="checkbox"/>	172.16.1.227		172.16.1.227		PE2-ASR1002	UNKNOWN	ssh_tprnv		

Device Explorer

Device explorer view will provide the devices, its associated config and observability elements in logical hierarchy. This view contains the available device-groups and its associated devices . By default, all the devices are part of **AllDevices** Group.

Device group will have all the corresponding device details Each group and node will have the following sections:

1. **Summary** : It provides the device platform, version, serial number, current operating OS, Device hardware health, Interface summary, Config compliance violations and Active alerts and recent activity.

2. **Configuration** : it provides the entire summary of config related operations.
 - a. **[Config Archive](#)** : It shows the each config retrieval, type, retrieval & parsing status.
 - b. **Changelog** : provides the summary of change in configuration such as number of lines added, deleted or modified and at what time & corresponding changes.
 - c. **Config Data** : it will provide the entire config tree through YANG models parsing. This is not applicable for any device group as they can have heterogeneous models based on the grouping criteria & provisioning interface such as ATOM abstract device models, OC or Native models.
3. **Monitoring** : It contains all possible templates & charts through inheritance from its group or node level. It will show the default template by default as its monitoring summary. Refer Monitoring Guide for more details.
4. **Alerts** : It will show the all active alerts and its history by default. Alert filter view is also available to search & prioritise the alerts. Refer Alerting Guide for more details.

Each device-group view will have a Summary dashboard which can be customizable.

Device Actions

ATOM supports common actions on Device. These actions can be performed from Device Grid view on one or more devices or from within the Device specific view and will be discussed in [Device Summary](#) section.

Jobs & Subscriptions

Various Collection & Diagnostics jobs can be invoked.

1. Navigate to **Devices** > select one or more devices
2. Click on the **Jobs** and select the job to run
 - a. Jobs action -> Run Device Inventory
 - b. Jobs action -> Run Extended Inventory
 - c. Jobs action -> Run Topology Inventory
 - d. Jobs action -> Retrieve Configs
 - e. Jobs action -> Run Diagnostics
 - f. Jobs action -> Run Policy
 - g. Jobs action -> Run Profile
3. Click on the **Subscriptions** to configure Syslog Subscription on the Devices
 - a. This will result in ATOM being configured as a Syslog receiver and is a configuration change on the device.

ID	Status	Jobs	Fqdn-Name	Name	Device Type	Credential Set	Last-Extended-In-Time	Os-Version	Platform
172.16.17.135	●	Run Device Inventory	asr17_135	asr17_135	CiscoIOSXRv9000		10/30/20, 11:39:23 AM	6.3.3	6.3.3/CiscoIOSXRv9000/Cisco ASR 9000/IOSXE/Cisco Systems
172.16.3.30	●	Run Extended Inventory	wnacrp-dtss-0-gw.net.disney.com	wnacrp-dtss-0-gw.net.disney.com	Cisco CSR 1000V		11/01/20, 10:16:04 AM	15.6(1)S	15.6(1)S/Cisco CSR 1000V/Cisco CSR 1000V/IOSXE/Cisco Syst...
172.16.3.31	●	Run Topology Inventory	ana-buf-6-gw.anutacorp.com	ana-buf-6-gw.anutacorp.com	Cisco CSR 1000V		10/30/20, 11:05:26 AM	15.6(1)S	15.6(1)S/Cisco CSR 1000V/Cisco CSR 1000V/IOSXE/Cisco Syst...
172.16.3.32	●	Retrieve Configs	enacbb-kmtc.net.disney.com	enacbb-kmtc.net.disney.com	Cisco CSR 1000V		10/07/20, 10:20:02 AM	15.6(1)S	15.6(1)S/Cisco CSR 1000V/Cisco CSR 1000V/IOSXE/Cisco Syst...
172.16.3.40	●	Run Diagnostics	wnacrp-dtss-0-gw.test.com	wnacrp-dtss-0-gw.test.com	Cisco CSR 1000V		10/07/20, 11:03:49 AM	15.6(1)S	15.6(1)S/Cisco CSR 1000V/Cisco CSR 1000V/IOSXE/Cisco Syst...
172.16.3.42	●	Run Policy	n7-cbb-0-gw	n7-cbb-0-gw	Cisco CSR 1000V		10/06/20, 12:53:36 PM	15.6(1)S	15.6(1)S/Cisco CSR 1000V/Cisco CSR 1000V/IOSXE/Cisco Syst...
172.16.3.43	●	Run Profile	wnacrp-dtss-0-gw.net.disney.com	wnacrp-dtss-0-gw.net.disney.com	Cisco CSR 1000V		10/21/20, 10:14:14 PM	15.6(1)S	15.6(1)S/Cisco CSR 1000V/Cisco CSR 1000V/IOSXE/Cisco Syst...
172.16.3.46	●		aanbcb-ana-0-gw	aanbcb-ana-0-gw	Cisco CSR 1000V		10/08/20, 8:42:04 AM	15.6(1)S	15.6(1)S/Cisco CSR 1000V/Cisco CSR 1000V/IOSXE/Cisco Syst...
172.16.3.73	●		ana-buf-2-gw	ana-buf-2-gw	Cisco Nexus 9000v		10/05/20, 4:45:50 PM	7.0(3)(7)(4)	7.0(3)(7)(4)/Cisco Nexus 9000v/Cisco Nexus 9000/NXOS/Cisco ...
172.16.3.76	●		172.16.3.76	172.16.3.76	Panorama			ALL	ALL/Panorama/Panorama/PANOS/Panorama Systems
172.16.4.154	●		172.16.4.154	172.16.4.154	vMX		10/10/20, 2:07:18 AM	20.2R1.10	20.2R1.10/vMX/Jumper MX/JUNOS/Jumper Networks
172.16.4.60	●		172.16.4.60	172.16.4.60	bigipVirtual			ALL	ALL/bigipVirtual(Bigip)TMOS/F5 Networks
172.16.4.61	●		172.16.4.61	172.16.4.61	bigipVirtual			ALL	ALL/bigipVirtual(Bigip)TMOS/F5 Networks
172.16.5.104	●		172.16.5.104	172.16.5.104	JuniperVSRX		10/06/20, 3:02:39 PM	17.3R2.10	17.3R2.10/JuniperVSRX/Juniper SRX Series Firewall/JUNOS/Ju...
172.16.5.80	●		172.16.5.80	172.16.5.80	Router123		10/06/20, 12:49:10 PM	15.7(3)M6	15.7(3)M6/Cisco 2911/Cisco 2900/IOS/Cisco Systems
172.16.5.80_swim	●		172.16.5.80	172.16.5.80	Router123		10/06/20, 10:53:12 AM	15.7(3)M6	15.7(3)M6/Cisco 2911/Cisco 2900/IOS/Cisco Systems
172.16.5.89	●		172.16.5.89	172.16.5.89	vMX		10/08/20, 1:53:52 PM	20.2R1.10	20.2R1.10/vMX/Jumper MX/JUNOS/Jumper Networks
172.16.5.95	●		172.16.5.95	172.16.5.95	vMX		10/13/20, 1:39:54 PM	17.4R1.16	17.4R1.16/vMX/Jumper MX/JUNOS/Jumper Networks

atom

> Devices-List

Devices

Credential Sets

Credential Profiles

Credential Maps

Discovery

8 of 8

Search

Id	Status	Notifications	Fqdn-Name	Name	Device Type	Credential Set	Last-Extended-Inv-Time	Os-Version	Platform
<input checked="" type="checkbox"/> 172.16.18.176	<div></div>	Subscribe To SysLog Events		eor-cbb-2-gw.net.disney.com	CiscoIOSXRv9000		05/26/20, 12:16:25 PM	6.1.4	6.1.4/CiscoIOSXRv9000/Cisco ASR 9000/IOSXR/Cisco Systems
<input type="checkbox"/> 172.16.3.32	<div></div>	Unsubscribe To SysLog Events		anuta-lab02.anutacorp.com	Cisco CSR 1000V		05/26/20, 12:17:06 PM	15.6(1)S	15.6(1)S/Cisco CSR 1000V/Cisco CSR 1000V/IOSXE/Cisco Syst...
<input type="checkbox"/> 172.16.3.33	<div></div>			wbucbb-bur-0-gw.net.disney.com	Cisco CSR 1000V		05/26/20, 11:58:42 AM	15.6(1)S	15.6(1)S/Cisco CSR 1000V/Cisco CSR 1000V/IOSXE/Cisco Syst...
<input type="checkbox"/> 172.16.3.41	<div></div>	172.16.3.41		CSR_3.41.anutacorp.com	Cisco CSR 1000V		05/26/20, 12:15:45 PM	15.6(1)S	15.6(1)S/Cisco CSR 1000V/Cisco CSR 1000V/IOSXE/Cisco Syst...
<input type="checkbox"/> 172.16.3.42	<div></div>	172.16.3.42		n7-cbb-0-gw.net.disney.com	Cisco CSR 1000V		05/26/20, 12:09:09 PM	15.6(1)S	15.6(1)S/Cisco CSR 1000V/Cisco CSR 1000V/IOSXE/Cisco Syst...
<input type="checkbox"/> 172.16.3.43	<div></div>	172.16.3.43		ana-svc-0-gw.net.disney.com	Cisco CSR 1000V		05/26/20, 12:09:45 PM	15.6(1)S	15.6(1)S/Cisco CSR 1000V/Cisco CSR 1000V/IOSXE/Cisco Syst...
<input type="checkbox"/> 172.16.3.44	<div></div>	172.16.3.44		wbucbb-bur-1-gw.net.disney.com	Cisco CSR 1000V				ALL/Cisco CSR 1000V/Cisco CSR 1000V/IOSXE/Cisco Systems
<input type="checkbox"/> 172.16.3.49	<div></div>	172.16.3.49		wbu.anuta.com	Cisco CSR 1000V		05/26/20, 12:15:13 PM	15.6(1)S	15.6(1)S/Cisco CSR 1000V/Cisco CSR 1000V/IOSXE/Cisco Syst...

Exporting Device Information:

You can export the device information of the devices either in the XML or JSON format.

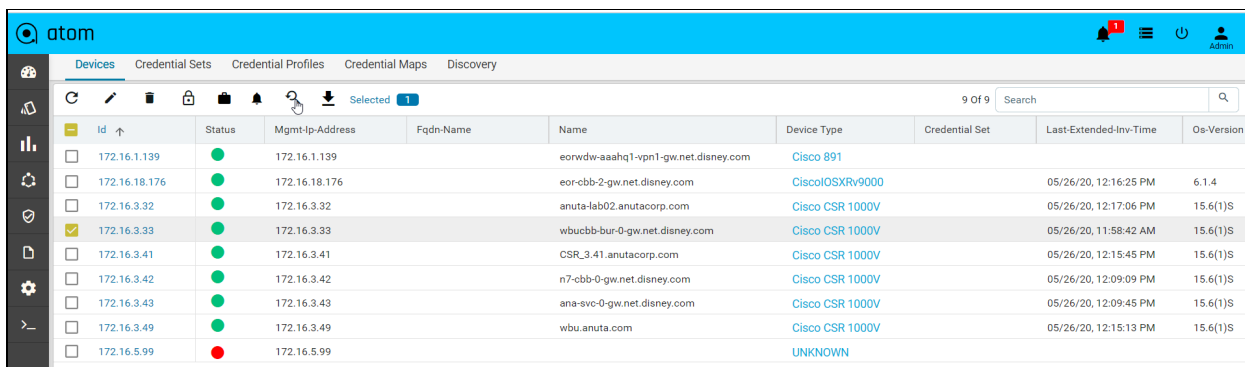
1. Navigate to **Resource Manager > Devices > Grid View(Icon) > Devices**
2. Select one or more devices

- Click the **View/Download** button and select either the XML or JSON

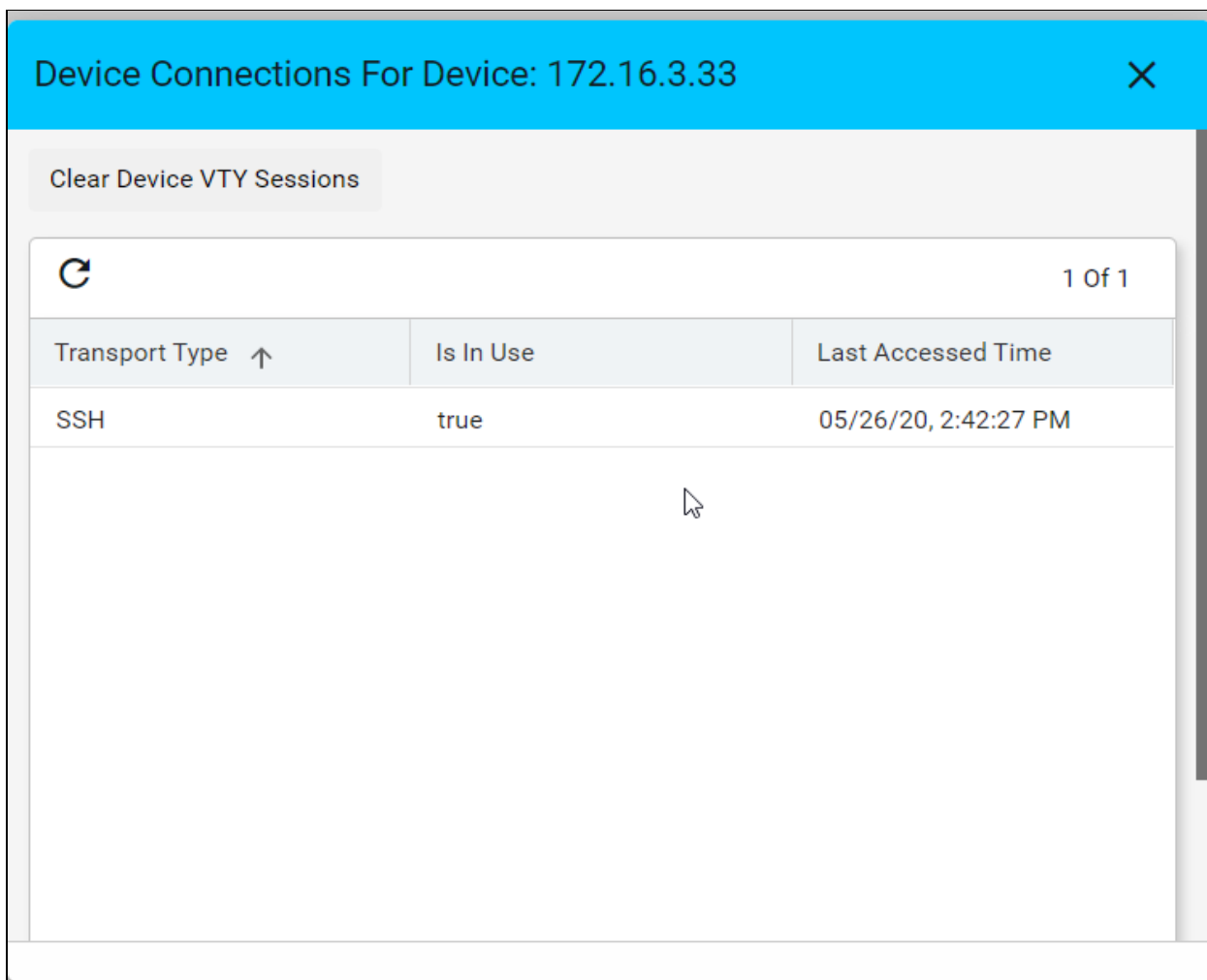
VTY Sessions

This is used to view the active vty sessions.

- Navigate to **Resource Manager > Devices > Grid View(Icon) > Devices**
- Select any device
- Click the **VTY Sessions** button



Id	Status	Mgmt-Ip-Address	Fqdn-Name	Name	Device Type	Credential Set	Last-Extended-Inv-Time	Os-Version
172.16.1.139	●	172.16.1.139		eorwdw-aaahq1-vpn1-gw.net.disney.com	Cisco 891			
172.16.18.176	●	172.16.18.176		eor-cbb-2-gw.net.disney.com	CiscoIOSXrv9000		05/26/20, 12:16:25 PM	6.1.4
172.16.3.32	●	172.16.3.32		anuta-lab02.anutacorp.com	Cisco CSR 1000V		05/26/20, 12:17:06 PM	15.6(1)S
172.16.3.33	●	172.16.3.33		wbucbb-bur-0-gw.net.disney.com	Cisco CSR 1000V		05/26/20, 11:58:42 AM	15.6(1)S
172.16.3.41	●	172.16.3.41		CSR_3.41.anutacorp.com	Cisco CSR 1000V		05/26/20, 12:15:45 PM	15.6(1)S
172.16.3.42	●	172.16.3.42		n7-cbb-0-gw.net.disney.com	Cisco CSR 1000V		05/26/20, 12:09:09 PM	15.6(1)S
172.16.3.43	●	172.16.3.43		ana-svc-0-gw.net.disney.com	Cisco CSR 1000V		05/26/20, 12:09:45 PM	15.6(1)S
172.16.3.49	●	172.16.3.49		wbu.anuta.com	Cisco CSR 1000V		05/26/20, 12:15:13 PM	15.6(1)S
172.16.5.99	●	172.16.5.99			UNKNOWN			



Device Connections For Device: 172.16.3.33

Clear Device VTY Sessions

1 Of 1

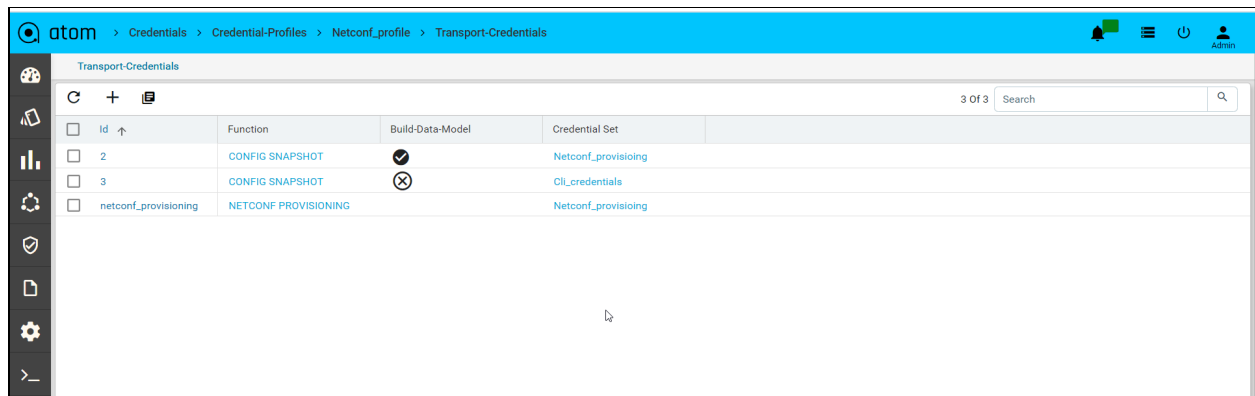
Transport Type	Is In Use	Last Accessed Time
SSH	true	05/26/20, 2:42:27 PM

Default Jobs

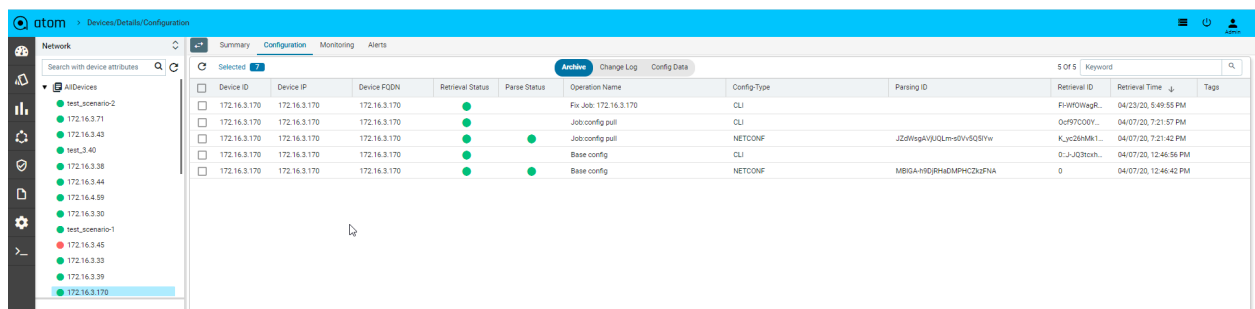
Below are the jobs which run during the device onboarding process in the mentioned order.

1. **Device Inventory** : It gathers the Platform, OS Version through SNMP and gets the Device to ONLINE. If the platform is not found in ATOM then check Platform guide on
2. **Device Extended Inventory** : It collects the Serial Number, Interface performance, health, availability etc.,
3. **Device Diagnostics** : ATOM will perform the reachability check through Ping, SNMP and Telnet/SSH if they are applicable.
4. **Base Config Pull or Config Retrieval** : It will retrieve the configuration and persist in the database. Configuration will be collected if the credential function is set to Config SNAPSHOT or any of the PROVISIONING functions. Build data model flag is used to parse the configuration into YANG entities from the specified config source snapshot.

Below is the example, to backup cli and netconf xml config and parse the xml version.



Config Type column will show us the source of config retrieval.



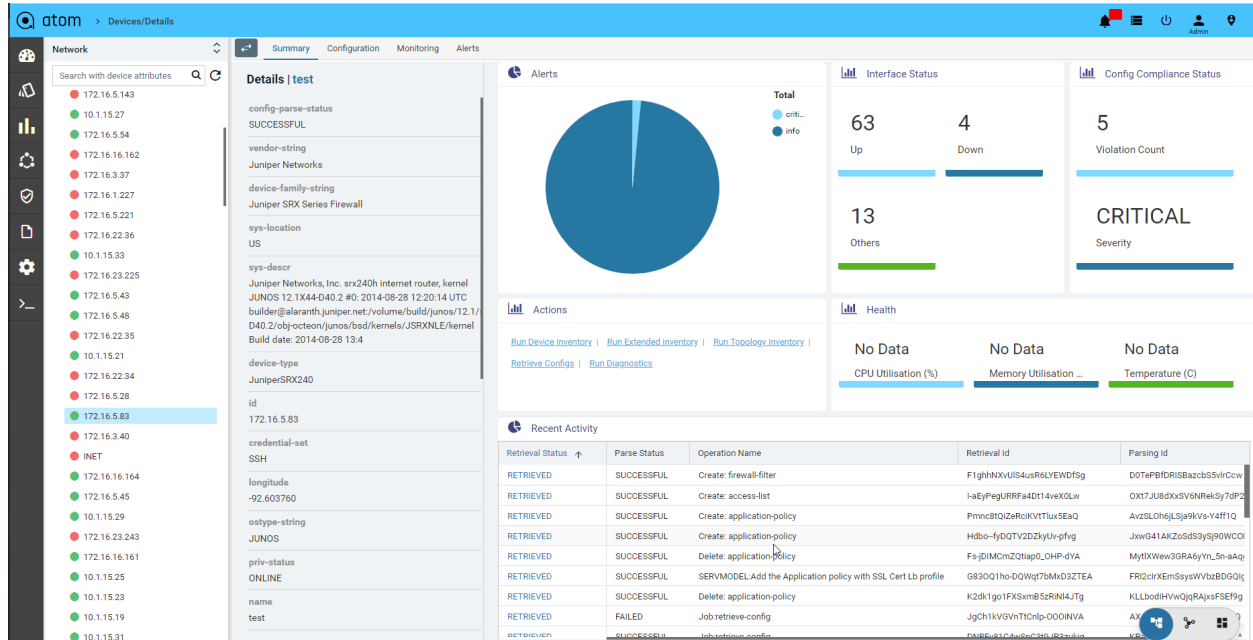
All the above operations can be customized for any platform as required and scheduled similar to other collection jobs.

Device Summary

Device summary view provides a quick snapshot of important device attributes including Alarm summary, interface summary, recent configuration change history and health.

Device Summary also provides access to most popular device actions and quick links to frequently used activities.

1. Navigate to **Devices** > select a device
2. Click on the **Device > Summary**> to view the details associated with each attribute.



Configuration Management

Configuration Archive

ATOM Collects Network or Server Configuration through API, NETCONF or CLI over Telnet or SSH. ATOM Provides the following Configuration Management relations functions:

1. Fetch, archive, and deploy device configurations
2. Build Stateful Configuration Model for:
 - a. Devices that support YANG Over NETCONF
 - b. Devices where Device YANG Model is mapped to Concrete API or CLI
3. Stateful Configuration - Support Create, Update, Delete
4. Stateful Configuration - Configuration Drift & Transactions
5. Compare Startup vs Running
6. Compare Running vs Latest Archived
7. Compare Two Versions of the Same Device
8. Compare Two Versions of Different Devices
9. Base Config vs Latest Version of Multiple Devices
10. Search and generate reports on archived data
11. Compare and label configurations, compare configurations with a baseline, and check for compliance.

12. You can use the Baseline template to compare with other device configurations and generate a report that lists all the devices that are non-compliant with the Baseline template.
13. You can easily deploy the Baseline template to the same category of devices in the network with dynamic inputs.
14. You can import or export a Baseline template/Config archives.
15. Set Up Event-Triggered Archiving
16. Synchronize Running and Startup Device Configurations
17. Deploy an External Configuration File to a Device
18. Roll Back a Device's Configuration To an Archived Version

ATOM Collects Device configuration periodically as configured in Jobs->Configuration or upon a config change event from the device. To trigger configuration collection through config change notification, ATOM should be configured to receive config change notification through SNMP Trap or Syslog.

1. To view Device(s) Configuration - Navigate to **Devices** > select a device(s)
2. Click on the **“Configuration > Archive”** Tab
3. Select an Entry in the Grid
4. In Details view - CLI/XML Configuration is displayed

The screenshot shows the ATOM Configuration Archive grid and the Configuration Details view. The grid displays a list of devices with their IP addresses, FQDNs, and retrieval/parse status. The Configuration Details view shows the CLI configuration for a selected device.

Device ID	Device IP	Device FQDN	Retrieval Status	Parse Status	Operation Name	Config-Type
172.16.1.139	172.16.1.139	eor-cbb-0-gw.net.dl...	●	●	Create: IS-service	
172.16.3.170	172.16.3.170	172.16.3.170	●	●	Fix Job: 172.16.3.170	NETCONF
172.16.3.170	172.16.3.170	172.16.3.170	●	●	Base config	NETCONF
172.16.1.139	172.16.1.139	eor-cbb-0-gw.net.dl...	●	●	Job retrieve-config	
172.16.5.41	172.16.5.41	ase-svc-1-gw.anuta...	●	●	Base config	
172.16.1.139	172.16.1.139	eor-cbb-0-gw.net.dl...	●	●	Job retrieve-config	
172.16.3.40	172.16.3.40		●	●	Job retrieve-config	
172.16.22.97	172.16.22.97		●	●	Config-Pull	
172.16.22.97	172.16.22.97		●	●	Config-Pull	
172.16.22.97	172.16.22.97		●	●	Config-Pull	
172.16.22.97	172.16.22.97		●	●	Config-Pull	
172.16.22.97	172.16.22.97		●	●	Config-Pull	
172.16.22.97	172.16.22.97		●	●	Fix Job: 172.16.22.97	
172.16.22.97	172.16.22.97		●	●	Config-Pull	
172.16.22.97	172.16.22.97		●	●	Config-Pull	
172.16.22.97	172.16.22.97		●	●	Config-Pull	
172.16.1.139	172.16.1.139		●	●	Job retrieve-config	
172.16.22.97	172.16.22.97		●	●	Config-Pull	
172.16.22.97	172.16.22.97		●	●	Config-Pull	
172.16.22.97	172.16.22.97		●	●	Config-Pull	
172.16.3.40	172.16.3.40		●	●	Job retrieve-config	
172.16.3.40	172.16.3.40		●	●	Config-Pull	
172.16.22.97	172.16.22.97		●	●	Config-Pull	
172.16.1.139	172.16.1.139		●	●	Create: logging-hosts	
172.16.1.139	172.16.1.139		●	●	Job retrieve-config	
172.16.3.40	172.16.3.40		●	●	Delete: IS-service	
172.16.3.40	172.16.3.40		●	●	Update: IS-service	

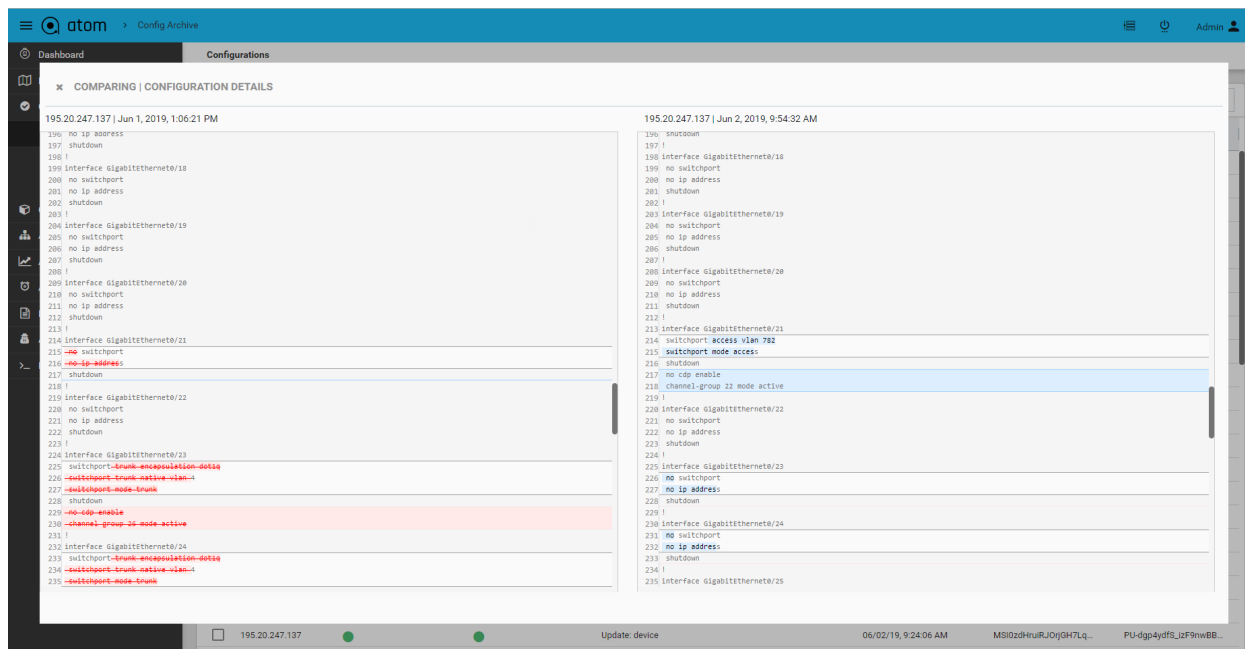
The Configuration Details view shows the CLI configuration for the selected device (172.16.3.40). The configuration includes various services and settings, such as 'show running-config', 'Building configuration...', 'Current configuration: 48217 bytes', 'Last configuration change at 01:18:32 UTC Tue Feb 18 2020 by admin', 'version 15.6', 'no service pad', 'service tcp-keepalives-in', 'service tcp-keepalives-out', 'service timestamps debug datetime msec localtime show-timezone', 'service timestamps log datetime msec localtime show-timezone', 'service password-encryption', 'service sequence-numbers', 'no platform punt-keepalive disable-kernel-core', 'platform console virtual', 'platform hardware throughput level MB 500', 'hostname test', 'boot-start-marker', 'boot system flash:/c800-universalk9-mz.SPA.154-3.M3', 'boot-end-marker', 'no shell processing', 'vrf definition 111', and 'address-family ipv4'.

Configuration Diff:

Configuration differences across various revisions can be viewed by selecting two versions from the Configuration archive grid.

1. To view Device(s) Configuration - Navigate to **Devices** > select a device(s)
2. Click on the **“Configuration > Archive”** Tab

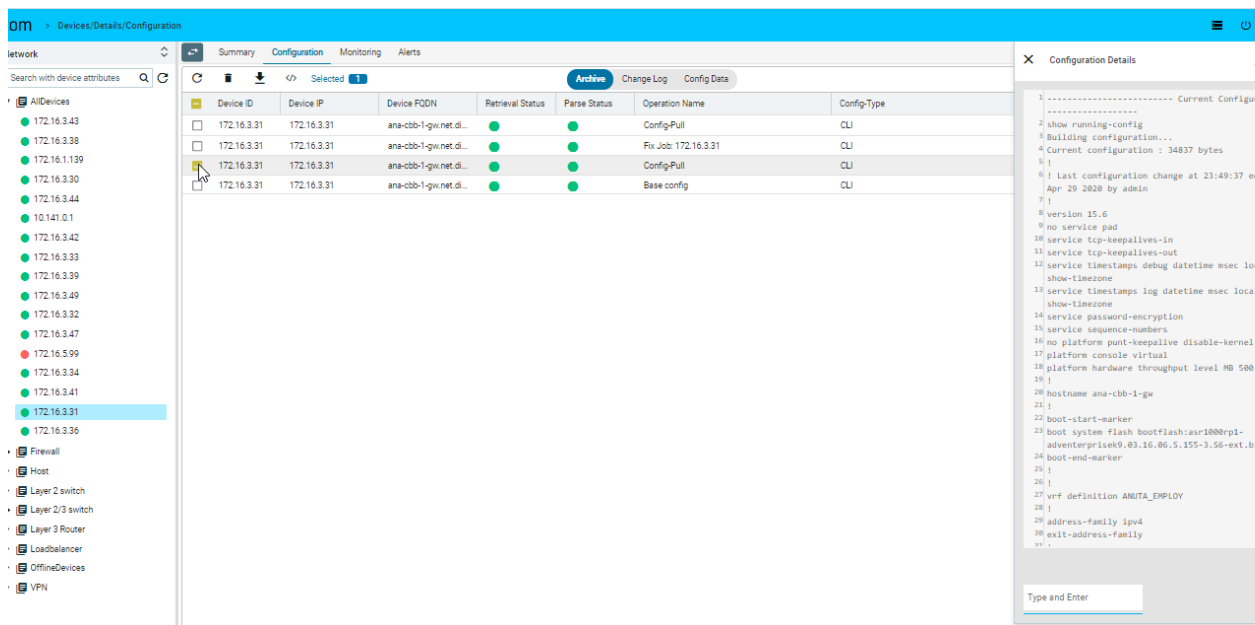
3. Search configuration grid using tags or other attributes
4. Select two configuration revisions
5. Click on **“Compare”** to launch configuration diff view



Configuration Tagging:

Configuration version can be tagged using user provided flags or tags. This can be used for filtering and comparison of configuration revisions.

1. To view Device(s) Configuration - Navigate to **Devices** > select a device(s)
2. Click on the **“Configuration > Archive”** Tab
3. Select an entry from the configuration revision grid
4. Click on **“Update Tags”**
5. Enter one or more tags in the lower right of the configuration details view



Configuration Change Log

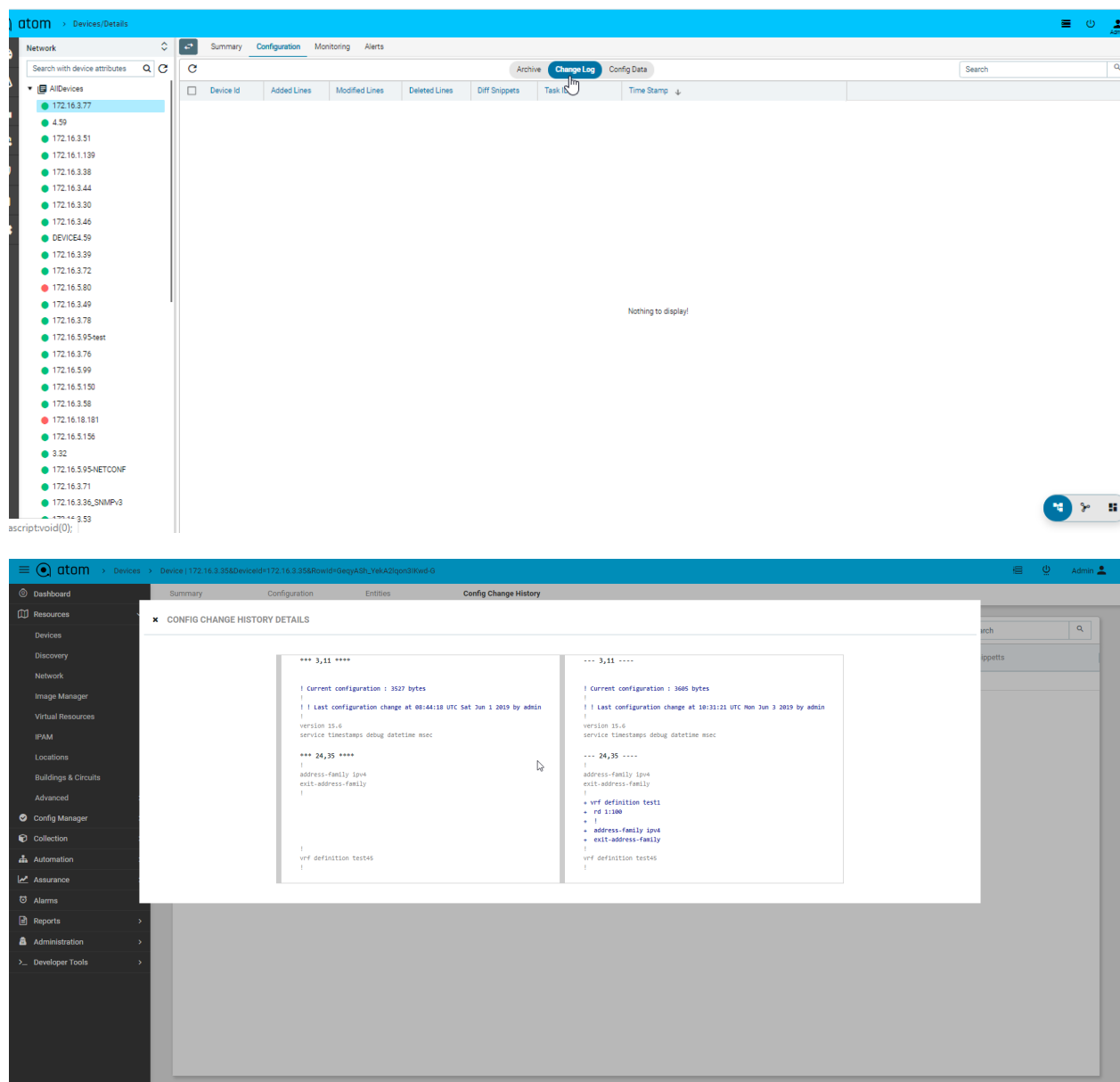
Configuration archive provides full comparison of device configuration changes across revisions. ATOM provides another view to see only config modifications only.

This can be enabled from Admin Settings.

1. Administrations > System> General settings> Admin settings
2. Edit “**Admin Settings**”
3. Set “**generate-config-inventory-event**” to true

Config change history for devices can be tracked as follows:

1. Navigate to **Devices (Tree View)** > select device(s)
2. Click on the “Configuration” Tab
3. Click on the “Change Log” Tab



Configuration Change Management - Create/Update/Delete

Configuration archive discussed in the “Configuration Management” section provides a Read-Only view of Device CLI configuration. Additionally, ATOM provides Model driven configuration for create, update & delete. This includes the following:

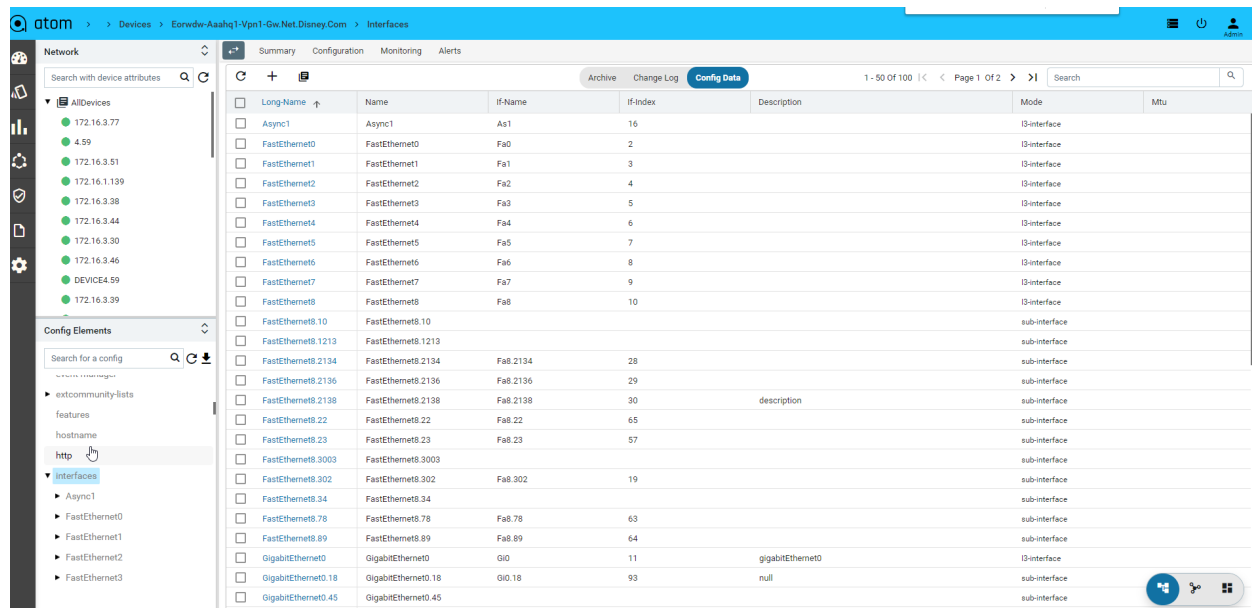
1. Discovery of Device configuration
2. Show a tree view of the configuration
3. Create/Edit/Delete of Device configuration

Configuration Editing can be done from “**Config Data**” view:

1. To view Device Configuration - Navigate to **Devices** > select a device

2. Click on the **Configuration> Config Data** Tab
3. From the Tree view select a node and possible operations are shown on the right hand side

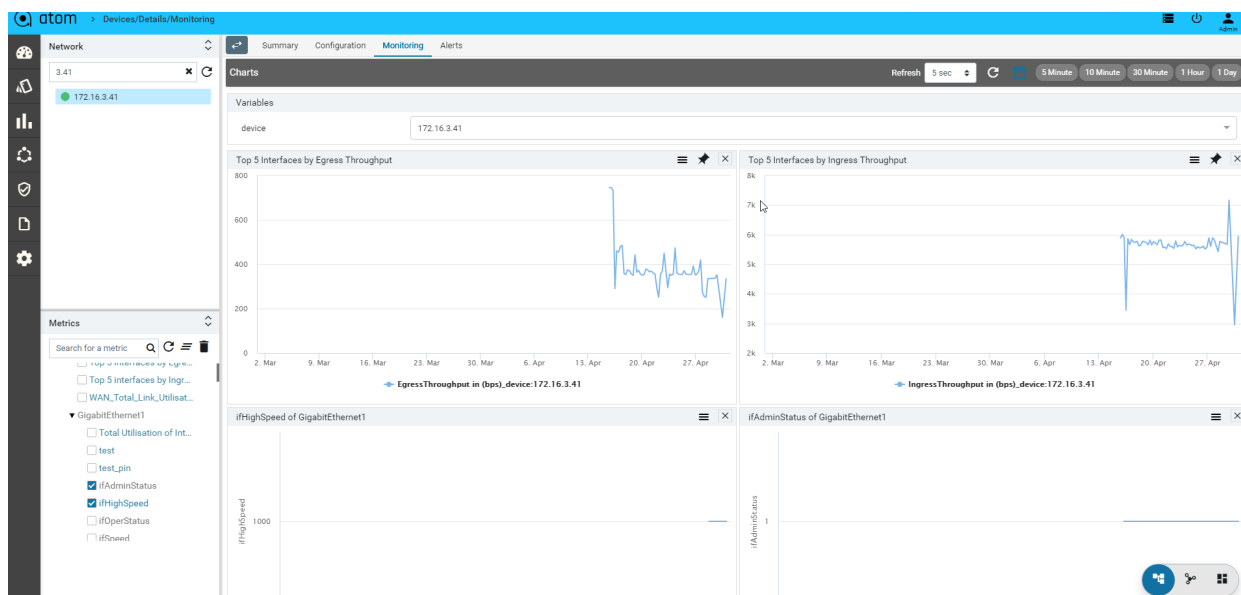
Note: Create/Edit/Delete from here will send configuration instructions to the device. ATOM should be set to interactive mode from the Administration page.



Device Inventory (SNMP)

All Device inventory collected through SNMP Collection job is shown in Entities view. Following provides guidance on

1. To view Device Configuration - Navigate to **Devices** > select a device
2. Click on the **“Monitoring”** Tab
3. Collected data will be shown under MIB-name



Adding Unmanaged devices

Some devices, with feature capabilities such as L2 only, L2 and L3 both, L3 only, can be manually added to the Devices table. Such devices are not managed by ATOM and it does neither generate configurations nor push any configurations on them. Multiple unmanaged devices can be on-boarded into the resource pool and each such device can be used during service instantiation.

To add an Unmanaged device, do the following:

1. Navigate to **Resources > Devices > Add Device**
2. In the **Create device** screen, select the Unmanaged option

Enter values in the following fields:

- **Host Name:** Enter a name for the device
- **Device Capability:** Select one or more capabilities from the available list.

For example, if you want the device to behave as a L3 device, choose **L3Router** from the list.

- **Device Type:** Select the category of the device that it belongs to.
- 3. Add network connections between the null device and it's peer device as follows:
 - **Source Interface:** Select the interface, on the null device, from which the network connection should originate.
 - **Peer Device:** Select the device, managed by ATOM, as the peer device.
 - **Peer Interface:** Select the interface on the peer device where the network connection should terminate.

Adding Dummy devices

In some scenarios, you may have to create devices for which configurations are created as a part of a service but are not pushed to any actual device. These logical entities are termed Dummy Devices and they do not have any real world counterparts with a pingable IP address.

Network Topology

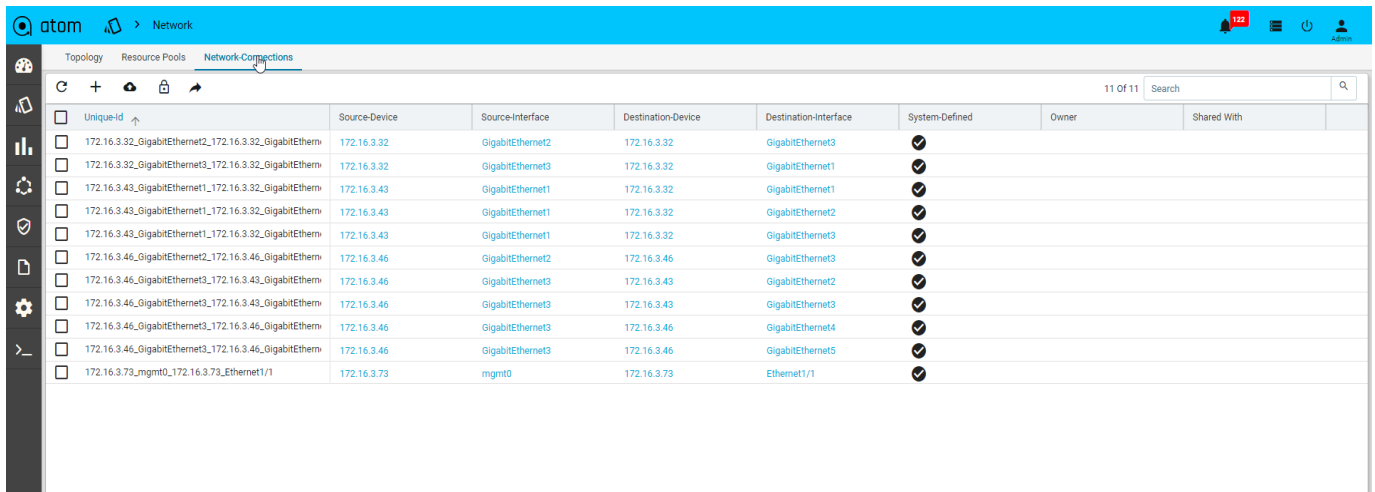
Network Connections

Network connectivity is discovered between devices using Layer 2 discovery protocols - CDP & LLDP. In cases where CDP/LLDP is not supported or enabled on the device, Network connections can be added Manually using Network connections .

NOTE: Network connections should be added manually between the devices that have LACP port channels configured on them.

To add a Network Connection, do the following:

1. Go to **Resource Manager > Network**
2. Click **Network Connections** and click **Add**.
3. In the Create Network Connection screen, enter the values in the following fields:
 - **Unique ID:** This is a system-generated ID for a network connection.
 - **Source Device:** Select a Device (origin of the network connection)
 - **Source Interface :** Enter a name for the interface on the source device
 - **Destination Device:** Select a Device (the end of the network connection)
 - **Destination Interface:** Enter a name for the interface on the destination device.A Network Connection is established between the interfaces of the source and the destination devices.



The screenshot shows the ATOM Network-Connections tab. The table lists network connections with columns: Unique-Id, Source-Device, Source-Interface, Destination-Device, Destination-Interface, System-Defined, Owner, and Shared With. There are 11 connections listed.

Unique-Id	Source-Device	Source-Interface	Destination-Device	Destination-Interface	System-Defined	Owner	Shared With
172.16.3.32_GigabitEthernet2_172.16.3.32_GigabitEthernet3	172.16.3.32	GigabitEthernet2	172.16.3.32	GigabitEthernet3	✓		
172.16.3.32_GigabitEthernet3_172.16.3.32_GigabitEthernet2	172.16.3.32	GigabitEthernet3	172.16.3.32	GigabitEthernet2	✓		
172.16.3.43_GigabitEthernet1_172.16.3.32_GigabitEthernet3	172.16.3.43	GigabitEthernet1	172.16.3.32	GigabitEthernet3	✓		
172.16.3.43_GigabitEthernet1_172.16.3.32_GigabitEthernet2	172.16.3.43	GigabitEthernet1	172.16.3.32	GigabitEthernet2	✓		
172.16.3.43_GigabitEthernet1_172.16.3.32_GigabitEthernet3	172.16.3.43	GigabitEthernet1	172.16.3.32	GigabitEthernet3	✓		
172.16.3.46_GigabitEthernet2_172.16.3.46_GigabitEthernet3	172.16.3.46	GigabitEthernet2	172.16.3.46	GigabitEthernet3	✓		
172.16.3.46_GigabitEthernet3_172.16.3.43_GigabitEthernet2	172.16.3.46	GigabitEthernet3	172.16.3.43	GigabitEthernet2	✓		
172.16.3.46_GigabitEthernet3_172.16.3.43_GigabitEthernet3	172.16.3.46	GigabitEthernet3	172.16.3.43	GigabitEthernet3	✓		
172.16.3.46_GigabitEthernet3_172.16.3.46_GigabitEthernet4	172.16.3.46	GigabitEthernet3	172.16.3.46	GigabitEthernet4	✓		
172.16.3.46_GigabitEthernet3_172.16.3.46_GigabitEthernet5	172.16.3.46	GigabitEthernet3	172.16.3.46	GigabitEthernet5	✓		
172.16.3.73_mgmt0_172.16.3.73_Ethernet1/1	172.16.3.73	mgmt0	172.16.3.73	Ethernet1/1	✓		

Network Topology

All the devices for each of which network connections are available are displayed in the topology view.

Resource Pools

A resource pool is a logical abstraction for flexible management of resources managed by ATOM. A resource pool can contain child resource pools and you can create a hierarchy of shared resources. The resource pools at a higher level are called parent resource pools. Users can create child resource pools of the parent resource pool or of any user-created child resource pool. Each child resource pool owns some of the parent's resources and can, in turn, have a hierarchy of child resource pools to represent successively smaller units of resources.

Resource pools allow you to delegate control over the resources of a host and by creating multiple resource pools as direct children of the host, you can delegate control of the resource pools to tenants or users within the organizations.

Using resource pools can yield the following benefits to the administrator:

- Flexible hierarchical organization
- Isolation between pools, sharing within pools
- Access control and delegation

Creating a Resource Pool

1. Navigate to **Resource Manager > Network > Resource Pools**
2. In the right pane, click the **Add Resource Pool** button to create a Resource Pool
3. In the **Create Resource Pool**, enter values in the fields are displayed: .

The screenshot shows the 'New Resource Pool' form in the ATOM interface. The form is titled 'New Resource Pool' and has a sidebar on the left with navigation icons. The form fields are as follows:

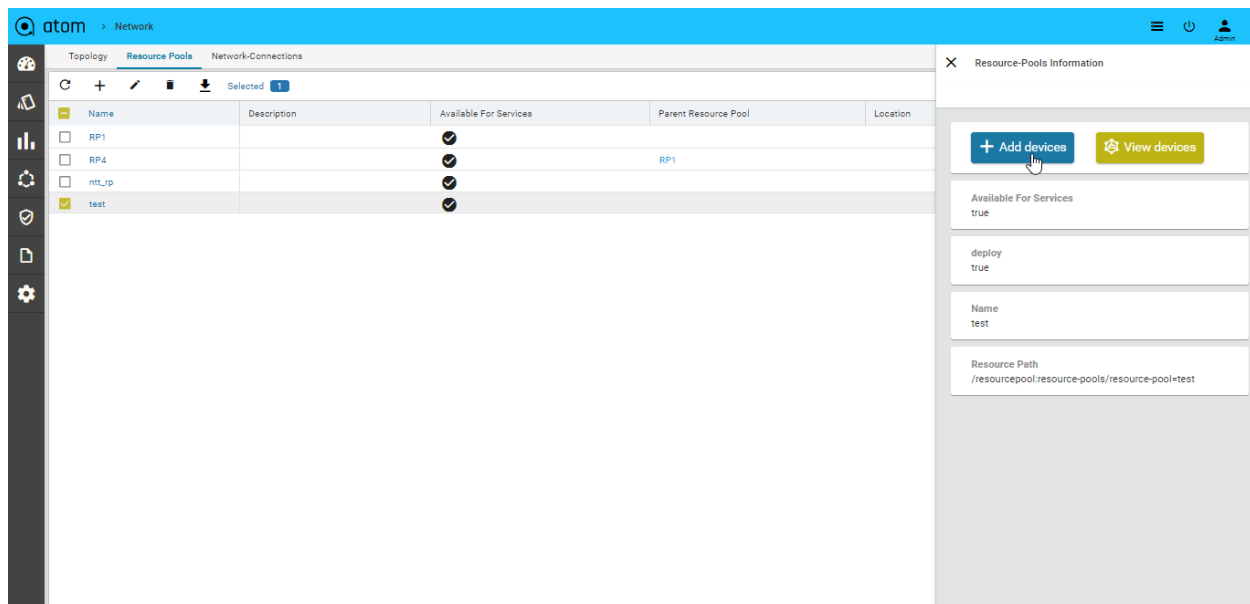
- Name:** A text input field containing 'test'. A tooltip above it says 'Name of the resource pool. This also acts as Unique Key. Allows AlphaNumeric...'. There is a clear (X) button and a green checkmark icon.
- Description:** A text input field with a placeholder 'Description of the resource pool' and a clear (X) button.
- Parent Resource Pool:** A dropdown menu with a clear (X) button.
- Location:** A dropdown menu with a clear (X) button.
- Available For Services:** A checkbox labeled 'Flag to indicate whether this resource-pool can be used for services. Typically s...'. It is checked.
- Deploy:** A checkbox labeled 'Set to false if the resource pool needs to be decommissioned'. It is checked.

- **Name:** Enter a name for the resource pool
- **Description:** Enter some descriptive text for the created resource pool
- **Available for Services:** Select this option if the resource pool can be used for creating services.
- **Parent Resource Pool:** Select a resource pool that should act as the parent for this resource pool that is being created.
- **Location:** Select the name of the site or the geographical location where this resource pool should be created. See the section, "Locations" for more information about creating Locations and Location types.
- **Deploy:** Select this option if the resource pool should be deployed or used in services.

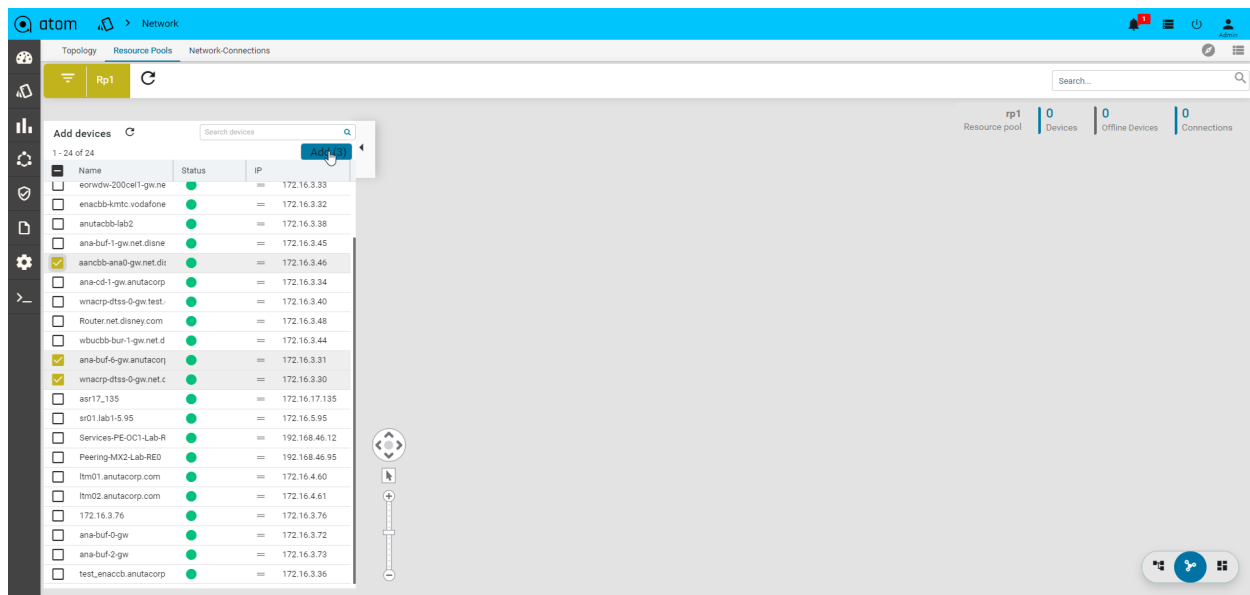
Adding Devices to a Resource Pool

1. Click the created resource pool to add the required devices to it.

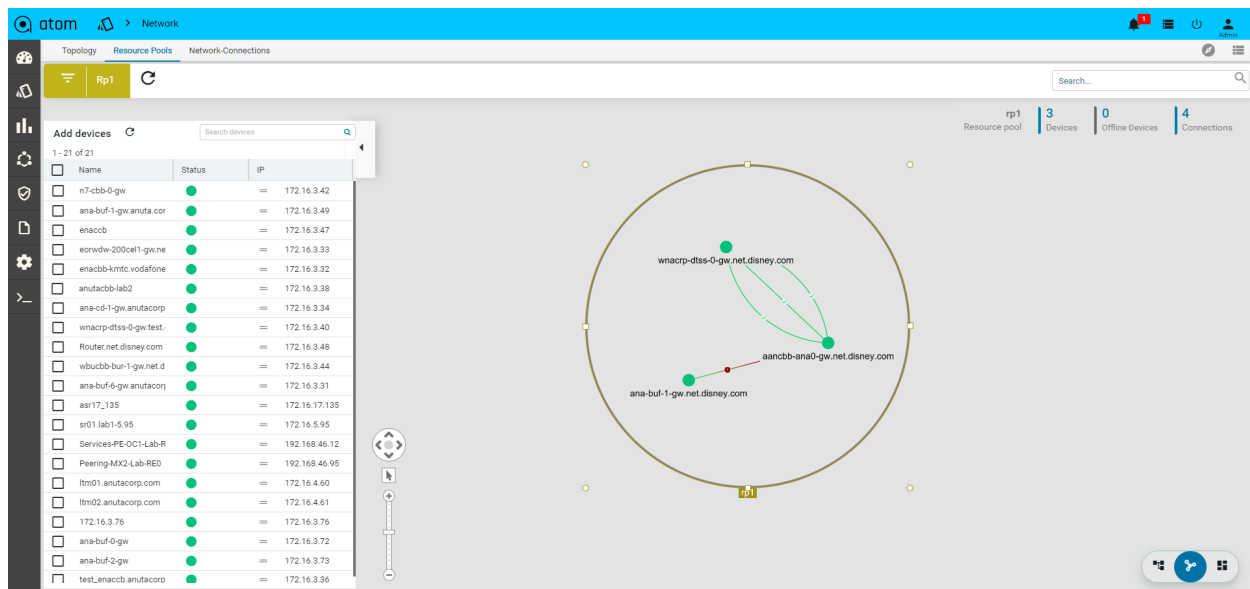
Select Resource pool > Add Devices



2. All the devices available in ATOM are displayed in the left pane.
3. Click **Add** to include the required devices in the resource pool



4. Select the device from the **Drag and Drop the devices** pane All the selected devices are now part of the resource pool created earlier.



Locations

Devices & Resource Pools can be attached to a Physical Location. Location tagging allows devices and resource pools to be visualized on a Geographical Map in topology view.

1. To Create/Edit a Location - **Resource Manager > Locations > Resources-Location** >click **Add Location**
2. In the **Create Location** screen, enter values in the fields described:
 - **Name:** Add the name of for the data center or the site that you want to create.
 - **Type:** Select from the pre-defined location types from the drop-down menu. (preferably select Site)
 - **Block:** Enter the name of the block where the location is situated
 - **Building:** Enter the name of the Building
 - **Street Number:** Enter the number of the street where the Building is located
 - **Latitude:** Enter the latitude of the site.
 - **Longitude:** Enter the longitude of the site.
 - **Street:** Enter the street name where the building is located.
 - **Country:** Select a country from a pre populated list available in ATOM
 - **City:** Select a specific city contained in the chosen country.
 - **State:** Enter the name of the State or province to which the city belongs.
 - **Zip Code:** Enter the zip code of the City where the Site is located.
 - **Parent Location:** Select one of the predefined locations (of the type, Region or Country) defined earlier.

For assigning the created Location to a Resource pool, refer to section, "Creating a Resource Pool".

After the successful allocation of the Resource Pool to the given Location, you can view it on the map. Select the created Resource Pool and click **View on Map**

Location Types

Add the types of the location that should be associated with a Location.

Navigate to **Resource Manager > Locations > Location Types**.

The default location types available in ATOM are Region, Country, and Data Center

Name	Owner	Shared With	Created-On	Created-By	Last-Modified-On	Last-Modified-By
Country	system	system.*	2021-03-03 05:23:52.591		2021-03-03 05:23:52.591	
Region	system	system.*	2021-03-03 05:23:52.591		2021-03-03 05:23:52.591	
Site	system	system.*	2021-03-03 05:23:52.591		2021-03-03 05:23:52.591	

IPAM

IP Address Pool Group

For effective management of IP addresses, you can arrange IP addresses as an ordered collection and use them while instantiating a service.

1. Navigate to **Resource Manager > IPAM > IP Address Pool Groups**
2. Click **Add IP Address Pool Group** in the right pane

3. In the **Create IP Address Pool Group** screen, enter values in the fields:
 - a. **Name:** Enter the name of the IP address pool group
 - b. **Label:** Enter the name of the label that describes the IP address pool group
 - c. Click **Add** to add IP Address Pools to be included in the IP Address Pool Group

IP Address Pools

A range of IP addresses can be assigned to a pool and associated with a resourcepool. All these IP addresses will be used during the instantiation of the service.

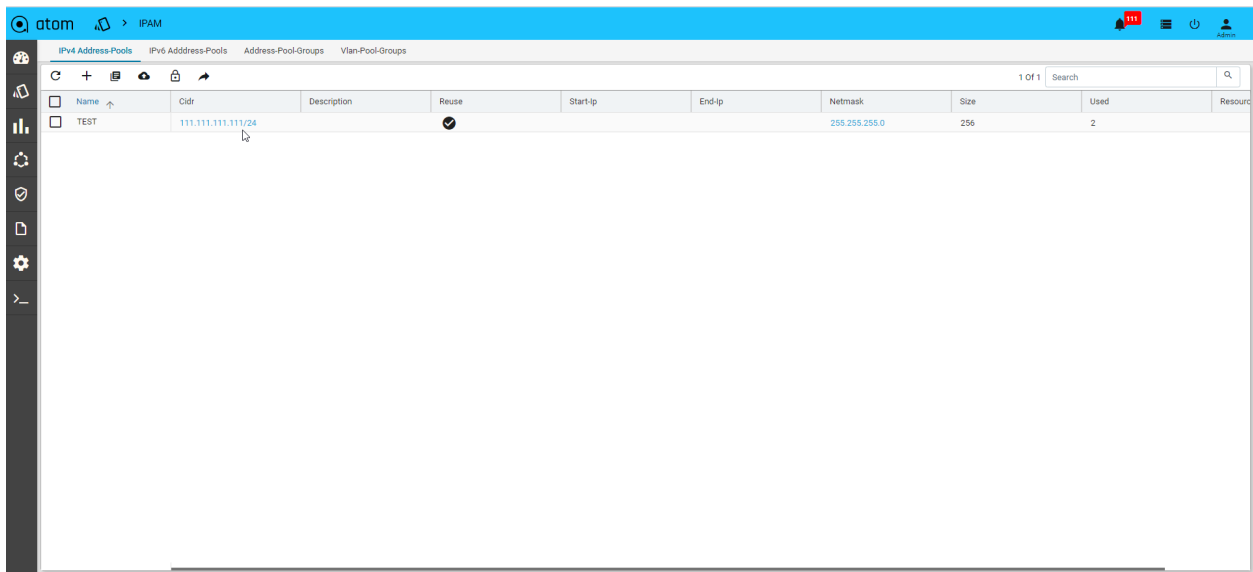
1. Navigate to **Resource Manager > IPAM > IP4- Address Pools**
2. Click **Add IP Address Pool** on the right pane and enter values in the following fields:
 - **Name:** Enter a unique name for the IP address pool
 - **CIDR:** Enter the CIDR (IP address followed by a slash and number)
 - **Description:** Enter the description for the created IP Address Pool
 - **Reuse:** Select this option if the IP addresses contained in this pool should be reused across different services.
 - **Start IP:** Enter the start IP address of the range of IP addresses
 - **End IP:** Enter the last IP address in the IP address range.
 - **Resource Pool:** Select the Resource pool to which these IP addresses should be assigned. All the services that are created in these Resource Pools will use these IP addresses.

Creating IP address entries

IP Address entries are the IP Address Pools that have been reserved for a service.

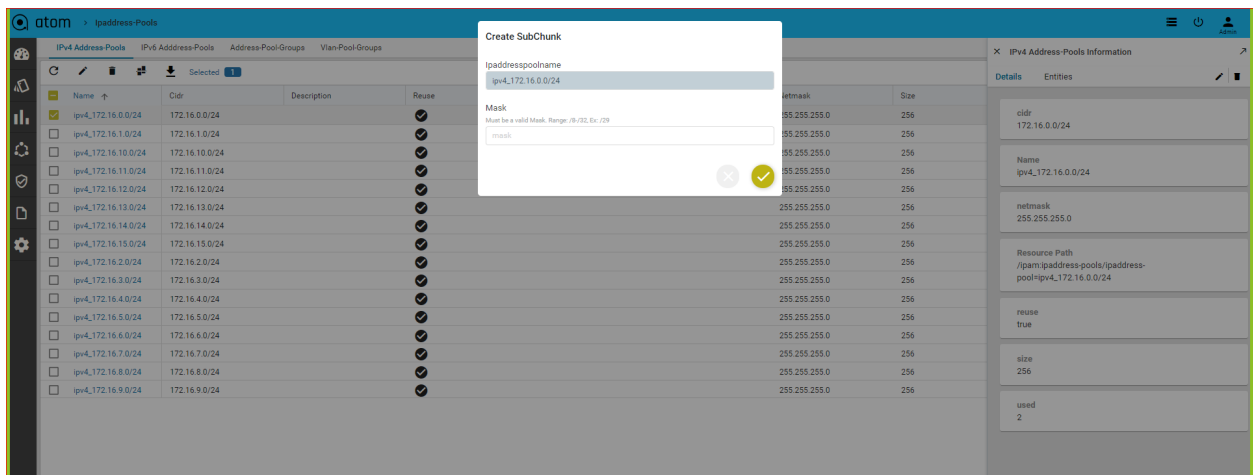
- Click **IP address pool > Action > IP address entries**

IPV4



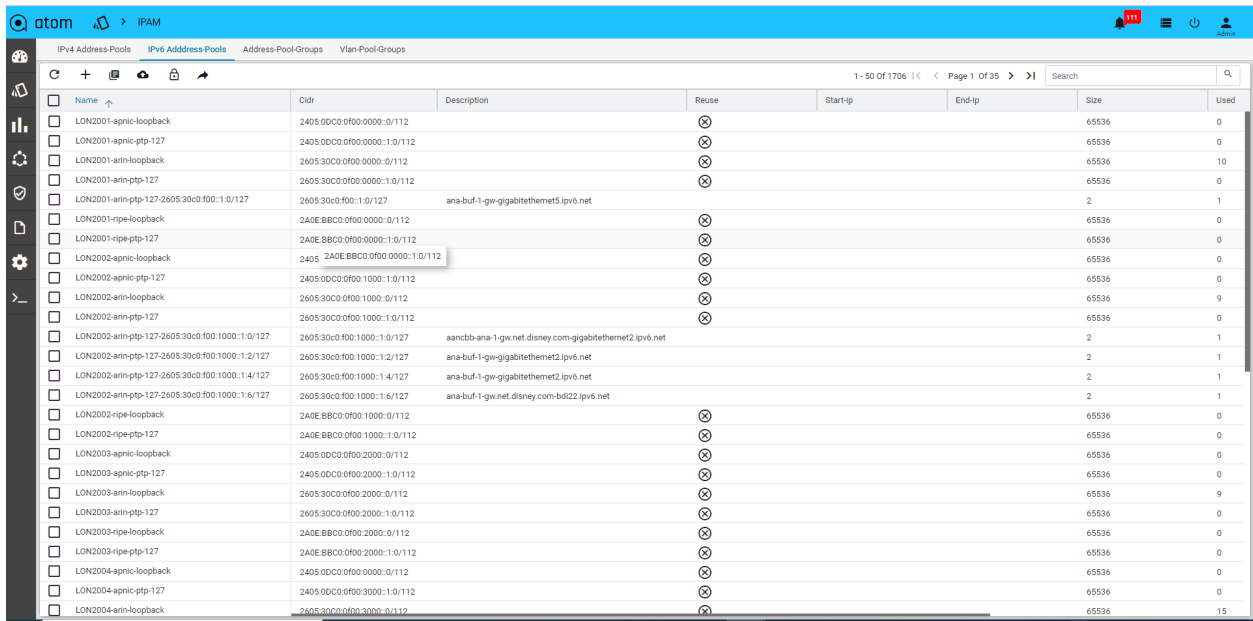
Creating Sub Chunks of the IP Address Pools:

The network contained in an IP address pool can be divided into two or more networks within it. The resulting sub chunks can be used for different services to be configured on a resource pool tied with the parent IP address pool.



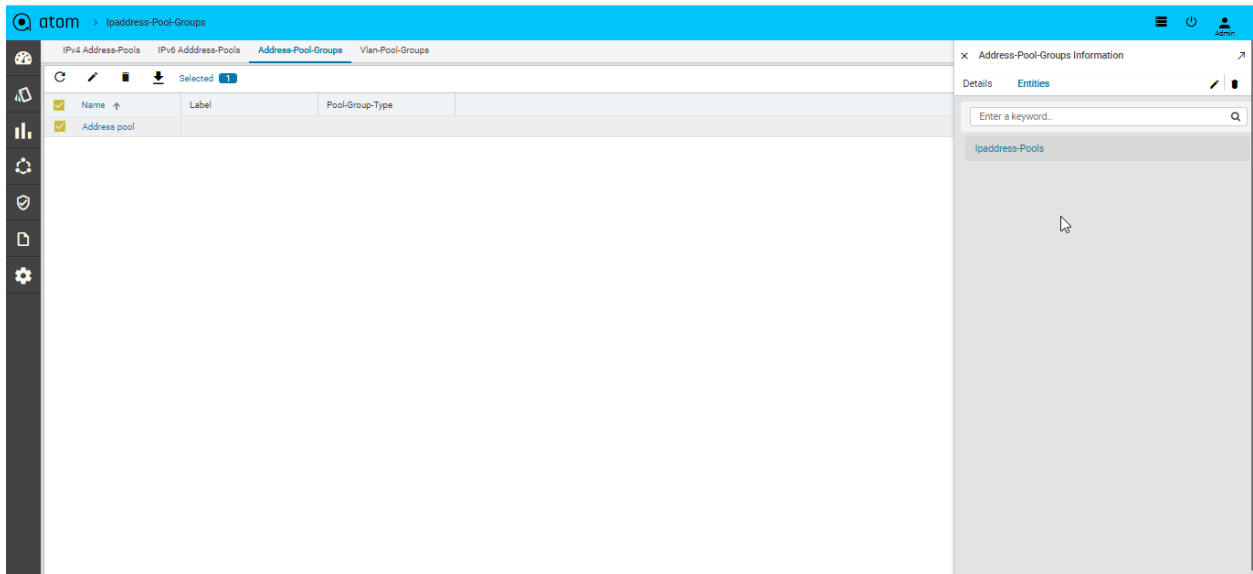
IPv6:

Resource Manager > IPAM > IP6- Address Pools



Name	Cidr	Description	Reuse	Start-ip	End-ip	Size	Used
LON2001-apric-loopback	2405.0D00.0F00.0000:0/112		⊗			65536	0
LON2001-apric-ptp-127	2405.0D00.0F00.0000:1.0/112		⊗			65536	0
LON2001-arin-loopback	2605.30C0.0F00.0000:0/112		⊗			65536	10
LON2001-arin-ptp-127	2605.30C0.0F00.0000:1.0/112		⊗			65536	0
LON2001-arin-ptp-127-2605.30c0.f00:1.0/127	2605.30c0.f00:1.0/127	ana-buf-1-gw-gigabitethernet5.ipv6.net				2	1
LON2001-ripe-loopback	2A0E.BB00.0F00.0000:0/112		⊗			65536	0
LON2001-ripe-ptp-127	2A0E.BB00.0F00.0000:1.0/112		⊗			65536	0
LON2002-apric-loopback	2405.2A0E.BB00.0F00.0000:1.0/112		⊗			65536	0
LON2002-apric-ptp-127	2405.0D00.0F00.0000:1.0/112		⊗			65536	0
LON2002-arin-loopback	2605.30C0.0F00.0000:0/112		⊗			65536	9
LON2002-arin-ptp-127	2605.30C0.0F00.0000:1.0/112		⊗			65536	0
LON2002-arin-ptp-127-2605.30c0.f00:1000:1.0/127	2605.30c0.f00:1000:1.0/127	aanbb-ana-1-gw.net.disney.com-gigabitethernet2.ipv6.net				2	1
LON2002-arin-ptp-127-2605.30c0.f00:1000:1.2/127	2605.30c0.f00:1000:1.2/127	ana-buf-1-gw-gigabitethernet2.ipv6.net				2	1
LON2002-arin-ptp-127-2605.30c0.f00:1000:1.4/127	2605.30c0.f00:1000:1.4/127	ana-buf-1-gw-gigabitethernet2.ipv6.net				2	1
LON2002-arin-ptp-127-2605.30c0.f00:1000:1.6/127	2605.30c0.f00:1000:1.6/127	ana-buf-1-gw.net.disney.com-bd122.ipv6.net				2	1
LON2002-ripe-loopback	2A0E.BB00.0F00.0000:0/112		⊗			65536	0
LON2002-ripe-ptp-127	2A0E.BB00.0F00.0000:1.0/112		⊗			65536	0
LON2003-apric-loopback	2405.0D00.0F00.2000:0/112		⊗			65536	0
LON2003-apric-ptp-127	2405.0D00.0F00.2000:1.0/112		⊗			65536	0
LON2003-arin-loopback	2605.30C0.0F00.2000:0/112		⊗			65536	9
LON2003-arin-ptp-127	2605.30C0.0F00.2000:1.0/112		⊗			65536	0
LON2003-ripe-loopback	2A0E.BB00.0F00.2000:0/112		⊗			65536	0
LON2003-ripe-ptp-127	2A0E.BB00.0F00.2000:1.0/112		⊗			65536	0
LON2004-apric-loopback	2405.0D00.0F00.0000:0/112		⊗			65536	0
LON2004-apric-ptp-127	2405.0D00.0F00.3000:1.0/112		⊗			65536	0
LON2004-arin-loopback	2605.30C0.0F00.3000:0/112		⊗			65536	15

Address-pool-Groups



Name	Label	Pool-Group-Type
Address pool		

Address-Pool-Groups Information

Details Entities

Enter a keyword...

Ipaddress-Pools

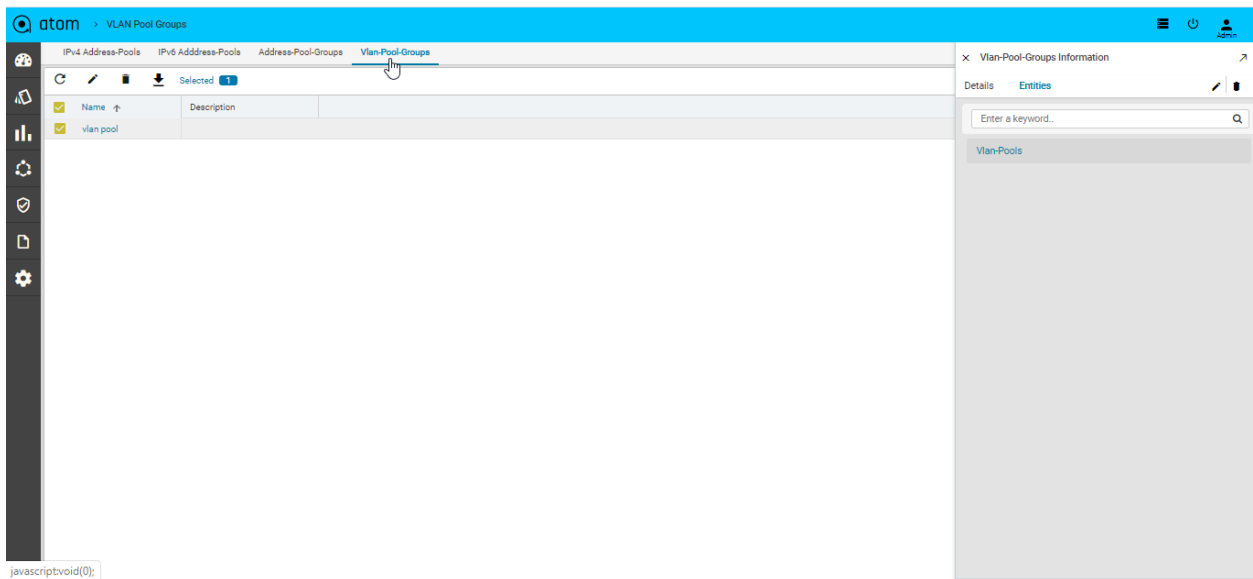
VLAN Groups

You can define VLAN groups and VLAN pools and define them as resource boundaries for a tenant in such a way that these VLAN Pools can be used during service instantiation on a resource pool.

Adding VLAN Groups

1. Navigate to **Resources > IPAM > VLAN Groups**

2. In the right pane, click **Add VLAN Pool Group**
3. In the **Create VLAN Pool Group** screen, enter values in the following fields:
 - Name: Enter a name for VLAN Group
 - Description:
4. Click **Actions** > **vlan pools** > **vlan pool** to create VLAN pools in the VLAN group:
5. Enter values in the following fields:
 - **Start VLAN**: Enter a number from the valid VLAN range. (1-4096)
 - **End VLAN**: Enter a number from the valid range (1-4096)
 - Click **Add** to add the required resource pools to the VLAN Pools
 - Click the **vlan pool** > click **Actions** to add allocated VLAN.

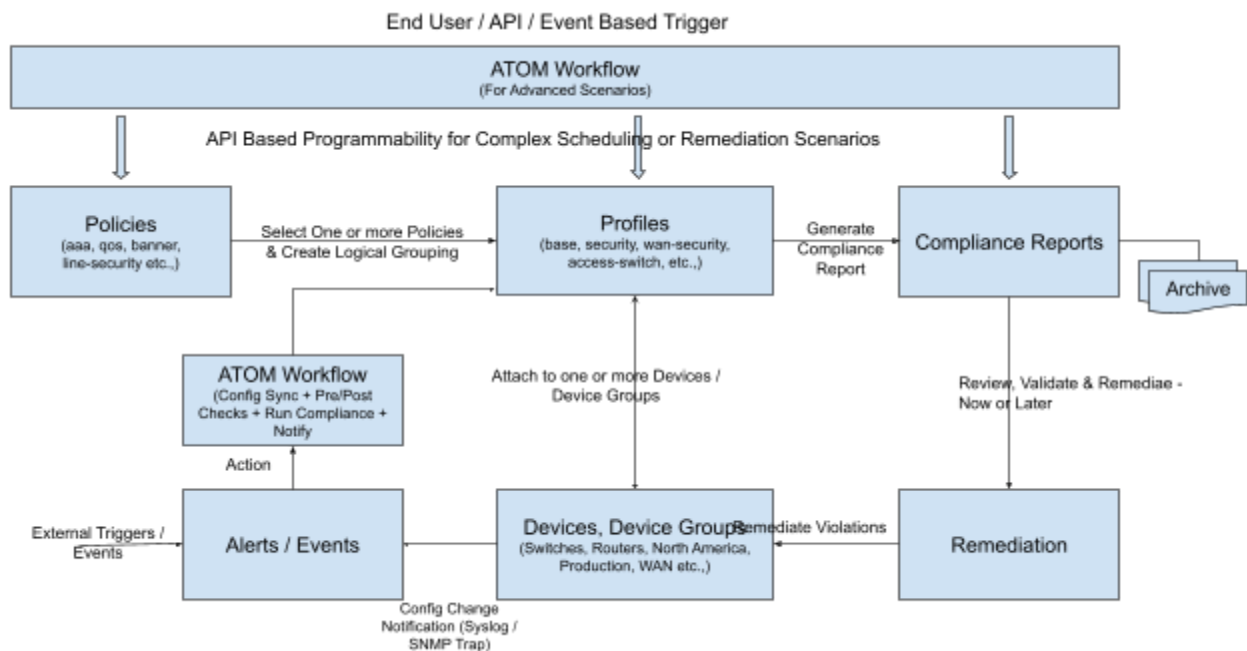


Configuration Compliance

Configuration Compliance feature allows users to Define & Enforce Configuration Compliance Standards. This is realized within ATOM using the following primitives.

1. Policies - Define Configuration standards & Remediation Policy by Device Family, Device Type, OS Type etc.,
2. Profiles - Group multiple policies and apply configuration standards on one or more devices
3. Reports - Comprehensive compliance reporting view at device level
4. Remediation - Fix Policy Violations on one or more devices

Following diagram summarizes the overall flow.



Configuration Compliance Can address the following scenarios:

1. Check If a particular configuration is present
2. Check If a particular configuration contains a given pattern / should NOT contain a given pattern.
3. Check If a particular configuration contains stale/unwanted configuration.
4. Check for User defined parameterized values (Dynamic inputs) in configuration.
5. Arithmetic checks to enforce thresholds on resource usage and capacity planning.
Example:- Per device Max 100 vrfs to be configured or 20 Vlans to be enabled or 10 bgp sessions per vrf
6. Group parameterized values to apply the policy. Example valid Values: 'Any', 'AnyEthernet', 'FastEthernet0/.*' etc.

7. Regex and Jinja2 Parsers In & Between conditions
8. Apply filters on configuration to categorize config blocks. Ex:- Access vs trunk based on link speed, Ports description satisfying regex classifiers, Ports which are admin up and contain IP.
9. Inventory checks for NETCONF/YANG parameters using XPath based expressions.
10. Parse as Blocks to split the entire running configuration into blocks and search for the condition match criteria value within each block.
11. Custom Block split definitions based on the start and end expressions you provide in the Block Start Expression and Block End Expression text boxes.
12. Evaluate each block against a set of conditions with individual actions/severities by using the Condition Scope as Previously Matched Blocks to parse.
13. Raise single violations for condition violation by any block or multiple violations per block of violation with individual remediation actions defined.

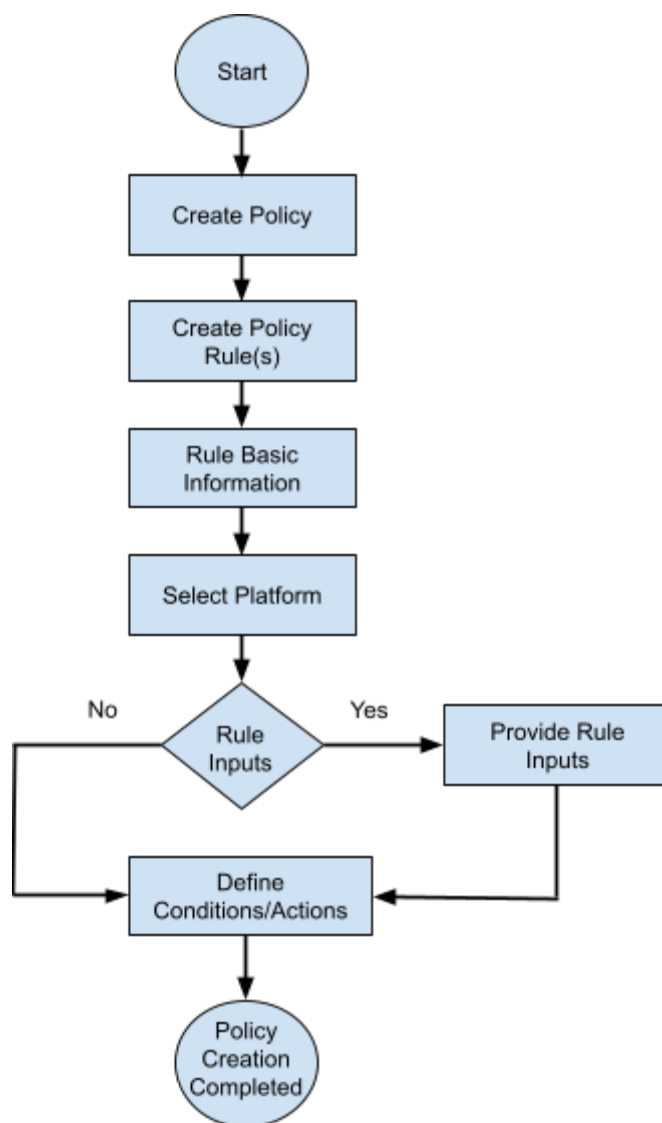
ATOM supports Configuration Compliance for the following Vendors:

- Cisco Systems
- Juniper Networks
- Fortinet
- Force10 Networks
- Brocade
- PaloAlto Networks
- Riverbed Technology
- F5 Networks

Policies

Compliance policy allows configuration standards to be defined in CLI format and YANG format(x-path or xml). Following provides a high level overview of a Policy:

- Policy is a collection of Rules
- A Rule contains one or more Conditions
- Condition describes
 - Expected Configuration. Configuration can be parameterized through Rule Variables.
 - Action to be taken on a condition evaluation includes CLI commands or Netconf XML RPC format to be used to remediate a violation.
- A Rule can be attached to one or More device platforms - Vendor, OS Type, Device Family, Device Type and OS Version



Use Cases

#	Configuration Standard Style	Example	Reference
1	Static Configuration	<p>Example: All Devices in Target Network should Contain a specific Domain Name</p> <p>Expected Configuration:</p> <pre>ip domain-name anutacorp.com</pre> <p>Fix Configuration:</p> <pre><<If missing, configure the above command>></pre>	Scenario1
	XPath Expression	<p>Xpath Expression:</p> <pre>Cisco-IOS-XR-native:native/ip/domain/name='anutacorp.com'</pre>	Scenario6
	XML Template Payload	<p>Template Payload:</p> <pre><native xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-native" > <ip> <domain> <name>net.disney.com</name> </domain> </ip> </native></pre>	Scenario11
2	Dynamic Configuration with User provided values	<p>Example: Devices in Target Network should have a specific Loopback interface - Loopback0 or Loopback1 based on user input.</p> <p>Expected Configuration:</p> <pre>interface {{ interface_name }}</pre> <p>Fix Configuration:</p> <pre><<If missing, Configure the specific Loopback interfaces>></pre>	Scenario3
	X-path	<p>Xpath Expression:</p>	Scenario9

	Expression	Cisco-IOS-XE-native:native/interface/Loopback/name='0' and Cisco-IOS-XE-native:native/interface/Loopback[name=0]/ip/address/primary/address='{{ lo0_ipv4addr }}' and Cisco-IOS-XE-native:native/interface/Loopback[name=0]/ip/address/primary/mask='255.255.255.255' and Cisco-IOS-XE-native:native/interface/Loopback[name=0]/ipv6/address/prefix-list/prefix='{{ lo0_ipv6addr }}'	
	XML Template Payload	Template Payload: <pre> <native xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-native" > <interface> <Loopback> <ip> <address> <primary> <address>10.100.99.98</address> <mask>255.255.255.255</mask> </primary> </address> </ip> <ipv6> <address> <prefix-list> <prefix>2605:30C0::3B/128</prefix> </prefix-list> </address> </ipv6> <name>0</name> </Loopback> </interface> </native> </pre>	Scenario13
3	Configuration with Patterns, Wildcards, etc.that require Regular expressions	<p>Example: All the VTY lines should have specific exec-timeout and session-timeout configured.</p> <p>Expected Configuration:</p> <pre> line vty (.*) session-timeout 10 </pre>	Scenario4

		<pre>exec-timeout 10 0</pre> <p>Fix Configuration: <<If missing, Configure the timeouts under all matching VTY lines>></p>	
4	Configuration with sub-modes	<p>Example: The physical Interface should not be shut down and show be in auto-negotiation mode</p> <p>Expected Configuration:</p> <pre>interface {{ interface_name }} no shutdown negotiation auto</pre> <p>Fix Configuration: <<If missing, Configure the above commands for one or more interfaces>></p>	Scenario3
5	Removing unwanted extra configuration	<p>Example: Finding the Devices having extra ntp-server addresses configured and removing those other than expected server addresses.</p> <p>Expected Configuration:</p> <pre>ntp-server 10.1.1.1</pre> <p>Fix Configuration: << Configure above ntp-server if not found. Remove any ntp server other than 10.1.1.1 >></p>	Scenario2
6	Advanced: Presence of an entity value from one block in another	<p>Example: Finding the devices in the network which doesn't contain the OSPF router-id configured as per loopback0 ip address.</p> <p>Expected Configuration:</p> <pre>interface Loopback0 ip address 45.45.45.5 255.255.255.255 ! router ospf 100 router-id 45.45.45.5</pre> <p>Fix Configuration:</p> <pre>router ospf 100</pre>	Scenario5

		router-id 45.45.45.5	
--	--	----------------------	--

Scenario 1: IP Domain Name

Scenario: Network Devices must have domain name configured. In this example we are looking for the domain name as **anutacorp.com** across all devices in the lab.

Platform:

Cisco IOS-XE

Expected Configuration:

```
ip domain-name anutacorp.com
```

Fix-CLI Configuration:

```
ip domain-name anutacorp.com
```

Follow the steps below to configure Compliance Policy for the above scenario.

1. Configure Policy

Steps:

- Navigate to Resource Manager > Config Compliance -> Policies
- Click '+' to create new Policy and provide the following information
 - Policy Name
 - Description

Add Policy

• mandatory information

Policy Name •

Policy name. Can contain AlphaNumerics and underscore characters o...

IP_Domain_Name

Description

Description of the policy

Check whether ip domain name is present in the device or not

Create Policy

2. Configure Rules

One or more rules can be configured to express configuration standards. Based on the complexity of the scenario, configuration standards can be broken up into more than one condition.

Steps:

- Navigate to Resource Manager > Config Compliance -> Policies
- Create/Select a Policy
- Click '+' to create new Rule
- ATOM opens up new wizard as shown below

Add Rule



- Rule has four components
 - Basic Information
 - Platform Selection
 - Rule Variables
 - Conditions & Actions

Basic Information

Provide basic information as described below. Information provided here is for documentation purposes only.

Rule Name: Provide any Name

Description: Brief explanation of the configuration evaluation that the rule is going to perform.

Impact: If the device configuration does not meet the rule or rules in the policy, type it in the Impact field.

Suggested Fix: Using which non-compliance can be corrected and device returns to a state of Compliance.

Edit Policy | IP_Domain_Name

Add Rule

Basic Information Platform Selection

Rule Name ●

Check_Ip_Domain_Name

Description

Check domain name for Cisco devices

Impact

If domain name is not present in the device. Device will be non-compliant.

Suggested fix

ip domain name anutacorp.com

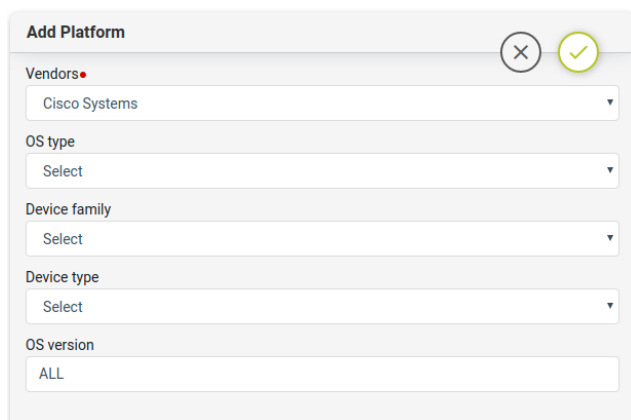
Create Rule

Platform Selection

Rules contain configuration standards expressed in CLI Configuration format. Configuration standard can be at Vendor level, Device Type, Device Platform, OS Type or OS Version.

Steps:

- Navigate to Config Manager > Config Compliance -> Policies -> Rules
- Create/Select a Rule & Provide the following information
 - Vendor
 - OS type
 - Device Family
 - Device Type
 - OS Version

A dialog box titled "Add Platform" with a close button (X) and a confirm button (checkmark). It contains several dropdown menus: "Vendors" with "Cisco Systems" selected, "OS type" with "Select", "Device family" with "Select", "Device type" with "Select", and "OS version" with "ALL".

Add Platform

Vendors •
Cisco Systems

OS type
Select

Device family
Select

Device type
Select

OS version
ALL

Note: Platform Selection will be used during Policy execution. Devices that don't match the above criteria are skipped.

Note: It's not common to have more than one Platform

Rule Variables

Rule variables allow configuration to be parameterized.

Steps:

- Navigate to Resource Manager > Config Compliance -> Policies -> Rules
- Create/Select a Rule Variables
 - Key - Provide unique name to identify rule variable
 - Description - Describe rule-input configuration
 - Default Value - Default value. Can be overridden during Policy execution time

Conditions and Actions

Expected configuration & actions to be taken when violations are detected are specified in the *Conditions & Actions* section.

Based on the complexity of the scenario, configuration standards can be broken up into more than one condition.

Steps:

- Navigate to Resource Manager > Config Compliance -> Policies -> Rules
- Create/Select Conditions & Actions
 - Condition Details - Described Below
 - Action Details - Described Below

Condition Details

Condition section provides users to specify the expected configuration and various options on how to match the expected configuration including option to identify sub mode configuration blocks.

Condition Name: Name of the Condition

Sequence Number : Order of the condition execution.

Scope Details

Condition scope details: Scope could be either full configuration copy or configuration matched in prior condition.

- *Configuration* - Full Configuration
- *Previously_Matched_Blocks* - Subset of configuration matched by prior condition

Block Options

Start Expression - Regular expression indicating the start of the sub-block.

End Expression - Regular expression indicating the end of the sub-block.

Condition Match Criteria

Operator:

MATCHES_THE_EXPRESSION - Checks whether the condition value exactly matches with device configuration or not.

DOESNOT_MATCHES_THE_EXPRESSION - Checks whether the condition value does not match with the device configuration or not.

CONTAINS_STRING - Checks whether the device configuration contains condition value config or not.

Rule-Pass-Criteria:

All_SubBlocks - Checks whether the condition value matches in all the blocks or not.

Any_SubBlock - Checks whether the condition value matches in any of the blocks or not.

Value: Value field accepts Configuration Standard as CLI Configuration. Following types of configuration can be provided:

- Static Configuration
- Dynamic/Parameterized Configuration
- Configuration with Regular Expressions
- Configuration coupled with Jinja2 Templating

Note: For some Vendor Configurations like Cisco IOS-Style, whitespace in command prefix is mandatory to identify commands at sub-mode level.

For Scenario 1 - Provide value as **ip domain-name anutacorp.com** to search for a given domain name in the running configuration.

Test Config

Based on the complexity of the configuration standard, Value may be complex and may need to build up iteratively. Test Config utility helps the CLI configuration condition to be validated against Test Configuration.

Steps:

- Navigate to Resource Manager > Config Compliance -> Policies -> Rules ->
- Create/Select Conditions & Actions
- Click “Launch Test Config” will launch a form to Test Condition

Test Configuration and Results

Condition Match Operator:
 MATCHES_THE_EXPRESSION
 Value
 ip domain-name anutacorp.com

Rule Variables
 Please enter the rule variables in Key: Value => format Ex: Key1: Value1
 Key1: Value1
 Key2: Value2
 Key3: Value3

Test Configuration
 ip domain-name anutacorp.com

Test Results

```
{
  "compliance-policies": {
    "highest-severity": "",
    "rule-violation-count": 0,
    "compliance-status": "compliant",
    "compliant-rules-output": {
      "violated-conditions": "",
      "device-compliance-condition-output": {
        "block-start-unmatched-content": "<![CDATA[<br>]",
        "block-start-condition-search-output": "<![CDATA[<br> \\block_start_matched_contents\\': [\\n]]>",
        "condition-search-output": "<![CDATA[<br> \\matched_contents\\': [\\n]]>",
        "total-block-count": 1,
        "aggregated-condition-output": "<![CDATA[<br> \\condition_contents\\': [\\n \\condition_id\\': null\\n \\block_start_matched_content\\': null\\n \\block_start_unmatched_content\\': null\\n \\unmatched_content\\': null\\n \\matched_content\\': null\\n ]\\n]]>"
      }
    }
  }
}
```

Test

Condition Match Operator:

MATCHES_THE_EXPRESSION
 DOESNOT_MATCHES_THE_EXPRESSION
 CONTAINS_STRING

Value: Sample configuration to be tested. Value will be shown from the Condition Details. Value can be further refined

Note: Any Edits to Value will reflect in the Condition Details -> Value and Vice-versa.

Test Configuration: Sample device configuration

Rule Variables: The rule variables created in the rule will be shown here with default values. Values can be modified.

Note: Any Edits to Values will not be reflected in the Rule Variable default values provided in Rule Variables Section.

Test Results: Based on Condition Match Operator test results will be shown on the right hand side

Action Details

Action can be taken after Condition evaluation. Condition can result in either a “Match” or “Non-Match”. Depending on the scenario one or both criteria may apply.

Select Match Action - This option is applicable when Condition evaluates to a Match

Select action:

Continue - continue execution to next condition

Donot_raise_violation - skip execution and don't raise violation

Raise_violation_and_continue - raise violation and continue execution to next conditions

Raise_violation - raise violation and skip execution

For Scenario 1, no action needs to be taken during a match condition, so select **continue** as action.

Violation severity:

LOW

MEDIUM

HIGH

CRITICAL

Violation message type:

Default_violation_message

User_defined_violation_message

Derive fix cli commands:

Use_unmatched_block - unmatched config from the block

Use_matched_block - matched config from the block

Use_complete_block - total block config

Select Non-Match Action

Select action:

Continue

Don't raise violation

Raise violation and continue

Raise violation

For Scenario 1, Action is required when Condition is not matched. Select **Raise violation and continue**.

Violation severity:

LOW

MEDIUM

HIGH
CRITICAL

Violation message type:

Default_violation_message
User_defined_violation_message

Fix CLI: Provide the CLI Configuration to be used for remediation. Fix CLI can be either provided here or derived.

Option - 1 - Explicit Remediation / Fix CLI

For Scenario 1, Provide “ip domain-name anutacorp.com” in Fix CLI.

The screenshot shows the 'Conditions and Actions' configuration interface. The 'Action Details' tab is selected. The 'Select Match Action' dropdown is set to 'continue'. The 'Select Non-Match Action' dropdown is set to 'Raise_violation'. The 'Violation severity' dropdown is set to 'CRITICAL'. The 'Violation message type' dropdown is set to 'Default_violation_message'. The 'Fix CLI' text area contains the text 'ip domain-name anutacorp.com'. The 'Derive fix cli commands' dropdown is set to 'Select'.

Option - 2 - Remediation Commands can be derived from Condition evaluation.

Derive fix cli commands:

Options below:

Use_unmatched_block
Use_matched_block
Use_complete_block

For Scenario 1, Select “Use_unmatched_block”. Since this is non-match Action, unmatched_block will be Condition Details->Value and can be used as Fix CLI.

Scenario 2: NTP Server configuration check

Scenario:

1. All devices in the network should contain the designated ntp server.
2. Remove all other ntp servers
3. In this example
 - a. Expected ntp-server = 10.0.0.1

Platform:

Cisco IOS-XE

Expected Configuration:

```
ntp server 10.0.0.1
```

Fix-CLI Configuration:

```
ntp server 10.0.0.1
```

```
<<Remove Any Other ntp server other than 10.0.0.1>>
```

This use case uses regular expressions and contains two conditions.

1. Condition-1 - Check for expected config & if not found remediate using Fix CLI.

Fix-cli Configuration :

```
ntp server 10.0.0.1
```

2. Condition 2- Check for unwanted ntp-servers and remove them.

Fix-cli Configuration :

```
no ntp server 10.0.0.2 //Derived
```

```
no ntp server 10.0.0.3 //Derived
```

Steps:

- Navigate to Resource Manager > Config Compliance -> Policies
- Click '+' to create new Policy and provide the following information
 - Policy Name - NTP_Common_Peer_Configuration
 - Description
- Select the Policy and Click '+' to create new Rule
 - Rule Name - Check_NTP_Common_Peer_Configuration
- Navigate to Config Manager > Config Compliance -> Policies -> Rules
- Select a Rule & Provide the following information
 - Vendor - Cisco Systems
 - OS type - IOSXE
 - Device Family - ALL
 - Device Type - ALL
 - OS Version - ALL
- Rule variables are not required for this scenario.
- Now fill the Conditions and Actions

Condition1

The Verify_NTP condition will check if the NTP server config is present in the device or not.

Conditions and Actions

Condition Details | Action Details

Condition Name •
Condition Name, can contain Alphanumerics, underscore, space and hyphen ...
Verify_NTP

Sequence Number •
Sequence Number controls the order of execution of the Conditions.
1

Scope Details
Condition scope details
Configuration

Block Options
Start Expression

Condition Match Criteria
Operator
MATCHES_THE_EXPRESSION
Rule-pass-criteria
All_SubBlocks

Value •
ntp server 10.0.0.1

Launch Test Config

Here Non-Match Action can be done either using the commands in Fix CLI or using the Derive fix cli commands.

- Using the Fix CLI user needs to provide the configuration commands manually.
- Using the Derive Fix CLI Commands user needs to select the use_unmatched_block as shown below.

The screenshot shows a configuration window with two tabs: 'Rule Variables' and 'Conditions and Actions'. The 'Conditions and Actions' tab is active, and the 'Action Details' sub-tab is selected. The window is divided into two main sections: 'Select Match Action' and 'Select Non-Match Action'. In the 'Select Match Action' section, the 'Select action' dropdown is set to 'continue'. In the 'Select Non-Match Action' section, the 'Select action' dropdown is set to 'Raise_violation_and_continue'. Below this, the 'Violation severity' dropdown is set to 'CRITICAL', and the 'Violation message type' dropdown is set to 'Default_violation_message'. The 'Fix CLI' field is a large empty text area. At the bottom, the 'Derive fix cli commands' dropdown is set to 'use-unmatched-block'. There are close (X) and confirm (checkmark) buttons in the top right corner.

Here on Match Action it will Continue and on Non-Match Action the Derive fix cli commands uses the use-unmatched-block to remediate the device.

Condition2

Remove_NTP_Extra_Config condition will use the regex to match and capture the extra NTP server ip configured in the device other than the expected ip.

Conditions and Actions

Condition Details

Action Details

Condition Name

Condition Name, can contain Alphanumerics, underscore, space and hyphen ...

Remove_NTP_Extra_Config

Sequence Number

Sequence Number controls the order of execution of the Conditions.

1

Scope Details

Condition scope details

Configuration

Block Options

Start Expression

Condition Match Criteria

Operator

MATCHES_THE_EXPRESSION

Rule-pass-criteria

All_SubBlocks

Value

ntp server (?10.0.0.1)(\d+.\d+.\d+.\d+)

Launch Test Config

The extra NTP server ip captured will be stored in the backend data structure which is shown in the Test Results tab.

Test Configuration and Results

Condition Match Operator:

MATCHES_THE_EXPRESSION

Value

ntp server (?10.0.0.1)(\d+.\d+.\d+.\d+)

Rule Variables

Please enter the rule variables in Key: Value => Format Ex: Key1: Value1

Key1: Value1

Key2: Value2

Key3: Value3

Test Configuration

ntp server 10.0.0.1

ntp server 10.0.0.2

ntp server 10.0.0.3

Test Results

```

"compliant-rules-output": {
  "violated-conditions": "",
  "device-compliance-condition-output": {
    "block-start-unmatched-content": "<[CDATA]]>",
    "block-start-condition-search-output": "<[CDATA[(\n \bblock_start_matched_contents\": [{(n \bgroups\": [{(n \bindex\": 1,(n \bgrep_content\": \"10.0.0.2\"(n \bgrep_group\": \"1\" )}] )}]>",
    "condition-search-output": "<[CDATA[(\n \bmatched_contents\": [{(n \bgroups\": [{(n \bindex\": 1,(n \bgrep_content\": \"10.0.0.2\"(n \bgrep_group\": \"1\" )}] )}]>",
    "total-block-count": 2,
    "aggregated-condition-output": "<[CDATA[(\n \bcondition_id\": null,(n \bblock_start_matched_content\": null,(n \bblock_start_unmatched_content\": null,(n \bmatched_content\": null,(n \b )}]>",
    "template-substituted-content": "<[CDATA[ntp server (?10.0.0.1)(\d+.\d+.\d+.\d+)]>"
  }

```

Test

The captured data will be stored in the condition-search-output

48
70

65

Edit Policy | NTP_Common_Peer_Configuration

Edit Rule | Check_NTP

Basic Information | Platform Selection | Rule Variables | Conditions and Actions

Conditions and Actions

Condition Name

Verify_NTP

Remove_NTP_Extra_Con...

Condition Details

Action Details

Select Match Action

Select action

Raise_violation

Violation severity

CRITICAL

Violation message type

Default_violation_message

Fix CLI

```
{% for content in matched_contents -%}
{% for group in content["groups"] -%}
no ntp server {{ group["group_content"] }}
{% endfor %}
{% endfor %}
```

Derive fix cli commands

Select

Select Non-Match Action

Select action

continue

Finally if different NTP servers are present on the device, for Non-Compliant device Fix CLI will show up as below

```
ntp server 10.0.0.1
no ntp server 10.0.0.2
no ntp server 10.0.0.3
```

Scenario 3: Interface configuration check

Scenario: All devices in the network should have a specific interface in no shutdown state with auto negotiation enabled. The interface block can have extra configuration commands under it but should be in no shutdown state and auto negotiation enabled.

Platform:

Cisco IOS-XE

Expected Configuration:

```
interface {{ interface_name }}
no shutdown
negotiation auto
```

Fix-CLI Configuration:

```
interface {{ interface_name }}
  no shutdown
  negotiation auto
```

This use case is an interface block configuration having rule variables. In this use case as no shutdown is generally not visible on device running config, we will check whether the interface is in shutdown or not. If shutdown it will remediate to no shutdown.

Steps:

- Navigate to Resource Manager > Config Compliance -> Policies
- Click '+' to create new Policy and provide the following information
 - Policy Name - Interfaces
 - Description
- Select the Policy and Click '+' to create new Rule
 - Rule Name - Check_Interfaces
- Navigate to Config Manager > Config Compliance -> Policies -> Rules
- Select a Rule & Provide the following information
 - Vendor - Cisco Systems
 - OS type - IOSXE
 - Device Family - ALL
 - Device Type - ALL
 - OS Version - ALL
- Now create the Rule variables for this scenario.

Edit Policy | [Interfaces](#)

Edit Rule | [Check_Interfaces](#)

Basic Information
Platform Selection
Rule Variables
Conditions and Actions

Rule Variables

<input type="checkbox"/> Key	Description	Default Value
<input type="checkbox"/> interface_name		GigabitEthernet5

Conditions and Actions

Condition Details

Action Details

Condition Name●

Condition Name, can contain Alphanumerics, underscore, space and hyphen ...

Verify_Interface

Sequence Number●

Sequence Number controls the order of execution of the Conditions.

1

Scope Details

Condition scope details

Configuration

Block Options

Start Expression

Condition Match Criteria

Operator

CONTAINS_STRING

Rule-pass-criteria

Any_SubBlock

Value●

interface {{ interface_name }}
no shutdown
negotiation auto

Launch Test Config

In the policy we will have a jinja rule variable `interface_name`. Here `Verify_Interfaces` condition will check if the interface block config is present in the device or not and under that interface if `no shutdown` and `negotiation auto` is present.

For Scenario3 Condition Match Operator as `CONTAINS_STRING` will check whether the device configuration contains condition value or not. If device configuration contains value, the result will be Compliant, else Non-Compliant.

Test Configuration and Results

Condition Match Operator:

CONTAINS_STRING

Value

```
interface {{ interface_name }}
shutdown
negotiation auto
```

Test Configuration

```
interface GigabitEthernet5
description TEST
ip address 94.1.1.9 255.255.255.252
shutdown
negotiation auto
```

Rule Variables

Please enter the rule variables in Key: Value or Format: Key: Value1

```
interface_name: GigabitEthernet5
```

Test Results

```
{
  "compliance-policies": {
    "highest-severity": "",
    "rule-violation-count": 0,
    "compliance-status": "compliant",
    "compliance-rules-output": {
      "violated-conditions": "",
      "device-compliance-condition-output": {
        "block-start-unmatched-content": "<[CDATA]]>",
        "block-start-condition-search-output": "<[CDATA]]>",
        "condition-search-output": "<[CDATA]]>[\\n \\matched_contents\\": []\\n]]>",
        "total-block-count": 1,
        "aggregated-condition-output": "<[CDATA]]>[\\n \\condition_contents\\": [\\n \\condition_id\\": null\\n \\block_start_matched_content\\": null\\n \\block_start_unmatched_content\\": null\\n \\unmatched_content\\": null\\n \\matched_content\\": null\\n ]]>"
      }
    }
  }
}
```

Test

Here on Non-Match Action select Continue and on Match Action add Fix cli commands to remediate on the device

Rule Variables

Conditions and Actions

Condition Details

Action Details

Select Match Action

Select action

Raise_violation

Violation severity

CRITICAL

Violation message type

Default_violation_message

Fix CLI

```
interface {{ interface_name }}
no shutdown
negotiation auto
```

Derive fix cli commands

Select

Select Non-Match Action

Select action

continue

For Non-Compliant devices Fix CLI will show up later-on as below.

```
interface GigabitEthernet5
no shutdown
negotiation auto
```

Scenario 4: Enforce VTY Session Timeouts

Scenario: All devices in the network should contain the network admin preferred VTY **session-timeout** and **exec-timeout** on all vty lines. If VTY session-timeout and exec-timeout is not configured on the device or mis-match with the network admin preferred timeouts, ATOM CLI compliance can configure the devices with the user preferred VTY timeouts on all the vty lines.

In this example we are considering the VTY session-timeout and exec-timeout as 10 sec.

Platform:

Cisco IOS-XE

Expected Configuration:

```
line vty (.*)
session-timeout 10
exec-timeout 10 10
```

Fix-CLI Configuration:

```
line vty <>
session-timeout 10
exec-timeout 10 10
```

This use case is using the regex and rule variables and uses jinja2 template for fix-cli configuration.

Steps:

- Navigate to Resource Manager > Config Compliance -> Policies
- Click '+' to create new Policy and provide the following information
 - Policy Name - Enforce_VTY_Session_Timeouts
 - Description
- Select the Policy and Click '+' to create new Rule
 - Rule Name - Check_Enforce_VTY_Session_Timeouts
- Navigate to Resource Manager > Config Compliance -> Policies -> Rules
- Select a Rule & Provide the following information
 - Vendor - Cisco Systems
 - OS type - IOSXE
 - Device Family - ALL
 - Device Type - ALL
 - OS Version - ALL
- Now create the Rule variables for this scenario.

Edit Policy | Enforce_VTY_Session_Timeouts

Edit Rule | Check_vty_session_timeouts

Basic Information | Platform Selection | **Rule Variables** | Conditions and Actions

Rule Variables

🔄 +

<input type="checkbox"/>	Key	Description	Default Value
<input type="checkbox"/>	exec_timeout		10
<input type="checkbox"/>	session_time...		10

Here created user defined rule variables **vty_exec_timeout** and **vty_session_timeout** with default timeout as 10. These rule variables will be used in the condition value.

The **verify_session_exec_timeouts** condition will check whether the device in the network is configured with user preferred VTY timeouts or not.

Conditions and Actions

Condition Details | Action Details

Condition Name •
Condition Name, can contain Alphanumerics, underscore, space and hyphen ...
verify_session_timeouts

Sequence Number •
Sequence Number controls the order of execution of the Conditions.
1

Scope Details
Condition scope details
Configuration

Block Options
Start Expression

Condition Match Criteria
Operator
CONTAINS_STRING
Rule-pass-criteria
All_SubBlocks

Value •
line vty (*)
session-timeout {{ session_timeout }}
exec-timeout {{ exec_timeout }} 0

Launch Test Config

Here under Condition Match Criteria the Operator used was CONTAINS_STRING to check for session-timeout and exec-timeout in line vty config.

Here Rule-pass-criteria used All_SubBlocks to check the condition config in all line vty configurations of the device. If all the line vty is matching with the condition then compliant. If any of the line vty is not matching then non-compliant.

The launch Test Config will check values with the Test configuration and gives the Test Result whether compliant or not.

Test Configuration and Results

Condition Match Operator:

CONTAINS_STRING

Value

```
line vty (*)
session-timeout (( session_timeout ))
exec-timeout (( exec_timeout )) 0
```

Rule Variables

Please enter the rule variables in Key: Value <- Format Ex: Key1: Value1

```
session_timeout: 10
exec_timeout: 10
```

Test Configuration

```
line vty 0 4
session-timeout 5
access-class ssh-permit-aci in
exec-timeout 5 0
privilege level 15
transport input ssh
line vty 5 98
session-timeout 5
access-class ssh-permit-aci in
exec-timeout 5 0
privilege level 15
transport input ssh
!
```

Test Results

```
{
  "unmatched-content": "<CDATA[\\n \\unmatched_contents\\":{\\n \\groups\\":{\\n \\index\\":1,\\n \\grep_content\\":\\n 0 4\\n \\grep_group\\":1\\n  })\\n },\\n \\groups\\":{\\n \\index\\":1,\\n \\grep_content\\":\\n 5 98\\n \\grep_group\\":1\\n  })\\n  }\\n ]>",
  "id": "verify_session_timeouts",
  "block-match-count": 0,
  "cli-unmatch-output": "<CDATA[\\n line vty 0 4\\n session-timeout 5 \\n access-class ssh-permit-aci in\\n exec-timeout 5 0\\n privilege level 15\\n transport input ssh\\n line vty 5 98\\n session-timeout 5 \\n access-class ssh-permit-aci in\\n exec-timeout 5 0\\n privilege level 15\\n transport input ssh\\n ]>"
},
{
  "name": "test-condition",
  "failed-conditions": "",
  },
{
  "compliance-status": "non-compliant"
}
}
```

Test

Here the unmatched line vty will be captured and stored in the backend data structure. The captured data structure maximizes the view shown below.

test-results

```
{
  "compliance-policies": {
    "highest-severity": "",
    "rule-violation-count": 0,
    "noncompliant-rules-output": {
      "violated-conditions": "",
      "device-compliance-condition-output": {
        "block-start-unmatched-content": "<![CDATA[]]>",
        "block-start-condition-search-output": "<![CDATA[]]>",
        "condition-search-output": "<![CDATA[{\\n  \\\"matched_contents\\\" : [ ]\\n}]]>",
        "total-block-count": 2,
        "aggregated-condition-output": "<![CDATA[{\\n  \\\"condition_contents\\\" : [ {\\n    \\\"condition_id\\\" : null,\\n    \\\"block_start_matched_content\\\" : null,\\n    \\\"block_start_unmatched_content\\\" : null,\\n    \\\"unmatched_content\\\" : null,\\n    \\\"matched_content\\\" : null\\n  } ]\\n}]]>",
        "template-substituted-content": "<![CDATA[line vty (.*)\\n session-timeout 10\\n exec-timeout 10 0]]>",
        "block-unmatch-count": 2,
        "cli-match-output": "<![CDATA[]]>",
        "condition-status": false,
        "unmatched-content": "<![CDATA[{\\n  \\\"unmatched_contents\\\" : [ {\\n    \\\"groups\\\" : [ {\\n      \\\"index\\\" : 1,\\n      \\\"grep_content\\\" : \\\"0 4\\\",\\n      \\\"grep_group\\\" : 1\\n    } ]\\n }, {\\n    \\\"groups\\\" : [ {\\n      \\\"index\\\" : 1,\\n      \\\"grep_content\\\" : \\\"5 98\\\",\\n      \\\"grep_group\\\" : 1\\n    } ]\\n } ]\\n}]]>",
        "id": "verify_session_timeouts",
        "block-match-count": 0,
        "cli-unmatch-output": "<![CDATA[line vty 0 4\\n session-timeout 5 \\n access-class ssh-permit-acl in\\n exec-timeout 5 0\\n privilege level 15\\n transport input ssh\\nline vty 5 98\\n session-timeout 5 \\n access-class ssh-permit-acl in\\n exec-timeout 5 0\\n privilege level 15\\n transport input ssh\\n]]>"
      },
      "name": "test-condition",
      "failed-conditions": ""
    },
    "compliance-status": "non-compliant"
  }
}
```

On Match action will continue and on Non-Match Action fix-cli will use the jinja2 template configuration written based on the above captured data structure.

Rule Variables

Conditions and Actions

Condition Details

Action Details

Select Match Action

Select action

continue

Select Non-Match Action

Select action

Raise_violation

Violation severity

CRITICAL

Violation message type

Default_violation_message

Fix CLI

{% for content in unmatched_contents %}
{% for group in content["groups"] %}
line vty {{ group["grep_content"] }}
session-timeout {{ session_timeout }}
exec-timeout {{ exec_timeout }} 0
exit
{% endfor %}
{% endfor %}

Derive fix cli commands

Select

The Non-compliant device fix-cli configurations derived from above jinja2 snippet will look like below.

```

line vty 0 4
  session-timeout 10
  exec-timeout 10 0
exit

line vty 5 98
  session-timeout 10
  exec-timeout 10 0
exit

```

Scenario 5: Enforce OSPF Router Id as Loopback0

Scenario: All devices in the network should contain the OSPF router-id configured with loopback0 ip address. If OSPF router-id is not configured on the device it will configure the OSPF router-id with the value of loopback0 ip address on the devices.

Platform:

Cisco IOS-XE

Expected Configuration:

```
interface Loopback0
 ip address 45.45.45.5 255.255.255.255
!
router ospf 100
 router-id 45.45.45.5
```

Fix-CLI Configuration:

```
router ospf 100
 router-id 45.45.45.5
```

This use case is using the regex and contains two conditions.

1. First condition is to capture and store loopback0 ip address. It will not have a fix-cli configuration as the intention of the condition is to capture loopback0 ip address.

Fix-cli Configuration :

<< no fix cli configuration >>

2. Second condition will check whether the OSPF router id is the same as the first condition's captured loopback0 ip address or not. if not matching then it will configure the OSPF router id with loopback0.

Fix-cli Configuration :

```
router ospf 100
 router-id 45.45.45.5
```

Steps:

- Navigate to Resource Manager > Config Compliance -> Policies
- Click '+' to create new Policy and provide the following information
 - Policy Name - Enforce_OSPF_Router_Id_as_Loopback
 - Description
- Select the Policy and Click '+' to create new Rule
 - Rule Name - Check_OSPF_Router_Id_Cisco
- Navigate to Resource Manager > Config Compliance -> Policies -> Rules
- Select a Rule & Provide the following information
 - Vendor - Cisco Systems
 - OS type - IOSXE
 - Device Family - ALL
 - Device Type - ALL
 - OS Version - ALL
- Rule variables are not required for this scenario.
- Now fill the Conditions and Actions

Edit Policy | Enforce_OSPF_Router_Id_as_Loopback

Edit Rule | Check_OSPF_Router_Id_Cisco

Basic Information

Platform Selection

Rule Variables

Conditions and Actions

Conditions and Actions



- ☐ Condition Name
- ☐ Verify_Loopback0_Ip
- ☐ Verify_OSPF_Router_Id_as_Loopback

Condition1

Conditions and Actions

Condition Details

Action Details

Condition Name

Condition Name, can contain Alphanumerics, underscore, space and hyphen ...

Verify_Loopback0_Ip

Sequence Number

Sequence Number controls the order of execution of the Conditions.

1

Scope Details

Condition scope details

Configuration

Block Options

Start Expression

Condition Match Criteria

Operator

CONTAINS_STRING

Rule-pass-criteria

Any_SubBlock

Value

interface Loopback0
ip address (\d+.\d+.\d+.\d+) (\d+.\d+.\d+.\d+)

Launch Test Config

Another way of writing the above block configuration using the **Block Options** Start Expression is shown below.

The first line “interface Loopback0” can be written in the start Expression with regex symbol ^ to indicate the block starts with interface Loopback0. The remaining configuration lines can be written in value.

Conditions and Actions

Condition Details | Action Details

Condition Name •
Condition Name, can contain Alphanumerics, underscore, space and hyphen ...
Verify_Loopback0

Sequence Number •
Sequence Number controls the order of execution of the Conditions.
1

Scope Details
Condition scope details
Configuration

Block Options
Start Expression
^interface Loopback0

Condition Match Criteria
Operator
MATCHES_THE_EXPRESSION
Rule-pass-criteria
All_SubBlocks

Value •
ip address (\d+\.\d+\.\d+\.\d+) (\d+\.\d+\.\d+\.\d+)

Launch Test Config

The launch test config will check the condition value with the Test configuration and will give the Test Result. Here the captured loopback0 ip address will be stored in the backend data structure as shown below.

Test Configuration and Results

Condition Match Operator:

CONTAINS_STRING

Value

```
interface Loopback0
ip address (ld+.ld+.ld+.ld+) (ld+.ld+.ld+.ld+)
```

Rule Variables

Please enter the rule variables in Key: Value or Format Ex: Key1: Value1

Key1: Value1

Key2: Value2

Key3: Value3

Test Configuration

```
!
interface Loopback0
ip address 45.45.45.5 255.255.255.255
!
interface Loopback1
no ip address
!
interface Loopback200
ip address 94.1.1.1 255.255.255.255
!
```

Test Results

```
{
  "highest-severity": "",
  "rule-violation-count": 0,
  "compliance-status": "compliant",
  "compliant-rules-output": {
    "violated-conditions": "",
    "device-compliance-condition-output": {
      "block-start-unmatched-content": "<![CDATA]]>",
      "block-start-condition-search-output": "<![CDATA]]>",
      "condition-search-output": "<![CDATA[(\n \\'matched_contents\': [(\n \\'groups\': [(\n \\'index\': 1,\n \\'grep_content\': \"45.45.45.5\");(\n \\'grep_group\': 1\n ),(\n \\'index\': 2,\n \\'grep_content\': \"255.255.255.255\");(\n \\'grep_group\': 2\n )]\n )]\n]]>",
      "total-block-count": 1,
      "aggregated-condition-output": "<![CDATA[(\n \\'condition_contents\': [(\n \\'condition_id\': null,\n \\'block_start_matched_content\': null,\n \\'block_start_unmatched_content\': null,\n \\'unmatched_content\': null,\n \\'matched_content\': null\n )]\n )]\n]]>",
      "template-substituted-content": "<![CDATA[interface Loopback0\n ip address (ld+.ld+.ld+.ld+) (ld+.ld+.ld+.ld+)]>"
    }
  }
}
```

Test

The Test result in maximize view is shown below. This output will be used in condition2.

test-results

```

{
  "compliance-policies": {
    "highest-severity": "",
    "rule-violation-count": 0,
    "compliance-status": "compliant",
    "compliant-rules-output": {
      "violated-conditions": "",
      "device-compliance-condition-output": {
        "block-start-unmatched-content": "<![CDATA[]]>",
        "block-start-condition-search-output": "<![CDATA[]]>",
        "condition-search-output": "<![CDATA[{\n  \"matched_contents\" : [ {\n    \"groups\" : [ {\n      \"index\" : 1,\n      \"grep_content\" : \"45.45.45.5\", \n      \"grep_group\" : 1\n    }, {\n      \"index\" : 2,\n      \"grep_content\" : \"255.255.255.255\", \n      \"grep_group\" : 2\n    } ]\n} ]\n}]]>",
        "total-block-count": 1,
        "aggregated-condition-output": "<![CDATA[{\n  \"condition_contents\" : [ {\n    \"condition_id\" : null,\n    \"block_start_matched_content\" : null,\n    \"block_start_unmatched_content\" : null,\n    \"unmatched_content\" : null,\n    \"matched_content\" : null\n  } ]\n}]]>",
        "template-substituted-content": "<![CDATA[interface Loopback0\n ip address (\\d+\\.\\d+\\.\\d+\\.\\d+)\n(\\d+\\.\\d+\\.\\d+\\.\\d+)]]>",
        "block-unmatch-count": 0,
        "cli-match-output": "<![CDATA[interface Loopback0\n ip address 45.45.45.5 255.255.255.255\n]]>",
        "condition-status": true,
        "unmatched-content": "<![CDATA[{\n  \"unmatched_contents\" : [ ]\n}]]>",
        "id": "Verify_Loopback0_ip",
        "block-match-count": 1,
        "cli-unmatch-output": "<![CDATA[]]>"
      },
    },
    "name": "test-condition",
    "failed-conditions": ""
  }
}

```

For Non-Match Action violation is being raised and fix-cli is having no commands as this condition is to capture the loopback0 ip.

Rule Variables

Conditions and Actions

Condition Details

Action Details

Select Match Action

Select action

continue

Select Non-Match Action

Select action

Raise_violation

Violation severity

CRITICAL

Violation message type

Default_violation_message

Fix CLI

Derive fix cli commands

Select

Condition2

The Verify_OSPF_Router_Id_as_Loopback condition will check whether the OSPF router id is the same as the first condition's captured loopback0 ip address or not. if not matching then in fix-cli it will configure the OSPF router id with loopback0.

Conditions and Actions

Condition Details

Action Details

Condition Name

Condition Name, can contain Alphanumerics, underscore, space and hyphen ...

Verify_OSPF_Router_Id_as_Loopback

Sequence Number

Sequence Number controls the order of execution of the Conditions.

1

Scope Details

Condition scope details

Configuration

Block Options

Start Expression

Condition Match Criteria

Operator

CONTAINS_STRING

Rule-pass-criteria

All_SubBlocks

Value

router ospf (*)

router-id {{ condition_contents[0]["matched_content"]["matched_contents"][0]["groups"][0]["grep_content"] }}

Launch Test Config

On Match Action it will continue. On Non-Match Action it will use the jinja2 template configuration in fix-cli to configure the OSPF router id with loopback0 ip.

Rule Variables

Conditions and Actions

Condition Details

Action Details

Select Match Action

Select action

continue

Select Non-Match Action

Select action

Raise_violation

Violation severity

CRITICAL

Violation message type

Default_violation_message

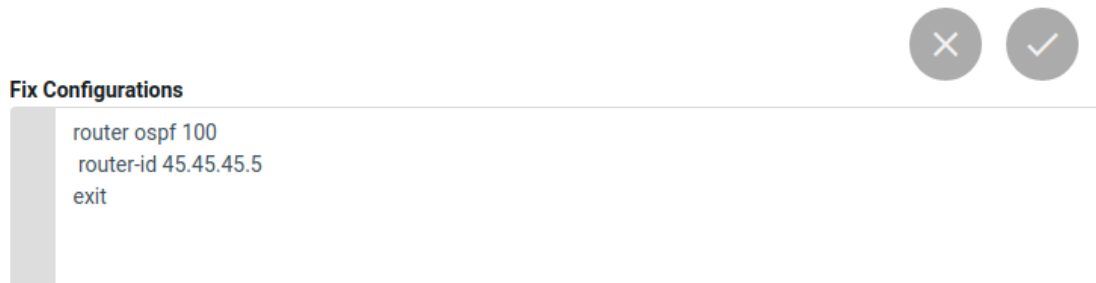
Fix CLI

```
{% for content in unmatched_contents %}
{% for group in content["groups"] %}
router ospf {{ group["grep_content"] }}
router-id {{ condition_contents[0]["matched_content"]
["matched_contents"][0]["groups"][0]["grep_content"] }}
exit
{% endfor %}
{% endfor %}
```

Derive fix cli commands

use-unmatched-block

The Non-compliant device fix-cli configurations from the jinja2 template configuration is given below.



Scenario 6: BGP TTL Hop-count

Scenario: All devices in the network should contain the network admin preferred BGP ttl-security hops. If hops is not configured on the device or mis-match with the network admin preferred ttl-security hops, ATOM CLI compliance can configure the devices with the user preferred hops.

In this example we are considering the ttl-security hops as 5.

Platform:

Cisco IOS-XE

Expected Configuration:

```
router bgp 65535
  bgp log-neighbor-changes
  neighbor 2.3.2.6 ttl-security hops 5
```

Fix-CLI Configuration:

```
router bgp 65535
  bgp log-neighbor-changes
  neighbor 2.3.2.6 ttl-security hops 5
```

This use case is using the regex and rule variables and contains two conditions.

1. First condition is to match the block. It will not have a fix-cli configuration as the intention of the condition is to match the block.

Fix-cli Configuration :

<< no fix cli configuration >>

2. Second condition will check whether the BGP ttl-security hops is in the first condition's matched block or not. if not matching in the block then it will configure the BGP hops.

Fix-cli Configuration :

```
router bgp 65535
  neighbor 2.3.2.6 ttl-security hops 5
```

Steps:

- Navigate to Resource Manager > Config Compliance -> Policies
- Click '+' to create new Policy and provide the following information
 - Policy Name - BGP_TTL_Hop_Count
 - Description
- Select the Policy and Click '+' to create new Rule
 - Rule Name - Check_BGP_TTL
- Navigate to Resource Manager > Config Compliance -> Policies -> Rules
- Select a Rule & Provide the following information
 - Vendor - Cisco Systems
 - OS type - IOSXE
 - Device Family - ALL
 - Device Type - ALL
 - OS Version - ALL
- Now create the Rule variables for this scenario.

Edit Policy | BGP_TTL_Hop_Count

Edit Rule | Check_BGP_TTL

Basic Information > Platform Selection > **Rule Variables** > Conditions and Actions

Rule Variables

🔄 +

<input type="checkbox"/> Key	Description	Default Value
<input type="checkbox"/> asn		65535
<input type="checkbox"/> hops		5
<input type="checkbox"/> neighbor		1.1.1.1

Condition1

The first line "router bgp (.*)" to be written in the start Expression with regex to indicate the block starts with router bgp. The remaining configuration lines can be written in value.

Edit Policy | BGP_TTL_Hop_Count

Edit Rule | Check_BGP_TTL

Basic Information | Platform Selection | Rule Variables | **Conditions and Actions**

Conditions and Actions

Condition Name	Sequence Number
<input checked="" type="checkbox"/> Verify_BGP	1
<input type="checkbox"/> Verify_BGP_TTL	1

Condition Details

Condition Name: Verify_BGP

Sequence Number: 1

Scope Details

Condition scope details: Configuration

Block Options

Start Expression: router bgp (*)

Condition Match Criteria

Operator: CONTAINS_STRING

Rule-pass-criteria: Any_SubBlock

Value: bgp log-neighbor-changes

Launch Test Config

On **Match Action** execution continues to the next condition. On **Non-Match Action** it will raise a violation and continue next condition. The “Fix-CLI” for the condition was written based on the test results obtained from “Launch Test Config”.

When the start Expression is used the regex captured data will be stored in “condition_contents” of “aggregated-condition-output” in test results.

Test Configuration And Results

Condition Match Operator: CONTAINS_STRING

Value: bgp log-neighbor-changes

Rule Variables

```
asn: 65535
neighbor: 1.1.1.1
hops: 5
```

Test Configuration

```
router bgp 65535
bgp log-neighbor-changes
```

Test Results

```
{
  "compliance-policies": {
    "highest-severity": "",
    "rule-violation-count": 0,
    "compliance-status": "compliant",
    "compliance-rules-output": {
      "violated-conditions": ""
    },
    "device-compliance-condition-output": {
      "block-start-unmatched-content": "<[CDATA[\\n \\block_start_unmatched_contents': {}]]>",
      "block-start-condition-search-output": "<[CDATA[\\n \\block_start_matched_contents': {}]]>",
      "grep_content": "65535",
      "condition-search-output": "<[CDATA[\\n \\matched_contents': {}]]>",
      "total-block-count": 1
    },
    "aggregated-condition-output": "<[CDATA[\\n \\condition_contents': {}]]>"
  }
}
```

Test

test-results

```

{
  "compliance-policies": {
    "highest-severity": "",
    "rule-violation-count": 0,
    "compliance-status": "compliant",
    "compliant-rules-output": {
      "violated-conditions": "",
      "device-compliance-condition-output": {
        "block-start-unmatched-content": "<![CDATA[{\\n  \\\"block_start_unmatched_contents\\\": [ ]\\n}]]>",
        "block-start-condition-search-output": "<![CDATA[{\\n  \\\"block_start_matched_contents\\\": [ {\\n    \\\"groups\\\": [ {\\n      \\\"index\\\": 1,\\n      \\\"grep_content\\\": \\\"65535\\\",\\n      \\\"grep_group\\\": 1\\n    } ]\\n } ]\\n}]]>",
        "condition-search-output": "<![CDATA[{\\n  \\\"matched_contents\\\": [ ]\\n}]]>",
        "total-block-count": 1,
        "aggregated-condition-output": "<![CDATA[{\\n  \\\"condition_contents\\\": [ {\\n    \\\"condition_id\\\": \\\"Verify_BGP\\\",\\n    \\\"block_start_matched_content\\\": {\\n      \\\"block_start_matched_contents\\\": [ {\\n        \\\"groups\\\": [ {\\n          \\\"index\\\": 1,\\n          \\\"grep_content\\\": \\\"65535\\\",\\n          \\\"grep_group\\\": 1\\n        } ]\\n      } ]\\n    },\\n    \\\"block_start_unmatched_content\\\": {\\n      \\\"block_start_unmatched_contents\\\": [ ]\\n    },\\n    \\\"unmatched_content\\\": {\\n      \\\"unmatched_contents\\\": [ ]\\n    },\\n    \\\"matched_content\\\": {\\n      \\\"matched_contents\\\": [ ]\\n    } ]\\n} ]\\n}]]>",
        "enforcement-time": 1597311923441,
        "condition-input": "<![CDATA[bgp log-neighbor-changes]]>",
        "template-substituted-content": "<![CDATA[bgp log-neighbor-changes]]>",
        "block-unmatch-count": 0,
        "cli-match-output": "<![CDATA[router bgp 65535\\n bgp log-neighbor-changes\\n]]>",
        "condition-status": true,
        "unmatched-content": "<![CDATA[{\\n  \\\"unmatched_contents\\\": [ ]\\n}]]>",
        "id": "Verify_BGP",
        "block-match-count": 1,
        "cli-unmatch-output": "<![CDATA[]]>"
      },
    },
    "name": "test-condition",
    "failed-conditions": ""
  }
}

```



Basic Information > Platform Selection > Rule Variables > Conditions and Actions

Conditions and Actions

Condition Name	Sequence Number
Verify_BGP	1
Verify_BGP_TTL	1

Condition Details

Select Match Action

Select action: continue

Select Non-Match Action

Select action: Raise_violation_and_continue

Violation severity: CRITICAL

Violation message type: Default_violation_message

Fix CLI

```
bgp router (( condition_contents[@block_start_matched_content]
[block_start_matched_contents[@groups[@grep_content]])
bgp log-neighbor-changes
```

Derive fix cli commands

use-unmatched-block

Condition2

The Verify_BGP_TTL condition will check whether the router bgp block config matched in the previous condition has the ttl-security hops or not. if not matching then in fix-cli it will configure the ttl-security hops.

This condition uses the **condition scope details** as **Previously_Matched_Blocks** to check on previous condition matched block.

Edit Rule | Check_BGP_TTL

Basic Information > Platform Selection > Rule Variables > Conditions and Actions

Conditions and Actions

Condition Name	Sequence Number
Verify_BGP	1
Verify_BGP_TTL	1

Condition Details

Condition Name: Verify_BGP_TTL

Sequence Number: 1

Condition scope details: Previously_Matched_Blocks

Condition Match Criteria

Operator: CONTAINS_STRING

Rule-pass-criteria: Any_SubBlock

Value: neighbor ([neighbor]) ttl-security hops ([hops])

Launch Test Config

Basic Information > Platform Selection > Rule Variables > Conditions and Actions

Conditions and Actions

Condition Name	Sequence Number
<input type="checkbox"/> Verify_BGP	1
<input checked="" type="checkbox"/> Verify_BGP_TTL	1

Condition Details

Selected: 2 Of 2

Enter a keyword

Select Match Action

Select action: continue

Select Non-Match Action

Select action: Raise_violation

Violation severity: CRITICAL

Violation message type: Default_violation_message

Fix CLI

```
bgp router {{ condition_contents[0] }} {
  block_start_matched_content [
    block_start_matched_contents [0] {{ groups[0] }} {{ grep_content }}
  ]
  neighbor {{ neighbor }} ttl-security hops {{ hops }}
```

Derive fix cli commands

use-unmatched-block

YANG Compliance

Note: In order to use Yang Compliance make sure that the config-snapshot is provided in the Credential profile, which lets ATOM to parse the configuration and store it. For more information on Credential profile please refer to credential profile section in ATOM User guide.

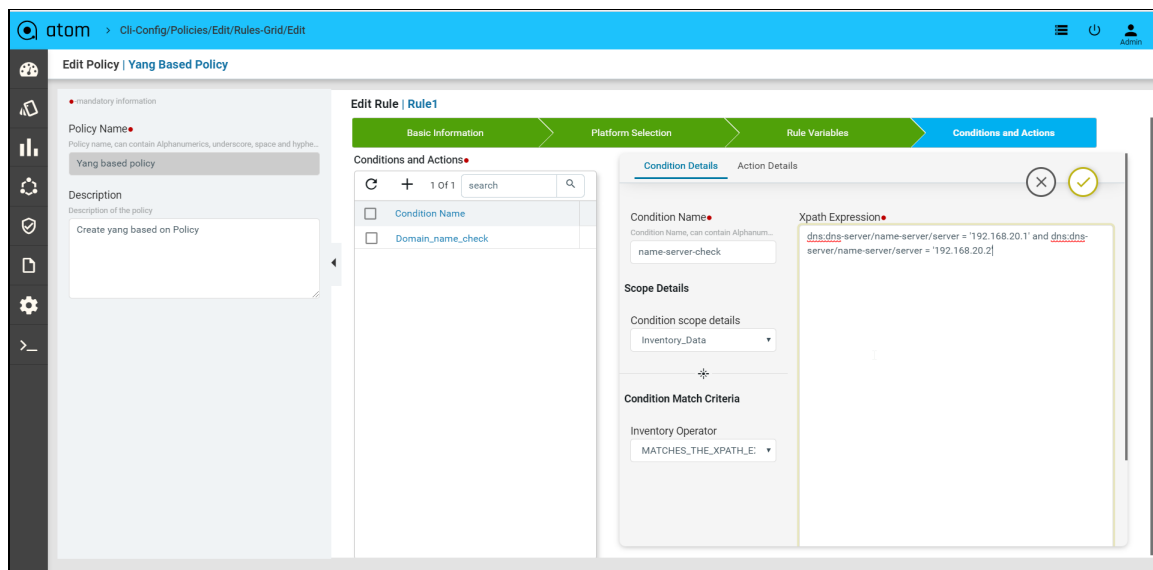
For Yang based Configuration Compliance, make sure to select the option of **Inventory_Data** for **Condition scope** during Compliance **Policy** creation. This gives two ways of defining the **Condition Match Criteria**

- Xpath Expressions
- XML Template Payload

Policy creation with Xpath Expressions

- Within **Condition Match Criteria** select “Matches_the_Xpath_Expression” / “Doesn’t_Matches_the_Xpath_Expression” option for **Inventory Operator** field
- The Fix Mutation Payload is in Netconf xml RPC format written using the XML template details for the yang parsed entities.

Navigate to Resource Manager > Config Compliance > Policy > + (Add Policies)



Few examples

Scenario 7: IP Domain Name

In this example we are looking for the domain name as **anutacorp.com** across all devices in the lab using X-path expression.

Xpath Expression:

```
Cisco-IOS-XR-native:native/ip/domain/name='anutacorp.com'
```

Fix Mutation Payload:

Note: we can use **ATOM_DEVICE_ID** or **inputDeviceId** for substituting the deviceId.

```
<config>
  <devices xmlns="http://anutanetworks.com/controller">
    <device>
      <id>{{ inputDeviceId }}</id>
      <native xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-native">
        <ip>
          <domain nc:operation='create'>
            <name>anutacorp.com</name>
          </domain>
        </ip>
      </native>
    </device>
  </devices>
</config>
```

Defining Xpath Expression

Edit Policy | Yang_IP_Domain_Name

Edit Rule | Check_Yang_IP_Domain_Name

Basic Information Platform Selection Rule Variables Conditions and Actions

Conditions and Actions

Condition Name	Sequence Number
Verify_Yang_IP_Domain_Name	1

Condition Details

Condition Name: Verify_Yang_IP_Domain_Name

Sequence Number: 1

Scope Details

Condition scope details: Inventory_Data

Condition Match Criteria

Inventory Operator: MATCHES_THE_XPATH_EXPRESSION

Xpath Expression

Xpath Expression Ex: `property[Name='value', starts-with[property/Name='value'], contains[property/Name='value']]`

Cisco-IOS-XE-native:ip/domain/name="anutacorp.com"

Launch Test Config

Defining Fix Payload

Edit Policy | Yang_IP_Domain_Name

Edit Rule | Check_Yang_IP_Domain_Name

Basic Information Platform Selection Rule Variables Conditions and Actions

Conditions and Actions

Condition Name	Sequence Number
Verify_Yang_IP_Domain_Name	1

Action Details

Select Match Action

Select action: continue

Select Non-Match Action

Select action: Raise_violation

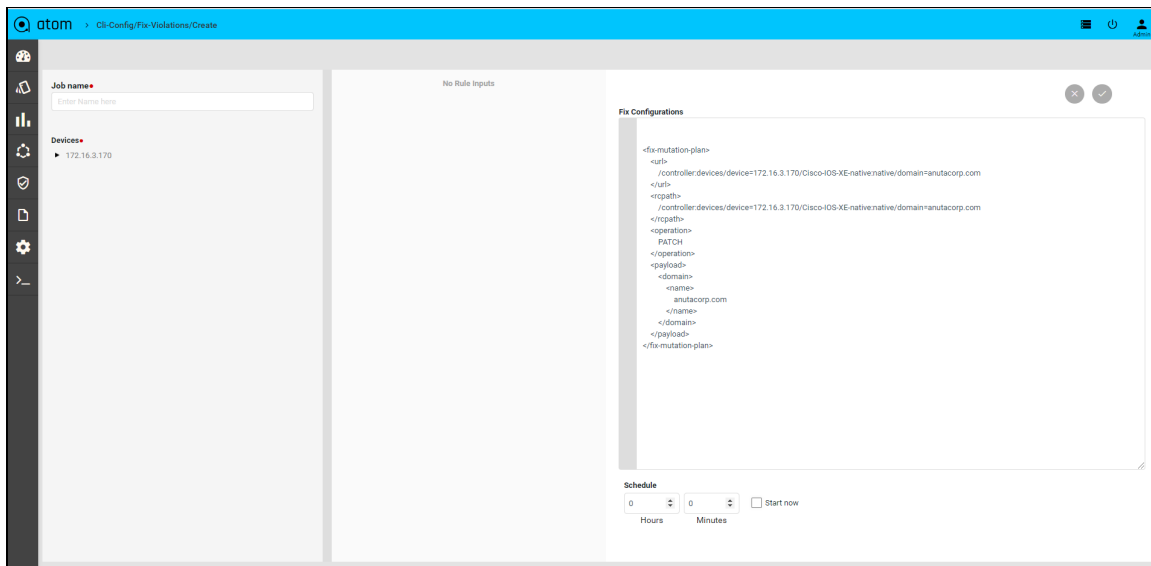
Violation severity: CRITICAL

Violation message type: Default_violation_message

Fix Mutation Payload

```
<config>
  <devices xmlns="http://anutanetworks.com/controller">
    <device>
      <id-{ { inputDeviceId } }>
      <native xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-native">
        <ip>
          <domain nc:operation=create>
            <name=anutacorp.com>
          </domain>
        </ip>
      </native>
    </device>
  </devices>
</config>
```

Fix Configuration Display in Remediation



Scenario 8: IP Name-server check

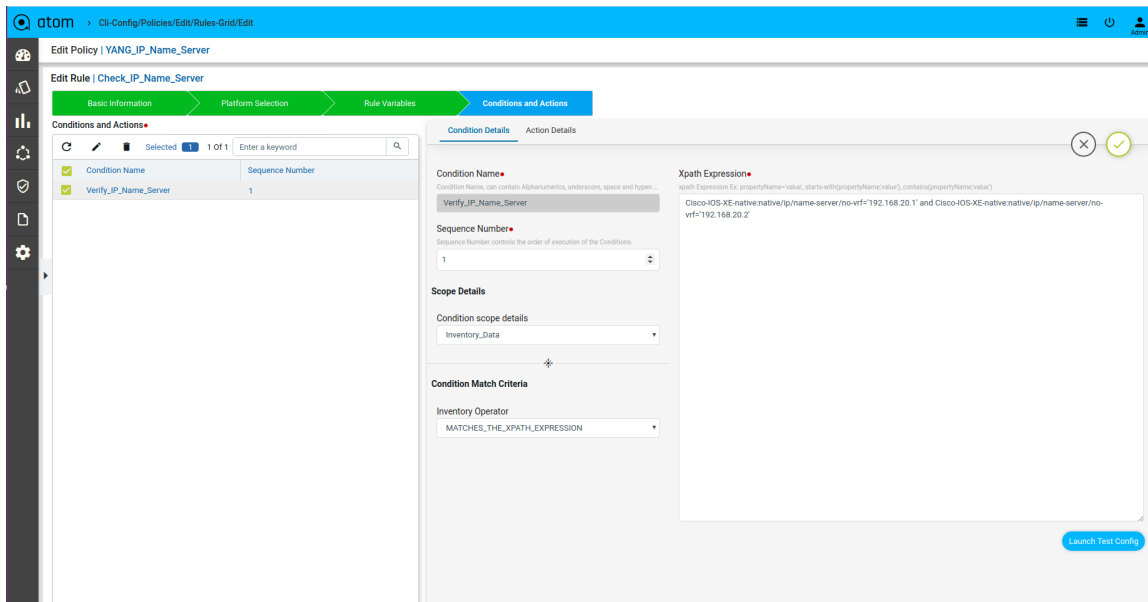
Xpath Expression:

Cisco-IOS-XE-native:native/ip/name-server/no-vrf='192.168.20.1' and
Cisco-IOS-XE-native:native/ip/name-server/no-vrf='192.168.20.2'

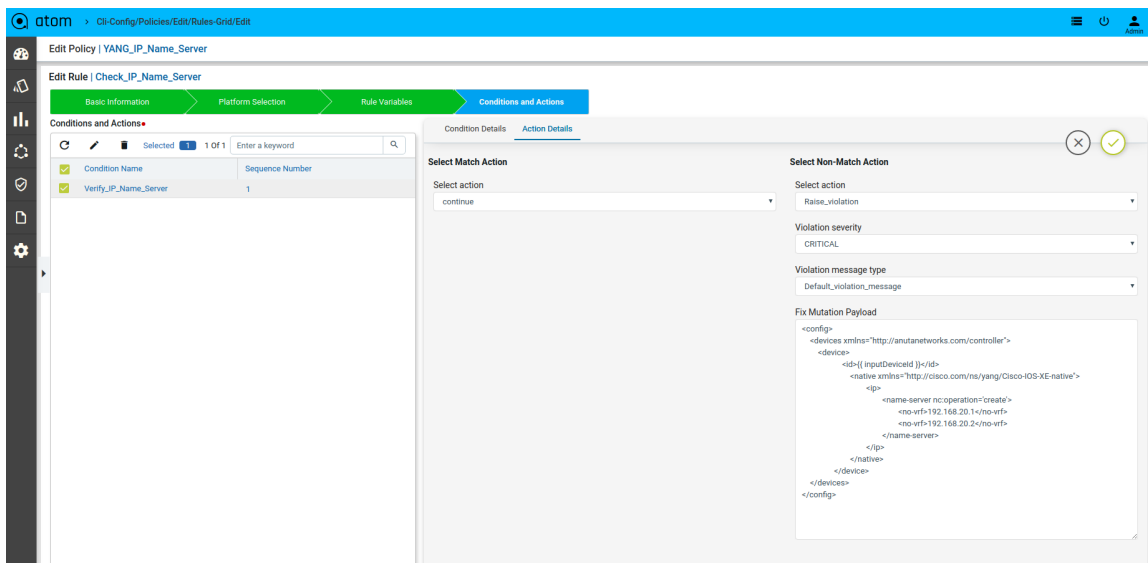
Fix Mutation Payload:

```
<config>
  <devices xmlns="http://anutanetworks.com/controller">
    <device>
      <id>{{ inputDeviceId }}</id>
      <native xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-native">
        <ip>
          <name-server nc:operation='create'>
            <no-vrf>192.168.20.1</no-vrf>
            <no-vrf>192.168.20.2</no-vrf>
          </name-server>
        </ip>
      </native>
    </device>
  </devices>
</config>
```

Defining Xpath Expression



Defining Fix Payload



Scenario 9 : NTP server Check

Xpath Expression:

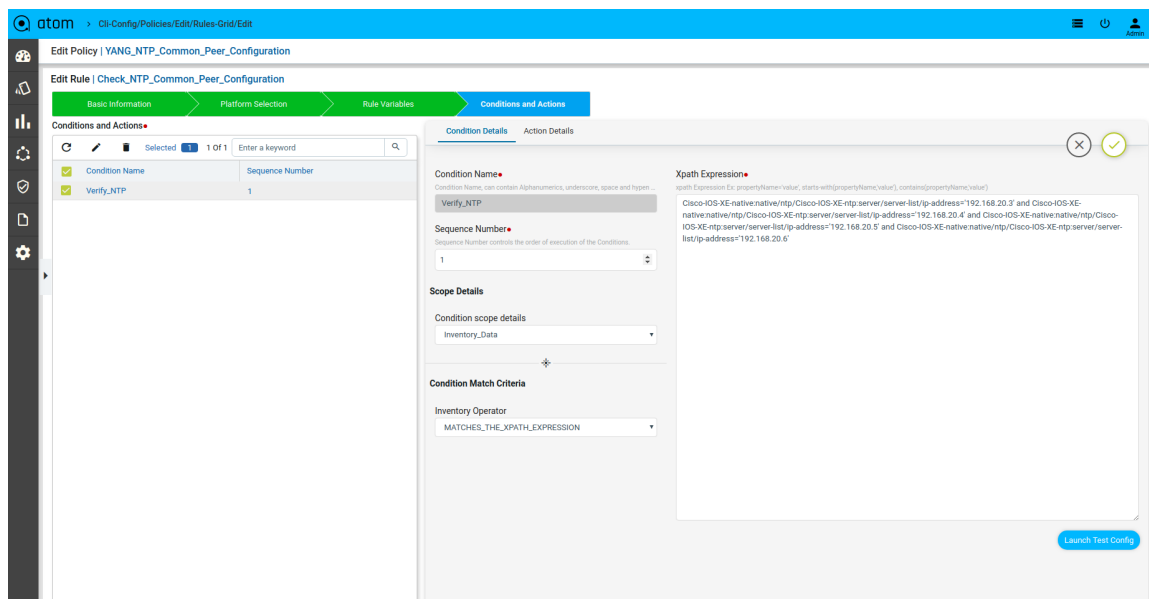
Cisco-IOS-XE-native:ntp/Cisco-IOS-XE-ntp:server/server-list/ip-address='192.16
8.20.3' and
Cisco-IOS-XE-native:ntp/Cisco-IOS-XE-ntp:server/server-list/ip-address='192.16
8.20.4' and
Cisco-IOS-XE-native:ntp/Cisco-IOS-XE-ntp:server/server-list/ip-address='192.16
8.20.5' and


```
Cisco-IOS-XE-native:native/ntp/Cisco-IOS-XE-ntp:server/server-list/ip-address='192.168.20.6'
```

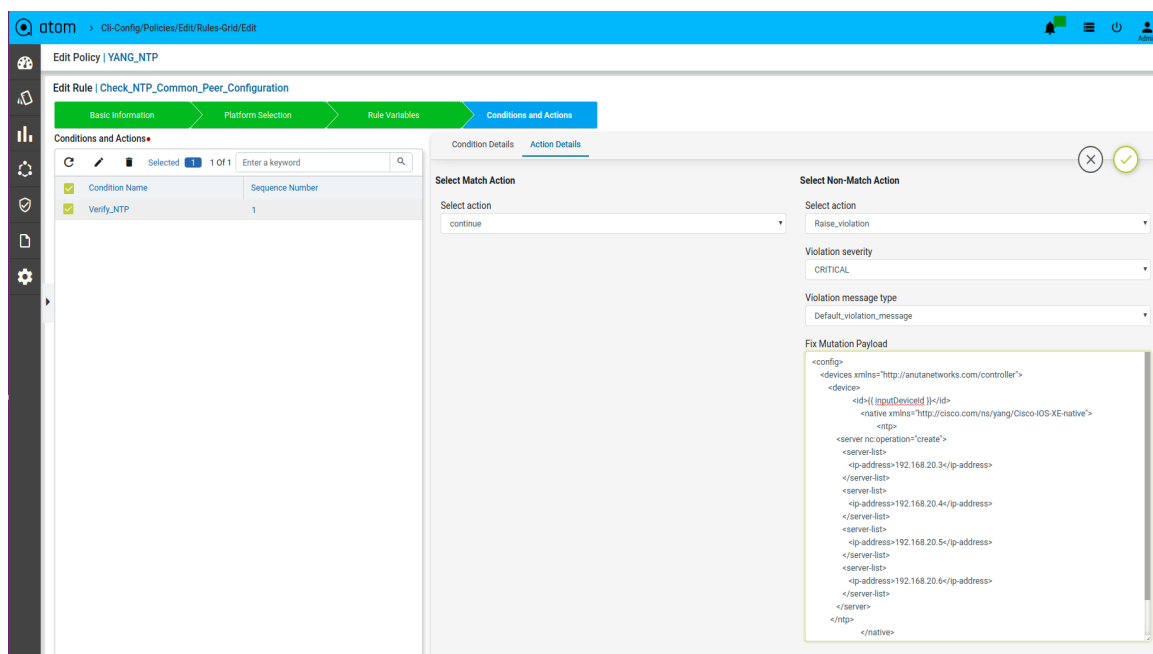
Fix Mutation Payload:

```
<config>
  <devices xmlns="http://anutanetworks.com/controller">
    <device>
      <id>{{ inputDeviceId }}</id>
      <native xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-native">
        <ntp>
          <server nc:operation="create">
            <server-list>
              <ip-address>192.168.20.3</ip-address>
            </server-list>
            <server-list>
              <ip-address>192.168.20.4</ip-address>
            </server-list>
            <server-list>
              <ip-address>192.168.20.5</ip-address>
            </server-list>
            <server-list>
              <ip-address>192.168.20.6</ip-address>
            </server-list>
          </server>
        </ntp>
      </native>
    </device>
  </devices>
</config>
```

Defining Xpath Expression



Defining Fix Payload



Scenario 10 : Interface Check with rule_variable

Xpath Expression:

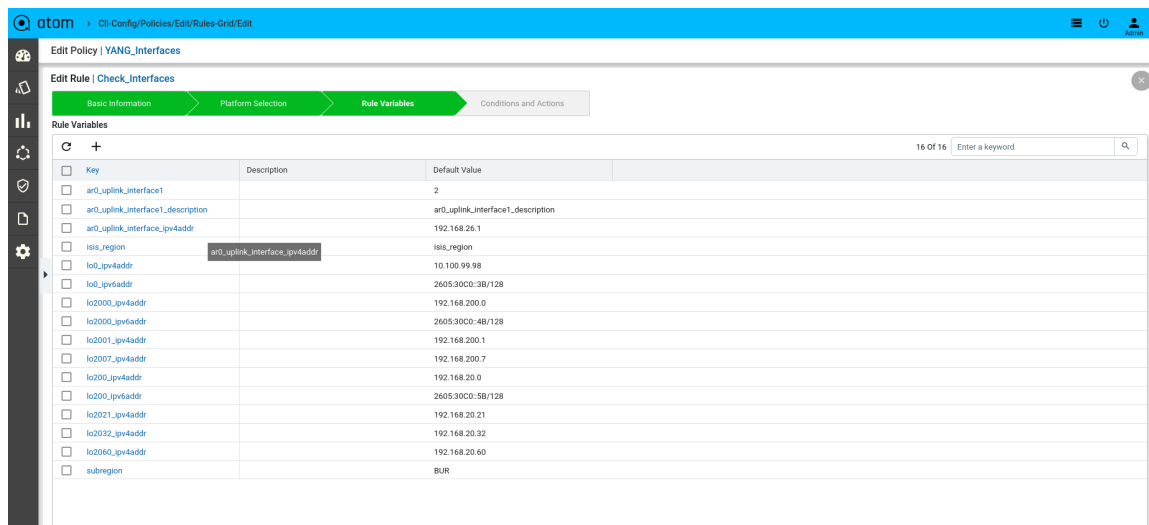
Cisco-IOS-XE-native:native/interface/Loopback[name='0'] and
Cisco-IOS-XE-native:native/interface/Loopback[name=0]/ip/address/primary/address=
'{{ lo0_ipv4addr }}' and

Cisco-IOS-XE-native:native/interface/Loopback[name=0]/ip/address/primary/mask='255.255.255.255' and
 Cisco-IOS-XE-native:native/interface/Loopback[name=0]/ipv6/address/prefix-list/prefix='{{ lo0_ipv6addr }}'

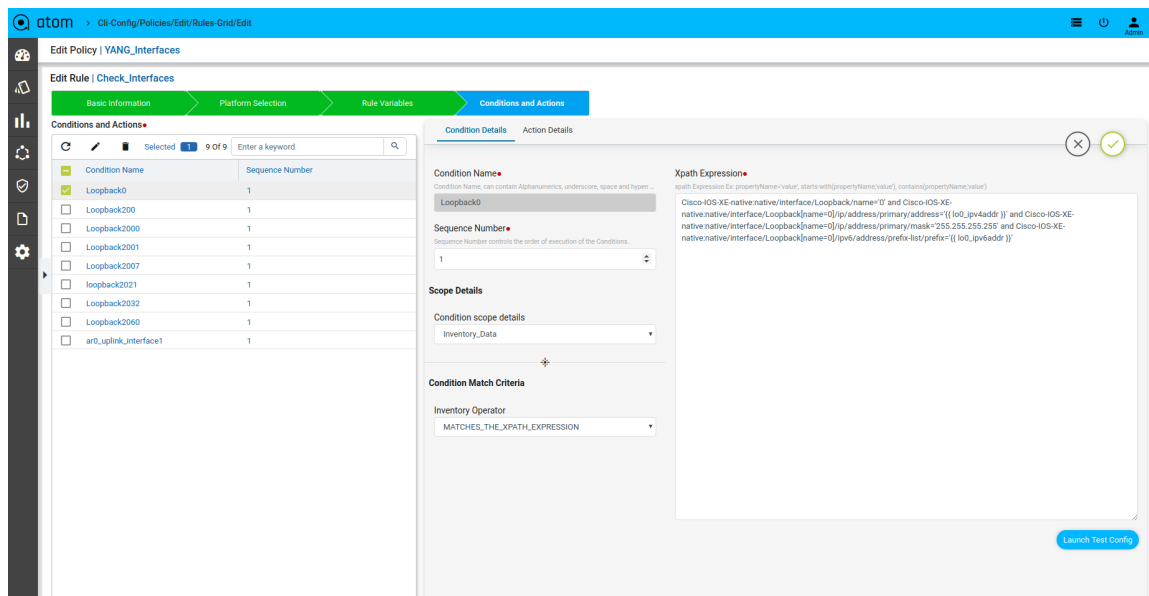
Fix Mutation Payload:

```
<config>
  <devices xmlns="http://anutanetworks.com/controller">
    <device>
      <id>{{ inputDeviceId }}</id>
      <native xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-native">
        <interface>
          <Loopback nc:operation="create">
            <ip>
              <address>
                <primary>
                  <address>10.100.99.98</address>
                  <mask>255.255.255.255</mask>
                </primary>
              </address>
            </ip>
            <ipv6>
              <address>
                <prefix-list>
                  <prefix>2605:30C0::3B/128</prefix>
                </prefix-list>
              </address>
            </ipv6>
            <name>0</name>
          </Loopback>
        </interface>
      </native>
    </device>
  </devices>
</config>
```

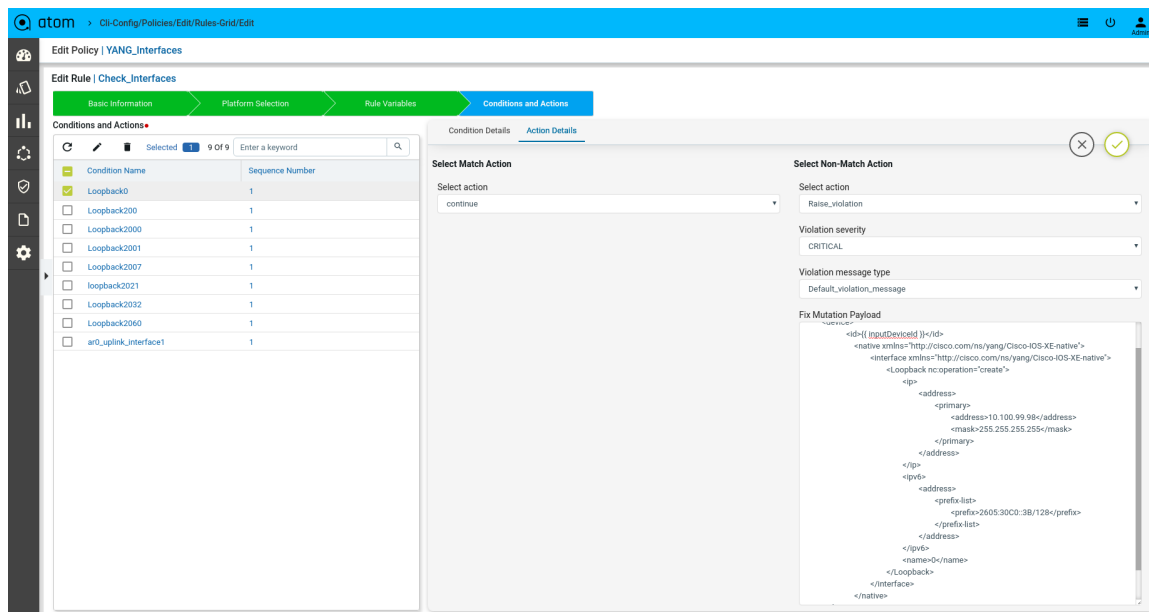
Defining Rule Variables



Defining Xpath Expression



Defining Fix Payload



Scenario 11 : VRF Check with rule_variable

Xpath Expression:

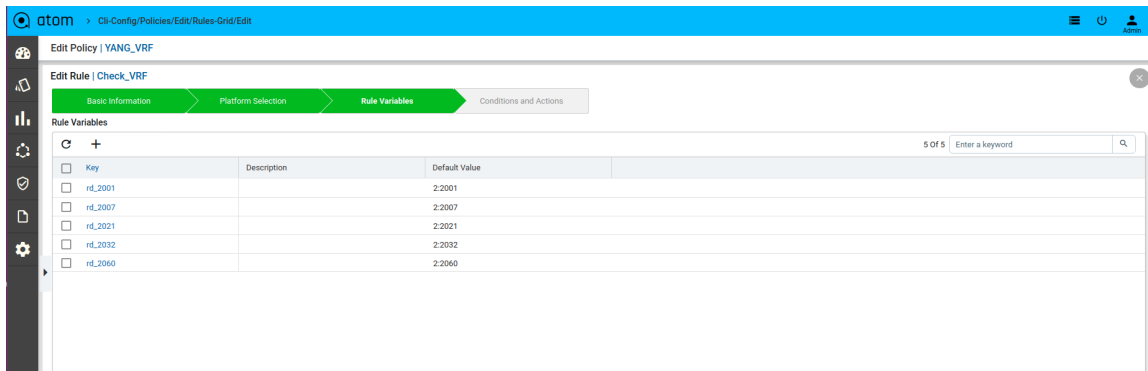
Cisco-IOS-XE-native:native/vrf/definition/name='LON2001' and
 Cisco-IOS-XE-native:native/vrf/definition/rd='{{ rd_2001 }}' and
 Cisco-IOS-XE-native:native/vrf/definition/address-family/ipv4/mdt/default/address=23
 9.232.0.1' and
 Cisco-IOS-XE-native:native/vrf/definition/address-family/ipv4/mdt/data/multicast/add
 ress='239.232.1.0' and
 Cisco-IOS-XE-native:native/vrf/definition/address-family/ipv4/mdt/data/multicast/wil
 dcard='0.0.0.255' and
 Cisco-IOS-XE-native:native/vrf/definition/address-family/ipv4/route-target/export/asn
 -ip='2:2001' and
 Cisco-IOS-XE-native:native/vrf/definition/address-family/ipv4/route-target/import/asn
 -ip='2:2001'

Fix Mutation Payload:

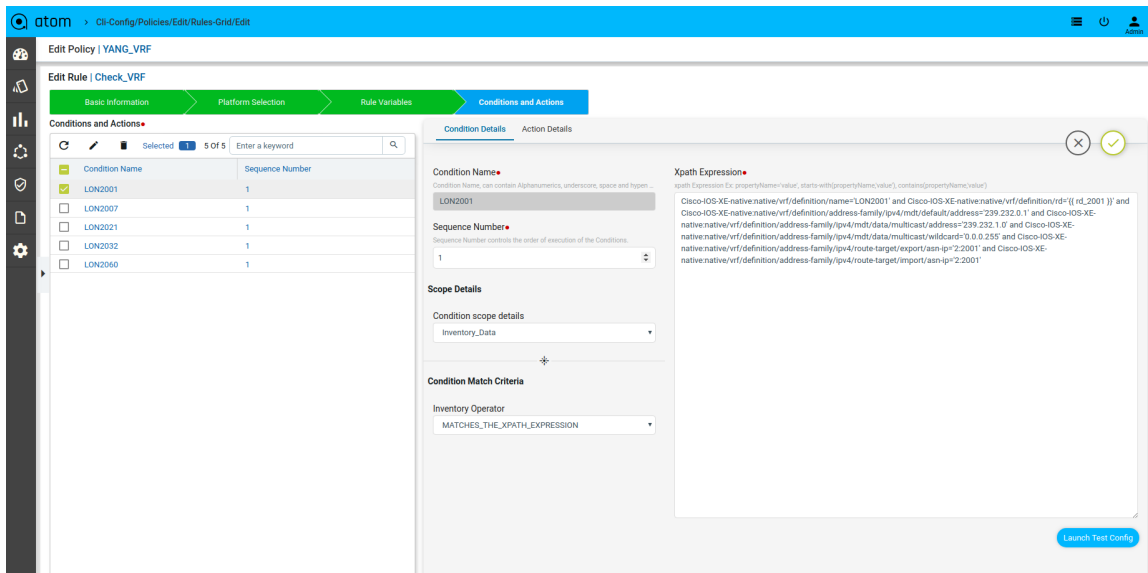
```
<config>
  <devices xmlns="http://anutanetworks.com/controller">
    <device>
      <id>{{ inputDeviceId }}</id>
      <native xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-native">
        <vrf>
          <definition nc:operation="create">
```

```
<rd>2:2001</rd>
<name>LON2001</name>
<address-family>
  <ipv4>
    <route-target>
      <export>
        <asn-ip>2:2001</asn-ip>
      </export>
      <import>
        <asn-ip>2:2001</asn-ip>
      </import>
    </route-target>
    <mdt>
      <default>
        <address>239.232.0.1</address>
      </default>
      <data>
        <multicast>
          <address>239.232.1.0</address>
          <wildcard>0.0.0.255</wildcard>
        </multicast>
      </data>
    </mdt>
  </ipv4>
</address-family>
</definition>
</vrf>
</native>
</device>
</devices>
</config>
```

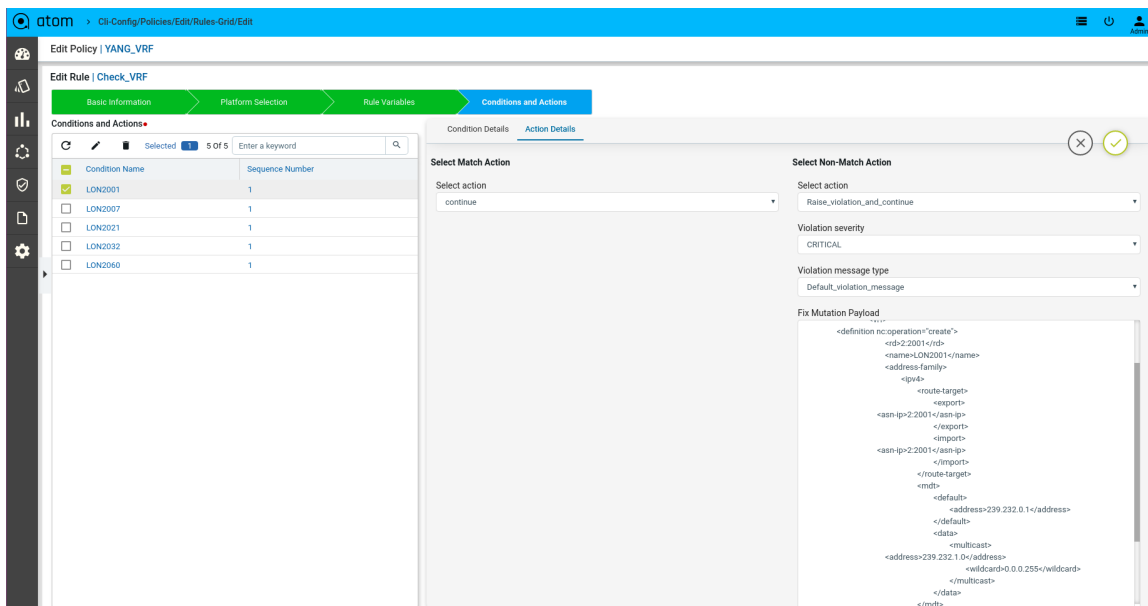
Defining Rule Variables



Defining Xpath Expression



Defining Fix Payload



- **Snmp-string with rule_variable :**
basicDeviceConfigs:snmp/snmp-community-list/snmp-string = "{{ community }}"
- **Logical A|B :** starts-with(vendor-string,'Cisco') or contains(device-family-string,'Cisco 800')
- **Logical A&B :** starts-with(vendor-string,'Cisco') and contains(device-family-string,'Cisco 800')
- **Logical A&(B|C) :** contains(vendor-string,'Cisco Systems') and (contains(device-family-string,'Cisco 800') or contains(device-family-string,'Cisco CSR 1000V'))
- **Logical A&(B|(C&D)) :**
contains(interface:interfaces/interface/if-name,'GigabitEthernet1') and (contains(os-version,'15.6(1)S') or (contains(vendor-string,'Cisco Systems') and contains(device-family-string,'Cisco CSR 1000V')))
- **Logical not(A&B) :**
not(contains(basicDeviceConfigs:local-credentials/local-credential/name , 'admin') and contains(basicDeviceConfigs:local-credentials/local-credential/name , 'cisco'))

How to derive the X-path expressions

There can be two ways by which you can derive the X-path expressions

- Navigate to the Device profile page to get the X-path Expression Details for the yang parsed entities

Resource Manager → Devices → Select a Device → Configuration → Config Data → Entities → Select Entity

For Example: If we want to write xpath expression for VRF name to match as “**anuta**”, then below is how condition needs to be written

l3features:vrfs/vrf/name = 'anuta'

l3features:vrfs/vrf : **This is x-path derived based on model under device**

name : **Attribute of vrf name.**

- Navigate to **Schema Browser** to see all yang models under path /controller:devices/device

Policy creation with XML Template Payload

- Within **Condition Match Criteria** select “Matches_the_template_payload”
/“Doesn't_matches_the_template_payload” option for **Inventory Operator** field

- The Fix Mutation Payload is a Jinja2 template configuration in Netconf xml RPC format written using the unmatched content from the test results tab.

Navigate to Resource Manager > Config Compliance > Policy > + (Add Policies)

Few examples

Scenario 12 : IP Domain name check

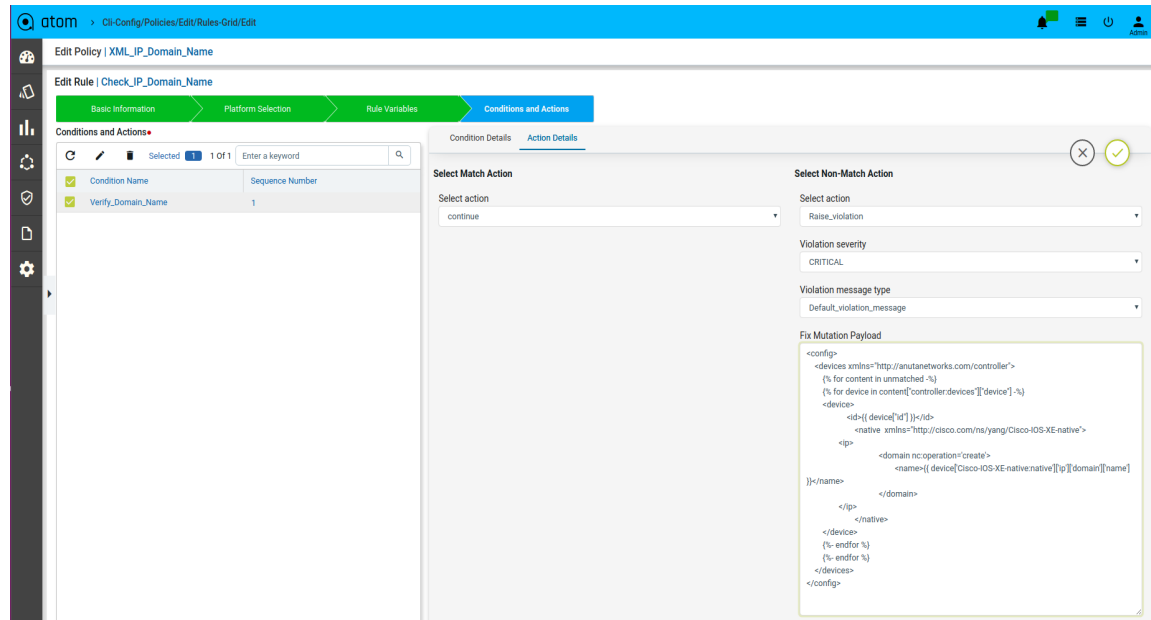
Template Payload:

```
<native xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-native">
  <ip>
    <domain>
      <name>net.disney.com</name>
    </domain>
  </ip>
</native>
```

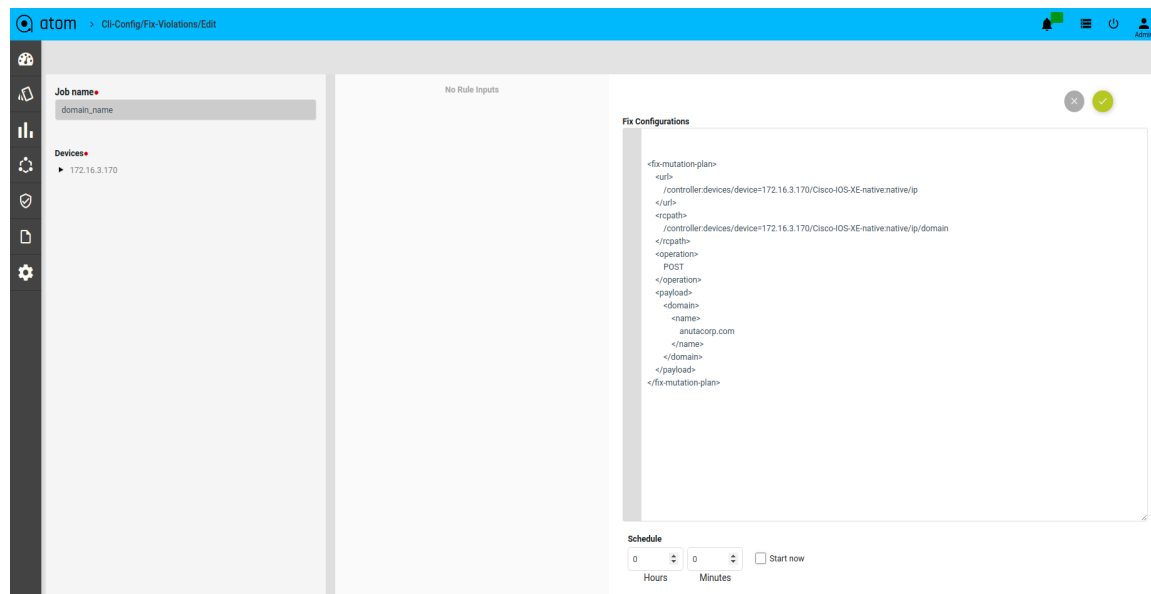
Fix Mutation Payload :

```
<config>
  <devices xmlns="http://anutanetworks.com/controller">
    {% for content in unmatched -%}
    {% for device in content["controller:devices"]["device"] -%}
    <device>
      <id>{{ device["id"] }}</id>
      <native xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-native">
        <ip>
          <domain nc:operation='create'>
            <name>{{ device['Cisco-IOS-XE-native:native']['ip']['domain']['name'] }}</name>
          </domain>
        </ip>
      </native>
    </device>
    {%- endfor %}
    {%- endfor %}
  </devices>
</config>
```


Defining Fix Payload



Fix Configuration Display in Remediation



Scenario 13 : IP Name Server check

Template Payload:

```
<native xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-native">
  <ip>
    <name-server>
```

```

    <no-vrf>192.168.20.1</no-vrf>
    <no-vrf>192.168.20.2</no-vrf>
  </name-server>
</ip>
</native>

```

Fix Mutation Payload :

```

<config>
  <devices xmlns="http://anutanetworks.com/controller">
    {% for content in unmatched -%}
    {% for device in content["controller:devices"]["device"] -%}
    <device>
      <id>{{ device["id"] }}</id>
      <native xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-native">
        <ip>
          <name-server nc:operation='create'>
            {% for name_server in
device['Cisco-IOS-XE-native:native']['ip']['name-server']['no-vrf'] -%}
              <no-vrf>{{ name_server }}</no-vrf>
            {%- endfor %}
          </name-server>
        </ip>
      </native>
    </device>
    {%- endfor %}
    {%- endfor %}
  </devices>
</config>

```

atom > Cti-Config/Polices/Add/Rules-Grid/Add

Edit Policy | XML_IP_Name_Server

Edit Rule | Check_IP_Name_Server

Basic Information Platform Selection Rule Variables **Conditions and Actions**

Conditions and Actions

Condition Name	Sequence Number
Verify_Name_Server	1

Condition Details

Condition Name: Verify_Name_Server

Sequence Number: 1

Scope Details

Condition scope details: Inventory_Data

Condition Match Criteria

Inventory Operator: MATCHES_THE_TEMPLATE_PAYLOAD

Template Payload

```
<native xmlns="http://cisco.com/ns/ietf/Cisco-IOS-XE-native">
  <ip>
    <name-server>
      <no-vrf>192.168.20.1</no-vrf>
      <no-vrf>192.168.20.2</no-vrf>
    </name-server>
  </ip>
</native>
```

Launch Test Config

Defining Fix Payload

atom > Cti-Config/Polices/Add/Rules-Grid/Add

Edit Policy | XML_IP_Name_Server

Edit Rule | Check_IP_Name_Server

Basic Information Platform Selection Rule Variables **Conditions and Actions**

Conditions and Actions

Condition Name	Sequence Number
Verify_Name_Server	1

Action Details

Select Match Action

Select action: continue

Select Non-Match Action

Select action: Raise_violation

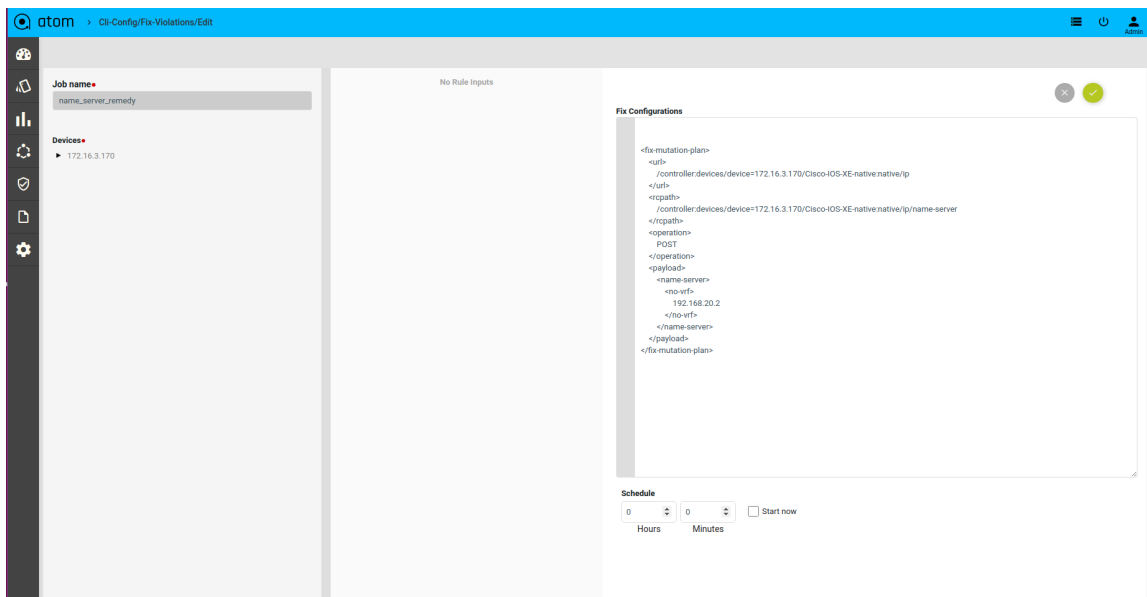
Violation severity: CRITICAL

Violation message type: Default_violation_message

Fix Mutation Payload

```
<config>
  <devices xmlns="http://anutanetworks.com/controller">
    (% for content in unmatched -%)
    (% for device in content["controller/devices"] ["device"] -%)
    <device>
      <id>[% device["id"] %]</id>
      <native xmlns="http://cisco.com/ns/ietf/Cisco-IOS-XE-native">
        <ip>
          <name-server nc:operation=create>
            (% for name_server in device["Cisco-IOS-XE-native/native"] ["ip"]
              [% name_server["no-vrf"] -%])
            <no-vrf>[% name_server %]</no-vrf>
          </name-server>
        </ip>
      </native>
    </device>
  </endfor %>
</devices>
</config>
```

Fix Configuration Display in Remediation



Scenario 14 : Interface check

Template Payload:

```
<native xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-native">
  <interface>
    <Loopback>
      <ip>
        <address>
          <primary>
            <address>10.100.99.98</address>
            <mask>255.255.255.255</mask>
          </primary>
        </address>
      </ip>
      <ipv6>
        <address>
          <prefix-list>
            <prefix>2605:30C0::3B/128</prefix>
          </prefix-list>
        </address>
      </ipv6>
      <name>0</name>
    </Loopback>
  </interface>
```

</native>
Fix Mutation Payload :

```

<config>
  <devices xmlns="http://anutanetworks.com/controller">
    {% for content in unmatched -%}
    {% for device in content["controller:devices"]["device"] -%}
    <device>
      <id>{{ device["id"] }}</id>
      <native xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-native">
        <interface>
          {% for loopback in
device['Cisco-IOS-XE-native:native']['interface']['Loopback'] -%}
          <Loopback nc:operation="create">
            {% if loopback['ip'] -%}
            <ip>
              <address>
                <primary>
                  <address>{{ loopback['ip']['address']['primary']['address']
}}</address>
                  <mask>{{ loopback['ip']['address']['primary']['mask'] }}</mask>
                </primary>
              </address>
            </ip>
            {%- endif %}
            {% if loopback['ipv6'] -%}
            <ipv6>
              <address>
                <prefix-list>
                  {% for prefix in loopback['ipv6']['address']['prefix-list'] -%}
                  <prefix>{{ prefix['prefix'] }}</prefix>
                  {%- endfor %}
                </prefix-list>
              </address>
            </ipv6>
            {%- endif %}
            <name>{{ loopback['name'] }}</name>
          </Loopback>
        {%- endfor %}
      </interface>
    </native>

```

```

</device>
{% - endfor %}
{% - endfor %}
</devices>
</config>

```

Defining Template Payload

Condition Name

Condition Name	Sequence Number
Loopback0	1
Loopback200	1
Loopback2000	1
Loopback2001	1
Loopback2007	1
Loopback2021	1
Loopback2032	1
Loopback2060	1
ar0_uplink_interface1	1

Condition Name: Loopback0

Sequence Number: 1

Scope Details: Inventory_Data

Condition Match Criteria: Inventory Operator: MATCHES_THE_TEMPLATE_PAYLOAD

Template Payload:

```

<native xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-native">
  <interface>
    <Loopback>
      <address>
        <primary>
          <address>10.100.99.98</address>
        </primary>
        <address>255.255.255.255</mask>
      </address>
    </Loopback>
  </interface>
</native>

```

Defining Template Payload

Condition Name

Condition Name	Sequence Number
Loopback0	1
Loopback200	1
Loopback2000	1
Loopback2001	1
Loopback2007	1
Loopback2021	1
Loopback2032	1
Loopback2060	1
ar0_uplink_interface1	1

Condition Name: Loopback0

Sequence Number: 1

Select Match Action: continue

Select Non-Match Action: Raise_violation_and_continue

Violation severity: CRITICAL

Violation message type: Default_violation_message

Fix Mutation Payload:

```

<id>{{ device|id }}</id>
<native xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-native">
  <interface xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-native">
    {% for loopback in device["Cisco-IOS-XE-native"]["interface"] %}
      [Loopback] -%>
        <Loopback nc:operation="create">
          <ip>
            <address>
              <primary>
                <address>{{ loopback["ip"]|address }}</primary>
              </primary>
              <address>{{ loopback["ip"]|address }}</mask>
            </address>
          </ip>
          <prefix-list>
            <prefix-list>
              {% for prefix in loopback["ipv6"]|address }}</prefix-list>
                <prefix>{{ prefix["prefix"] }}</prefix>
                {% endfor %}
              </prefix-list>
            </prefix-list>
          </prefix-list>
        </Loopback>
      </interface>
    </native>
  </native>
</native>

```

Scenario 15 : VRF check

Template Payload:


```

<native xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-native">
  <vrf>
    <definition>
      <address-family>
        <ipv4>
          <mdt>
            <default>
              <address>239.232.0.1</address>
            </default>
            <data>
              <multicast>
                <address>239.232.1.0</address>
                <wildcard>0.0.0.255</wildcard>
              </multicast>
            </data>
          </mdt>
          <route-target>
            <export>
              <asn-ip>2:2001</asn-ip>
            </export>
            <import>
              <asn-ip>2:2001</asn-ip>
            </import>
          </route-target>
        </ipv4>
      </address-family>
      <name>LON2001</name>
      <rd>2:2001</rd>
    </definition>
  </vrf>
</native>

```

Fix Mutation Payload :

```

<config>
  <devices xmlns="http://anutanetworks.com/controller">
    {% for content in unmatched -%}
    {% for device in content["controller:devices"]["device"] -%}
    <device>
      <id>{{ device["id"] }}</id>
      <native xmlns="http://cisco.com/ns/yang/Cisco-IOS-XE-native">
        <vrf>

```

```

        {% for vrf_def in device['Cisco-IOS-XE-native:native']['vrf']['definition']
-%}

        {% if vrf_def['name'] == 'LON2001' -%}
        <definition nc:operation="create">
            {% if vrf_def['rd'] -%}
            <rd>{{ rd_2001 }}</rd>
            {%- endif %}
            <name>{{ vrf_def['name'] }}</name>
            {% if vrf_def['address-family'] -%}
            <address-family>
                <ipv4>
                    {% if vrf_def['address-family']['ipv4']['route-target'] -%}
                    <route-target>
                        {% for export in
vrf_def['address-family']['ipv4']['route-target']['export'] -%}
                            <export>
                                <asn-ip>{{ export['asn-ip'] }}</asn-ip>
                            </export>
                        {%- endfor %}
                        {% for import in
vrf_def['address-family']['ipv4']['route-target']['import'] -%}
                            <import>
                                <asn-ip>{{ import['asn-ip'] }}</asn-ip>
                            </import>
                        {%- endfor %}
                    </route-target>
                    {%- endif %}
                    {% if vrf_def['address-family']['ipv4']['import'] -%}
                    <import>
                        <map>{{
vrf_def['address-family']['ipv4']['import']['map'] }}</map>
                    </import>
                    {%- endif %}
                    {% if vrf_def['address-family']['ipv4']['mdt'] -%}
                    <mdt>
                        {% if
vrf_def['address-family']['ipv4']['mdt']['default'] -%}
                        <default>
                            <address>{{
vrf_def['address-family']['ipv4']['mdt']['default']['address'] }}</address>

```

```

                                </default>
                                {% - endif %}
                                {% if vrf_def['address-family']['ipv4']['mdt']['data']
- %}

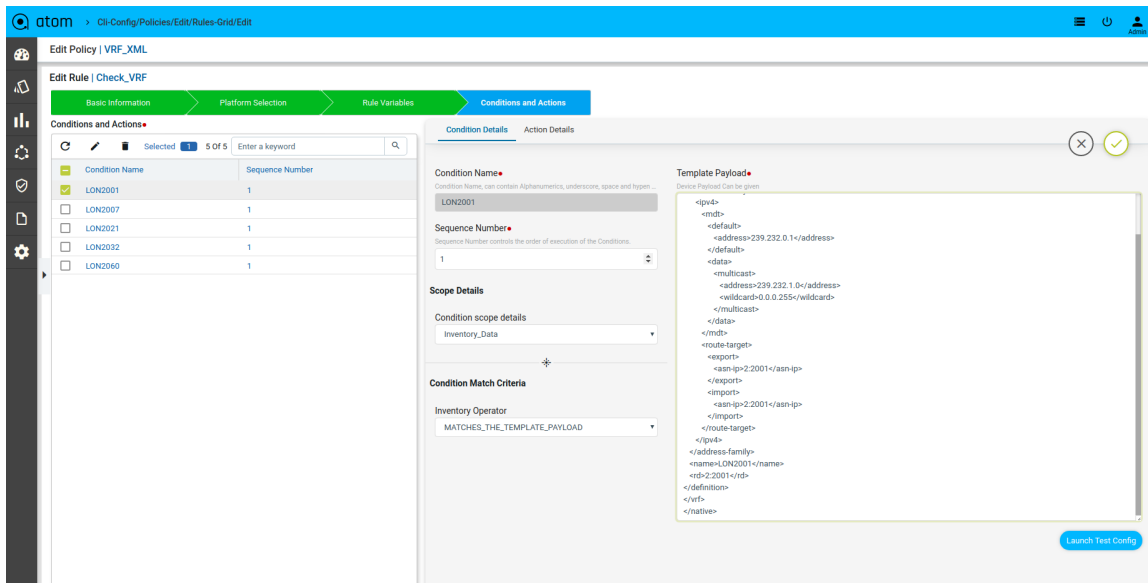
                                <data>
                                {% for multicast in
vrf_def['address-family']['ipv4']['mdt']['data']['multicast'] - %}
                                <multicast>
                                    <address>{{ multicast['address']
}}</address>

                                    <wildcard>{{ multicast['wildcard']
}}</wildcard>

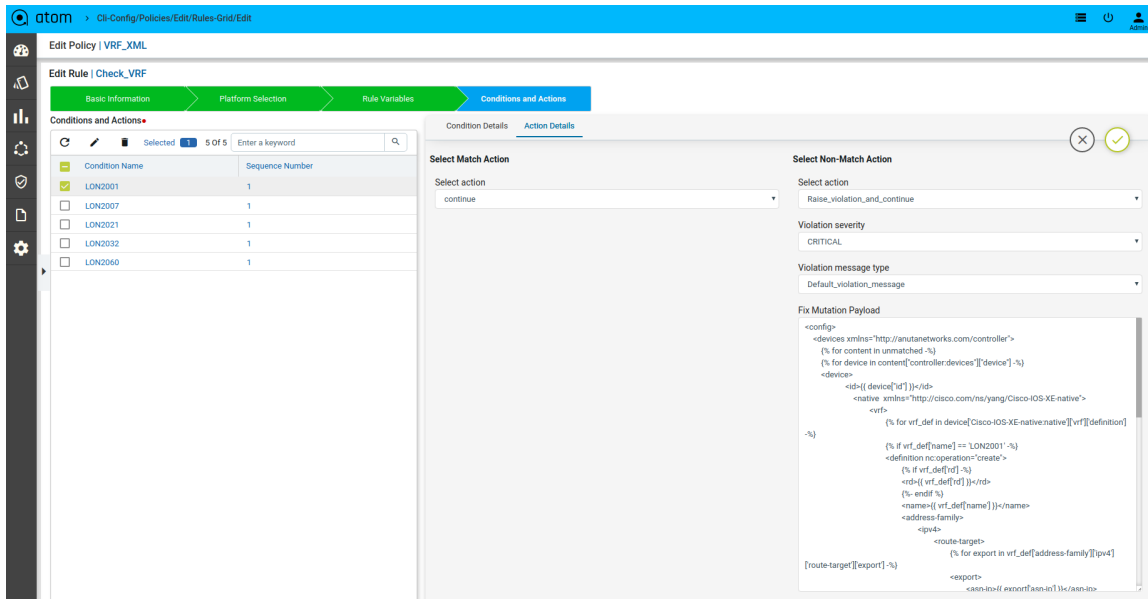
                                </multicast>
                                {% - endfor %}
                                </data>
                                {% - endif %}
                                </mdt>
                                {% - endif %}
                                </ipv4>
                                </address-family>
                                {% - endif %}
                                </definition>
                                {% - endif %}
                                {% - endfor %}
                                </vrf>
                                </native>
                                </device>
                                {% - endfor %}
                                {% - endfor %}
                                </devices>
</config>

```

Defining Template Payload



Defining Fix Payload



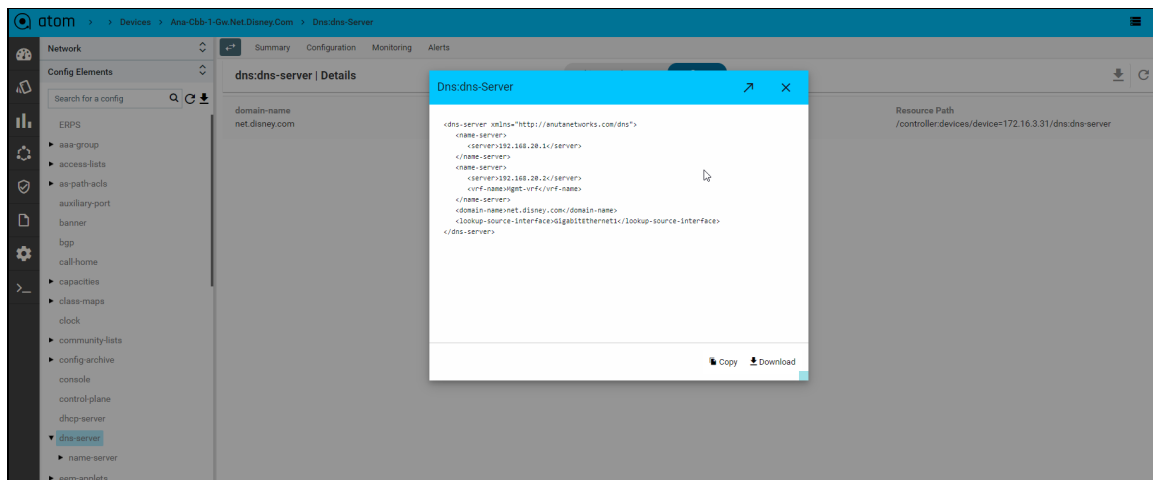
How to derive the XML Template payload

- Navigate to the Device profile page and export the XML template details for the yang parsed entities

Resource Manager → Devices → Select a Device → Configuration → Config Data → Entities → Select Abstract entity → Use Download button to export/copy the XML payload

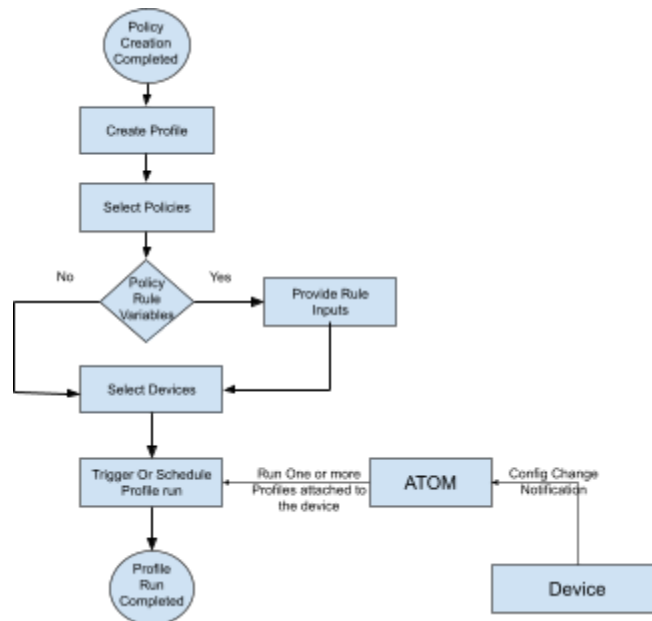
Example : Let's derive domain-name XML Template payload

Navigate to **Devices** → **select a device** → **configuration** -> **Config Data** → **Entities** → **dns-server** → **Export XML** payload using download button.



Profiles

A profile allows one or more Policies to be grouped and executed on one or more devices either on-demand or as per Schedule. Profile execution results in a per-device compliance report included in the execution.



Steps:

a. Navigate to Resource Manager > Config Compliance -> Profiles

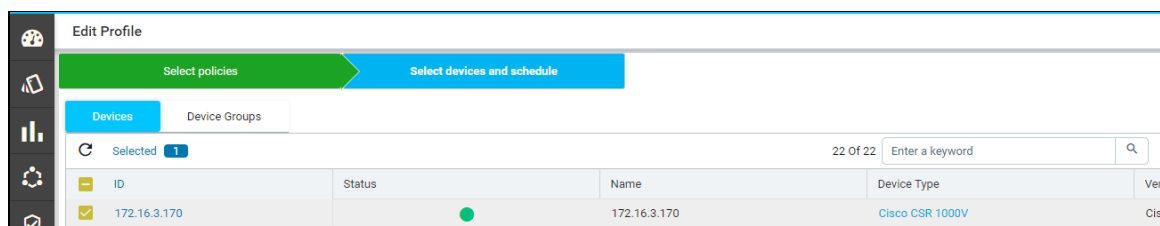
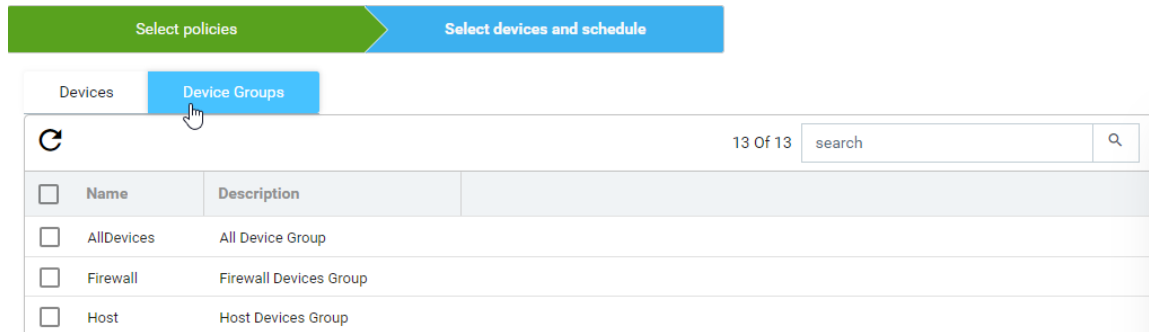
- Select “+” to Create a Profile
- ATOM opens up a new wizard and displays 2 sections.
 - Policies - Select one/more policies
 - Devices & Schedule - Select one/more devices or Device groups

b. Create profile by providing name, description and select policy which was created previously

IP_Domain_Name.

The screenshot shows the 'Select policies' step of the ATOM wizard. On the left, there are fields for 'Profile name' (containing 'Day0_Config') and 'Description'. On the right, a table titled 'Select policies' shows a list of policies. The first policy is 'IP_Domain_Name', which is selected (indicated by a green checkmark) and has a description 'Check whether the domain name is present in the...'. The table has columns for 'Name' and 'Description'. Above the table, it says 'Selected 1' and '33 Of 33'.

c. Navigate to the next tab, Select devices and schedule. We can select either device(s) from *Devices* or *Device Groups* tab



d. After device(s) are selected, choose if the compliance checks need to be run against an archived config or current running-configuration of the device. By default Latest From Config Archive is selected.

Schedule: The profile job can be scheduled in Hours or Minutes. Alternatively, a job can be started right away by enabling *Start now* option.

Configuration:

- Current Config: This will pull the current device configuration and evaluate against the polices.
- Latest From Config Archive: This will use the latest configuration that is stored in the ATOM. Optionally you can add a check to skip the compliance check when the configuration is older than **n** hrs

Select Configuration
☐ Current Config
☒ Latest From Config Archive
Skip when config older than

0

Hours

Schedule
Frequency

0






Hours

0

Minutes

☐ Start now

Or the profile job can triggered at a later point of time using run job icon on the profiles view

<div>     Selected 1</div>			
Name	Description	Policies	
<input checked="" type="checkbox"/> ip_domain_profile		IP_Domain_Name	

Report

Navigate to Resource Manager > Config Compliance -> reports

Compliance report is generated upon completion of Profile run. For each device, the report lists the compliant and non-compliant policies, rules and conditions .

After profile job is run, audit details can be viewed in Report View

Host Name	Device Type	Severity	Device Compliance Status	Execution	Condition Status	Device Id	Vendor	Policy Name	Rule Name	Condition Name
wnacrp-dtss-0-gw.net.disney.com	Cisco CSR 1000V	Critical	●	●	●	172.16.3.30	Cisco Systems	IP_Domain_N...	Check_IP_D...	Verify_IP_Domain...

Since *IP_Domain_Name policy* has a condition named *Verify_IP_Domain_name*, where it didn't meet the required criteria. The condition is marked as *Non_compliant*.

Severity: Severity the condition where the condition Match Action or Non Match Action is of type *Raise_violation* or *Raise_violation_and_continue*.

Upon checking the row you can see the expected and the fix commands for that condition along with action-severity, action-type and other metadata related to device & condition.

Host Name	Device Type	Severity	Device Compliance Status	Execution	Condition Status	Device Id	Vendor	Policy Name	Rule Name	Condition Name
wnacrp-dtss-0-gw.net.disney.com	Cisco CSR 1000V	Critical	●	●	●	172.16.3.30	Cisco Systems	IP_Domain_N...	Check_IP_D...	Verify_IP_Domain...

Compliance details

Information

- Device ID: 172.16.3.30
- Device host name: wnacrp-dtss-0-gw.net.disney.com
- Device Type: Cisco CSR 1000V
- Vendor: Cisco Systems
- Device Compliance Status: Non Compliant
- Execution Status: SUCCESSFUL
- Config Time: Nov 5, 2020, 10:46:16 PM
- Policy Name: IP_Domain_Name
- Rule Name: Check_IP_Domain_Name
- Condition ID: Verify_IP_Domain_name
- Condition Status: Non Compliant

Expected Pattern

Host Name	Device Type	Severity	Device Compliance Status	Execution	Condition Status	Device Id	Vendor	Policy Name	Rule Name	Condition Name
wnacrp-dtss-0-gw.net.disney.com	Cisco CSR 1000V	Critical	●	●	●	172.16.3.30	Cisco Systems	IP_Domain_N...	Check_IP_D...	Verify_IP_Domain...

Compliance details

Information

Expected Pattern

ip domain-name net.disney.com

Action Details

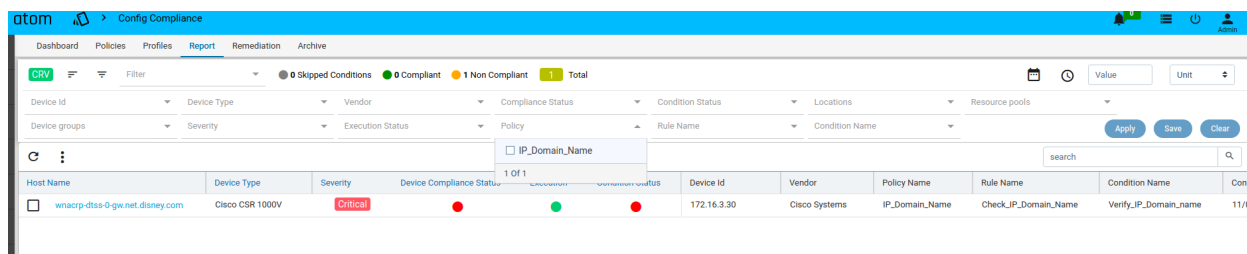
- Action Type: NON_MATCH_ACTION
- Action Severity: CRITICAL

Remediation Commands

```
ip domain-name net.disney.com
```

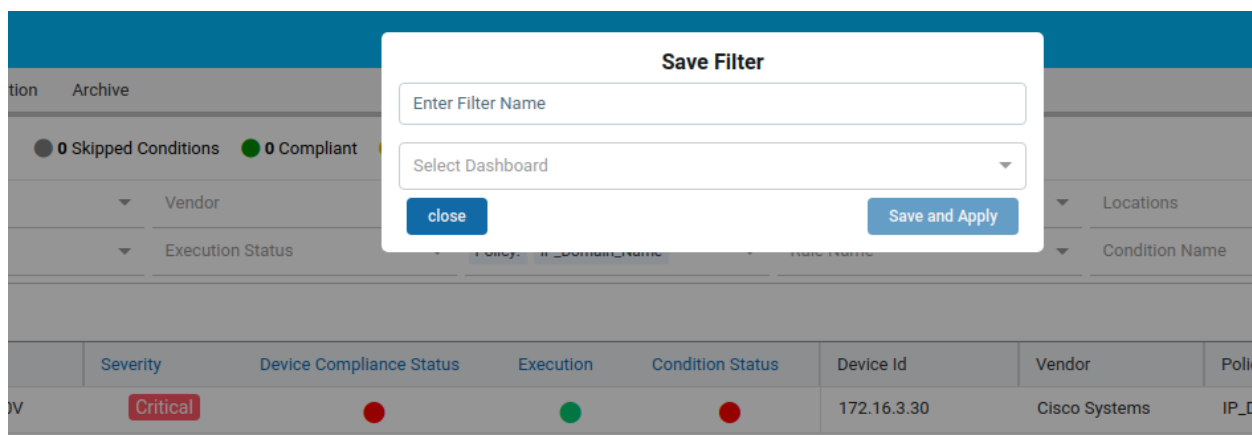
The reports section also facilitates users to filter the results of what user is interested in. The dropdown will display all the possible values for the filters. Users can try out any combination and see the results. By clicking on the **apply** button.

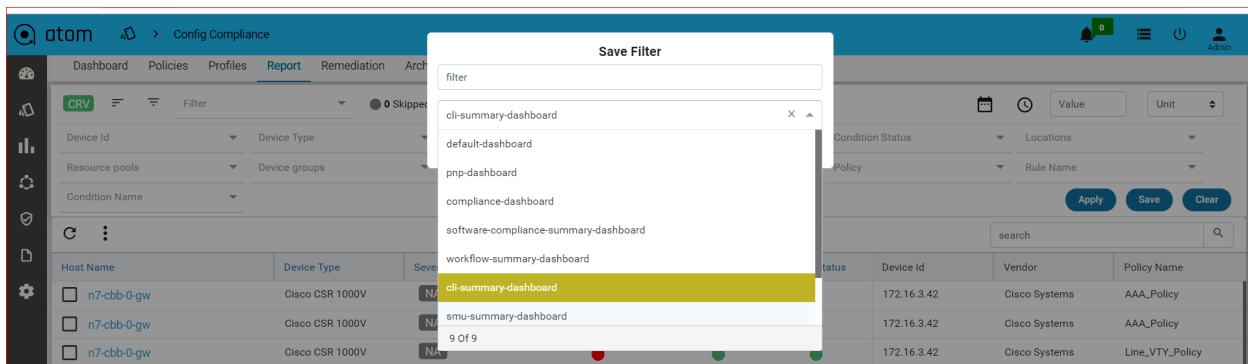
Inorder to revert the filter that are applied you can click on the **clear** button.



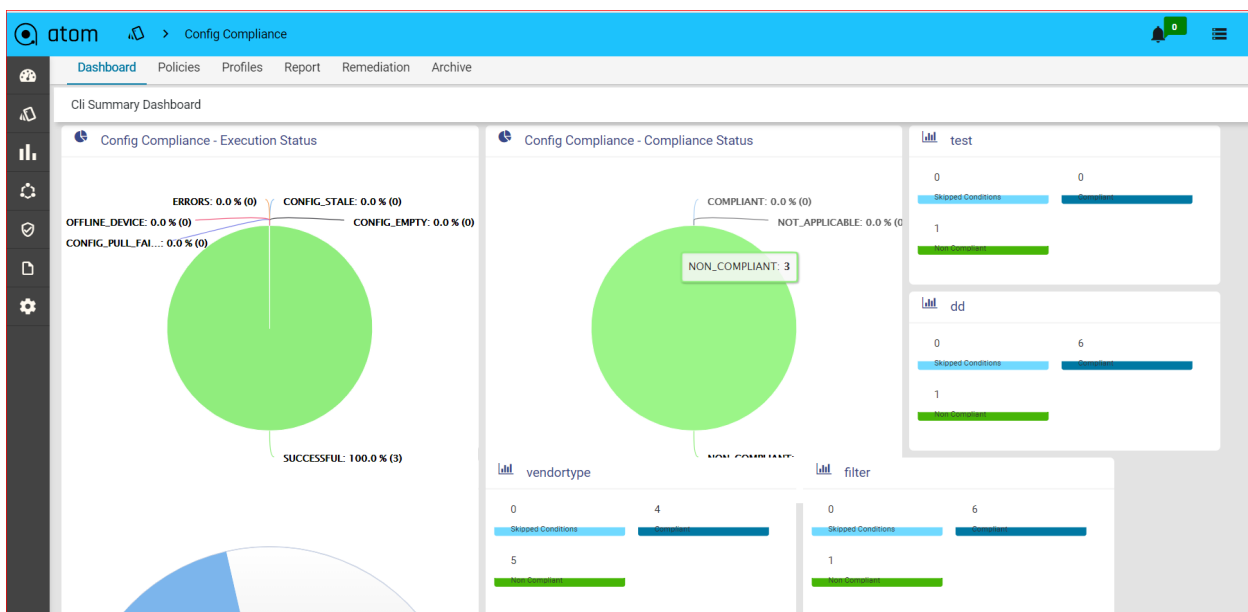
Tired of filtering the results every time for more frequent data. We got you covered ATOM provides an option to save the filter that you applied again with a single click.

- Select filters that you are interested in and click on the **save** button
- This will show a new pop-up box prompting for the filter name and the dashboard where the user wants to pin it.
- Click on the **save and apply** button will save this filter and the resulting data will be populated.



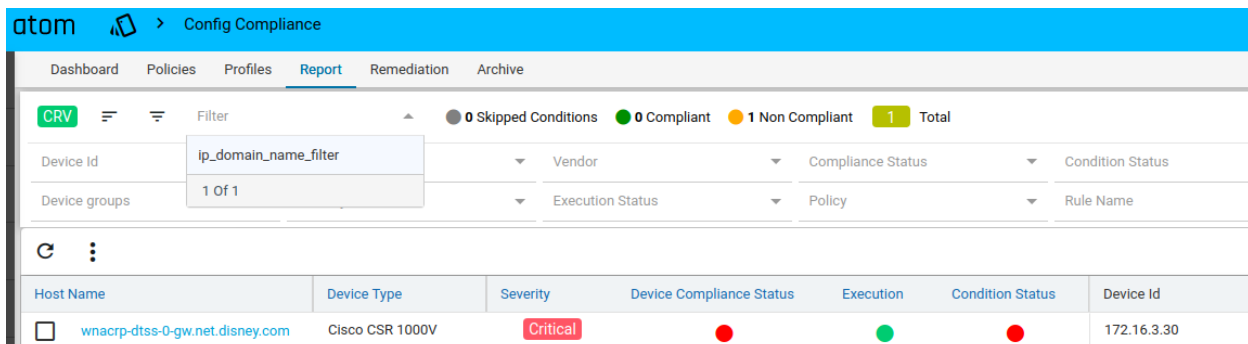


We can pin to the dashboard, upon saving the filter using dropdown. we are able to see the filter under the dashboard.

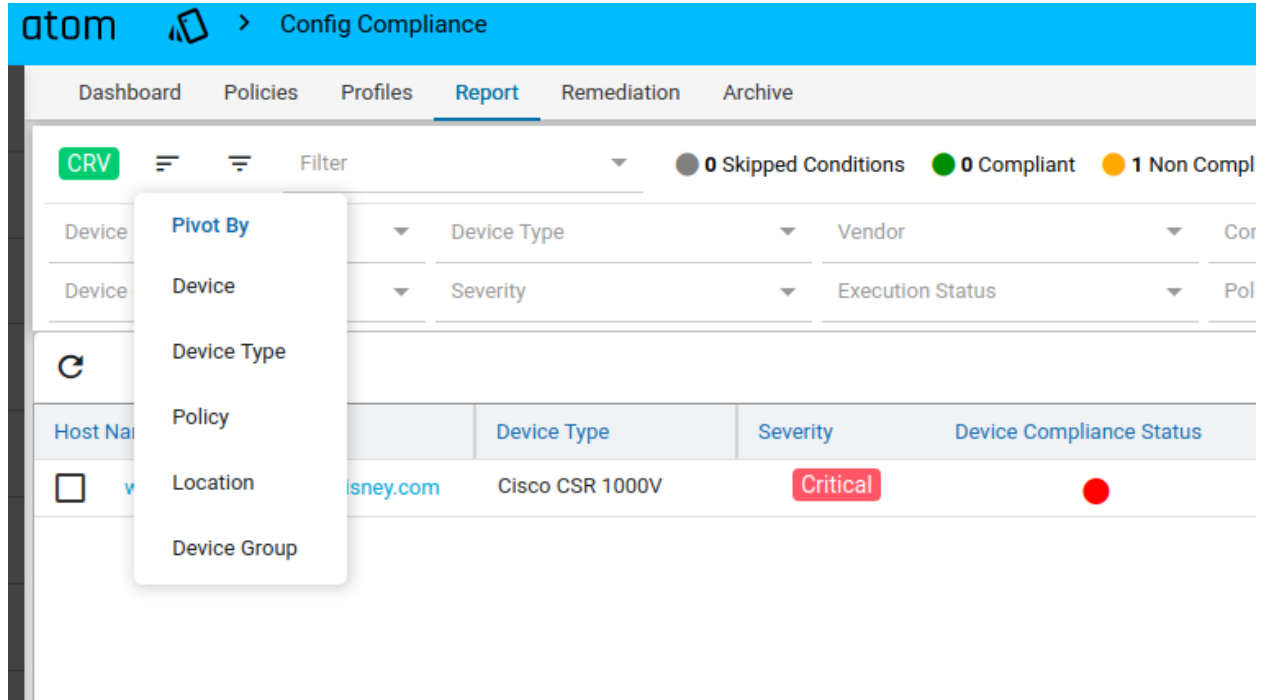


From where I can access these saved filters?

- They are easily accessible. All the filters that are saved are listed under the dropdown on the top.
- Click on the interested filter and you see the data getting filtered.



ATOM also provides an option to view the statistics based on Pivot by Device, Device Type, Policy, Location and Device Group.



Users can opt for any view that they are interested in.

Pivot by device

Device Id	Device Type	Vendor	Compliance Status	Condition Status	Locations	Resource pools
Device groups	Severity	Execution Status	Policy	Rule Name	Condition Name	
Device Compliance Status	Severity	Execution	Host Name	Device ID	Device Type	Compliant Policies
						Non-Compliant Policies
						Compliant Conditions
						Non-Compliant Conditions
						Vendor
<input type="checkbox"/>	Critical	●	wnacrp-dtss-0-gw.net.disney...	172.16.3.30	Cisco CSR 1000V	0
						1
						0
						1
						Cisco Systems

Severity: The aggregated severity of that particular device.

Compliant Policies: The number of policies that are compliant against the device.

Non-Compliant Policies: The number of policies that are non-compliant against the device.

Compliant Conditions: The number of Conditions that are complaint against the device

Non-Compliant Conditions: The number of Condition that are non-compliant against the device

Device ID: Displays all Device Ids for which compliance has run. This is the context column for pivot by device.

Execution status: Based on execution on device it is updated as Successful, errors, stale config, empty config, config pull failed, offline device.

Hostname: Hostname of a device is displayed here.

Device compliance status: Based on compliance run, for a device it is updated as Compliant, non-compliant, Not applicable.

Device Type: The device type of a device is displayed(like cisco csr 1000v , cisco 891).

Vendor: vendors of device are displayed here (like cisco, juniper)

On clicking on the Device ID, Device ID filter gets applied and the user will be navigated to CRV.

Dashboard

Policies

Profiles

Report

Remediation

Archive

CRV

Filter

2 Skipped Conditions

3 Compliant

22 Non Compliant

27 Total

Value

Unit

Device Id: 172.16.22.101

Device Type

Vendor

Compliance Status

Condition Status

Locations

Resource pools

Device groups

Severity

Execution Status

Policy

Rule Name

Condition Name

Apply

Save

Clear

search

Host Name	Severity	Execution	Device Type	Device Compliance Status	Condition Status	Device Id	Vendor	Policy Name	Rule Name	Condition Name
<input type="checkbox"/> csr101.anutanetworks.com	Critical		Cisco CSR 1000V			172.16.22.101	Cisco Systems	ACL_on_Line...	Check_Line...	Verify_Line_VTY
<input type="checkbox"/> csr101.anutanetworks.com	Critical		Cisco CSR 1000V			172.16.22.101	Cisco Systems	Clock_Synchr...	NTP_TEMPL...	Check_NTP_Server
<input type="checkbox"/> csr101.anutanetworks.com	Critical		Cisco CSR 1000V			172.16.22.101	Cisco Systems	Clock_Synchr...	NTP_TEMPL...	Check_NTP_Aso...
<input type="checkbox"/> csr101.anutanetworks.com	Critical		Cisco CSR 1000V			172.16.22.101	Cisco Systems	Clock_Synchr...	NTP_TEMPL...	Check_NTP_ACL
<input type="checkbox"/> csr101.anutanetworks.com	Critical		Cisco CSR 1000V			172.16.22.101	Cisco Systems	Clock_Synchr...	CLOCK_TE...	Check_Summer...
<input type="checkbox"/> csr101.anutanetworks.com	Critical		Cisco CSR 1000V			172.16.22.101	Cisco Systems	Enable_Paas...	Check_pass...	verify_Password...
<input type="checkbox"/> csr101.anutanetworks.com	Critical		Cisco CSR 1000V			172.16.22.101	Cisco Systems	IP_Domain_N...	Check_IP_D...	Verify_IP_Domain...
<input type="checkbox"/> csr101.anutanetworks.com	Critical		Cisco CSR 1000V			172.16.22.101	Cisco Systems	IP_Name_Ser...	Check_IP_N...	IP_Name_Server
<input type="checkbox"/> csr101.anutanetworks.com	NA		Cisco CSR 1000V			172.16.22.101	Cisco Systems	NTP_configur...	NTP_config...	Remove_extra_N...
<input type="checkbox"/> csr101.anutanetworks.com	Critical		Cisco CSR 1000V			172.16.22.101	Cisco Systems	NTP_configur...	NTP_config...	NTP_server_check
<input type="checkbox"/> csr101.anutanetworks.com	NA		Cisco CSR 1000V			172.16.22.101	Cisco Systems	Standard_Aud...	Check_Privil...	verify_VTY
<input type="checkbox"/> csr101.anutanetworks.com	NA		Cisco CSR 1000V			172.16.22.101	Cisco Systems	Standard_Aud...	Check_Privil...	verify_Console
<input type="checkbox"/> csr101.anutanetworks.com	Critical		Cisco CSR 1000V			172.16.22.101	Cisco Systems	Standard_Aud...	Check_vty_s...	verify_session_t...

Pivot By Policy

PV

Policy

Filter

0 Skipped Conditions

0 Compliant

1 Non Compliant

1 Total

Device Id

Device Type

Vendor

Compliance Status

Condition Status

Device groups

Severity

Execution Status

Policy

Rule Name

Compliance Status	Severity	Policy Name	Compliant Devices	Non-Compliant Devices	Non-Compliant Conditions	
<div><div></div><div></div></div>	Critical	IP_Domain_Name	0	1	1	

Severity: The aggregated severity of that particular policy.

Compliant Devices: The number of devices that are compliant against the policy.

Non-Compliant Devices: The number of devices that are non-compliant against the policy.

Non-Compliant Conditions: The number of conditions that are non-compliant against the policy.

Policy Name: Displays all policies for which compliance is run. This is the context column for pivot by policy.

Compliance status: Based on policy, it is updated as compliant or non-compliant.

On clicking on the policy, the policy filter gets applied and the user will be navigated to CRV.

CRV

Filter

0 Skipped Conditions0 Compliant15 Non Compliant15 Total

Value

Unit

Device Id

Device Type

Vendor

Compliance Status

Condition Status

Locations

Resource pools

Device groups

Severity

Execution Status

Policy: ACL_on_Line_VTY

Rule Name

Condition Name

Apply

Save

Clear

C

:

search

Q

Host Name	Device Type	Severity	Device Compliance Status	Execution	Condition Status	Device Id	Vendor	Policy Name	Rule Name	Condition Name
<input type="checkbox"/> aancbb-ana-0-gw	Cisco 871	Critical				172.16.1.138	Cisco Systems	ACL_on_Line...	Check_Line...	Verify_Line_VTY
<input type="checkbox"/> aancbb-ana-1gw.net.dianey.com	Cisco 891	Critical				172.16.1.139	Cisco Systems	ACL_on_Line...	Check_Line...	Verify_Line_VTY
<input type="checkbox"/> car161.anutacorp.com	Cisco CSR 1000V	Critical				172.16.16.161	Cisco Systems	ACL_on_Line...	Check_Line...	Verify_Line_VTY
<input type="checkbox"/> car162.anutacorp.com	Cisco CSR 1000V	Critical				172.16.16.162	Cisco Systems	ACL_on_Line...	Check_Line...	Verify_Line_VTY
<input type="checkbox"/> car163.anutacorp.com	Cisco CSR 1000V	Critical				172.16.16.163	Cisco Systems	ACL_on_Line...	Check_Line...	Verify_Line_VTY
<input type="checkbox"/> car164.anutacorp.com	Cisco CSR 1000V	Critical				172.16.16.164	Cisco Systems	ACL_on_Line...	Check_Line...	Verify_Line_VTY
<input type="checkbox"/> ana-buf-4-gw	CiscoIOSXRv9000	Critical				172.16.17.133	Cisco Systems	ACL_on_Line...	Check_Line...	Verify_Line_VTY
<input type="checkbox"/> loasn-6.5.1-134	CiscoIOSXRv9000	Critical				172.16.17.134	Cisco Systems	ACL_on_Line...	Check_Line...	Verify_Line_VTY
<input type="checkbox"/> asr17_135	CiscoIOSXRv9000	Critical				172.16.17.135	Cisco Systems	ACL_on_Line...	Check_Line...	Verify_Line_VTY
<input type="checkbox"/> eor-cbb-2-gw.net.dianey.com	CiscoIOSXRv9000	Critical				172.16.18.176	Cisco Systems	ACL_on_Line...	Check_Line...	Verify_Line_VTY
<input type="checkbox"/> car101.anutanetworks.com	Cisco CSR 1000V	Critical				172.16.22.101	Cisco Systems	ACL_on_Line...	Check_Line...	Verify_Line_VTY
<input type="checkbox"/> car102.anutacorp.com	Cisco CSR 1000V	Critical				172.16.22.102	Cisco Systems	ACL_on_Line...	Check_Line...	Verify_Line_VTY

Pivot By Device Type:

Severity:The aggregated severity of that particular device type

Device type: Displays all device types available in compliance. This is the context column for pivot by device type.

Vendor: Displays the vendors available in compliance .

Compliant Device: The number of devices that are compliant against the device type

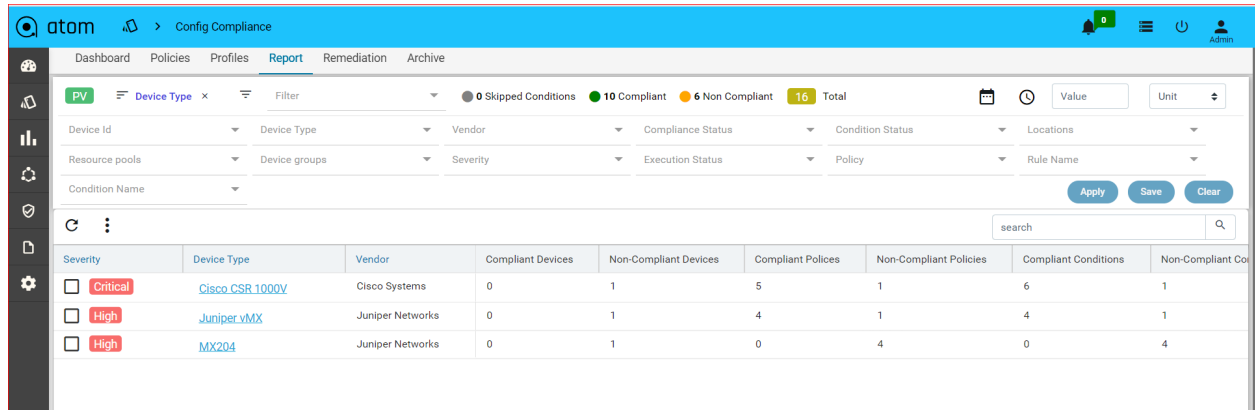
Non-compliant device:The number of devices that are non-compliant against the device type.

Compliant policies:The number of policies that are compliant against the device type

Non-Compliant policies:The number of policies that are non-compliant against the device type

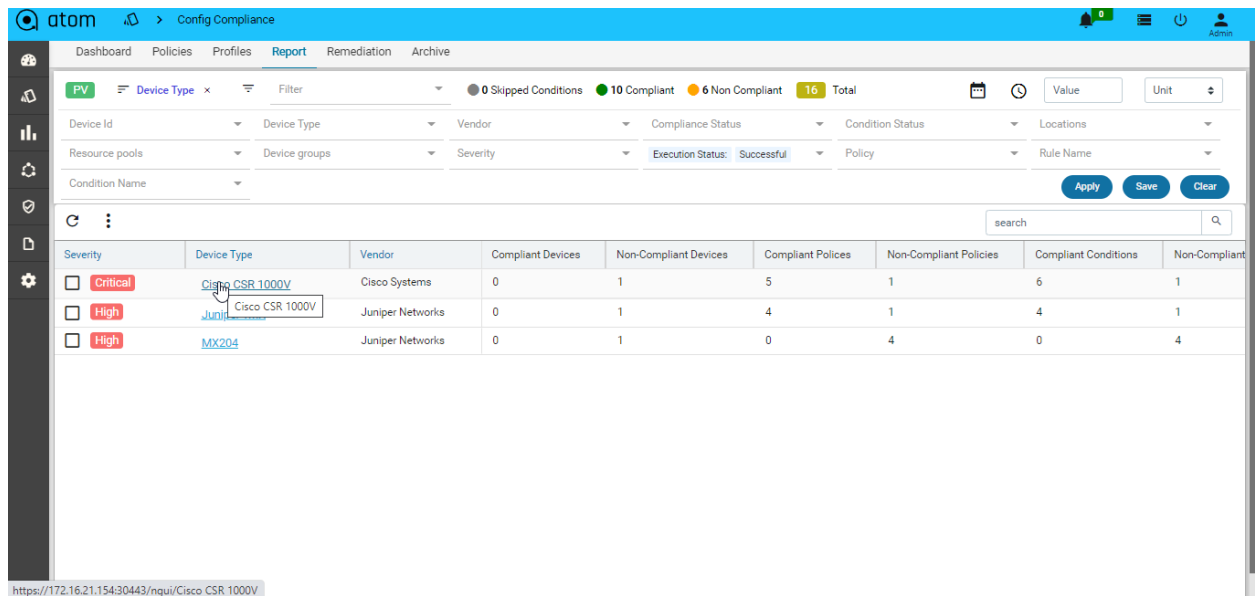
Compliant Condition:The number of conditions that are compliant against the device type.

Non-compliant Condition:The number of conditions that are non-compliant against the device type.



Severity	Device Type	Vendor	Compliant Devices	Non-Compliant Devices	Compliant Policies	Non-Compliant Policies	Compliant Conditions	Non-Compliant Conditions
Critical	Cisco CSR 1000V	Cisco Systems	0	1	5	1	6	1
High	Juniper vMX	Juniper Networks	0	1	4	1	4	1
High	MX204	Juniper Networks	0	1	0	4	0	4

Click on device type> Cisco CSR 1000v here



Severity	Device Type	Vendor	Compliant Devices	Non-Compliant Devices	Compliant Policies	Non-Compliant Policies	Compliant Conditions	Non-Compliant Conditions
Critical	Cisco CSR 1000V	Cisco Systems	0	1	5	1	6	1
High	Juniper vMX	Juniper Networks	0	1	4	1	4	1
High	MX204	Juniper Networks	0	1	0	4	0	4

On clicking on the Device Type, Device Type filter gets applied and the user will be navigated to CRV.

Host Name	Device Type	Severity	Device Compliance Status	Execution	Condition Status	Device Id	Vendor	Policy Name
<input type="checkbox"/> n7-cbb-0-gw	Cisco CSR 1000V	NA	●	●	●	172.16.3.42	Cisco Systems	AAA_Policy
<input type="checkbox"/> n7-cbb-0-gw	Cisco CSR 1000V	NA	●	●	●	172.16.3.42	Cisco Systems	AAA_Policy
<input type="checkbox"/> n7-cbb-0-gw	Cisco CSR 1000V	NA	●	●	●	172.16.3.42	Cisco Systems	Line_VTY_Policy
<input type="checkbox"/> n7-cbb-0-gw	Cisco CSR 1000V	NA	●	●	●	172.16.3.42	Cisco Systems	Logging_Policy
<input type="checkbox"/> n7-cbb-0-gw	Cisco CSR 1000V	NA	●	●	●	172.16.3.42	Cisco Systems	NTP_server_cisco_policy
<input type="checkbox"/> n7-cbb-0-gw	Cisco CSR 1000V	NA	●	●	●	172.16.3.42	Cisco Systems	SNMP_server_cisco_Policy
<input type="checkbox"/> n7-cbb-0-gw	Cisco CSR 1000V	Critical	●	●	●	172.16.3.42	Cisco Systems	Cisco_domain_name_policy

Pivot By Location:

Severity	Location Name	Compliant Devices	Non-Compliant Devices	Compliant Policies	Non-Compliant Policies	Compliant Conditions	Non-Compliant Conditions
<input type="checkbox"/> Critical	india_loc	0	1	5	1	6	1

Severity: The aggregated severity of that particular locations

Location Name: Displays all available locations over which compliance is run.This is the context column for pivot by location.

Compliant Devices: The number of devices that are compliant against locations

Non-Compliant Devices: The number of devices that are non-compliant against locations

Compliant Policies: The number of policies that are compliant against locations

Non-compliant Policies: The number of policies that are non-compliant against locations

Compliant Condition: The number of conditions that are compliant against locations

Non-Compliant Condition: The number of conditions that are non- compliant against locations

Click on any of location name

Dashboard	Policies	Profiles	Report	Remediation	Archive						
<div> <div>CRV</div> <div>Filter</div> <div> <div>206 Skipped Conditions</div> <div>108 Compliant</div> <div>635 Non Compliant</div> <div>949 Total</div> </div> <div>Value</div> <div>Unit</div> </div>											
Device Id	Device Type	Vendor	Compliance Status	Condition Status	Locations: location_california						
Device groups	Severity	Execution Status	Policy	Rule Name	Condition Name						
					<div>Apply</div> <div>Save</div> <div>Clear</div>						
<div> <div>C</div> <div>:</div> <div>search</div> <div>Q</div> </div>											
Host Name	Device Type	Severity	Device Compliance Status	Execution	Condition Status	Device Id	Vendor	Policy Name	Rule Name	Condition Name	Configuration
<input type="checkbox"/> Cisco2951	Cisco 2951		NOT_APPLICABLE	CONFIG_E...		10.1.1.1					11/05/20, 12:
<input type="checkbox"/> Cisco3845	Cisco 3845		NOT_APPLICABLE	CONFIG_E...		10.1.1.2					11/05/20, 12:
<input type="checkbox"/> cat385024P-2	cat385024P		NOT_APPLICABLE	CONFIG_E...		10.1.1.3					11/05/20, 12:
<input type="checkbox"/> ciscoAircT3508K9	ciscoAircT3508K9		NOT_APPLICABLE	CONFIG_E...		10.1.1.4					11/05/20, 12:
<input type="checkbox"/> cat385024P-3	cat385024P		NOT_APPLICABLE	CONFIG_E...		10.1.1.5					11/05/20, 10:
<input type="checkbox"/> cat385024P-1	cat385024P		NOT_APPLICABLE	CONFIG_E...		10.1.1.6					11/05/20, 12:
<input type="checkbox"/> Cisco Virtual ASA	Cisco Virtual ASA		NOT_APPLICABLE	CONFIG_E...		10.1.15.21					11/05/20, 12:
<input type="checkbox"/> ciscoASA5510	ciscoASA5510		NOT_APPLICABLE	CONFIG_E...		10.1.15.23					11/05/20, 12:
<input type="checkbox"/> Cisco Nexus 7004	Cisco Nexus 7004		NOT_APPLICABLE	CONFIG_E...		10.1.15.25					11/05/20, 12:
<input type="checkbox"/> Cisco Nexus 5010 Switch	Cisco Nexus 5010 S...		NOT_APPLICABLE	CONFIG_E...		10.1.15.27					11/05/20, 12:
<input type="checkbox"/> Cisco Nexus 3064 Switch	Cisco Nexus 3064 S...		NOT_APPLICABLE	CONFIG_E...		10.1.15.29					11/05/20, 12:
<input type="checkbox"/> Cisco ASR 9006	Cisco ASR 9006		NOT_APPLICABLE	CONFIG_E...		10.1.15.31					11/05/20, 12:
<input type="checkbox"/> N/A	APIC		NOT_APPLICABLE	CONFIG_E...		10.1.15.34					11/05/20, 12:
<input type="checkbox"/> CiscoISR4331	CiscoISR4331		NOT_APPLICABLE	CONFIG_E...		10.1.2.1					11/05/20, 12:

Adding the same device in two resource pools and associate one RP with location and another RP with no location, only location associated with RP will be listed.

Pivot By Device Group:

Severity	Device Group	Compliant Devices	Non-Compliant Devices	Compliant Policies	Non-Compliant Policies	Compliant Conditions	Non-Compliant Conditions
High	Layer 2/3 switch	0	2	4	5	4	5
High	Layer 2 switch	0	2	4	5	4	5
High	Layer 3 Router	0	2	4	5	4	5
High	OfflineDevices	0	2	9	2	10	2

Severity: The aggregated severity of that particular Device group

Device Group: Displays all available device groups over which compliance is run. This is the context column for pivot by device group.

Devices in group: This gives the number of devices in a group

Compliant Devices: The number of devices that are compliant against device groups

Non-Compliant Devices: The number of devices that are non-compliant against device groups

Compliant Policies: The number of policies that are compliant against device groups

Non-compliant Policies: The number of policies that are non-compliant against device groups

Compliant Condition: The number of conditions that are compliant against device groups

Non-Compliant Condition: The number of conditions that are non-compliant against device groups.

Click on any of device groups

Severity	Compliant Devices	Non-Compliant Devices	Device Group	Compliant Policies	Non-Compliant Policies	Compliant Conditions	Non-Compliant Conditions
NA	0	3	Layer 2 switch	1	27	9	51
NA	0	38	Layer 3 Router	4	78	103	601
NA	0	2	Firewall	0	12	5	35
NA	0	6	VPN	0	29	17	122
NA	0	7	Layer 2/3 switch	3	44	22	96
NA	0	1	OfflineDevices	0	11	3	21
NA	0	0	gre1	0	0	0	0
Critical	0	38	cisco_gre	4	83	105	613
Critical	0	1	Host	0	7	2	17
NA	0	0	gre1_device	0	0	0	0

On clicking on the device group, the device group filter gets applied and the user will be navigated to CRV.

CRV

Filter

18 Skipped Conditions

9 Compliant

51 Non Compliant

78 Total

Value

Unit

Device Id

Device Type

Vendor

Compliance Status

Condition Status

Locations: location_california

Resource pools

Device groups: Layer 2 switch

Severity

Execution Status

Policy

Rule Name

Condition Name

Apply

Save

Clear

C

:

search

Q

Host Name	Device Type	Severity	Device Compliance Status	Execution	Condition Status	Device Id	Vendor	Policy Name	Rule Name
<input type="checkbox"/> cat385024P-2	cat385024P		NOT_APPLICABLE	CONFIG_EMPTY		10.1.1.3			
<input type="checkbox"/> cat385024P-3	cat385024P		NOT_APPLICABLE	CONFIG_EMPTY		10.1.1.5			
<input type="checkbox"/> cat385024P-1	cat385024P		NOT_APPLICABLE	CONFIG_EMPTY		10.1.1.6			
<input type="checkbox"/> aancbb-ana-0-gw	Cisco 871	NA				172.16.1.138	Cisco Systems	Enable_Password_Encryption	Check_L
<input type="checkbox"/> aancbb-ana-0-gw	Cisco 871	Critical				172.16.1.138	Cisco Systems	ACL_on_Line_VTY	Check_L
<input type="checkbox"/> aancbb-ana-0-gw	Cisco 871	Critical				172.16.1.138	Cisco Systems	Clock_Synchronization	NTP_TE
<input type="checkbox"/> aancbb-ana-0-gw	Cisco 871	Critical				172.16.1.138	Cisco Systems	Clock_Synchronization	NTP_TE
<input type="checkbox"/> aancbb-ana-0-gw	Cisco 871	Critical				172.16.1.138	Cisco Systems	Clock_Synchronization	NTP_TE
<input type="checkbox"/> aancbb-ana-0-gw	Cisco 871	Critical				172.16.1.138	Cisco Systems	Clock_Synchronization	CLOCK_
<input type="checkbox"/> aancbb-ana-0-gw	Cisco 871	Critical				172.16.1.138	Cisco Systems	IP_Domain_Name_Global	Check_L
<input type="checkbox"/> aancbb-ana-0-gw	Cisco 871	Critical				172.16.1.138	Cisco Systems	IP_Name_Server_Global	Check_L
<input type="checkbox"/> aancbb-ana-0-gw	Cisco 871	NA				172.16.1.138	Cisco Systems	NTP_configurations_Global	NTP_co

pivot view :

- Single pivot view can be selected.(no multi pivot view selection)
- Upon selecting a pivot view, further filters like device id, device type, vendor, compliance status,condition status,location,resource pool, device groups,severity,execution status,policy, rule name, condition name can be applied.
- Just with the pivot views, a filter can't be saved, Based on further filters applied, a filter can be saved. The saved filter can be deleted.
- Bulk delete is not supported in pivot views
- Remediation is not supported
- Sorting is not supported on any column in pivot view.
- Searching is not supported in pivot view.
- The counts on top are related to the non- pivot views and labels are from condition status
- Export: based on pivot views, the records can be exported.

CRV(conditional report view) :

Dashboard

Policies

Profiles

Report

Remediation

Archive

CRV

Filter

0 Skipped Conditions

12 Compliant

2 Non Compliant

14 Total

Value

Unit

Device Id

Device Type

Vendor

Compliance Status

Condition Status

Locations

Resource pools

Device groups

Severity

Execution Status

Policy

Rule Name

Condition Name

Apply

Save

Clear

search

Q

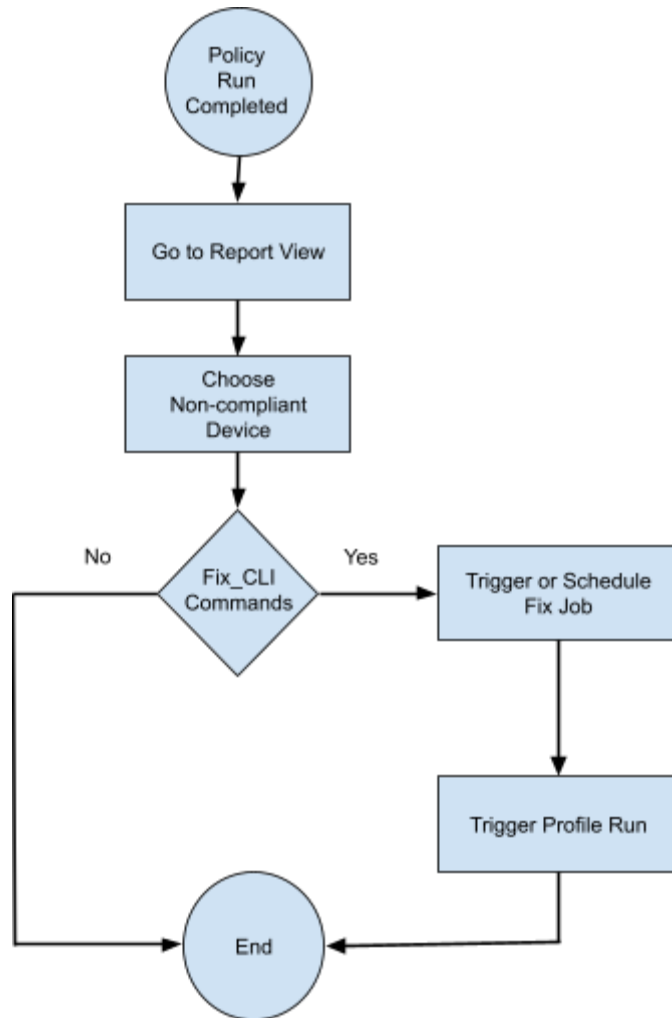
Host Name	Device Type	Severity	Device Compliance Status	Execution	Condition Status	Device Id	Vendor	Policy Name	Rule Name	Condition Name
<input type="checkbox"/> aancbb-ana-1gw.net.danay.com	Cisco 891	NA				172.16.1.139	Cisco Systems	ACL_Global	Check_ACL_Configuration	aancmpm3_cnmbs
<input type="checkbox"/> aancbb-ana-1gw.net.danay.com	Cisco 891	NA				172.16.1.139	Cisco Systems	ACL_Global	Check_ACL_Configuration	aancmpm3_cnmbs
<input type="checkbox"/> aancbb-ana-1gw.net.danay.com	Cisco 891	NA				172.16.1.139	Cisco Systems	ACL_Global	Check_ACL_Configuration	aancmpm3_cnmbs
<input type="checkbox"/> aancbb-ana-1gw.net.danay.com	Cisco 891	NA				172.16.1.139	Cisco Systems	ACL_Global	Check_ACL_Configuration	aancmpm3_cnmbs
<input type="checkbox"/> aancbb-ana-1gw.net.danay.com	Cisco 891	NA				172.16.1.139	Cisco Systems	ACL_Global	Check_ACL_Configuration	aancmpm3_cnmbs
<input type="checkbox"/> aancbb-ana-1gw.net.danay.com	Cisco 891	NA				172.16.1.139	Cisco Systems	ACL_Global	Check_ACL_Configuration	aancmpm3_cnmbs
<input type="checkbox"/> aancbb-ana-1gw.net.danay.com	Cisco 891	Critical				172.16.1.139	Cisco Systems	ACL_Global	Check_ACL_Configuration	aancmpm3_cnmbs
<input type="checkbox"/> ana-buf-1-gw.net.danay.com	Cisco CSR 1000V	NA				172.16.3.45	Cisco Systems	ACL_Global	Check_ACL_Configuration	aancmpm3_cnmbs
<input type="checkbox"/> ana-buf-1-gw.net.danay.com	Cisco CSR 1000V	NA				172.16.3.45	Cisco Systems	ACL_Global	Check_ACL_Configuration	aancmpm3_cnmbs
<input type="checkbox"/> ana-buf-1-gw.net.danay.com	Cisco CSR 1000V	NA				172.16.3.45	Cisco Systems	ACL_Global	Check_ACL_Configuration	aancmpm3_cnmbs
<input type="checkbox"/> ana-buf-1-gw.net.danay.com	Cisco CSR 1000V	NA				172.16.3.45	Cisco Systems	ACL_Global	Check_ACL_Configuration	aancmpm3_cnmbs
<input type="checkbox"/> ana-buf-1-gw.net.danay.com	Cisco CSR 1000V	NA				172.16.3.45	Cisco Systems	ACL_Global	Check_ACL_Configuration	aancmpm3_cnmbs
<input type="checkbox"/> ana-buf-1-gw.net.danay.com	Cisco CSR 1000V	NA				172.16.3.45	Cisco Systems	ACL_Global	Check_ACL_Configuration	aancmpm3_cnmbs
<input type="checkbox"/> ana-buf-1-gw.net.danay.com	Cisco CSR 1000V	Critical				172.16.3.45	Cisco Systems	ACL_Global	Check_ACL_Configuration	aancmpm3_cnmbs

- The available columns are: Hostname, Severity, Execution, Device Type, Device compliance status, condition status, Device id, Vendor, Policy name, Rule name, Condition name, Configuration retrieved at, Expected pattern, Enforcement time.
- The data can be filtered by applying filters on device id, device type, vendor, compliance status, condition-status, location, resourcepool, device groups, severity, execution status, policy, rule name and condition name.
- Time based filtering : For timing based, user can use value and units(Days, weeks, months, hours, minutes) fields.
Date wise filter: User can choose the date from the calendar symbol and click on 'apply' in the calendar.
- Count band - Represents counts for skipped, compliant, and non compliant conditions
 - Skipped conditions - Platform mismatch and Execution Status (Stale Config, Empty Config, Erros, Config Pull Failed, Offline Device) fall into this category.
 - Complaint conditions - Based on conditions meeting the criteria
 - Non- compliant conditions -Based on conditions meeting the criteria and violations chosen
- Multi-selection on filters is supported - More than one entry for a given filter can be selected.
- Sorting is enabled on all columns.
- Searching is enabled for all columns.
- The record details are listed when the checkbox for a given entry is selected. Device ID, Device host name, Device type, Vendor, Device compliance status, Execution status, Config time, Policy name, Rule name, Condition ID, Condition status, Expected pattern, Action Details, Remediation commands(if it's non-compliant) are shown as part of details.

Remediation :

Navigate to Resource Manager > Config Compliance -> Remediation

Fix CLI Action will generate the remediation CLI to be applied for each non-compliant device. Users can schedule remediation on one or more devices or execute it right away. For each device selected, ATOM will push remediation CLI to the device.



- a. Select a Report and click on the highlighted arrow for navigating to the Remediation screen.
Only if it is non- compliant, user will be taken to the remediation screen.

CRV

Filter

0 Skipped Conditions

0 Compliant

1 Non Compliant

1 Total

Device Id

Device Type

Vendor

Compliance Status

Condition Status

Device groups

Severity

Execution Status

Policy

Rule Name

Host Name	Severity	Device Type	Device Compliance Status	Condition Status	Execution	Device Id
<div>✓</div> wnacrp-dtss-0-gw.net.disney.com	Critical	Cisco CSR 1000V	<div></div>	<div></div>	<div></div>	172.16.3.30

- b. Fix Violations by providing a Job name and verify rule-input values and fix CLI commands under *Fix Configurations*. Click on the tick button to complete fix-job.

Fix Violations

Job name

fix_ip_domain_name

Devices

▶ 172.16.3.46

No Rule Inputs

Fix Configurations

ip domain-name anutacorp.com

c. Fix Job can be scheduled using the Schedule option or can be initiated immediately by enabling Start now. And click on the Tick button to initiate a fix-job.

Schedule

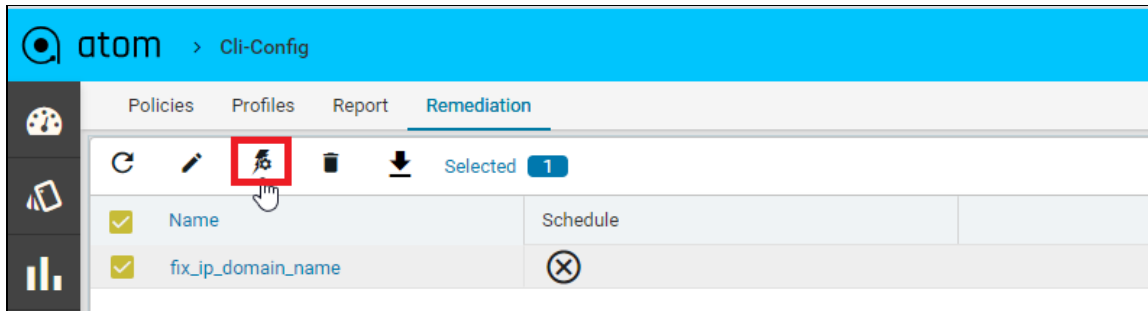
0

0

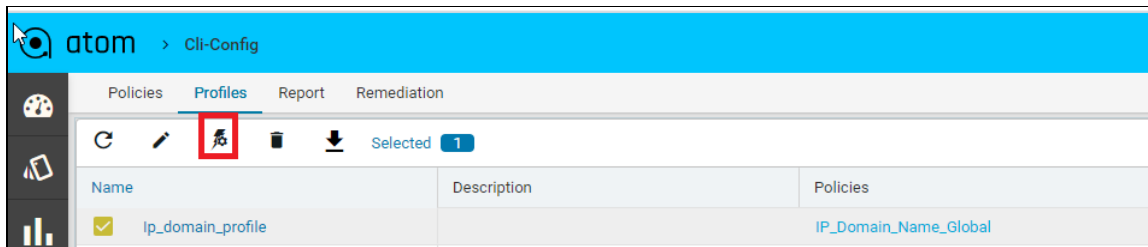
☒ Start now

Hours Minutes

1. Fix job can be initiated from the Remediation tab as well at a later point of time.



d. After remediation, trigger profile job and validate the report to see if the device came back as a complaint.



e. Below we can see the device is now back to complaint after fixing violations.

Host Name	Device Type	Severity	Device Compliance Status	Execution	Condition Status	Device Id	Vendor	Policy Name
<input type="checkbox"/> wncrp-dtss-0-gw.net.disney.com	Cisco CSR 1000V	NA	●	●	●	172.16.3.30	Cisco Systems	IP_Domain_N...

Remediation is not supported when the user is in pivot view.

Bulk delete:

Navigate to Resource Manager > Config Compliance -> reports(CRV)

Host Name	Device Type	Severity	Device Compliance Status	Execution	Condition Status	Device Id	Vendor	Policy Name
<input type="checkbox"/> aancbb-ana-1gw.net.disney.com	Cisco 891	NA	●	●	●	172.16.1.139	Cisco Systems	ACL_Global
<input type="checkbox"/> aancbb-ana-1gw.net.disney.com	Cisco 891	NA	●	●	●	172.16.1.139	Cisco Systems	ACL_Global
<input type="checkbox"/> aancbb-ana-1gw.net.disney.com	Cisco 891	NA	●	●	●	172.16.1.139	Cisco Systems	ACL_Global
<input type="checkbox"/> aancbb-ana-1gw.net.disney.com	Cisco 891	NA	●	●	●	172.16.1.139	Cisco Systems	ACL_Global
<input type="checkbox"/> aancbb-ana-1gw.net.disney.com	Cisco 891	NA	●	●	●	172.16.1.139	Cisco Systems	ACL_Global
<input type="checkbox"/> aancbb-ana-1gw.net.disney.com	Cisco 891	NA	●	●	●	172.16.1.139	Cisco Systems	ACL_Global
<input type="checkbox"/> aancbb-ana-1gw.net.disney.com	Cisco 891	NA	●	●	●	172.16.1.139	Cisco Systems	ACL_Global

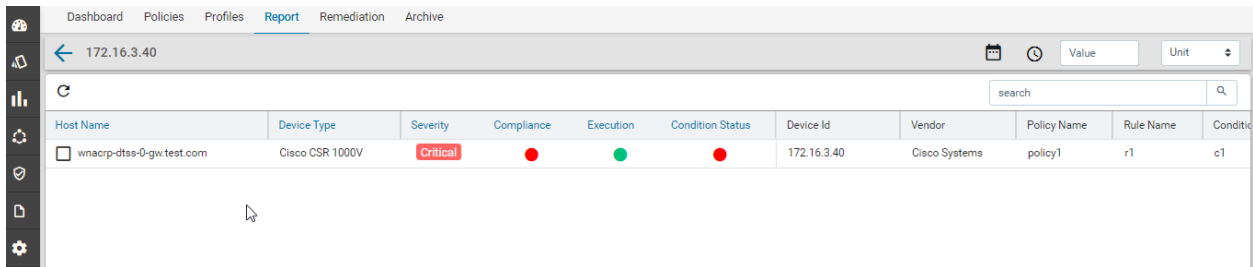
In order to delete the records at shot bulk delete is used.

Based on filters applied, records can be deleted

- If the filter is on Device(device id/ device group/ device Type) And invoke Bulk delete, all the records related to devices are deleted.
- If the filter is on Policy and Bulk delete is invoked,all the records Related to policies across all devices are deleted.(columns like Policy, rule, condition, expected pattern, enforcement time- empty)
- If the filter is on device + policy and bulk delete is invoked, all the records related to policies across all devices are deleted but Device is not deleted.(columns like Policy, rule, condition, expected pattern, enforcement time- empty)
- If the filter is policy+rule+condition, where there are many rules or many conditions, records are deleted at policy level only.(records having the selected policies are deleted irrespective of rule or condition)

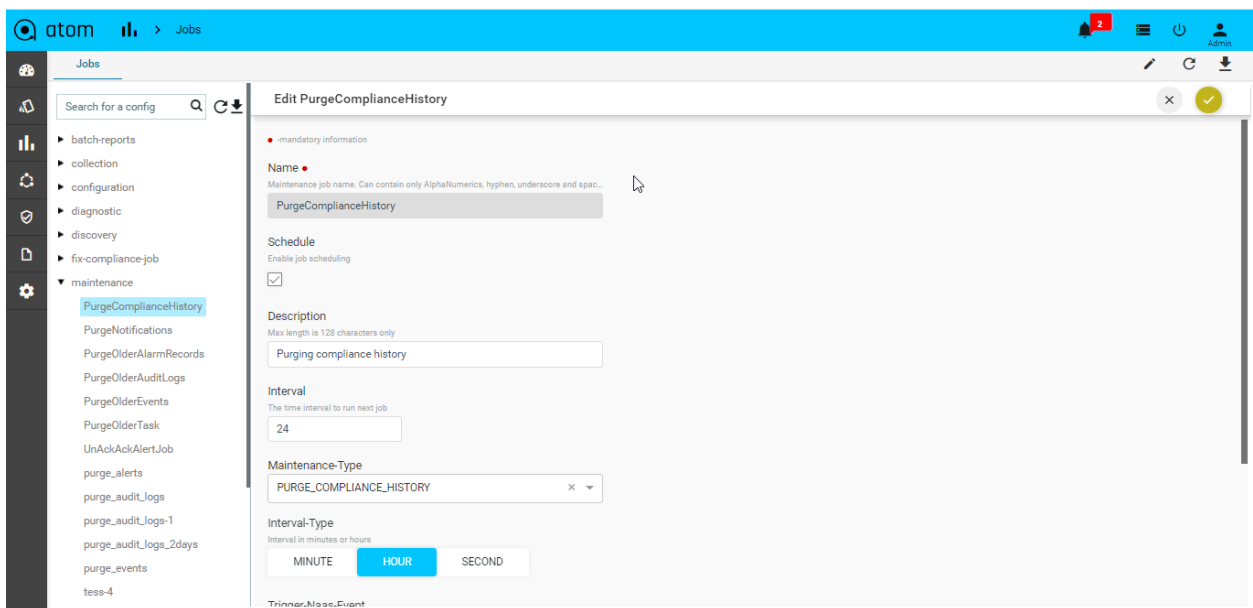
Purge compliance history:

In order to delete history details, we can use purge compliance history under Monitoring--> jobs→ Maintenance→ purge compliance history..



The screenshot shows the 'Report' tab in the ATOM web interface. A table displays compliance data for a device named 'wnacrp-dtss-0-gw.test.com'. The table has columns for Host Name, Device Type, Severity, Compliance, Execution, Condition Status, Device Id, Vendor, Policy Name, Rule Name, and Condition. The 'Severity' is 'Critical', 'Compliance' is represented by a red dot, 'Execution' by a green dot, and 'Condition Status' by a red dot.

Host Name	Device Type	Severity	Compliance	Execution	Condition Status	Device Id	Vendor	Policy Name	Rule Name	Condition
wnacrp-dtss-0-gw.test.com	Cisco CSR 1000V	Critical	●	●	●	172.16.3.40	Cisco Systems	policy1	r1	c1



The screenshot shows the 'Edit PurgeComplianceHistory' configuration page. The left sidebar lists various jobs under the 'maintenance' category, with 'PurgeComplianceHistory' selected. The main form contains the following fields:

- Name:** A text field containing 'PurgeComplianceHistory'.
- Schedule:** A checkbox labeled 'Enable job scheduling' which is checked.
- Description:** A text field containing 'Purging compliance history'.
- Interval:** A text field containing '24'.
- Maintenance-Type:** A dropdown menu set to 'PURGE_COMPLIANCE_HISTORY'.
- Interval-Type:** Radio buttons for 'MINUTE', 'HOUR' (selected), and 'SECOND'.

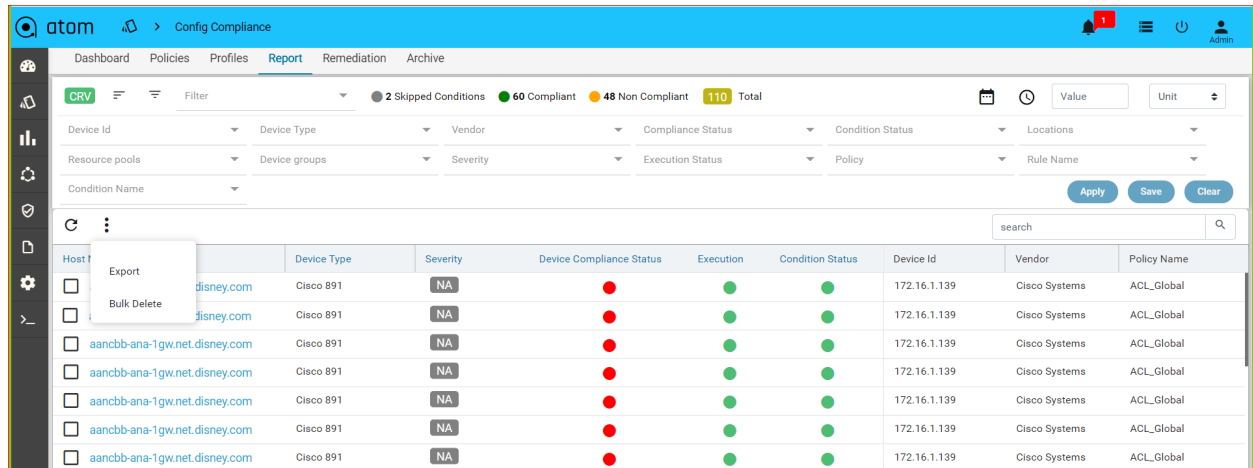
Say, the threshold is 60 and it is set for days. When this job is run, it deletes all the record history details that are older than 60 days.

It can be set in hours too.

Export:

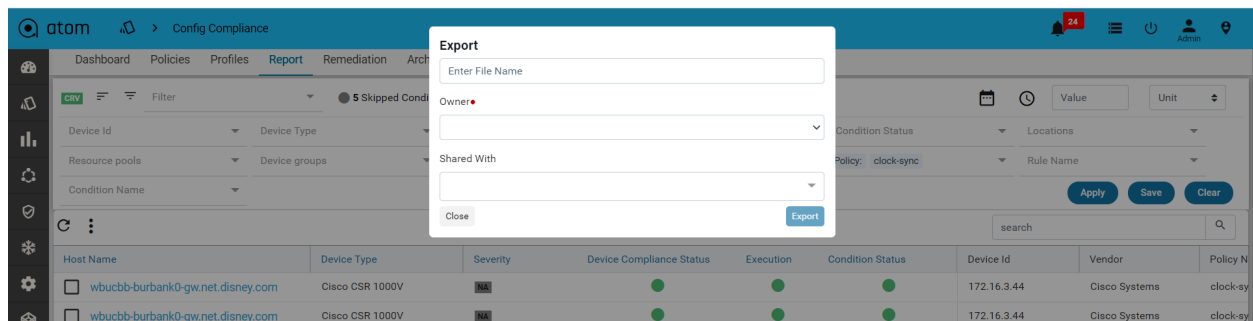
Navigate to Resource Manager > Config Compliance -> reports(CRV/Pivot)

Using export will get report for all the records at one shot



Upon applying filters and the user does export, then the user can get only those records. Without applying filters, if the user does export, all records are exported.

Provide a name to export file to be saved



Export file will be saved in the archive tab.

Archive:

Navigate to Resource Manager > Config Compliance -> Archive

From here we can download the report and delete as well. File download format is CSV.

The headers of the downloaded csv report are according to the filter applied.
If pivot views are applied, the headers are according to it.

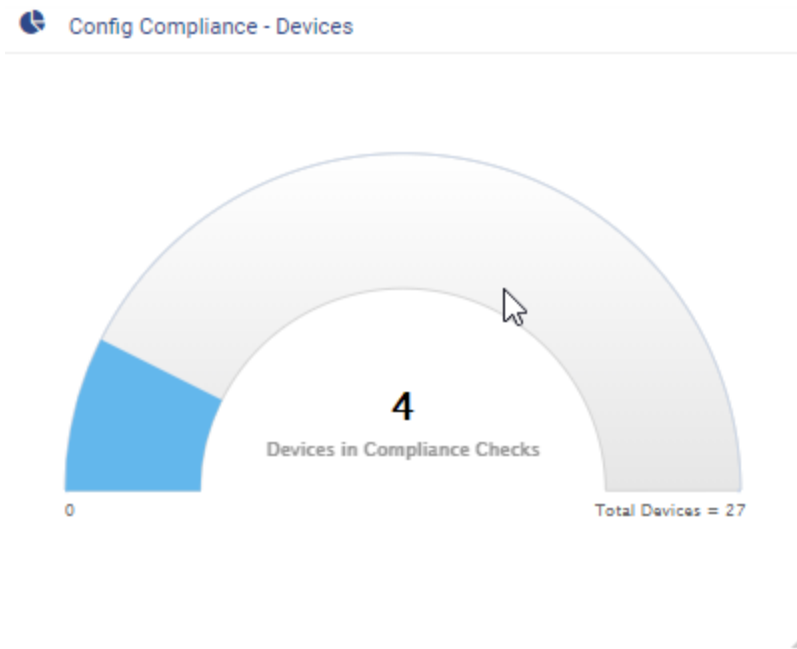
Dashboard Policies Profiles Report Remediation Archive				
↻				
Name	ReportType	Size	Owner	Shared With
<input type="checkbox"/> ConditionView_April 7	CONDITIONS_REPORT	900 bytes	coke	coke.*
<input type="checkbox"/> Device Report	DEVICE_SUMMARY	352 bytes	coke	coke.*
<input type="checkbox"/> Location Report	LOCATION_SUMMARY	242 bytes	coke	coke.*,coke.north

atom > Config Compliance				
Dashboard Policies Profiles Report Remediation Archive				
↻ ⬇️ 🗑️ Selected 1				
Name	ReportType	Size	Owner	Shared With
<input type="checkbox"/> ConditionView_April 7	CONDITIONS_REPORT	900 bytes	coke	coke.*
<input checked="" type="checkbox"/> Device Report	DEVICE_SUMMARY	352 bytes	coke	coke.*
<input type="checkbox"/> Location Report	LOCATION_SUMMARY	242 bytes	coke	coke.*,coke.north

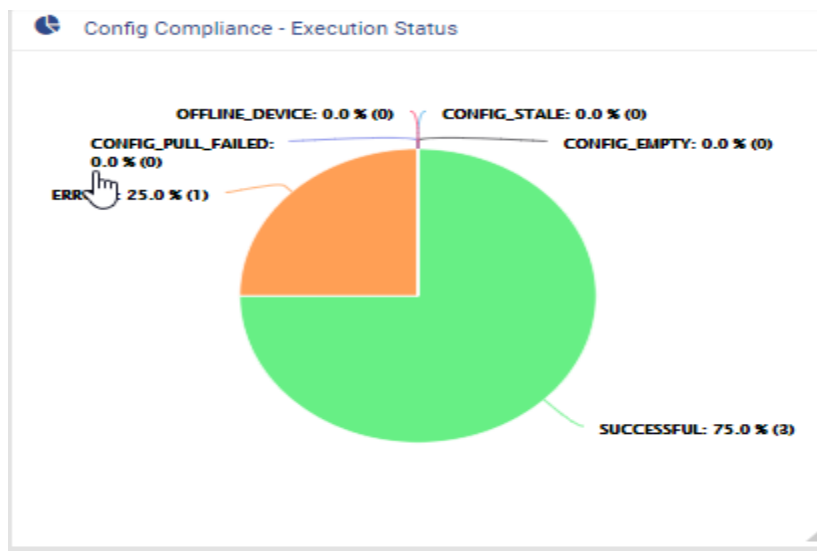
Dashboard

There are 5 Dashboards which gives a quick information about compliance status

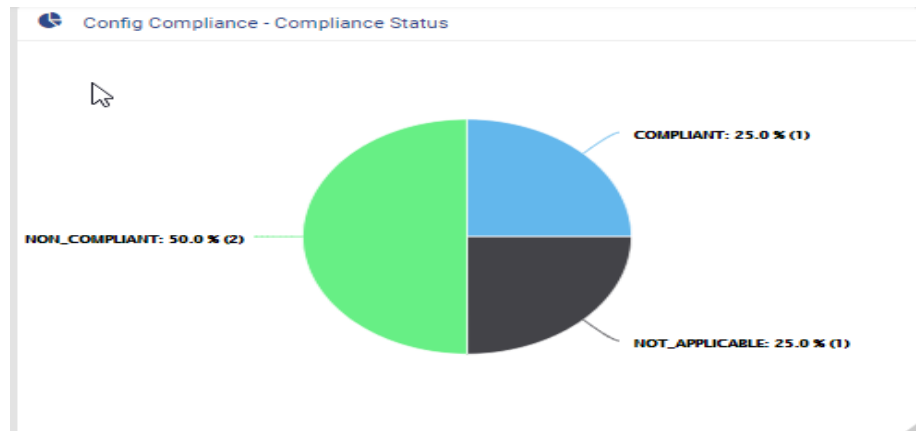
- Config compliance -devices: Representing the number of devices participated in compliance v/s all the available devices in ATOM.



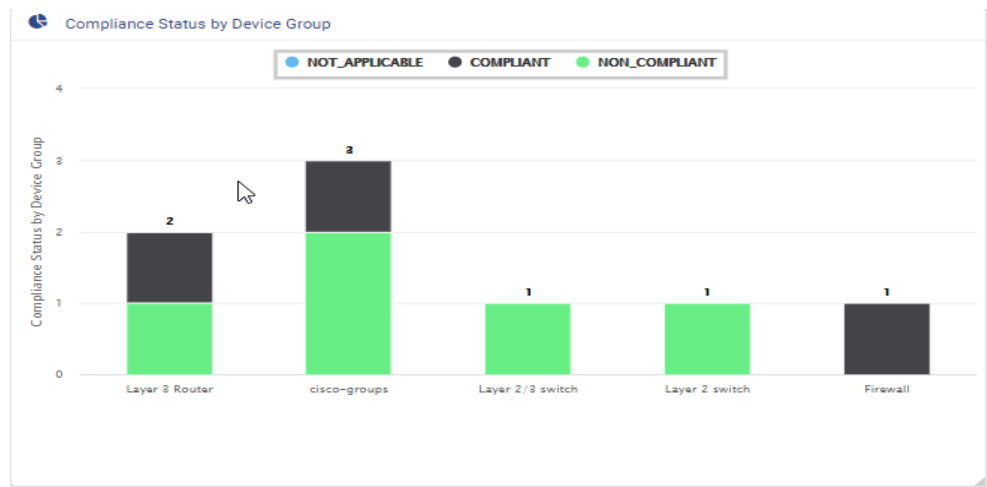
- Config compliance- execution status: A pie chart representing the percentage of execution status in terms of successful, config_empty, config_stale, errors, config_pull_failed, offline_device.



- Config compliance- compliance status: A pie chart representing the percentage of compliance in terms of Complaint, Non- compliant, Not_applicable.

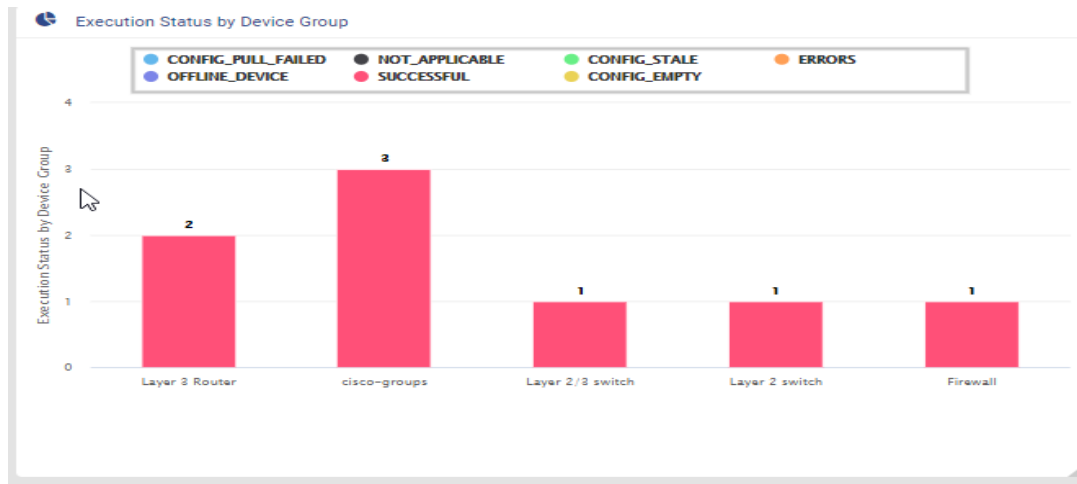


- Compliance status by group: A bar graph representing compliance status such as complaint, non-compliant, not_applicable for each and every device group on which compliance profile jobs are run.



- Execution status by device group: A bar graph representing execution status such as config pull failed, not applicable, config_stale, errors, offline device, successful, config_empty for each and every device group on which compliance profile jobs are run.

Upon clicking on any of the dashboards, the user is navigated to CRV.



Upon clicking on any of the dashboards, the user is navigated to CRV.

FAQ's

- We have a multi-vendor network, can ATOM help ensure compliance in my environment?

ATOM supports compliance management for Cisco, Juniper and Fortigate at this point in time.

- We want to standardize the network configs even before introducing automation, can ATOM help?

The standard configurations can be defined explicitly in ATOM's compliance framework. ATOM will perform compliance checks against the network and perform remediation in case of non-compliance to standardize the network configurations.

- We already have a platform which checks for compliance, but the remediation is manual, can ATOM help ?

Yes, ATOM is capable of performing remediation on non-compliant devices. The Fix-CLI or derived CLIs can be scheduled or executed immediately to fix all non-compliance issues in the network.

- Can we schedule compliance checks periodically using ATOM ?

Yes, the profiling section in ATOM's compliance framework supports scheduling of compliance checks against a device or a group of devices

- Can ATOM's compliance help in achieving regulatory compliance ?

Yes, based on the policies that regulatory authorities specify, ATOM's compliance management framework can be configured to meet these requirements.

- We have multiple checks that need to be run on the network, can ATOM help handle this scenario ?

ATOM's profiling section supports grouping of multiple policies

Appendix

- Writing Jinja template configurations based on Test Result output

Below is a sample output from the test result obtained in Launching Test Config. This helps writing jinja2 templates as required in use case requirements.

```
{
  "compliance-policies": {
    "highest-severity": "",
    "rule-violation-count": 0,
    "compliance-status": "compliant",
    "compliant-rules-output": {
      "violated-conditions": "",
      "device-compliance-condition-output": {
        "block-start-unmatched-content": "<![CDATA[]]>",
        "block-start-condition-search-output": "<![CDATA[{
          \"block_start_matched_contents\" : []
        }]]>",
        "condition-search-output": "<![CDATA[{
          \"matched_contents\" : [ {
            \"groups\" : [ {
              \"index\" : 1,
              \"grep_content\" : \"1.1.1.1\",
              \"grep_group\" : 1
            } ]
          }, {
            \"groups\" : [ {
              \"index\" : 1,
              \"grep_content\" : \"2.2.2.2\",
              \"grep_group\" : 1
            } ]
          } ]
        }]]>",
        "total-block-count": 2,
        "aggregated-condition-ouput": "<![CDATA[{
          \"condition_contents\" : [ {
```

```

        "condition_id" : null,
        "block_start_matched_content" : null,
        "block_start_unmatched_content" : null,
        "unmatched_content" : null,
        "matched_content" : null
    ]]
  ]]]>",
  "template-substituted-content": "<![CDATA[ntp server (?!10.0.0.1)(\d+.\d+.\d+.\d+)]]>",
  "block-unmatch-count": 0,
  "cli-match-output": "<![CDATA[ntp server 1.1.1.1
                        ntp server 2.2.2.2
  ]]]>",
  "condition-status": true,
  "unmatched-content": "<![CDATA[]]>",
  "id": "Remove_NTP_Extra_Config",
  "block-match-count": 2,
  "cli-unmatch-output": "<![CDATA[]]>"
},
"name": "test-condition",
"failed-conditions": ""
}
}

```

Keys	Condition value	Jinja2 template
matched_contents - when matched with regex in the condition value with the test configuration.	ntp server (?!10.0.0.1)(\d+.\d+.\d+.\d+)	{% for content in matched_contents -%} {% for group in content["groups"] -%} no ntp server {{ group["grep_content"] }} {%- endfor %} {% endfor %}
unmatched_contents - when matches with the regex and does not match with the block config in the condition value with the test configuration.	line vty (.*) session-timeout {{ session_timeout }} exec-timeout {{ exec_timeout }} 0	{% for content in unmatched_contents %} {% for group in content["groups"] %} line vty {{ group["grep_content"] }} session-timeout {{ session_timeout }} exec-timeout {{ exec_timeout }} 0

		<pre>exit {% endfor %} {% endfor %}</pre>
<p>condition_contents - condition1 captured data will be accessible to condition2 using condition_contents.</p>	<p>condition1: interface Loopback0 ip address (\d+.\d+.\d+.\d+) (\d+.\d+.\d+.\d+)</p> <p>condition2: router ospf (.*) router-id {{ condition_contents[0]["matched_content"]["matched_contents"][0]["groups"][0]["grep_content"] }}</p>	<pre>{% for content in unmatched_contents %} {% for group in content["groups"] %} router ospf {{ group["grep_content"] }} router-id {{ condition_contents[0]["matched_content"]["matched_contents"][0]["groups"][0]["grep_content"] }} exit {% endfor %} {% endfor %}</pre>

Collections in ATOM

ATOM collects network operational & performance data from multiple data sources such as SNMP, Streaming Telemetry, SNMP Traps and Syslog.

Appropriate data source and data stream can be chosen based on device capabilities, and throughput and latency requirements. Model-Driven Telemetry uses a push model and provides near real-time access to operational & performance statistics.

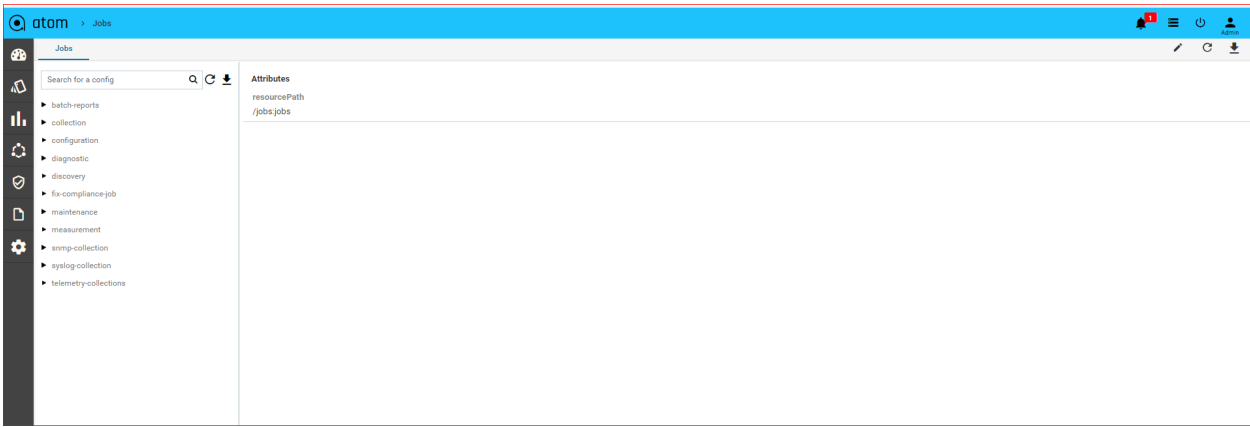
The collected Operational and Performance data can be visualized using Grafana (available as part of ATOM package) or in ATOM UI (by using various built-in reports available in Report section or under the device view). Users can build additional dashboards customized to their interests. (See ATOM Platform Guide to know more on how to create custom dashboards).

Jobs

A Job is the configurable task on a device that can be managed by ATOM. A job thus created can be a piece of work that can be created, executed and tracked in ATOM. Depending on the need, the administrator can schedule and manually run various Jobs to collect data about the device state.

Jobs are classified into the following types:

- [Collection Job](#)
- [Configuration Job](#)
- [Diagnostics Job](#)
- [Discovery Job](#)
- [Inventory Job](#)
- [Maintenance Job](#)
- [Batch-reports Job](#)
- [SNMP-collections Job](#)
- [Syslog-collection Job](#)
- [Telemetry-Collections Job](#)
- [Telemetry-measurement Job](#)



Collection Job

ATOM collects or retrieves the status of the device (OFFLINE or ONLINE) . By default, this job is scheduled to run every 6 hours. Starting with 6.0 release, you can model the collection job to collect information about the device using the SNMP OIDs. For more information, refer section, “Modelling of Collection Job” in the “*ATOM Platform Guide*.”

Configuration Job

Configuration job retrieves the running configurations from the device, or is triggered in the event of configuration discrepancies (either at the device or the service level) between the device and ATOM.

Creating a Configuration Job

1. Navigate to **Monitoring > Jobs > Configuration**
2. Select the Configuration folder > **Add Configuration**
3. In the **Create configuration** screen, enter the values in the following fields:

- **Name:** Enter the name of the Job
- **Description:** Enter an appropriate description for the Job
- **Configuration Type:** Select the type of Configuration Job from the menu:
 - **CONFIG_RETRIEVAL** - Retrieves the basic device configuration. This option is the default value.

- **DEVICE_COMPLIANCE** - The device compliance job is triggered when there is a violation of policy configured on the device or a set of devices. For information about the Device Compliance Policies, refer "Creating Compliance Templates"
- **SERVICE_COMPLIANCE** - The service compliance job is triggered when there is a discrepancy in the service configurations available on ATOM and the device.

By default, 'ServiceInventory' Job of type Service Compliance is triggered every 5 minutes.

- **Resource Type:** Select one of the following resource entities where the job should be triggered:
 - **DEVICE:** If the selected resource type is a Device, click **Add** to enter the IP address of the device for ATOM to communicate with it.
 - **DEVICE GROUP:** If the selected resource type is Device Group, click **Add** to enter the device group for which the configuration jobs should be triggered.
 - **RESOURCE POOL:** Click **Add** to select from the available resource pools where the job should be run.
- **Config Pull type:** If the selected configuration type is Config Retrieval, you can opt for one of the following methods to be used while retrieving the configurations from the device:
 - **TFTP_EXPORT** - Select this option when ATOM should retrieve configs from the TFTP server
 - **SHOW_COMMAND** - Select this option if ATOM should retrieve configs from the running configuration of the device. This will be useful when in some customer environments where the TFTP port is disabled.
- **Parse Config:** Select this option if the parsing of the configurations should be enabled on the device/devices after the successful run of the config retrieval job.

Note: Use this option if you want to override the value set in the global parameters of ATOM. By default, at the global level config parsing is enabled for all devices. However, using this option you can disable config parsing at the device level.

- **Schedule:** Select the checkbox and schedule the job to be run at intervals
- **Interval:** Enter the time period for which the job should be scheduled
- **Interval Type:** Select the units of time when the job should be scheduled (HOUR or MINUTE)

Click the Task viewer to check for the status of the executed job.

Option 1: A successful run of the Config Retrieval job, where **'TFTP Export'** enabled, fetches the following details from the device:

Config Pull 172.16.3.31
×

Task ID BudL9Oc83_Rb-klyzt2PhN4w
Time Taken 30/04/2020, 18:14:12 - 30/04/2020, 18:14:19 (6 seconds)

```

Apr 30, 2020, 6:14:12 PM   RPC Operation retrieve-configs started.
Apr 30, 2020, 6:14:12 PM   Request
{"input": {"device-id": "172.16.3.31"}}
Apr 30, 2020, 6:14:12 PM   Config Pull Job started
Apr 30, 2020, 6:14:12 PM   Config pull job started for 1 devices
Apr 30, 2020, 6:14:12 PM   Processing in Agent Name: default_agent
Apr 30, 2020, 6:14:12 PM   Retrieving config for device: 172.16.3.31, osType=IOSXE
Apr 30, 2020, 6:14:12 PM   The following command(s) to be executed on device ana-cbb-1-
gw.net.disney.com(172.16.3.31)
1. copy running-config tftp://172.16.21.19/BudL9Oc83_Rb-klyzt2PhN4w
Apr 30, 2020, 6:14:18 PM   Posted config-parse request on kafka. running-config = 34 KB
Apr 30, 2020, 6:14:18 PM   Initiated config parsing, refer config parsing task details
Apr 30, 2020, 6:14:18 PM   Done retrieving configuration for the device : 172.16.3.31
Apr 30, 2020, 6:14:18 PM   Config request saved successfully

```

Case 2: A successful run of the Config Retrieval Job, with **'show run config'**, is shown as below:

Config Pull 172.16.3.31
×

Task ID H6hAAF69gwTi6EgPM0Xvf9w
Time Taken 30/04/2020, 18:11:07 - 30/04/2020, 18:11:31 (24 seconds)

```

Apr 30, 2020, 6:11:07 PM   RPC Operation retrieve-configs started.
Apr 30, 2020, 6:11:07 PM   Request
{"input": {"device-id": "172.16.3.31"}}
Apr 30, 2020, 6:11:07 PM   Config Pull Job started
Apr 30, 2020, 6:11:07 PM   Config pull job started for 1 devices
Apr 30, 2020, 6:11:08 PM   Processing in Agent Name: default_agent
Apr 30, 2020, 6:11:08 PM   Retrieving config for device: 172.16.3.31, osType=IOSXE
Apr 30, 2020, 6:11:30 PM   Posted config-parse request on kafka. running-config = 35 KB
Apr 30, 2020, 6:11:30 PM   Initiated config parsing, refer config parsing task details
Apr 30, 2020, 6:11:30 PM   Done retrieving configuration for the device : 172.16.3.31
Apr 30, 2020, 6:11:30 PM   Config request saved successfully

```

Diagnostics Job

Diagnostics job collects various CPU, memory utilization and interface performance data that is used to provision the service. By creating a Diagnostics Job, you can run the basic device Telnet or SNMP connections to the device and also perform module- level diagnostics.

Creating a Diagnostics Job

1. Navigate to **Monitoring > Jobs > Diagnostic**

The screenshot shows the 'Create Diagnostic' form in the ATOM web interface. The form is titled 'Create Diagnostic' and has a 'Jobs' tab selected. The form includes the following fields:

- Name:** A text input field with the value 'diagnostics'.
- Resource-Type:** A dropdown menu with three options: 'DEVICE_GROUP', 'DEVICE' (selected), and 'RESOURCE_POOL'.
- Schedule:** A checkbox labeled 'Enable job scheduling'.
- Description:** A text input field with the value 'Description'.
- Interval:** A text input field with the value 'Interval'.
- Device ID:** A list of device IDs with a search bar. The selected device ID is '172.16.3.32'.

The list of device IDs shown is:

- 172.16.1.139
- 172.16.18.176
- 172.16.3.32 (selected)
- 172.16.3.33
- 172.16.3.41
- 172.16.3.42
- 172.16.3.43

2. Select **Diagnostic** and click **Actions > Add Job**

3. In the **Add Diagnostic** screen, enter values in the following fields:

- **Name:** Enter a name not exceeding 64 characters
- **Description:** Type an appropriate description for the job.
- **Resource Type:** Select one of the entities where the job should be run
 - **Device:** Enter the IP address of the device
 - **Device Group:** Select a device group from the drop-down list.
 - **Resource Pool:** Select the resource pool from the available resource pools in ATOM.
- **Schedule:** Select this option to run the job in specific intervals of time
 - **Interval:** Enter a number representing a span of time.
 - **Interval Type:** Select the units of time (minute or hour)

Click the Task Viewer pane and search for the Diagnostics job. A successful run of the Diagnostic Job displays the following information in the task details:

```

Create: diagnostic

Task ID      FU3QDugYbTQCCVNrX1wA01Vg
Time Taken   30/04/2020, 18:05:47 - 30/04/2020, 18:05:49 (1 seconds)

Apr 30, 2020, 6:05:49 PM   Reserved datanodes
Apr 30, 2020, 6:05:49 PM   Resuming commit
Apr 30, 2020, 6:05:49 PM   Triggering data model commit
Apr 30, 2020, 6:05:49 PM

Notification spec: SYSTEM, Generator type: DEFAULT, Timestamp: 2020-04-30T12:35:49.614Z
Streams: ACLN
Payload:

<notification xmlns="urn:ietf:params:netconf:capability:notification:1.0">
  <eventTime>2020-04-30T12:35:49.614Z</eventTime>
  <task-id>FU3QDugYbTQCCVNrX1wA01Vg</task-id>
  <report>
    <notification-spec>SYSTEM</notification-spec>
    <anchor-object></anchor-object>
    <source-datanode></source-datanode>
    <source-datanode-owner>system</source-datanode-owner>
    <matcher-type>DEFAULT</matcher-type>
    <generator-type>DEFAULT</generator-type>
    <timestamp>2020-04-30T12:35:49.614Z</timestamp>
  </report>
  <task-commit-notification>
    <task-id>FU3QDugYbTQCCVNrX1wA01Vg</task-id>
    <datanode-count>10</datanode-count>
    <updated-datanode-count>1</updated-datanode-count>
    <deleted-datanode-count>0</deleted-datanode-count>
  </task-commit-notification>
</notification>

Apr 30, 2020, 6:05:49 PM   Create: /jobs:jobs/diagnosticjob:diagnostic/diagnostic=Job1
+ diagnostic:
+   device-id: 172.16.1.139
+   end-time: -1
+   interval-type: MINUTE
+   job-type: DIAGNOSTIC
+   name: Job1
+   repeat-for-ever: true
+   resource-type: DEVICE
+   schedule: false
+   synchronized: true
Apr 30, 2020, 6:05:49 PM
Apr 30, 2020, 6:05:49 PM   Operation completed successfully

```

Discovery Job

Discovery job is used in discovery of the devices falling within a range of IP addresses.

The first step in provisioning a network is discovering the devices in the network. ATOM discovers the devices in the pod using either CDP or LLDP. Based on this discovery, ATOM automatically draws a network topology diagram.

If only SNMP is enabled, the topology diagram cannot be drawn as SNMP does just the sweep, which is not a methodical way of discovering device hierarchy. Therefore, it should be ensured that either CDP or LLDP is enabled on all the devices managed by ATOM.

A SEED device is the starting point from which ATOM discovers the network and its peers or neighbor devices. SEED discovery type should be selected when devices in a smaller range are required in the topology. This method of discovery is quicker, but fewer number of devices are discovered.

If the selected discovery type is SWEEP, the devices within a range of IP addresses are discovered.

Creating a Discovery Job

1. Navigate to **Monitoring > Jobs > Discovery > Add Discovery**
2. In the **Create Discovery** screen, enter values for the mandatory fields:

- **Name:** Enter an alphanumeric string to identify the created discovery job
- **Description:** Enter some text that describes the Job
- **Discovery Type:** Select one of the discovery protocols that used for discovering devices:
 - SEED - By default, the discovery type is SEED.
 - SWEEP - Change the value to SWEEP, if you want more devices to be discovered.
- **Seed Type:** Select the type of the seed protocol, either CDP or LLDP. In case the Discovery type is selected as SWEEP, enter the SWEEP IP range in the field.

NOTE: This IP range should be the same or a subset of the range of IP addresses defined in the Credential Map. IP addresses can be expressed in CIDR notation as well.

- **Hop Count:** Enter the number of hops (devices) that ATOM should discover from the seed device while using CDP.
- **Seed IP Address:** Enter the IP Address of the seed device from which the discovery of the neighbouring devices should be initiated.
- **Auto manage:** Select this option to add the discovered devices to ATOM automatically.

If this option is selected, a Managed Task is generated after the successful run of the discovery job.

- **Schedule:** Select this option if this job should be scheduled at prescribed time intervals.
 - Interval: Enter the period of time within which the job should be scheduled
 - Interval Type: Enter the units of time (HOUR or MINUTE)

A successful run of the Discovery Job with the SWEEP protocol is as shown as below:

```

Create: discovery discovery
Task ID: OCdSpK-U55QmAFmRmDPg
Time Taken: 26/05/2020, 12:00:10 - 26/05/2020, 12:00:17 (6 seconds)
May 26, 2020, 12:00:14 PM Pasted on Kafka: {"taskId": "OCdSpK-U55QmAFmRmDPg", "timestamp": 1588474614368, "commit": true, "autoRollback": true, "force": false, "skipNotify": false, "verbose": false, "skipUniqueConstraintValidation": false, "stackTrace": "", "payload": "
{
  \"operation\": \"\",
  \"systemTask\": false,
  \"taskStatusChanged\": false,
  \"actionContextPath\": \"\",
  \"errorMsg\": \"\",
  \"logUpdateReferences\": false,
  \"targetApi\": \"DEFAULT\",
  \"type\": \"CREATE\",
  \"yangState\": \"COMMIT_REQUESTED\"
}
\"input\": transaction-policy=full-fast=true/full-fast=validation-scope=COMMITTED_DATA/validation-scope=command-sequence-policy=DEPENDENCY_BASED/command-sequence-policy=do-not-send-commands-to-devices=true/do-not-send-commands-to-devices=service-discovery-enabled=false/se
May 26, 2020, 12:00:14 PM atom-core-74d9685cc-39399: Processing from kafka
May 26, 2020, 12:00:15 PM Reserved datanodes
May 26, 2020, 12:00:15 PM Recalling commit
May 26, 2020, 12:00:15 PM Triggering data model commit
May 26, 2020, 12:00:16 PM
Notification spec: SYSTEM, Generator type: DEFAULT, Timestamp: 2020-05-26T06:30:16.397Z
Streams: ACN
Payload:
{
  \"notification\": {
    \"name\": \"urn:ietf:params:netconf:capability:notification:1.0\",
    \"eventTime\": 2020-05-26T06:30:16.397Z,
    \"task\": {
      \"taskId\": \"OCdSpK-U55QmAFmRmDPg\",
      \"timestamp\": 1588474614368,
      \"commit\": true,
      \"autoRollback\": true,
      \"force\": false,
      \"skipNotify\": false,
      \"verbose\": false,
      \"skipUniqueConstraintValidation\": false,
      \"stackTrace\": \"\",
      \"payload\": \"
      {
        \"operation\": \"\",
        \"systemTask\": false,
        \"taskStatusChanged\": false,
        \"actionContextPath\": \"\",
        \"errorMsg\": \"\",
        \"logUpdateReferences\": false,
        \"targetApi\": \"DEFAULT\",
        \"type\": \"CREATE\",
        \"yangState\": \"COMMIT_REQUESTED\"
      }
      \"input\": transaction-policy=full-fast=true/full-fast=validation-scope=COMMITTED_DATA/validation-scope=command-sequence-policy=DEPENDENCY_BASED/command-sequence-policy=do-not-send-commands-to-devices=true/do-not-send-commands-to-devices=service-discovery-enabled=false/se
    }
  }
}
May 26, 2020, 12:00:16 PM Create: /jobs/jobs/discoveryJob/discovery/discovery=discovery
+ discovery:
+ auto-manage: true
+ discovery-type: SWEEP
+ end-time: -1
+ job-type: DISCOVERY
+ name: discovery
+ repeat-for-ever: true
+ schedule: false
+ sleep-ls-range: 172.16.3.40-172.16.3.45
+ synchronized: true
May 26, 2020, 12:00:16 PM Task commit db stats: Total: 42, created = 20, updated = 4, deleted = 18, natural-Id = 2, flush = 6
May 26, 2020, 12:00:17 PM Operation completed successfully
  
```

Details of the Executed Job along with the devices that are discovered by ATOM in this job are displayed as follows:

Discovery

Task ID IqGF_GrkdQSn2lUf0NMdNghg

Time Taken 30/04/2020, 18:19:38 - Unknown

```

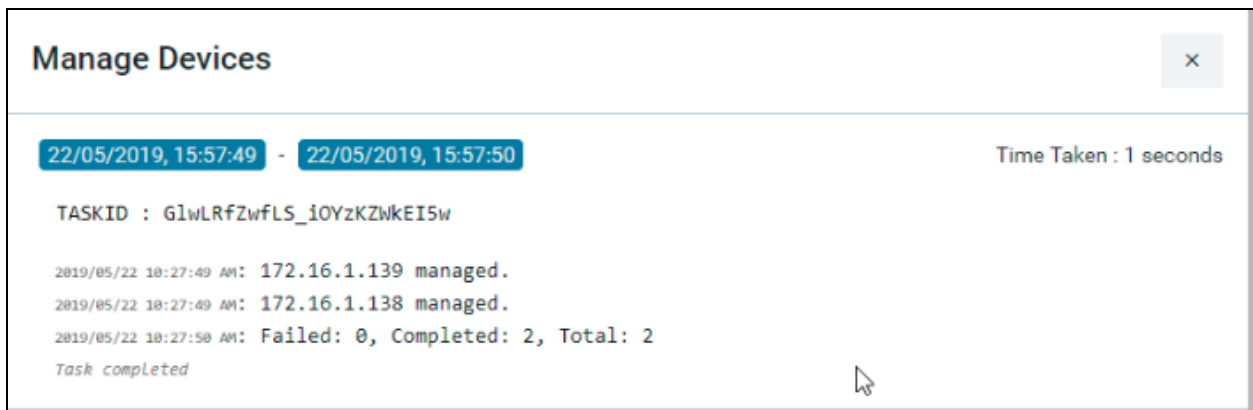
Apr 30, 2020, 6:19:38 PM RPC Operation jobs:runjob started.
Apr 30, 2020, 6:19:38 PM Request
{"input": {"job-id": "DfLGEQou5Pfi-EVtgtk6h-ow"}}
Apr 30, 2020, 6:19:38 PM Discovery Job RPC started.
Apr 30, 2020, 6:19:38 PM default_agent: Setting State to:MANAGED for Discovered Device:172.16.3.39
Apr 30, 2020, 6:19:39 PM Discovered Device 172.16.3.39 with Serial Number eb91162f3e86
Apr 30, 2020, 6:19:39 PM default_agent:: Discovery completed for: 172.16.3.39
Apr 30, 2020, 6:19:39 PM 172.16.3.39 managed.
Apr 30, 2020, 6:19:44 PM default_agent: Setting State to:MANAGED for Discovered Device:172.16.3.38
Apr 30, 2020, 6:19:44 PM Discovered Device 172.16.3.38 with Serial Number 75cc23872709
Apr 30, 2020, 6:19:44 PM default_agent:: Discovery completed for: 172.16.3.38
Apr 30, 2020, 6:19:44 PM 172.16.3.38 managed.
Apr 30, 2020, 6:19:44 PM default_agent: Setting State to:MANAGED for Discovered Device:172.16.3.34
Apr 30, 2020, 6:19:44 PM Discovered Device 172.16.3.34 with Serial Number 94WBQ1DSUUE
Apr 30, 2020, 6:19:44 PM default_agent:: Discovery completed for: 172.16.3.34
Apr 30, 2020, 6:19:44 PM 172.16.3.34 managed.
Apr 30, 2020, 6:19:44 PM default_agent: Setting State to:MANAGED for Discovered Device:172.16.3.32
Apr 30, 2020, 6:19:45 PM Discovered Device 172.16.3.32 with Serial Number 9XXKSSPQRLS
Apr 30, 2020, 6:19:45 PM default_agent:: Discovery completed for: 172.16.3.32
Apr 30, 2020, 6:19:45 PM 172.16.3.32 managed.
Apr 30, 2020, 6:19:45 PM default_agent: Setting State to:MANAGED for Discovered Device:172.16.3.49
Apr 30, 2020, 6:19:45 PM Discovered Device 172.16.3.49 with Serial Number 92GZ2OUX9BF
Apr 30, 2020, 6:19:45 PM default_agent:: Discovery completed for: 172.16.3.49
Apr 30, 2020, 6:19:45 PM 172.16.3.49 managed.
Apr 30, 2020, 6:19:45 PM default_agent: Setting State to:MANAGED for Discovered Device:172.16.3.33
Apr 30, 2020, 6:19:46 PM Discovered Device 172.16.3.33 with Serial Number 9JZEN71306E
Apr 30, 2020, 6:19:46 PM default_agent:: Discovery completed for: 172.16.3.33
Apr 30, 2020, 6:19:46 PM 172.16.3.33 managed.
Apr 30, 2020, 6:19:46 PM default_agent: Setting State to:MANAGED for Discovered Device:172.16.3.42
Apr 30, 2020, 6:19:46 PM Discovered Device 172.16.3.42 with Serial Number 9H7EKRH94UD
Apr 30, 2020, 6:19:46 PM default_agent:: Discovery completed for: 172.16.3.42
Apr 30, 2020, 6:19:46 PM 172.16.3.42 managed.
Apr 30, 2020, 6:19:46 PM default_agent: Setting State to:MANAGED for Discovered Device:172.16.3.47
Apr 30, 2020, 6:19:47 PM Discovered Device 172.16.3.47 with Serial Number 9H2XAVJ3XLO
Apr 30, 2020, 6:19:47 PM default_agent:: Discovery completed for: 172.16.3.47
Apr 30, 2020, 6:19:47 PM 172.16.3.47 managed.

```

Discovered Devices						
C						
<input type="checkbox"/>	Device	Ip Address	Serial Number	State	Credential Profile	Hops
<input type="checkbox"/>	CSR_3.41.anutacorp.com	172.16.3.41	96TJOYAIBUJ	MANAGED	cli	0
<input type="checkbox"/>	GRE-VMX-5.95	172.16.3.39	eb91162f3e86	MANAGED	cli	0
<input type="checkbox"/>	Router.anuta.com	172.16.3.48	9158G13RXQS	MANAGED	cli	0
<input type="checkbox"/>	ana-buf-2-gw.anutacorp.com	172.16.3.46	922DZREOR9U	MANAGED	cli	0
<input type="checkbox"/>	ana-cbb-1-gw.net.disney.com	172.16.3.31	9LSJZSL2T2G	MANAGED	cli	0
<input type="checkbox"/>	ana-cd-1-gw.net.disney.com	172.16.3.34	94WBQ1DSUUE	MANAGED	cli	0
<input type="checkbox"/>	ana-ivc-0-gw.net.disney.com	172.16.3.43	9VQBKYATJ6F	MANAGED	cli	0
<input type="checkbox"/>	anuta-lab02.anutacorp.com	172.16.3.32	9XXKSSPQRLS	MANAGED	cli	0
<input type="checkbox"/>	anuta-cbb-lab2	172.16.3.38	75cc23872709	MANAGED	cli	0
<input type="checkbox"/>	csr_322.net.disney.com	172.16.3.30	9E156NLIB2D	MANAGED	cli	0
<input type="checkbox"/>	enacbb	172.16.3.47	9H2XAVJ3XLO	MANAGED	cli	0
<input type="checkbox"/>	n7-cbb-0-gw.net.disney.com	172.16.3.42	9H7EKRH94UD	MANAGED	cli	0
<input type="checkbox"/>	test_enacbb.na.steelcase.net	172.16.3.36	9A3H4VF94DG	MANAGED	cli	0
<input type="checkbox"/>	wbu.anuta.com	172.16.3.49	920Z2OUX9BF	MANAGED	cli	0
<input type="checkbox"/>	wbucbb-bur-0-gw.net.disney.com	172.16.3.33	9JZEN71306E	MANAGED	cli	0
<input type="checkbox"/>	wbucbb-bur-1-gw.net.disney.com	172.16.3.44	9XJDK04726	MANAGED	cli	0

Managed Task

This task will be triggered in ATOM after the successful run of the discovery job. All the discovered devices are added to the device table maintained in the ATOM inventory are marked as “Managed” devices



Inventory Job

Inventory job is used for detecting and adding device Interfaces, interface capabilities, and interface addresses.

Extended Inventory: Retrieves the lost network connections, establishes the new network connections between the devices, retrieves the configurations from the device, By default, this job is scheduled to run every 12 hours.

Maintenance Job

You can configure the maintenance jobs to remove unwanted records of the tasks or the alarms in ATOM. The maintenance jobs can be scheduled on a one-time basis or run periodically.

Purge Older Alarm Records (You can remove unwanted, older records of the Alarms generated in ATOM.)

Creating a Purge Older Alarm Records Job

1. Navigate to **Resource Manager > Jobs**
2. In the left pane, navigate to the **MAINTENANCE** folder
3. Click the **MAINTENANCE** folder > **Actions > Add Job**
4. In the **Create MAINTENANCE** screen, enter the values for each field described below:
 - **Maintenance Job Name:** Enter a name for the maintenance job to be created.
 - **Description:** Enter a suitable description for the job
 - **Maintenance Type:** Select the type as " **PURGE_OLDER_ALARM_RECORDS**" to create a job to clean all the old Alarms from ATOM
 - **Threshold (in days):** Enter a number of days, of which the records for which history should be maintained. All the records before the prescribed days will be deleted.

- **Schedule:** In order to schedule the job to run periodically at specified intervals of time, select the Schedule option.
- **Interval:** Enter the number for the interval
- **Interval Type:** Select either Hour or Minute as units of time.

Example:

If 30, 24, and HOUR are entered as values in the fields - Threshold, Interval, and Interval Type respectively, a maintenance job is executed every 24 hours that will remove all the Alarm records older than 30 days. That is, all the records of the previous month before the 30th day will be deleted..

Purge Older Task Details Records

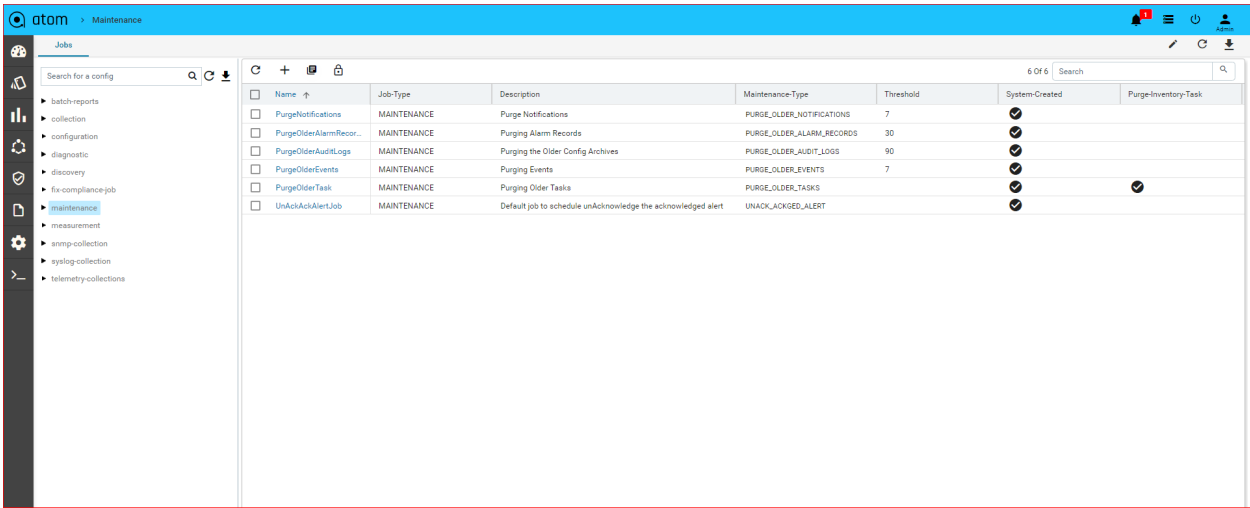
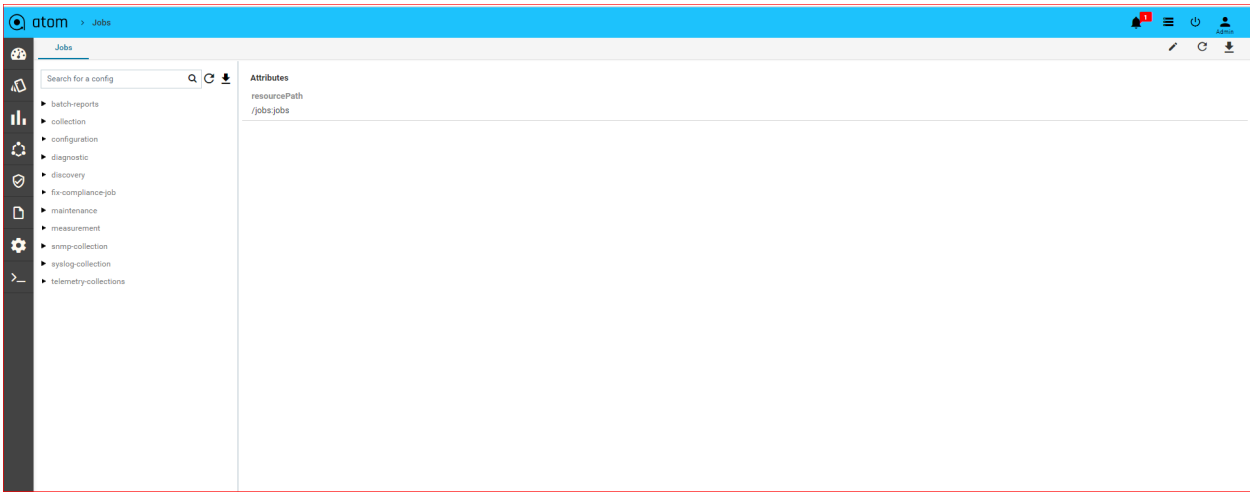
You can schedule a maintenance job that can be run to remove all the details of the tasks run before a specified period in time.

Creating a Purge Older Task Details Records

1. Navigate to **Monitoring> Jobs**
2. In the left pane, navigate to the maintenance folder
3. Click the maintenance folder > **Actions > Add Job**
4. In the **Create** maintenance screen, enter the values for each field described below:
 - **Maintenance Job Name:** Enter a name for the maintenance job to be created.
 - **Description:** Enter a suitable description for the job
 - **Maintenance Type:** Select the type as " **PURGE_OLDER_TASK_RECORDS**" to create a job to clean all the details of the tasks
 - **Threshold (in days):** Enter a number of days, of which the records for which history should be maintained. All the records before the prescribed days will be deleted.
 - **Schedule:** In order to schedule the job to run periodically at specified intervals of time, select the Schedule option.
 - **Interval:** Enter the number for the interval
 - **Interval Type:** Select either Hour or Minute as units of time.

Example

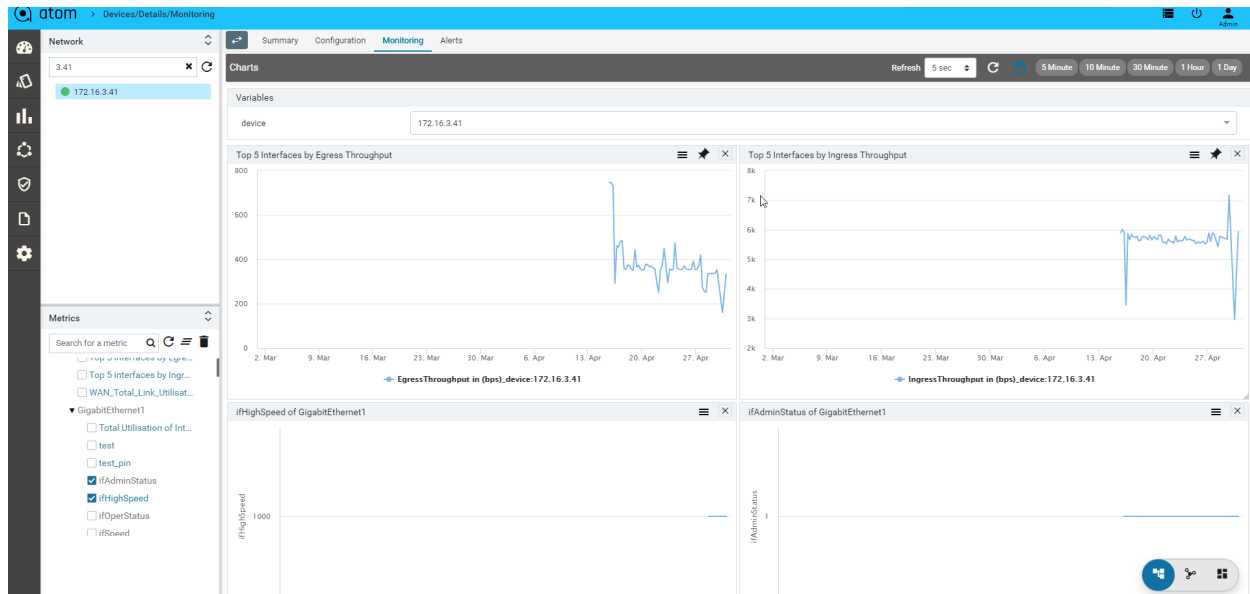
If 30, 24, and HOUR are entered as values in the fields - Threshold, Interval, and Interval Type respectively, a maintenance job is executed every 24 hours that will remove all the details of the tasks older than 30 days. That is, all those task details of the previous month before the 30th day will be deleted.



Device Inventory (SNMP)

All Device inventory collected through SNMP Collection jobs is shown in the Entities view. Following provides guidance on

1. To view Device Configuration - Navigate to **Devices** > select a device
2. Click on the “**Monitoring**” Tab
3. Collected data will be shown under MIB-name



Monitoring

[[[ATOM enables you to create Assurance profiles to facilitate 24x7 uptime of your network. Closed loop automation (CLA) framework allows you to define policies and remediation actions in violation of those policies.

ATOM collects operational & performance metrics from multiple data sources such as SNMP, SNMP traps, Syslog and Streaming Telemetry and stores them in a time-series database.

Following are the different activities on the metrics:

- Visualize Data Using Charts & Reports
- Alerts against thresholds defined on the Metrics
- Alert Dashboards - Collection of Predefined & User Defined Dashlets
- Alert Routing to Email, Slack etc.,
- Actions on Alerts
- Closed Loop Automation Actions on the Alerts

Please refer to “ATOM User Guide - Performance Management & Alerting” for further details.]]]

ATOM enables you to create Assurance profiles to facilitate 24x7 uptime of your network. Closed loop automation (CLA) framework allows you to define policies and remediation actions in violation of those policies.

ATOM collects operational & performance metrics from multiple data sources such as SNMP, SNMP traps, Syslog and Streaming Telemetry and stores them in a time-series database.

Following are the different activities on the metrics:

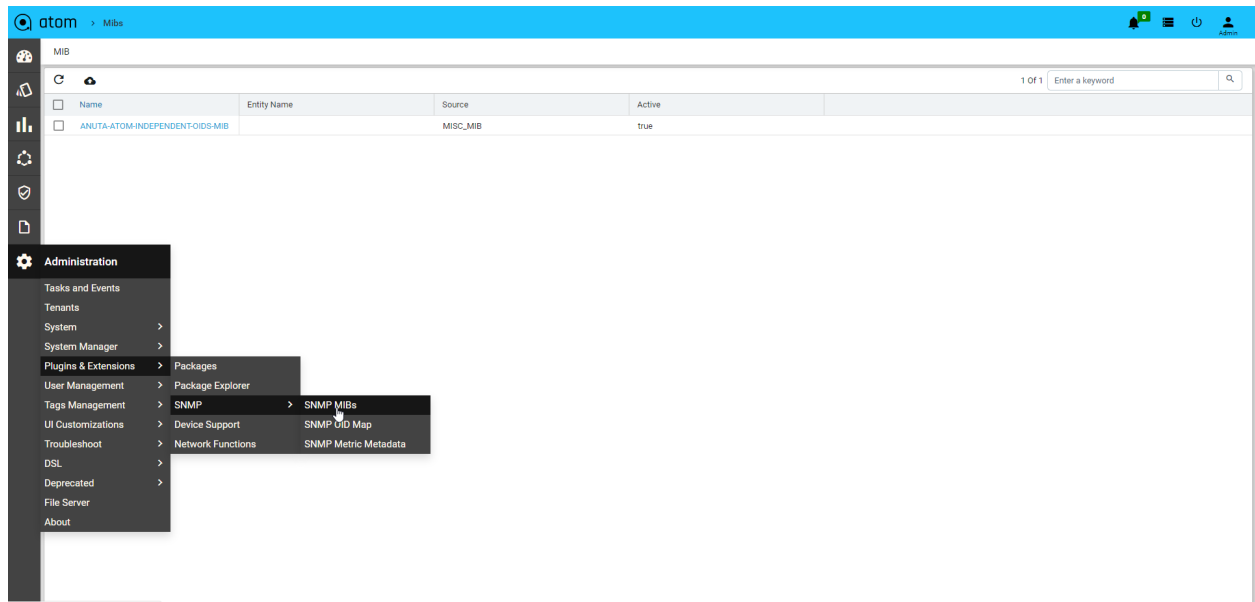
- Visualize Data Using Charts & Reports
- Alerts against thresholds defined on the Metrics
- Alert Dashboards - Collection of Predefined & User Defined Dashlets
- Alert Routing to Email, Slack etc.,
- Actions on Alerts
- Closed Loop Automation Actions on the Alerts

Import SNMP MIBs

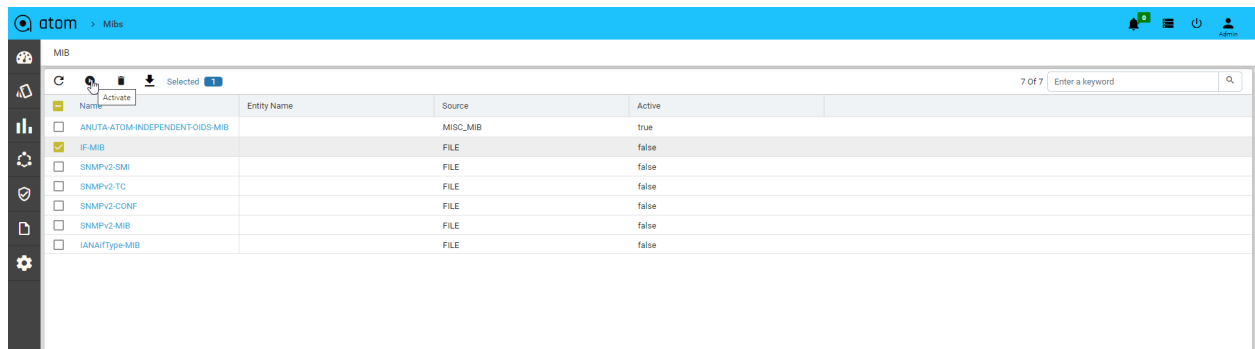
ATOM allows users to import any SNMP MIBs. To use the Object identifiers (OID's) which are part of the uploaded MIB need to be compiled by activating them.

During activation, it compiles all the dependent MIBs automatically if they are already available on the system. Otherwise, the user has to upload the dependent MIBs to compile properly.

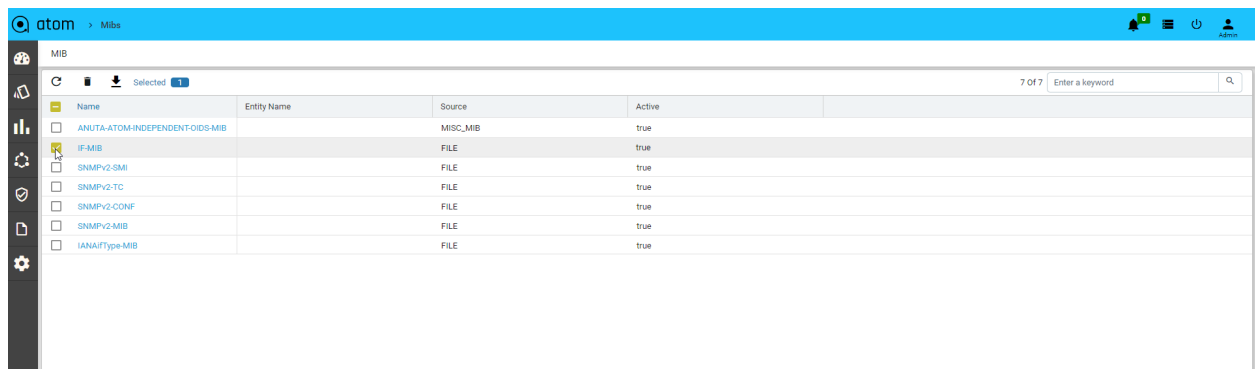
Navigate to Administration -> Plugins and Extensions -> SNMP -> SNMP MIBs and upload a mib file



Below snapshot shows IFMIB along with its dependencies after uploading. All of them are still inactive. To activate a MIB, select a MIB entry and click on the activate button.

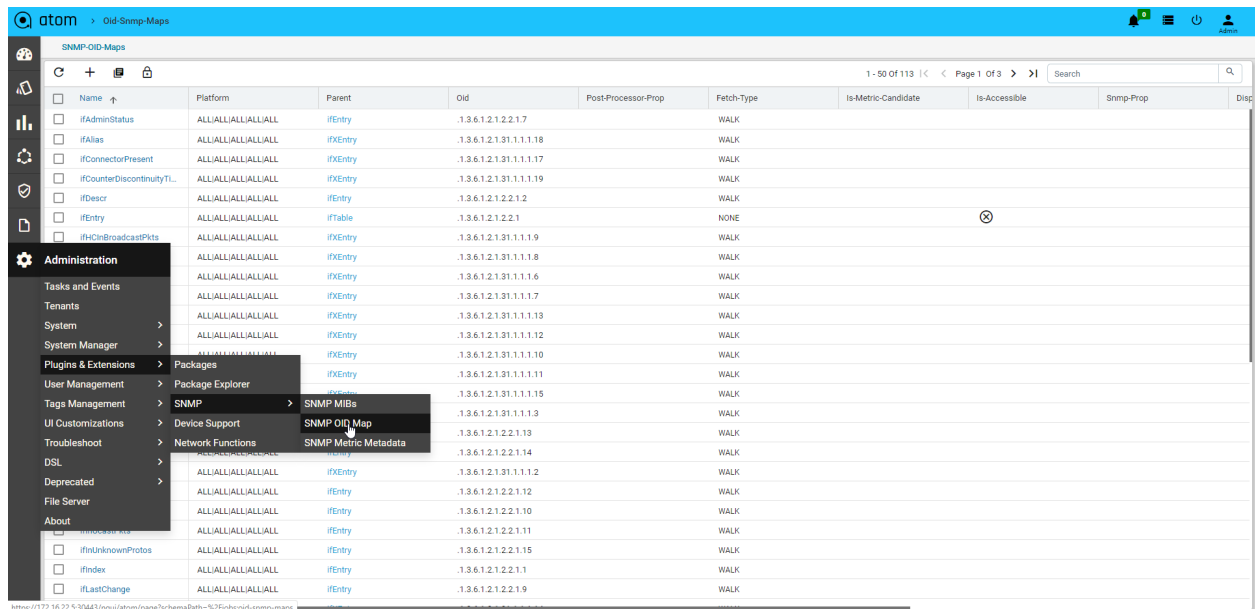


Once the MIB is activated, automatically its dependencies get activated as shown in the snapshot below.



Name	Entity Name	Source	Active
ANUTA-ATOM-INDEPENDENT-OIDS-MIB		MISC_MIB	true
IF-MIB		FILE	true
SNMPv2-SMI		FILE	true
SNMPv2-TC		FILE	true
SNMPv2-CONF		FILE	true
SNMPv2-MIB		FILE	true
IANAIfType-MIB		FILE	true

After compilation (activation), it will extract all the OIDs from each MIB and can be seen on Administration -> Plugins and Extensions -> SNMP -> SNMP OID Map.



Name	Platform	Parent	OID	Post-Processor-Prop	Fetch-Type	Is-Metric-Candidate	Is-Accessible	Snmp-Prop	Disp
ifAdminStatus	ALLIALLIALLIALL	ifEntry	.1.3.6.1.2.1.2.2.1.7		WALK				
ifAlias	ALLIALLIALLIALL	ifEntry	.1.3.6.1.2.1.31.1.1.1.18		WALK				
ifConnectorPresent	ALLIALLIALLIALL	ifEntry	.1.3.6.1.2.1.31.1.1.1.17		WALK				
ifCounterDiscontinuityTime	ALLIALLIALLIALL	ifEntry	.1.3.6.1.2.1.31.1.1.1.19		WALK				
ifDescr	ALLIALLIALLIALL	ifEntry	.1.3.6.1.2.1.2.2.1.2		WALK				
ifEntry	ALLIALLIALLIALL	ifTable	.1.3.6.1.2.1.2.2.1		NONE				
ifHCInBroadcastPkts	ALLIALLIALLIALL	ifEntry	.1.3.6.1.2.1.31.1.1.1.9		WALK				
	ALLIALLIALLIALL	ifEntry	.1.3.6.1.2.1.31.1.1.1.8		WALK				
	ALLIALLIALLIALL	ifEntry	.1.3.6.1.2.1.31.1.1.1.6		WALK				
	ALLIALLIALLIALL	ifEntry	.1.3.6.1.2.1.31.1.1.1.7		WALK				
	ALLIALLIALLIALL	ifEntry	.1.3.6.1.2.1.31.1.1.1.13		WALK				
	ALLIALLIALLIALL	ifEntry	.1.3.6.1.2.1.31.1.1.1.12		WALK				
	ALLIALLIALLIALL	ifEntry	.1.3.6.1.2.1.31.1.1.1.10		WALK				
	ALLIALLIALLIALL	ifEntry	.1.3.6.1.2.1.31.1.1.1.11		WALK				
	ALLIALLIALLIALL	ifEntry	.1.3.6.1.2.1.31.1.1.1.15		WALK				
	ALLIALLIALLIALL	ifEntry	.1.3.6.1.2.1.31.1.1.1.3		WALK				
	ALLIALLIALLIALL	ifEntry	.1.3.6.1.2.1.2.2.1.13		WALK				
	ALLIALLIALLIALL	ifEntry	.1.3.6.1.2.1.2.2.1.14		WALK				
	ALLIALLIALLIALL	ifEntry	.1.3.6.1.2.1.31.1.1.1.2		WALK				
	ALLIALLIALLIALL	ifEntry	.1.3.6.1.2.1.2.2.1.12		WALK				
	ALLIALLIALLIALL	ifEntry	.1.3.6.1.2.1.2.2.1.10		WALK				
	ALLIALLIALLIALL	ifEntry	.1.3.6.1.2.1.2.2.1.11		WALK				
	ALLIALLIALLIALL	ifEntry	.1.3.6.1.2.1.2.2.1.15		WALK				
	ALLIALLIALLIALL	ifEntry	.1.3.6.1.2.1.2.2.1.1		WALK				
	ALLIALLIALLIALL	ifEntry	.1.3.6.1.2.1.2.2.1.9		WALK				

For each OID, it will extract the following properties:

1. Fetch Type: Identifies the snmp fetch type such as GET or WALK
2. Is Accessible : Is it actual OID or the tabular header
3. Key OID's : Lists out the all key oids for each oid

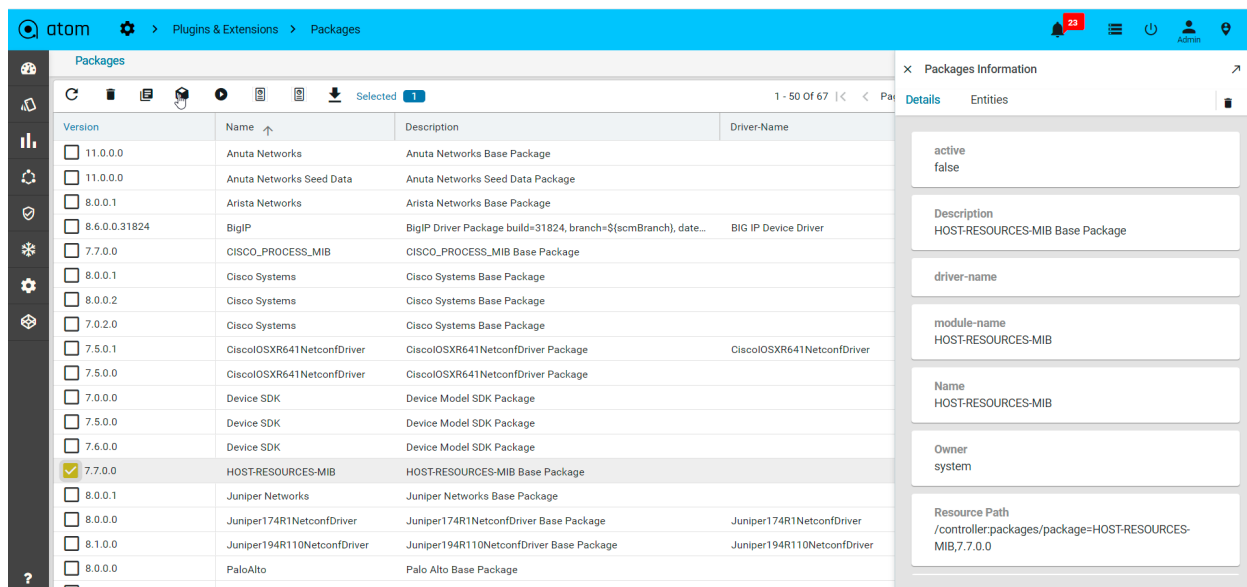
Name	Platform	Parent	Old	Post-Processor-Prop	Fetch-Type	Is-Metric-Candidate	Is-Accessible	Snmp-Prop	Dis
<input type="checkbox"/> ifLastChange	ALLIALLIALLIALL	ifEntry	.1.3.6.1.2.1.2.2.1.9		WALK				
<input type="checkbox"/> ifLinkUpDownTrapEnable	ALLIALLIALLIALL	ifXEntry	.1.3.6.1.2.1.31.1.1.1.14		WALK				
<input type="checkbox"/> ifMtu	ALLIALLIALLIALL	ifEntry	.1.3.6.1.2.1.2.2.1.4		WALK				
<input type="checkbox"/> ifName	ALLIALLIALLIALL	ifXEntry	.1.3.6.1.2.1.31.1.1.1.1		WALK				
<input type="checkbox"/> ifNumber	ALLIALLIALLIALL		.1.3.6.1.2.1.2.1.0		GET				
<input type="checkbox"/> ifOperStatus	ALLIALLIALLIALL	ifEntry	.1.3.6.1.2.1.2.2.1.8		WALK				
<input type="checkbox"/> ifOutBroadcastPkts	ALLIALLIALLIALL	ifXEntry	.1.3.6.1.2.1.31.1.1.1.5		WALK				
<input type="checkbox"/> ifOutDiscards	ALLIALLIALLIALL	ifEntry	.1.3.6.1.2.1.2.2.1.19		WALK				
<input type="checkbox"/> ifOutErrors	ALLIALLIALLIALL	ifEntry	.1.3.6.1.2.1.2.2.1.20		WALK				
<input type="checkbox"/> ifOutMulticastPkts	ALLIALLIALLIALL	ifXEntry	.1.3.6.1.2.1.31.1.1.1.4		WALK				
<input type="checkbox"/> ifOutUnicastPkts	ALLIALLIALLIALL	ifEntry	.1.3.6.1.2.1.2.2.1.18		WALK				
<input type="checkbox"/> ifOutOctets	ALLIALLIALLIALL	ifEntry	.1.3.6.1.2.1.2.2.1.16		WALK				
<input type="checkbox"/> ifOutQLen	ALLIALLIALLIALL	ifEntry	.1.3.6.1.2.1.2.2.1.21		WALK				
<input type="checkbox"/> ifOutcastPkts	ALLIALLIALLIALL	ifEntry	.1.3.6.1.2.1.2.2.1.17		WALK				
<input type="checkbox"/> ifPhysAddress	ALLIALLIALLIALL	ifEntry	.1.3.6.1.2.1.2.2.1.6		WALK				
<input type="checkbox"/> ifPromiscuousMode	ALLIALLIALLIALL	ifXEntry	.1.3.6.1.2.1.31.1.1.1.16		WALK				
<input type="checkbox"/> ifRcvAddressAddress	ALLIALLIALLIALL	ifRcvAddressEntry	.1.3.6.1.2.1.31.1.4.1.1		WALK				
<input type="checkbox"/> ifRcvAddressEntry	ALLIALLIALLIALL	ifRcvAddressTable	.1.3.6.1.2.1.31.1.4.1		NONE				
<input type="checkbox"/> ifRcvAddressStatus	ALLIALLIALLIALL	ifRcvAddressEntry	.1.3.6.1.2.1.31.1.4.1.2		WALK				
<input type="checkbox"/> ifRcvAddressTable	ALLIALLIALLIALL		.1.3.6.1.2.1.31.1.4		NONE				
<input type="checkbox"/> ifRcvAddressType	ALLIALLIALLIALL	ifRcvAddressEntry	.1.3.6.1.2.1.31.1.4.1.3		WALK				
<input type="checkbox"/> ifSpecific	ALLIALLIALLIALL	ifEntry	.1.3.6.1.2.1.2.2.1.22		WALK				
<input type="checkbox"/> ifSpeed	ALLIALLIALLIALL	ifEntry	.1.3.6.1.2.1.2.2.1.5		WALK				
<input type="checkbox"/> ifStackEntry	ALLIALLIALLIALL	ifStackTable	.1.3.6.1.2.1.31.1.2.1		NONE				
<input type="checkbox"/> ifStackHigherLayer	ALLIALLIALLIALL	ifStackEntry	.1.3.6.1.2.1.31.1.2.1.1		WALK				
<input type="checkbox"/> ifStackUnitName	ALLIALLIALLIALL		.1.3.6.1.2.1.31.1.2.1.6		GET				

By Default, ATOM uses the MIB name as the default in device monitoring trees. To provide an alias, specify the entity name.

Note:: Navigate->Administration->Plugin & Extension->Packages:: We have other option to upload the mib packages are IF-MIB, Host -resource-mib,Cisco Process-Mib, bgp4 mib are etc.

Version	Name	Description	Type	System-Created	State
<input type="checkbox"/> 11.0.0.0	Anuta Networks	Anuta Networks	DEVICE		
<input type="checkbox"/> 11.0.0.0	Anuta Networks Seed Data	Anuta Networks	DEVICE		
<input type="checkbox"/> 8.0.0.1	Arista Networks	Arista Networks	DEVICE		VALIDATED
<input type="checkbox"/> 8.6.0.0.31824	BigIP	BigIP	DEVICE		VALIDATED
<input type="checkbox"/> 7.7.0.0	CISCO_PROCESS_MIB	CISCO	DEVICE		VALIDATED
<input type="checkbox"/> 7.0.2.0	Cisco Systems	Cisco	DEVICE		VALIDATED
<input type="checkbox"/> 8.0.0.2	Cisco Systems	Cisco	DEVICE		VALIDATED
<input type="checkbox"/> 8.0.0.1	Cisco Systems	Cisco	DEVICE		VALIDATED
<input type="checkbox"/> 7.5.0.1	CiscoIOSXR641NetconfDriver	Cisco	DEVICE		VALIDATED
<input type="checkbox"/> 7.5.0.0	CiscoIOSXR641NetconfDriver	Cisco	DEVICE		VALIDATED
<input type="checkbox"/> 7.0.0.0	Device SDK	Device	DEVICE		VALIDATED
<input type="checkbox"/> 7.5.0.0	Device SDK	Device	DEVICE		VALIDATED
<input type="checkbox"/> 7.6.0.0	Device SDK	Device Model SDK Package	DEVICE		VALIDATED
<input type="checkbox"/> 8.0.0.1	Juniper Networks	Juniper Networks Base Package	DEVICE		VALIDATED
<input type="checkbox"/> 8.0.0.0	Juniper174R1NetconfDriver	Juniper174R1NetconfDriver Base Package	DEVICE		VALIDATED
<input type="checkbox"/> 8.1.0.0	Juniper194R110NetconfDriver	Juniper194R110NetconfDriver Base Package	DEVICE		VALIDATED
<input type="checkbox"/> 8.0.0.0	PaloAlto	Palo Alto Base Package	DEVICE		VALIDATED

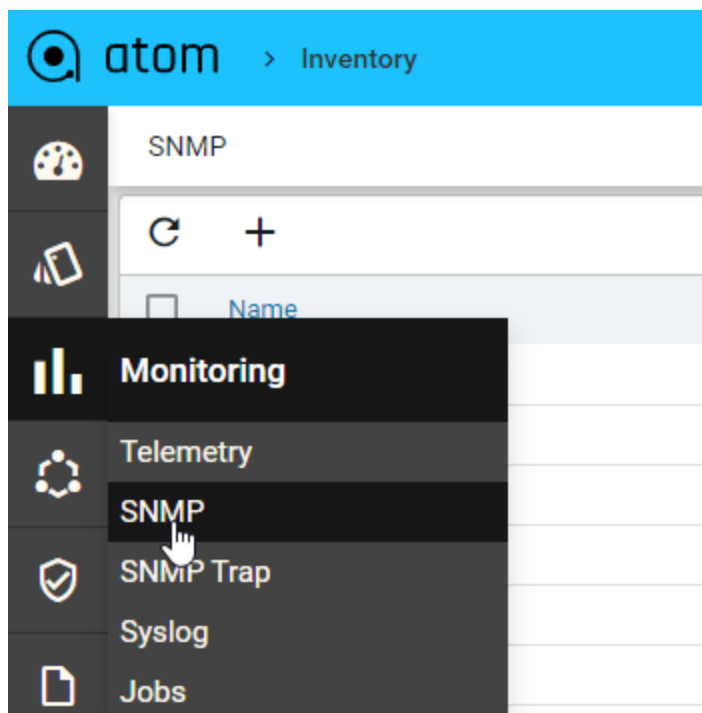
To activate a MIB, select a MIB entry and click on the activate button.

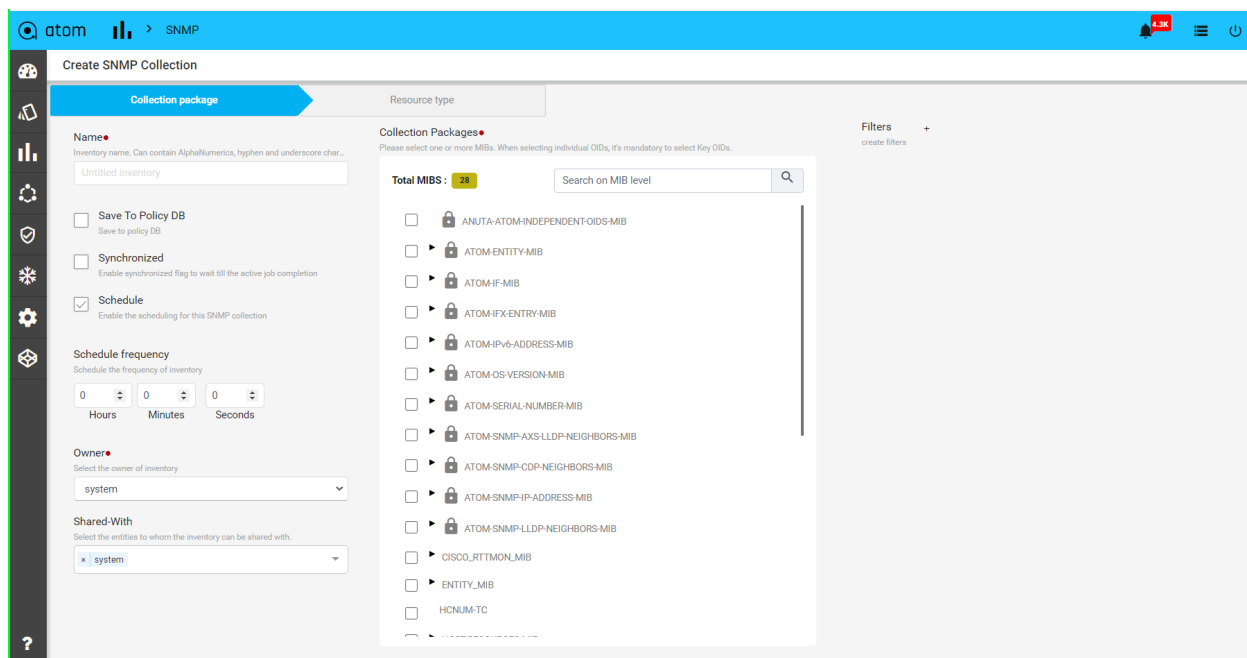


SNMP

SNMP Collections can be created for collecting Inventory information from the devices, checking the health of the devices, to measure performance of Network Objects and discovering the neighbours of the device.

1. From the left pane, navigate to **Monitoring > SNMP**
2. Click Add (+) to create a new SNMP collection profile
3. In the **New Inventory Collection** screen, enter the following details in the **Collection package** tab





- **Name:** Enter the appropriate name for the collection.
- **Synchronized:** It makes scheduled collection at regular tight intervals when it is set to true.

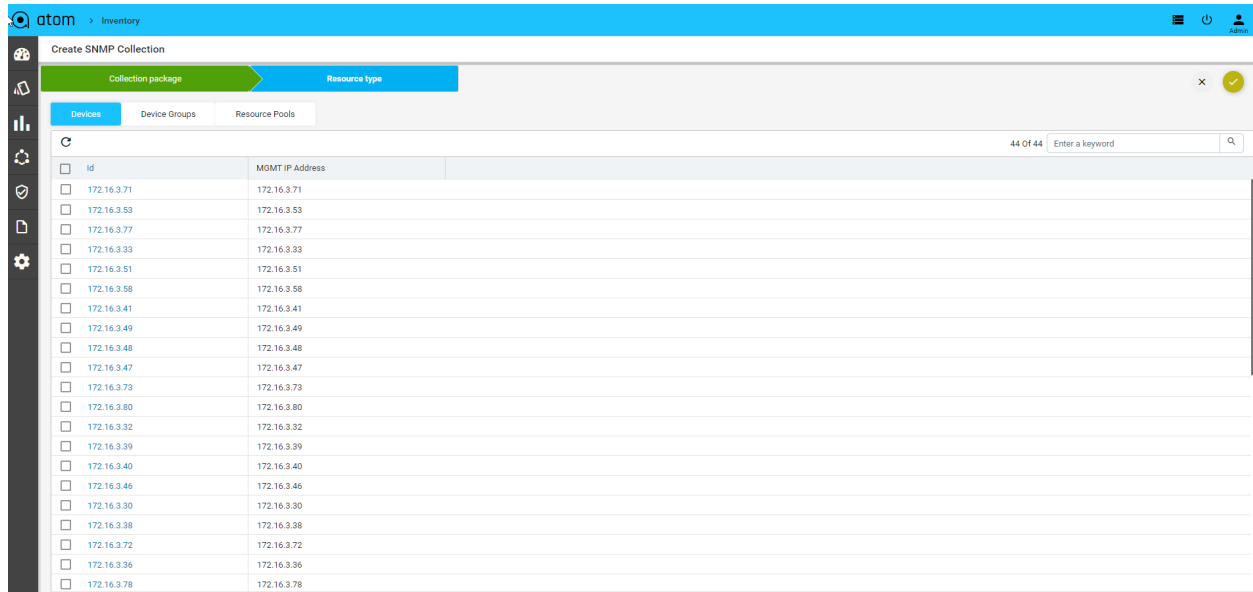
eg: if we are polling the device every 1min to collect Interface counters but the device hasn't responded for previous requests as it has too many objects to gather. So, when we set the synchronized flag to true, it waits until the previous request is completed by the device. Otherwise, it makes continuous requests at regular intervals irrespective of the time spent on the device.

- **Save to Policy DB:** By default, all the metrics data will be stored in a time series database. To store anything as inventory with overridden policy, this has to be enabled. It stores the data on the ATOM policy database and it requires the YANG mappings to OID's which can be defined as part of the device package. Refer to the ATOM SDK to generate the YANG mappings for MIB.
- **Collection Packages:** Choose one or more Object identifiers which can be spread across the MIBs. ATOM will automatically let the users know about mandatory key OIDs for selected Objects. For example, if a user wants to collect InterfaceInErrors then they must require the interface names to map the values correctly.
- **Schedule frequency:** Choose the SNMP polling frequency

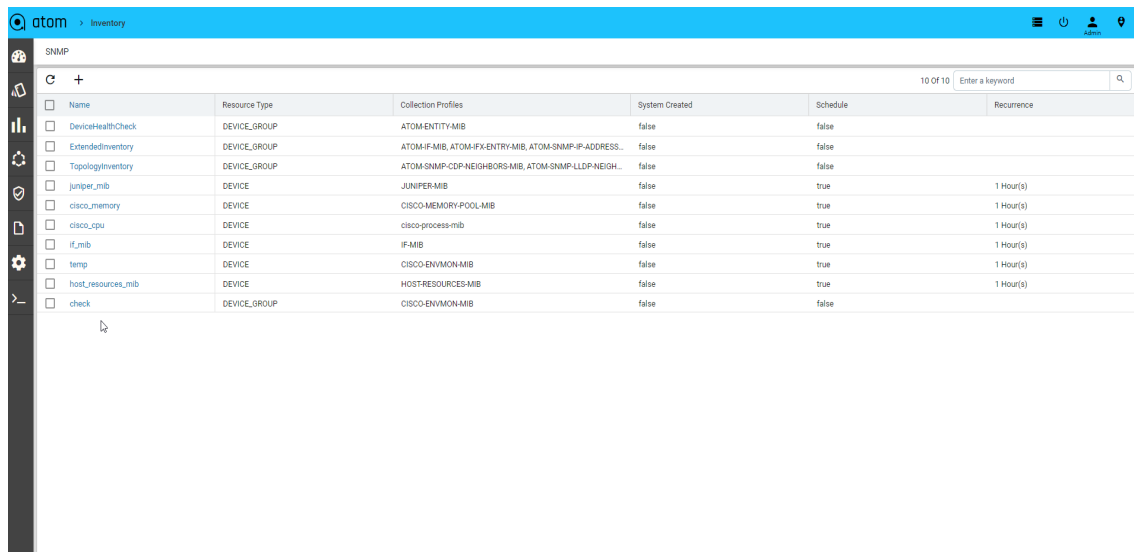
NOTE:

1. Don't select all the OID's in each MIB with less frequency, choose what is useful. Otherwise, it leads to HIGH CPU UTILIZATION on devices.

- In the Resources type tab, choose the devices, device groups or resource pools to which the profile has to be applied



- Click ✓ to save the collection profile



- Navigate to **Monitoring > SNMP** to view/modify the collection profile.

For each new collection, ATOM creates the metric schema for each MIB on every Device to showcase the metrics in logical hierarchy in device-explorer. Refer to the **Metric schema customisation**.

Telemetry

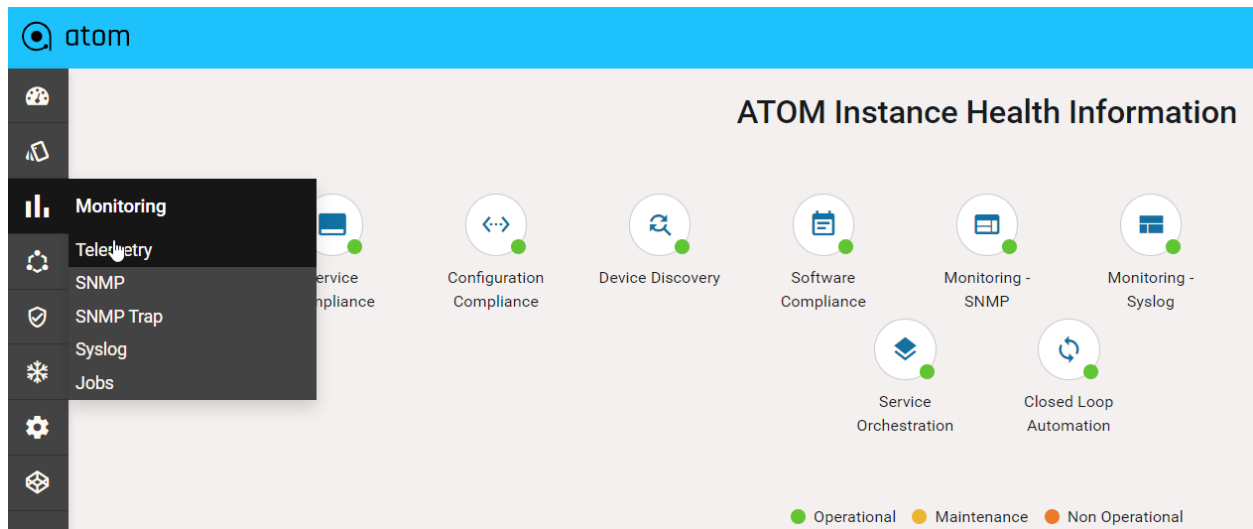
ATOM can act as a Telemetry receiver. ATOM Supports Model driven Telemetry, various Transport options, encoding and frequency of collection.

- Telemetry configuration in ATOM is optional if Telemetry is enabled manually on the devices

- ATOM can enable Telemetry subscription on the device. The section below discusses the steps to be followed.

Configure Telemetry Collection

1. Ensure that the **dry run** option is unchecked in **Administration > General Settings** tab.
2. For configuring the sensors on the device, navigate to **Monitoring > Telemetry**.



3. Click Add (+) to create a new streaming telemetry collection profile
4. In the **Create Telemetry Collection > Platform** tab, enter the following information
 - **Name:** Enter a name for the collection profile
 - **Platform:** Select the device platform that supports Telemetry
 - **Transport type:** Select the mode of transmission of telemetry data between the sender (device) and the receiver (ATOM's Agent), depending on your requirement
 - i. gRPC
 - ii. TCP
 - **Dial Mode:** Select **Dial Out** if the session establishment between the destination and the device should be initiated by the device itself.

NOTE: By default, the Dial In option is enabled.

- **Packet Encoding:** Select the encoding format for the streamed data between the sender (device) and the receiver (ATOM's Agent)
 - i. Compact gpb
 - ii. Self-describing gpb

- **Owner:** Select the owner/tenant who owns the telemetry data
- **Shared-With:** Add the tenant names to share the data or alternatively leave it as “all” to make it globally available

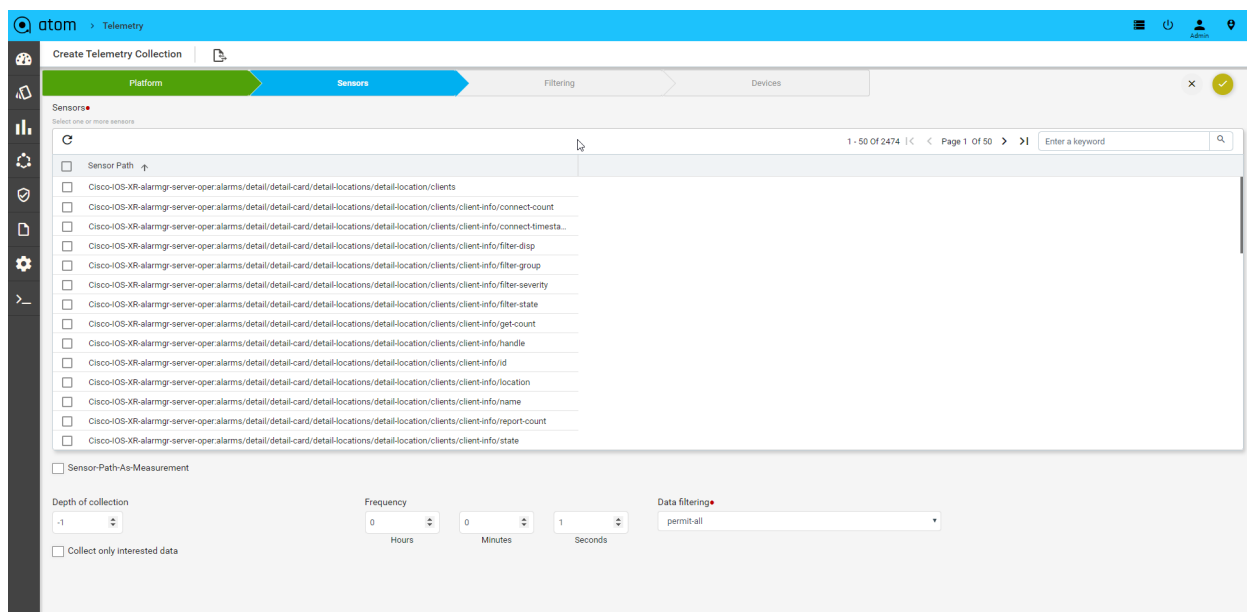
The screenshot shows the 'Create Telemetry Collection' dialog in the ATOM interface. The dialog has a blue header bar with the ATOM logo and 'Telemetry' text. Below the header, there are four tabs: 'Platform' (selected), 'Sensors', 'Filtering', and 'Devices'. The 'Platform' tab contains the following fields:

- Name:** A text input field with a red asterisk. Below it, a small text says 'Telemetry name. Can contain Alphabets, hyphen and underscore characters only. Max L...'. The input field contains 'Untitled Telemetry'.
- Platform:** A dropdown menu with a red asterisk. Below it, a small text says 'Select a platform (Ex: IOS-XR, Junos)'. The dropdown shows 'Select platform'.
- Transport Type:** A section with a red asterisk. Below it, a small text says 'Select a transport type (Ex: gRPC, TCP)'. There are two buttons: 'gRPC' (selected) and 'Tcp'.
- Dial Mode:** A section with a red asterisk. Below it, a small text says 'Check the dial mode to Dial in or Dial Out'. There are two buttons: 'Dial in' (selected) and 'Dial Out'.
- Packet-Encoding:** A section with a red asterisk. Below it, a small text says 'Select a packet encoder'. There are two buttons: 'Compact-Opb' and 'Self Describing Opb' (selected).
- Owner:** A dropdown menu with a red asterisk. Below it, a small text says 'Select the owner of telemetry'. The dropdown shows 'system'.
- Shared-With:** A dropdown menu with a red asterisk. Below it, a small text says 'Select the entities to whom the telemetry can be shared with'. The dropdown shows 'all'.

5. In the **Create Telemetry Collection > Sensors** tab, enter the following information

- **Sensors:** Select the sensor paths for the device platform selected

NOTE: All the Xpaths of the entities described in the YANG model that the receiver (ATOM) has subscribed to in the sender (device) are displayed here. The sensors are the abstract paths defined in YANG mapped to device model paths published by the vendor. To view all the available device model paths or sensors published by the vendor and imported into ATOM, navigate to the **Administration > Plugins & Extensions > Device Support > Edit > Telemetry seed data**



- Depth of collection:** In the selected sensor paths, enter the level in the container hierarchy at which the data should be streamed from the device.
 For example, a depth of collection of 0 returns only the leaf entities in the specified container and a depth of 1 returns the data from depth 0 and depth 1 levels the data from the container specified in the sensor path.

Example:

Sensor path: Cisco-IOS-XR-pfi-im-cmd-oper:interfaces/interface-xr/interface

Depth of collection = -1

Result:

Cisco-IOS-XR-pfi-im-cmd-oper:interfaces/interface-xr/interface/arp-information/arp-is-learning-disabled (boolean)

Cisco-IOS-XR-pfi-im-cmd-oper:interfaces/interface-xr/interface/arp-information/arp-timeout (float)

Cisco-IOS-XR-pfi-im-cmd-oper:interfaces/interface-xr/interface/arp-information/arp-type-name (string)

Depth of collection = 0

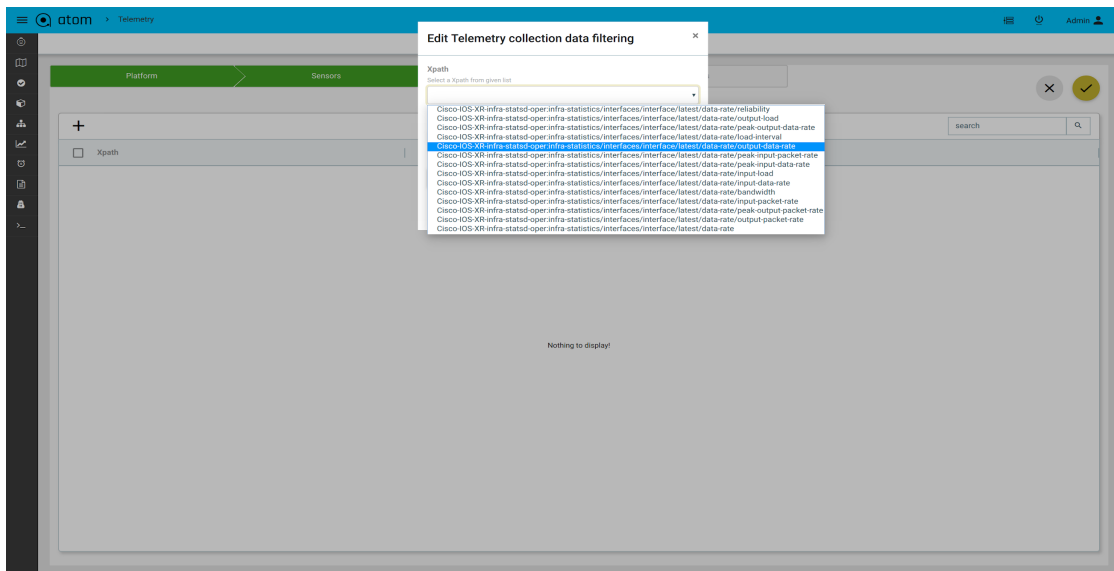
Result:

Cisco-IOS-XR-pfi-im-cmd-oper:interfaces/interface-xr/interface/bandwidth (float)

Cisco-IOS-XR-pfi-im-cmd-oper:interfaces/interface-xr/interface/crc-length (float)

Cisco-IOS-XR-pfi-im-cmd-oper:interfaces/interface-xr/interface/description (string)

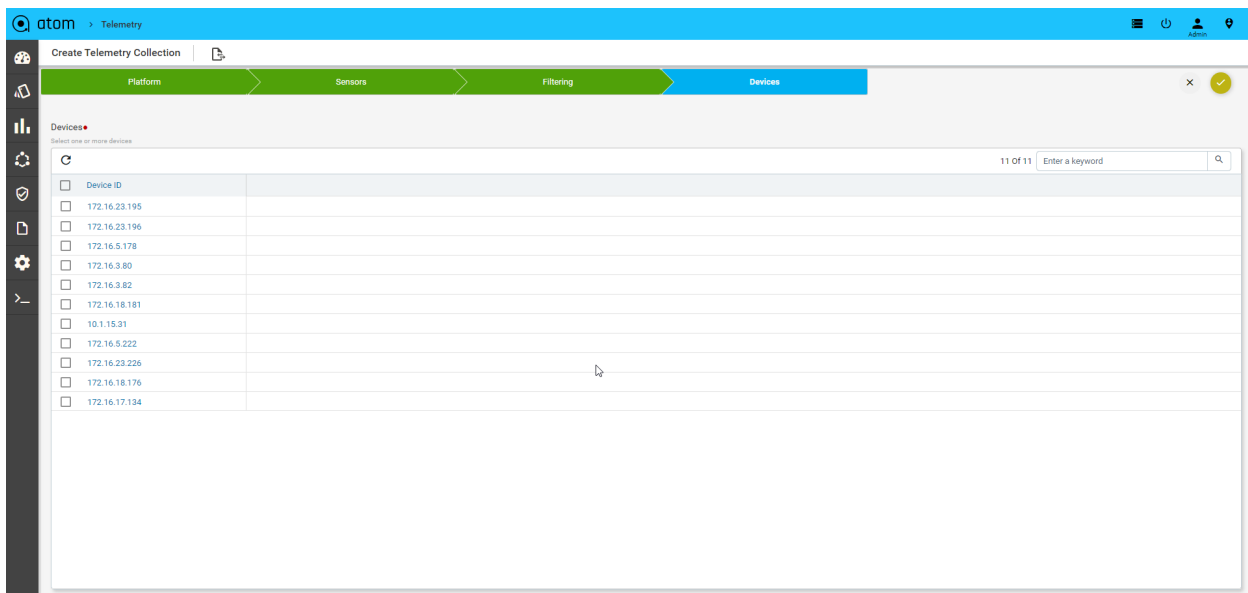
- **Collect only interested data:** Check this box to collect the data selectively only for the required sensors.
 - **Data filtering:** In the global filtering policy, permit or deny the collected data from the device to the Time-series database.
 - **Frequency:** Enter the rate at which sensor data needs to be collected from the device to be saved into a Time-series database.
6. In the **Create Telemetry Collection > Filtering** tab, enter the following information
- Click Add (+) to create Telemetry collection data filtering
 - In the **Xpath** field, choose the sensor path to be filtered from the drop-down list



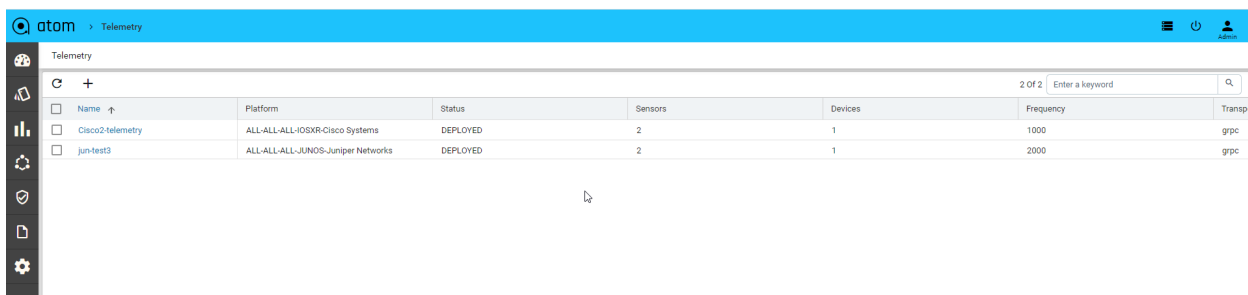
- In the **Subtree** field, check or uncheck the value of subtree
- In the **Filter** type, select Permit or Deny

NOTE: You can exercise the filtering options(either permit or deny) at the granular level. The filtering option set here overrides the global filtering policy described in step 5

7. In the **Create Telemetry Collection > Devices** tab, select the Device (multiple devices can be selected), where the subscription services are running and the data should be pushed to the device.



8. Click ✓ to save the streaming telemetry collection profile.
9. After successful creation of collection, provision collection and deploy if it is dial in mode. If it is dial out mode then provision the collection only.
10. After successful deployment of the collection make sure that the status of the collection will be shown as DEPLOYED in the UI.



11. Navigate to **Monitoring > Telemetry** to view/modify the collection profile

SNMP Traps

Navigate->Administration->Plugins & Extension->Packages::

Upload the mib dependencies package & load it.

Navigate->Monitoring->SNMP::

Add snmp collection with if-mib & csr devices and schedule it every two minutes & ran job it

Check snmp collection data is shown in the monitoring tab.

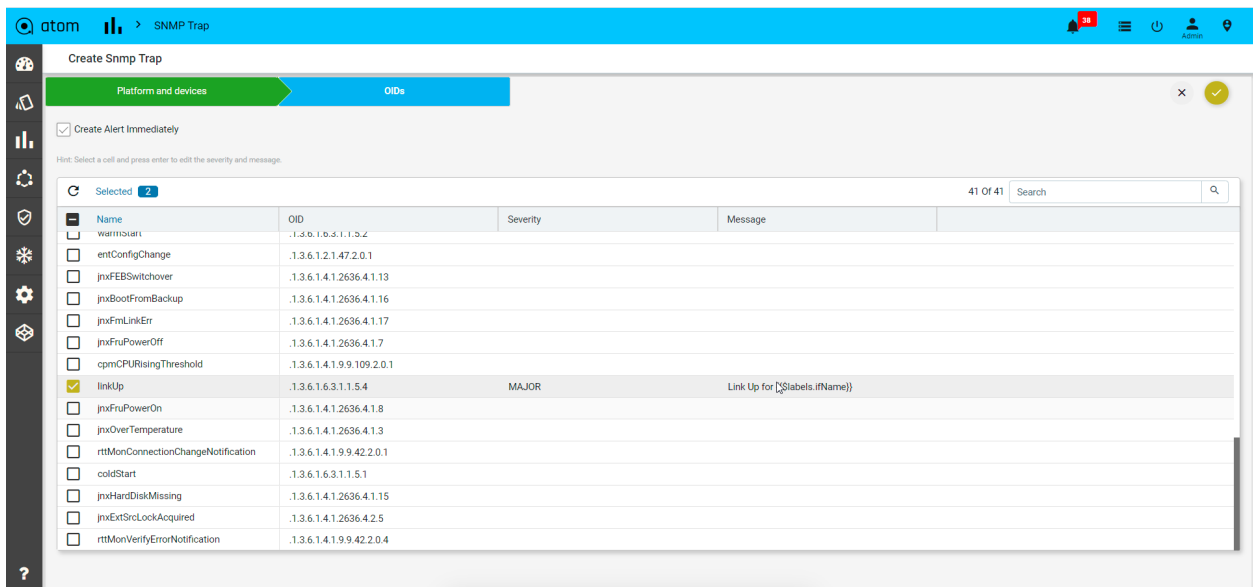
SNMP traps generated by the device can be collected and visualized in ATOM

1. Navigate to **Monitoring > SNMP trap** on the main menu
2. Click Add (+) to create a new SNMP trap collection profile

3. In the **Platform and devices** tab, enter the following information

- **Name:** Enter a name for the SNMP trap profile
- **Platform:** Select the device platform from the drop-down list
- **Devices:** Select one or more devices from which SNMP traps need to be collected

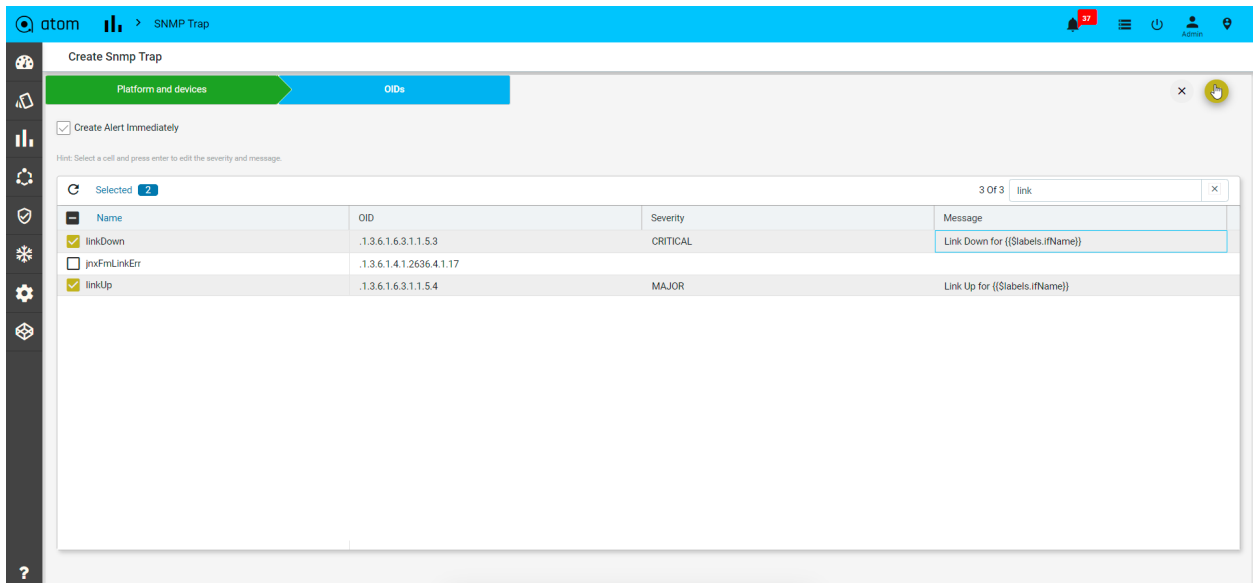
Click on create alert immediately & enter link is up and down in mib oids

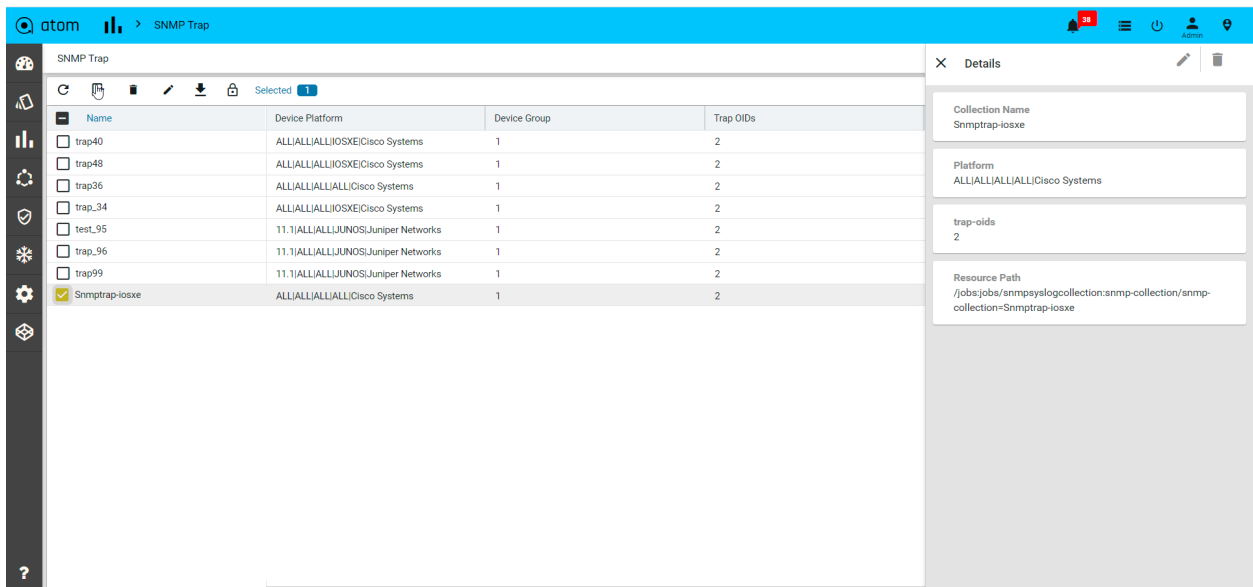


4. In the OIDs tab, enter the following information

- In the list provided, select the OIDs to be enabled for SNMP trap collection to enter link is up & link is down as above screen shot.
- **Owner:** Choose the owner/tenant who owns the SNMP trap data
- **Shared-With:** Add the tenant names to share the data or alternatively leave it as "all" to make it globally available

5. Click ✓ to save the collection profile





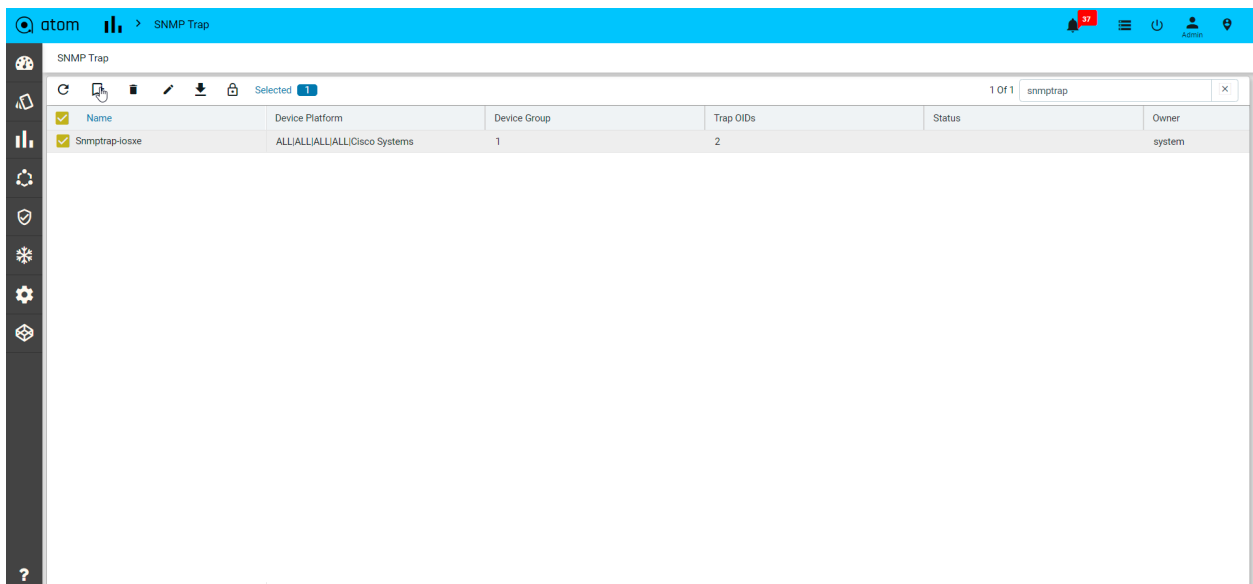
The screenshot shows the ATOM web interface for the 'SNMP Trap' configuration page. The main table lists several traps, with 'Snmpttrap-iosxe' selected. The details panel on the right shows the configuration for the selected trap.

Name	Device Platform	Device Group	Trap OIDs
<input type="checkbox"/> trap40	ALL ALL ALL IOSXE Cisco Systems	1	2
<input type="checkbox"/> trap48	ALL ALL ALL IOSXE Cisco Systems	1	2
<input type="checkbox"/> trap36	ALL ALL ALL ALL Cisco Systems	1	2
<input type="checkbox"/> trap_34	ALL ALL ALL IOSXE Cisco Systems	1	2
<input type="checkbox"/> test_95	11.1 ALL ALL JUNOS Juniper Networks	1	2
<input type="checkbox"/> trap_96	11.1 ALL ALL JUNOS Juniper Networks	1	2
<input type="checkbox"/> trap99	11.1 ALL ALL JUNOS Juniper Networks	1	2
<input checked="" type="checkbox"/> Smmpttrap-iosxe	ALL ALL ALL ALL Cisco Systems	1	2

Details panel for 'Snmpttrap-iosxe':

- Collection Name: Smmpttrap-iosxe
- Platform: ALL|ALL|ALL|ALL|Cisco Systems
- trap-oids: 2
- Resource Path: /jobs/jobs/snmptsyslogcollection:snmp-collection/snmpt-collection=Snmpttrap-iosxe

6. Navigate to **Monitoring > SNMP Trap** to view/modify the collection profile
7. Click the **Subscribe** button to provision the SNMP traps configured on the devices



The screenshot shows the ATOM web interface for the 'SNMP Trap' configuration page after the subscription process. The table now includes a 'Status' column, and the 'Snmpttrap-iosxe' trap is shown with a status of 'system'.

Name	Device Platform	Device Group	Trap OIDs	Status	Owner
<input checked="" type="checkbox"/> Smmpttrap-iosxe	ALL ALL ALL ALL Cisco Systems	1	2	system	system

8. The subscription of snmp trap can be seen as shown as below commands

The screenshot shows the ATOM Alerts interface. A modal window titled "Enable-Trap for device: 172.16.3.42" is open, displaying task details. The task ID is EV-ankXL0PQzuK1HcHttPfv, and the parent ID is P9bZDLxV59Rkg/bEa1IH3QZw. The user name is admin. The time taken is 23/06/2021, 13:01:30 - 23/06/2021, 13:03:31 (120 seconds). The operation is SnmpTrapDest, and the device name/IP is n7-cbb-0-gw.anutanetworks.com / 172.16.3.42. The status is PROVISIONED. The commands are snmp-server host 172.16.21.194 traps version 2c public udp-port 162 and snmp-server enable traps snmp. A "Download as Config" button is at the bottom right of the modal.

9. Check login device to shutdown the interface & check the alerts can be generated for snmp trap

The screenshot shows the ATOM Alerts interface with a table of alerts. The table has columns: Alert Name, Device/NodeName, Device Name, Device FQDN, Resource, Severity, Ack, Status, and Message. There are three alerts listed:

Alert Name	Device/NodeName	Device Name	Device FQDN	Resource	Severity	Ack	Status	Message
linkDown	172.16.3.42	n7-cbb-0-gw.anutanetworks...		{collection=TRAP_42, device=172.16.3.42, ifD...	Critical	⊗	Open	LINK IS DOWN
linkDown	172.16.3.42	n7-cbb-0-gw.anutanetworks...		{collection=TRAP_42, device=172.16.3.42, ifD...	Critical	⊗	Open	LINK IS DOWN
linkUp	172.16.3.42	n7-cbb-0-gw.anutanetworks...		{collection=TRAP_42, device=172.16.3.42, ifD...	Minor	⊗	Open	LINK IS UP

Syslogs

Syslogs generated by the device can be collected and visualized in ATOM

1. Navigate to Monitoring> Syslog on the main menu
2. Click Add (+) to create a new Syslog collection profile
3. In the **Platform and devices** tab, enter the following information
 - **Name:** Enter a name for the syslog collection profile

- **Platform:** From the drop-down list, select the device platform from which Syslog messages need to be collected
- **Devices:** select one or more devices from which Syslog need to be collected

Create Syslog

Platform and devices | Event specs

Name
Syslog name. Can contain AlphaNumeric, hyphen and underscore characters only. Max length is 64.
syslog

Platform
Select a platform (Ex: IOS-XR, Junos).
ALLIANCE/All Cisco Systems

Devices
Select one or more devices.
21 of 21 | Search

Id	Owner	Shared With	Status
<input type="checkbox"/> 172.16.1.139	system	system	
<input type="checkbox"/> 172.16.17.133	system	system	
<input type="checkbox"/> 172.16.18.176	system	system	
<input type="checkbox"/> 172.16.3.30	system	system	
<input type="checkbox"/> 172.16.3.33	system	system	
<input type="checkbox"/> 172.16.3.34	system	system	
<input type="checkbox"/> 172.16.3.36	system	system	
<input type="checkbox"/> 172.16.3.40	system	system	
<input checked="" type="checkbox"/> 172.16.3.42	system	system	
<input type="checkbox"/> 172.16.3.43	system	system	
<input type="checkbox"/> 172.16.3.44	system	system	
<input type="checkbox"/> 172.16.3.45	system	system	
<input type="checkbox"/> 172.16.3.46	system	system	
<input type="checkbox"/> 172.16.3.49	system	system	
<input type="checkbox"/> 172.16.3.72	system	system	

Owner
Select the owner of syslog syslog.
system

Shared-With
Select the entities to whom the syslog syslog can be shared with.
system

4. In the **Event specs** tab, enter the following information

- In the list provided, select the Event specs to be enabled for Syslog collection
- **Owner:** Choose the owner/tenant who owns the Syslog data
- **Shared-With:** Add the tenant names to share the data or alternatively leave it as “all” to make it globally available

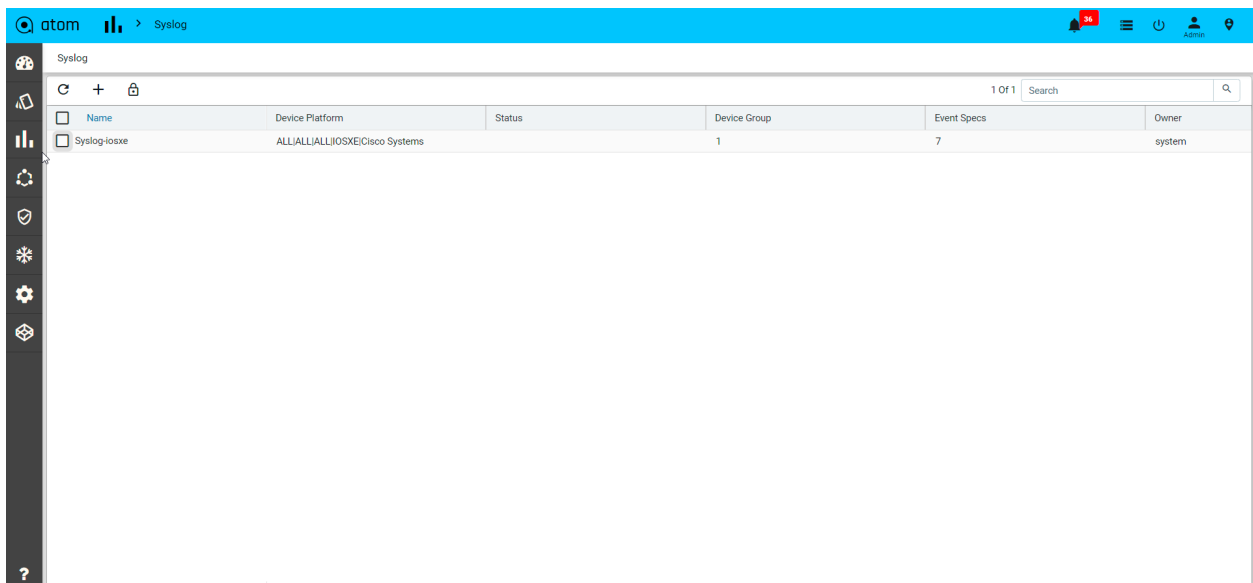
Edit Syslog

Platform and devices | **Event specs**

Selected 6 | Search

Name
<input checked="" type="checkbox"/> %SYS-S-RELOAD
<input type="checkbox"/> %PARSER-S-CFGLOG_LOGGEDCMD
<input checked="" type="checkbox"/> %SYS-S-RESTART
<input checked="" type="checkbox"/> %LINK-S-CHANGED
<input checked="" type="checkbox"/> %LINEPROTO-S-UPDOWN
<input checked="" type="checkbox"/> %LINK-3-UPDOWN
<input checked="" type="checkbox"/> %SYS-S-CONFIG

5. Click ✓ to save the collection profile

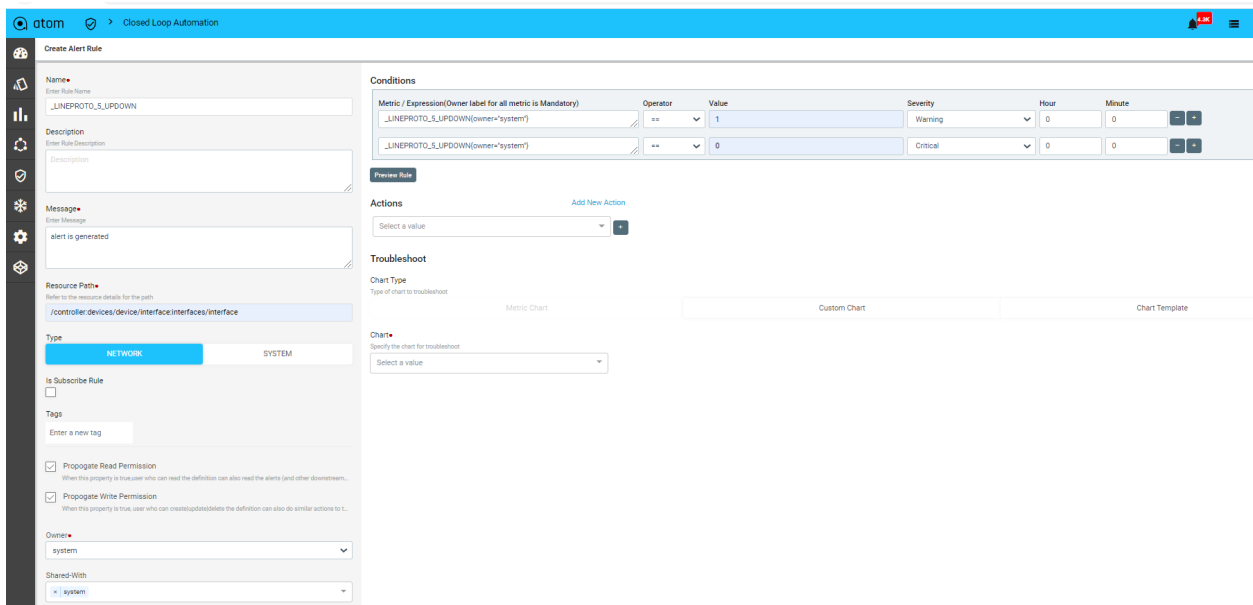


6. Navigate to **Monitoring > Syslog** to view/modify the collection profile

7. Click the **Subscribe** button to provision the Syslog collection on the devices

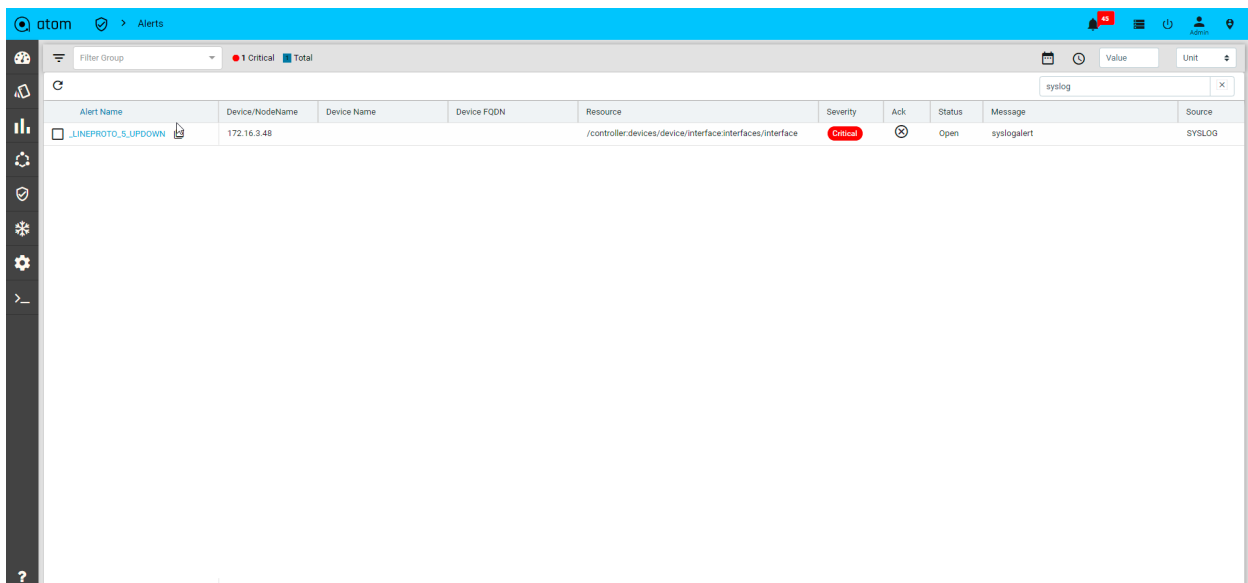
Navigate->Assurance->Closed loop automation:

Create alert definition for syslog is link is up and down & activa it.



8. Check login device to make the interface is up/down

9. Syslog alerts can be generated after the interface is up/down.



The screenshot shows the ATOM Alerts interface. At the top, there's a header bar with the ATOM logo and a navigation menu. Below the header, there's a filter bar with a dropdown for 'Filter Group' and a status indicator showing '1 Critical' and 'Total'. The main area contains a table of alerts. The table has columns for Alert Name, Device/NodeName, Device Name, Device FQDN, Resource, Severity, Ack, Status, Message, and Source. A single alert is visible with the name '_LINEPROTO_S_UPDOWN', device '172.16.3.48', and severity 'Critical'. The message is '/controller/devices/device/interface/interfaces/interface' and the source is 'SYSLOG'.

Alert Name	Device/NodeName	Device Name	Device FQDN	Resource	Severity	Ack	Status	Message	Source
<input type="checkbox"/> _LINEPROTO_S_UPDOWN	172.16.3.48			/controller/devices/device/interface/interfaces/interface	Critical	<input checked="" type="checkbox"/>	Open	syslogalert	SYSLOG

Charts & Templates

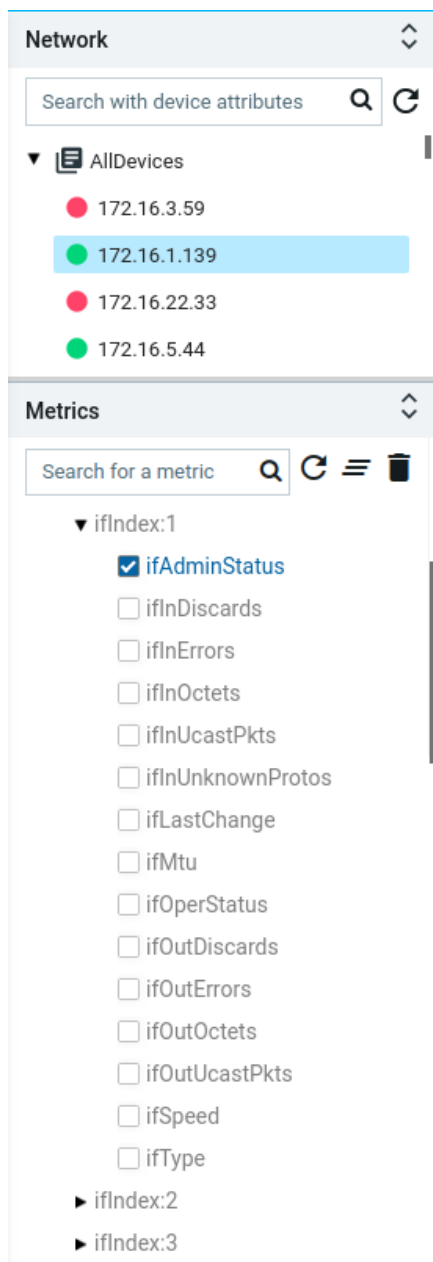
ATOM provides various customisations to visualise the collected metrics efficiently. Users can customise the mib names, metric names, logical grouping with human understandable keys, unified view of metric in single chart, context selection, multiple views through templates and also via dashboards.

Key Customization for each metric :

To change the keys in the device monitoring tree. By default key-oids will be shown as parents in a logical hierarchy. Most of the MIB's have Index numbers as the keys, so we can't keep track of them while looking at each metric. It is always required to keep well known objects to map the rest of the metrics.

eg: Interface Counters collected through the IFMIB will result below the tree by default. Here, it will have ifindex as the key by default which sits on top in the hierarchy.

Below is the snapshot without any customisations which has default key-oids at higher level in the tree. Here, Ifindex is the default key for IFMIB entries.



To change the keys to represent this data more human understandable, navigate to the *Administration -> Plugins & Extensions -> SNMP -> SNMP OID Map*

Here, choose each metric and add/modify the metric-tag-oids to reflect them in the tree.

We can add multiple metric tag OID's, after each entry press ENTER to add another. To choose which property has to be primary for tree representation among multiple metric-tag-oids, we have to change the field settings. To change this, select the oid entry as shown in snapshots and select the entities -> fields on the right hand side.

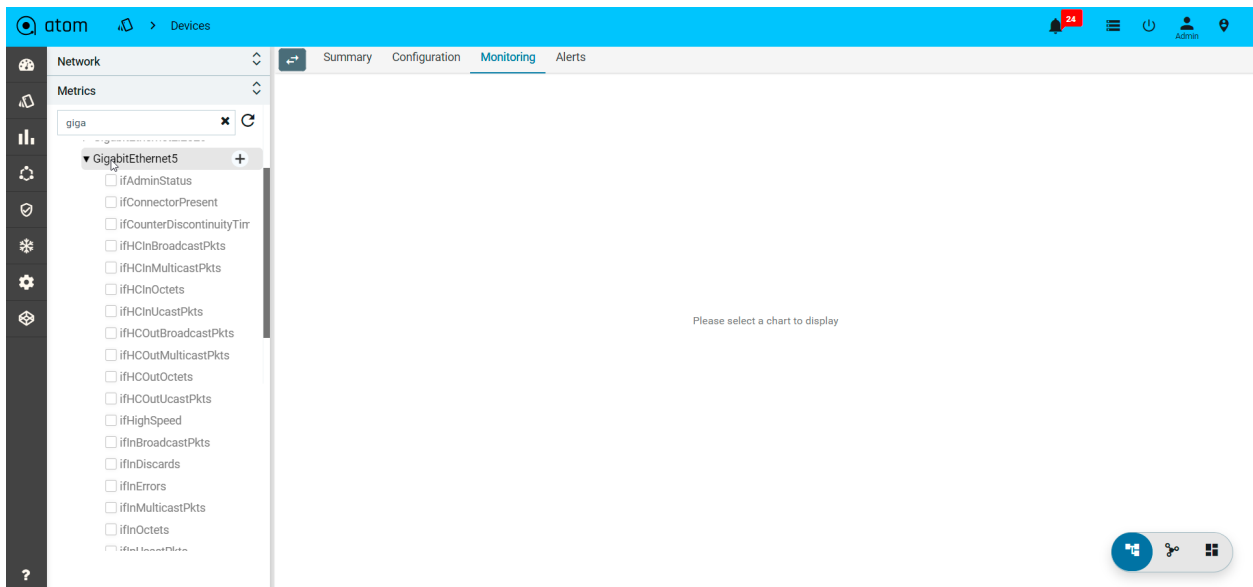
Name	Platform	Parent	Old	Post-Processor-Prop	Fetch-Type	Is-Metric-Candidate	Is-Metric-Tag
ifInErrors	ALLIALLIALLIALL	ifEntry	.1.3.6.1.2.1.2.2.1.14		WALK		

Here, add the fields by providing the oid name and select the label-field checkbox to true to make this as primary.

If we add multiple fields, only one of the fields will become the primary.

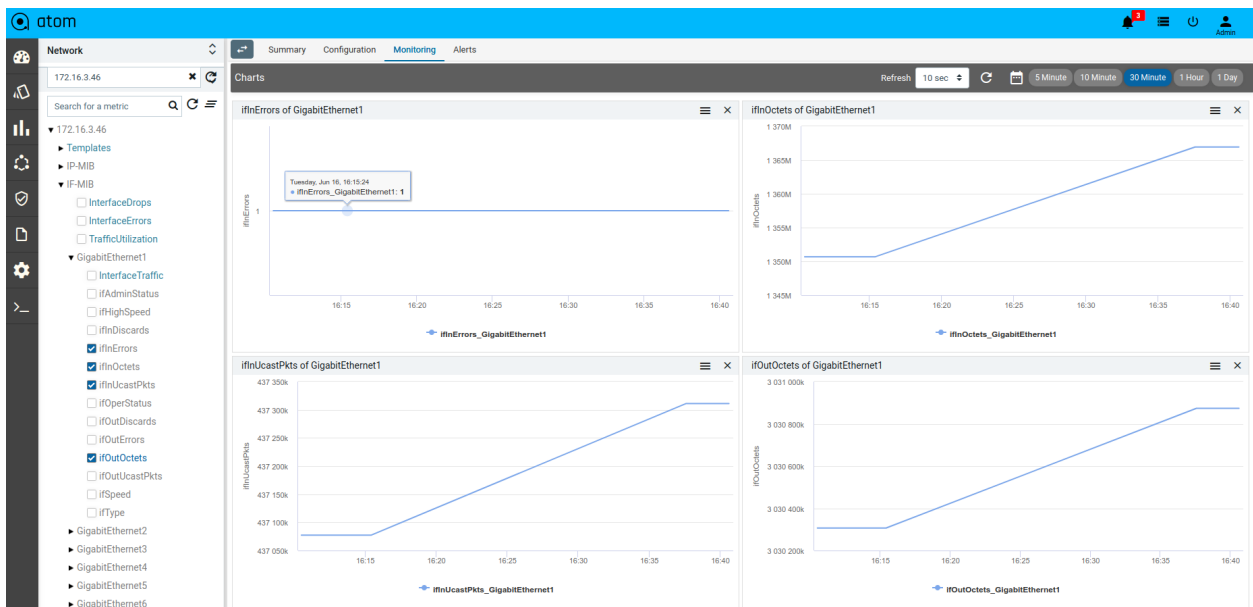
Name	Namespace	Label-Field	Type	Owner	Shared With
ifAlias	device	<input type="checkbox"/>		system	all
ifDescr	device	<input checked="" type="checkbox"/>		system	all

All the remaining fields will be shown on mouse over of the key on the device monitoring tree.



Metric Visualization on Device Explorer:

Navigate to Resource Manager -> Devices -> select the device -> Monitoring tab on the right hand side.



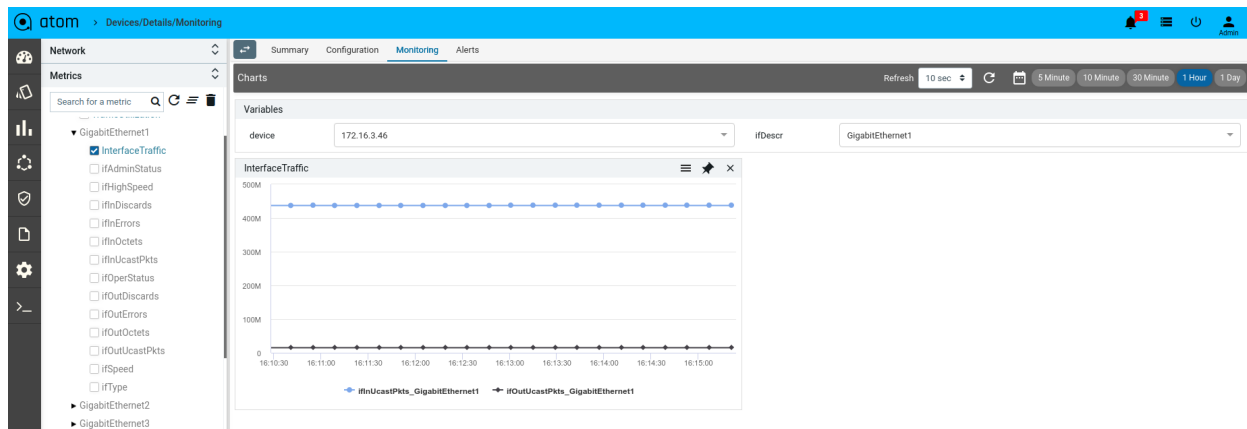
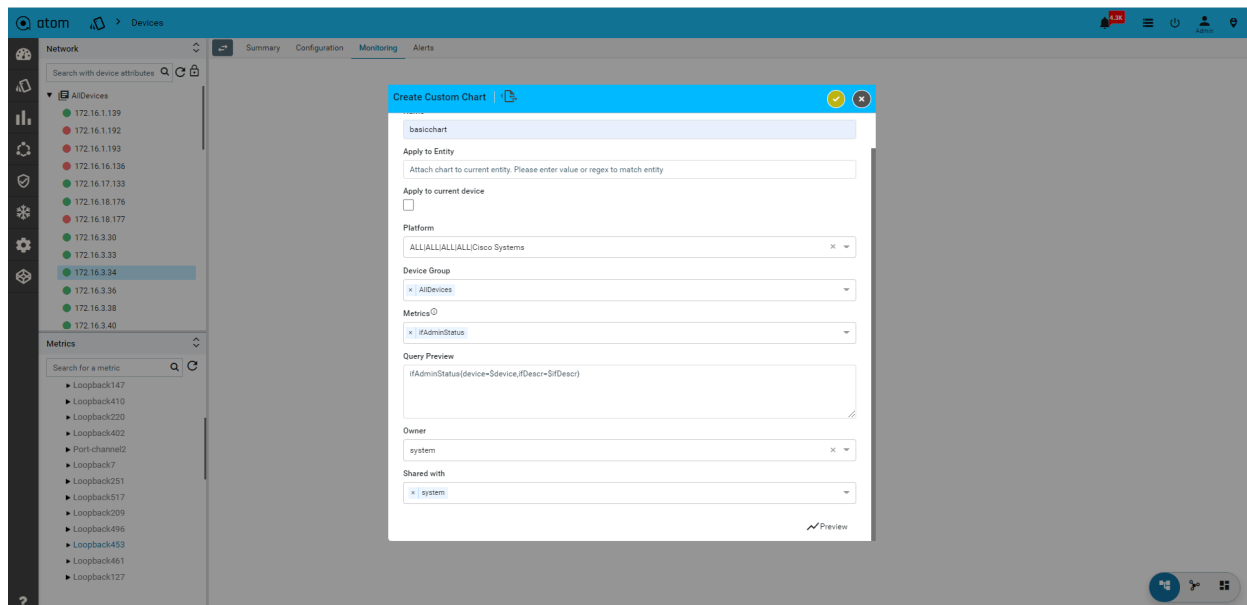
Refer [Chart Actions](#) section for various interactions.

Chart Creation

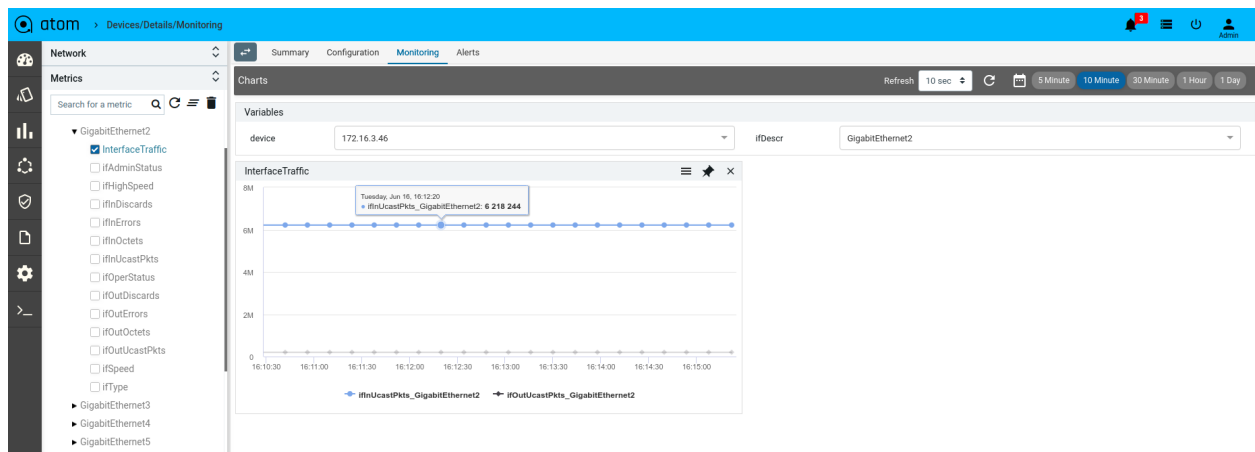
Create Simple Custom Chart:

Simple custom chart is to show multiple metrics in a single chart without further filtering. It will create a corresponding PromQL and retrieve the data.

eg: create a chart to show interface traffic which includes interface in & out packet counters, use preview option to see the data before saving.



Once the chart is created at any node level and it will be applicable to all other siblings automatically. In the above snapshot, we have created Interface Traffic chart at Gi1 Interface and it will be shown under all other interfaces automatically with the right context selected.



Create an Advanced Custom Chart

Advanced custom charts can have any valid promQL expression with/without variables. To use any variables, we need to define them beforehand.

Create the Variables for Charts

Variables are useful to move across instances or context easily with the dropdown selection. Variables can be used across the charts or templates.

To check the predefined variables or to create new global variable navigate to the *Administration -> UI Customizations -> Chart Variables*

Below are system defined variables.

Metric-Charts:variables							
<input type="checkbox"/> Name ↑	Description	Default-Value	System-Defined	Type	Metric	Filter	Tsdb-Query
<input type="checkbox"/> device			✓		device	\$device	
<input type="checkbox"/> ifDescr			✓		ifDescr	{'device':"\$device"}	

Provide below fields to create a variable.

1. **Name:** Provide the name of variable
2. **Description:** Specify the description for the variable
3. **Default-value:** provide the default value for the variable. It is used when the given expression or query doesn't result in any value.
4. **Variable Type:** Variable value can be derived using following ways:
 - a. **data-type :** It can be selected when the variable is known
 - b. **query:** it is used to fetch the variable value by executing the query on a metric instance database.
 - c. **top-n:** it is to get the current top n result and use those labels to plot a graph through selections.
5. **Metric:** to specify the key/column which is of interest to store in a variable.
6. **Filter:** It is equivalent to the where condition while fetching the metric value
7. **Tsdb-query:** To execute the top N query on tsdb and use the top n keys in the graph to render for any metric.

eg: To get the Top N Interface utilisation, we need to get the interface names which are utilizing high bandwidth in specified time periods.

Variable creation example:

The screenshot shows the 'Create Variable' form in the ATOM application. The breadcrumb navigation at the top indicates the path: 'UI Customizations > Chart-Variable'. The form is titled 'Create Variable' and includes a sidebar with various icons. The form fields are as follows:

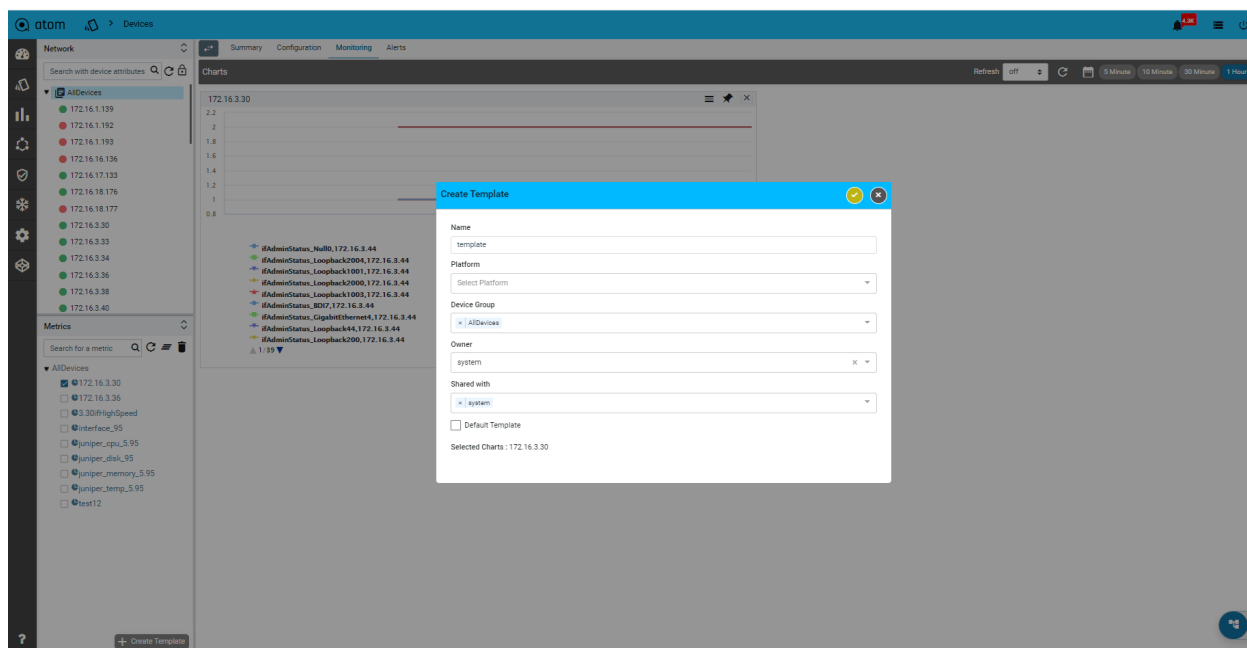
- Name** (mandatory):
- Description**:
- Variable-Type**:
- Metric**:
supports multiple metrics (ifDescr,ifIndex) selection from filter result where first need t...
- Filter**:
used to filter the results based the device and metric value conditions. Ex: {device:"10.1...
- Default-Value**:
- Owner** (mandatory):
- Shared-With**:

Create Chart Templates:

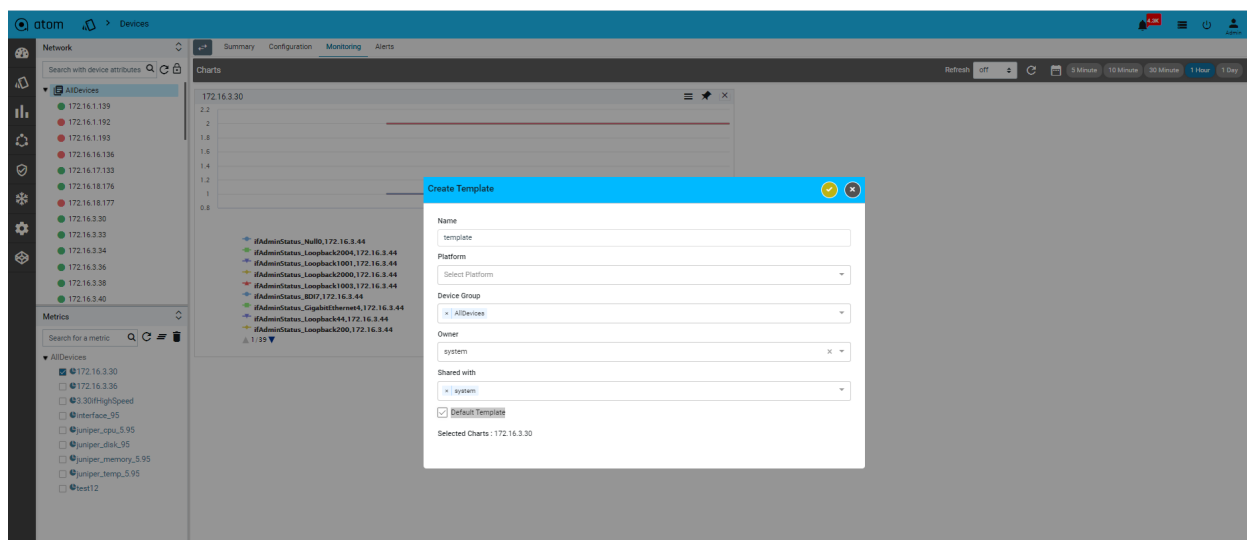
Chart templates are used to combine one or more charts with/without variables and save it as a view. Templates can be restricted to specific platforms or Groups. Any one of the templates can be set as default, to render automatically when we move to a particular logical level. Applies-to function determine where the template has to show in the device monitoring tree.

Templates can be created from the Admin screen or from device explorer view.

To create a template from device explorer, select one more chart and click on create template bottom left side. User has to provide the name for each template and optionally we can restrict the scope of template to platform or device group.



Select any template as default template which will be shown at each device or group level without any further selection.



To create template from Admin Screen, Navigate to the Administration -> UI Customizations -> Chart Templates

atom > UI Customizations > Chart Templates

Create Template

Entities

Template - Interface_overview

Name *

Description

Applies To

Device Platform

Device Group

Device Group	Description	Owner
<input type="checkbox"/> Host	Host Devices Group	system
<input type="checkbox"/> Layer 2 switch	Layer 2 switch Devices Group	system
<input type="checkbox"/> Layer 2/3 switch	Layer 2/3 switch Devices Group	system
<input type="checkbox"/> Layer 3 Router	Layer 3 Router Devices Group	system
<input type="checkbox"/> Loadbalancer	Loadbalancer Devices Group	system
<input type="checkbox"/> OfficeDevices	Office Devices Group	system
<input type="checkbox"/> VPNs	VPN Devices Group	system
<input type="checkbox"/> con	con	system
<input type="checkbox"/> groupclass	groupclass	system
<input type="checkbox"/> gpp-11	gpp-11	system
<input type="checkbox"/> gpp-jumper	gpp-jumper	system

Default Template ☐

Node-Level ☐

Owner

Shared With

atom > > Chart-Templates

Create Template

Entities

Template - Interface_overview

Chart Spec

Interface_overview

Description

Description

Platform

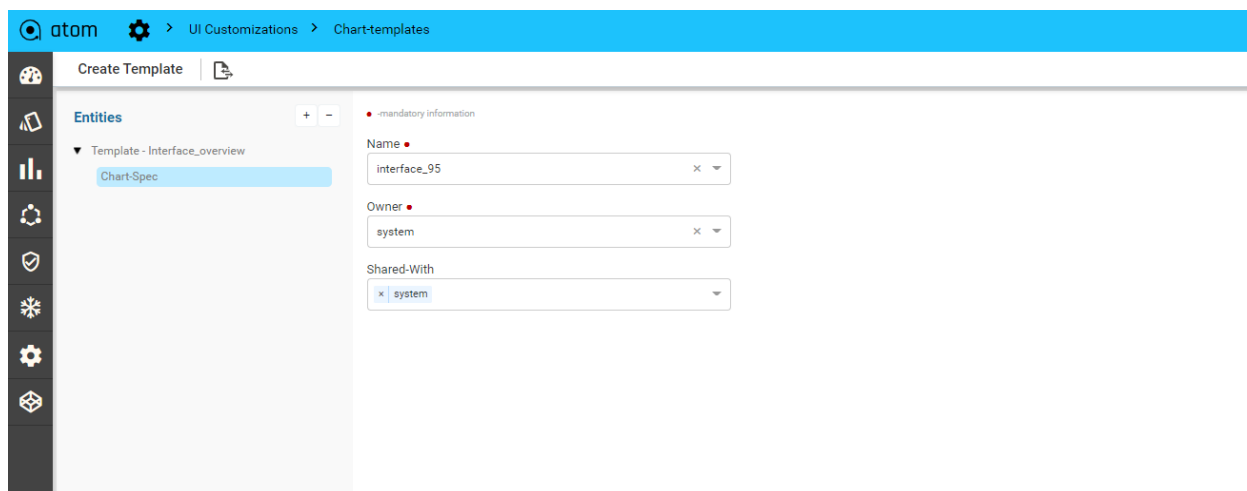
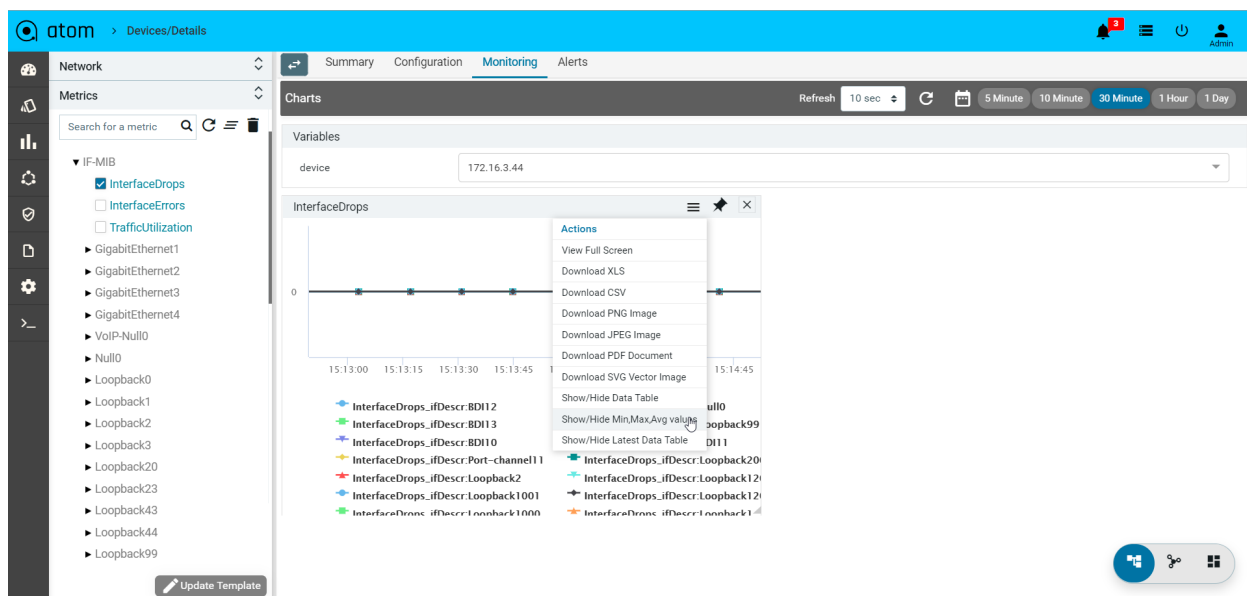


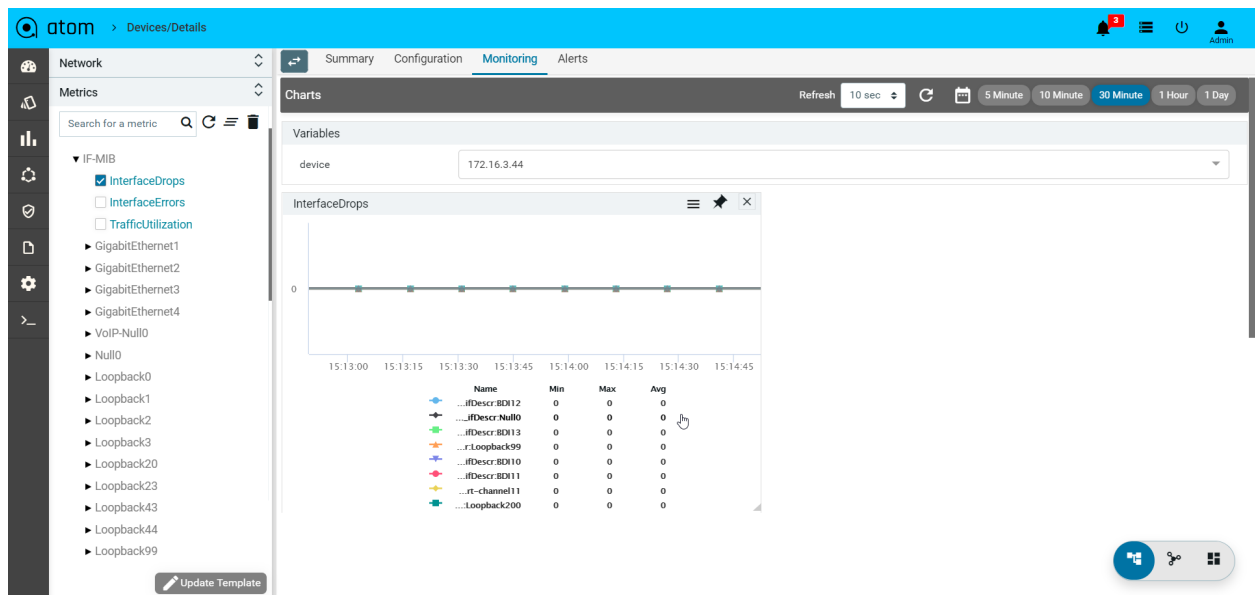
Chart Actions:

Each chart can be seen in full screen, downloaded in PNG, JPEG, SVG formats, export Time series data in csv or xls format, min/max/avg can be seen for each time series and also latest snapshot of time series data can be seen in a table below the graph. Currently, time selection is global which is applicable to all the charts or templates which are selected.

To see available actions for each chart.. Click on three vertical lines as shown below.



min/max/avg:

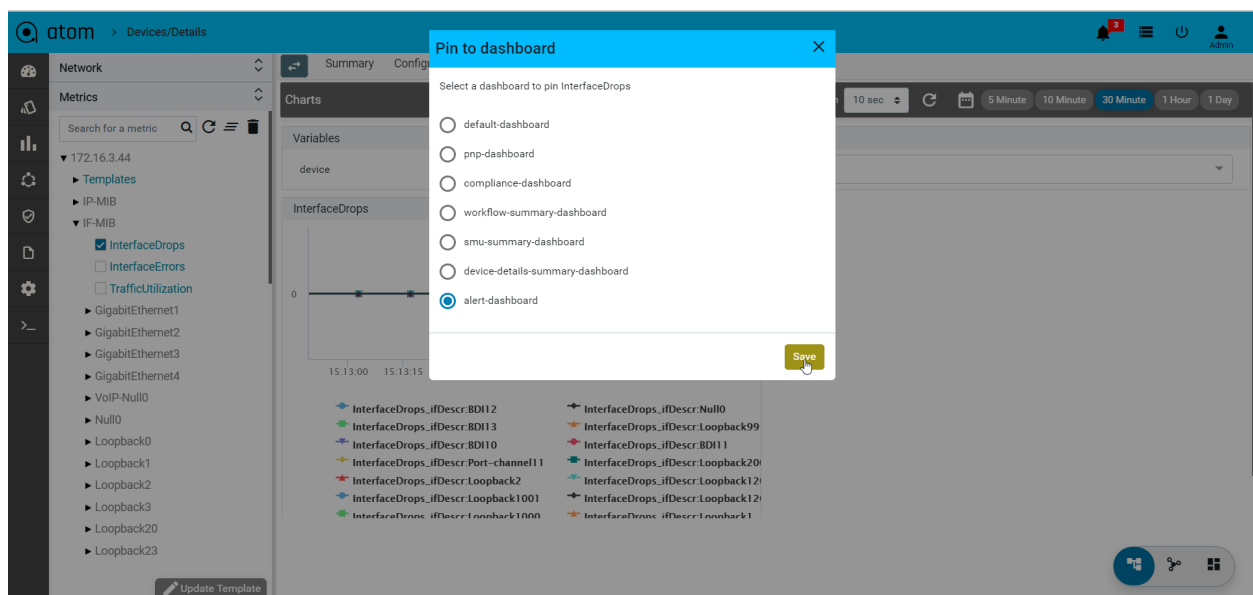
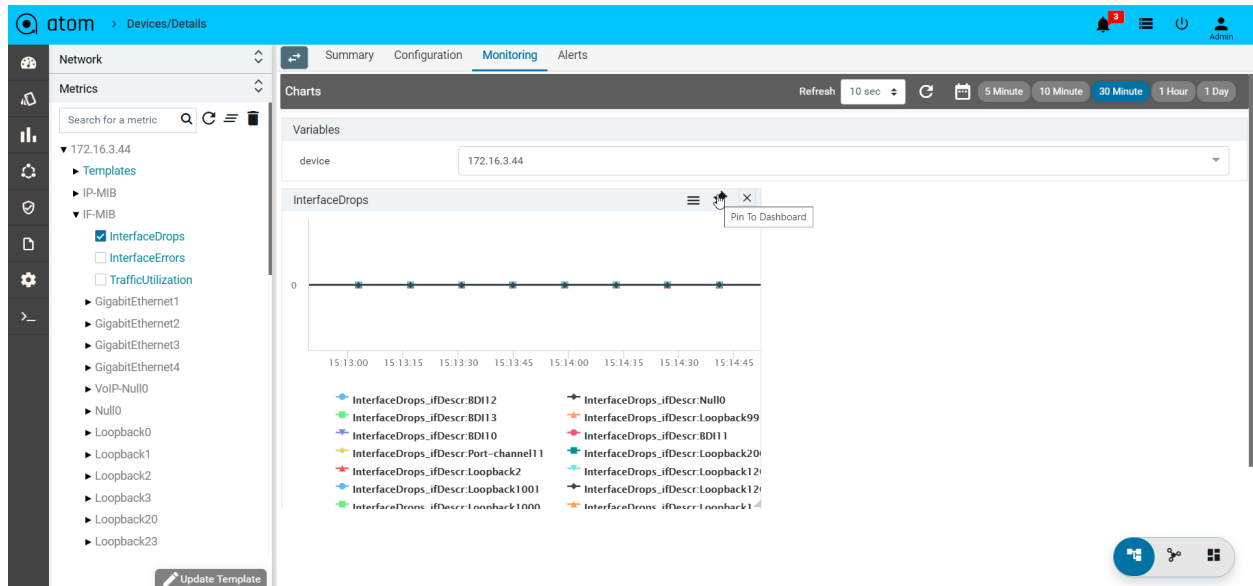


Latest data in table format:

The screenshot shows the ATOM Monitoring page with a table of the latest data for InterfaceDrops metrics. The table has two columns: 'Timestamp' and 'Legend'.

Timestamp	Legend
2020-06-16 15:15:09	InterfaceDrops_ifDescr:BDI12
2020-06-16 15:15:09	InterfaceDrops_ifDescr:Null0
2020-06-16 15:15:09	InterfaceDrops_ifDescr:BDI13
2020-06-16 15:15:09	InterfaceDrops_ifDescr:Loopback99
2020-06-16 15:15:09	InterfaceDrops_ifDescr:BDI10
2020-06-16 15:15:09	InterfaceDrops_ifDescr:BDI11
2020-06-16 15:15:09	InterfaceDrops_ifDescr:Port-channel11

Chart pinning to a dashboard



Note: To have the static values for the variables in monitoring charts, create the Advanced charts by following the below steps.

1. Create Advanced chart

Create Custom Chart

✓

✕

Basic

Advanced

Name

JunosTemp

Apply to current device

☐

Platform

ALL|ALL|ALL|ALL|Juniper Networks

✕ ▼

Device Group

✕ AllDevices

▼

Unit

Enter the unit

Query Type

Widget Type

Timeseries

Instant

Line

Table

Owner

system

✕ ▼

Shared with

✕ system

▼

Queries

+

2. Provide the variables as follows in the queries.

Create Custom Chart

✓

✕

Owner

system

✕

▼

Shared with

✕ system

▼

Queries

+

Name

✕

Temp

Query

jnxOperatingTemp{device=\$device,jnxOperatingContentsIndex=\$jnxOperatingContentsIndex,jnxOperatingL1Index=\$jnxOperatingL1Index}

Legend Name

Enter legend name (legend name should be unique for the chart)

Variables

+

device

10.225.254.101

✕

jnxOperatingContentsIndex

20

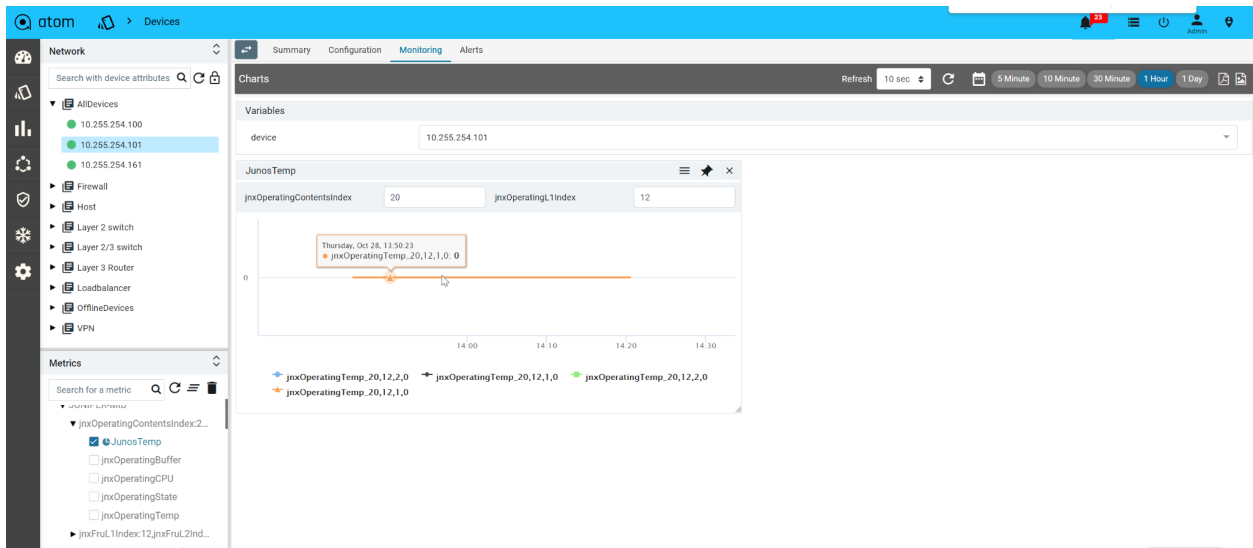
✕

jnxOperatingL1Index

12

✕

3. Charts can be viewed with static values as below.



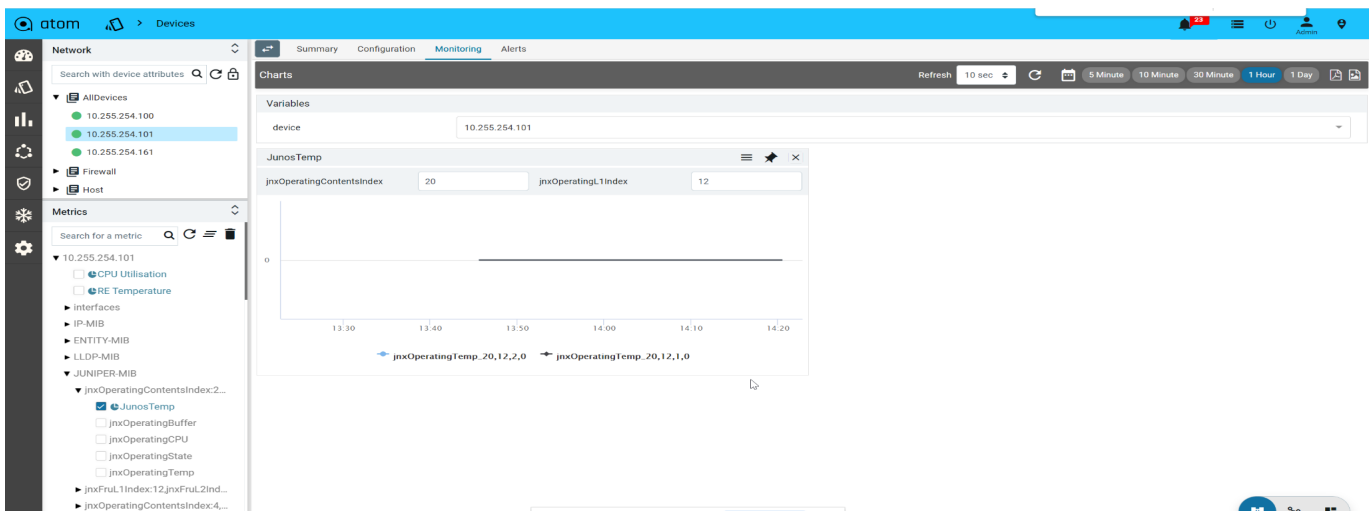
Alert, Actions & Closed Loop Automation

ATOM enables you to create Assurance profiles to facilitate 24x7 uptime of your network. Closed loop automation (CLA) framework allows you to define policies and remediation actions in violation of those policies.

ATOM collects operational & performance metrics from multiple data sources such as SNMP, SNMP traps, Syslog and Streaming Telemetry and stores them in a time-series database.

Following are the different activities on the metrics:

- Visualize Data Using Charts & Reports - To be covered in [Monitoring](#)
- Alerts against thresholds defined on the Metrics
- Alert Dashboards - Collection of Predefined & User Defined Dashlets
- Alert Routing to Email, Slack etc.,
- Actions on Alerts
- Closed Loop Automation Actions on the Alerts - Alerts can be routed to ATOM CLA framework where further correlation can be done against the defined behavior and remediation steps to resolve issues in the network. The actions to be taken for any breach in threshold values range from sending an email alert, a slack notification or



raising the alarm in the system. For more advanced use cases, you can define auto-remediation actions such as executing a workflow to shut down an interface due to excessive BGP neighbor flaps or seek approvals by sending requests to ITSM tools such as Service Now. Each action can be triggered automatically or on demand by the user after analysing the event. Actions can be attached as generic (all events) or restricted scope by using alert filters.

Follow the below steps to get the event driven closed loop automation works.

1. Define the Alert Rule - When to trigger an alert
 - a. Each alert rule consists of set of conditions, severity, duration to raise or clear an alert
2. Create the Alert Filter to Alert definition - Who gets affected
 - a. It defines the scope for actions such as node level, severity based, location or hybrid etc. These are reusable entities.
3. Attach the action with suitable filter and type - What needs to be done after the event
 - a. Attach the actions such as email, remediation, RCA/Diagnostics on set of alerts

Below are some of the use cases:

1. Generate an alert for the interface packet drop events & diagnose the issue (by performing a set of operational checks on the device through workflow) and perform the right action such as upgrading the link or changing the QoS policy etc..
2. Notify the management team on all critical events which needs immediate attention.
3. Send location specific alerts to corresponding slack channels and email groups.
4. Raise a servicenow incident for the hardware issues
5. Alert BGP adjacency state stuck issues and diagnose & perform the available actions (check & change BGP port issues and AS Numbers, Peer IP, update source etc..) on demand with manual triggers.
6. Create a dashboard with long standing issues on specific resource pools which are triggered from cisco IOS XR streaming telemetry sources.

Define Alert Rules

1. Navigate to **Assurance > Closed Loop Automation** to define the alert rules
2. Click **Add** and provide values for Name, Description, Message, Resource Path
 - a. **Name** : Name of the alert definition which will contain the set of rules and associated actions
 - b. **Description** : Provide the description about the what the rule is about
 - c. **Message**: Define the message template which you would want to see as part of notification with variables replaced with corresponding values. It is based on Prometheus Go Templating.
https://prometheus.io/docs/prometheus/latest/configuration/template_examples/
 - d. **Resource Path**: Provide the unique XPATH using alert resulted variables. Resource Path and Name are the composite key to define the alert ingestion, maintain the history and resolve events appropriately. To use variables in the resource path, keep key names inside curly braces.check the following example:
Device={device}/Interface_name={ifDescr}. Here, device and ifDescr are the variables
Eg: Notifying each Interface Flap

- i. **Name : *InterfaceDown* and Resource Path : */device/interfacename* ,**
 1. all the interface alerts will be treated as similar. If Gi1 Interface goes down on 10.0.0.1 and Gi2 comes to operationally UP on 10.0.0.2 then it will clear the Gi1 Active alert on other devices.
- ii. **Name : *InterfaceDown* and Resource Path : */device={device}* then all the interface alerts will be aggregated by device**
 1. Every InterfaceDown alert of the same device will become the history of other alerts and clear the events raised by other interfaces on that device.

Alert Name	Device/NodeName	Device Name	Device FQDN	Resource	Severity	Ack	Status	Message
Interfaceflap	172.16.3.44	wbucbb-burbank0-gw.net.d...		/controller:devices/device=172.16.3.44/interf...	Major	⊗	Open	Interface has b...
Interfaceflap	172.16.3.44	wbucbb-burbank0-gw.net.d...		/controller:devices/device=172.16.3.44/interf...	Major	⊗	Open	Interface has b...

In both (i) & (ii) cases, manual action can be performed only on the latest instance of that since they are grouped by upper level or ad hoc entities.

- iii. **Name : *InterfaceDown* and Resource Path : */device={device}/Interface_name={ifDescr}* then all the interface alerts will be aggregated by device & Interface**

Name: Interfaceflap

Description: Notify Interface Flaps to NetOps Team

Message: Interface {{ \$device.\$desc }} has been flapping more than 5 times in the last 5 mins on Device {{ \$device.device }}

Resource Path: /controller/devices/device={{device}}/interface/interfaces/interface={{desc}}

Type: NETWORK

Is Subscribe Rule: ☒

Tags: Enter a new tag

Permissions: ☒ Propagate Read Permission
☒ Propagate Write Permission

Owner: system

Shared With: system

Conditions:

Metric / Expression (Owner label for all metric is Mandatory)	Operator	Value	Severity	Hour	Minute
changes(/OperatorStatus(Owner="system"))[5m]>=2	==	true	Critical	0	0
changes(/OperatorStatus(Owner="system"))[5m]>=1	==	true	Major	0	0
changes(/OperatorStatus(Owner="system"))[5m]>=0	==	true	Resolved	0	0

Actions: Select a value

Troubleshoot: Chart Type: Metric Chart, Custom Chart, Chart Template

Filter Group: 1 Critical Total

Severity: Critical **Ack Status:** Both **Resolved:** Closed **Alert Group:** Primary **Alert Group Name:** Name: Interfaceflap **Type:** Network **Source:**

Alert Name	Device/Node Name	Device Name	Device FQDN	Resource	Severity	Ack	Status	Message
Interfaceflap	172.16.4.99	mx204-ztp.anutacorp.com	/controller/devices/device=172.16.4.99/interf...		Critical	⊗	Closed	Interface has been flapping more than 5 times

- It makes every event independent and manual actions can be performed at any time since we have the right set of affected object information available instead of just the latest event from that group.
 - Auto clearing of events will happen properly. If Gi1 goes down on device 10.0.0.1 will raise an alert and it gets cleared only when it comes operationally on the same Interface on the correct device.
- Type:** By Default, it is of type *NETWORK*. Please, don't switch this flag for device related alert rules as it defines which database query for metrics and alerts.
 - Tags:** Provide a new tag name or choose the existing ones. it will be useful to create or apply the filter efficiently in an aggregated way.
- Configure the condition for each rule with the required thresholds.
 - Metric/Expression:** Provide the metric or expression in the condition

- i. If the condition has the expression with boundaries then no need to change the operator and value. It should be == true in that case.
 - ii. If the condition has only the metric or expression which doesn't have any logical boundaries then operator and value will be useful. Refer below snapshot for more details.
- b. **Severity:** A severity level can be attached to the alerts if the sensors being monitored breach the threshold values.
 - c. **Duration:** Alerts will be fired once they have been returned for this long

Conditions

Metric / Expression(Owner label for all metric is Mandatory)	Operator	Value	Severity	Hour	Minute
changes(ifOperStatus[owner="system"])[5m]==2	==	true	Critical	0	0
changes(ifOperStatus[owner="system"])[5m]==1	==	true	Major	0	0
changes(ifOperStatus[owner="system"])[5m]==0	==	true	Resolved	0	0

Preview Rule

Preview

```
group:
  name: Interfaceflap
  alert: Interfaceflap-1
    expr: changes(ifOperStatus[owner="system"])[5m]==2
    for: 0m
    labels:
      severity: CRITICAL
      alertDefName: Interfaceflap
    annotations:
      description: Interface {{ $labels.ifDescr }} has been flapping more than 5 times in the last 5 mins on Device {{ $labels.device }}
  alert: Interfaceflap-2
    expr: changes(ifOperStatus[owner="system"])[5m]==1
    for: 0m
    labels:
      severity: MAJOR
      alertDefName: Interfaceflap
    annotations:
      description: Interface {{ $labels.ifDescr }} has been flapping more than 5 times in the last 5 mins on Device {{ $labels.device }}
```

4. 'Live Preview' shows the generated prometheus rule with the defined thresholds and its metadata.

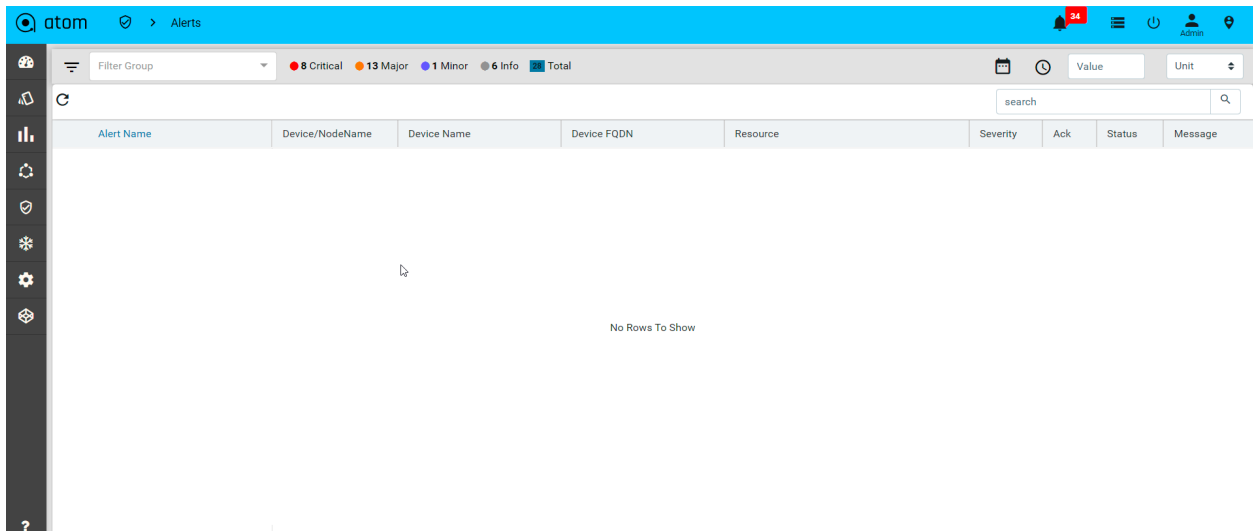
Define Alert Filter

Alert filters are to define the scope for any action to be performed or group them together based on certain criteria. We have the following attributes to create the filters : Name, Acknowledge State, Resolved State, Alert Name, Severity, Device-Groups, Resource Pools, Locations, Tags and Relative or Absolute times.

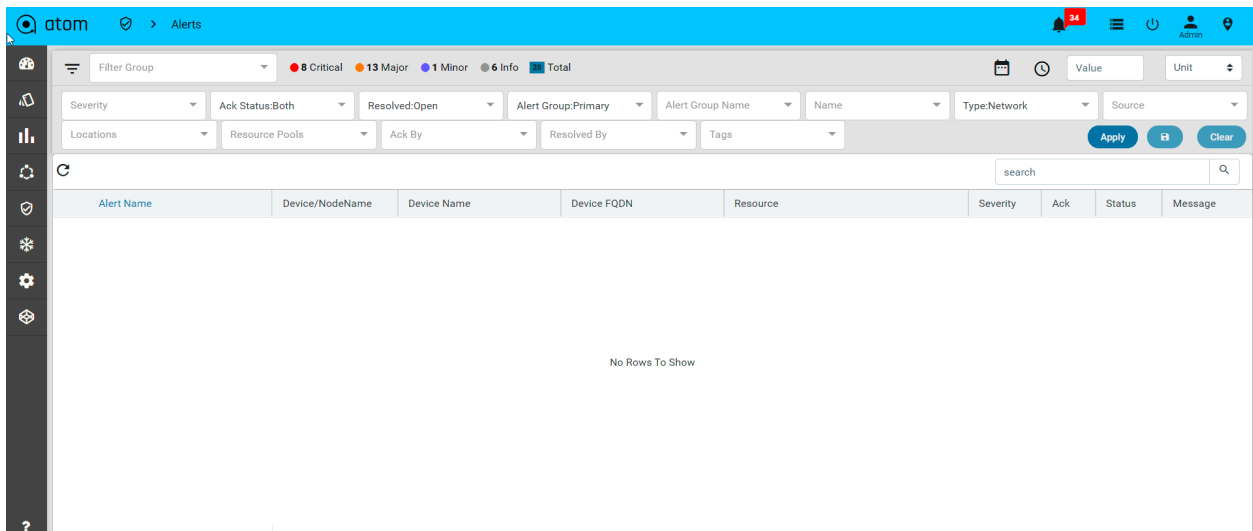
Follow below steps to create and view filters:

1. Navigate to Assurance -> Alerts
2. Click on top left corner icon to see the available attributes to define the alert filter
3. Choose the values from the dropdowns and apply the filter to see existing alerts.
4. If step #3 provides the desired filtered result then go ahead and save the filter by providing the name and optionally save as a dashlet in the dashboard. Refer to the Dashboards section for more details.

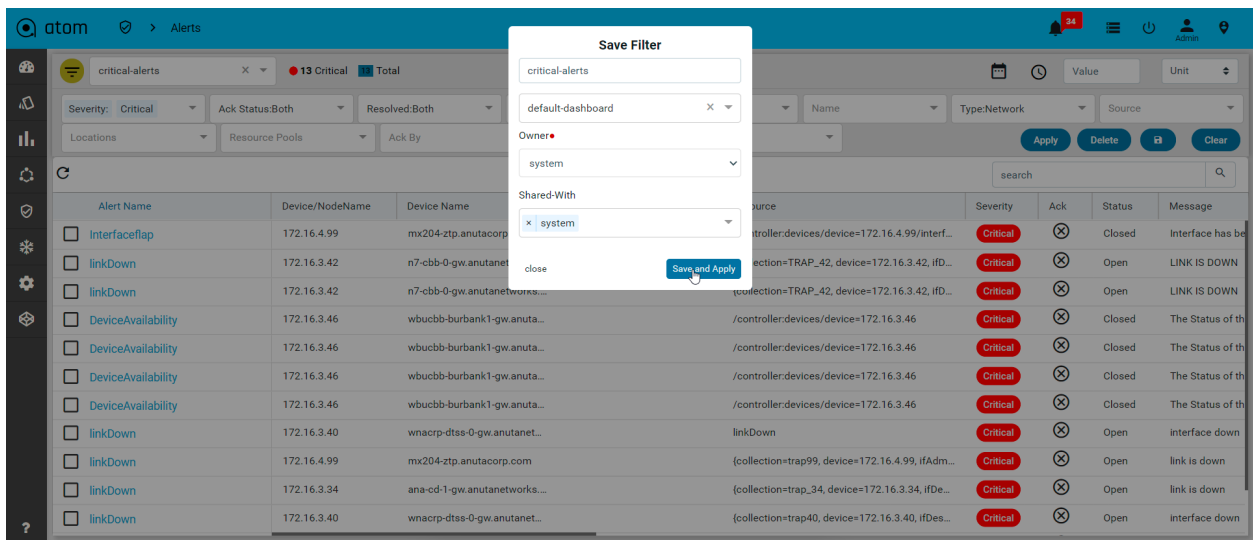
Navigate to the top left corner and click on the filter icon to see possible filter attributes.



Choose the filter attributes and click on the Save button to **'Save Filter'**.

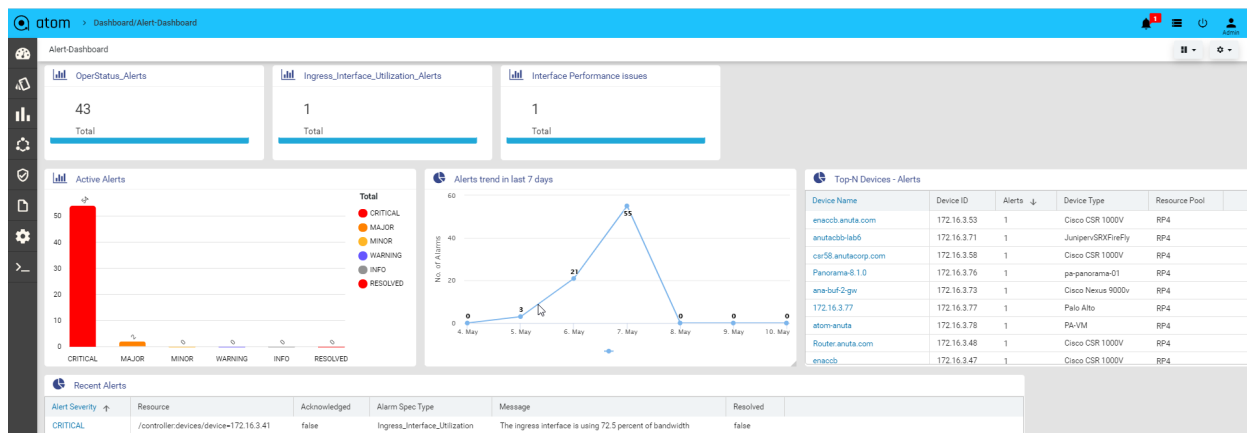
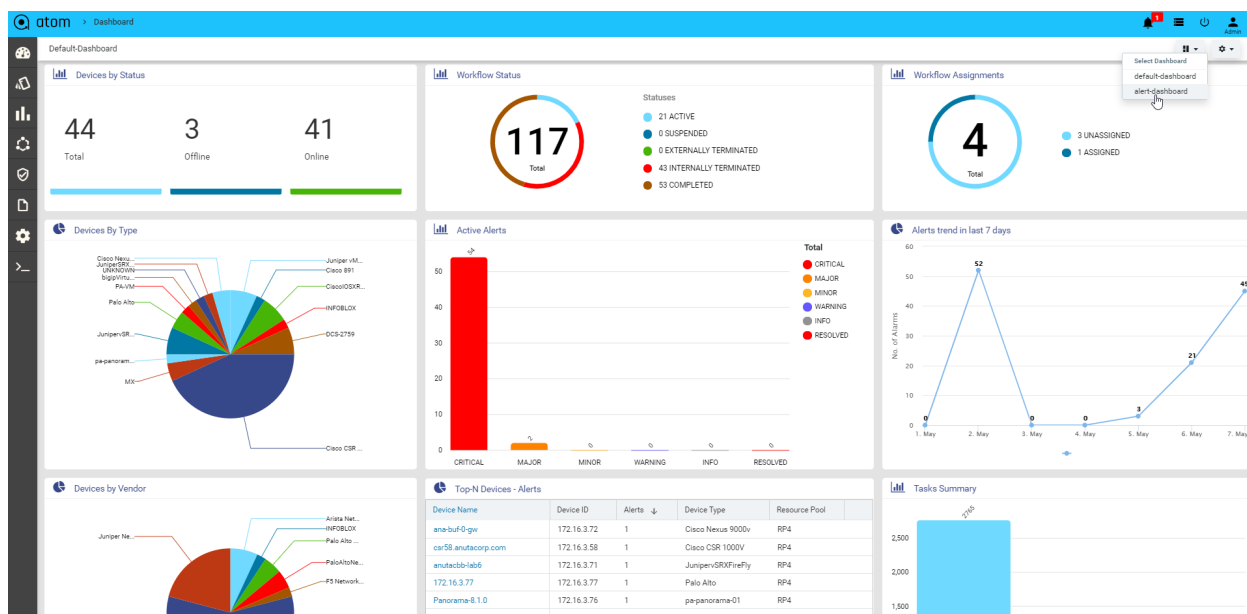


Save filter will provide an option to provide a name and to attach to the dashboard.



To clear selected filter attributes, click on the **Clear** button in left down corner

Navigate Dashboard> select alert-Dashboard in right corner and click on it to view the saved filter dashboard



Attach Remediation Actions

Once you define the baseline behavior by specifying the sensors to be monitored, the next step is to define the actions to be automated. The actions range from simple notifications to complex remediation actions. Here, we will bridge the alert to action behavior by using alert filters.

The steps below will help you define the necessary actions for the rules defined

1. Navigate to the Assurance -> Closed Loop Automation
2. Click on the Alert Actions tab
3. Click the Add (+) button to create the action template and provide the Name, Description.
 - a. **Name** : provide unique name to for action template
 - b. **Description** : provide the description of action and what is does

- c. **Trigger For** : It is used to define scope of action. *All Filters* will enable this action for all the alerts and *Alert filters* will restrict scope to defined criteria as per selected named filters.
- d. **Alert Filters**: Select the name of the alert filter to attach an action
- e. **Type**: choose the type of action such as slack, email and workflow or rpc
- f. **Trigger: Auto** Trigger is used to perform closed loop automation use cases and **Manual** trigger is used to perform on demand diagnostics and notifications. However, any type of trigger can be used based on the use case and criticality of the affect.
- g. **Is Action Enabled**: It is to make sure the defined Action template is ready to attach for an alert or not. If we disable this, this particular action template won't be shown or get triggered when the event happens.

Below is the snapshot to send an email notification for every alert

Create Alert Action

Name

Enter Action Name

email-notification

Description

Enter Action Description

Description

Trigger For

All Alerts

Alert Filters

Alerts Filters

majoralerts

Type

Email

Email To

Email@company.com

test@mail.com

Subject

Enter Rule Description

Alert{{definitionName}}Status{{status}}Severity{{severity}}

Trigger

Auto

Manual

Is Action Enabled

Owner

system

Shared-With

x

system

If Action is enabled then those all enabled actions will show in Alerts action window as shown below

The screenshot displays the ATOM web interface. The top section, 'Alert Rules', lists various rules with columns for Name, Description, Type, Trigger, Action Enabled, Owner, and Shared With. Rules include 'Execute Commands and Update Ticket', 'InterfaceUp-FromServiceNow', 'Ping - From ATOM', 'Raise ServiceNow Ticket', 'Raise Update ServiceNow Ticket', 'Traceoute - From ATOM', 'email-notification', 'jira', 'juniper_interface_shutdown', 'rpc', 'slack-alerts', and 'slack-notification'.

The bottom section, 'Alerts', shows a list of alerts with columns for Device FQDN, Resource, Severity, Ack, Message, and Source. A dropdown menu for 'Actions' is open, showing options like 'OpenSt', 'Raise/Update ServiceNow Ticket', 'Ping - From ATOM', 'shut_interface', 'Raise ServiceNow Ticket', 'Traceoute - From ATOM', and 'OperStatus_CLA'. The right sidebar provides 'Alert details | OperStatus_CLA' with information about the resource, device, severity, and a list of actions taken, such as 'shut_interface' and 'Raise/Update ServiceNow Ticket'.

Actions

Slack

1. **Slack Channel:** Provide the channel name
2. **Webhook:** Provide the slack channel webhook, Refer below link to create incoming webhook in slack https://api.slack.com/messaging/webhooks#posting_with_webhooks
3. **Body:** Provide the template to format the message while notifying via slack, here you can use the keys from alert

Name •
Enter Action Name
slack-notification

Description
Enter Action Description
Description

Trigger For
All Alerts Alert Filters

Type •
Slack

Slack Channel •
alert-notifications

Webhook •
<https://hooks.slack.com/services/T02TAQP5R/BRZUD2480/ubnFuc0F0uxMqoiBV9>

Body •
Enter Rule Description
Alert name: {{definitionName}}
Severity: {{severity}}
Resource: {{resource}}

Trigger
Auto Manual

Is Action Enabled
☐

Owner •
system

Shared-With
x system.*

Below the slack notification

alert-notifications
2:49 PM

Alert name: OperStatus_CLA
Severity: MAJOR
Resource: /controller:devices/device=10.83.153.40/interface:interfaces/interface=x-3/0/0
Device: 10.83.153.40
Message: Interface has been flapping more than 5 times in the last 5 mins on Device 10.83.153.40
Status: Raised

2:49
Alert name: OperStatus_CLA
Severity: MAJOR
Resource: /controller:devices/device=10.83.153.38/interface:interfaces/interface=x-1/0/1.0
Device: 10.83.153.38
Message: Interface has been flapping more than 5 times in the last 5 mins on Device 10.83.153.38
Status: Raised

Alert name: OperStatus_CLA
Severity: MAJOR
Resource: /controller:devices/device=10.83.153.38/interface:interfaces/interface=x-1/0/1
Device: 10.83.153.38
Message: Interface has been flapping more than 5 times in the last 5 mins on Device 10.83.153.38
Status: Raised

Email

1. Email to: Provide the email address
2. Subject: Provide the subject for an email

Create Alert Action

Name

Enter Action Name

email-notification

Description

Enter Action Description

Description

Trigger For

All Alerts

Alert Filters

Alerts Filters

majoralerts

Type

Email

Email To

Email@company.com

test@mail.com

Subject

Enter Rule Description

Alert{{definitionName}}Status{{status}}Severity{{severity}}

Trigger

Auto

Manual

Is Action Enabled

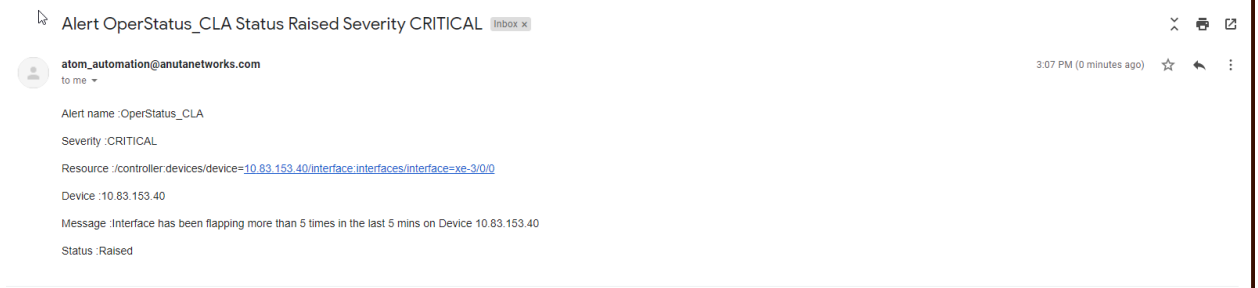
Owner

system

Shared-With

x

system

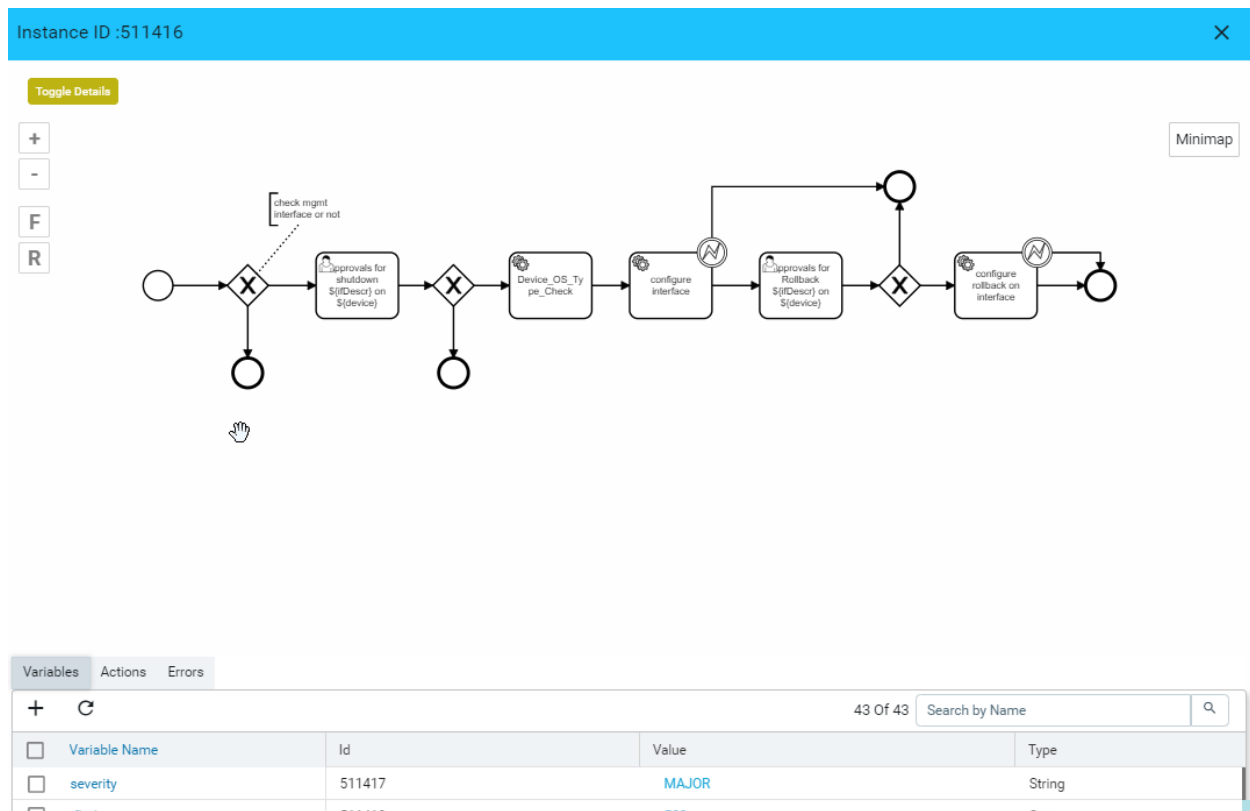


Currently, templatzation for email body is not supported in ATOM.

Workflow

Workflow: Choose the available workflow by its name to diagnose or remediate the issue or refer the workflow section to create new workflow and packaging

The screenshot displays the ATOM workflow editor interface. On the left, a list of alerts is shown, with 'OperStatus_CLA' selected for device 10.83.153.40. The main area shows a workflow diagram for 'Instance ID: 512172'. The workflow starts with a decision diamond 'check mgmt interface or not'. If 'Yes', it proceeds to 'approve for shutdown (approved on S(device))', then 'Device OS Type Check', 'configure interface', 'approve for Rollback (approved on S(device))', and finally 'configure rollback on interface'. If 'No', it proceeds to 'configure interface', 'approve for Rollback (approved on S(device))', and finally 'configure rollback on interface'. The workflow ends with a final connector. Below the diagram, a table shows variables: 'Variable Name' (severity), 'Id' (512173), 'Value' (CRITICAL), and 'Type' (String). On the right, the 'Alert details' panel for 'OperStatus_CLA' shows resource, device, severity, acknowledged status, source, and message. Below this, the 'Action taken' section lists several notifications (shut, Interface; slack notification; email notification) with their completion status (IN PROGRESS or COMPLETE).



RPC

1. **RPC:** choose the available RPC's or refer the developer guide to create new rpc's
2. **Inputs:** Choose input method key-value or json, xml to provide the payload

Create Alert Action

Name

Enter Action Name

rpc

Description

Enter Action Description

Description

Trigger For

All Alerts

Alert Filters

Type

Rpc

RPC

developerutils:alertaction-test-rpc

Inputs

Key Value

JSON

XML

Key

Value

input

key-value-input

+

Trigger

Auto

Manual

Is Action Enabled

Owner

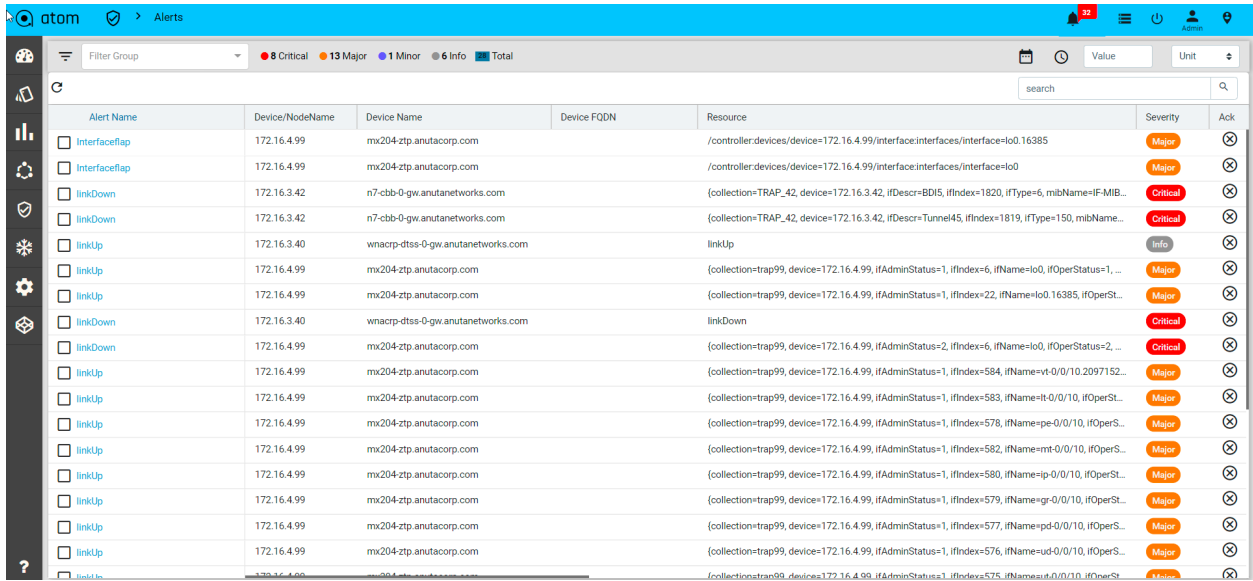
system

Shared-With

x system

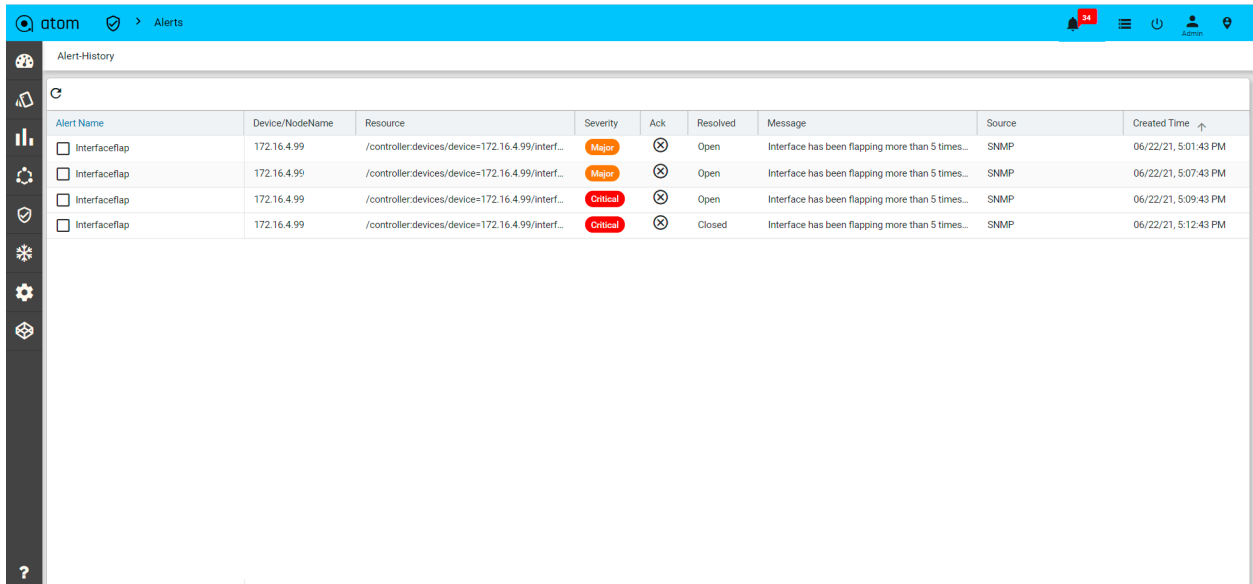
Alerts Summary

Below is the snapshot for alert summary



Alert Name	Device/NodeName	Device Name	Device FQDN	Resource	Severity	Ack
<input type="checkbox"/> Interfaceflap	172.16.4.99	mx204-ztp.anutacorp.com		/controller.devices/device=172.16.4.99/interface.interfaces/interface=lo0.16385	Major	<input type="checkbox"/>
<input type="checkbox"/> Interfaceflap	172.16.4.99	mx204-ztp.anutacorp.com		/controller.devices/device=172.16.4.99/interface.interfaces/interface=lo0	Major	<input type="checkbox"/>
<input type="checkbox"/> linkDown	172.16.3.42	n7-cbb-0-gw.anutanetworks.com		{collection=TRAP_42, device=172.16.3.42, ifDescr=BDI5, ifIndex=1820, ifType=6, mibName=IF-MIB...	Critical	<input type="checkbox"/>
<input type="checkbox"/> linkDown	172.16.3.42	n7-cbb-0-gw.anutanetworks.com		{collection=TRAP_42, device=172.16.3.42, ifDescr=Tunnel45, ifIndex=1819, ifType=150, mibName=...	Critical	<input type="checkbox"/>
<input type="checkbox"/> linkUp	172.16.3.40	wnacrp-dtss-0-gw.anutanetworks.com		linkUp	Info	<input type="checkbox"/>
<input type="checkbox"/> linkUp	172.16.4.99	mx204-ztp.anutacorp.com		{collection=trap99, device=172.16.4.99, ifAdminStatus=1, ifIndex=6, ifName=lo0, ifOperStatus=1, ...	Major	<input type="checkbox"/>
<input type="checkbox"/> linkUp	172.16.4.99	mx204-ztp.anutacorp.com		{collection=trap99, device=172.16.4.99, ifAdminStatus=1, ifIndex=22, ifName=lo0.16385, ifOperSt...	Major	<input type="checkbox"/>
<input type="checkbox"/> linkDown	172.16.3.40	wnacrp-dtss-0-gw.anutanetworks.com		linkDown	Critical	<input type="checkbox"/>
<input type="checkbox"/> linkDown	172.16.4.99	mx204-ztp.anutacorp.com		{collection=trap99, device=172.16.4.99, ifAdminStatus=2, ifIndex=6, ifName=lo0, ifOperStatus=2, ...	Critical	<input type="checkbox"/>
<input type="checkbox"/> linkUp	172.16.4.99	mx204-ztp.anutacorp.com		{collection=trap99, device=172.16.4.99, ifAdminStatus=1, ifIndex=584, ifName=vt-0/0/10.2097152...	Major	<input type="checkbox"/>
<input type="checkbox"/> linkUp	172.16.4.99	mx204-ztp.anutacorp.com		{collection=trap99, device=172.16.4.99, ifAdminStatus=1, ifIndex=583, ifName=rt-0/0/10, ifOperSt...	Major	<input type="checkbox"/>
<input type="checkbox"/> linkUp	172.16.4.99	mx204-ztp.anutacorp.com		{collection=trap99, device=172.16.4.99, ifAdminStatus=1, ifIndex=578, ifName=pe-0/0/10, ifOperSt...	Major	<input type="checkbox"/>
<input type="checkbox"/> linkUp	172.16.4.99	mx204-ztp.anutacorp.com		{collection=trap99, device=172.16.4.99, ifAdminStatus=1, ifIndex=582, ifName=mt-0/0/10, ifOperSt...	Major	<input type="checkbox"/>
<input type="checkbox"/> linkUp	172.16.4.99	mx204-ztp.anutacorp.com		{collection=trap99, device=172.16.4.99, ifAdminStatus=1, ifIndex=580, ifName=ip-0/0/10, ifOperSt...	Major	<input type="checkbox"/>
<input type="checkbox"/> linkUp	172.16.4.99	mx204-ztp.anutacorp.com		{collection=trap99, device=172.16.4.99, ifAdminStatus=1, ifIndex=579, ifName=gr-0/0/10, ifOperSt...	Major	<input type="checkbox"/>
<input type="checkbox"/> linkUp	172.16.4.99	mx204-ztp.anutacorp.com		{collection=trap99, device=172.16.4.99, ifAdminStatus=1, ifIndex=577, ifName=pd-0/0/10, ifOperSt...	Major	<input type="checkbox"/>
<input type="checkbox"/> linkUp	172.16.4.99	mx204-ztp.anutacorp.com		{collection=trap99, device=172.16.4.99, ifAdminStatus=1, ifIndex=576, ifName=ud-0/0/10, ifOperSt...	Major	<input type="checkbox"/>
<input type="checkbox"/> linkUp	172.16.4.99	mx204-ztp.anutacorp.com		{collection=trap99, device=172.16.4.99, ifAdminStatus=1, ifIndex=575, ifName=sd-0/0/10, ifOperSt...	Major	<input type="checkbox"/>

To see the alert history, click on the entry.



Alert Name	Device/NodeName	Resource	Severity	Ack	Resolved	Message	Source	Created Time
<input type="checkbox"/> Interfaceflap	172.16.4.99	/controller.devices/device=172.16.4.99/interf...	Major	<input checked="" type="checkbox"/>	Open	Interface has been flapping more than 5 times...	SNMP	06/22/21, 5:01:43 PM
<input type="checkbox"/> Interfaceflap	172.16.4.99	/controller.devices/device=172.16.4.99/interf...	Major	<input checked="" type="checkbox"/>	Open	Interface has been flapping more than 5 times...	SNMP	06/22/21, 5:07:43 PM
<input type="checkbox"/> Interfaceflap	172.16.4.99	/controller.devices/device=172.16.4.99/interf...	Critical	<input checked="" type="checkbox"/>	Open	Interface has been flapping more than 5 times...	SNMP	06/22/21, 5:09:43 PM
<input type="checkbox"/> Interfaceflap	172.16.4.99	/controller.devices/device=172.16.4.99/interf...	Critical	<input checked="" type="checkbox"/>	Closed	Interface has been flapping more than 5 times...	SNMP	06/22/21, 5:12:43 PM

Manual actions will be shown on click of alert record with vertical ellipsis on top

The screenshot shows the ATOM Alerts interface. The top bar includes the ATOM logo, a shield icon, and the word 'Alerts'. Below this is a filter group section with a dropdown menu and a summary of alert counts: 13 Critical, 18 Major, 1 Minor, and 6 Info. A table of filters is visible, including Severity, Ack Status, Resolved Both, Alert Group, and Name. The main table lists alerts with columns for Alert Name, Device/NodeName, Device Name, Device FQDN, Resource, and Severity. The 'Interfaceflap' alert is selected, showing details on the right side. The details panel includes information about the Resource, Device, Severity, Acknowledged status, Source, and Message. It also shows a timeline of actions taken, including 'Ping - From ATOM' and 'Juniper_workflow_shutdown_the_interface', both marked as 'COMPLETE'.

Alert Name	Device/NodeName	Device Name	Device FQDN	Resource	Severity
Interfaceflap	172.16.3.44	wbucbb-burbank0-gw.net.dl...		/controller.devices/device=172.16.3.44/interf...	Major
Interfaceflap	172.16.4.99	mx204-ztp.anutacorp.com		/controller.devices/device=172.16.4.99/interf...	Critical
Interfaceflap	172.16.4.99	mx204-ztp.anutacorp.com		/controller.devices/device=172.16.4.99/interf...	Major
Interfaceflap	172.16.4.99	mx204-ztp.anutacorp.com		/controller.devices/device=172.16.4.99/interf...	Major
Interfaceflap	172.16.3.44	wbucbb-burbank0-gw.net.dl...		/controller.devices/device=172.16.3.44/interf...	Major
Interfaceflap	172.16.3.44	wbucbb-burbank0-gw.net.dl...		/controller.devices/device=172.16.3.44/interf...	Major
Interfaceflap	172.16.3.44	wbucbb-burbank0-gw.net.dl...		/controller.devices/device=172.16.3.44/interf...	Major
Interfaceflap	172.16.4.99	mx204-ztp.anutacorp.com		/controller.devices/device=172.16.4.99/interf...	Major
linkDown	172.16.3.42	n7-cbb-0-gw.anutanetworks...		(collection=TRAP_42, device=172.16.3.42, ifD...	Critical
linkDown	172.16.3.42	n7-cbb-0-gw.anutanetworks...		(collection=TRAP_42, device=172.16.3.42, ifD...	Critical
DeviceAvailability	172.16.3.46	wbucbb-burbank1-gw.anuta...		/controller.devices/device=172.16.3.46	Critical
DeviceAvailability	172.16.3.46	wbucbb-burbank1-gw.anuta...		/controller.devices/device=172.16.3.46	Critical
DeviceAvailability	172.16.3.46	wbucbb-burbank1-gw.anuta...		/controller.devices/device=172.16.3.46	Critical
DeviceAvailability	172.16.3.46	wbucbb-burbank1-gw.anuta...		/controller.devices/device=172.16.3.46	Critical
linkUp	172.16.3.40	wnacrp-dtss-0-gw.anutanet...		linkUp	Info

Filter objects can be seen on click on the action in top left

This screenshot is similar to the one above, but it shows a filter object in the top left corner. The filter object is a small window that appears when a user clicks on an action in the top left. It contains a list of filter objects, including 'Interfaceflap', 'linkDown', 'DeviceAvailability', and 'linkUp'. The 'Interfaceflap' filter object is selected, showing details about the alert, including the Resource, Device, Severity, Acknowledged status, Source, and Message. The details panel also shows a timeline of actions taken, including 'Ping - From ATOM' and 'Juniper_workflow_shutdown_the_interface', both marked as 'COMPLETE'.

The screenshot displays the 'Alert-Dashboard' with the following components:

- Alert-Dashboard Header:** Includes navigation icons and user information (Admin).
- OperStatus_Alerts:** A card showing 21 total alerts with a blue progress bar.
- Critical_OperStatus_Alerts:** A card showing 3 total alerts with a blue progress bar.
- Active Alerts:** A bar chart showing the count of alerts by severity. The 'MAJOR' category has the highest count at 5. A legend on the right lists severity levels: CRITICAL (red), MAJOR (orange), MINOR (yellow), WARNING (blue), INFO (grey), and RESOLVED (green).
- Alerts trend in last 7 days:** A line chart showing the number of alerts over time from May 4 to May 10. The count starts at 0 and rises to 22 by May 10.
- Top-N Devices - Alerts:** A table showing the top devices by alert count. The first entry is '10.83.153.38' with 5 alerts, categorized as 'MX'.
- Recent Alerts:** A table listing the most recent alerts, including severity, resource path, acknowledgment status, alarm specification type, message, and resolution status.

Inventory

[[Introduction - Chassis, Interface, Software]]

iSoftware Image Management

Image Upgrade Workflow

Upload an Image and work through a Business process to take a device from V1 to V2 with pre-checks and Post-checks to ensure the device is working as expected after the upgrade.

Image Repository

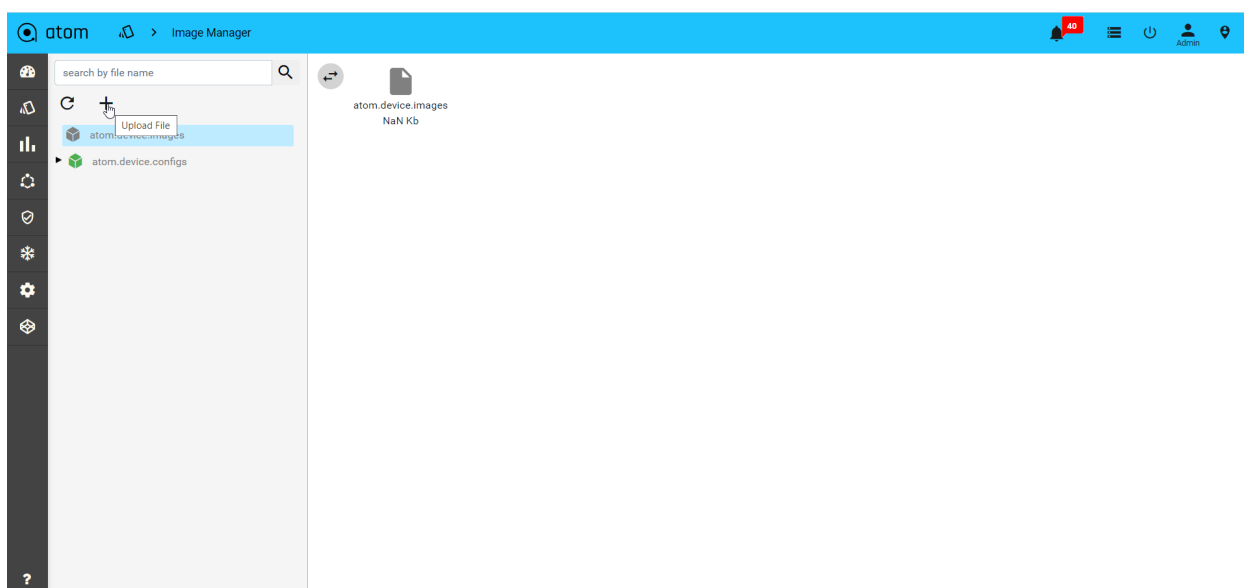
ATOM acts as a Network Element Image Repository (Image Server) as well as Config Repository. Devices can boot an Image from ATOM Image Server manually or through ATOM Network Element Software Image Upgrade Workflow. Image Manager is a replica of the Minio repository. All the images are saved to a minio bucket named 'atom.device.images' and all the configs are saved to 'atom.device.configs'. The supported protocols are Http, anonymous ftp, tftp.

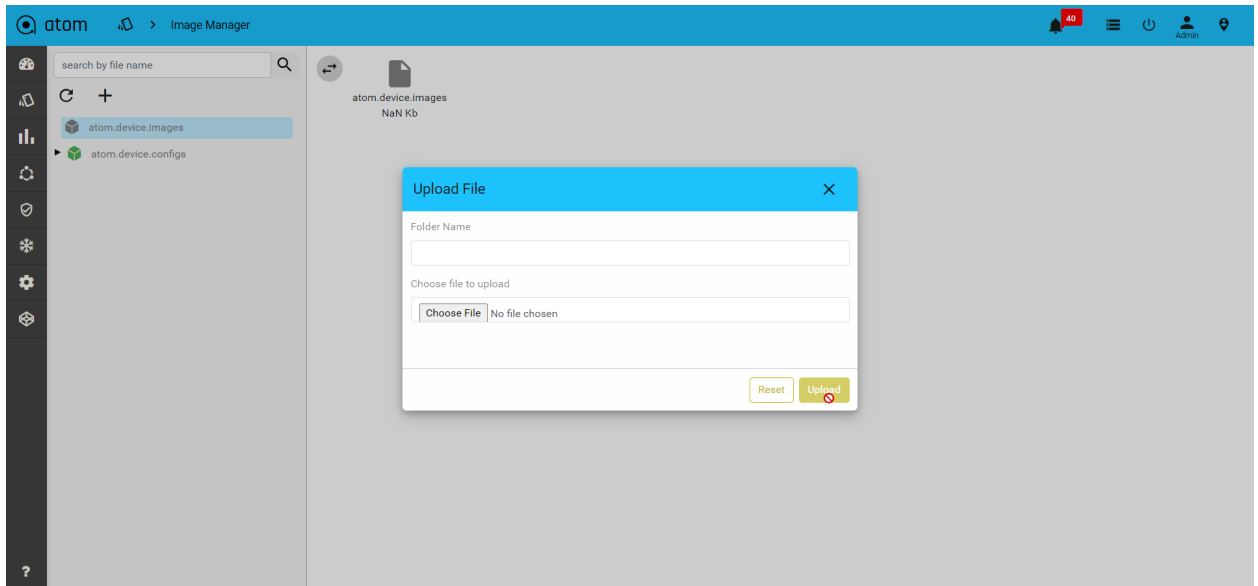
Following steps on ATOM & Upload Images:

1. To upload a file to the Image Manager navigate to **Resource Manager > Image Manager**

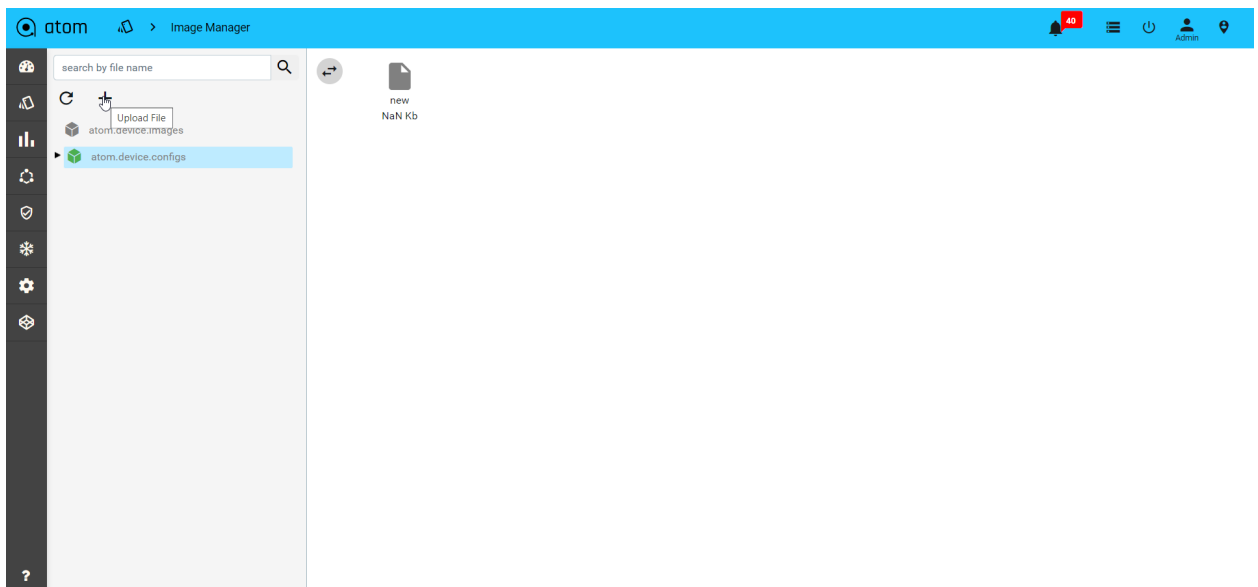


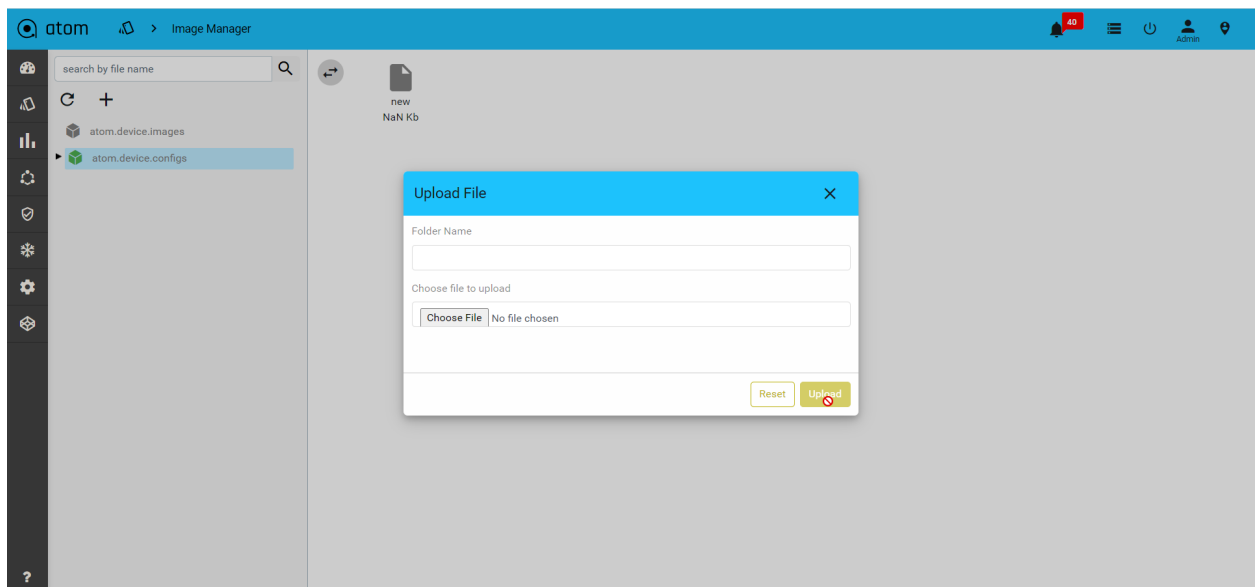
2. To upload image file, click on the upload option after selecting 'atom.device.images' option





3. To upload config file, click on the upload file option after selecting 'atom.device.configs' option





Once files are uploaded to the image-manager then login to the client device, then transfer the files to a particular location on the device by using scp protocol..

Syntax : file copy

`scp://{user-name}@{agent-node-ip}/{path}/{file-name} {Destination path}/{file-name}`

Example :

`scp://admin@10.113.10.44/pub/images/junos-vmhost-install-mx-x86-64-18.3R1.9.tgz
re0:/var/tmp/junos-vmhost-install-mx-x86-64-18.3R1.9.tgz`

```
anuta@Services-PE-OC1-Lab-RE1> file copy scp://admin@10.113.10.44:32222/pub/images/junos-vmhost-install-mx-x86-64-18.3R1.9.tgz re0:/var/tmp/junos-vmhost-install-mx-x86-64-18.3R1.9.tgz
The authenticity of host '[10.113.10.44]:32222 ([10.113.10.44]:32222)' can't be established.
RSA key fingerprint is SHA256:3ug+6dW5AfCvMH0yfA48DBB0YdW+f37cYe5hvTORleM.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '[10.113.10.44]:32222' (RSA) to the list of known hosts.
Password authentication
Password:
junos-vmhost-install-mx-x86-64-18.3R1.9.tgz 100% 3014MB 39.1MB/s 01:17

(master)
anuta@Services-PE-OC1-Lab-RE1> file list /var/tmp
/var/tmp:
10-15-19_Services-PE-OC1-Lab-RE0-logs.tgz
10-15-19_Services-PE-OC1-Lab-RE1-logs.tgz
```

```

(master)
anuta@Services-PE-0C1-Lab-RE1> file list re0:/var/tmp
re0:
-----
/var/tmp:
LOCK FILE
appidd_cust_app_trace
appidd_trace_debug
etc/
juniper.conf.sync.gz
junos-install-mx-x86-64-18.4R1.8.tgz*
junos-install-mx-x86-64-20.1R1-S1.2.tgz*
junos-vmhost-install-mx-x86-64-18.3R1.9.tgz
krt_rpf_filter.txt
mmcq_mmdb_rep_mmcq
mmcq_sdb_bbe_mmcq
netproxy
package.log
pc /
pfs_debug_commands
pics/
pkg_cleanup.log
pkg_cleanup.log.err
pkg_cleanup.log.old
ppe_trap_fpc1_LU_0_00.0
ppe_trap_fpc1_LU_0_00.1
ppe_trap_fpc1_LU_1_00.0
ppe_trap_fpc1_LU_1_00.1
preinstall_boot_loader.conf
rtsdb/
sampled.pkts
sd-upgrade/
sec-download/
ttrace_fpc1_LU_0_00.0
ttrace_fpc1_LU_0_00.1
ttrace_fpc1_LU_0_01.0
ttrace_fpc1_LU_0_01.1
ttrace_fpc1_LU_0_02.0
ttrace_fpc1_LU_0_02.1
ttrace_fpc1_LU_0_03.0
ttrace_fpc1_LU_0_03.1
ttrace_fpc1_LU_1_00.0
ttrace_fpc1_LU_1_00.1
ttrace_fpc1_LU_1_01.0
ttrace_fpc1_LU_1_01.1
ttrace_fpc1_LU_1_02.0
ttrace_fpc1_LU_1_02.1
ttrace_fpc1_LU_1_03.0
ttrace_fpc1_LU_1_03.1
(master)
anuta@Services-PE-0C1-Lab-RE1>

```

Software Version Compliance

Software Version Compliance feature enables defining policies and enforcing software currency checks for network devices. The policies support defining N-1 (previous), N (current), N+1 (next) compliant versions for vendor and device models. Reports are generated for the software version checks. Summary reports by device model and vendor are available. The out of compliance devices can then be upgraded using Software Upgrade workflows available in ATOM.

Network Automation

ATOM provides stateful or Service and stateless (MOP) automation framework.

Stateless Low Code or MOP automation - Low Code Workflow automation enables network administrators to perform method-of-procedures involving different actions configuration, operations including show & exec commands on the device. Multiple actions can be stitched together to form a flow. Such flow is executed on one more device with appropriate user inputs.

Example:

1. Device Software Image Upgrade

2. Protocol Migration [IPV4 TO V6, OSPF to ISIS]
3. Hardware RMA/ Refresh [Moving from one vendor to another]

MOP Automation can be a combination of Stateless Action and Staful actions as well. In such scenarios MOP will contain stateless actions like pre-checks while performing API invocations against Device or Service Models to perform stateful transactional action.

Such tasks have no requirement for statefulness and can be best developed using Workflow Automation.

Example:

1. Application Deployment in Data Center with Pre-checks and Post-Checks
2. Branch Config Deployment with Pre-Checks & Post-Checks

Stateful Service Automation - ATOM Service automation helps administrators develop stateful and atomic transactions. Admins can create service models that enable Create, Update and Delete operations (CRUD). Such operations can be carried out throughout the life of the service. Brownfield service discovery is also supported.

Example:

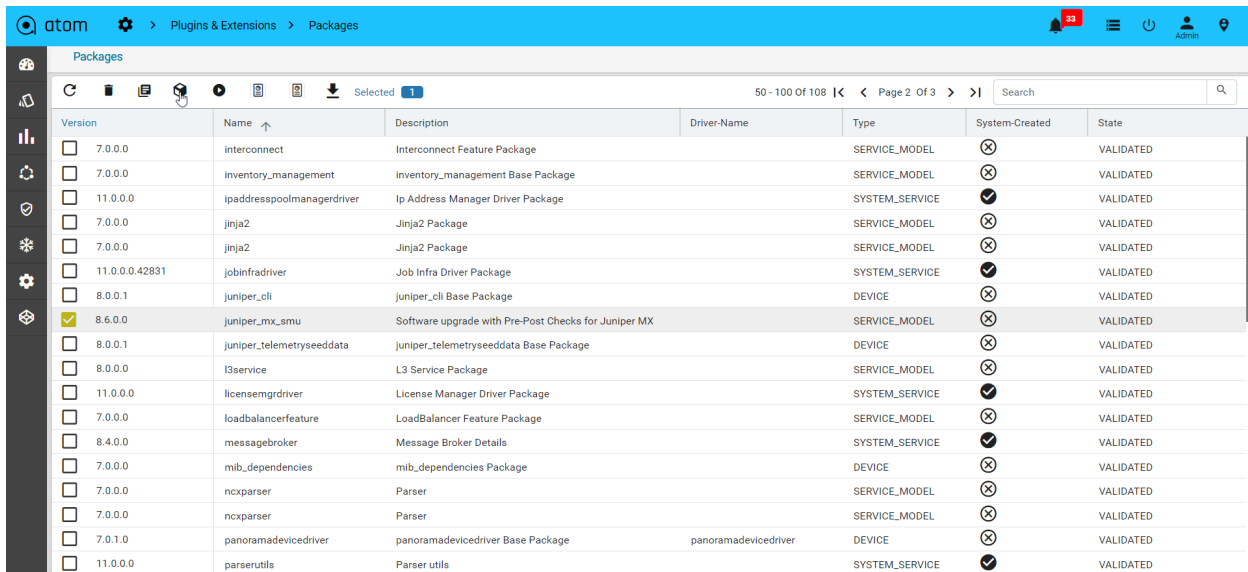
1. Application Deployment in Data Center
2. Layer-3 VPN
3. Layer-2 VPN
4. Private Cloud to Public Cloud Interconnect

Network Workflow & Low Code Automation

Workflow breaks down an activity into subtasks and ties them together with network events, provisioning actions, show-commands, pre-checks, post-checks, user forms and approvals, timed background tasks, inventory checks etc.

Workflow Automation offers an intuitive graphical designer to automate network provisioning and maintenance activities.

Administrators can create simple or complex flows using ATOM Workflow's drag and drop interface. ATOM Workflow has prebuilt adaptors to enable integration with ticketing, billing, OSS, BSS and many other network elements. Workflow can also automate multi-level approval sequences. Use workflows for a one time project or for repetitive tasks. Workflow development is covered in "Workflow Modelling" section in the ATOM Platform Guide guide. For automation of tasks that require stateful and atomic transactions it is advised to use ATOM Service Models discussed in ATOM Platform Guide.



Version	Name	Description	Driver-Name	Type	System-Created	State
7.0.0.0	interconnect	Interconnect Feature Package		SERVICE_MODEL	⊗	VALIDATED
7.0.0.0	inventory_management	Inventory Management Base Package		SERVICE_MODEL	⊗	VALIDATED
11.0.0.0	ipaddresspoolmanagerdriver	Ip Address Manager Driver Package		SYSTEM_SERVICE	✔	VALIDATED
7.0.0.0	jinja2	Jinja2 Package		SERVICE_MODEL	⊗	VALIDATED
7.0.0.0	jinja2	Jinja2 Package		SERVICE_MODEL	⊗	VALIDATED
11.0.0.0.42831	jobinfradriver	Job Infra Driver Package		SYSTEM_SERVICE	✔	VALIDATED
8.0.0.1	juniper_cli	Juniper CLI Base Package		DEVICE	⊗	VALIDATED
8.6.0.0	juniper_mx_smu	Software upgrade with Pre-Post Checks for Juniper MX		SERVICE_MODEL	✔	VALIDATED
8.0.0.1	juniper_telemetryseeddata	Juniper Telemetryseeddata Base Package		DEVICE	⊗	VALIDATED
8.0.0.0	l3service	L3 Service Package		SERVICE_MODEL	⊗	VALIDATED
11.0.0.0	licensemgrdriver	License Manager Driver Package		SYSTEM_SERVICE	✔	VALIDATED
7.0.0.0	loadbalancerfeature	LoadBalancer Feature Package		SERVICE_MODEL	⊗	VALIDATED
8.4.0.0	messagebroker	Message Broker Details		SYSTEM_SERVICE	✔	VALIDATED
7.0.0.0	mib_dependencies	mib_dependencies Package		DEVICE	⊗	VALIDATED
7.0.0.0	ncxparser	Parser		SERVICE_MODEL	⊗	VALIDATED
7.0.0.0	ncxparser	Parser		SERVICE_MODEL	⊗	VALIDATED
7.0.1.0	panoramadevicedriver	Panorama Device Driver Base Package	panoramadevicedriver	DEVICE	⊗	VALIDATED
11.0.0.0	parserutils	Parser utils		SYSTEM_SERVICE	✔	VALIDATED

Workflow Lifecycle Management

A workflow definition defines the structure of a workflow. A workflow instance is an individual execution of a workflow definition. The relation of the workflow instance to the workflow definition is the same as the relation between Object and Class in Object Oriented Programming. The workflow engine is responsible for creating workflow instances and managing their state.

Workflow Instances traverse different states as they progress from the start to end. The various states are as listed below:

- **Active** : Once the workflow is started it gets into an active state. Through-out the different tasks, workflow continues to be in an active state and indicates an error free execution.
- **Error State** : If there are unhandled exceptions in the scripts and programmatic/syntactic errors in inline scripts the workflow execution goes to an error state.
- **Internally Terminated** : If there are any errors in communication with the device or any custom RPCs throw exceptions which don't have explicit error handling defined in the workflow they are internally terminated by ATOM and state is updated accordingly.
- **Externally Terminated**: If the Network Administrator finds any unexpected behavior during any point in the workflow execution he has an option of manually terminating the workflow instance. This is the only state which the end user can manually state to terminate the flow.
- **Completed**: Once the workflow is terminated and has reached the stop event, the workflow goes to a completed stage and indicates a successful positive flow execution.

The screenshot displays the ATOM Workflows interface. At the top, there's a navigation bar with 'atom' and 'Workflows'. Below it, a workflow diagram for 'Software_Maintenance_Upgrade_Juniper_MX:1:1907' is shown. The diagram consists of a sequence of tasks connected by arrows, with some tasks having sub-processes. Below the diagram, there's an 'Advanced Search' bar with a 'Search' button. Underneath, a table titled 'Instances' shows 10 of 10 instances. The table has columns for 'Workflow Instance Name', 'Workflow Id', 'Id', 'State', and 'Start Time'.

Workflow Instance Name	Workflow Id	Id	State	Start Time
<input type="checkbox"/> wqewq	Software_Maintenance_Upgrade_Juniper_MX:1:1907	211887	COMPLETED	06/23/21, 4:58:11 PM
<input type="checkbox"/> wqeqwe	Software_Maintenance_Upgrade_Juniper_MX:1:1907	106665	INTERNALLY_TERMINATED	06/21/21, 1:18:25 PM
<input type="checkbox"/> wqee	Software_Maintenance_Upgrade_Juniper_MX:1:1907	16364	INTERNALLY_TERMINATED	06/17/21, 2:56:53 PM
<input type="checkbox"/> wqe	Software_Maintenance_Upgrade_Juniper_MX:1:1907	24425	INTERNALLY_TERMINATED	06/17/21, 5:23:10 PM
<input type="checkbox"/> qwerty	Software_Maintenance_Upgrade_Juniper_MX:1:1907	13657	INTERNALLY_TERMINATED	06/17/21, 11:29:17 AM
<input type="checkbox"/> qwer	Software_Maintenance_Upgrade_Juniper_MX:1:1907	13499	ACTIVE	06/17/21, 11:13:30 AM

Start Workflows :

To start a workflow instance follow the steps below.

1. Navigate to **Automation > Workflows > Workflows**
2. Select the workflow package from the list
3. Click on **Start** to start an instance of the workflow and provide valid Instance Name

The screenshot displays the ATOM Workflows interface with a list of workflow packages. The interface includes a navigation bar with 'atom' and 'Workflows'. Below it, there's a 'Summary' tab and a 'Filter Group' dropdown. A table lists various workflow packages with columns for 'Name', 'Resource', 'Description', 'Key', and 'Package Name An'.


Name	Resource	Description	Key	Package Name An
<input type="checkbox"/> image_success	ZTP		image_success	ztp.11.0.0.5
<input type="checkbox"/> Interface Rate Limit SNMP	Interface_ratelimit_SNMP	MAIN : Configure rate-limit on interface by SNMP	Interface_Rate_Limit_SNMP	cla_ratelimit_remu
<input type="checkbox"/> Interface_Shutdown_with_Approval_MDT	Interface_Shutdown_with_Approval_MDT	Interface Shutdown With Approval MDT	Interface_Shutdown_with_Approval_MDT	cla_workflow_rerr
<input type="checkbox"/> Interface_Shutdown_with_Approval_OC	Interface_Shutdown_with_Approval_OC	Interface Shutdown With Approval OC	Interface_Shutdown_with_Approval_OC	cla_workflow_rerr
<input type="checkbox"/> Interface_Shutdown_with_Approval_SNMP	Interface_Shutdown_with_Approval_SNMP	Interface Shutdown with Approval SNMP	Interface_Shutdown_with_Approval_SNMP	cla_workflow_rerr
<input type="checkbox"/> Internal Atom Package Wf		Internal Atom Package For Bpmn Deployment during package I...	InternalAtomPackageWf	
<input type="checkbox"/> Inventory Management	Inventory_mngmt	Inventory Management and Status of Juniper Devices	Inventory_mngmt	Inventory_managi
<input type="checkbox"/> iosxr	ZTP		iosxr	ztp.11.0.0.5
<input type="checkbox"/> iPX	ZTP		iPX	ztp.11.0.0.5
<input type="checkbox"/> Jira Ticket	jira	MAIN : Open Incident in JIRA	JiraTicket	cla_workflow_rerr
<input type="checkbox"/> Juniper MX DiskSpace Checks	diskspaceCheck_juniper_mx	Subprocess : Disk Space Checks for Juniper MX	Juniper_MX_Diskspace_Checks	juniper_mx_smu.8
<input type="checkbox"/> Juniper MX Pre Post Checks	prepostchecks_juniper_mx	Subprocess : Pre-Post Checks for Juniper MX	Juniper_MX_Pre_Post_Checks	juniper_mx_smu.8
<input checked="" type="checkbox"/> Juniper MX Upgrade	juniper_mx_smu	MAIN : Software upgrade with Pre-Post Checks for Juniper MX	Software_Maintenance_Upgrade_Juniper_MX	juniper_mx_smu.8
<input type="checkbox"/> L3Service_CLI	L3servicecli	MAIN : L3 service commands are directly pushed on cisco platf...	L3Service_CLI	L3servicecli.8.0.0.6

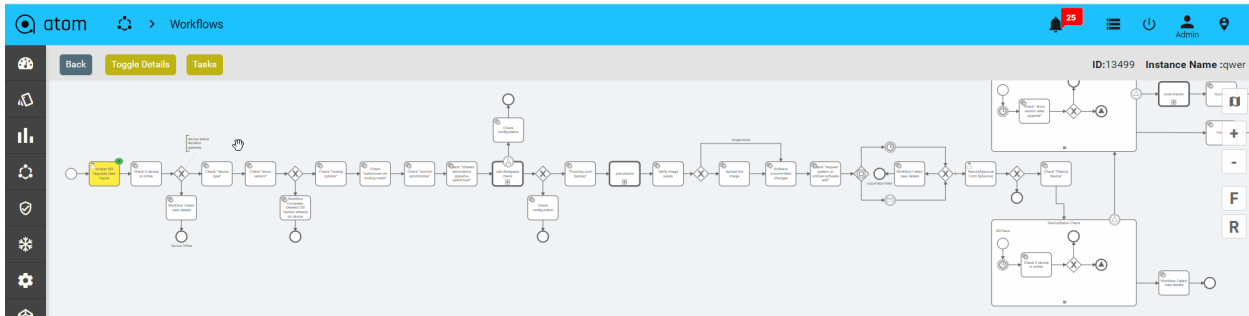
1. Navigate to **Workflows > Instances**
2. Click on **Inspect**

This opens a window with the workflow elements. Green indicates successfully completed tasks. Yellow indicates the current task being executed

Instances

1 Of 1

<input type="checkbox"/>	Workflow Instance Name	Workflow Id	Id	State	Start Time 
<input type="checkbox"/>	smu	Software_Maintenance_Upgrade_Juniper_MX:1:1907	401172	ACTIVE	06/25/21, 11:43:11 AM

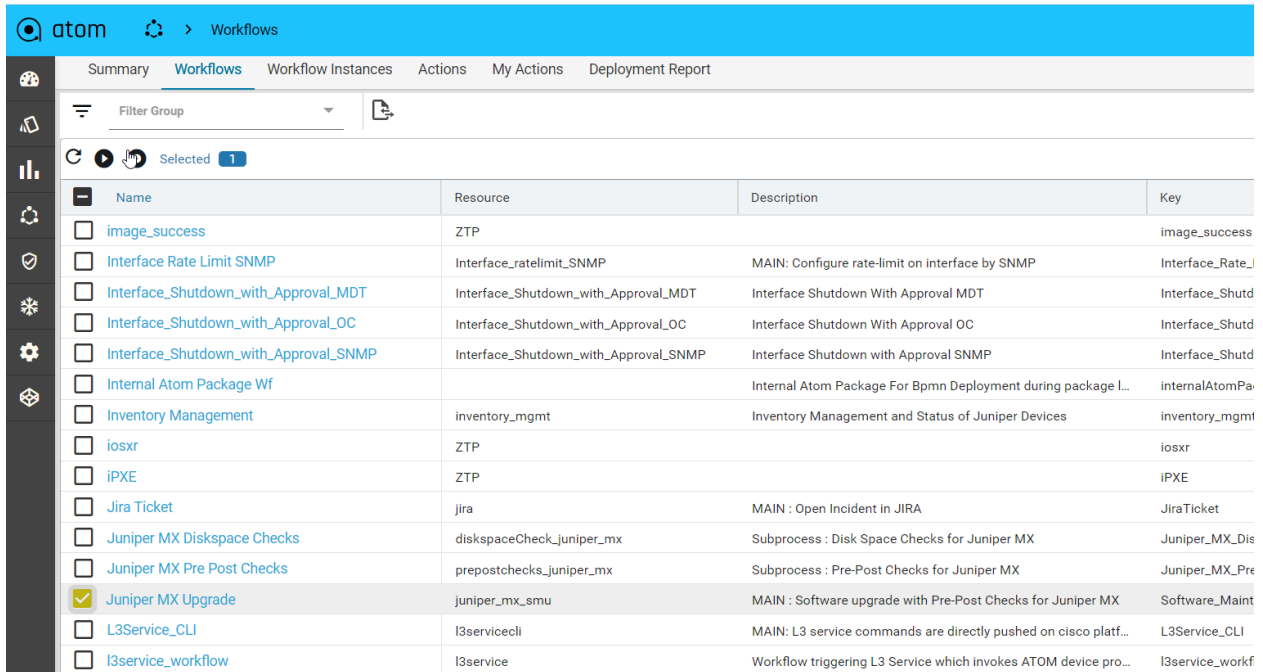


Suspend/Pause Workflows:

In the workflow definition view and in the workflow instance view, can suspend the selected workflow definition or workflow instance by using the suspend button on the panel.

Workflow Definition Suspension

If you suspend the workflow definition, you prevent the workflow definition from being instantiated. No further operations can be done while the workflow definition is in the suspended state. You can simply re-activate the workflow definition .



Name	Resource	Description	Key
<input type="checkbox"/> image_success	ZTP		image_success
<input type="checkbox"/> Interface Rate Limit SNMP	Interface_ratelimit_SNMP	MAIN: Configure rate-limit on interface by SNMP	Interface_Rate_Limit_SNMP
<input type="checkbox"/> Interface_Shutdown_with_Approval_MDT	Interface_Shutdown_with_Approval_MDT	Interface Shutdown With Approval MDT	Interface_Shutdown_with_Approval_MDT
<input type="checkbox"/> Interface_Shutdown_with_Approval_OC	Interface_Shutdown_with_Approval_OC	Interface Shutdown With Approval OC	Interface_Shutdown_with_Approval_OC
<input type="checkbox"/> Interface_Shutdown_with_Approval_SNMP	Interface_Shutdown_with_Approval_SNMP	Interface Shutdown with Approval SNMP	Interface_Shutdown_with_Approval_SNMP
<input type="checkbox"/> Internal Atom Package Wf		Internal Atom Package For Bpmn Deployment during package l...	internalAtomPackage
<input type="checkbox"/> Inventory Management	inventory_mgmt	Inventory Management and Status of Juniper Devices	inventory_mgmt
<input type="checkbox"/> iosxr	ZTP		iosxr
<input type="checkbox"/> iPX	ZTP		IPXE
<input type="checkbox"/> Jira Ticket	Jira	MAIN : Open Incident In JIRA	JiraTicket
<input type="checkbox"/> Juniper MX DiskSpace Checks	diskspaceCheck_juniper_mx	Subprocess : Disk Space Checks for Juniper MX	Juniper_MX_DiskSpace_Checks
<input type="checkbox"/> Juniper MX Pre Post Checks	prepostchecks_juniper_mx	Subprocess : Pre-Post Checks for Juniper MX	Juniper_MX_Pre_Post_Checks
<input checked="" type="checkbox"/> Juniper MX Upgrade	juniper_mx_smu	MAIN : Software upgrade with Pre-Post Checks for Juniper MX	Software_Maintenance_Juniper_MX_Upgrade
<input type="checkbox"/> L3Service_CLI	l3servicecli	MAIN: L3 service commands are directly pushed on cisco platf...	L3Service_CLI
<input type="checkbox"/> l3service_workflow	l3service	Workflow triggering L3 Service which invokes ATOM device pro...	l3service_workflow

Workflow Instance Suspension:

If you suspend the workflow instance, you can prevent the workflow instance from being executed any further. This includes suspending all tasks included in the process instance. You can re-activate the process instance at any later point of time.

The screenshot shows the ATOM Workflows interface. At the top, there's a navigation bar with 'atom' and 'Workflows'. Below it, a sidebar contains various icons. The main area displays a workflow diagram for 'Juniper_MX_Upgrade:1:15043'. Below the diagram, there's a section for 'Instances' with a filter group and a table of workflow instances.

Workflow Instance Name	ID	Workflow Id	Business State	Workflow State	Duration	SLA Time (S)	SLA Compliant	Progress(%)
start	354216	Juniper_MX_Upgrade:1:15043	IN_PROGRESS	ACTIVE	5m 4s			
wqwq	301425	Juniper_MX_Upgrade:1:15043	COMPLETE	COMPLETED	37m 24s			Complete
wqe	103572	Juniper_MX_Upgrade:1:15043	COMPLETE	COMPLETED	21m 6s			Complete
wqe	102712	Juniper_MX_Upgrade:1:15043	COMPLETE	COMPLETED	20m 12s			Complete

Workflow Instance Error:

Unresolved programmatic/syntactic errors of a process instance or a sub process instance are indicated by Atom workflow engine as errors. The Errors tab in the workflow instance view lists the failed activities with additional information.

The screenshot shows the ATOM Workflows interface with the 'Errors' tab selected. It displays a workflow diagram for 'ID:402265 Instance Name :smu1'. Below the diagram, there's a section for 'Errors' with a table of error messages.

Message	Timestamp	Activity	Cause Process Incident ID	Root Cause Process Incident ID
CallActivity_01kcf4o	2021-06-25T07:14:44.253+0000	CallActivity_01kcf4o	402786	402786

Retry a Failed Job

To resolve an error you can use the **Retry** button on the top panel. Select the corresponding instance, so the atom-engine will re-trigger this job and increment its retry value in the database.

Variables

Actions

Errors

Selected 1

1 Of 1

Enter a keyword

<div><div></div></div> Message	Timestamp	Activity	Cause Process Incident ID	Root Cause Process Incident ID
<div><div></div></div>	2021-06-25T07:14:44.253+0000	CallActivity_01kcf4o	402786	402786

Workflow Variables

Workflow Instance Variables can be used to add data to workflow runtime state. Various API methods/Service Tasks that change the state of these entities allow updating of the attached variables. In general, a variable consists of a name and a value. The name is used for identification across workflow constructs. For example, if one activity sets a variable named var, a follow-up activity can access it by using this name. The value of a variable is the value held by that particular named variable in the Atom engine for that particular workflow instance context.

To view the workflow variables

1. Select the particular workflow instance that is active.

The screenshot shows the ATOM Workflows interface. The top panel displays a complex workflow diagram with various activities and connectors. The bottom panel shows a table of workflow instances with the following data:

Workflow Instance Name	ID	Workflow Id	Business State	Workflow State	Duration	SLA Time (s)	SLA Compliant	Progress(%)
qwdsa	355536	Juniper_MX_Upgrade:1:15043	IN_PROGRESS	ACTIVE	42s			
wqe	355066	Juniper_MX_Upgrade:1:15043	ERROR	INTERNALLY_TERMINATED	4m 39s			Error
qweqe	354703	Juniper_MX_Upgrade:1:15043	CANCELLED	INTERNALLY_TERMINATED	4m			Cancelled
wqeqe	354589	Juniper_MX_Upgrade:1:15043	COMPLETE	COMPLETED	49s			Complete

2. View the workflow variables in the bottom panel.

Variables	Activities	Actions	Errors
26 Of 26	Search by Name		
Variable Name	Id	Value	Type
PERM_AUDIT_SESSION_ID	355537	fb12ff57-a4b0-4a62-8a0d-231d52b7ca12	String
atom_user_id	355538	GVpBgCazlxKc9BAsVmThDZLAAAI	String
businessKey	355539	qwdsa	String
atom_user_name	355540	admin	String
atom_user_owner	355541	system	String
request_url	355542	process-definition/start	String
span.context	355545	{ "uber-trace-id": "e603e40fd46028e:45a58b614a79756..."	Object

3. Users can also edit the variable values during runtime.

The screenshot shows the ATOM interface with a workflow diagram in the background. A modal dialog titled "Edit Instance Variable" is open. It contains the following fields:

- Variable Name:** span.context
- Variable Type:** Object
- Variable Value:** { "uber-trace-id": "a1684d8c5b1ea418:86510b448c077038:a1684d8c5b1ea418:1" }

The background shows a workflow diagram with various tasks and a variables table at the bottom. The table lists variables like PERM_AUDIT_SESSION_ID, atom_user_id, businessKey, atom_user_name, atom_user_owner, request_url, and span.context.

4. Alternatively User can compare two variable values and see the difference on the screen.

The screenshot shows the ATOM interface with a workflow diagram in the background. A modal dialog titled "Comparing | Variable Values" is open. It displays the values of two variables side-by-side for comparison:

- span.context:** { "uber-trace-id": "a1684d8c5b1ea418:86510b448c077038:a1684d8c5b1ea418:1" }
- request_url:** process-definition/start

The background shows a workflow diagram with various tasks and a variables table at the bottom. The table lists variables like businessKey, atom_user_name, atom_user_owner, request_url, span.context, and TaskID.

User Inputs:

Some workflows may require the administrator to enter some values at particular stages.

Workflow execution will be stalled until the values are entered.

To view if any Action items are pending against a particular workflow instance we can view it under the specific workflow instance view :

For viewing such tasks:

1. Select the particular workflow instance that is active.

The screenshot displays the ATOM Workflows interface. At the top, there's a navigation bar with 'atom' and 'Workflows'. Below it, a status bar shows '5 Completed', '5 Internally Terminated', and '1 Active'. The main area features a complex workflow diagram for 'Juniper_MX_Upgrade:1:15043'. Below the diagram, there's a tabbed interface with 'Instances' and 'Activities'. The 'Instances' tab is active, showing a table of workflow instances.

Workflow Instance Name	ID	Workflow Id	Business State	Workflow State	Duration	SLA Time (S)	SLA Compliant	Progress(%)
start1	355671	Juniper_MX_Upgrade:1:15043	IN_PROGRESS	ACTIVE	1m 10s			
qwdsd	355536	Juniper_MX_Upgrade:1:15043	ERROR	INTERNALLY_TERMINATED	1m 42s			Error
wqe	355066	Juniper_MX_Upgrade:1:15043	ERROR	INTERNALLY_TERMINATED	4m 39s			Error
qwede	354703	Juniper_MX_Upgrade:1:15043	CANCELLED	INTERNALLY_TERMINATED	4m			Cancelled
wqege	354589	Juniper_MX_Upgrade:1:15043	COMPLETE	COMPLETED	49s			Complete
qwede	354506	Juniper_MX_Upgrade:1:15043	COMPLETE	COMPLETED	26s			Complete

2. Once Selected navigate to the action tabs to view all pending action items against this particular workflow instance.

The screenshot shows the ATOM Workflows interface. At the top, there's a navigation bar with 'atom' and 'Workflows'. Below it, a sidebar contains various icons. The main area displays a workflow diagram with multiple steps and decision points. On the right, a panel shows details for a specific workflow instance (Automation: 00:00:00.236, Instance ID: 355671, Instance Name: start1). Below the diagram, there's a tabbed interface with 'Variables', 'Activities', 'Actions', and 'Errors'. The 'Actions' tab is active, showing a list of workflow instances with columns: Workflow Instance Id, Workflow Instance Name, Name, Id, Workflow Id, Start Time, End Time, and Duration. The table shows one instance: 355671, start1, Juniper MX Upgrade User Inputs, 355686, Juniper_MX_Upgrade:1:15043, 08/31/21, 4:11:05 PM.

For completing such tasks

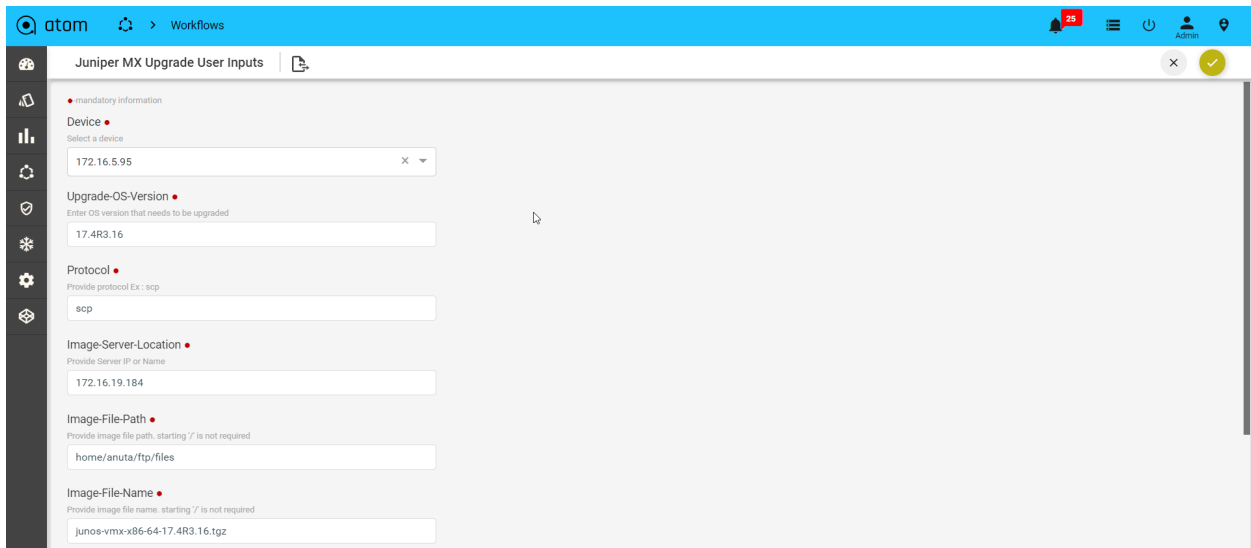
1. Navigate to **Workflows > Actions**
2. Select the workflow task and click on **Claim** to claim the task

The screenshot shows the ATOM Workflows interface with the 'Actions' tab selected. The top navigation bar includes 'Summary', 'Workflows', 'Workflow Instances', 'Actions', 'My Actions', and 'Deployment Report'. Below the navigation bar, there's a filter group showing 1 Unclaimed, 0 Assigned, 14 Completed, and 0 Cancelled. A table lists workflow instances with columns: Workflow Instance Id, Workflow Instance Name, Name, Id, Workflow Id, and Start Time. The table shows several instances, with the one having Id 355671 and Name 'Juniper MX Upgrade User Inputs' selected.

3. Navigate to **Workflow > My Actions**
4. Select the task claimed at step 2 and click on **Complete**

The screenshot shows the ATOM Workflows interface with the 'My Actions' tab selected. The top navigation bar includes 'Summary', 'Workflows', 'Workflow Instances', 'Actions', 'My Actions', and 'Deployment Report'. Below the navigation bar, there's a filter group showing 1 Selected. A table lists workflow instances with columns: Name, Workflow Instance Name, Id, Workflow Instance Id, Workflow Id, and Start Time. The table shows one instance: Juniper MX Upgrade User Inputs, start1, 355686, 355671, Juniper_MX_Upgrade:1:15043, 08/31/.

5. Enter values and click on the tick mark



The screenshot shows the ATOM Workflows interface. The top bar is blue with the ATOM logo and 'Workflows' text. Below it, a sidebar contains various icons. The main area displays a form titled 'Juniper MX Upgrade User Inputs'. The form includes several fields with red error indicators:

- Device**: Select a device. Value: 172.16.5.95.
- Upgrade-OS-Version**: Enter OS version that needs to be upgraded. Value: 17.4R3.16.
- Protocol**: Provide protocol Ex: scp. Value: scp.
- Image-Server-Location**: Provide Server IP or Name. Value: 172.16.19.184.
- Image-File-Path**: Provide image file path, starting '/' is not required. Value: home/anuta/ftp/files.
- Image-File-Name**: Provide image file name, starting '/' is not required. Value: junos-vmx-x86-64-17.4R3.16.tgz.

Network Service Automation

Stateful services are developed using ATOM SDK and involve Service model developed in YANG and optional business logic in Python. Such services have a continuous life cycle and undergo multiple changes over a period.

Services can be deployed in two modes -

Greenfield Mode - A user can instantiate the service, a set of network configurations, using the service template. These service templates are rendered from schema files that have been developed as a part of the Service package. ATOM automatically generates and applies relevant configurations on to the devices.

Brownfield Mode - ATOM automatically discovers services running on the device and maps it to the service template. For detailed information about service packages and how to write your own service models, and usage of “maps-to” extension, refer to the “*ATOM Platform Guide*”.

Ordering Greenfield Services in ATOM:

To order a service, that was modelled earlier, do the following:

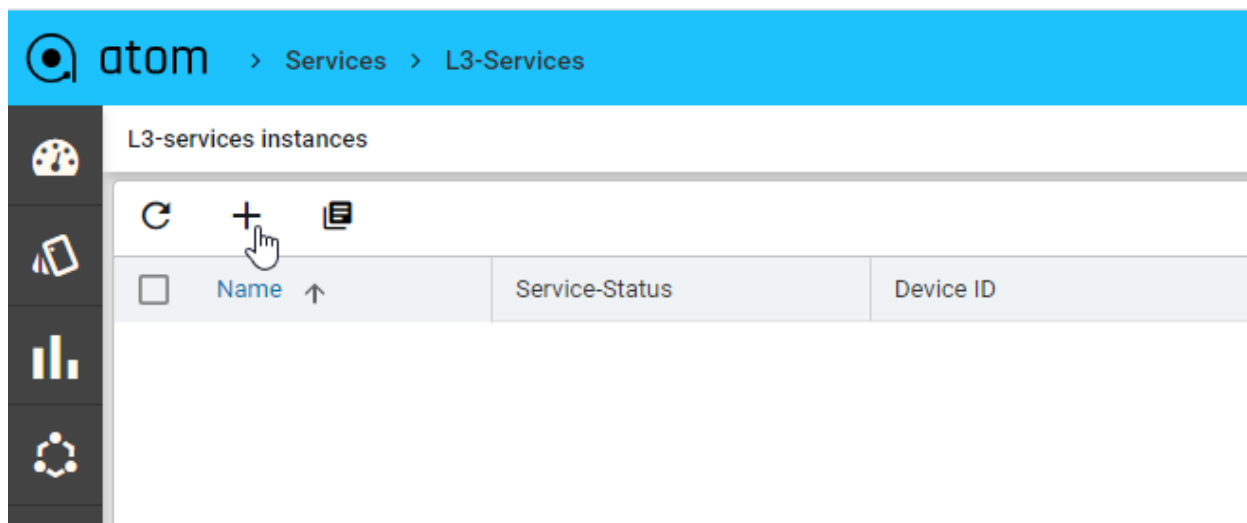
1. Navigate to **Automation > Services**
2. In the right pane, click Add
3. In the ensuing form, enter values for the fields that are displayed.
4. Click **OK**

ATOM automatically generates relevant network configurations.

Note: If “Dry Run” is enabled in the Administration tab, the generated configurations will not be applied to the devices.

Let us take an example of creating an instance of the “L3 Services” in ATOM. The schematic representation of the service is defined in the .yang file (in this case, *l3service.yang* file). This file is contained in the model folder of the corresponding service package (l3 service package) uploaded as a plugin to ATOM.

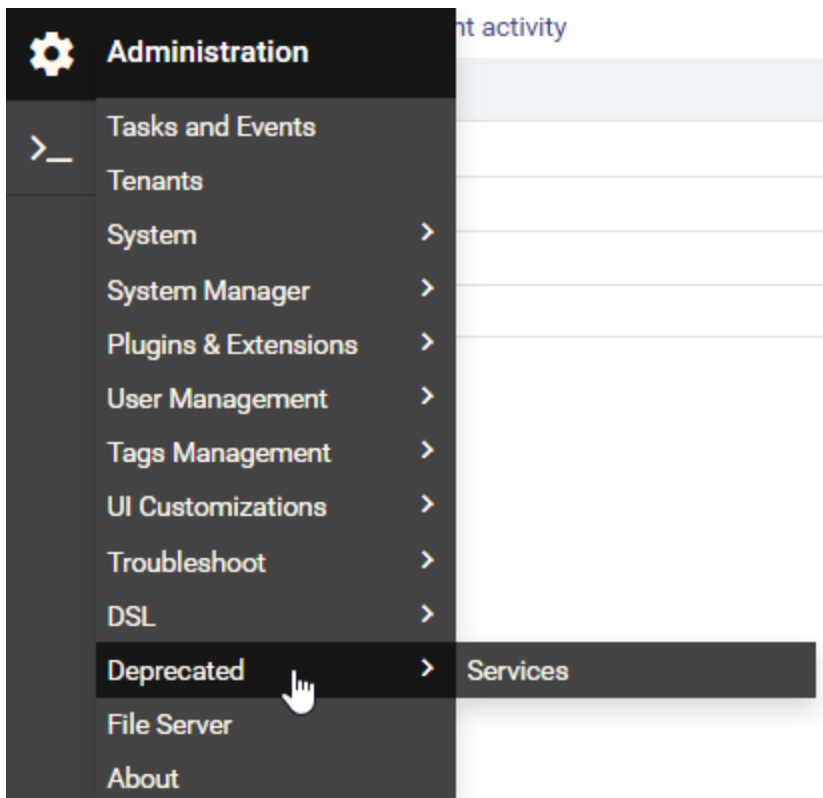
1. Navigate to **Administration > Plugins & Extensions**
2. Navigate to **Automation > Services > l3-services** and click on **Add**



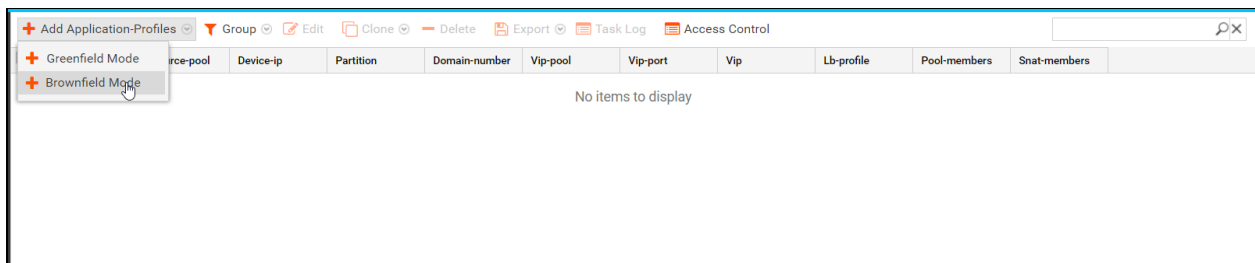
Deploying BSD services in ATOM:

Let us take an example of the deploying the “Application Profiles” service in Brownfield deployment mode:

1. Obtain the appropriate service package from Anuta Networks
2. Upload the service package into ATOM.
3. Navigate to **Administration > Deprecated > Services** to view the uploaded service package.



4. In the **Add Application Profiles** pane, select the **Brownfield Mode** as shown below:



5. In the **Create Application Services** form, all the values discovered from the device are populated in the parameters shown below:

Enter values in the fields that have been marked mandatory.

Create application-profiles

Name* ⓘ appservice1

Resource-Pool* ⓘ RP

Device-IP* ⓘ 172.16.3.37

Partition* ⓘ

Domain-Number* ⓘ

Vip-Pool* ⓘ

Vip

Vip-Port* ⓘ

+ Add + Add All Edit Delete Delete All Up Down

pool-members ⓘ

No Records Found

Transaction Policies OK Cancel Templates

Note: The borders of the fields that contain the auto discovered values are coloured in brown color.

6. Click **OK** after selecting the requisite values in each of the fields.

The commands that are generated in ATOM are not pushed to the device because of the mode of Brownfield deployment.

Transactional control at the Service level

For every service, the admin can control whether the corresponding configurations, generated by ATOM, should be pushed to the device. This gives an admin a granular level of control wherein some services can be sent to the device and a few can be retained on ATOM.

1. Navigate to **Automation > Services**
2. Click the service that you want to configure the transaction policies.
3. You can either enter the values of the fields or import the values from a template to fill the form.
4. In the Create service template, click **Transaction Policies > Transaction Policy Configs** screen to set the control at the transaction level as shown:

Transaction Policy Configs

Do not send commands to devices
Flag to control if commands can be sent to the device. 'true' ...

Fail Fast
Flag to control if reference validation to be done immediately...

Validation Scope
flag to control the data validation scope, this is similar transa...

Command Sequence

Auto Rollback

Clonable Device
Flag to control if commands can be sent to the device. 'true' ...

COMMITTED_DATA

DEPENDENCY_BASED

Save

Option	Type	Description
do-not- send-commands- to -devices	boolean	<p>Controls whether commands can be sent to the device. devices</p> <p>Select this option to commit the data to ATOM datastore, but no configuration changes will be applied on the device. Useful for testing or in the case of a brown-field environment to create services.</p> <p>Note: The value set for this option at the transaction policy overrides the value at the global level (in the General Settings)</p>
fail-fast	boolean	<p>Controls whether the reference validation should be done immediately.</p> <p>False: Defers the validation to after 'commit-task' state of the transaction</p>
validation-scope-type	enum	Controls whether data validation scope is across transactions. This flag is similar to

<ul style="list-style-type: none"> • COMMITTED_DATA • UNCOMMITTED_DATA 		<p>isolation control in traditional RDBMS, but limited to just data validation. Allowed values are "COMMITTED_DATA" and "UNCOMMITTED_DATA".</p> <p>Validation will be done only using the committed data. Current transaction will not see changes done by other parallel transactions</p> <p>Data validation will be done using the uncommitted data. Current transaction will see changes done by other parallel transactions</p>
<ul style="list-style-type: none"> • Command-sequence-policy • DEPENDENCY_BASED • NONE 	enum	<p>Controls whether the generated commands need to be ordered according to the dependencies specified in the model.</p> <p>Generated commands will be re-ordered based on the dependencies specified in the data model.</p> <p>Generated commands reflect the order of the requests sent from the client, no re-ordering is done</p>

The values for following options can be cross-verified before creation of each service

- Fail Fast
- Validation Scope
- Command Sequence Policy

5. Click the task created for the created service to view the commands generated by ATOM in the Task Details.

In the Task details, click Commands to view the generated commands by ATOM. As the

commands should not be sent to the device, (if do-not-send-commands-to-device option is selected), the status of the commands is set to “TO_BE_PROVISIONED” as shown below:

The screenshot shows a window titled "Create: I3-service" with a close button (X) in the top right corner. Below the title bar, there is a timestamp range "21/05/2019, 18:24:39 - 21/05/2019, 18:24:41" and "Time Taken : 2 seconds". The "TASKID" is "Maxe9xtYtcTXaaajRHgbVDvA". There are two tabs: "Logs" and "Commands", with "Commands" being the active tab. The content area displays the following information:

```
Result: DEVICE: name = CSR3.31.Anuta.com ip-address = 172.16.3.31

Operation: CreateVrf
Status: TO_BE_PROVISIONED

vrf definition test
address-family ipv4
exit-address-family

Operation: UpdateInterface
Status: TO_BE_PROVISIONED

interface GigabitEthernet2
vrf forwarding test
ip address 10.10.40.19 255.255.255.0
no shutdown

RollbackCommands:
```

At the bottom right, there is a button labeled "Download as Config".

The generated commands can be downloaded and verified with the expected configurations for that service.

Cancelling an ordered Service:

1. Select the service and click **Delete**.
2. In the **Confirmation** window, before selecting the **Yes** button, click the **Transaction Policies**.
3. Select the option , “**Do-not-send-commands-to-devices**” in the policy

In the corresponding task generated, in the Task Details pane, click **Commands** to view the generated commands by ATOM. As the commands should not be sent to the device (if “do not send commands to the device” option selected in the transaction policy config), the status of the command is set to “TO_BE_PROVISIONED”.

IMPORTANT: If this option is not selected properly as per create behavior, the service deletion might fail.

Service Approvals

You can create policies for approving creation, deletion or updation of the service configurations on devices. In addition, you can add approvers who must approve the operations defined in the service approval policy. Apart from seeking approvals for services, you can set approvers for any of the operations for any entity in ATOM.

1. Navigate to **Automation -> Services-> Approvals**
2. In the right pane, click **Add Policy** to create the details as shown below:

The screenshot shows the 'Create Policy' form in the ATOM interface. The form is titled 'Create Policy' and has a sidebar on the left with icons for various functions. The main form area contains the following fields and options:

- Policy-Name:** A text input field with the value 'policy'.
- Service-Target:** A dropdown menu with the selected value '/controller/services/13service/13-services/13-service'.
- Provision-Approval-Needed:** A checkbox that is checked.
- Delete-Approval-Needed:** A checkbox that is checked.
- Update-Approval-Needed:** A checkbox that is checked.
- Policy-Type:** A dropdown menu with the selected value 'ANY_ONE'.

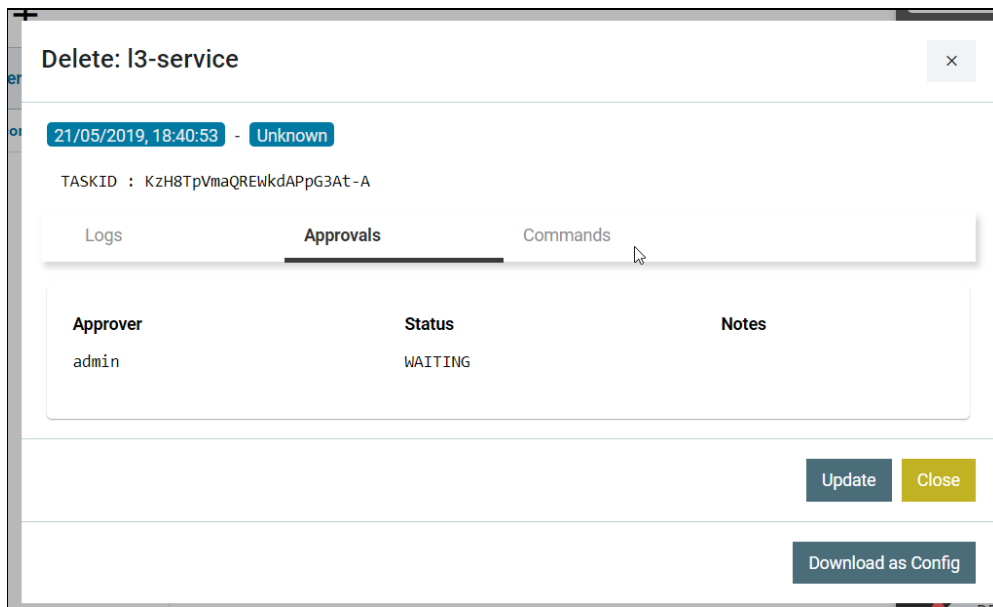
At the top right of the form, there is a 'Create policy' button with a green checkmark icon.

3. Navigate to the right pane to add details as described below:
 - i. **Policy Name:** Enter a name for the approval policy
 - ii. **Service Target:** Enter the path for the object in the data model tree for which approval is required.
For example: If the object of interest is a service, enter the path of the service.

/controller:services/l3service:l3-services, which means that the operation of interest on this managed-cpe -service will be sent to the approver or approvers for their perusal before being pushed to the device.

- iii. **Provision Approval Needed:** Check this option if the user selected as the approver should approve the configurations before they are pushed to the respective device or devices.
 - iv. **Delete Approval Needed:** Check this option if the approver should approve the configurations that are required for deletion of the service configurations from the device or devices.
 - v. **Update Approval Needed:** Check this option if the admin should approve the configurations that are required to update the service configurations on the device or devices.
 - vi. **Policy Type:** This option enables you to set if approvals are required from a single approver or multiple approvers.
- **ALL:** The task that is generated as a result of a service operation awaits the approval of all the approvers who have been added for that service .

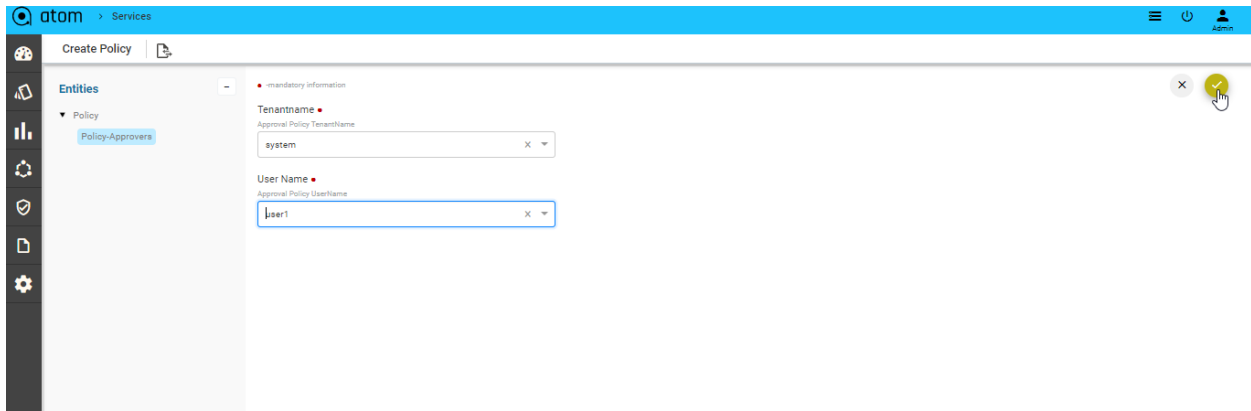
In the following example, the operation of creating a 'customer' needs approval of two approvers, 'admin' and 'User1'. The task is completed successfully after receiving the approval of all the approvers as shown below:



- **ANY_ONE:** The task generated as a result of a service operation awaits the approval of any of the multiple approvers added for that service.

NOTE: All changes made in the service approval policy will come into effect only for the subsequent service instantiations and will not affect the ongoing service operations.

4. Navigate to the left pane to add the tenant users who should approve the configurations generated by ATOM for any of the service operations (create, delete or update).



NOTE: Do not edit the name of the user (UserName) who has been added as an approver in the service policy.

Configuration Drift (Network Services)

Whenever there is a configuration change in the device that does not match with the generated configuration on ATOM, a reconciliation task is generated in ATOM. After viewing the config diff generated, the administrator can decide how to reconcile these config differences so that the device and ATOM are always in sync with respect to the configuration states.

- Navigate to **Automation->Services->Reconciliation** in the left pane
- In the right pane, click **Reconciliation > Entities** to view all the reconciliation entities that are generated at the device level and service level.

Device	Entity Name	Service	Operation	Detected Time
172.16.3.170	mpls		DELETE	2020-04-16 08:49:39.108
172.16.1.139	mode=mtsp		DELETE	2020-04-09 09:35:59.606
172.16.1.139	vlan=567		DELETE	2020-04-09 09:35:59.6
172.16.1.139	vlan=124	IS-service=mtsp3	DELETE	2020-04-09 09:35:59.594
172.16.1.139	vrf=mtsp1	IS-service=mtsp3	DELETE	2020-04-09 09:35:59.579
172.16.1.139	interface=02.124	IS-service=mtsp3	DELETE	2020-04-09 09:35:59.541
172.16.1.139	local-credential=admin1		DELETE	2020-04-09 09:35:59.52
172.16.18.176	subscription=Test056		DELETE	2020-04-09 09:35:22.365
172.16.3.40	vrf=networksanuta	IS-service=anuta	DELETE	2020-04-09 09:35:16.995
172.16.3.40	interface=ipip	IS-service=anuta	DELETE	2020-04-09 09:35:16.894
172.16.5.156	scheduler=Nil-Nep-pon-sched		DELETE	2020-04-09 09:28:46.549

- Double click the reconciliation entity of your choice, to view the Reconciliation details:

The screenshot shows the ATOM Reconciliation interface. The main table lists reconciliation entities with columns: Device, Entity Name, Service, Operation, and Detected Time. A right-hand pane titled 'Reconciliation-Details' shows the details for a selected device (172.16.3.45). It includes the entity name (interface=GigabitEthernet3), service (IS-service=test2), and configuration differences between ATOM and the device. The configuration to be reconciled is also shown.

Device	Entity Name	Service	Operation	Detected Time
172.16.3.45	interface=GigabitEthernet3	IS-service=test2	UPDATE	2020-04-30 06:05
172.16.3.45	vrf=vrf4	IS-service=test2	DELETE	2020-04-30 06:05
172.16.3.170	mpls		DELETE	2020-04-16 08:40
172.16.1.139	mode=matp		DELETE	2020-04-09 09:35
172.16.1.139	vlan=567		DELETE	2020-04-09 09:35
172.16.1.139	vlan=124	IS-service=test3	DELETE	2020-04-09 09:35
172.16.1.139	vrf=ast1	IS-service=test3	DELETE	2020-04-09 09:35
172.16.1.139	interface=22.124	IS-service=test3	DELETE	2020-04-09 09:35
172.16.1.139	local-credential=admin1		DELETE	2020-04-09 09:35
172.16.18.176	subscription=Test566		DELETE	2020-04-09 09:35
172.16.3.40	vrf=networksanuta	IS-service=anuta	DELETE	2020-04-09 09:35
172.16.3.40	interface=ipip	IS-service=anuta	DELETE	2020-04-09 09:35
172.16.5.156	scheduler=Ni-Net-port-sched		DELETE	2020-04-09 09:26

Reconciliation-Details

Device: 172.16.3.45

Entity Name: interface=GigabitEthernet3
Service: IS-service=test2

Configuration Difference

- 'description': 'MPLS ATT'
- 'ip-address': '83.0.0.13'
- 'ipv6-address': '26530c0:1:2:1'
- 'ipv6-prefix-length': '126'
- 'isfeatures.vrf': 'vrf4'
- 'netmask': '255.0.0.0'

Configuration to be reconciled

operation: UpdateInterface
commands:
interface GigabitEthernet3
ip address 26530c0:1:2:1/126
ipv6 enable
vrf forwarding vrf4
ip address 83.0.0.13 255.0.0.0
description MPLS ATT

The configuration difference between ATOM and the device is shown on the left pane where as the right pane displays the configurations that should be pushed to the device to reconcile with the state of ATOM.

- Click the **Reconciliation Policy** to create the policies for reconciling the config differences either with the state of ATOM or with that of the device.
 - **OVERWRITE SERVER** - The generated config diff is pushed to the database of ATOM to reconcile with the state of the device
 - **OVERWRITE DEVICE** - The generated config diff is pushed to the device to reconcile with the state of ATOM.
 - **WAIT FOR APPROVAL** - Select this option if the generated reconciliation entities require a review by an administrator. The generated config diff is sent to an approver who can take the decision of either pushing the configurations to the device or overwriting the ATOM database.

atom > Services > L3-Services

Create L3-Service

mandatory information

Name
atmg

Device ID
csr342

Interface Mode
Sub-interface

Description
atmg

Vrf
vrf

IP Address
Must be a valid IP Address. Ex: 172.16.1.24.
172.16.1.24

Netmask
Must be a valid IP Address. Ex: 172.16.1.24.
255.255.255.0

IPv6-Address
Must be a valid IPv6 Address.
2001:db8::1

IPv6-Prefix-Length
Must be a valid IPv6 Prefix Length.
64

2. Login to the device console and delete the “test VRF” from the device

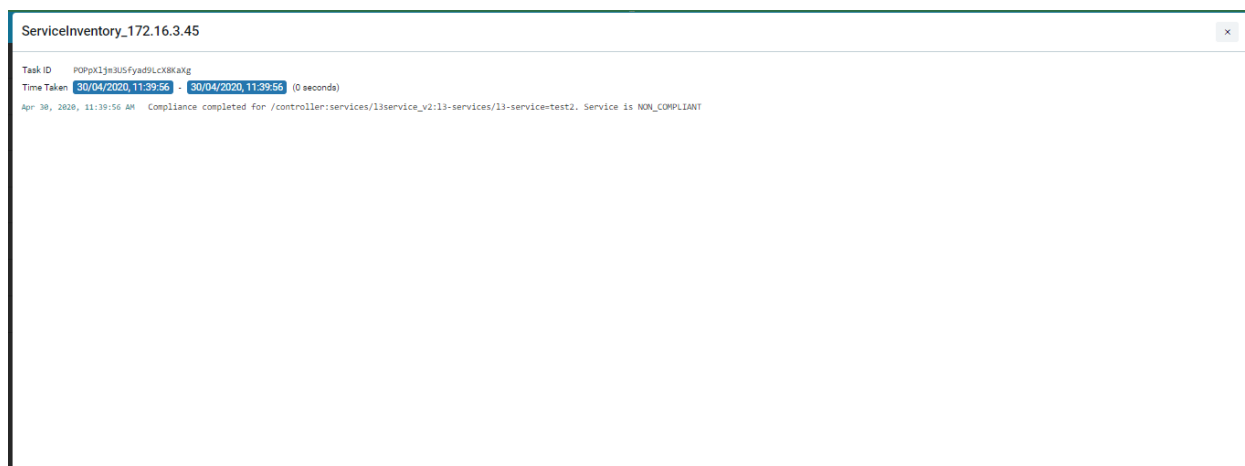
```

172.16.3.42 - PuTTY
login as: admin
executing bUsing keyboard-interactive authentication.
Password:
exe executing
csr342#sh vrf | inc test
test                <not set>
test1               <not set>          ipv4
test100             <not set>
test1000            <not set>
test125             <not set>          ipv4
test56              <not set>          ipv4      Gi1.876
test90              <not set>
test_vrf            <not set>          ipv4      Gi4
csr342#config term
Enter configuration commands, one per line.  End with CNTL/Z.
csr342(config)#no vrf defi
csr342(config)#no vrf definition test_vrf
% IPv4 and IPv6 addresses from all interfaces in VRF test_vrf have been removed
csr342(config)#do sh vrf | inc test
test                <not set>
test1               <not set>          ipv4
test100             <not set>
test1000            <not set>
test125             <not set>          ipv4
test56              <not set>          ipv4      Gi1.876
test90              <not set>
csr342(config)#

```

3. As there is a config difference between the device and ATOM, a Reconciliation task is triggered in ATOM.

As the config change in the device is related to the created service in ATOM, a Service Inventory task is created.



The Service is marked as “Non Compliant” service as shown in the “[Compliance](#)” Dashboard.

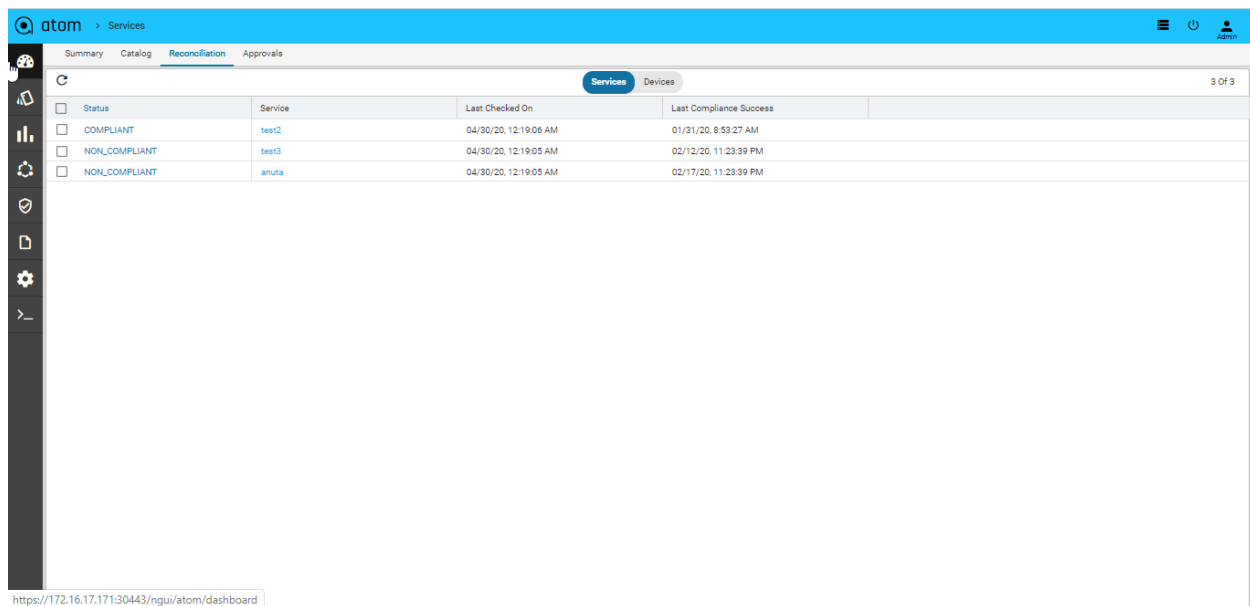
You can either resolve the service violation or look at the Reconciliation entities created.

Service Compliance

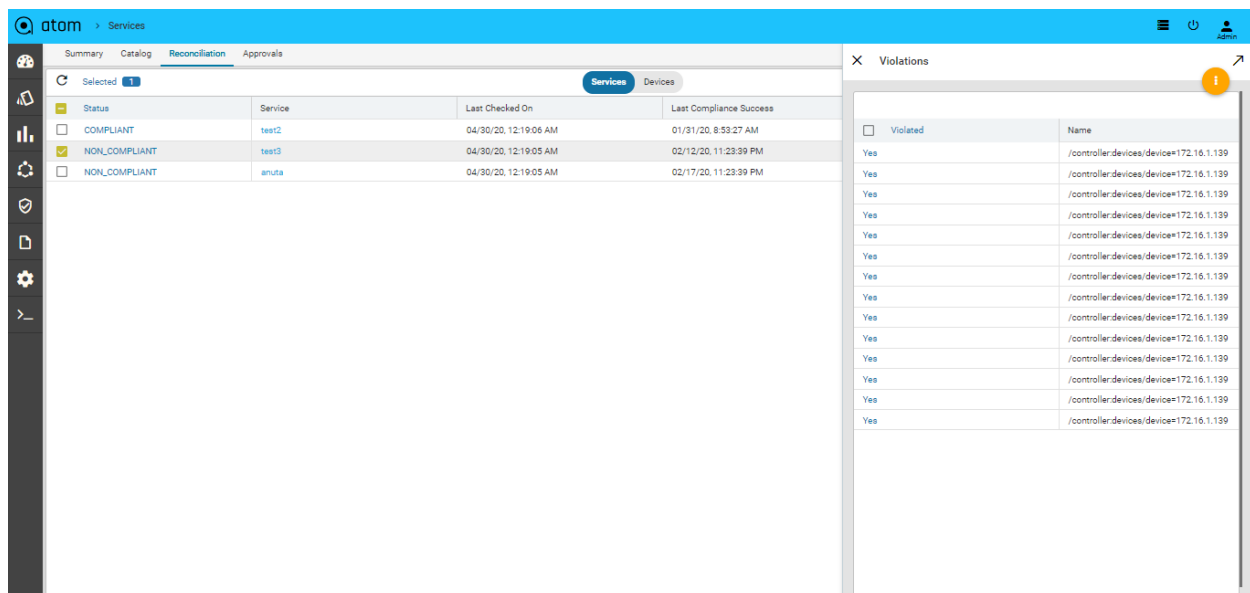
ATOM helps to detect any configuration deviations in network at the service level. ATOM detects the missing, deleted, violated configurations of the services that have been instantiated in ATOM and sends the reconciliation report.

When a service is instantiated on a device, all the necessary configurations are generated by ATOM and pushed to the device. After the successful creation of the service on the device, ATOM compares the running configuration on the device, compares this with the services that were generated, flags the violations and marks the service as non-compliant.

If there is any service that is non-compliant, navigate to **Automation -> Services->Reconciliation->Services**



Click **Non Compliant Services** to view those services



Resolving Service Violations

ATOM generates the config diff, in the service created through ATOM. and pushes the deleted configurations (that were either removed intentionally or accidentally) to the device, thus enabling the administrator to maintain the same state of configuration in both.

1. Select **Non Compliant Services** > Click **Resolve Violations**

The screenshot shows the ATOM web interface. The main panel displays a table of services under the 'Reconciliation' tab. The table has columns for Status, Service, Last Checked On, and Last Compliance Success. Two services are listed: 'test2' (COMPLIANT) and 'test3' (NON_COMPLIANT). A side panel titled 'Violations' is open on the right, showing a list of violations with a 'Resolve Violations' button at the top right.

Status	Service	Last Checked On	Last Compliance Success
COMPLIANT	test2	04/30/20, 12:19:06 AM	01/31/20, 8:53:27 AM
NON_COMPLIANT	test3	04/30/20, 12:19:05 AM	02/12/20, 11:23:59 PM
NON_COMPLIANT	anuta	04/30/20, 12:19:05 AM	02/17/20, 11:23:59 PM

- Click the Task Viewer and look for the task named **“RPC Operation: Compliance:fix-service-violations”**

The screenshot shows the Task Viewer for the task 'Compliance:fix-service-violations'. The task ID is 'EykyQinoF7R8Gq10Hhyet4g'. The time taken is '30/04/2020, 11:26:32 - 30/04/2020, 11:26:33 (0 seconds)'. The task log shows the following steps:

```

Apr 30, 2020, 11:26:32 AM RPC Operation Compliance:fix-service-violations started.
Apr 30, 2020, 11:26:32 AM Request
{"input": [{"service-compliance-id": "/controller:services/l3service_v2:l3-services/l3-service-test3"}]}
Apr 30, 2020, 11:26:33 AM Device reconciliation triggered for /controller:devices/device=172.16.1.139/l2features:vlan=124
Apr 30, 2020, 11:26:33 AM Device reconciliation triggered for /controller:devices/device=172.16.1.139/interfaces:interfaces/interface=22.124
Apr 30, 2020, 11:26:33 AM Device reconciliation triggered for /controller:devices/device=172.16.1.139/l3features:vrf=vrf-test1
Apr 30, 2020, 11:26:33 AM RPC Operation Output : {"output": [{"result": "SUCCESS", "description": "Reconcile Service /controller:services/l3service_v2:l3-services/l3-service-test3", "task-id": "EykyQinoF7R8Gq10Hhyet4g"}]}
Apr 30, 2020, 11:26:33 AM RPC Operation Compliance:fix-service-violations Completed.
  
```

Agents

ATOM Agent handles all device communication which communicates with Other ATOM Components either remotely or locally based on deployment mode.

Each ATOM Agent manages multiple network devices. ATOM agents can be assigned with multiple CIDR blocks to manage the devices. It is used to communicate, collect and monitor the networking devices in your infrastructure using standard protocols. Once the agent collects the data, it gets encrypted and sent to Anuta ATOM Server over an outgoing SSL Connection.

One Agent can typically manage hundreds of devices. However, it depends on many other factors such as device type, data collection, size of the data, frequency etc. Checkout ATOM Agent Hardware requirements for further information.

ATOM Agent Deployment is discussed in detail in "[ATOM Agent Deployment Guide](#)".

Administration

As an administrator, you can manage changes in the ATOM that will affect the behavior of the system and have a global effect on all the components of ATOM.

- "[Tasks](#)" and "[Events](#)"
- "[Tenants](#)"
- "[System](#)"
- "[System Manager](#)"
- "[Plugins and Extensions](#)"
- "[User Management](#)"
- "[Tag management](#)"
- "[UI Customizations](#)"
- "[Troubleshoot](#)"
- "[DSL](#)"
- "[Deprecated](#)"
- "[File Server](#)"
- "[About](#)"

Tasks & Events

You can view any activity, "task" that is being executed in ATOM as a result of an user- initiated

action. Tasks are generated during the following operations such as:

- Adding or Deleting Devices
- Executing Jobs
- Validating the resource pool and running the Inventory
- Configuration out-of-sync between the device and ATOM
- Creating or Deleting Networks

1. Select any Task and click **Details** to view the configurations associated with that task.
2. You can search for any Task by entering a query in the Search field.
3. Select any task and click **Cancel** to view the task is to be cancelled
4. Select any task and click **Download Log** to view the system related logs and message.

For example, enter “Create” in the Search field, if you want to query for all the Create operations that have been executed so far. All the Create tasks that have been triggered in various operations are displayed as shown below:

5. Click **Retry** when the creation of a Service (during instantiation of the Service) fails due to deficit in the operational resources or during provisioning. 4. Click Task log to view the system related logs and messages

Events

Events represent an important part of an operation or a change in the state of an object in ATOM. For example, an event is generated when a user logs in to ATOM. In addition, login attempts to a device using any of the transport types is also displayed.

Select a task and click Details to view the schema of the service, click Commands to view the configurations associated with the service generated by ATOM.

TraceLogs

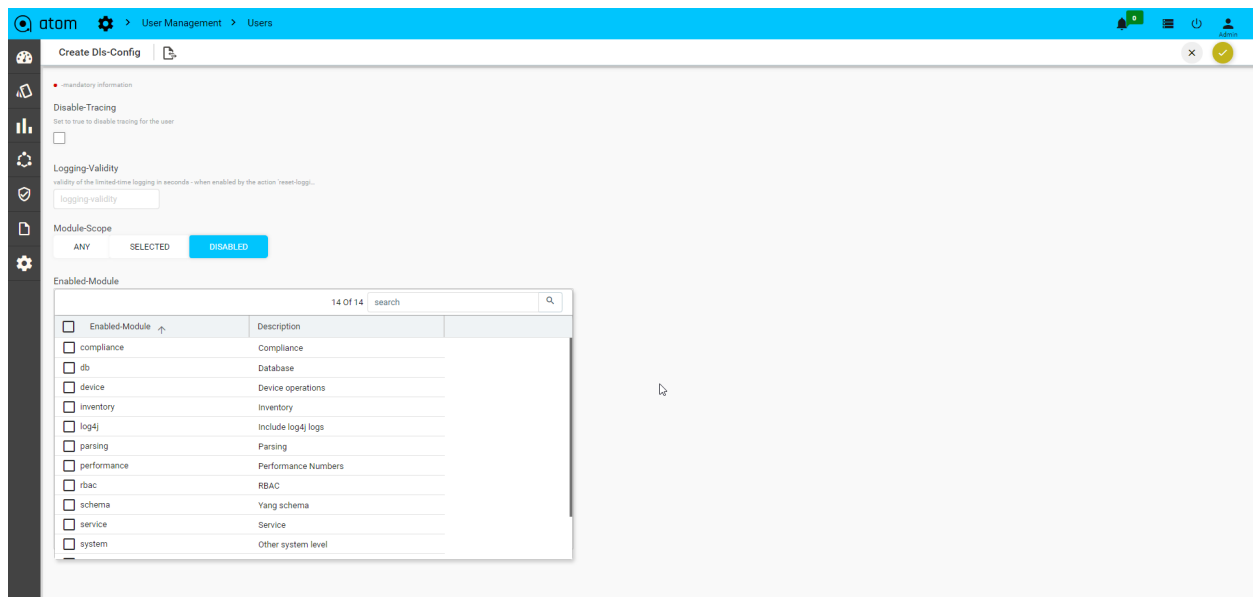
Trace Logs enables users to end-to-end distributed tracing of a task. User can monitor the performance of the task and latency optimisation can be done. It actually helps users to encounter the root cause analysis. The Tracelog option was enabled in tasks UI and also in tasks and events.

Select any task and click **Trace Logs** to view the task in distributed tracing.

Trace logs UI can be visualized from jaeger UI, this shows a complete cycle of the task and all the components involved in it.

To Enable TraceLogs Navigate to **Administration > User Management > Users**

Here select a User and **Entities > DLS-Config**



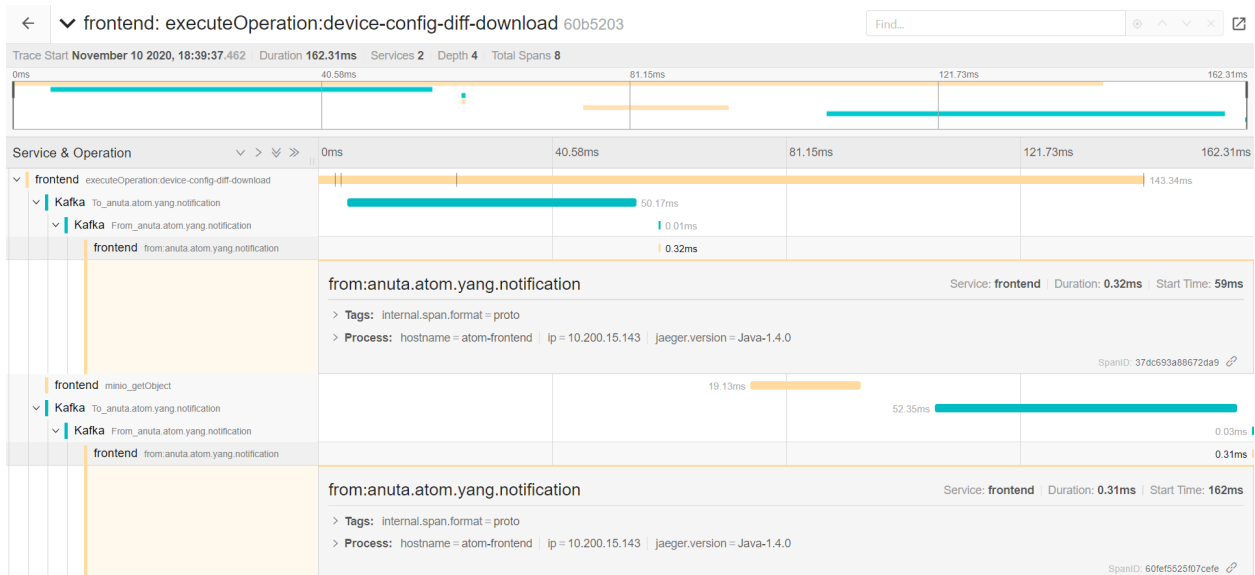
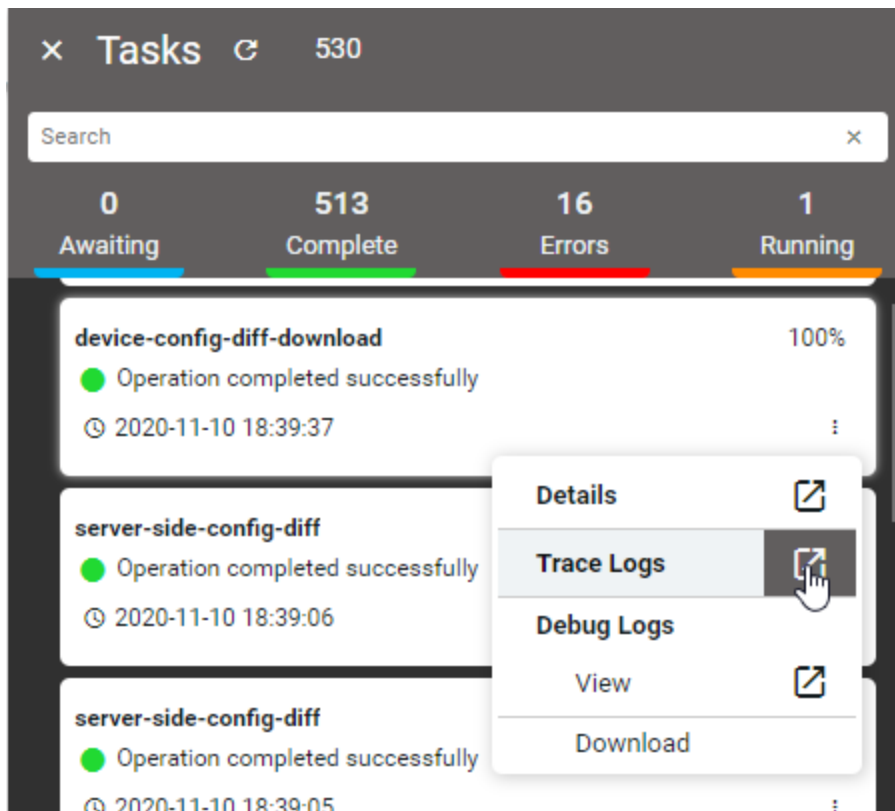
- **Disable-Tracing:** Set true to disable tracing for the user.
- **Logging-Validity:** Validity of the limited-time logging in seconds.
- **Module-Scope:** List of the modules that are supported for tracing.

Any: Enables any of the modules selected.

Selected: Enables only selected modules to tracelogs.

Disabled: Selected Module will disabled while tracing.

Note: When you select the trace logs from tasks UI it opens in the new tab as jaeger UI with SSO URLs. When you select trace logs from Tasks and Events then it opens in the ATOM application itself as a new window. To enable trace logs from deployment jaeger-tracing pods should be up.



System

As an administrator, you may want to configure or modify the system settings or customize these settings after installing ATOM.

- ["Rule Engine"](#)
- ["License"](#)

- ["General Settings"](#)
- ["Look and Feel"](#)
- ["Event Summary"](#)
- ["Notifications"](#)
- ["Message Brokers"](#)

Rule Engine

Rule engine is a functionality in which the user-defined business logic is executed to bring about changes in the state of the resources managed by ATOM. The logic describes the sequence of operations that is associated with data in a database to carry out the rule. You can create rules in the Rule engine for ATOM to handle changes in devices in a maintainable, reusable, extensible way. Rule engines support rules, conditions, priority (based on index), and other functions. Rules can be constructed to serve various functions, some of which are listed below:

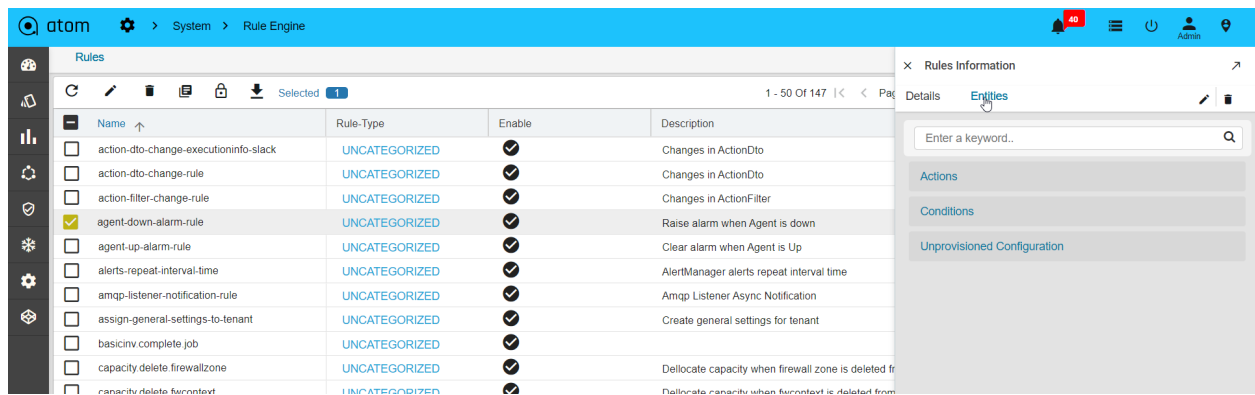
- Resources Validation
- Triggering different actions based on some user defined conditions

All the system defined rules available in ATOM as shown in the following snippet:

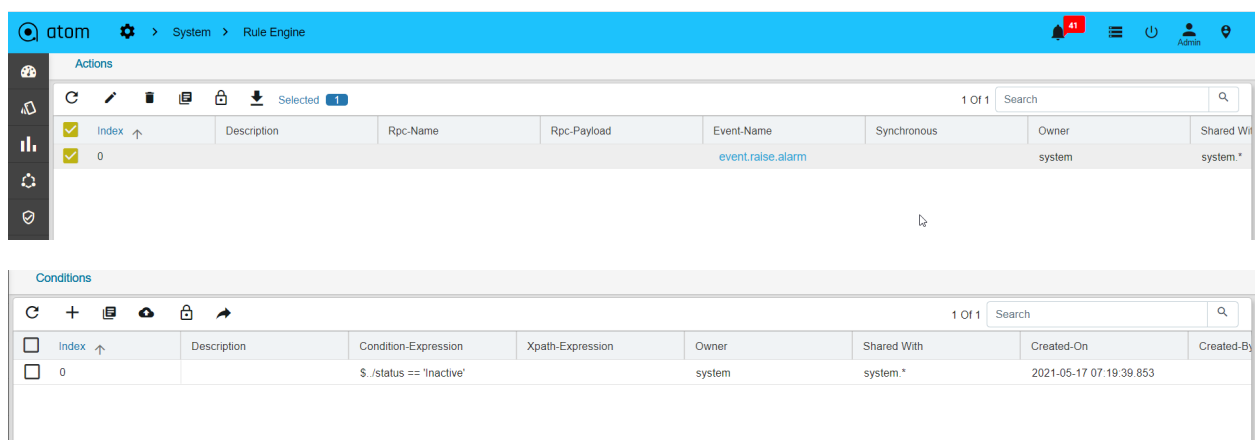
Name	Rule-Type	Enable	Description	Match-Type	Context-Path
Cache-Flush-Rule	UNCATEGORIZED	✓		MATCH_ALL	/controller.devices/de
Capacity Max Limit Create	UNCATEGORIZED	✗		MATCH_ALL	/capacities device-ca
Capacity Max Limit Delete	UNCATEGORIZED	✗		MATCH_ALL	/capacities device-ca
Capacity Max Limit Update	UNCATEGORIZED	✗		MATCH_ALL	/capacities device-ca
Compliance_Rule	COMPLIANCE	✓		MATCH_ANY	/controller.devices/de
	UNCATEGORIZED	✓		MATCH_ALL	/controller.devices/de
	UNCATEGORIZED	✓		MATCH_ALL	/controller.devices/de
	UNCATEGORIZED	✓		MATCH_ALL	/metric-instance-sche
	UNCATEGORIZED	✓		MATCH_ALL	/controller.devices/de
	UNCATEGORIZED	✓	Update Device Family Capability rule	MATCH_ALL	/controller.devices-sup
	UNCATEGORIZED	✓	Update Device Family Capability rule	MATCH_ALL	/controller.device-sup
	UNCATEGORIZED	✓		MATCH_ALL	/controller.devices/de
	UNCATEGORIZED	✓		MATCH_ALL	/controller.devices/de
	UNCATEGORIZED	✓		MATCH_ALL	/controller.device-sup
	UNCATEGORIZED	✓	Update system entity rule	MATCH_ALL	/topology network-co
	UNCATEGORIZED	✓		MATCH_ALL	/controller.devices/de
	UNCATEGORIZED	✓	Changes in ActionDto	MATCH_ALL	/alert-def actions/acti
	UNCATEGORIZED	✓	Changes in ActionDto	MATCH_ALL	/alert-def actions/acti

Click on any rule > Entities of your interest and view the Actions and Conditions associated with that rule.

For example, double click the 'agent-down-alarm' rule as shown below:



This rule comes to effect when any of the ATOM Agents goes OFFLINE and the status is set to INACTIVE.



In addition to the rules that are available by default, you can create a custom rule as per your requirement as described in the following section.

Rule

Rules are conditional statements that govern the conduct of business processes. A rule consists of a condition and a set of actions. If that condition is met, and is evaluated as true then the rule engine initiates one or more actions.

A rule is composed of three parts:

1. **Condition** - The condition part is a logical test that, if satisfied or evaluates to true, causes the action to be carried out
2. **Action** - The action part consists of one or more actions that need to be performed when the condition is met.
3. **Event** - The event part specifies the signal that triggers the invocation of the rule.

Create Rule:

1. Navigate to **Administration > System > Rule Engine > Rules**

2. Click **Add Rule** and fill the following fields:

- **Name:** Enter a string that will be used to identify the rule.
- **Rule Type:** Select the category that the rule should belong to.

There are two types of categories available now:

- **UNCATEGORIZED**
- **COMPLIANCE**
- **Enable:** Select this option if the rule should be enabled.
- **Rule Context:** Enter the context in which the rule has to be triggered:

The screenshot shows the 'Create Rule' dialog box. On the left, there's a sidebar with 'Entities' and a 'Rule - Rule' button. The main area contains the following fields:

- Name:** A text input field with the placeholder 'Rule'.
- Context-Path:** A text input field with the placeholder '/controller:devices/device'.
- Rule-Type:** A dropdown menu currently showing 'UNCATEGORIZED'.
- Change-Type:** A list box containing 'CREATE', 'UPDATE', 'DELETE', 'REPLACE', 'CREATE_UPDATE_REPLACE', and 'ANY'.
- Enable:** A checkbox that is checked.
- Rule-Context:** A dropdown menu currently showing 'DATAMODEL'.
- Description:** A text input field with the placeholder 'Description'.
- Match-Type:** A text input field with the placeholder 'Strategy to evaluate the Rule Conditions. In case of multiple rule conditions, an action is e'.

At the top right of the dialog, there are two circular buttons: a grey one with an 'X' and a yellow one with a checkmark.

- **DATAMODEL:** Select this option if the rule should be triggered on a ATOM managed entity.
 - Context path: Example: For this rule to be applicable on the devices, enter the context path as /controller:devices
- **EVENT:** Select this option if the rule should be triggered in the case of an event generated in ATOM.
 - Event Spec: Select from the available event specs in ATOM:

- Description: Enter descriptive text for the rule
- Change type The rule engine will check for the conditions defined in the rule when one of the following scenarios listed below:

Change -Type Description

- CREATE A component is created in ATOM
- UPDATE A component is updated in ATOM
- DELETE A component is deleted from ATOM
- Match-Type The conditions can be evaluated on an ANY or ALL basis.
 - Match-All: . All the conditions will be matched before executing the action.
 - Match-Any: Any condition of the condition-set will be matched before executing the action.

Create conditions?

Conditions are statements that should be qualified by the system before subsequent actions can take place. In other words, conditions are what the rule is looking for to trigger an action.

1. Navigate to **Administration > System > Rule Engine > Add Rule**
2. In the **Create Rule > Select Entity > click on + > conditions > click on + > rule-condition**
3. In the right pane, enter values in the following fields:
 - **Index:** Enter a unique number as an identifier

- **Condition-Expression:** Enter an expression that should be checked by the rule engine for the condition to be true.

Example: To check for a condition when the device is ONLINE, enter an expression: `/controller:devices/device/status == 'ONLINE'`

NOTE: Condition expression * means that the rule is triggered on all the conditions as defined in the context path.

- **Description:** Enter some text describing the condition.

Create actions?

Actions are operations that will be performed on the entities managed by ATOM once that the condition is evaluated as true by the Rule Engine.

1. Navigate to **Administration > System > Rule Engine > Add Rule**
2. In the **Create Rule > Select Entity > click on + > Actions**
3. In the right pane, enter values in the following fields:
 - **Index:** Enter a unique number that will be used as an identifier and also setting the priority
 - **Type:** Select the type of the component that should be acted upon, once the set condition is true.
 - **Description:** Enter a description for the rule-action
 - **Event-Name:** Select the appropriate event-name from the drop-down menu

Licensing & Entitlements

Usage limits in ATOM are enforced through a license file issued by Anuta Networks.

ATOM in Dedicated Mode

License File can be applied at System Level or at Each Tenant. This is applicable to ATOM Cloud Customers using a Silo/Dedicated Instance or an On-Premises instance. Following are the Admin & Tenant privileges:

1. Anuta Networks will issue the License
2. For On-Premises Deployment - Customer will apply the License
3. For ATOM Cloud Deployment - ATOM Cloud Administrator will apply the License
4. System Admin will have full access (View, Apply & Usage) to System and Tenant License Files

5. Tenant Admin/User will be able to view Available licenses and Usage for Tenant they are assigned to

ATOM in Multi-Tenant or Shared Mode

This is applicable only in ATOM Cloud Following are the Admin & Tenant privileges:

1. Anuta Networks will issue the License
2. Anuta Cloud Administrator will issue & Apply the License for each Tenant
3. System Admin will have full (View, Apply & Usage) access to System and Tenant License Files
4. Tenant Admin/User will be able to view Available licenses and Usage for Tenant they are assigned to

Uploading a License

We can upload the license using the upload button. Multiple license files can be uploaded to ATOM. Usage limits are cumulative of all License Files.

The screenshot shows the ATOM License management interface. At the top, there are six summary cards: Total Usage(C1) 0/130, Total Usage(C2) 0/120, Total Usage(C3) 0/130, Total Usage(C4) 0/130, Active Licenses 3, and Expiry Date Dec 31,2023. Below these is a table of licenses with columns: License-Tier, Device-Limit-C1, Device-Limit-C2, Device-Limit-C3, Device-Limit-C4, License-Type, Expiry-Date, Grace-Period, Customer-Name, and License. An 'Upload License(.lic) File' button is visible. An 'Upload license' dialog box is open, showing a 'Choose File' button, 'Cancel', and 'Upload' buttons.

License-Tier	Device-Limit-C1	Device-Limit-C2	Device-Limit-C3	Device-Limit-C4	License-Type	Expiry-Date	Grace-Period	Customer-Name	License
STANDARD	100	100	100	100	TRIAL	2021-12-31	5	system	Valid
STANDARD	20	10	10	20	TRIAL	2023-12-31	10	pepsi	Valid
STANDARD	10	10	20	10	TRIAL	2022-12-31	10	coke	Valid

License Summary

To view the License Summary & Details - Navigate to **Administration > License**

License Summary will show the overall summary across all the License Files.

This screenshot is identical to the one above, showing the ATOM License management interface with the same summary cards and license table.

Total Usage(C1): C1 Category Licenses Used vs Allowed

Total Usage(C2): C2 Category Licenses Used vs Allowed

Total Usage(C3): C3 Category Licenses Used vs Allowed

Total Usage(C4): C4 Category Licenses Used vs Allowed

Active licenses: Number of licenses which are active will be shown under active licenses

Expiry date: Farthest expiry date among all the License Files

Below the License Summary, all available License File details are shown as below:

The screenshot shows the ATOM License Summary dashboard. At the top, there are six summary cards: Total Usage(C1) 0/130, Total Usage(C2) 0/120, Total Usage(C3) 0/130, Total Usage(C4) 0/130, Active Licenses 3, and Expiry Date Dec 31,2023. Below these is a table with 3 rows and 10 columns. The columns are: License-Tier, Device-Limit-C1, Device-Limit-C2, Device-Limit-C3, Device-Limit-C4, License-Type, Expiry-Date, Grace-Period, Customer-Name, and License-Status. The rows show three different license tiers (STANDARD) with varying device limits and expiration dates.

License-Tier	Device-Limit-C1	Device-Limit-C2	Device-Limit-C3	Device-Limit-C4	License-Type	Expiry-Date	Grace-Period	Customer-Name	License-Status
STANDARD	100	100	100	100	TRIAL	2021-12-31	5	system	Valid
STANDARD	20	10	10	20	TRIAL	2023-12-31	10	pepsi	Valid
STANDARD	10	10	20	10	TRIAL	2022-12-31	10	coke	Valid

Tenant admin license

The screenshot shows the ATOM Tenant Admin License dashboard. At the top, there are six summary cards: Total Usage(C1) 0/100, Total Usage(C2) 1/100, Total Usage(C3) 0/100, Total Usage(C4) 0/100, Active Licenses 1, and Expiry Date Dec 31,2021. Below these is a table with 1 row and 7 columns. The columns are: Expiry-Date, Grace-Period, Customer-Name, License-Status, Deployment-Type, Dedicated-Deployment-Enabled, and License-Tenant. The row shows a single license entry for 'ubo' with an expiration date of 2021-12-31.

Expiry-Date	Grace-Period	Customer-Name	License-Status	Deployment-Type	Dedicated-Deployment-Enabled	License-Tenant
2021-12-31	10	ubo	Valid	ONPREMISE	YES	ubo

General Settings

You can edit and save the change the configuration parameters for each module in ATOM and these global changes are applicable to all the resources contained in each module.

1. Go to **Administration > System > General Settings**
2. In the **General Settings** panel, you can review the default settings of the following options and modify them.
3. Click **Edit** to modify the parameters arranged for each module.

URL Management

1. **Base URL:** This option enables the administrator to set the address (Base URL) for the third-party clients to make API request calls to ATOM server. The format of the Base URL

is `http[s]://ip|hostname`, where `ip` is the IP address of the ATOM server and the `hostname` is the host name of the ATOM server.

2. **Support URL:** Enter the URL of the support to login to the support portal of Anuta Networks
3. **User Session Timeout:** This is the time that you can set for the ATOM server to timeout if no activity takes place in the browser for a specified period of time. The user will be automatically logged out of the session, after the expiry of the specified time.

Alert Monitoring

1. **Unwanted-alertlabel-keys:** Each alert consists of multiple labels like alert name, app, collections_name etc, out of which some may not be persisted in Atom.

Chart-setting

1. **Chart-theme:** Select the chart theme from drop down, it should show in the monitoring custom chart.
2. **Chart-refresh-interval:** To set the default refresh interval, it should refresh the chart based on the given interval time in this global set.

Device Management

1. **Configuration Retrieval:** This option enables the server to retrieve configurations from the devices after each operation. By default, this option is selected.
2. **Syslog Configuration:** This option enables ATOM to configure the device to send syslog events. By default, this option is selected.
3. **Persist Configuration:** This option enables the configurations to persist in the NVRAM of the device after each provisioning.
4. **Dry Run:** This option enables ATOM to push the commands to the device or not. When selected the commands are pushed to the device..
5. **Auto Retry:** Select this option to enable ATOM to try establishing the connection with the device in case the connection is lost initially
 - a. **Number of Retries:** Enter the number of times that ATOM should try establishing the connection with the device in case of failure.
 - b. **Retry Wait Time:** Enter the time period that ATOM should wait between subsequent retries.
6. **Configuration Parsing:** This option enables ATOM to parse the configuration retrieved from the device and store the configuration data in the data model maintained in ATOM.
7. **Configuration Pull Type:** This option determines how the mode of retrieving the running configuration from the device

- TFTP_EXPORT - The running configurations are obtained from the TFTP server
 - SHOW_COMMAND - The current configuration on the device are obtained by ATOM
8. **Log Running Config:** This option enables ATOM to dump the retrieved configurations from the device in the logs obtained for the Config Retrieval Jobs.
 9. **Run-extended-inventory:** TConfiguration settings on disabling the extended inventory when device is added
 10. **Generate-config-inventory-event:** This option allows ATOM to enable or disable the config inventory event

Service now

- 1.**Snow-instance** : To enable service now option to perform the service now workflow
- 2.**Snow-url** : To provide the instance id for service-now
- 3.**Snow-username** : User name for service now
- 4.**Snow-password** : Password for service now

Service Management

1. **Service Auto Retry:** Select this option if ATOM should retry pushing the configurations to the device in the event of a service failure.
 - a. **Service Number of retries:** Enter the number of times ATOM should retry sending the configuration after initial failure of the service.
 - b. **Service Retry Wait Time:** Enter the duration of the time that ATOM should wait before trying to establish a connection with the device again.
2. **Auto Delete Stale Inv Data:** Select this option if all the available “stale” entries should be deleted from ATOM. Stale entries are the configurations that are available in the device and not seen in ATOM. These differences are not due to the service configurations created by ATOM and pushed to the device.
3. Delayed Event Buffer Time:

TSDB

- 1.**Retention-period:** Retention period of prometheus db
- 2.**Retention-size:** Retention period of prometheus db
- 3.**Namespace:** Namespace to get the stateful sets and config map
- 4.**Tsdb-url:** Url to get metrics
- 5.**Workflow-url:**The url to get workflow

6.**Tsdb-config-map-name**:The name of the tsdb config map

7.**Tsdb-stats-name**:The statefulset name of the tsdb server

8.**Tsdb-infra-alert-config-map-name**:The name of the tsdb config map for system alert

9.**Alert-repeat-interval**:The repeat interval time of the alerts

10.**Tsdb-alert-manager-config-map-name**:The name of the alert manager config map

SNMP v2 Configurations

1. **Enable Device Audit Trail Mode**: Select this option to view all the events generated in ATOM while communicating with the device using the SNMP protocol. SNMP events such as SNMP WALK, GET, DEVICE_LOGIN and DEVICE_COMMAND EXECUTION are captured in ATOM as Events.
2. **Enable Multi Tenancy**: Select this option to enable ownership of the resource. Once this option is enabled, the fields “Owner & Sharedwith” are displayed
3. **SNMP Configuration Contact**: Enter the mailing ID for contacting the admin (support) managing the SNMP server.
4. **SNMP Configuration Location**: Enter the location of SNMP server
5. **SNMP Community string**: Enter the community string required for authentication in SNMPv2 sessions..

SMTP Configurations

You may have to configure an external email server to send email notifications to the ATOM users.

SMTP Mail From: You can set up an external SMTP email server to send email notifications to the ATOM users. To do so, enter values in the fields described below:

1. **SMTP Host**: Enter the name of the server that will send the email.
2. **SMTP Port**: Enter the number of the port that is used to connect to the SMTP host
3. **SMTP Auth Required**: Enable this option if authentication is required to connect to the SMTP Host
4. **SMTP User Name**: Enter the name of SMTP user
5. **SMTP Password**: Enter the password to retrieve the email.
6. **SMTP Encryption SSL**: Select this option if the connection to the SMTP server should use SSL as the authentication method.
7. **SMTP Encryption TLS**: Select this option if the connection to the SMTP server should use TLS as the authentication method

Notification

Email Notifications: Select this option if you wish to be notified via email about changes taking place in the system.

License Expiry Threshold (days): Set the number of days to notify the user that the license is about to expire.

Python Remote Debug

Python Remote Debug: Select this option if you want to allow debugging of the logs in ATOM remotely.

Debug Server Port: Enter the port number of the remote debug server.

Developer Options

Enable Developer Mode: Select this option for the Developer Options to be visible in ATOM.

Using the Developer Options, the admin can view the all the ATOM entities represented in the data model tree, figure out the xpaths of the objects , all the device and service ATOM SDK

System Maintenance

Enable Maintenance Mode: Select this option if the system needs to be suspended for some time during which no operation can be performed on the ATOM VM. All the TASKS running in ATOM should be in “COMPLETE” state before enabling this option.

View Actions

Show Module Prefixes: Enable this option if the module prefixes for child entities on Profile and Action items should be made visible.

Request Sanitization

As an administrator , you can protect the data entered in ATOM from malicious attacks in the form of HTML tags. These tags when injected into the application’s HTML code can make ATOM vulnerable to these attacks and have a large impact as any user of the application can be a target.

The data entered in ATOM can be sanitized based on the

1. **Security Sanitizer Enabled:** Select this option if the sanitization filter should be enabled in ATOM.
2. **Sanitizer Exclude URL:** Enter the tags that should be excluded from filtering. The patterns mentioned here are allowed as values in the text fields in any of the HTML forms used in ATOM. For example, These tags can be added in the exclusion list
-/login,/initialize,/logout,/*.js,/controller:admin-settings\$
3. **Sanitizer Patterns:** Enter the patterns that will be used to sanitize the data and should not be allowed as values in any of the fields in any of the text fields in the HTML forms of ATOM.

An appropriate error message is displayed in the webpage when the user inputted data matches with the sanitizer pattern mentioned above.

Password Profile

The parameters required for a password that is used to authenticate logging into ATOM can be

1. **Password Expiry Days:** Enter the number of days for which the set password is valid.
2. **Password Pattern:**
3. **Password Min Length:** Enter the minimum number of characters that should be contained in the password used to authenticate the logging into ATOM.
4. **Password Max Length:** Enter the maximum number of characters that should be contained in a password used to authenticate the logging into ATOM.

Primary container load limit

1.Set the range of container load limit

Workflow

1.If true both delete both active and historical instances on package unload,if false, history will be maintained but unload will fail if active instances are there.

Look and Feel

You can change the "look and feel" of ATOM's GUI by uploading images of your choice to ATOM.

1. Navigate to **Administration > UI Customizations > Look & Feel**
 - **Product Logo:** Select the image that you should be displayed as the product logo, visible on all the screens.
 - **Login Screen Logo:** Select the image that should be displayed on the login screen.
2. Click **Update** for the uploaded images to come into effect on the UI
or
3. Click **Defaults** to revert to the default images.

Event Summary

ATOM generated events are grouped into different categories, (Alarm, Services and System) with an assigned severity to each category. ATOM maintains an event catalog and decides how and when an event is created and whether to associate an alarm with the event. Events are generated in ATOM through notifications received via the syslog and trap messages, inventory changes, discovery of the devices, changes in the ATOM server itself.

The severity of events can be classified into:

- **CRITICAL** - Events that demand the immediate attention of the system administrator. They are generally directed at the global (system-wide) level, such as System or

Application. They can also be used to indicate that an application or system has failed or stopped responding.

- **WARNING** - A warning indicates that a component or application is not in an ideal state and that some further actions could result in a critical error. These can be treated as forewarning of a problem that might occur.
- **ERROR** - Events that indicate problems, but in a category that does not require immediate attention.
- **INFORMATIONAL** - Events that pass noncritical information to the administrator.

Not all Events are associated with Alarms. Multiple events can be mapped to the same alarm. All Alarm Events are associated with an alarm that can be either state, CLEARED or ACTIVE.

Notifications

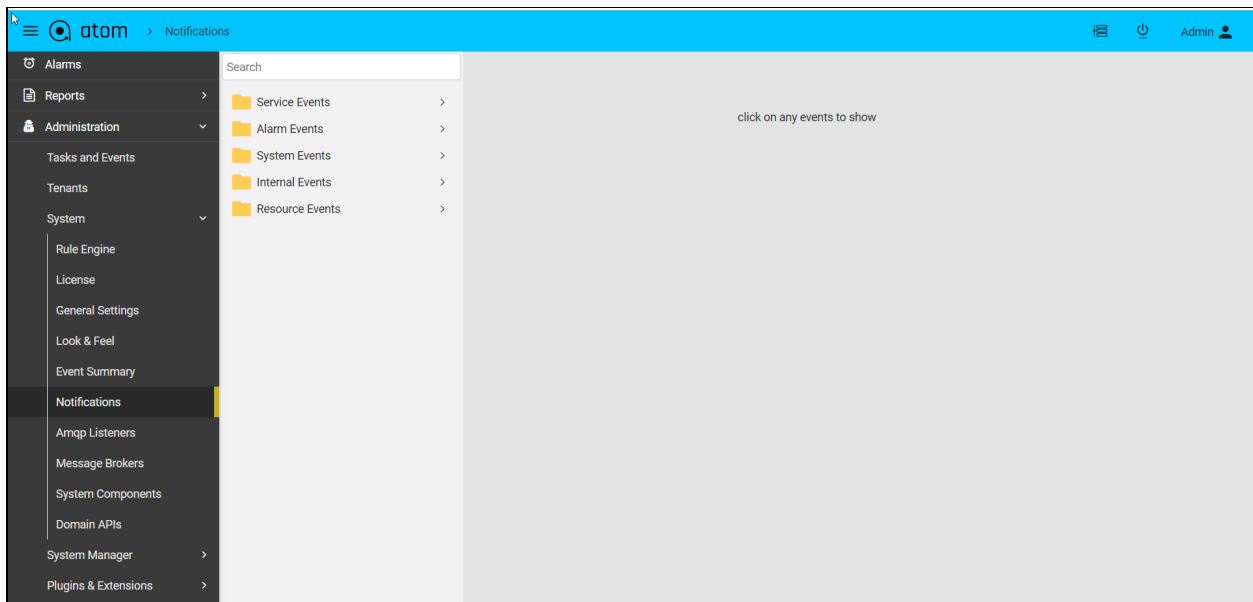
All the events that are triggered in ATOM due to various reasons such as change in the component state, device unreachability, high CPU usage of the system and so on can be notified to subscribers. As an administrator, you can create notifications such that users, message brokers can be notified when an event is triggered.

Subscribers can be added to the events falling in any of these categories:

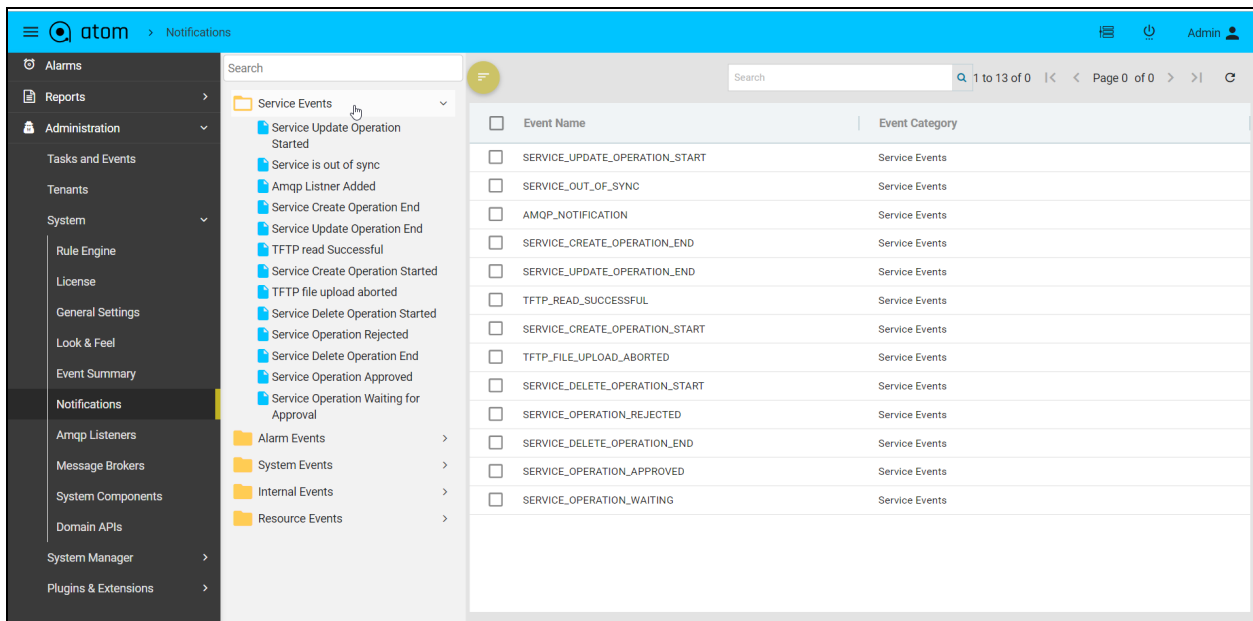
- Alarm Events
- Internal Events
- System Events
- Resource Events
- Service Events

Each of these categories contain many Events, pre-defined in ATOM. You can create subscribers or the recipients of a particular notification.

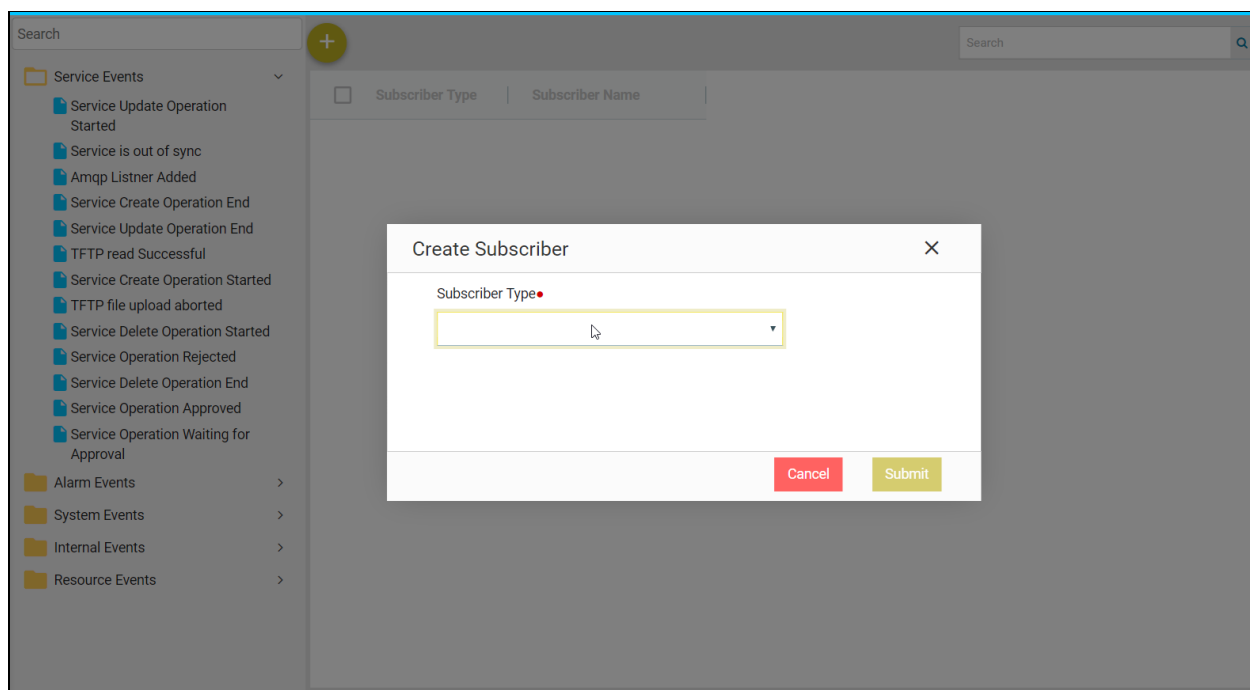
1. Navigate to **Administration > System > Notifications**



2. In the left pane, click a folder (of your choice). All the events are grouped into different categories
3. In the expanded view of the folder, select an Event as shown below:



4. In the **Events Subscribers** pane, click **Add** to the subscriber to the ATOM generated event.



5. In the **Create Subscriber (s)** screen, choose the type of the Subscriber:

- **User**
 - These users are the users created in ATOM . See the section, Creating Users in ATOM
- **AMQP Broker**
 - These are the message brokers where the events generated in ATOM are published. See the section, Creating Message Brokers for more information.

Message Brokers

As an administrator, you can configure a message broker to publish the events generated by ATOM. A message broker can be notified of all events or an event belonging to a specific event type.

In the current implementation of ATOM message brokers, RabbitMQ is the supported AMQP server where ATOM publishes the events and from where any AMQP listener consumes them.

Prerequisites:

Before creating Message Brokers in ATOM, check whether a virtual host exists in RabbitMQ.

If there is no existing vHost, create a new virtual host as shown below:

1. Login to RabbitMQ and go to the Virtual Hosts tab to add a new vHost

2. Add users for the created virtual host
3. Create users with required permissions in the vHost

Creating Message Brokers in ATOM:

1. Navigate to **Administration > System > Message Brokers** > click Add
2. In the **Create message broker** screen, enter the following fields as described below:

Create Message-Broker

• -mandatory information

Broker-Address •
Address of the broker
broker-address

Port Number •
Portnumber of the broker
Port Number

User Name •
Username of the broker
User Name

Password •
Password of the broker
Password

Vhost
Virtual Host
/

Exchange •
Exchange
exchange

Enable-Publishing

- **Broker Address:** Enter the IP address of the message broker (AMQP server which in our case is RabbitMQ).
- **Port Number:** This is the port number used to communicate with ATOM. By default, it is 5673.
- **Username and Password:** Enter the credentials for logging into the message broker.
- **vHost:** Enter the virtual host name that was created in RabbitMQ. Refer "Prerequisites" section.
- **Exchange:** Enter the name of the exchange in the message broker where the events generated in ATOM should be published. The default name is "ATOMNotifications".
- **Enable Publishing:** This option enables the administrator to enable or disable sending notifications from ATOM to the exchange. By default, the checkbox is selected, which means that the events can be sent to the broker.

- **Connection Status:** After saving the Message Broker, this field will be updated to True/False based on the success or failure of connectivity to RabbitMQ
- Click **Add** to select the events that should be sent to the message broker.

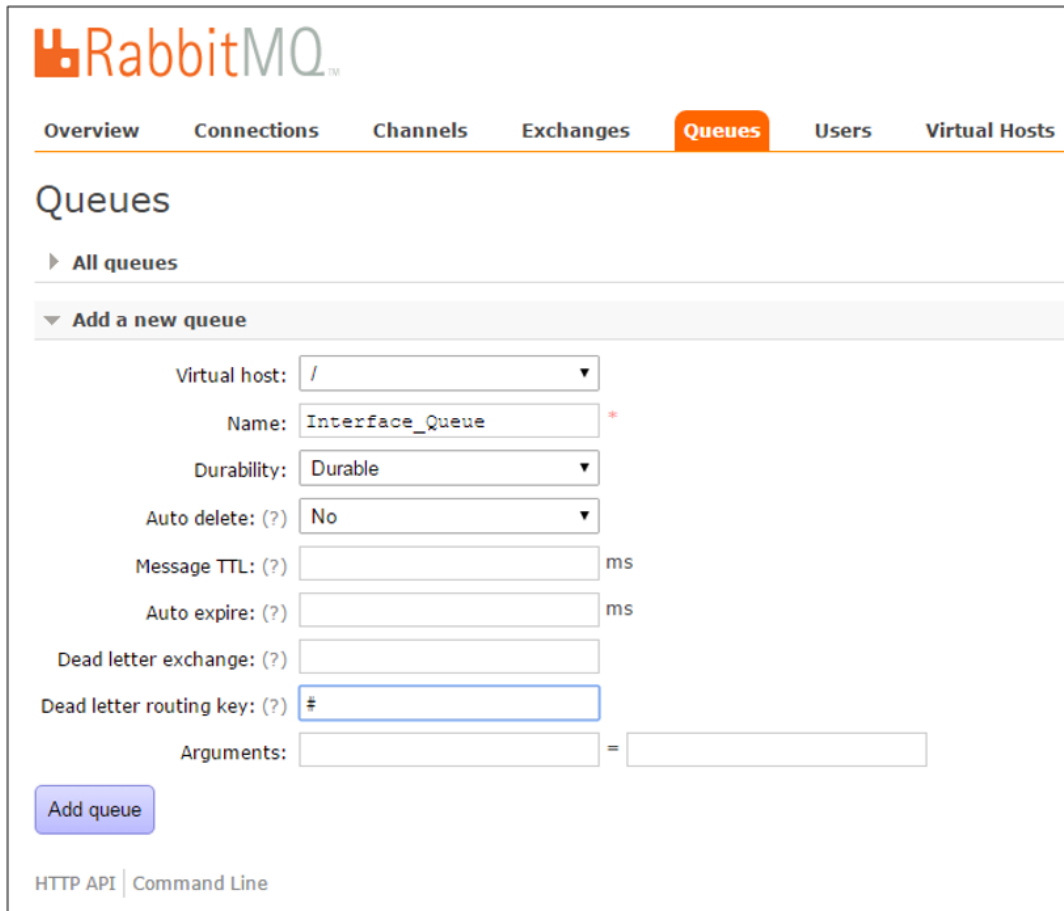
AMQP Listeners

You can create AMQP listeners in ATOM so that the messages (events) published in ATOM can be consumed by them.

In the current implementation of ATOM message brokers, RabbitMQ is the supported AMQP server where ATOM publishes the events and from where any AMQP listener consumes them.

Prerequisites:

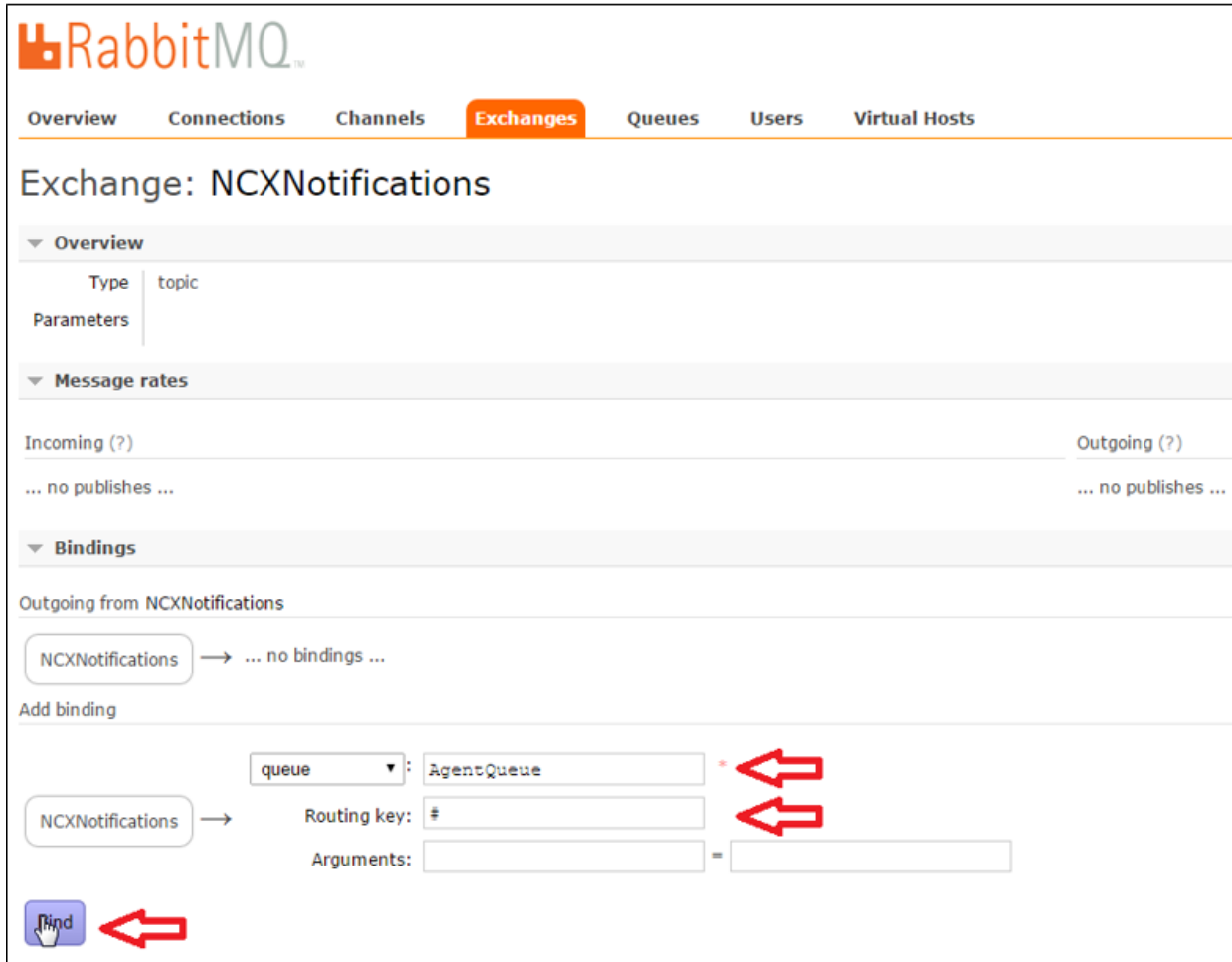
1. Login to RabbitMQ Message Broker using the appropriate credentials
2. Create Queues in RabbitMQ
 - a. Login to RabbitMQ and go to the Queues tab.
 - b. Add a new queue as shown below:



The screenshot displays the RabbitMQ web management interface. At the top, the RabbitMQ logo is visible. Below it, a navigation bar includes tabs for Overview, Connections, Channels, Exchanges, Queues (which is currently selected and highlighted in orange), Users, and Virtual Hosts. The main heading is 'Queues'. Underneath, there is a link for 'All queues' and a section titled 'Add a new queue' which is expanded. This section contains several form fields: 'Virtual host' with a dropdown menu showing '/'; 'Name' with a text input containing 'Interface_Queue' and a red asterisk indicating a required field; 'Durability' with a dropdown menu showing 'Durable'; 'Auto delete (?)' with a dropdown menu showing 'No'; 'Message TTL (?)' with a text input and 'ms' unit; 'Auto expire (?)' with a text input and 'ms' unit; 'Dead letter exchange (?)' with a text input; 'Dead letter routing key (?)' with a text input containing '#'; and 'Arguments' with two text inputs separated by an equals sign. At the bottom left of the form is a blue 'Add queue' button. At the very bottom of the interface, there are links for 'HTTP API' and 'Command Line'.

3. Bind the Queue with the Exchange in RabbitMQ

- i. In RabbitMQ, navigate to **Exchanges**
- ii. An Exchange entry is created in RabbitMQ and connection with ATOM is established
- iii. Select the Exchange, “**ATOMNotifications**”, created in ATOM and bind the created Queue to the same as follows:



4. Verify the messages in RabbitMQ

- i. Go to **Queue > Get messages**
- ii. In the **Messages** field, enter a number for the messages that you want to be displayed.
- iii. Click **Get Messages** as shown below:

Get messages

Warning: getting messages from a queue is a destructive action. (?)

Requeue: Yes
Encoding: Auto string / base64 (?)
Messages: 30

Get Message(s)

Message 1

The server reported 275 messages remaining.

Exchange

Routing Key

Redelivered

Properties

Payload

Encoding: string

NCXNotifications

*

delivery_mode: 1
headers:

0 bytes

Message 2

The server reported 274 messages remaining.

Exchange

Routing Key

Redelivered

Properties

Payload

Encoding: string

NCXNotifications

ncloudx.task.background

*

Task_Id=7f8604ca-4305-4694-8780-39ae98f3063d;Task_Component=Server;Task_ComponentType=SERVERJOB;Task_Operation=SystemHealthCheck;Task_Status=NOT_STARTED;

153 bytes

Message 3

To create the AMQP listener in ATOM, do the following:

1. Navigate to **Administration > System > AMQP Listeners**
2. Click **Add AMQP Listener** in the right pane to create a listener in ATOM

Enter values in the fields in the Create AMQP Listener screen as described below:

Create Amqp-Listener

mandatory information

Broker-Address

Address of the amqp listener

broker-address

Port Number

Portnumber of the amqp listener

Port Number

User Name

Username of the amqp listener

User Name

Password

Password of the amqp listener

Password

Vhost

Virtual host of amqp listener

/

Queue

Queue to get all messages from Message Broker

queue

Connection-Status

- **Broker Address:** Enter the IP address of the AMQP server that will receive the notifications from ATOM.
- **Port Number:** This is the port number that needs to be configured on the message broker to listen into the ATOM notifications.
- **Username:** Enter the username of the AMQP listener.
- **Password:** Enter the password to authenticate the AMQP listener.
- **vHost:** Enter the virtual host name that was created in RabbitMQ
- **Queue:** Enter the name of the Queue created in the AMQP server (example, RabbitMQ)
- **Agent Name:** Select the ATOM agent from which the ATOM notifications are generated on a given set of devices. All the notifications created on the device managed by the ATOM agent will be published in the queue.

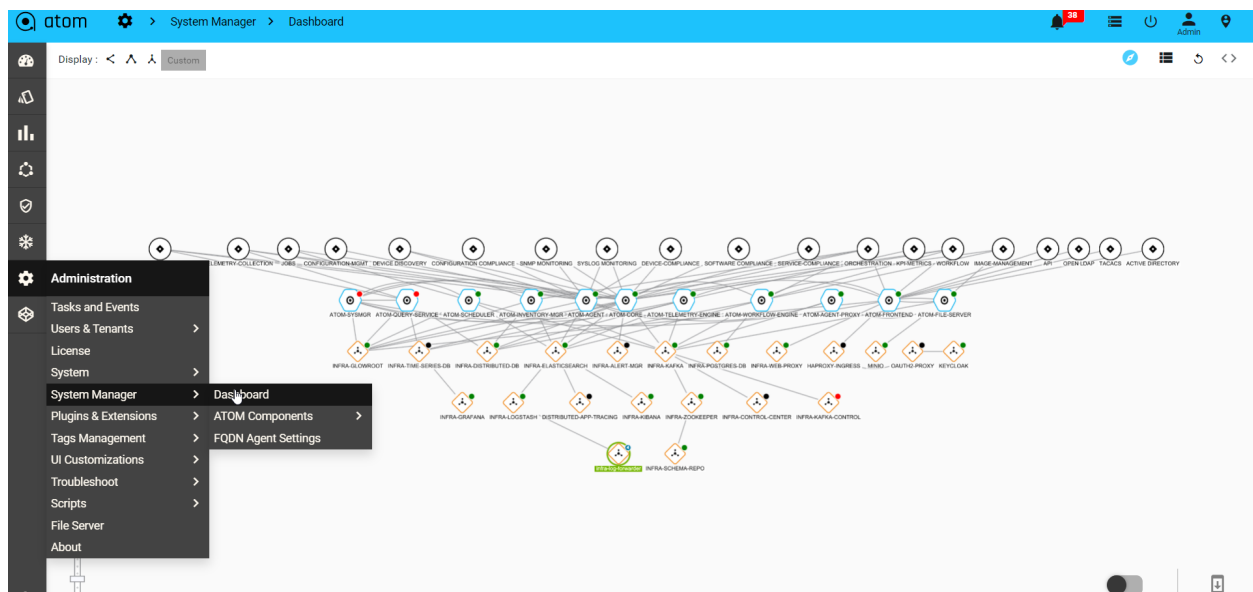
System Manager

To view the components, microservices and the applications managed by ATOM, navigate to **Administration > System Manager**

Dashboard:

View the graphical representation of the connection between the Applications, Components, and the underlying microservices.

Navigate > Administration > System Manager > Dashboard



ATOM Components:

Navigate > Administration > System Manager > Atom components > Atom components

The components, of ATOM, along with their dependencies are displayed as follows:

Name	Dependencies
ATOM-Frontend	ACTIVE DIRECTORY, ATOM-File-Server, OPEN LDAP, TACA...
ATOM-Agent	Device, infra-elasticsearch
ATOM-Query-Service	Kubernetes, infra-distributed-db, infra-time-series-db
ATOM-Telemetry-engine	ATOM-Frontend, infra-distributed-db, infra-elasticsearch, infra-...
ATOM-Workflow-engine	ATOM-Agent, ATOM-Frontend, infra-distributed-db, infra-elasti...
	ATOM-Agent, ATOM-File-Server, infra-kafka
	ATOM-Agent, ATOM-Frontend, ATOM-Inventory-mgr, ATOM-Q...
	ATOM-Frontend, infra-kafka, minio
	infra-distributed-db, infra-elasticsearch, infra-kafka
	ATOM-core, infra-elasticsearch, infra-kafka
	Alert-mgr, infra-distributed-db, in...

All Components:

Navigate > Administration > System Manager > Atom components > All components

The components of the Kubernetes cluster that are not owned by Anuta are displayed as follows:

Name	Description	Component-Owner	Prefix	Parent	Type
ACTIVE DIRECTORY	Defines Mode of Authentication and it will be communicated o...	CUSTOMER			third_party
ATOM-Agent	It is a microservice which communicates with the device direct...	Anuta			micro_service
ATOM-Agent-Proxy	A agent proxy for local and remote agents	Anuta			micro_service
ATOM-File-Server	Image server for serving images through FTP/SFTP to devices	Anuta			micro_service
ATOM-Frontend	It is the Pod which communicates with UI and send it's require...	Anuta			micro_service
	This pod is used for processing the config(Cli,Netconf)	Anuta			micro_service
	It is used to offload some batch report processing	Anuta			micro_service
	This take care of all the scheduled jobs in the atom(eg. job sc...	Anuta			micro_service
	It is used to monitor, over all system health and perform activi...	Anuta			micro_service
	Telemetry Pod will process all the telemetry data and converts ...	Anuta			micro_service
	the bpmn execution	Anuta			micro_service
		Anuta			micro_service
		3rd-party			device,infra
		Kubernetes			orchestration,infra
	Defines Mode of Authentication	CUSTOMER			third_party
	Defines Mode of Authentication	CUSTOMER			third_party
	Jaeger is a distributed tracing system	CNCF			analytics,infra
	It is used for SSO purpose	HAProxy			infra
	AlertManager used for monitoring of alerts and send those alerts	InfluxDB	Kanacitor		stream, monitoring,infra

Component Functions:

Navigate > Administration > System Manager > Atom components > Component Functions

The association of the components between the applications can be visualized as follows:

The screenshot shows the ATOM web interface with the breadcrumb path: Administration > System Manager > ATOM Components > Component Functions. The main content area displays a table titled "Comp-Functions" with 5 rows of data. The table columns are: Component, Name, Description, Owner, Shared With, Created-On, Created-By, and Last-Modified. The components listed are ATOM-Agent, ATOM-Inventory-mgr, ATOM-Telemetry-engine, ATOM-Telemetry-engine, and ATOM-core. The left sidebar shows the navigation menu with the path: Administration > System Manager > ATOM Components > Component Functions.

Component	Name	Description	Owner	Shared With	Created-On	Created-By	Last-Modified
ATOM-Agent	Configuration-Mgmt		system	system.*	2021-05-17 07:18:56.971		2021
ATOM-Inventory-mgr	Inventory		system	system.*	2021-05-17 07:18:56.971		2021
ATOM-Telemetry-engine	Telemetry		system	system.*	2021-05-17 07:18:56.971		2021
ATOM-Telemetry-engine	Assurance(CLA)		system	system.*	2021-05-17 07:18:56.971		2021
ATOM-core	Orchestration		system	system.*	2021-05-17 07:18:56.971		2021

Deployment Functions:

Navigate > Administration > System Manager > Atom components > Deployment Functions

The following deployment summaries to be displayed

The screenshot shows the ATOM web interface with the breadcrumb path: Administration > System Manager > ATOM Components > Deployment. The main content area displays a table titled "Deployment Summary" with 2 columns: Name and Replicas. The table lists various components and their replica counts. The left sidebar shows the navigation menu with the path: Administration > System Manager > ATOM Components > Deployment.

Name	Replicas
agent-lb	1
atom-agent-proxy	1
atom-core	1
atom-file-server	1
atom-frontend	1
	1
	0
	0
	0
	1
	1
	1
	1
infra-distributed-db-webagent	0

FQDN Agent Settings

Navigate > Administration > System Manager > FQDN Agent Settings

1.No need to restart any pod, it should discover any endpoints.

Command : `kubectrl edit cm -n kube-system coredns`

2.Content added as below:

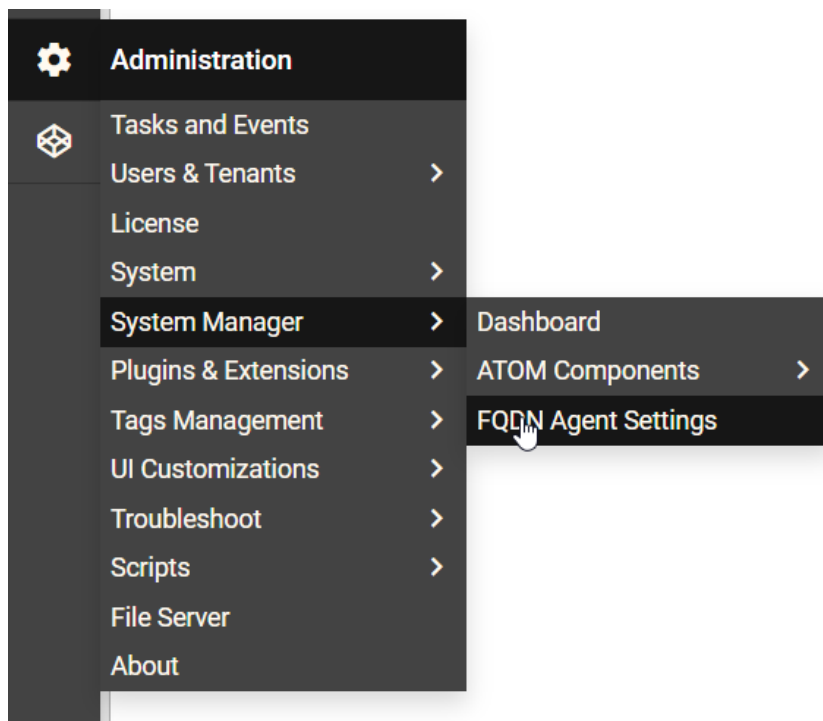
```
anutacorp.com:53 {
    errors
    cache 30
    forward . 172.16.100.5
}
```

```


Please edit the object below. Lines beginning with a '#' will be ignored,
# and an empty file will abort the edit. If an error occurs while saving this file will be
# reopened with the relevant failures.
#
apiVersion: v1
data:
  Corefile: |
    .:53 {
      errors
      health
      kubernetes cluster.local in-addr.arpa ip6.arpa {
        pods insecure
        upstream
        fallthrough in-addr.arpa ip6.arpa
      }
      prometheus :9153
      forward . /etc/resolv.conf
      cache 30
      loop
      reload
      loadbalance
    }
    anutacorp.com:53 {
      errors
      cache 30
      forward . 172.16.100.5
    }
kind: ConfigMap
metadata:
  creationTimestamp: "2019-07-02T06:07:23Z"
  name: coredns
  namespace: kube-system
  resourceVersion: "41690018"
  selfLink: /api/v1/namespaces/kube-system/configmaps/coredns
  uid: a862c69e-9c8f-11e9-8460-00505688cf0f

```

Go to Administration/System Manager/FQDN Agent setting for pattern



Click Add Symbol and Select **Default-Domain-Agent** and give proper **Pattern & Priority**

Create Fqdn-Config | 

Entities × ✓

Fqdn-Config - Agent1

Name •

agent1

Default-Domain-Agent •

default_agent × ▼

Patterns

fqdn matching regex-patterns

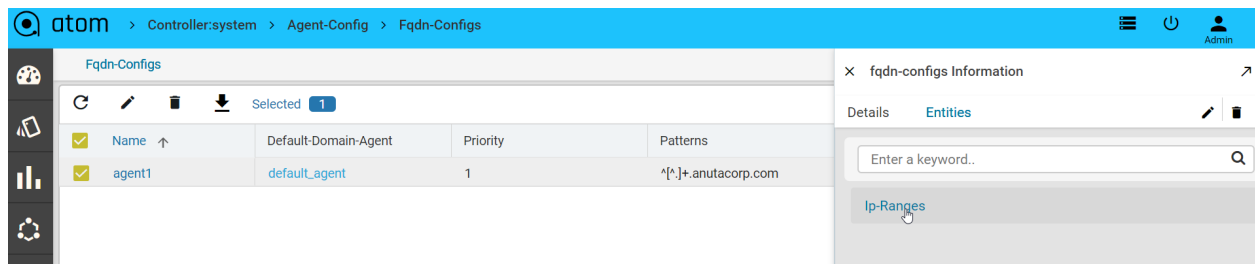
x

Priority

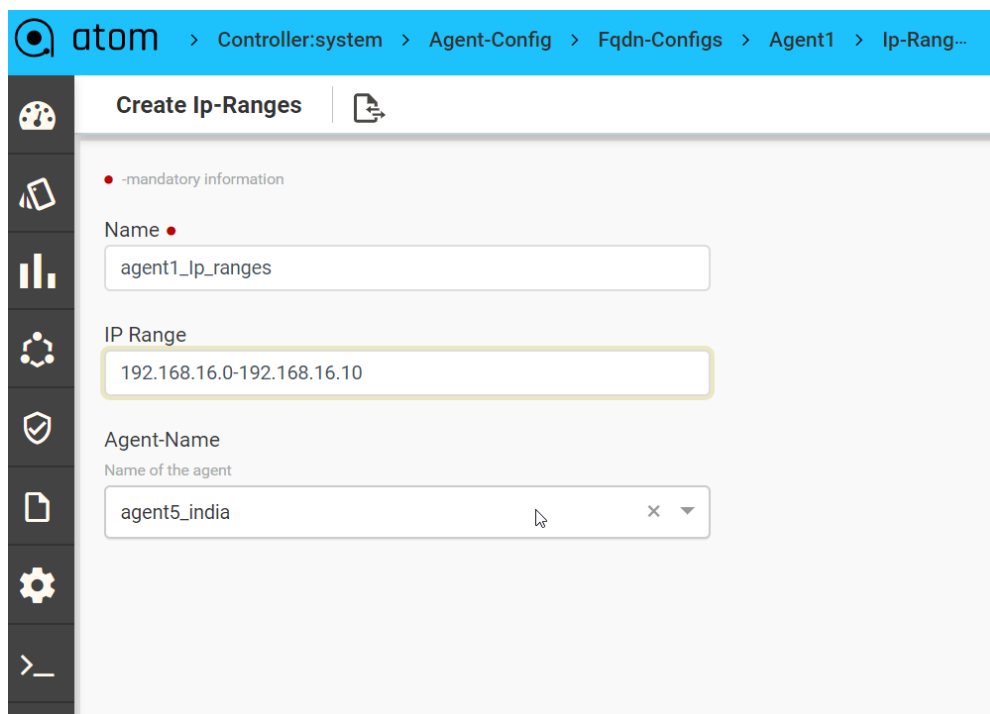
Sequence to consider regex-patterns

1 ↑ ↓


After creating FQDN then go to Entities/Ip-Ranges as shown.












Provide valid IP Ranges and select agent



Go to Devices and click add symbol and Provide **FQDN-Name** to resolve **FQDN**

 > Devices



Create Device

mandatory information

Id

Management-Mode

Managed device: ATOM manages and orchestrates such devices. UnManaged Devices:...

MANAGED

UNMANAGED

DUMMY

Name

the unique name

Fqdn-Name

ind-csr333.ind.anutacorp.com

Mgmt-Ip-Address

Must be a valid IP Address. Ex:172.16.1.24.

mgmt-ip-address

Credential Set

X

Device Type

UNKNOWN

X

Description

device type description

Description

Driver-Name

272

> Devices

Create Device

mandatory information

Id

Id

Management-Mode

Managed device: ATOM manages and orchestrates such devices. UnManaged Devices:...

MANAGED

UNMANAGED

DUMMY

Name

the unique name

Name

Fqdn-Name

ind-csr333.ind.anutacorp.com

Mgmt-Ip-Address

Must be a valid IP Address. Ex:172.16.1.24.

172.16.3.33

Credential Set

Device Type

UNKNOWN

Description

device type description

Description

Driver-Name

get-ip-from-fqdn

05/02/2020, 14:37:16 - 05/02/2020, 14:37:17

Time Taken : 0 seconds

TASKID : DgZ3xbco8xSR-PsFPGUC1PpA

2020/02/05 09:07:16 AM: RPC Operation get-ip-from-fqdn started.

2020/02/05 09:07:16 AM: **Request**

{"input": {"hostname": "ind-csr333.ind.anutacorp.com"}}

2020/02/05 09:07:16 AM: Matching FQDN config name: agent4

2020/02/05 09:07:16 AM: Domain-Agent for config name: agent4_india

2020/02/05 09:07:16 AM: Empty IP-Ranges for config name: agent4

2020/02/05 09:07:17 AM: RPC Operation Output :{"output":{"ipaddress":"172.16.3.33","resolved-by":"agent4_india"}}

2020/02/05 09:07:17 AM: RPC Operation get-ip-from-fqdn Completed.

Task completed

273

Plugins and Extensions

By utilizing the normalized device abstractions maintained in data stores written in YANG maintained in ATOM, the customers can write their own device models and applications to meet their specific operational needs, thereby utilizing the extensibility of ATOM.

The device and service packages are loaded as bundles or plugins to the ATOM container thereby making them modular. The packages can be installed, updated, or deleted without disrupting the operation of the device.

In addition to modeling the devices and service, you can model the features or network functions required to build a network service. These features thus modeled appear as icons in the feature palette of Service Designer pane of ATOM. The newly added features along with the included associated services can now be used to design the service in ATOM.

Packages

ATOM is packaged with many predefined device packages to enable you to work with many vendor devices. ATOM also provides capability to update the existing device packages and ability to add new device packages

A Device Model contains inventory models, communication model, and notification model that are packaged and uploaded to ATOM. A device package consists of models, a vendor -specific configuration data for all the different devices.

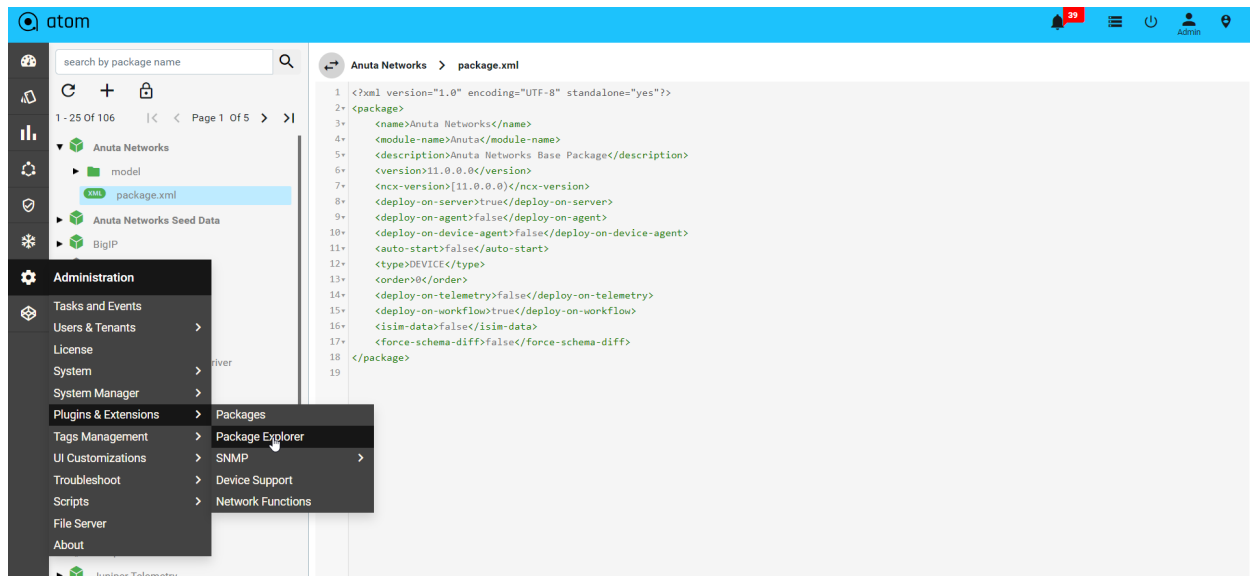
For details about what constitutes a device package and how to write it, refer to examples cited in the guide, *“ATOM Platform Guide”*.

A Service package contains the services models, service yang files and metadata information. For more information about Service Modeling, refer to examples cited in the guide, *“ATOM Platform Guide”*.

Package Explorer

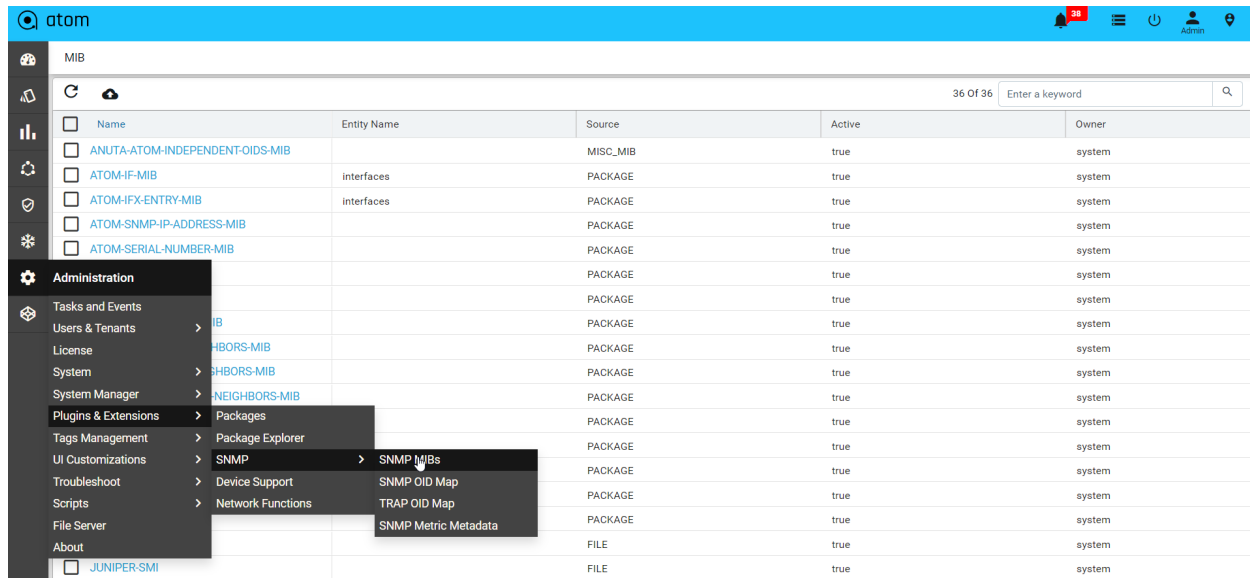
Users can manage and get the information of the packages that are currently available in the system. The packages that were uploaded and loaded can be viewed in minio UI.

Navigate to Administration > Plugins & Extensions > Package Explorer



SNMP

Navigate > Administration > Plugins & Extensions > Package > SNMP > SNMP Mibs > Upload



atom > MIB				
MIB				
27 Of 27 Enter a keyword				
<input type="checkbox"/>	Name	Entity Name	Source	Active
<input type="checkbox"/>	ANUTA-ATOM-INDEPENDENT-ODS-MIB		MISC_MIB	true
<input type="checkbox"/>	ATOM-IF-MIB	interfaces	PACKAGE	true
<input type="checkbox"/>	ATOM-IFX-ENTRY-MIB	interfaces	PACKAGE	true
<input type="checkbox"/>	ATOM-SNMP-IP-ADDRESS-MIB		PACKAGE	true
<input type="checkbox"/>	ATOM-SERIAL-NUMBER-MIB		PACKAGE	true
<input type="checkbox"/>	ATOM-OS-VERSION-MIB		PACKAGE	true
<input type="checkbox"/>	ATOM-ENTITY-MIB		PACKAGE	true
<input type="checkbox"/>	ATOM-IPv6-ADDRESS-MIB		PACKAGE	true
<input type="checkbox"/>	ATOM-SNMP-CDP-NEIGHBORS-MIB		PACKAGE	true
<input type="checkbox"/>	ATOM-SNMP-LLDP-NEIGHBORS-MIB		PACKAGE	true
<input type="checkbox"/>	ATOM-SNMP-AXIS-LLDP-NEIGHBORS-MIB		PACKAGE	true
<input type="checkbox"/>	SNMPv2-MIB		FILE	true
<input type="checkbox"/>	SNMPv2-CONF		FILE	true
<input type="checkbox"/>	SNMPv2-TC		FILE	true
<input type="checkbox"/>	SNMPv2-SMI		FILE	true
<input type="checkbox"/>	BGP4-MIB		FILE	true
<input type="checkbox"/>	BRIDGE-MIB		FILE	false
<input type="checkbox"/>	CISCO-SMI		FILE	true
<input type="checkbox"/>	cisco-process-mib		FILE	true
<input type="checkbox"/>	IANA-ENTITY-MIB		FILE	true
<input type="checkbox"/>	IF-MIB		FILE	true
<input type="checkbox"/>	IANAIfType-MIB		FILE	true
<input type="checkbox"/>	SNMP-FRAMEWORK-MIB		FILE	true
<input type="checkbox"/>	HCNUM-TC		FILE	true

Navigate > Administration > Plugins & Extensions > Package > SNMP > SNMP OID Maps > Add

atom > Old-Snmp-Maps								
SNMP-OID-Maps								
1 - 50 Of 479 < < Page 1 Of 10 > > Search								
<input type="checkbox"/>	Name	Platform	Parent	OID	Post-Processor-Prop	Fetch-Type	Is-Metric-Candidate	Is-Accessible
<input type="checkbox"/>	aristaEOSVersion	ALLIALLIALLIarista EOS/Arista Networks		.1.3.6.1.2.1.1.1.0	aristanetworks/aristaOSVersionPostProcessor.groovy	GET		
<input type="checkbox"/>	axsldpRemEntry	ALLIALLIALLIALLIALL	axsldpRemTable	.1.3.6.1.4.1.21839.2.2.1.100.4.1.1		NONE		
<input type="checkbox"/>	axsldpRemPortDesc	ALLIALLIALLIALLIALL	axsldpRemEntry	.1.3.6.1.4.1.21839.2.2.1.100.4.1.1.8		WALK		
<input type="checkbox"/>	axsldpRemRemotePort	ALLIALLIALLIALLIALL	axsldpRemEntry	.1.3.6.1.4.1.21839.2.2.1.100.4.1.1.7		WALK		
<input type="checkbox"/>	axsldpRemSysName	ALLIALLIALLIALLIALL	axsldpRemEntry	.1.3.6.1.4.1.21839.2.2.1.100.4.1.1.9		WALK		
<input type="checkbox"/>	axsldpRemTable	ALLIALLIALLIALLIALL		.1.3.6.1.4.1.21839.2.2.1.100.4.1		NONE		
<input type="checkbox"/>	bgp4PathAttrASPathSegment	ALLIALLIALLIALLIALL	bgp4PathAttrEntry	.1.3.6.1.2.1.15.6.1.5		WALK		
<input type="checkbox"/>	bgp4PathAttrAggregatorAS	ALLIALLIALLIALLIALL	bgp4PathAttrEntry	.1.3.6.1.2.1.15.6.1.10		WALK		
<input type="checkbox"/>	bgp4PathAttrAggregatorAddr	ALLIALLIALLIALLIALL	bgp4PathAttrEntry	.1.3.6.1.2.1.15.6.1.11		WALK		
<input type="checkbox"/>	bgp4PathAttrAtomicAggreg...	ALLIALLIALLIALLIALL	bgp4PathAttrEntry	.1.3.6.1.2.1.15.6.1.9		WALK		
<input type="checkbox"/>	bgp4PathAttrBest	ALLIALLIALLIALLIALL	bgp4PathAttrEntry	.1.3.6.1.2.1.15.6.1.13		WALK		
<input type="checkbox"/>	bgp4PathAttrCalcLocalPref	ALLIALLIALLIALLIALL	bgp4PathAttrEntry	.1.3.6.1.2.1.15.6.1.12		WALK		
<input type="checkbox"/>	bgp4PathAttrEntry	ALLIALLIALLIALLIALL	bgp4PathAttrTable	.1.3.6.1.2.1.15.6.1		NONE		⊗
<input type="checkbox"/>	bgp4PathAttrAddrPrefix	ALLIALLIALLIALLIALL	bgp4PathAttrEntry	.1.3.6.1.2.1.15.6.1.3		WALK		
<input type="checkbox"/>	bgp4PathAttrAddrPrefixLen	ALLIALLIALLIALLIALL	bgp4PathAttrEntry	.1.3.6.1.2.1.15.6.1.2		WALK		
<input type="checkbox"/>	bgp4PathAttrLocalPref	ALLIALLIALLIALLIALL	bgp4PathAttrEntry	.1.3.6.1.2.1.15.6.1.8		WALK		
<input type="checkbox"/>	bgp4PathAttrMultiExtDisc	ALLIALLIALLIALLIALL	bgp4PathAttrEntry	.1.3.6.1.2.1.15.6.1.7		WALK		
<input type="checkbox"/>	bgp4PathAttrNextHop	ALLIALLIALLIALLIALL	bgp4PathAttrEntry	.1.3.6.1.2.1.15.6.1.6		WALK		
<input type="checkbox"/>	bgp4PathAttrOrigin	ALLIALLIALLIALLIALL	bgp4PathAttrEntry	.1.3.6.1.2.1.15.6.1.4		WALK		
<input type="checkbox"/>	bgp4PathAttrPeer	ALLIALLIALLIALLIALL	bgp4PathAttrEntry	.1.3.6.1.2.1.15.6.1.1		WALK		
<input type="checkbox"/>	bgp4PathAttrTable	ALLIALLIALLIALLIALL		.1.3.6.1.2.1.15.6		NONE		⊗
<input type="checkbox"/>	bgp4PathAttrUnknown	ALLIALLIALLIALLIALL	bgp4PathAttrEntry	.1.3.6.1.2.1.15.6.1.14		WALK		
<input type="checkbox"/>	bgpIdentifier	ALLIALLIALLIALLIALL		.1.3.6.1.2.1.15.4.0		GET		
<input type="checkbox"/>	bgpLocalAs	ALLIALLIALLIALLIALL		.1.3.6.1.2.1.15.2.0		GET		

atom > Old-Snmp-Maps

Create Old-Snmp-Map

Entities

Old-Snmp-Map - Bgp4PathAttrBest

Platform *
Platform string for a specific oid
ALLIALLIALLIALLIALL

Name *
Name of the oid
bgp4PathAttrBest

Parent
bgp4PathAttrBest X

Old
Old value for a property
.1.3.6.1.2.1.15.6.1.13

Post-Processor-Prop
Post Processor groovy file name to invoke
post-processor-groovy

Fetch-Type
fetch-type of a oid
GET WALK NOTIFICATION NONE

Key-Old
Mapping related key-oid of a oid in case of WALK
X .1.3.6.1.2.1.15.6.1.2.1.3.6.1.2.1.15.6.1.1.1.3.6.1.2.1.15.6.1.3

Metric-Tag-Old
Old that needs to be considered as tag to the current oid

Snmp-Prop

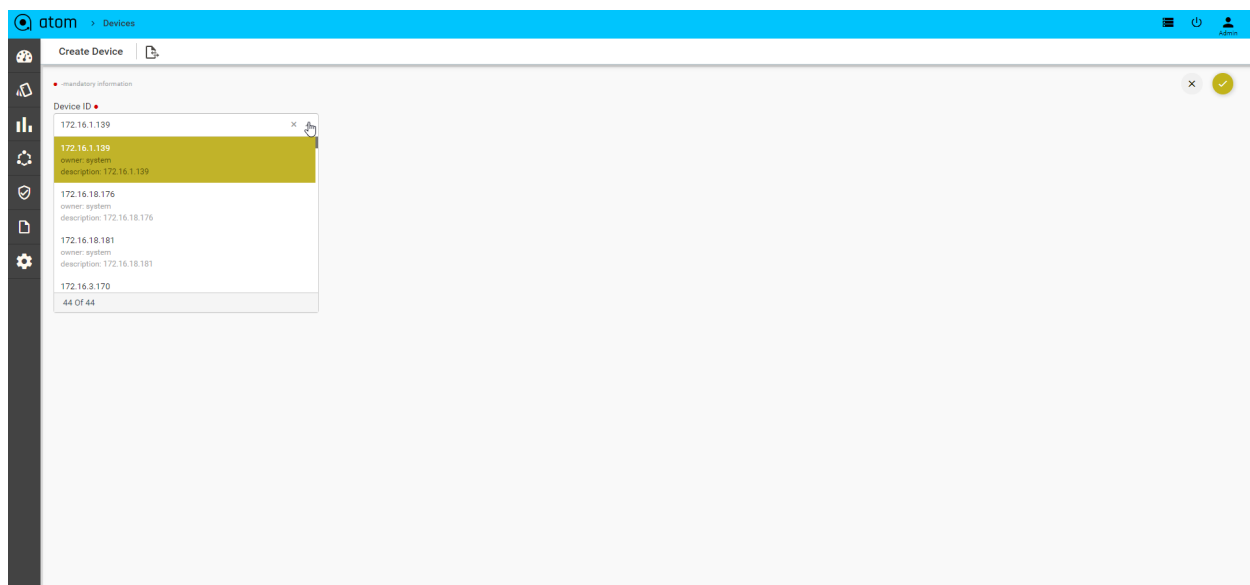
Navigate > Administration > Plugins & Extensions > Package > SNMP > SNMP Metric Metadata > Add

atom > Devices

Metric-Instance-Schema devices

40 Of 40 Search

Device ID	
<input type="checkbox"/> 172.16.1.139	
<input type="checkbox"/> 172.16.18.176	
<input type="checkbox"/> 172.16.18.181	
<input type="checkbox"/> 172.16.3.170	
<input type="checkbox"/> 172.16.3.30	
<input type="checkbox"/> 172.16.3.31	
<input type="checkbox"/> 172.16.3.32	
<input type="checkbox"/> 172.16.3.33	
<input type="checkbox"/> 172.16.3.34	
<input type="checkbox"/> 172.16.3.36	
<input type="checkbox"/> 172.16.3.36 SNMPv3	
<input type="checkbox"/> 172.16.3.38	
<input type="checkbox"/> 172.16.3.39	
<input type="checkbox"/> 172.16.3.40	
<input type="checkbox"/> 172.16.3.41	
<input type="checkbox"/> 172.16.3.44	
<input type="checkbox"/> 172.16.3.46	
<input type="checkbox"/> 172.16.3.47	
<input type="checkbox"/> 172.16.3.48	
<input type="checkbox"/> 172.16.3.49	
<input type="checkbox"/> 172.16.3.51	
<input type="checkbox"/> 172.16.3.53	
<input type="checkbox"/> 172.16.3.58	
<input type="checkbox"/> 172.16.3.71	
<input type="checkbox"/> 172.16.3.72	



Device Support

Device Support view allows users to create a new Vendor, Device Type, Device Family, OS Type, Device Capabilities, Terminal Handling Properties etc.,

Managing Tenants

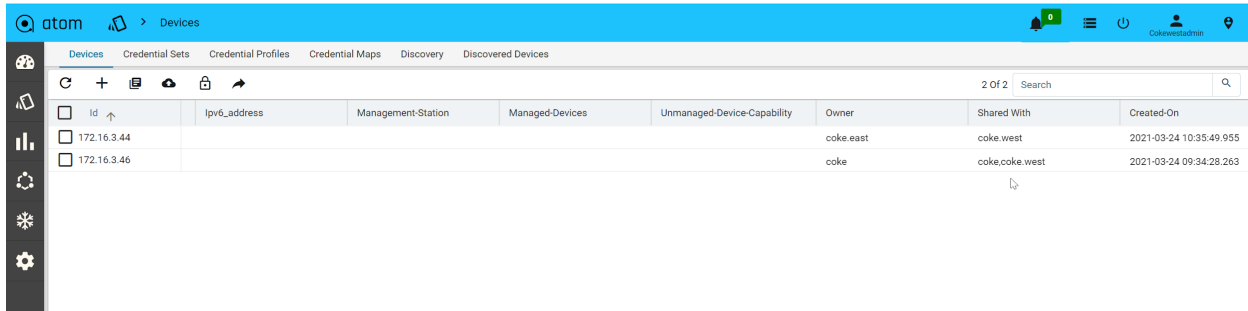
The administrator can make use of ATOM's multi-tenancy capability to share IT resources cost-efficiently and securely by creating Tenants. The Tenants share common infrastructure, yet utilize a defined set of highly secure services, with complete isolation from other tenants. The resources managed by ATOM can be securely shared among multiple applications and tenants (businesses, organizations, etc.) that use the resources of the datacenter.

Overview of Multi Tenancy

- Multi-Tenancy - Supported on ATOM On-Premises & ATOM Silo/Dedicated Deployment on ATOM Cloud.
 - Tenants (Coke, Pepsi etc.), are completely isolated from each other.
- Sub-Tenancy - Supported on All ATOM Deployments - On-Premises, ATOM Cloud Silo/Dedicated and also ATOM Cloud Shared.
 - Data sharing across sub-tenants (Coke.east, Coke.west, Coke.it etc.,) is controlled by Tenant Admin.
 - By Default Data at a higher Level Tenant is Visible to the Sub-Tenants.
 - By Default, Data under a sub-tenant is visible to the Tenant
 - By Default, Data under a sub-tenant is not visible to other Sub-tenant
 - Example - Coke.east owns a resource (credential set or device etc.) and wants to share with sub tenants (Coke.west but not with Coke.it). In this case, ATOM Multi Tenancy Infrastructure provides a facility to share a resource with particular sub-tenants. Upon sharing the resources as required, each individual ATOM User

interface will provide information on Resource sharing as shown below. ***This behaviour will be the same across all the resources in ATOM and will not be discussed specifically across features in the user guide.***

Sample View of Resource being shared from Coke.east to Coke.west



Id	ipv6_address	Management-Station	Managed-Devices	Unmanaged-Device-Capability	Owner	Shared With	Created-On
172.16.3.44					coke.east	coke.west	2021-03-24 10:35:49.955
172.16.3.46					coke	coke,coke.west	2021-03-24 09:34:28.263

Multitenancy feature is enabled in the system as a result of which every object managed by ATOM can be owned by an admin and shared with multiple users (tenants) simultaneously.

Aided by rules and roles that can be created in ATOM, the administrator can either assign or restrict access to the resources (resource pools, sites, locations, IPAM, devices) managed by ATOM.

All the created resources in ATOM are allocated to the system, the default admin user who is the owner of the resources. These resources managed by ATOM are available to all the Tenants in the system. The administrator can now share the system resources with the required tenant or tenants. From then on, all the resources that are created in each Tenant (parent) are available to only the users (child nodes) of a particular parent.

Root Tenant

‘System’ is the root tenant. Every other tenant is a child or in the child hierarchy of this root. There may be few objects which are kept ‘private’ to the system, meaning, those are not ‘shared’ to child tenants.

Top Level Tenants

Top Level Tenants (referred ‘tenants’ for simplicity) are the immediate children of system nodes.

For ex;

System

Company-1

Company-2

Company-3

In the above example, company-1, 2, 3 are top level tenants.

Simple Multi Tenancy

ATOM supports Sub tenancy where a tenant can subdivide their resources into a sub hierarchy. But, when there are no sub tenants in the deployment, it is referred to as 'Simple Multi Tenancy'.

For ex;

System

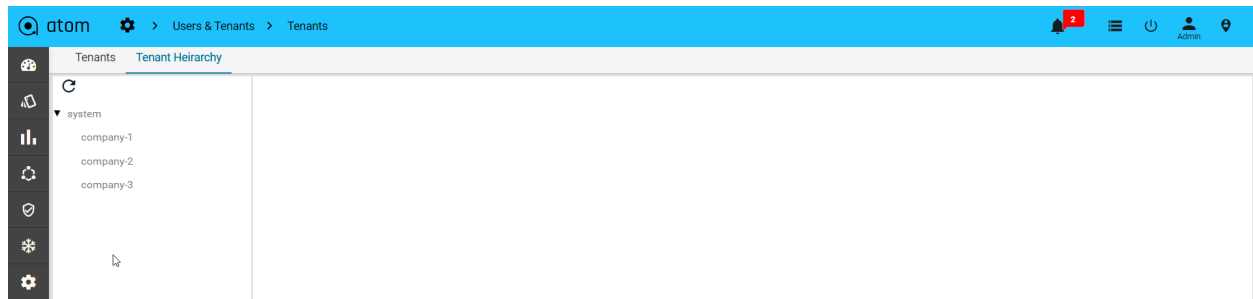
Company-1

Company-2

Company-3

In On-Prem deployment, it is up to the customer how they treat the root tenant.

If they don't create any child tenants to the system, then, system and customer are synonymous.



Hierarchical Multi Tenancy

Sub Tenant

A sub tenant is a child [directly or indirectly] of a Top Level [Not root] tenant.

System

Company-1

North

South

Campus-1

Company-2

Company-3

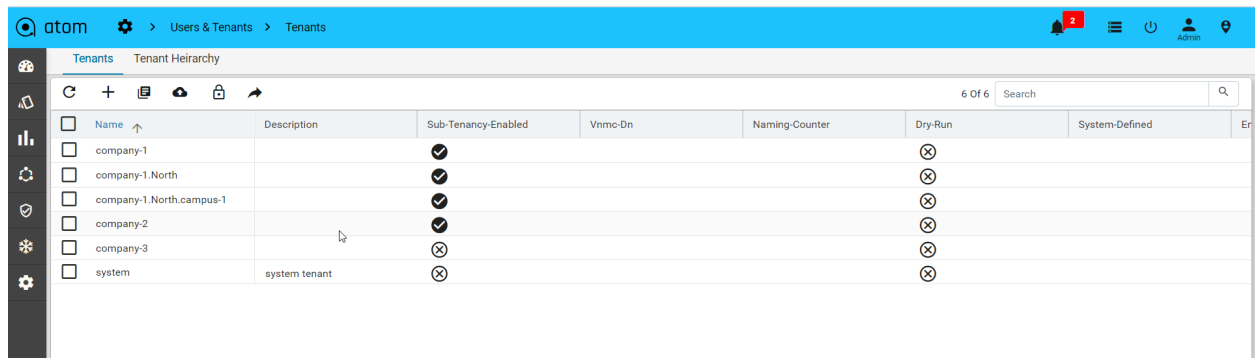
In the above example,

'System' is the root tenant

Company-1, Company-2, Company-3 are top level tenants

Company-1.north, Company-1.south are sub tenants of Company-1.

Company-1.south.campus-1 is a sub tenant of acme.south



The screenshot shows the ATOM web interface for managing users and tenants. The breadcrumb navigation is 'Users & Tenants > Tenants'. The interface includes a sidebar with various icons and a main content area with a table of tenants. The table has columns for Name, Description, Sub-Tenancy-Enabled, Vnmc-Dn, Naming-Counter, Dry-Run, System-Defined, and Error. The 'system' tenant is highlighted as the 'system tenant'.

Name	Description	Sub-Tenancy-Enabled	Vnmc-Dn	Naming-Counter	Dry-Run	System-Defined	Error
company-1		✓			⊗		
company-1.North		✓			⊗		
company-1.North.campus-1		✓			⊗		
company-2		✓			⊗		
company-3		⊗			⊗		
system	system tenant	⊗			⊗		

System users

System is the root tenant. System users are those, whose User.owner = 'system'.

Invite User(s)



●-mandatory information

External User



Add Users●



Username●

user

Email Address●

user@anutanetworks.com



Assign Roles to User

Select one or more Roles to assign to this user(s)

User Roles



Assign Groups to User

Select one or more User Groups that this user(s) belong(s) to

User Groups



Enable Developer Mode

Enable Developer Mode for this user

☐

Can-Read-Data-Of

Select the sub-set of tenants that are available to this u...

x system



Can-Change-Data-Of

Select the sub-set of tenants this user(s) is authorized t...

x system



Owner●

system



Shared-With

x system



Send Invites

Cancel

Tenant Users

Tenant users are customer users owned by individual tenants. User.owner != 'system'

Invite User(s)



• mandatory information

External User



Add Users •



Username •

companyadmin

Email Address •

companyadmin@anutanetworks.co



Assign Roles to User

Select one or more Roles to assign to this user(s)

User Roles



Assign Groups to User

Select one or more User Groups that this user(s) belong(s) to

User Groups



Enable Developer Mode

Enable Developer Mode for this user

☐

Can-Read-Data-Of

Select the sub-set of tenants that are available to this u...

x company-1



Can-Change-Data-Of

Select the sub-set of tenants this user(s) is authorized t...

x company-1



Owner •

company-1



Shared-With

x company-1



Send Invites

Cancel

Owner:

Owner is a tenant. And, we use 'tenant-id' to identify a tenant.

Tenant-id uses fully qualified names separated with dots, such as, Company-1.south.campus-1.

Multi Tenancy is all about keeping data private to a tenant. This means, data identified by a key can have one copy for each tenant. Suppose, 2 tenants want to bring in the same device-1 ? That counts to 2 instances with the same key. Clearly object id by itself is not sufficient. Hence, objects are identified by their id and 'owner'. Object key is formed by object id and owner.

Shared-with:

A resource can be shared with multiple tenants or kept private to the owner.

Sharing of resources applies only when a resource owned by one tenant is to be used by another tenant.

For example, a device owned by tenant-1 is used by a 'network service' created by a tenant-2.

Sharing across tenants is not supported. but, within a tenant sub hierarchy is supported.

For example, no data is ever shared among the 3 companies of the following hierarchy
System

Company-1

Company-2

Company-3

But there is, down the hierarchy sharing allowed. Such as, 'system' resources are shared to all the three [and sub tenants, if exist].

If Resource is shared-with system then it is private to system

If Resource is shared-with system.* then it is shared with all the sub tenants.

Concept Of Visibility And Usability

Visibility is the same as 'Readability'; Whether a resource is visible to a user.

Usability of a resource is with respect to another resource and it is about whether a resource-1 can be referred (in a relation for example) by another resource.

For example,

device-1.credential-set = 'cred-1'

#	resource	Owner	
1	device-1	Univ.Engg	
2	cred-1	Univ.Phy	

Device-1 wants to use 'cred-1' in a 'device.credential-set' relation.

Currently, to make cred-1 available (visible, usable) to resources of Univ.Engg, you have to share it with that tenant.

#	resource	Owner	shared-with
1	device-1	Univ.Engg	
2	cred-1	Univ.Phy	Univ.Engg

Shared With Variations

System

Company-1

Campus-1

Department-1

Department-2

Campus-2

Owner	Resource	Shared With	Details
Company-1	R1	Company-1	R1 becomes a private object
Company-1	R1	Company-1, system	A Tenant Resource shared with system [there are a few scenarios where this is useful]
Campus-2	R1	company1.Campus-1, company1.Campus-2	Sharing with other tenants [in the sub hierarchy]
Campus-1	R1	Campus-1.*	Using wildcards in sharing-with. R1 will be shared with all sub tenants. Since sub

			tenants can be added or removed during the life cycle of a deployment, sharing is spread to all the sub tenants available at the time of invocation.
--	--	--	--

User.Owner:

User object has an 'owner' property, just like any other resource.

But, there is a special meaning to 'user.owner'.

Users need to be authenticated in the system.

Authentication is done with an 'Identity Provider', such as an LDAP.

Identity Providers are associated with tenants.

So, a user is authenticated against the provider traced via User.owner

User.can-read-data-of And User.can-change-data-of:

A user could be created at a higher level (user.owner) but to limit the user to a subset of tenants there are two properties.

'can-read-data-of' controls which tenant data a user can read.

'can-change-data-of' controls which tenant data a user can change.

When a top level tenant does not have sub-tenants, user.can-read-data-of will be fixed to the tenant. User.can-change-data-of can be used to disable writes [by omitting a value]. If decided to allow writes , the value will be fixed to the tenant.

Invite User(s)



●-mandatory information

External User



Add Users●



Username●

user

Email Address●

user@anutanetworks.com



Assign Roles to User

Select one or more Roles to assign to this user(s)

User Roles



Assign Groups to User

Select one or more User Groups that this user(s) belong(s) to

User Groups



Enable Developer Mode

Enable Developer Mode for this user

☐

Can-Read-Data-Of

Select the sub-set of tenants that are available to this u...

x system



Can-Change-Data-Of

Select the sub-set of tenants this user(s) is authorized t...

x system



Owner●

system



Shared-With

x system



Send Invites

Cancel

NOTE: Except Monitoring and Alerts all other components are MT Enabled.

Creating Tenants

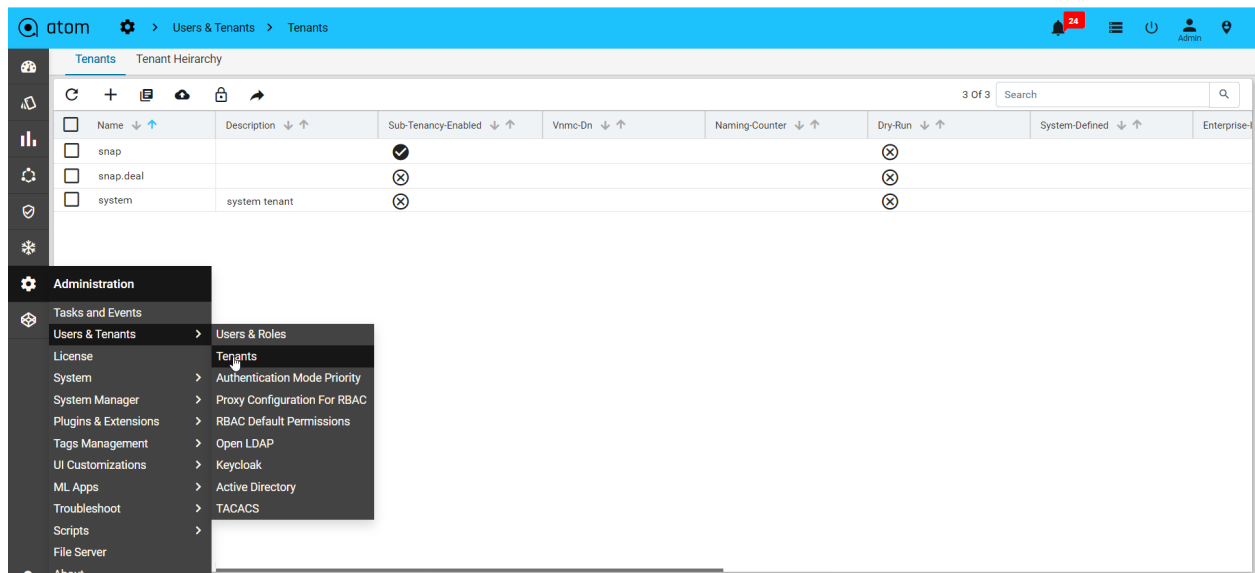
Before instantiating a service, there should be at least a single Tenant created in ATOM.

1. Navigate to **Administration > Tenants**
2. Select the Tenants folder > click **Add**
3. In the **Create Tenant** screen, enter the following:
 - **Name:** Enter an alphanumeric string of not more than 32 characters
 - **Description:** Enter a description for the Tenant
 - **TenantId:** Enter a unique identifier for the Tenant
 - **Dry Run:** This option does not allow ATOM to send the configurations to the devices while creating services. By default, this option is unselected.
 - Select the checkbox to allow ATOM to push the commands on to the devices.

Onboarding Tenants through Keycloak scripts

Before creating a tenant admin, We need to create a tenant manually in system admin(Ex:Coke).

Navigate->Administration->Users and tenants->Tenants:



Creating a tenant in Atom UI.

SSH into Master ip of ATOM to run onboard_tenant_lifecycle.py.

Configure like below in master ip.

```
sudo python onboard_tenant_lifecycle.py -r <tenant name> -c atom -tenant True -realm
True -vip <tenant ip>
```

EX : sudo python onboard_tenant_lifecycle.py -r coke -c atom -tenant True -realm True
-vip 172.16.5.65

Note:: tenant ip which is not pingable.

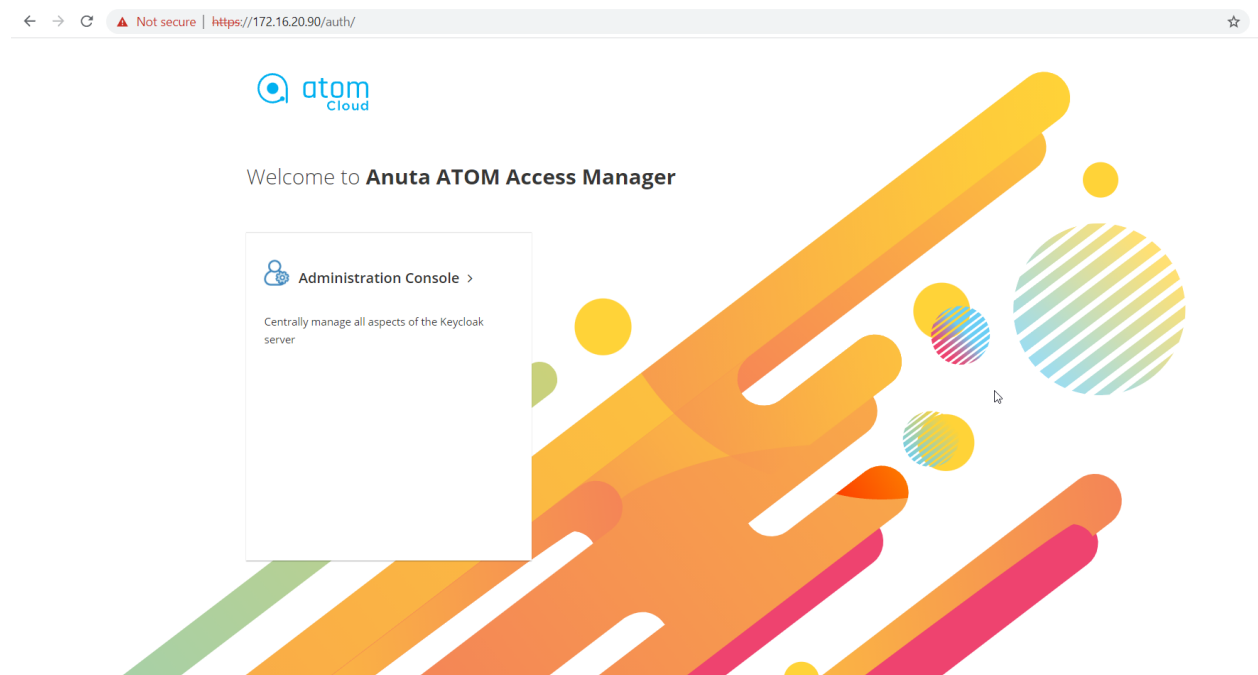
```
11.1.0.45545 etcd-download-test systemd-private-4ac039fa94dd47148b374c783ba70519-chronydw.service-r0y0m7 upgrade-11.1.45545.tgz
atom.lock k8s.conf terraform_1791523002.sh vmware-root_718-2957190230
[atom@raveiteja-master7107460 tmp]$ cd 11.1.0.45545
[atom@raveiteja-master7107460 11.1.0.45545]$ ls
atom-deployment upgrade-11.1.45545.tgz
[atom@raveiteja-master7107460 11.1.0.45545]$ cd atom-deployment/
[atom@raveiteja-master7107460 atom-deployment]$ ls
11.1.45545
agent_deployment.py aws_cluster_creation.py etcd-backup sanity
atom log aws_delete.py etcdctl_install.sh scale_values.yaml
atom-upgrade-2021-10-10-23-16-01.log backup.py functional_minimal.yaml scripts
atom-upgrade-2021-10-10-23-16-01.log.1 backup.pyc ha_cluster.py terraform-aws
atom-upgrade-2021-10-10-23-16-01.log.10 bootstrap_slaves.py ha_cluster_RHEL.py UpgradeDefinition.xml
atom-upgrade-2021-10-10-23-16-01.log.2 buildnumber.txt helm_validation.sh upgrade_handler.py
atom-upgrade-2021-10-10-23-16-01.log.3 centos_run_on_master.py _init_.py verify_deploy.py
atom-upgrade-2021-10-10-23-16-01.log.4 cleanup_atom_deployment.py ops_seed_cluster_automation_scripts workflow_scale_5k.yaml
atom-upgrade-2021-10-10-23-16-01.log.5 copingvaluesfile.py precheck_resources.py wrapper_minimal_centos.properties
atom-upgrade-2021-10-10-23-16-01.log.6 create_zip.py prepare_docker_engine.py wrapper_properties
atom-upgrade-2021-10-10-23-16-01.log.7 deploy_atom.py pv_snapshot_local wrapper_resilient_centos.properties
atom-upgrade-2021-10-10-23-16-01.log.8 deploy_k8s.py README.md wrapper_scale_centos.properties
atom-upgrade-2021-10-10-23-16-01.log.9 dummy.txt remove_unnecessary_files_for_custom_zip.sh zipignore.txt
aws_automation.py eks_cluster.sh run_helm_cluster_test.sh
[atom@raveiteja-master7107460 atom-deployment]$ cd scripts/
[atom@raveiteja-master7107460 scripts]$ ls
00-minio-deployment.yaml customer_setup_installation_sanity.py getlogsfrompod.sh keycloak remote-sanity-deployment.py
addUserstoSSO.py delete_and_create_volumes.py get_urls.sh kubevip-controller.yaml run_config.py
atom_sso_sanity_testcases.py delete_aws_loadbalancer.py heapster.yaml kubevip-rbac.yaml scale_ncx.py
aws_route53_tenant_onboarding.sh delete_aws_route53_cname.py images.yaml kubevip.yaml scp_state_file_to_remote.py
bootstrap-ha.sh deletelinstor.sh _init_.py kubevip.yaml script_create.sh
bootstrap-master.sh delete_pv_data.py install-calico.sh local_oauth.sh script_delete.sh
bootstrap-worker.sh delete_unused_aws_volumes.py install-docker.sh modify_extra_args.py setup-proxy.sh
broker-host.sh delete_volume.py install-helm.sh node_passwd_update ssh_passwordless_setup.py
buildnumber.txt delete_volume.py install-keepalived.sh onboard_tenant_lifecycle.py teardown-pv-pvc.sh
calico.yaml deploylinstor.sh install-kube.sh patch_security_fixes.sh tiller.yaml
cli.py deploylinstor.sh install-kubevip-cloud-controller.sh pod_failure_script.py update_elasticsearch_indices_disk_allocation.py
cluster-info deployk8s.sh install_ntp_centos.py precheck.py validate_webconfig_alertmanager.sh
config.yaml display_pvc.py docker-registry.py precheck.pyc vault
create_license_8.8.py elasticsearch-fix.sh pre-requisites-atom.sh velero_backup_and_restore.py
create_license.py fetch_build_number.py install-registry.sh velero_installation.py
create_loop_device.sh extractconfig.yaml install-terraform.sh velero_local_backup_and_restore.py
create-pv.sh fetch_build_number.py k8s-cluster-addons register_node.py wrapper_validation.py
create_volume.py gencert.sh k8s-master.sh
[atom@raveiteja-master7107460 scripts]$ sudo python onboard_tenant_lifecycle.py -r coke -c atom -tenant True -realm True -vip 172.16.20.95
```

```
[atom@raviteja-master7107460 scripts]$ sudo python onboard_tenant_lifecycle.py -r coke -c atom -tenant True -realm True -vip 172.16.20.95
-----System Specific Details -----
SYSTEM HAPROXY URL is : 172.16.20.90
SYSTEM CLIENT SECRET is : 46f5cf1e-a455-4399-8859-65f362abc946
-----Tenant Specific Details -----
TENANT HAPROXY URL is : https://172.16.20.95
TENANT CLIENT SECRET is : 949a1857-d3a1-44c0-8e6c-19ea7abddaf6
[atom@raviteja-master7107460 scripts]$ kubectl get pods -n coke
NAME                                READY   STATUS    RESTARTS   AGE
haproxy-ingress-gw-854688bb77-8z2np 2/2     Running   0           111s
oauth2-proxy-695cbd8899-4h8d7        1/1     Running   0           42s
[atom@raviteja-master7107460 scripts]$
```

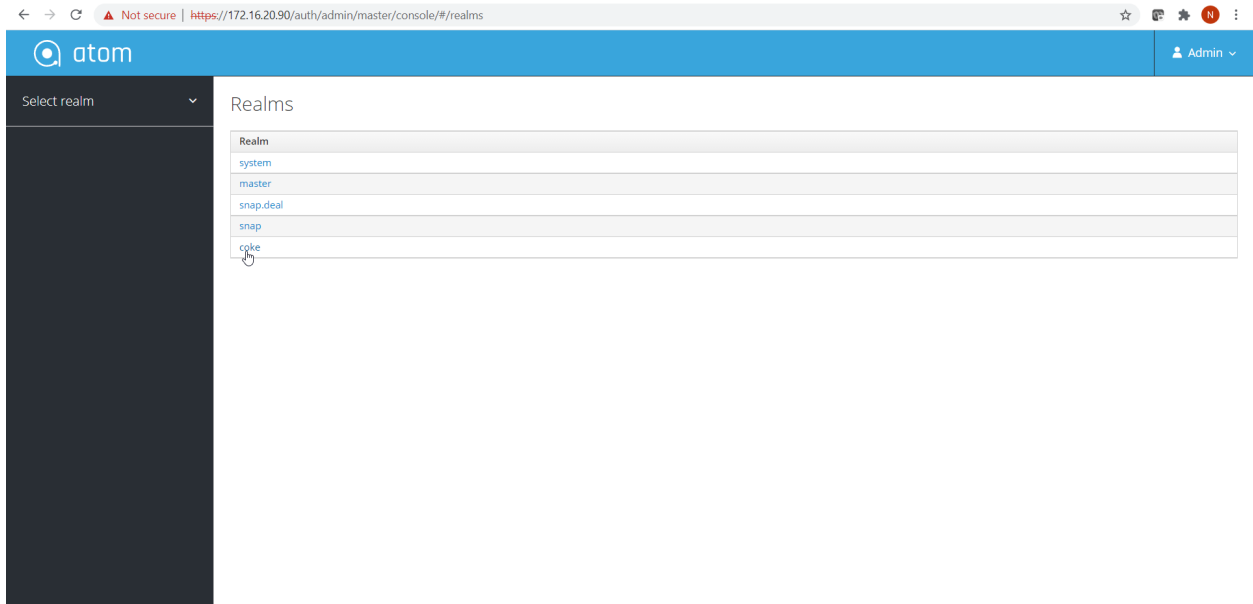
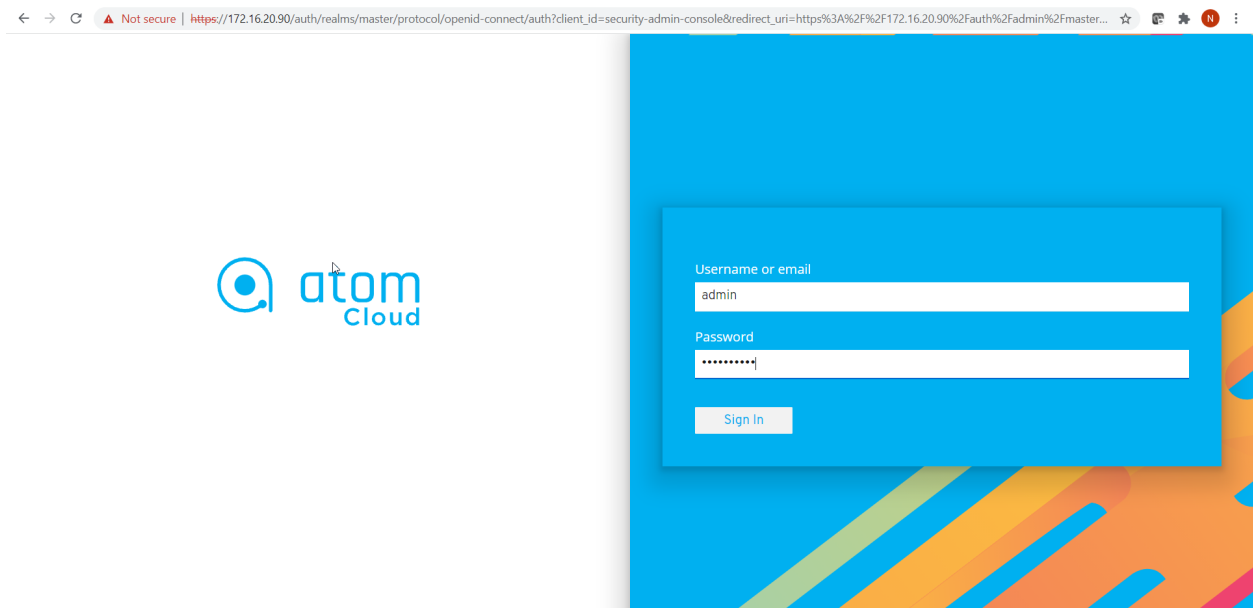
After running the commands, need to copy the Tenant login URL from `TENANT HAPROXY URL` : `https://172.16.20.95`.

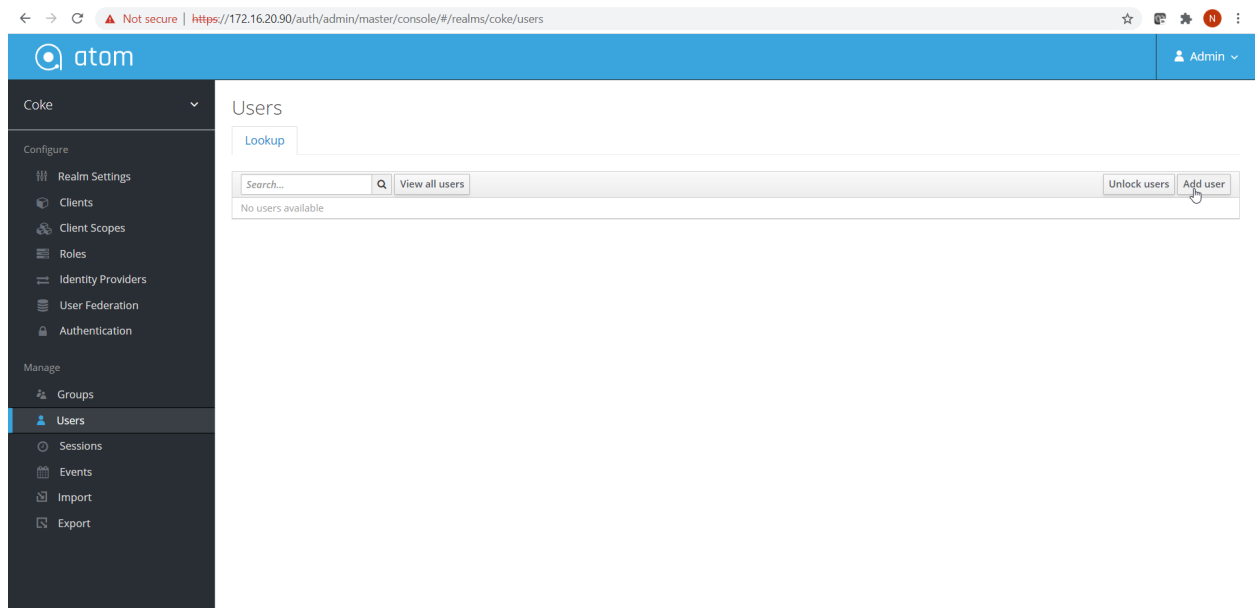
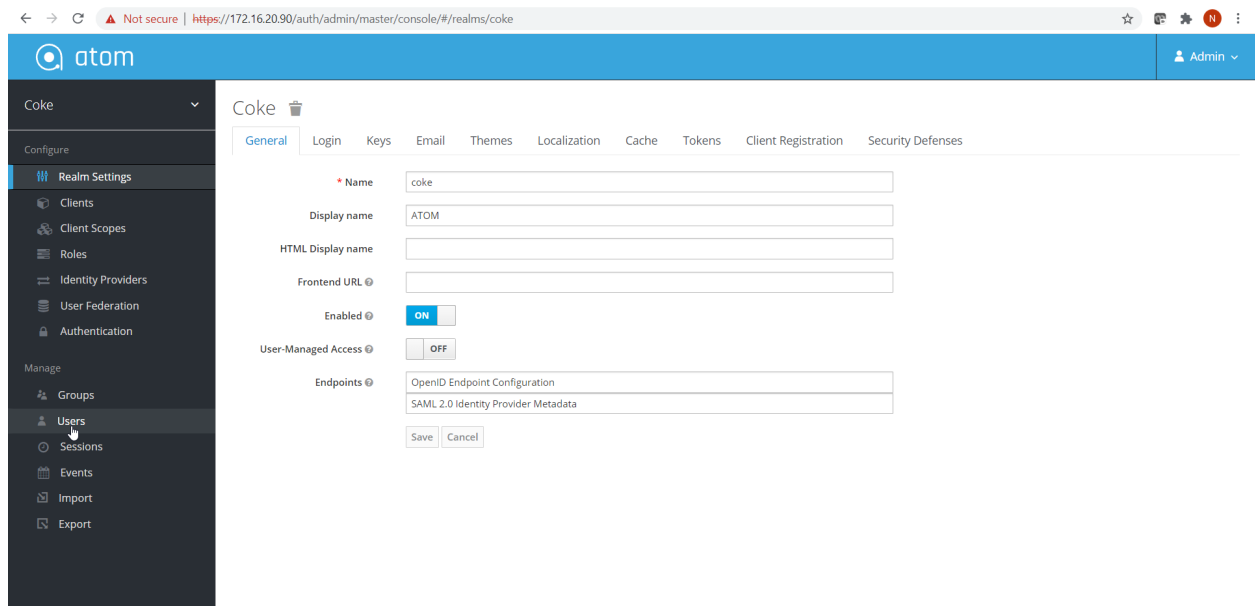
After getting the tenant haproxy url , need to connect the keycloak as:

`https://172.16.20.90/auth`



Login to keycloak using user/password : **admin/Secret@123**





Create tenant admin

atom

Admin

Coke

Users > Add user

Add user

ID

Created At

Username *

cokeadmin

Email

cokeadmin@anutanetworks.com

First Name

Last Name

User Enabled

ON

Email Verified

OFF

Groups

Select existing group...

No group selected

Required User Actions

Select an action...

Save Cancel

Set the password

atom

Admin

Coke

Users > cokeadmin

Success! The password has been set.

Cokeadmin

Details Attributes Credentials Role Mappings Groups Consents Sessions

Manage Credentials

Position	Type	User Label	Data	Actions
----------	------	------------	------	---------

Set Password

Password

Password Confirmation

Temporary

OFF

Set Password

Credential Reset

Reset Actions

Select an action...

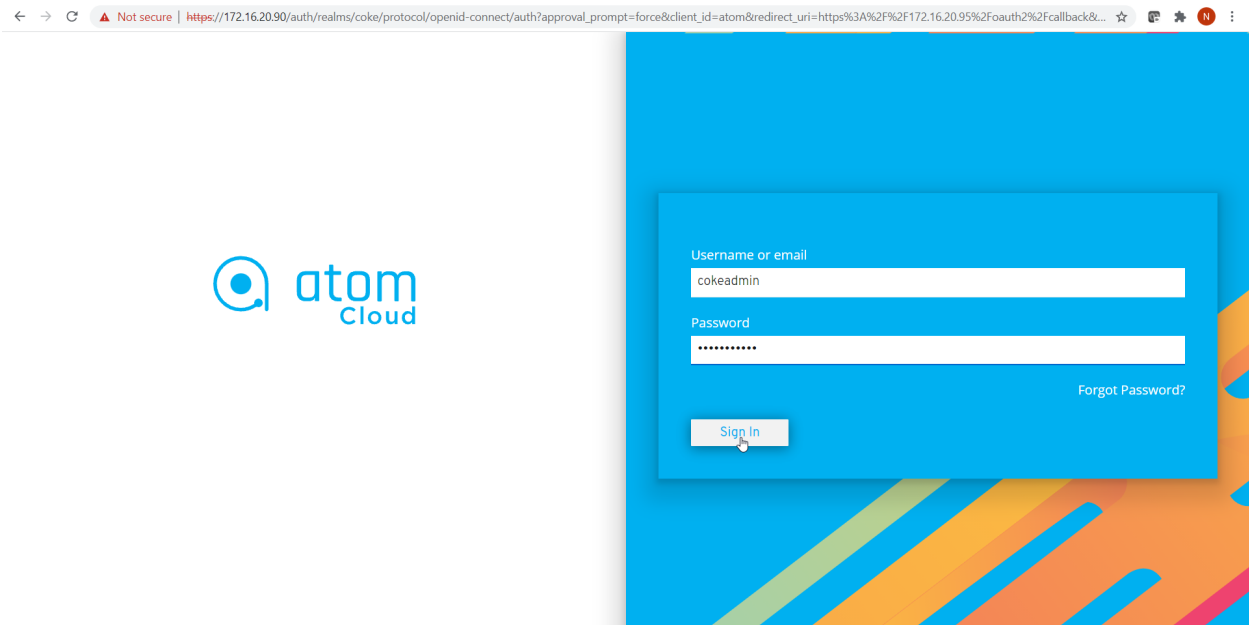
Expires In

12 Hours

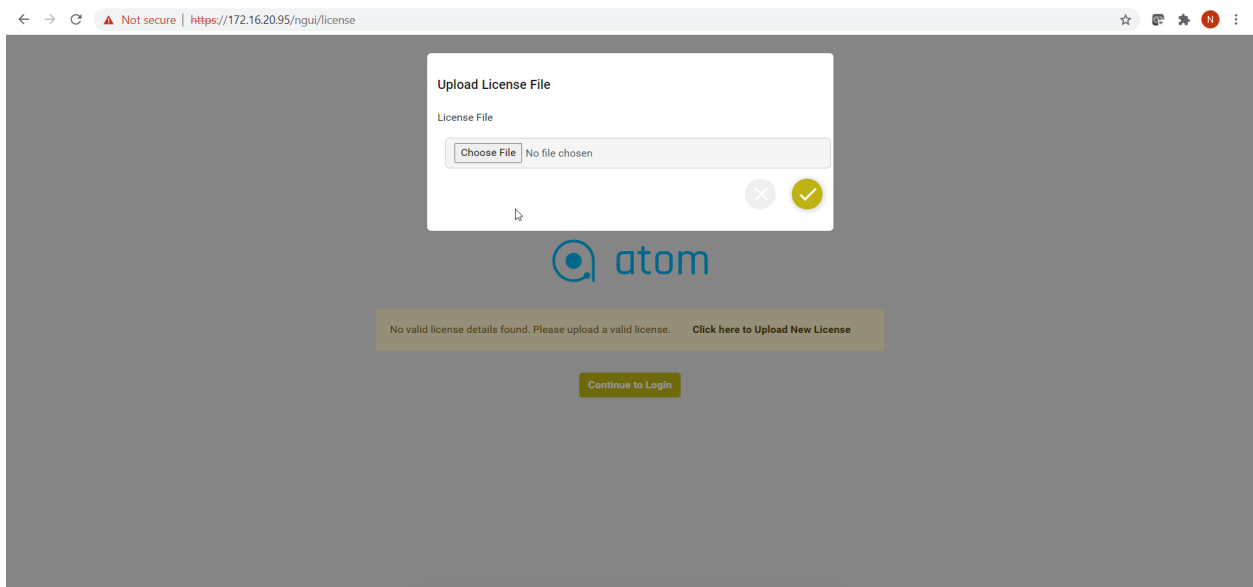
Reset Actions Email

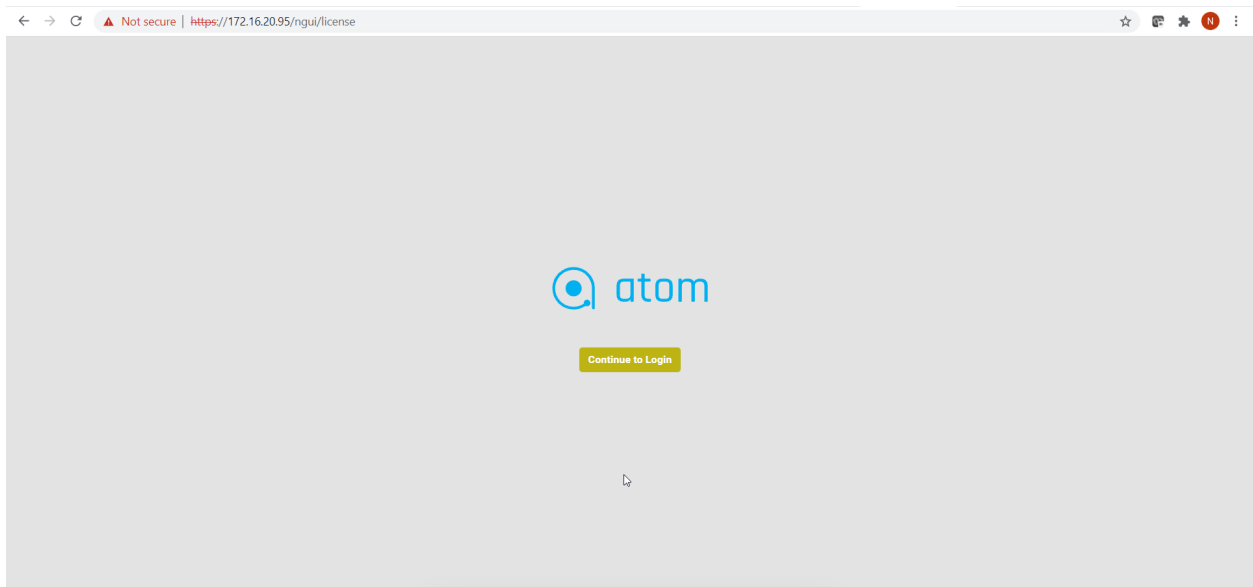
Send email

Login with tenant admin user

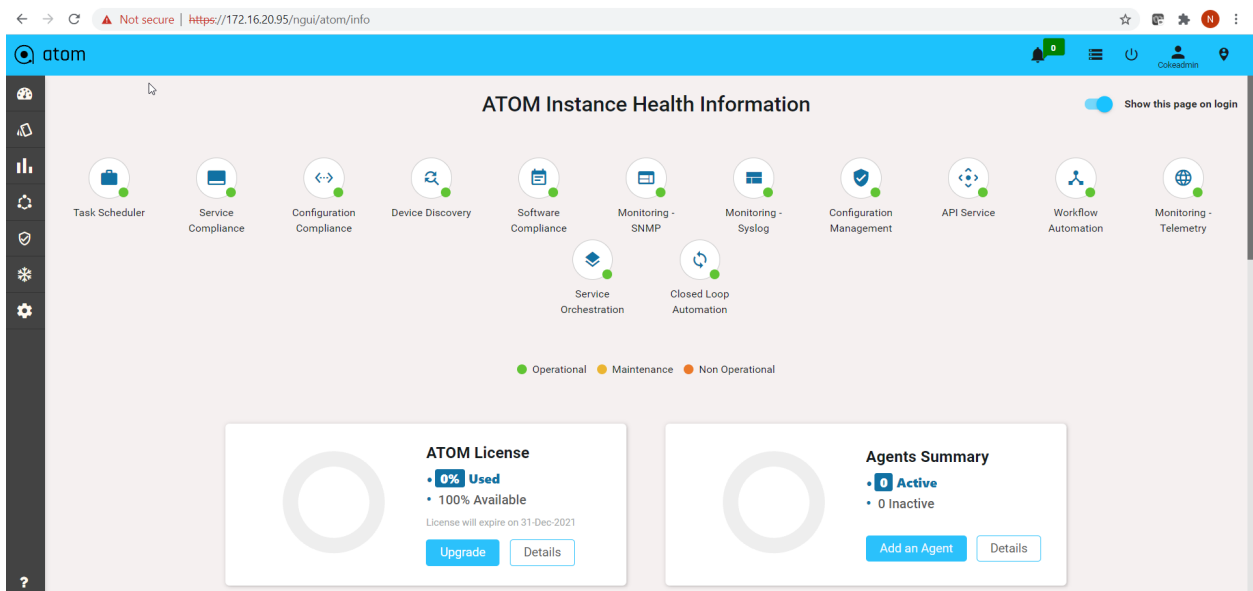


Upload the tenant license in atom





Check the tenant admin UI.

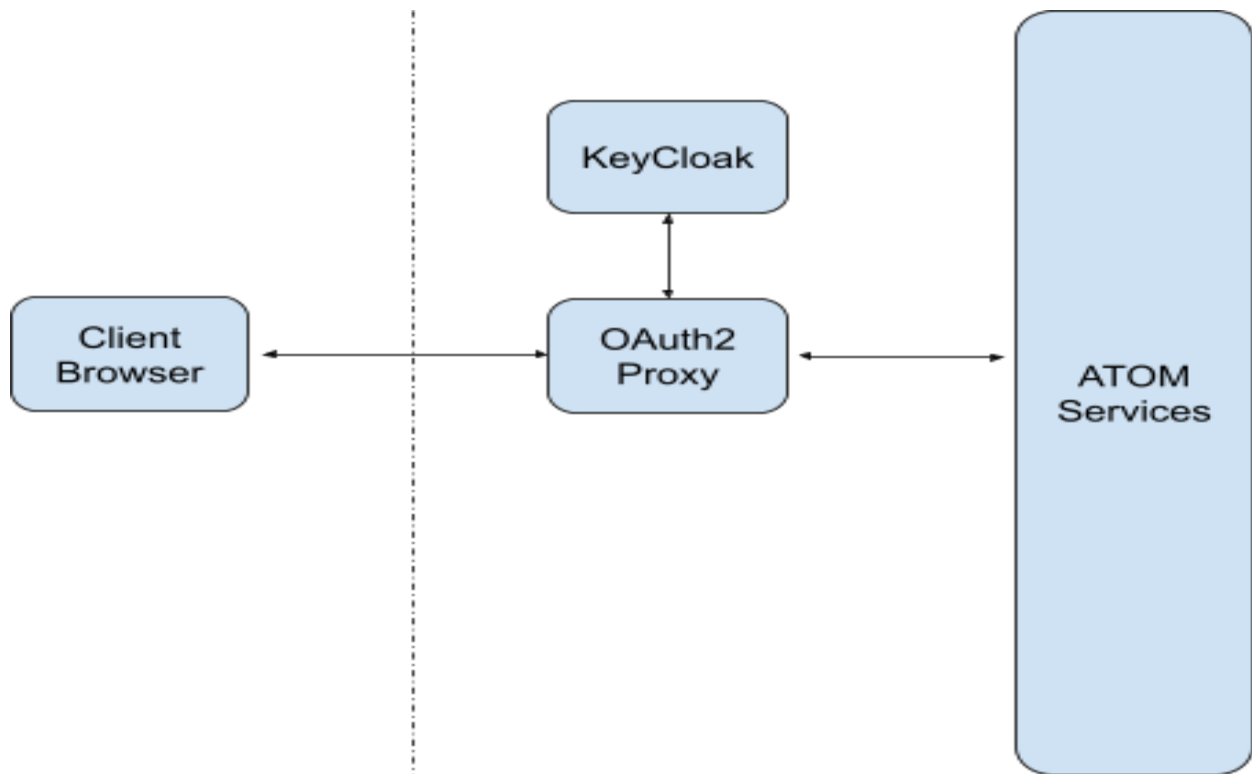


User Management

The control of users and groups is a core element of ATOM system administration. Users can also be grouped (based on the function) to have read permissions, write permissions, execute permissions or any combination of read/write/execute permissions for files owned by that group.

Managing Users in ATOM not only covers creating users but also configuring or assigning the privileges for each user or similar group of users to perform tasks in ATOM. Apart from creating the access or deny permissions in ATOM locally, you can import existing LDAP or AD users into ATOM and extend the necessary permissions to them too.

User management and Authentication works in ATOM as shown in the diagram below. The user information is saved in KeyCloak. OAuth Proxy acts as a gatekeeper checking all the incoming requests and ensuring the requests are coming from an authenticated client. If the proxy sees an non-authentic call, it redirects the request to a login screen served by the identity provider, which is KeyCloak.



User management in ATOM includes the following:

- "Roles"
- "Creating Authentication Mode Priority"
- "Managing Users"
- "Configuring Access Control"
- "Managing OpenLDAP Users"
- "Managing Active Directory Users"
- "Managing TACACS Users"

Roles

A set of system-defined permissions are grouped into roles and are available in ATOM by default. These roles can be assigned to a user during the creation of users in ATOM.

- **ROLE_SYSTEM_ADMIN:** Administrator of the root tenant ('system'). Every other tenant is either a direct child of the system or a sub tenant (descendent) of a top level tenant. System admin is given permission to onboard tenants and manage 'system' tenant resources.
- **ROLE_TENANT_ADMIN:** Is the equivalent of ROLE_SYSTEM_ADMIN (gets blanket permissions on all resources) but limited to resources of the specific tenant ('user.owner') .
- **ROLE_USER_ADMIN:** Gives all access to user mgmt objects (users, Groups, rbac rules [rule list and everything down] etc), but limited to user.owner.
- **ROLE_WORKFLOW_ADMIN:** Allows a user all permissions on all workflow resources.

This avoids having to create individual workflow permissions.

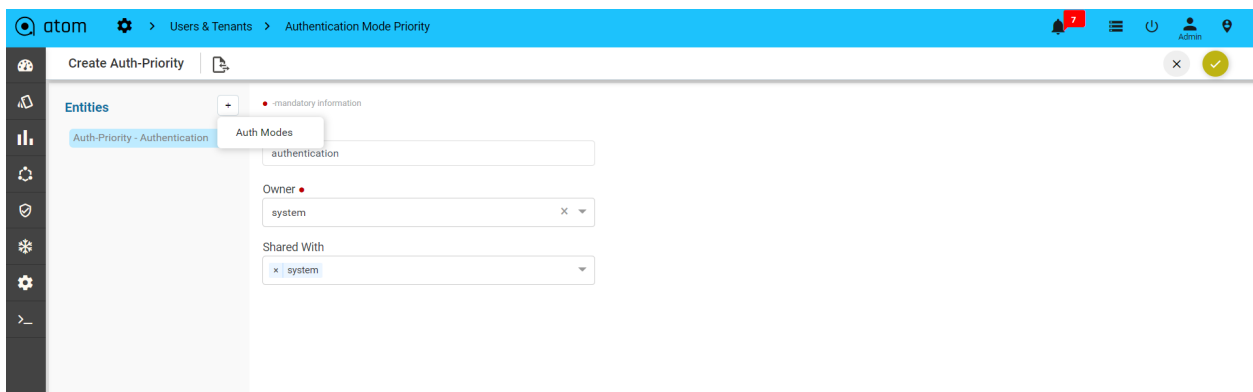
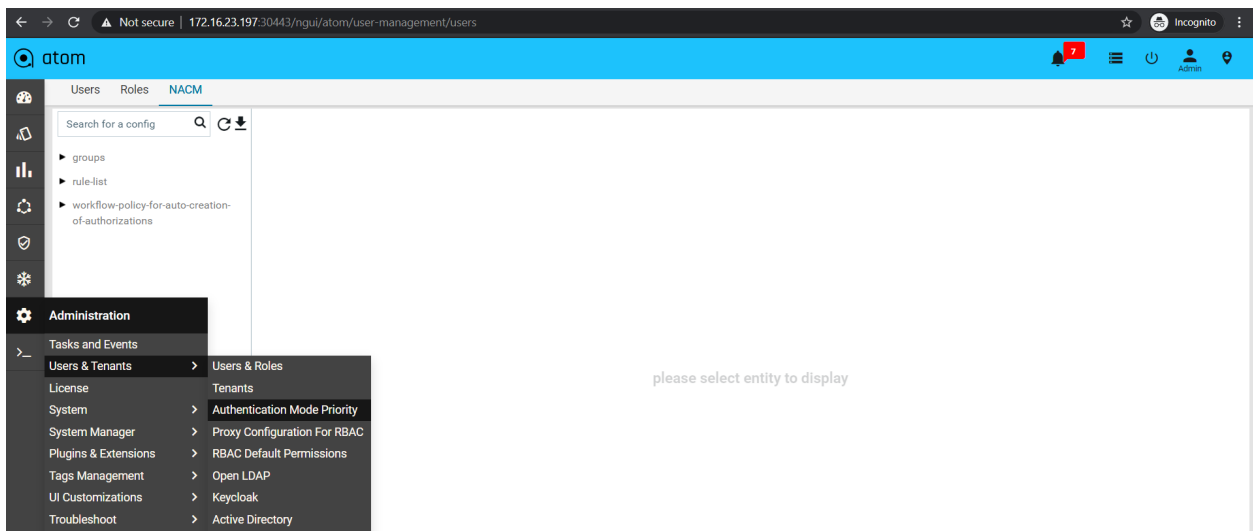
Note that other permissions (rpc, data node etc) are still needed to be given explicitly, because this role only covers 'Workflow resources' only.

Creating Authentication Mode Priority

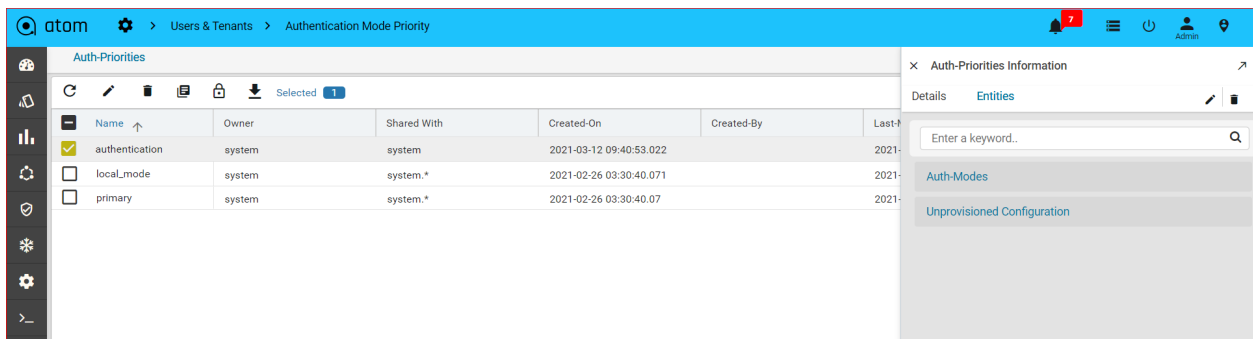
Starting from the 5.8.7 release, the admin can set the priority of the authentication modes in ATOM. By setting the priority of the authentication modes, the admin can enable the login failover to another authentication mode, if the first authentication mode (as arranged in the order of priority) fails. ATOM fails to the local authentication mode, if all the authentication modes as defined in the priority list fail.

To create the authentication mode priority in ATOM, do the following:

1. Navigate to **Administration > Users & Tenants> Authentication Mode Priority** in the left pane.
2. In the right pane click **Add Auth Priority** and in the **Create Auth Priority** screen, click **Entities > auth-modes**
3. Enter values in the fields as described below:
 - i. **Priority:** Set the priority for the authentication mode. (1 is the highest priority)
 - ii. **Authentication Mode** - Select the authentication mode from the available authentication modes (TACACS, OpenLDAP, Active Directory and Local)



After setting the priority for each of the authentication modes, you can view the list as shown below:



ATOM | Authentication Mode Priority

Not secure | 172.16.23.197:30443/ngui/atom/page/layout?schemaPath=/controlleraccess-control/auth-priorities

atom > Users & Tenants > Authentication Mode Priority

Auth-Priorities

Selected 1

Name	Owner	Shared With	Created-On	Created-By	Last-Modified
<input checked="" type="checkbox"/> authentication	system	system	2021-03-12 09:40:53.022		2021-03-12 09:40:53.022
<input type="checkbox"/> local_mode	system	system.*	2021-02-26 03:30:40.071		2021-02-26 03:30:40.071
<input type="checkbox"/> primary	system	system.*	2021-02-26 03:30:40.07		2021-02-26 03:30:40.07

Auth-Priorities Information

Details Entities

Enter a keyword..

Auth-Modes

Unprovisioned Configuration

atom > Users & Tenants > Authentication Mode Priority

Auth-Modes

Search

Priority	Authentication-Mode	Owner	Shared With	Created-On	Created-By	Last-Modified-On	Last-Modified-By
----------	---------------------	-------	-------------	------------	------------	------------------	------------------

atom > Users & Tenants > Authentication Mode Priority

Create Auth-Modes

mandatory information

Priority

1

Authentication-Mode

LOCAL OPENLDAP ACTIVE_DIRECTORY TACACS

Owner

system

Shared With

system

Priority	Authentication-Mode	Owner	Shared With	Created-On	Created-By	Last-Modified-On	Last-Modified-By
1	OPENLDAP	system	system	2021-03-12 09:46:39.995		2021-03-12 09:46:39.995	
2	ACTIVE_DIRECTORY	system	system	2021-03-12 09:47:21.283		2021-03-12 09:47:21.283	

The authentication mode priority thus set can be assigned to a user at the time of creation of the user in ATOM

Managing Users

An Administrator can add local users to ATOM and configure their email accounts to receive notifications from ATOM. Apart from local users, ATOM lets its customers integrate their central authentication servers to streamline the user login process and automate administrative tasks such as user creation and role assignment. User data is synchronized from customer authentication servers into ATOM.

Prerequisite of Adding New User:

Before adding user need to follow the below required steps

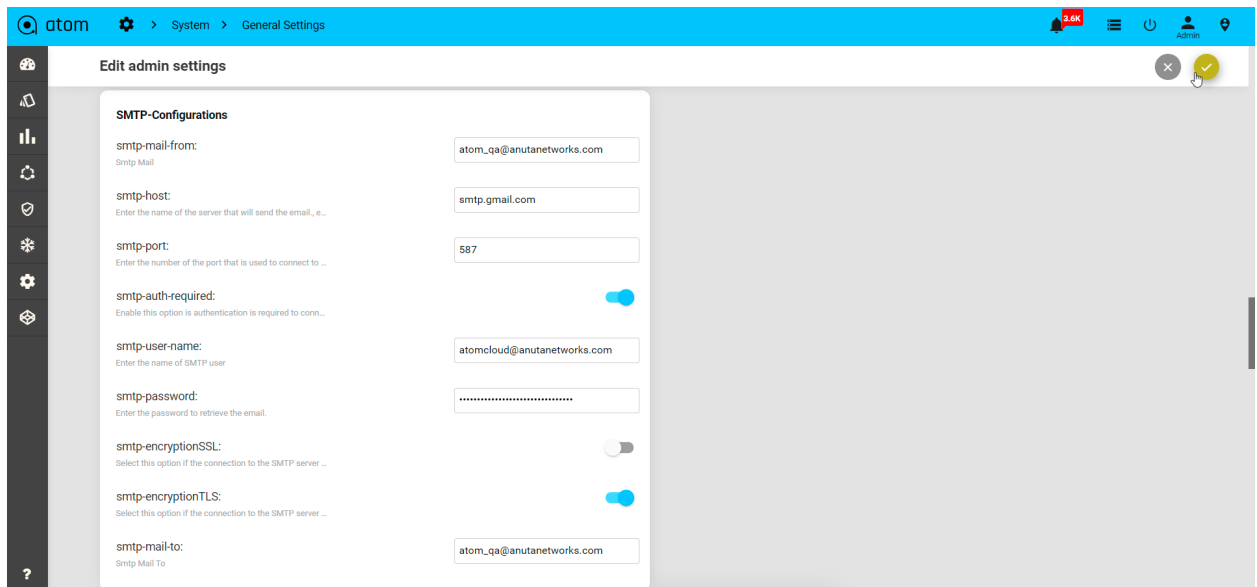
Navigate to Administration->System->General setting->Edit->Smtp configuration:

You have to configure an email server to send email notifications to the ATOM users.

SMTP Mail From: You can set up an external SMTP email server to send email notifications to the ATOM users. To do so, enter values in the fields described below:

1. **SMTP-MAIL-FROM** : Sample value like atomdev@anutanetworks.com

2. **SMTP Host:** Sample value like smtp.gmail.com
3. **SMTP Port:** Example 587
4. **SMTP Auth Required:** Enable this option if authentication is required to connect to the SMTP Host
5. **SMTP User Name:** Example atomcloud@anutanetworks.com
6. **SMTP Password:** xxxxxx
7. **SMTP Encryption SSL:** Disable/Enable this option according to your SMTP.
8. **SMTP Encryption TLS:** Select this option if the connection to the SMTP server should use TLS as the authentication method
9. **SMTP-MAIL-TO:** Example atom_qa@anutanetworks.com

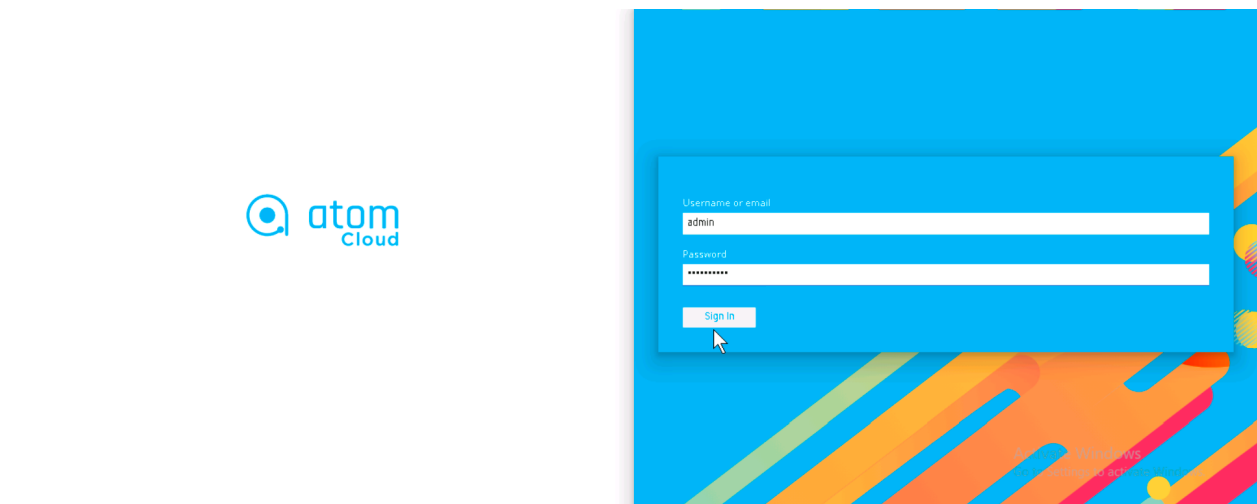


In keycloak admin has to update the smtp settings following the below steps.

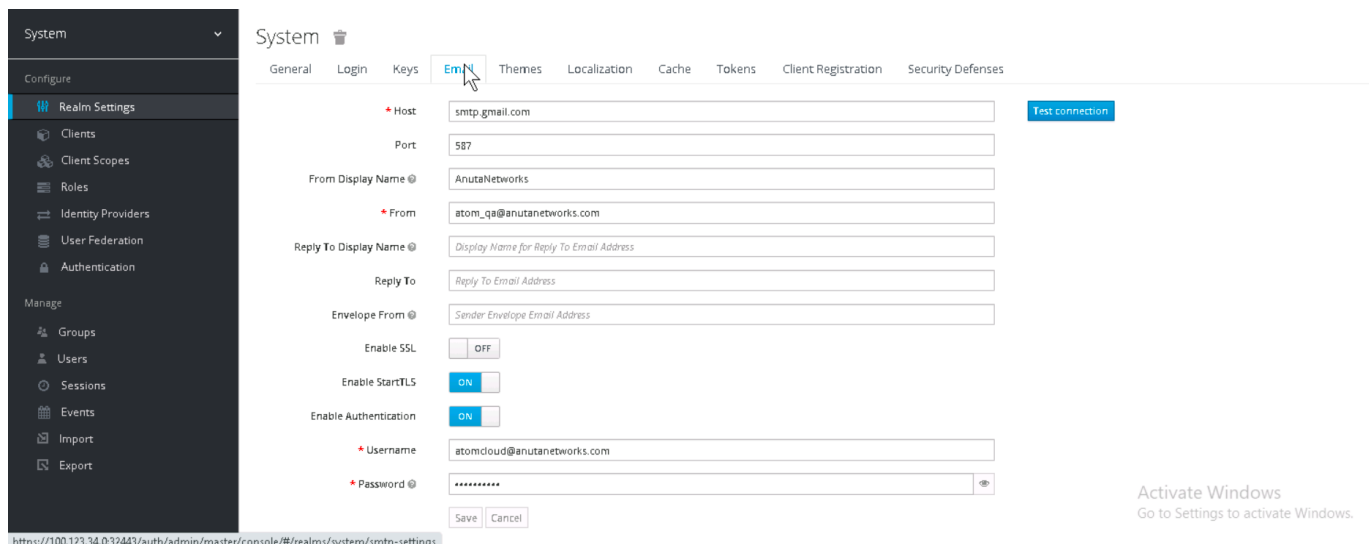
1. Navigate to keycloak url <https://<ATOM-Master-IP>:32443/auth/>:



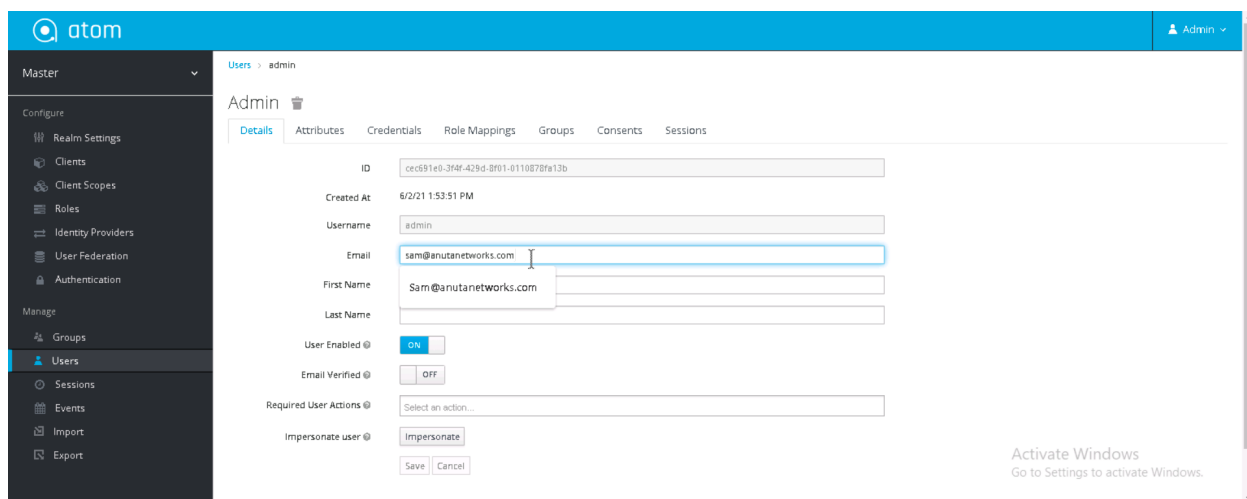
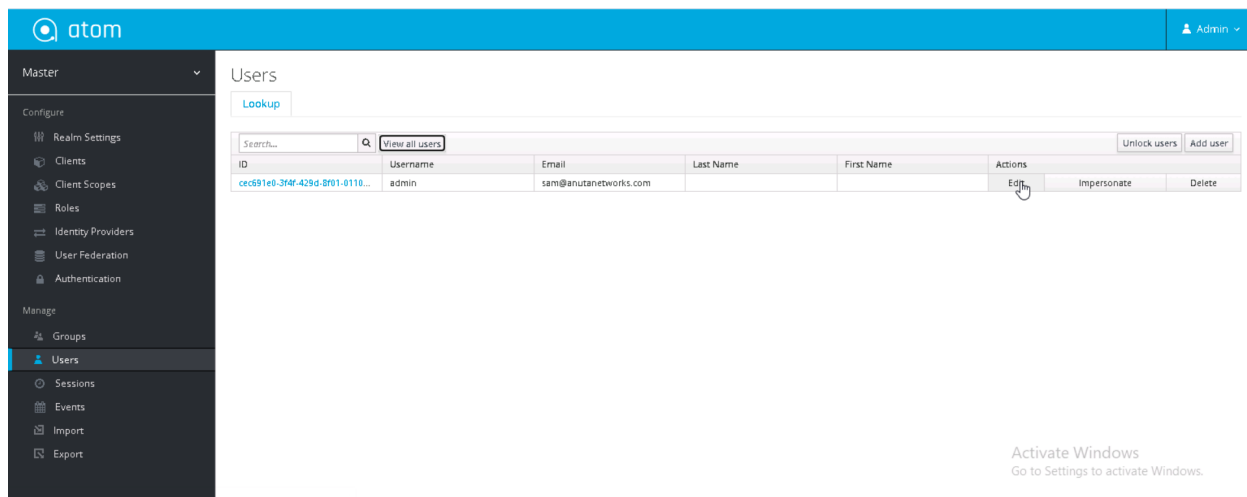
2. Click On the administration console and it goes to the login page if not given earlier. Enter the user name/password (admin/Secret@123)



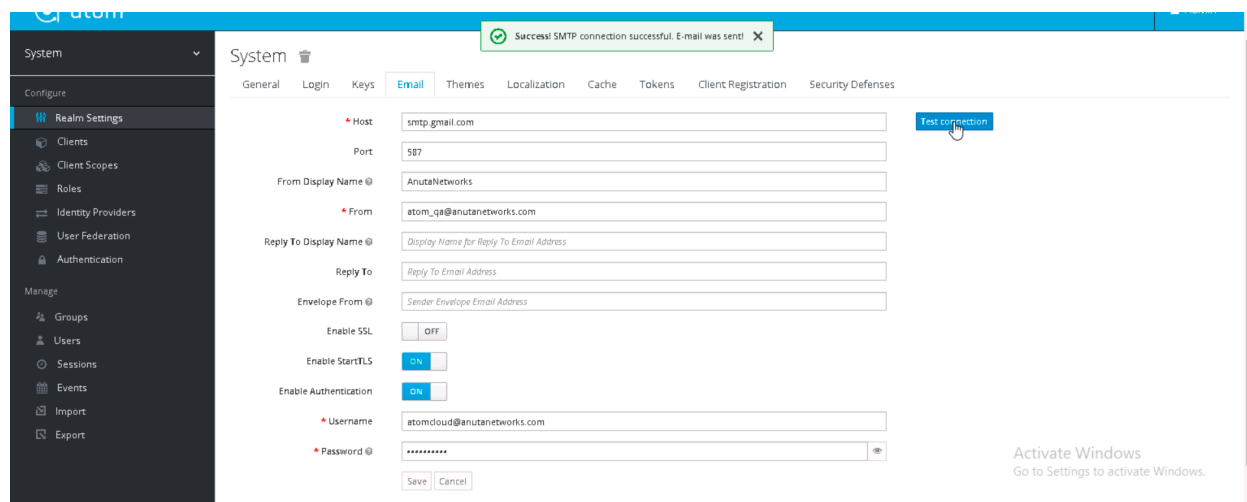
3. Navigate to **Keycloak->System->Realm setting->Emails**: Configure the below smtp setting in realm setting and save it.



4. Navigate to **keycloak->Master->Users->View All Users->Edit admin user**: Provide any dummy Email if needed, turn on the flag for “User enabled” and save it.



5. Navigate to **Keycloak->System->Realm Settings->Emails**: If we click on test connectivity it should be successful.



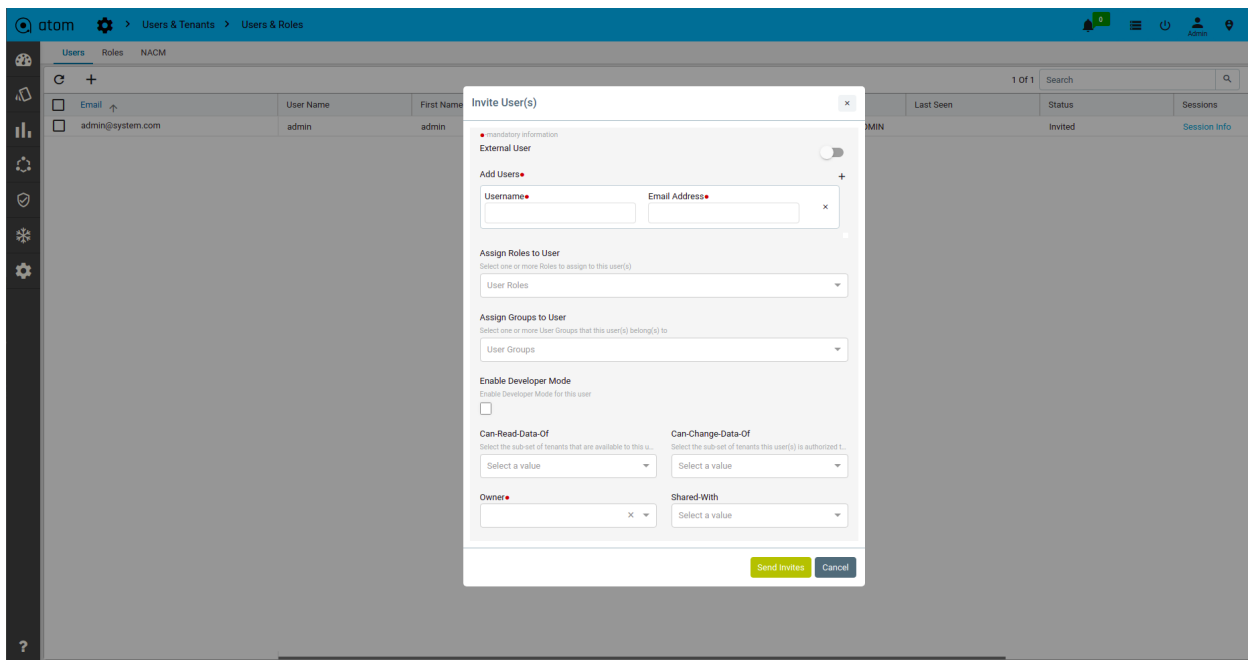
Adding a New User:

To add Users in ATOM, do the following:

1. Navigate to **Administration > Users & Tenants > Users & Roles** and click **Add**.
2. In the User Invitation screen, enter the values in the following fields:
 - a. **External User**: Enable this button to create AD, LDAP, or TACACS users in ATOM. When this button is not enabled, a local user is created in ATOM.

Note: A local user will be required to reset his account password on the first login attempt. An external user can directly login to ATOM using the link in their invitation email.

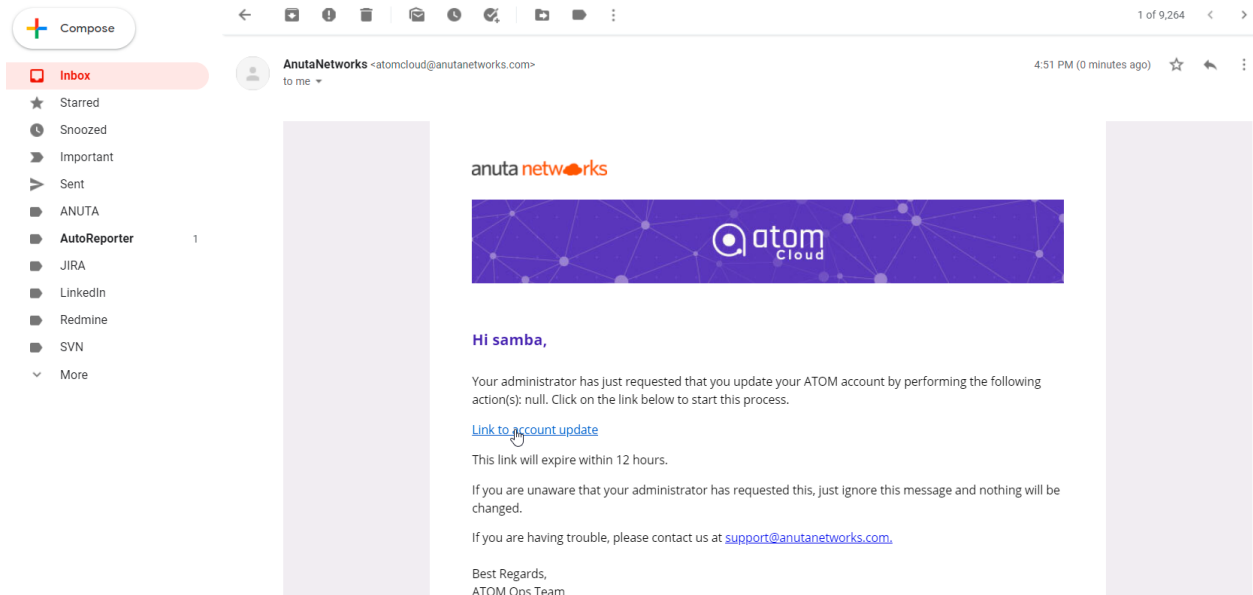
- b. **Add User**: The administrator can invite one or more users to ATOM at once. To add more than one user, click on the **+** button.



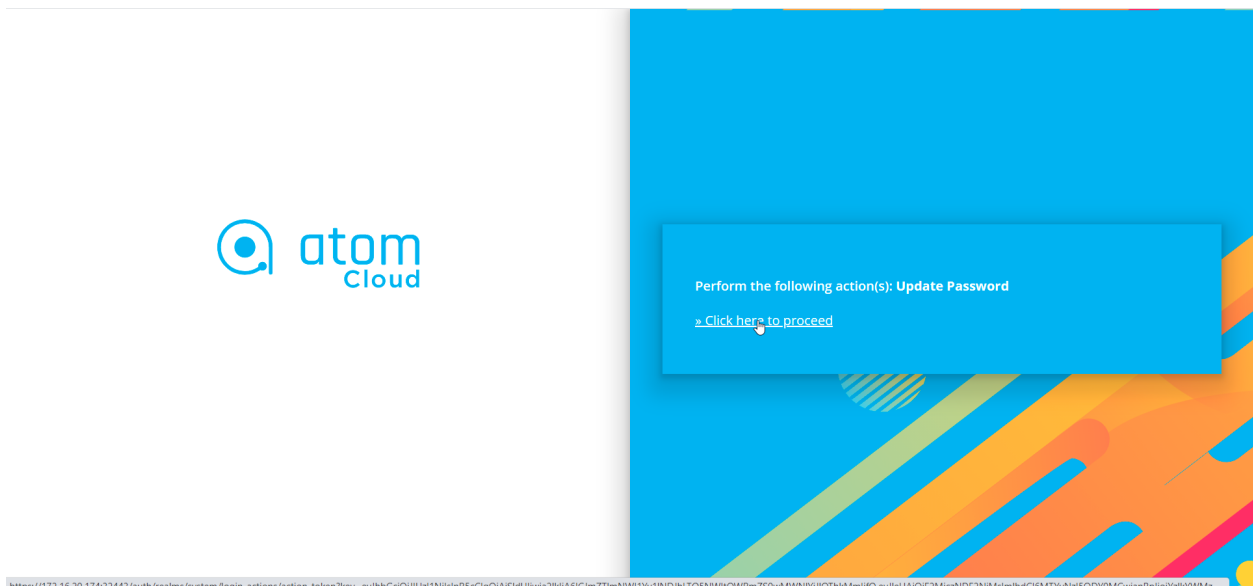
- i. **Username:** Enter an alphanumeric string of not more than 36 characters.

Note: While creating AD, LDAP, or TACACS users in ATOM, the name entered in the Username field should be the same as that created in the respective authentication server.

- ii. **Email Address:** Enter a valid email ID. Invitation emails to join ATOM Cloud will be sent to this address.
- c. **Assign Roles to User:** User can be assigned relevant Role(s) from the list of system-defined Roles. Each Role is a set of system-defined permissions given to the user.
- d. **Assign Groups to User:** Users can be put into pre-defined Group(s) to apply the access control privileges on them as a whole.
- e. **Enable Developer Mode:** In developer mode users have access to many tools to work with Atom platforms.
- f. **Can-Read-Data-Of:** Select the tenants, whose data this user can read.
- g. **Can-Change-Data-Of:** Select the tenants, whose data this user can change.
- h. **Owner:** Select the owner of this user.
- i. **Shared-With:** Select the tenants with whom this user should be shared.
3. If user creation is done, then check the mail sent to the given mail id of the new user created in ATOM for account update.




4. Click on here will take to atom page and it should show new update password options



5. Set the new password.



 You need to change your password.

New Password

Confirm password

6. Now new ATOM user login can login to atom UI with username/password

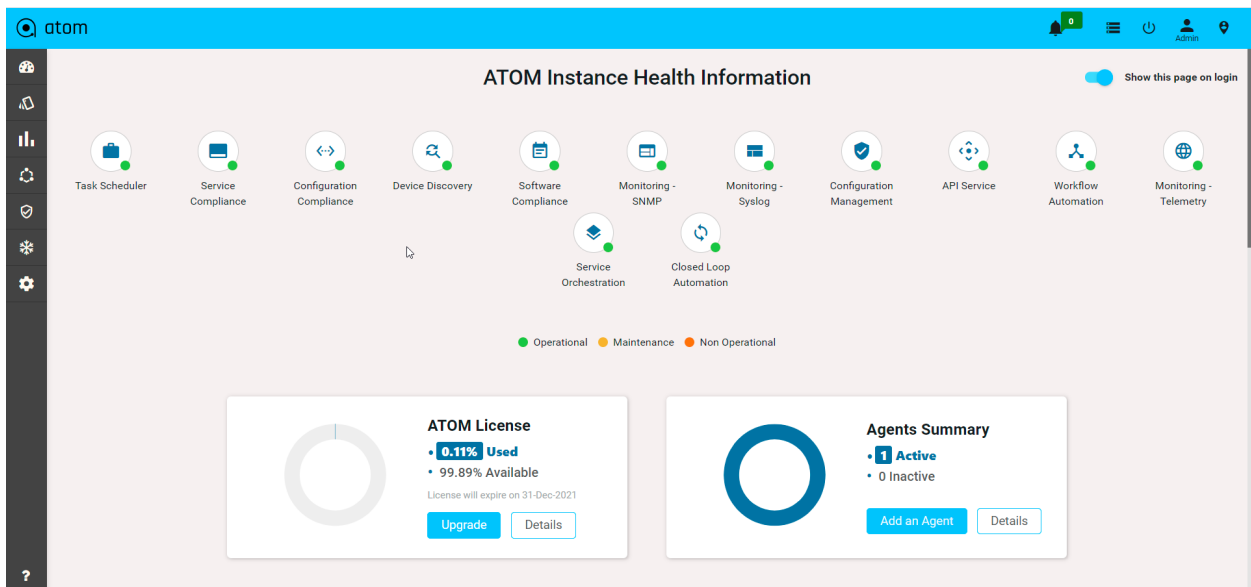


Username or email

Password

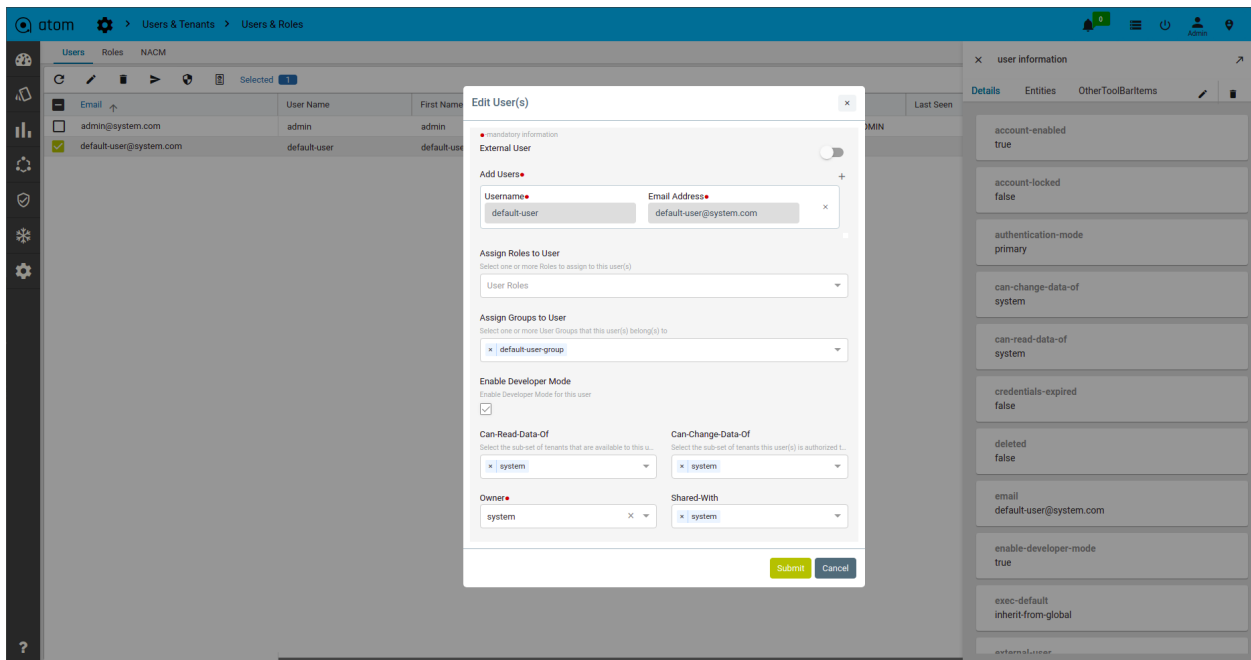
[Forgot Password?](#)

7. Access to the ATOM UI page possible after login with username/password.



Editing an Existing User:

User details can be edited by selecting an user. When a user is edited, the user is taken to the same form seen for adding a user. However, fields such as **Username**, **Email Address** and **External User** are non-editable as these form the fundamental identifiers for the user.




Resend User Invite:

At any instance, the administrator can **Enable** or **Disable** a user account by this option. When blocked, the user will not be allowed to login to ATOM.

From the drop-down menu, select the role that should be assigned to the user. These roles are defined earlier as in the section, [“Creating Roles”](#).

SNMPv3 users are required when the users or third party applications require additional authentication and access control provided by SNMPv3.

- 

The screenshot displays the Palo Alto Networks User Management interface. The top navigation bar shows the breadcrumb path: **Users > Roles > Users > Users & Roles**. Below this, there are tabs for **Users**, **Roles**, and **NACM**. The main content area is a table listing users, with columns for **Email**, **User Name**, **First Name**, and **Last Name**. The user **gshareef@anutanetworks.com** is selected, indicated by a yellow checkmark in the first column. On the right side, a sidebar titled **user information** is visible, containing a search bar and a list of links: **DIs-Config**, **Event-Subscription**, **Snmpv3** (highlighted with a mouse cursor), **Unprovisioned Configuration**, and **Workflow-User-Level-Authorizations-Policy**.

	Email	User Name	First Name	Last Name
<input type="checkbox"/>	abc@gmail.com	abc@gmail.com	abc@gmail.com	
<input type="checkbox"/>	admin@atom.local	admin		
<input type="checkbox"/>	admin@atom.local	admin		
<input checked="" type="checkbox"/>	gshareef@anutanetworks.com	gshareef@anutanetworks.com	gshareef@anutanetworks.com	
<input type="checkbox"/>	kpranav@anutanetworks.com	kpranav@anutanetworks.com	kpranav@anutanetworks.com	
<input type="checkbox"/>	krajashekar@anutanetworks.com	krajashekar@anutanetworks.com	krajashekar@anutanetworks.com	
<input type="checkbox"/>	ksubbareddy@anutanetworks.com	ksubbareddy@anutanetworks.com	ksubbareddy@anutanetworks.com	
<input type="checkbox"/>	ksurendra@anutanetworks.com	ksurendra@anutanetworks.com	ksurendra@anutanetworks.com	
<input type="checkbox"/>	mranya@anutanetworks.com	mranya@anutanetworks.com	mranya@anutanetworks.com	
<input type="checkbox"/>	pdivya@anutanetworks.com	pdivya@anutanetworks.com	pdivya@anutanetworks.com	
<input type="checkbox"/>	rlokeswaran@anutanetworks.com	rlokeswaran@anutanetworks.com	rlokeswaran@anutanetworks.com	
<input type="checkbox"/>	srikanthg@anutanetworks.com	srikanthg@anutanetworks.com	srikanthg@anutanetworks.com	
<input type="checkbox"/>	tswarupa@anutanetworks.com	tswarupa@anutanetworks.com	tswarupa@anutanetworks.com	
<input type="checkbox"/>	yharika@anutanetworks.com	yharika@anutanetworks.com	yharika@anutanetworks.com	

The screenshot shows the 'Create Snmpv3' configuration page in the ATOM interface. The page has a blue header with the ATOM logo and navigation links. The main content area is white and contains several configuration fields. On the left, there is a sidebar with icons for various functions. The configuration fields are as follows:

- Auth-Protocol:** A dropdown menu with three options: NO_AUTH, SHA, and MD5. SHA is currently selected.
- Auth-Password:** A text input field with a password strength indicator and a toggle to show/hide the password.
- Priv-Protocol:** A dropdown menu with one option: AES256.
- Priv-Password:** A text input field with a password strength indicator and a toggle to show/hide the password.
- Owner:** A dropdown menu with one option: system.
- Shared With:** A dropdown menu with one option: system.

- a. **Authentication Protocol:** Enter the mode of authentication when SNMPv3 is enabled.
- b. **Privacy Protocol:** Enter the requisite privacy protocol depending on the selected authentication protocol.

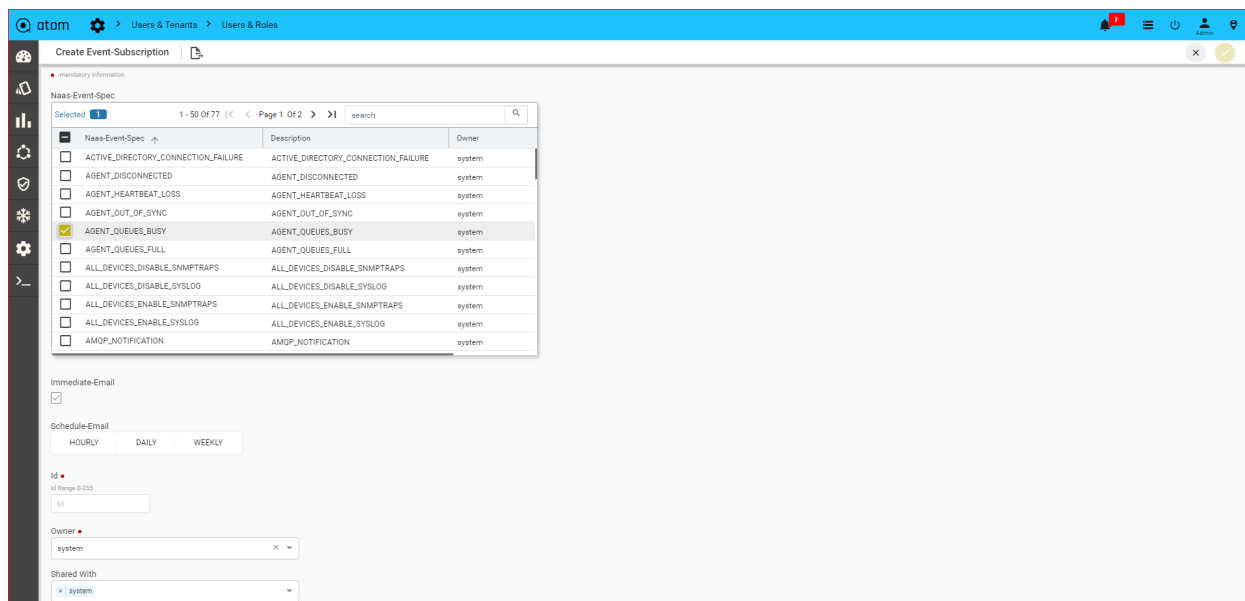
Subscribing to Events

Email notifications can be configured to be sent to the created user when a specific event or events occur in ATOM.

1. Navigate to **Administration > Users & Tenants > Add User**
2. In the Create user screen, click **user > entities > event subscription** in the right pane.
3. Fill the values in the fields described below:
4. Click **Add** to subscribe to a specific event or click **Add all** to add all the events available in ATOM.

An email is triggered and sent to the user when any of the events occur in ATOM.

You can configure the mail to be sent immediately after the occurrence of the event or schedule the mail notifications to be sent at periodic intervals as illustrated in the screenshot below:



Workflow-User-Level-Authorization

If it is decided to have user level permissions what should be the default permissions? That question is answered by the global configuration at `/controller:nacm/workflow-policy-for-auto-creation-of-authorization`

If it is needed to customize those permissions for a specific user? That question is answered by this model

Using this model you can use

- 1.enable the user level permissions
- 2.choose to use the global default or customize

Here three options are there

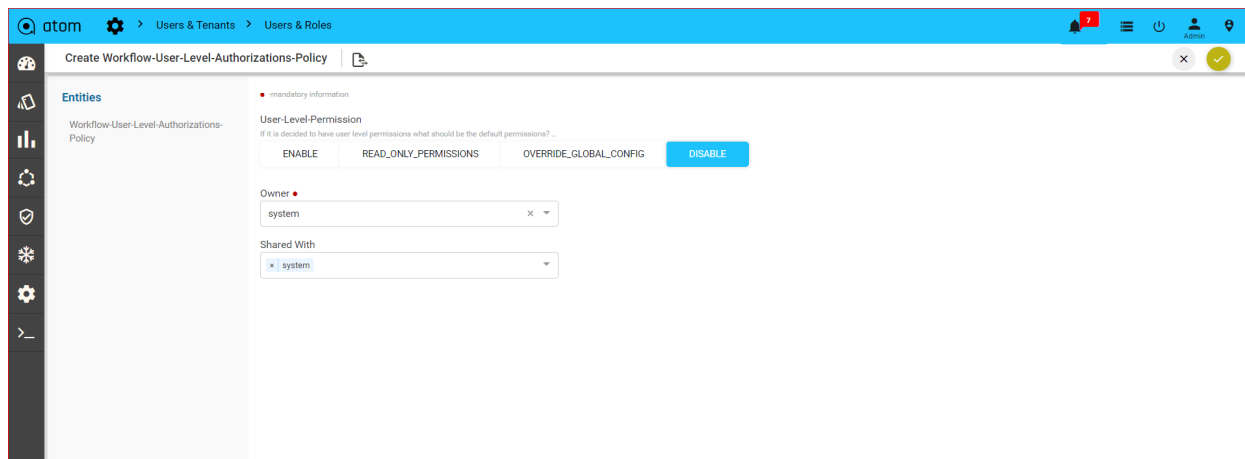
DISABLE:: Is the default selection.Using this option you can explicitly disable the auto creation of permission for this user.

ENABLE:Enables global defaults for this user.

READ-ONLY-PERMISSION:Enables the read only permissions out of the global defaults

OVERRIDE-GLOBAL-CONFIG:Enables user level permission creation for this user,and, for those permissions you're going to specify here, explicitly.

1. Navigate to **Administration > Users & Tenants > Add User**
2. In the Create user screen, click **user >entities> workflow-user-level-permission-authorization** in the right pane.
3. By default workflow-user-level- authorization is disabled it.



Configuring NACM

By implementing NACM developed by NETCONF, ATOM enables administrators to allow or deny access to protocol operations and data to a set of users. Access Control in ATOM is achieved by a combination of “Rule-list” and “User Groups”. Rules are grouped into Rule-list and users are assigned to User Groups to control access to resources managed by ATOM.

To set the global access control settings in ATOM, do the following:

1. Navigate to **Administration > Users & Tenants > Users & Roles**
2. Click Edit to modify the global settings for access control.

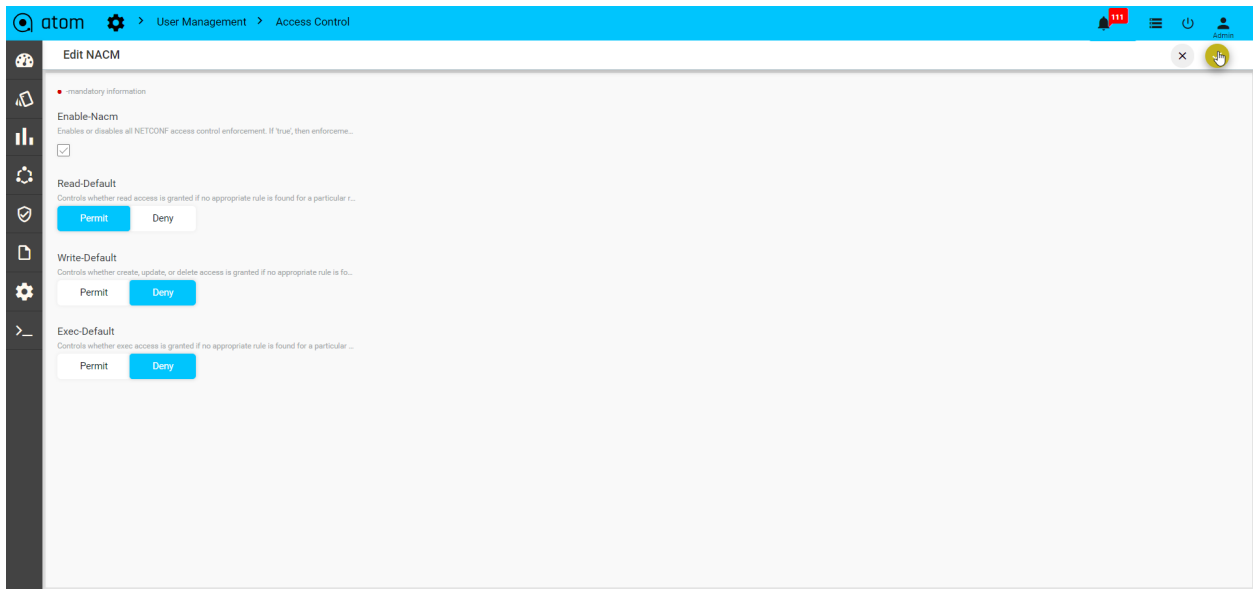
As an administrator, you can set the default access control settings which are applicable to all entities in ATOM

- **Enable NACM:** Select this option to set a group of read, write, and execute options that should be applicable by default to all the entities in ATOM.
- **Read Default:** The default value is “permit” for all the operations/objects(Controls whether read access is granted if no appropriate rule is found particular read request)
- **Write Default:** The default value is “deny” for all the operations/objects(Controls whether create,update or delete access is granted if no appropriate rule is found particular write request)
- **Exec Default:** The default value is “permit” for all the operations/objects(Controls whether exec access is granted if no appropriate rule is found particular protocol operation request)

The screenshot shows the ATOM web interface. The top navigation bar includes the ATOM logo, a settings icon, and the breadcrumb path: Users > Users & Roles. The main content area is titled 'Users' and displays a table of users. The table has columns for Email, User Name, First Name, Last Name, Role, and Last Seen. A sidebar menu on the left contains various system administration options, with 'Users & Roles' highlighted.

Email	User Name	First Name	Last Name	Role	Last Seen
admin@atom.local	admin	System	Administrator	ROLE_WORKFLOW_ADMIN,ROLE_SYSTEM_ADMIN	06/24/2021 04:03:30
cuser01@anuta.com	cuser01				
cuser02@anuta.com	cuser02				
dalekhya@anutanetworks.com	alekhya	alekhya	alekhya		
gsusmitha@anutanetworks.com	susmitha	susmitha	susmitha		
ks.com	narendra	narendra	narendra	ROLE_WORKFLOW_ADMIN	06/24/2021 06:36:27
s.com	raviteja	raviteja	raviteja		

This screenshot shows the same ATOM web interface, but the main content area is empty except for a search bar and a message that says 'please select entity to display'. The sidebar menu remains the same, with 'Users & Roles' highlighted.



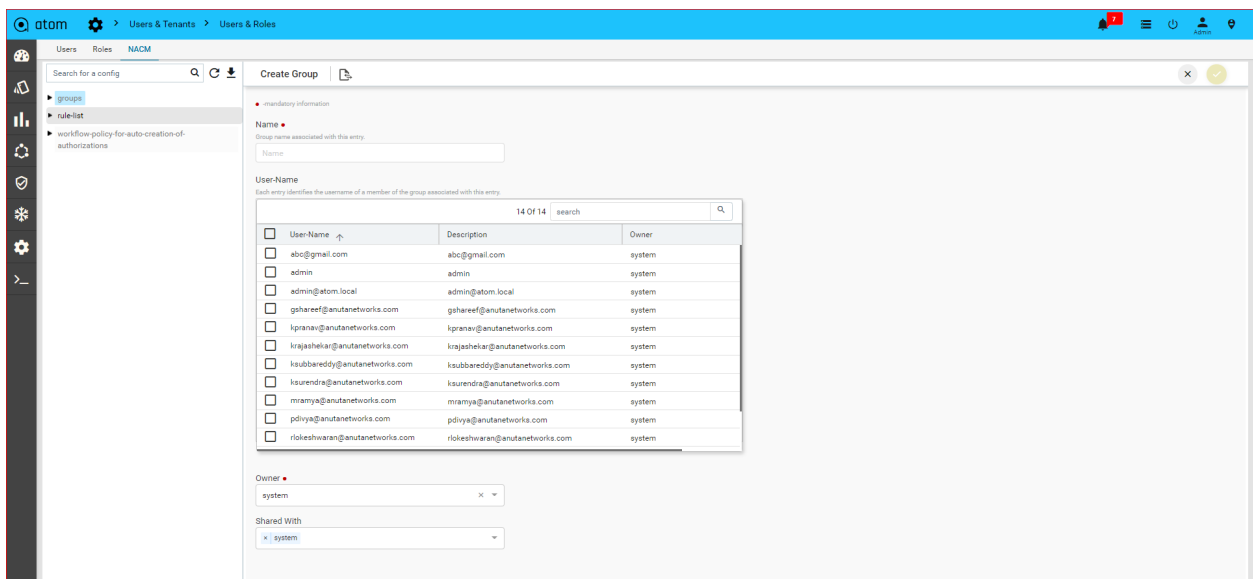
Note: These default settings that are entered in the above screen can be overridden by the values set by a specific rule created for an entity.

Adding User Groups

You can organize users to groups and apply the access control privileges on them as a whole.

To create a User Group:

1. Navigate to **Administration >Users & Tenants >Users & Roles>** click **NACM**
2. In the **Details** pane, on the right, click **Groups > Add Group**

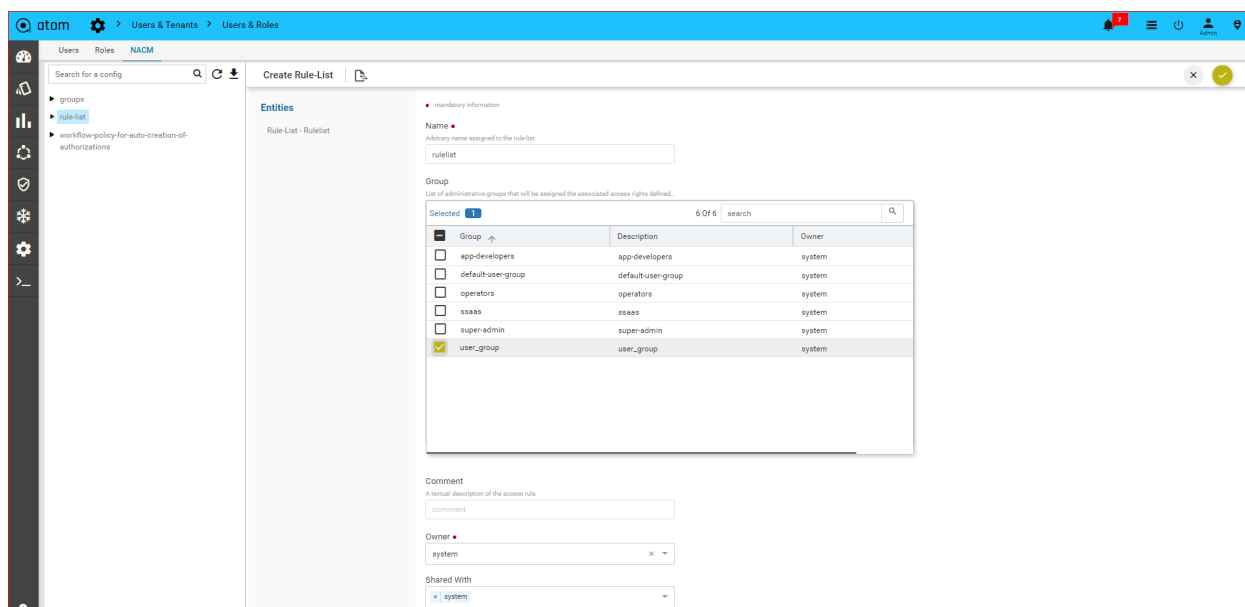


- **Name:** Enter the name for the User Group
- Select a user to this group.

Creating Rule lists

Rule lists are an aggregated list of rules created in ATOM.

1. Navigate to **Administration > Users & Tenants > Users & Roles > click NACM**
2. In the **Entities** pane, on the right, click **Rule-list > Add(+)**
3. **Name:** Enter the text that will be used as a name for the rule list.
4. Select a group (user group) to which the created rules in the Rule list should be assigned.



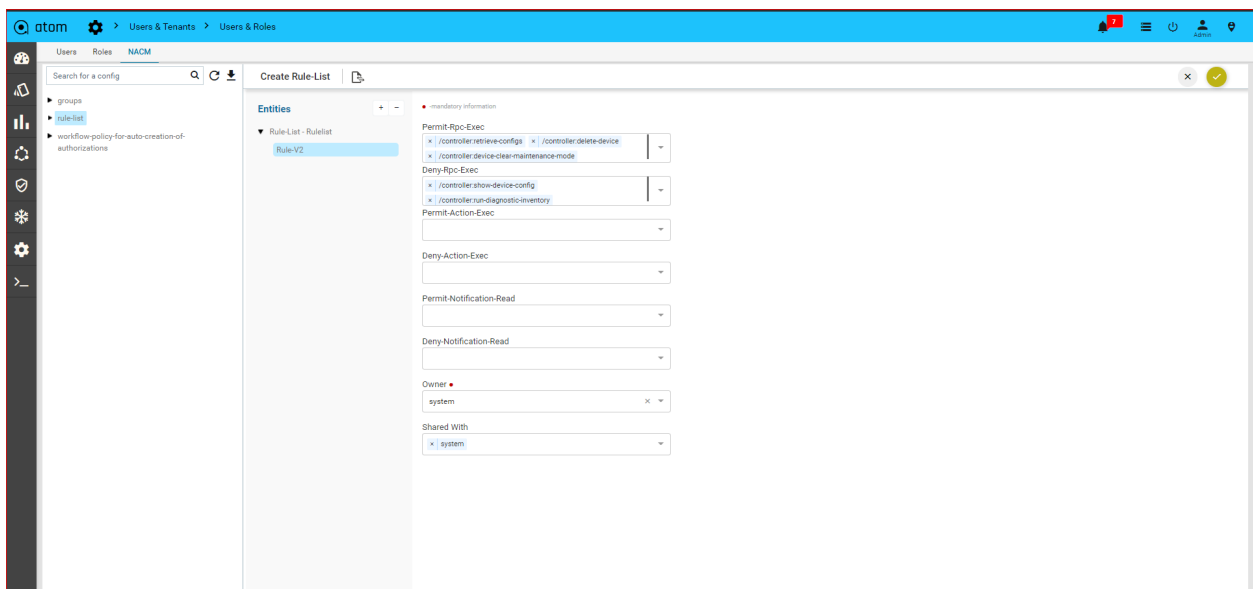
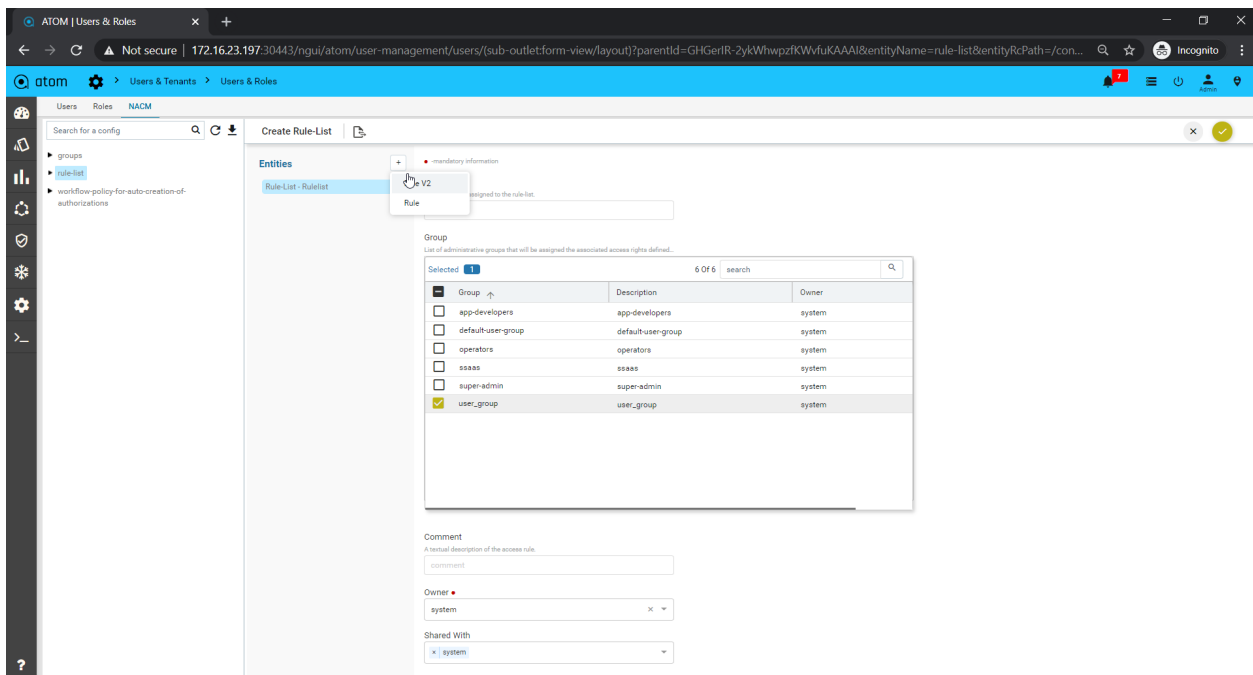
Creating Rule V2:

Rules are the conditions that identify individual objects in the system. Rules also capture whether a user of a network object should be “granted” or “denied” RPC permission.

RPC : Choose a remote procedure call (RPC) on which the access control needs to be applied.

1. Click the created **Rule-list > Entities > Rule V2 > Add(+)**
2. In the **Create rule** screen, enter values in the fields explained below:

Create a Rule v2, “rulev2” to be given the multiple RPCs in drop down Permit-RPC-Exec(Ex:Run device inventory, Extended inventory are etc) and Deny-RPC-Exec(Ex:Topology inventory, run dsl are etc)

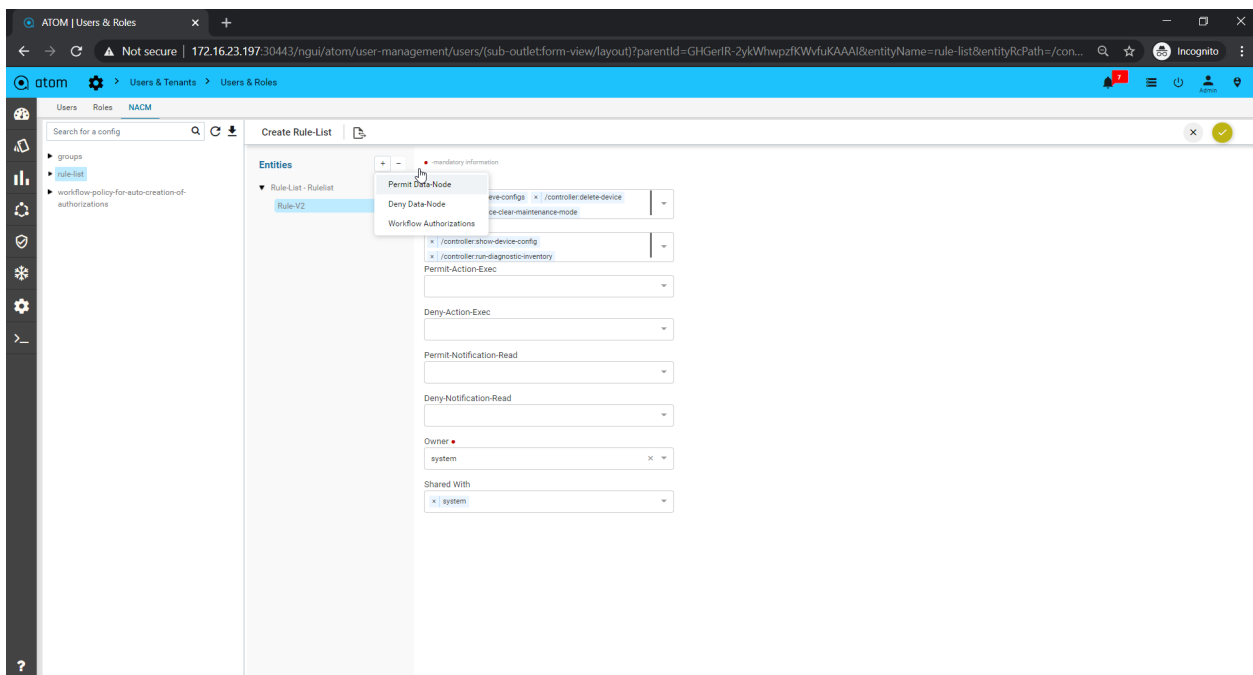


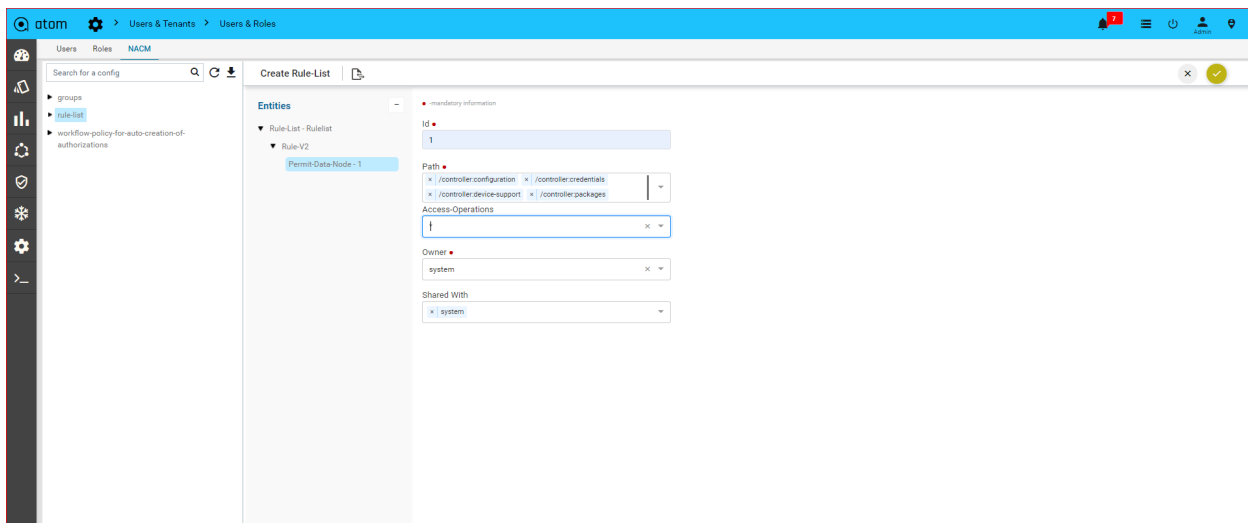
- Select a rule v2 that will be shown the below permit and deny data node, to grant either allow or deny access to the rule determined to match a particular request.
 - Permit datanode
 - Deny datanode

Click on rule v2 to add(+) permit data node to fill the below details:

- **Id** :Enter string or number(Ex:2)

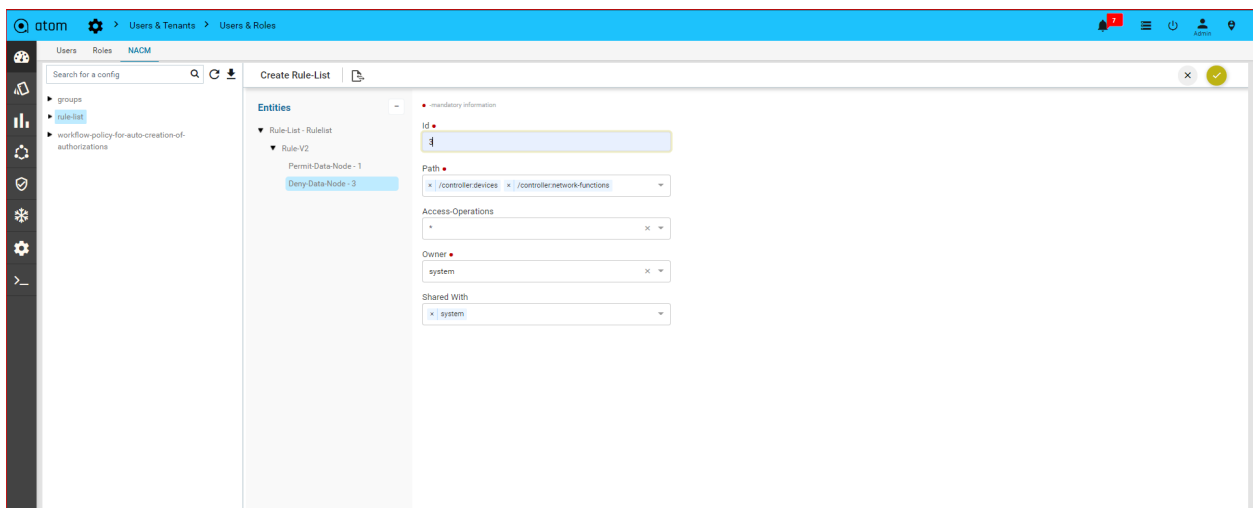
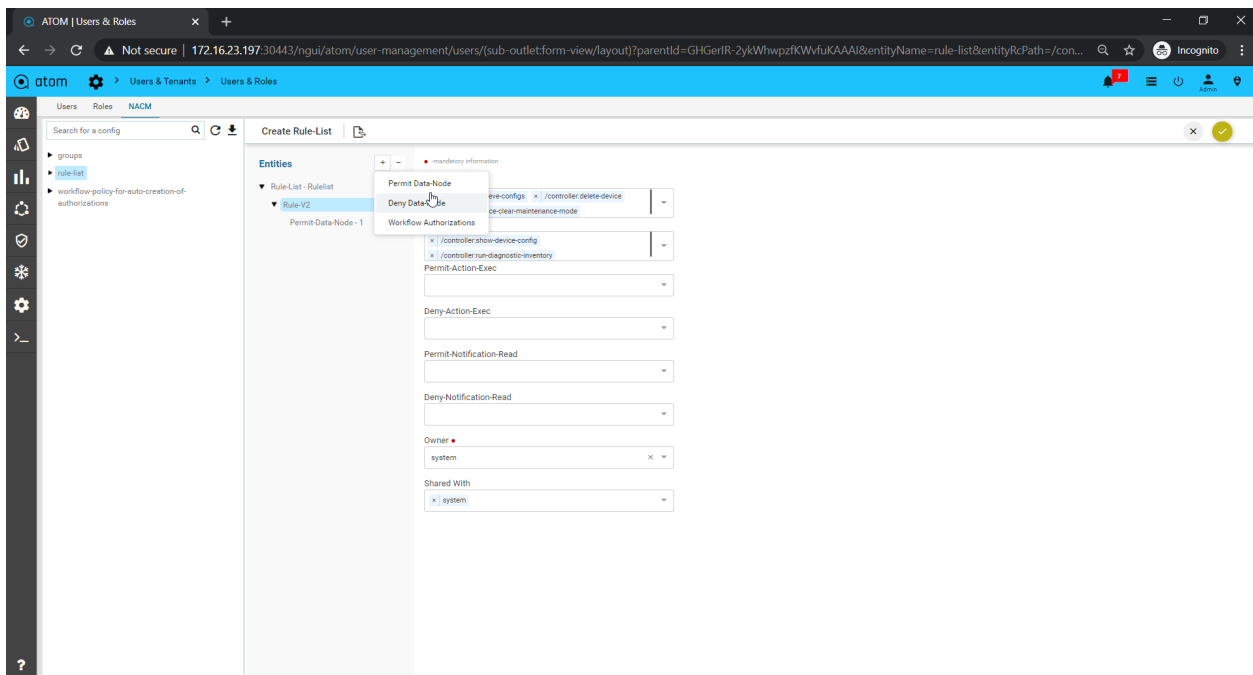
- **Path** : Enter the path of the object in the data tree on which the rule should be applied. This is applicable only for the rule type, 'data-node' Select the single or multiple Data node paths
- **Access operations** : Select any of the operations on the ATOM entity that needs to be controlled.
 - * [This symbol indicates all the operations (Create, Update, Delete, Read) are included]
 - Create read update delete
 - Read create
 - Read update
 - Read delete
 - Various operation to be tested





Click on rule v2 to add(+) deny data node to fill the below details:

- **Id** :Enter string or number(Ex:3)
- **Path** : Enter the path of the object in the data tree on which the rule should be applied. This is applicable only for the rule type, 'data-node' Select the single or multiple Data node paths
- **Access operations** : Select any of the operations on the ATOM entity that needs to be controlled.
 - * [This symbol indicates all the operations (Create, Update, Delete, Read) are included]
 - Create read update delete
 - Read create
 - Read update
 - Read delete
 - Various operation to be tested



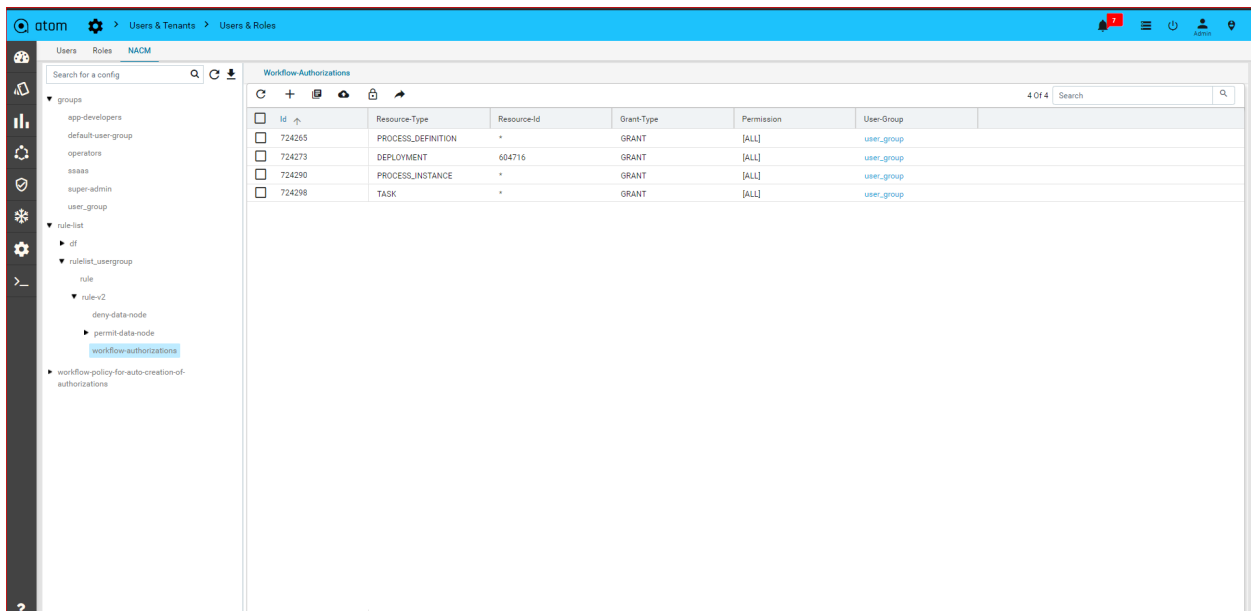
Workflow Authorization under rule v2:

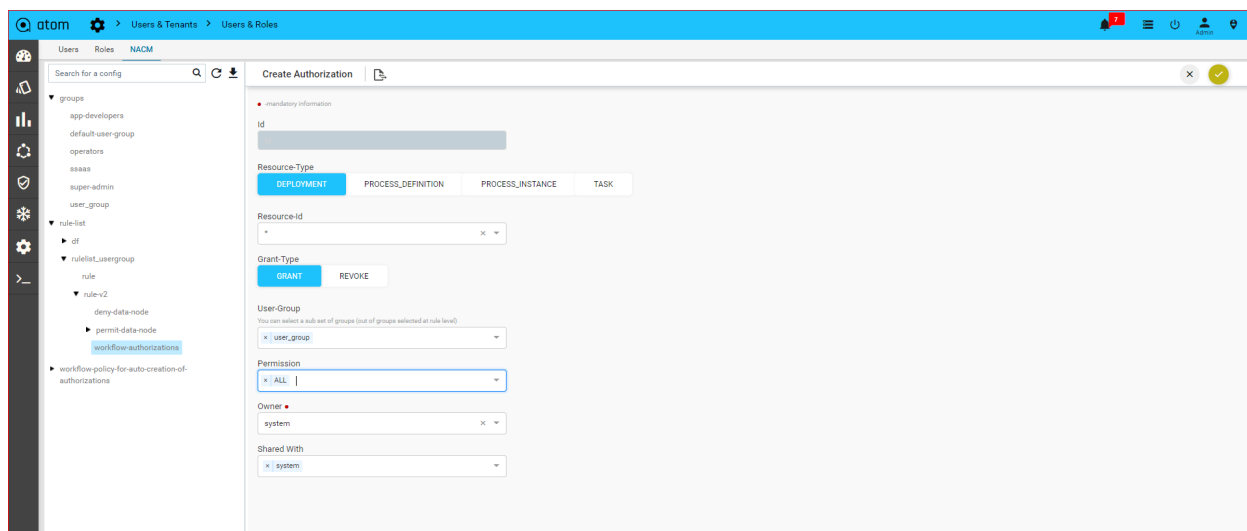
Camunda allows users to authorize access to the data it manages. This makes it possible to configure which user can access which process instances, tasks, etc... If it is needed to customize those permissions for a specific workflow user

Deployment: The option is weather package is upload/load/unload and delete the package

Click on Workflow authorization to add(+) to fill the below details for Deployment:

- **Id** :Id is the hardcoded
- **Resource type** : Select the resource type is **Deployment**
- **Resource id** : Select (*) from resource id drop down to be accessed by all packages or any specific package resource id.
- **Grant type:**
- **Grant**::Ranges over users and groups and grants a set of permissions. Grant authorizations are commonly used for adding permissions to a user or group that the global authorization revoked.
- **Revoke**::Ranges over users and groups and revokes a set of permissions. Revoke authorizations are commonly used for revoking permissions to a user or group that the global authorization grants.
- **User group**: You can select a subset of groups to mapped it.
- **Permissions**: A Permission defines the way an identity is allowed to interact with a certain resource. (ALL,Create, Read, Delete)

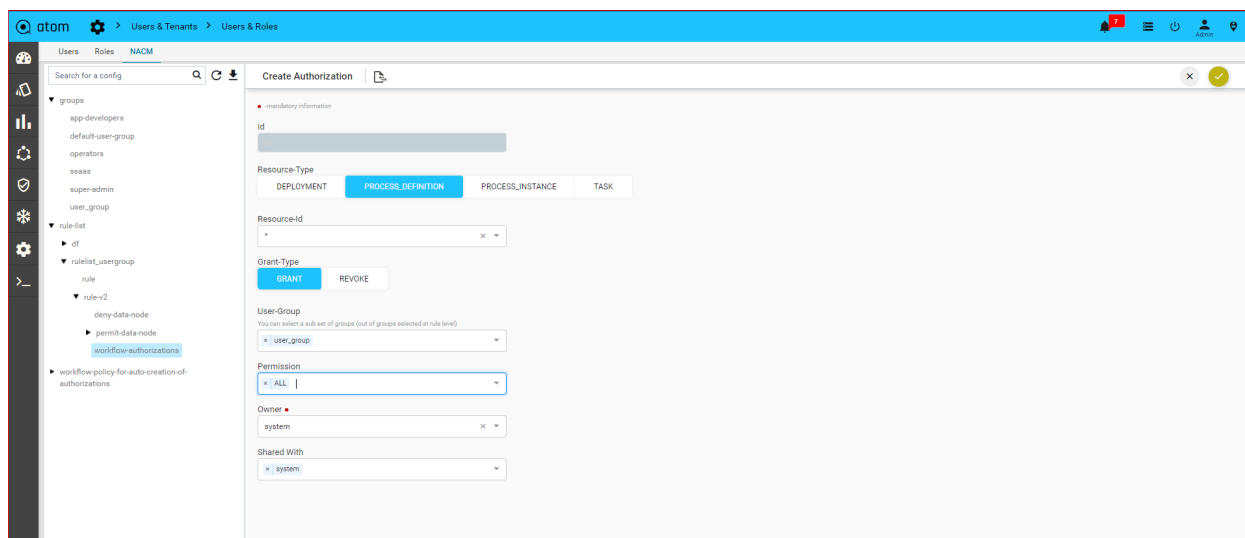




Process definition: This option is access the workflow grid with specific workflow or all based on given process definition

Click on Workflow authorization to add(+) to fill the below details for process definition:

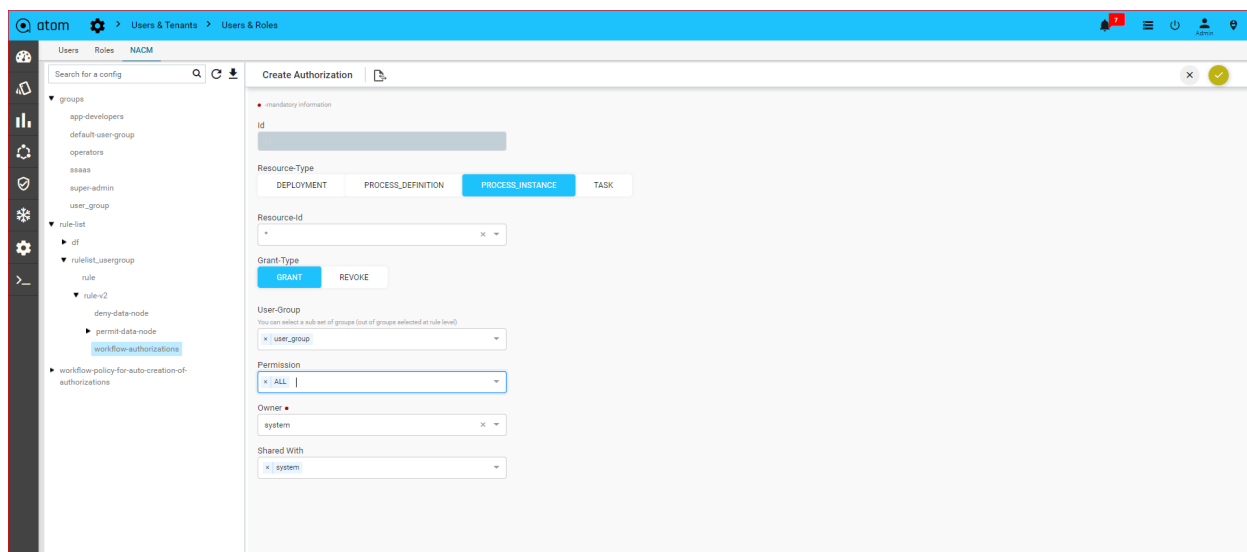
- **Id** :Id is the hardcoded
- **Resource type** : Select the resource type is **process definition**
- **Resource id** : Select (*) from resource id drop down to be accessed by all workflows or give any specific workflow of resource id.
- **Grant type:**
- **Grant**:Ranges over users and groups and grants a set of permissions. Grant authorizations are commonly used for adding permissions to a user or group that the global authorization revoked.
- **Revoke**::Ranges over users and groups and revokes a set of permissions. Revoke authorizations are commonly used for revoking permissions to a user or group that the global authorization grants.
- **User group**: You can select a subset of groups to mapped it.
- **Permissions**: A Permission defines the way an identity is allowed to interact with a certain resource.Choose the various permission (ALL,Create instance, Read, Delete-instance,Update-instance,update-history,Migrate-instance,Update-task-variable, Update-task)



Workflow instance: This option is a workflow instance and is a running instance of a workflow definition.

Click on **Workflow authorization to add(+)** to fill the below details for **Workflow instance**

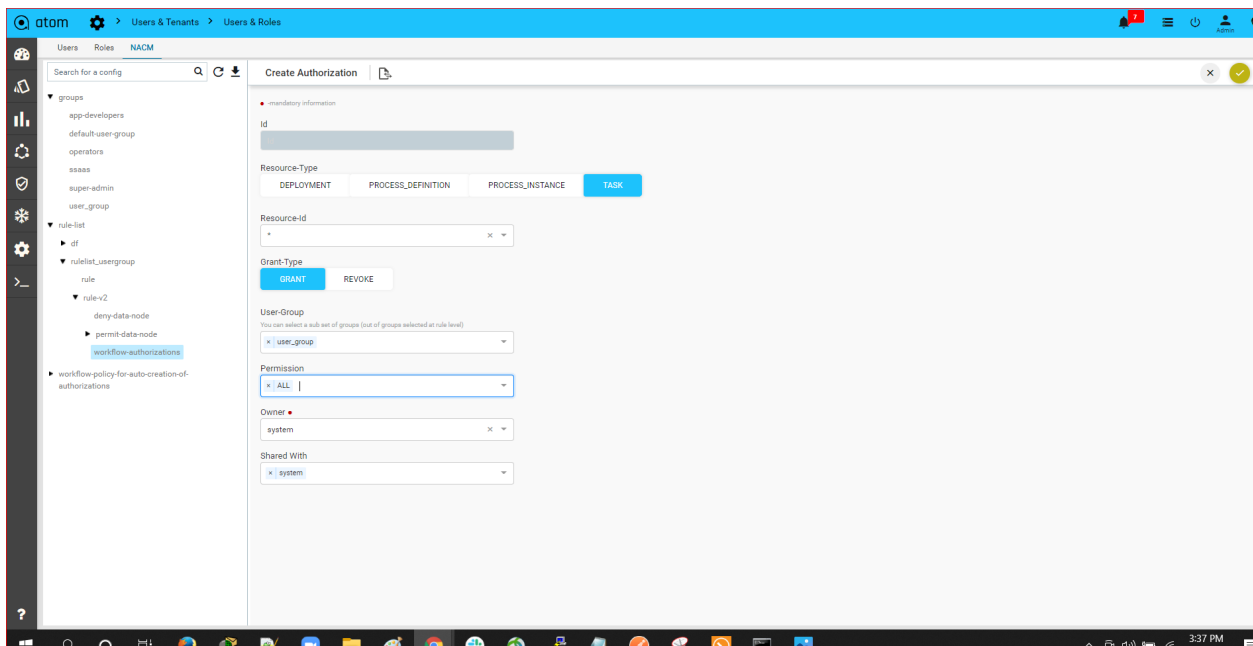
- **Id :** Id is the hardcoded
- **Resource type :** Select the resource type is **Workflow instance**
- **Resource id :** Select (*) from resource id drop down to be accessed by all workflow instances or any specific workflow instance of resource id.
- **Grant type:**
- **Grant:** Ranges over users and groups and grants a set of permissions. Grant authorizations are commonly used for adding permissions to a user or group that the global authorization revoked.
- **Revoke::** Ranges over users and groups and revokes a set of permissions. Revoke authorizations are commonly used for revoking permissions to a user or group that the global authorization grants.
- **User group:** You can select a subset of groups to mapped it.
- **Permissions:** A Permission defines the way an identity is allowed to interact with a certain resource. Choose the various permission(ALL, Create, Update, Read, Delete)



Tasks: This option is a user can perform different actions on a task, like assigning the task, claiming the task or completing the task. If a user has “Update” permission on a task (or “Update Task” permission on the corresponding process definition) then the user is authorized to perform all these task actions

Click on Workflow authorization to add(+) to fill the below details for Task

- **Id** :Id is the hardcoded
- **Resource type** : Select the resource type is **Task**
- **Resource id** : Select (*) from resource id drop down to be accessed by all workflow tasks or any specific workflow task of resource id.
- **Grant type:**
- **Grant:** Ranges over users and groups and grants a set of permissions. Grant authorizations are commonly used for adding permissions to a user or group that the global authorization revoked.
- **Revoke:** Ranges over users and groups and revokes a set of permissions. Revoke authorizations are commonly used for revoking permissions to a user or group that the global authorization grants.
- **User group:** You can select a subset of groups to mapped it.
- **Permissions:** A Permission defines the way an identity is allowed to interact with a certain resource. Choose the various permission (ALL, Create, Update, Read, Delete, Read-History, Task-Assign, Task-Work)



Note:: Need to be given the permit rpc exec in rule v2 based on given workflows & follow the workflow payload

<output>

<user-summary>

<username>srikanth</username>

<write-default>inherit-from-global</write-default>

<read-default>inherit-from-global</read-default>

<exec-default>inherit-from-global</exec-default>

<roles/>

<user-group>

<name>group_disney</name>

<rule-list>

<name>rulelist_disney</name>

<group>group_disney</group>

<rule-v2>

<permit-rpc-exec>/controller:retrieve-configs</permit-rpc-exec>

<permit-rpc-exec>/controller:run-device-inventory</permit-rpc-exec>

<permit-rpc-exec>/disney_ipv6_api:get-neighbor-devices</permit-rpc-exec>

```

    <permit-rpc-exec>/disney_ipv6_api:ipv6-junos-routing</permit-rpc-exec>
    <permit-rpc-exec>/disney_ipv6_api:isis-routing</permit-rpc-exec>
    <permit-rpc-exec>/disney_ipv6_api:servicemodel_update</permit-rpc-exec>
    <permit-rpc-exec>/disney_ipv6_config:append-task-details</permit-rpc-exec>
    <permit-rpc-exec>/disney_ipv6_config:execute-command</permit-rpc-exec>
    <permit-rpc-exec>/disney_ipv6_config:ipv6-addition</permit-rpc-exec>
    <permit-rpc-exec>/disney_ipv6_config:ipv6-routing</permit-rpc-exec>
    <permit-rpc-exec>/configarchive:config-diff</permit-rpc-exec>
  </workflow-authorizations/>
  <permit-data-node>
    <id>1</id>
    <access-operations>read create update delete</access-operations>
    <path>/ipam:ipv6-pools</path>
  </permit-data-node>
  <permit-data-node>
    <path>/controller:devices</path>
    <id>2</id>
    <access-operations>*</access-operations>
  </permit-data-node>
</rule-v2>
</rule-list>
<group-level-workflow-authorizations>
  <authorization>ID:3585112 GRANT PROCESS_DEFINITION ipv6_configs
[ALL]</authorization>
  <authorization>ID:3591653 GRANT PROCESS_DEFINITION
interface_ipv6_configuration [ALL]</authorization>
  <authorization>ID:3610357 GRANT PROCESS_DEFINITION
ipv6_routing_configuration [ALL]</authorization>
  <authorization>ID:3611232 GRANT PROCESS_DEFINITION bgp_peer_configuration
[ALL]</authorization>
  <authorization>ID:3585054 GRANT PROCESS_INSTANCE * [ALL]</authorization>

```

```
</group-level-workflow-authorizations>
</user-group>
<user-level-workflow-authorizations/>
</user-summary>
</output>
```

Integrating ATOM with Central Authentication Systems

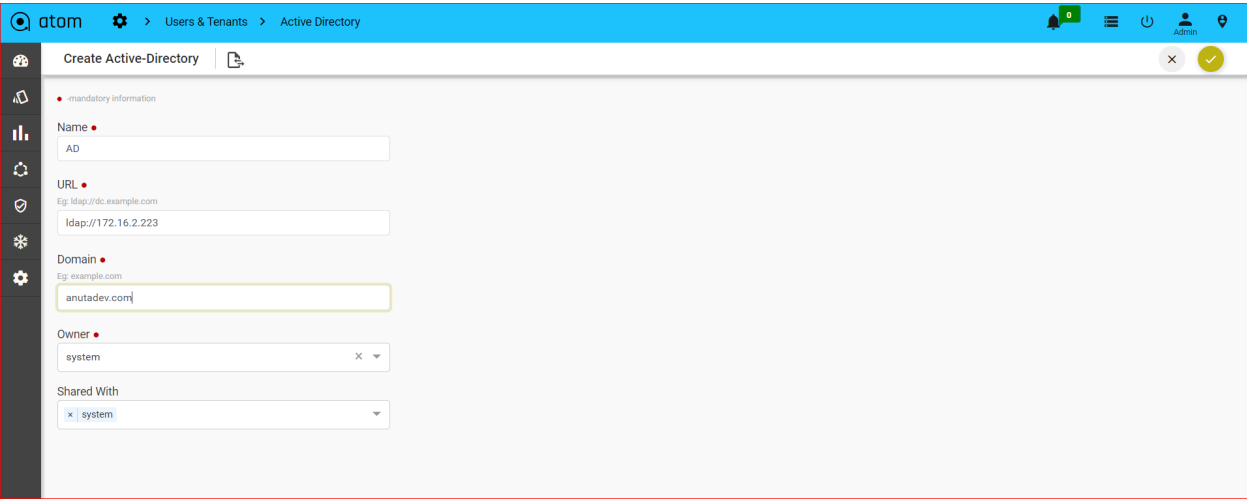
By integrating with central authentication systems such as LDAP, AD or TACACS, the users created in these servers can login to ATOM using their credentials created in their respective servers.

Managing Active Directory Users()

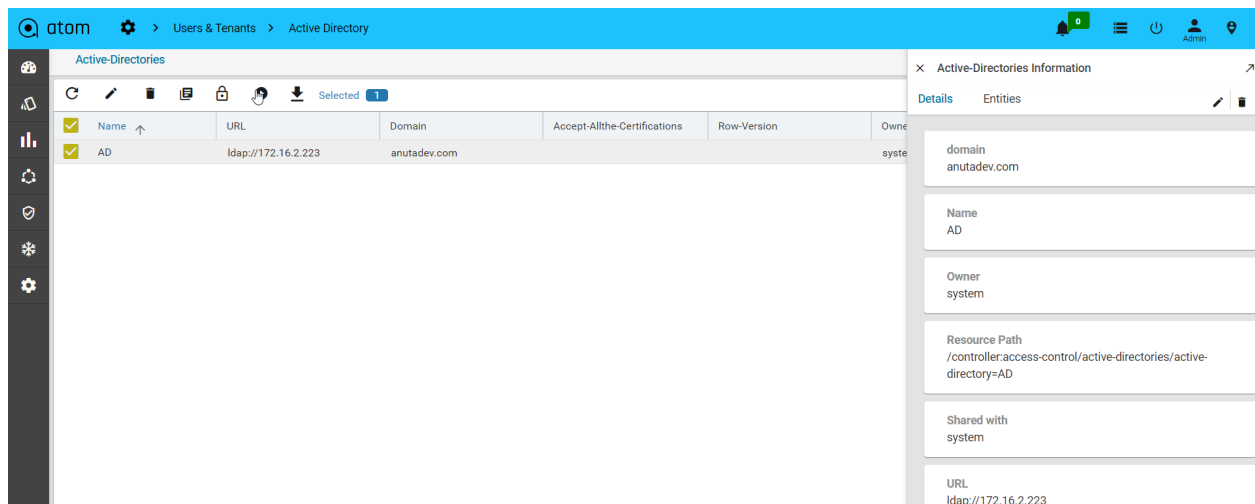
You can import users of Active Directory into ATOM and manage them as other users of ATOM.

For secure communication between the AD server and ATOM, import the security certificate into ATOM.

1. Navigate to **Administration >Users & Tenants > Active Directory**
2. In the **Create Active Directory** screen, enter the following:
 - **URL:** Enter the URL address of the Active Directory(Ex:ldap://172.16.2.223)
 - **Domain:** Enter the name of the domain(Ex:anutadev.com)



- 327



Managing OpenLDAP Users

You can import the existing LDAP users and user groups of a tenant, thus enabling the tenant users to login to ATOM using their LDAP credentials.

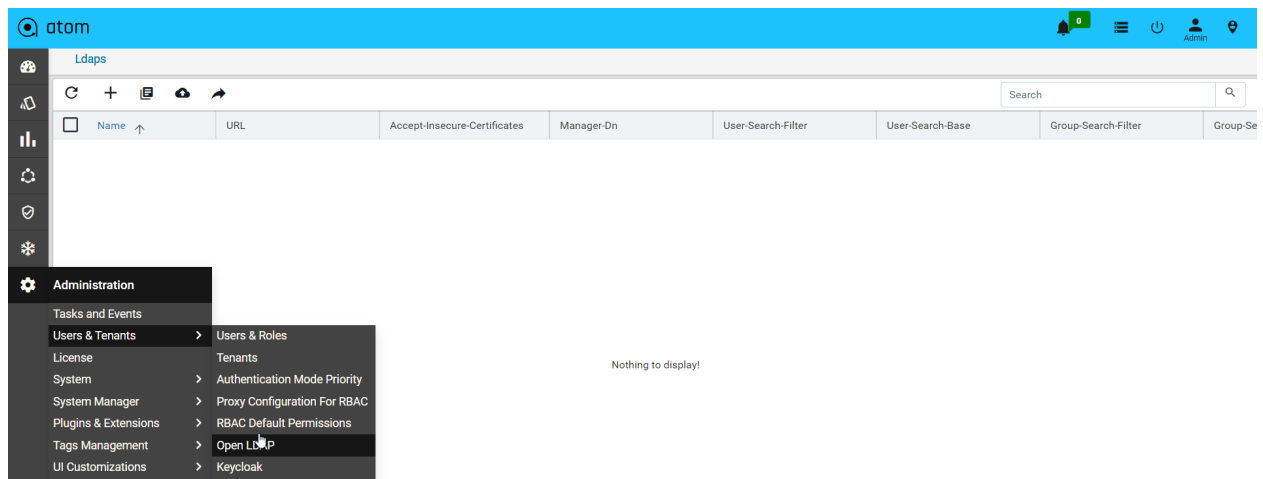
Importing OpenLDAP Users:

Before you begin, ensure that the following conditions are met with:

1. Users and groups are created in the LDAP server
2. OpenLDAP users are created as Tenants in ATOM

To create LDAP users in ATOM:

1. Navigate to **Administration > Users & Tenants > OpenLDAP**
2. Click **Add**



The screenshot shows the 'Create LDAP' configuration window. The left sidebar contains icons for various system components. The main area has the following fields:

- Name:** Text input field.
- URL:** Text input field with a placeholder: "ldap://172.16.2.22:10389/o=nCloud".
- Accept Insecure Certificates:** A checkbox.
- Manager DN:** Text input field with a placeholder: "uid=admin,ou=system".
- Manager Password:** Password input field with a toggle for visibility.
- User Search Filter:** Text input field with a placeholder: "(uid={0})".
- User Search Base:** Text input field with a placeholder: "ou=users".
- Group Search Filter:** Text input field with a placeholder: "(uniqueMember={0})".
- Group Search Base:** Text input field with a placeholder: "ou=groups".
- Group Role Attribute:** Text input field with a placeholder: "cn".
- User Lastname Attribute:** Text input field with a placeholder: "sn".
- User Firstname Attribute:** Text input field with a placeholder: "gn".
- Sync Period Hours:** Text input field with a placeholder: "1".
- Last Sync Time:** Text input field with a placeholder: "2023-10-27 10:00:00".
- Domain:** A dropdown menu with "system" selected.
- Shared With:** A text input field.

3. In the **Create OpenLDAP** screen, enter the values in the following fields:

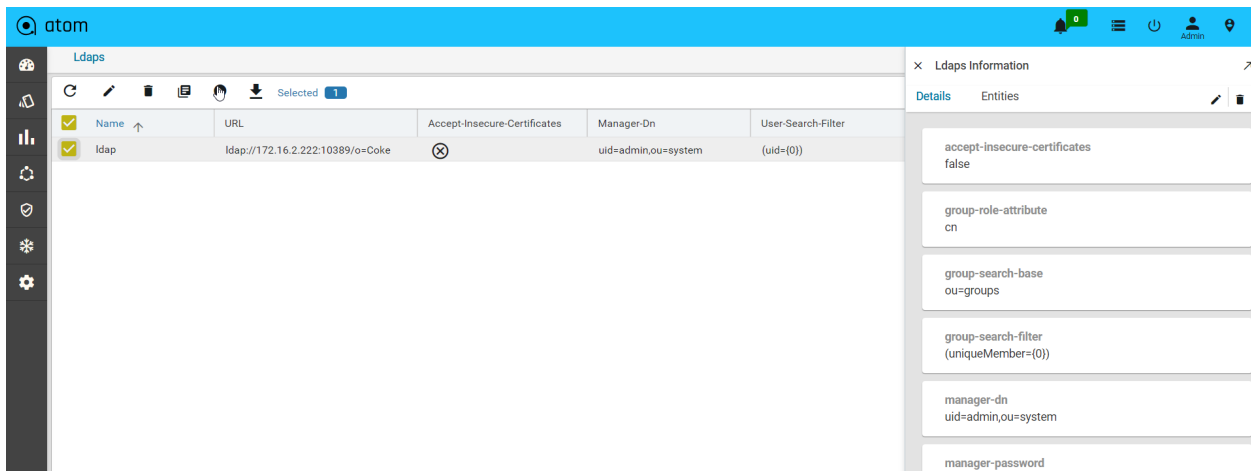
- i. **URL:** Enter the LDAP URL of the tenant. Include port number and base distinguished name (DN)
Example: "ldap://localhost:10389/o=nCloud"
- ii. **Manager Dn:** Type the distinguished name (DN) of LDAP manager. This manager should have at least read permission.
Example: "uid=admin, ou=system"
- iii. **Manager Password:** Type a password for the entered DN. Example: secret
- iv. **User Search Filter:** Specify a search filter. This field determines the query to be run to identify the user record. Always include a query in brackets '(' ')'. Example: "(uid={0})"
- v. **User Search Base:** Specify a relative DN (from the root/base DN) where users are located. Example: "ou=users"

NOTE: In LDAP, {0} is a placeholder (token) for login user ID.

- **Group Search Filter:** This field determines the query to be run to identify the user in a group. Always include a query in brackets '(' ')'. Example: "(uniqueMember={0})"
- **Group Search Base:** Specify a relative DN (from the root/base DN) where user groups are located. Example: "ou=groups"
- **Group Role Attribute:** Specify the attribute name of role in a group. Example: "cn"

- **User-Lastname Attribute:** Attribute that contains user's last name. Example: “sn”
- **User- Firstname Attribute:** Attribute that contains user's first name. Example: “cn”
- **Sync-Freq-in-Hours:** Type the interval, in hours, at which ATOM should query the AD/LDAP directory to schedule an automatic update.
- **Last-Syn-Req-Time:** The timestamp of the last successful synchronization with the LDAP server is displayed.

After adding the LDAP user in ATOM, click **Test Connectivity** to check the connectivity between the ATOM server and LDAP server.

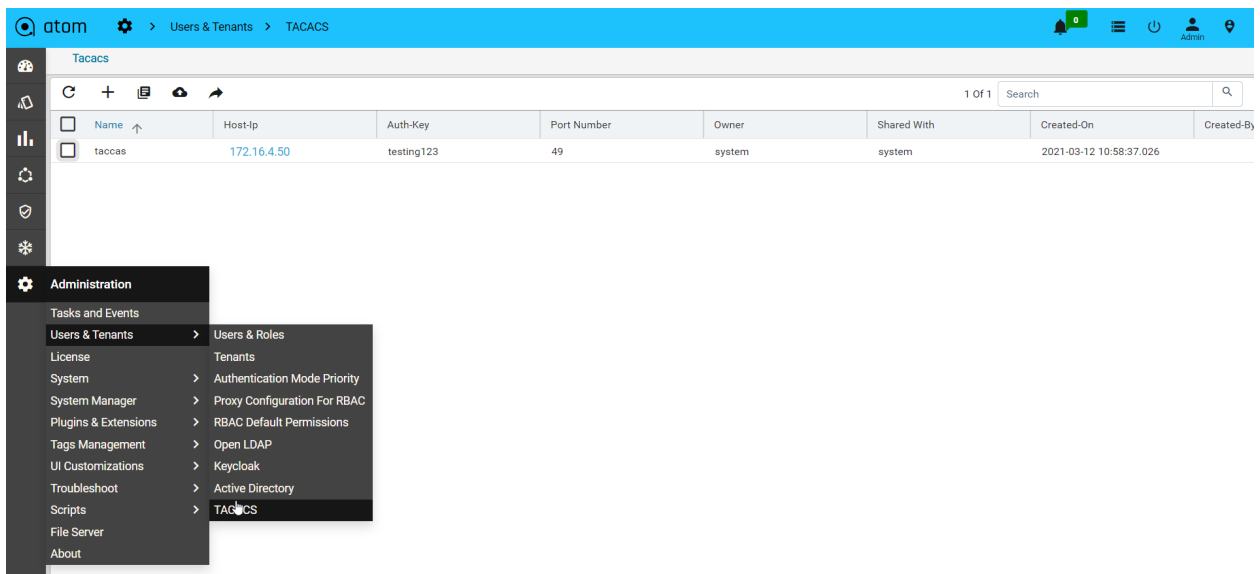


Managing TACACS Users

By integrating TACACS with ATOM, you can achieve a unified authentication system so that the same login credentials (username and password) can be used to access not only for managing network devices but for UNIX and Linux servers too. Therefore, the permissions and privileges to access the devices can be assigned and delegated through ATOM as systems rights.

To integrate the ATOM with TACACS, do the following:

1. Navigate to **Administration > Users & Tenants > TACACS > Add**



Create TACACS

• mandatory information

Name •
taccas

Host-IP •
Must be a valid IP Address. Ex: 172.16.1.24
172.16.4.50

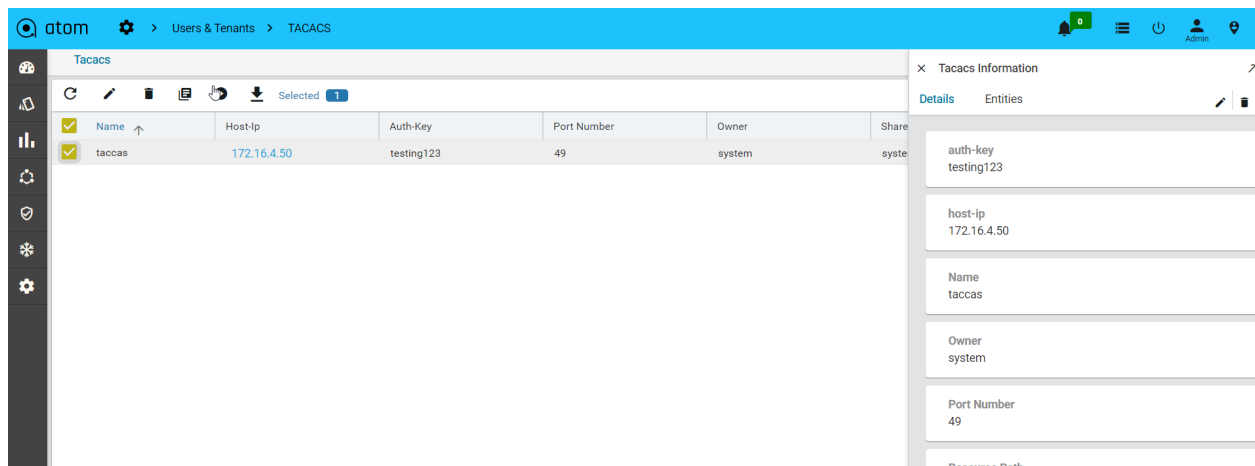
Auth-Key •
Allows Alphanumeric and underscore character only. Max length is 64
testing123

Port Number •
49

Owner •
system

Shared With
system

2. In the Create TACACS screen, enter values in the following fields:
 - i. Host Name: Enter the IP address of the host, which is hosting the TACACS server(172.16.4.50)
 - ii. Auth Key: Enter the key used to specify an encryption key for encrypting and decrypting all traffic between the ATOM server and the TACACS server(testing123)
 - iii. Port Number: Enter the TCP port number to be used when making connections to the TACACS+ daemon. The default port number is 49.
3. Click the Test Connectivity button to test the connection between the ATOM and the TACACS servers(enter the name and the password to validate this connection).



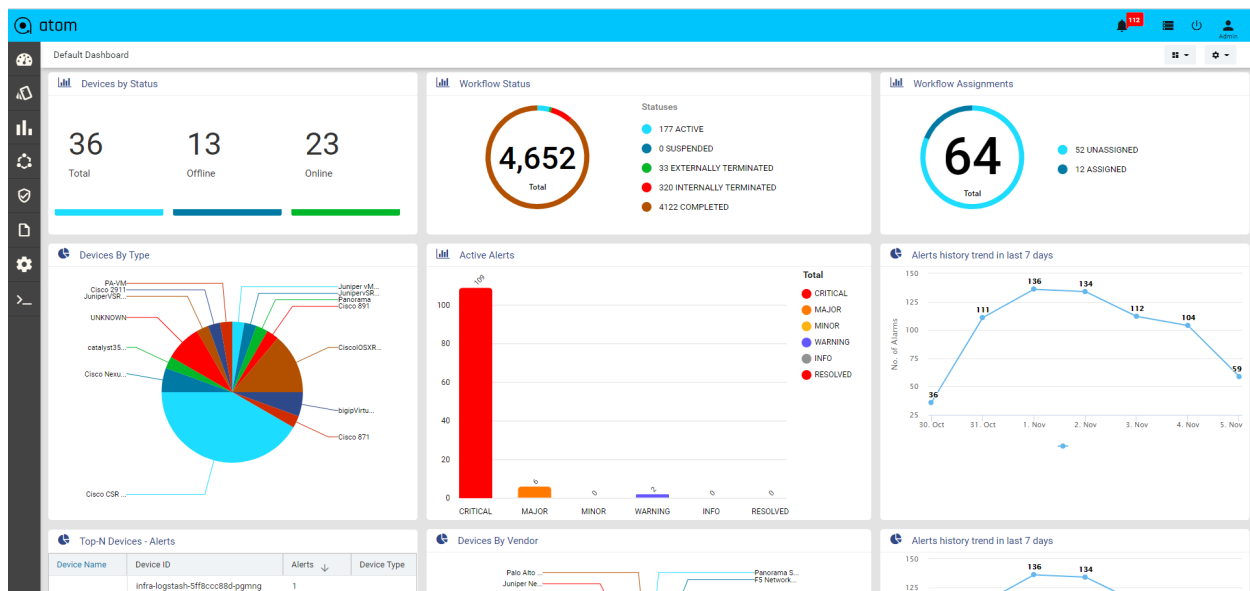
Customizing the Dashboard using DSL

DSL, Domain Specific Language, developed by Anuta can be used for representation and visualization of data derived from the devices managed by ATOM. DSL taps into YANG models, platform or third-party APIs to express code for model pre and post condition specification, rule expressions, rule logic, and RPC implementation logic. Some of the advantages that you can enjoy by implementing DSL are no more manual JAVA or Python code to carry out simple validations, side effect processing, ability to offer richer expressions than xpath 1.0 used by YANG, static analysis of business logic, side- effect analysis, advanced user experience, and ability to change logic on a live system.

Starting from the 7.x release, **Dashboard**, the landing page of ATOM, is organized into dashlets.

A dashlet is an individual component that can be added to or removed from a dashboard. Each dashlet is a reusable unit of functionality, providing a summary of the feature or the function supported by ATOM and is rendered as a result of the custom queries written in DSL.

You can customize the look of the Dashboard, by adding the dashlets of your choice, and dragging and dropping (the extreme right corner of the dashlet) to the desired location on the dashboard.



Each dashlet contains the summary or the overview of the feature or the functionality supported by ATOM.

For example, the dashlet “Device” displays the summary of devices managed by ATOM.

Some of the statistics that can be of the interest in this dashlet could be as follows:

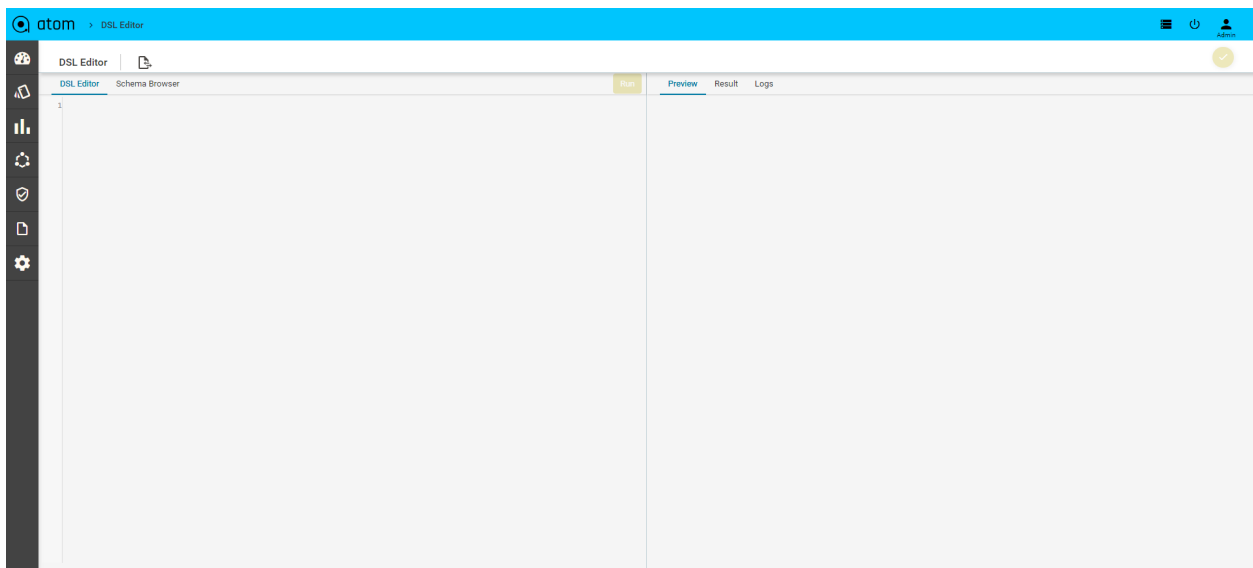
- Total number of devices
- Number of online devices
- Number of offline devices

These statistics can be gathered by ATOM and displayed in the corresponding dashlet depending on the DSL query written for each of them. For information about writing DSL queries, refer the section, “[Writing DSL Queries](#)”

You can save the layout containing the dashlets of your choice and set in a particular order.

Writing DSL Queries

1. For writing any new DSL query in the editor, browse to **Developer Tools > DSL > DSL Editor**.

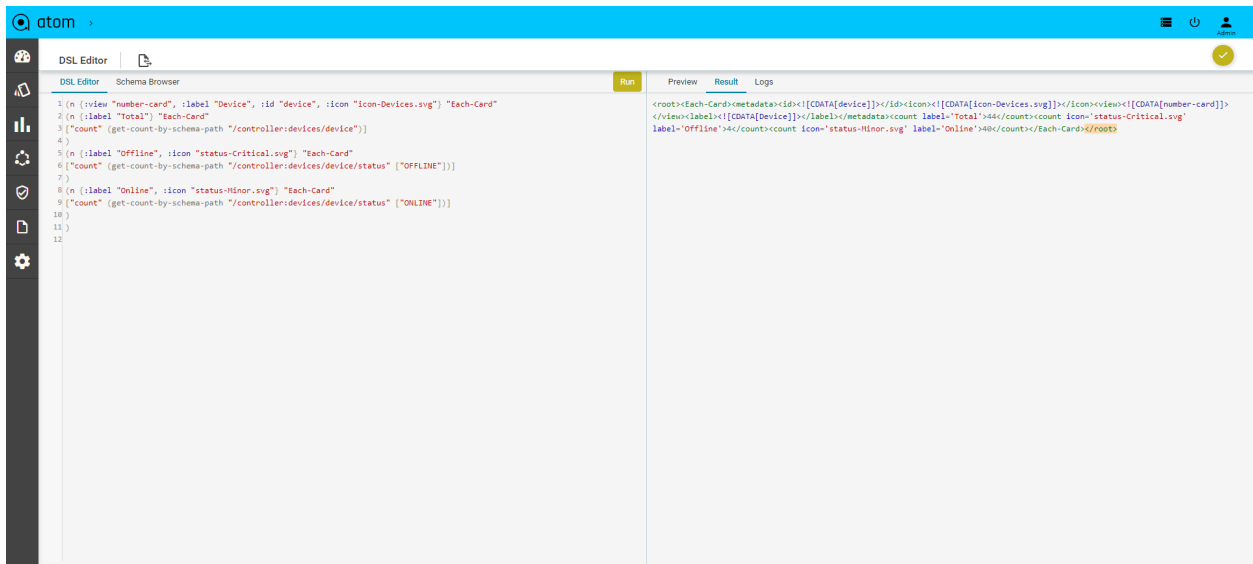


Sample DSL Query

Below is a sample DSL which will display Total/Offline/Online devices in ATOM as a card layout

```
(n {:view "number-card", :label "Device", :id "device", :icon "icon-Devices.svg"} "Each-Card"
  (n {:label "Total"} "Each-Card"
    ["count" (get-count-by-schema-path "/controller:devices/device")]
  )
  (n {:label "Offline", :icon "status-Critical.svg"} "Each-Card"
    ["count" (get-count-by-schema-path "/controller:devices/device/status" ["OFFLINE"])]
  )
  (n {:label "Online", :icon "status-Minor.svg"} "Each-Card"
    ["count" (get-count-by-schema-path "/controller:devices/device/status" ["ONLINE"])]
  )
)
```

2. Click to **Run** button on the top of the editor to check if DSL is working as expected.



Go to the right pane to view the result. There are three tabs in the right side panel **Preview**, **Result** and **Logs**

- Check the **Preview** of layout (Card, Grid, Pie Grid, Pie Chart) for the DSL in Preview tab.
- XML output of the DSL will be shown in the **Result** tab
- All errors of the DSL will be listed in the **Logs** tab in case of any failures.

After the successful execution of a DSL query, you can save and use this as a new Report or incorporate it into Dashboard view. All DSL queries will be saved in **Administration > DSL Queries**.

Customizing the Dashboard

After the successful execution of DSL query, if you want to incorporate the DSL into Dashboard, browse to **Developer Tools > DSL < DSL Queries** to view all the queries as shown below:

The screenshot shows the ATOM DSL Queries interface. The main panel displays a table of queries with columns: Name, Version, Id, Description, and Value. The 'Device' query is selected. The right panel shows the details for the selected query, including category, description, id, is-batch-mode, name, no-of-past-reports-to-show, owner, resource path, and shared with system.

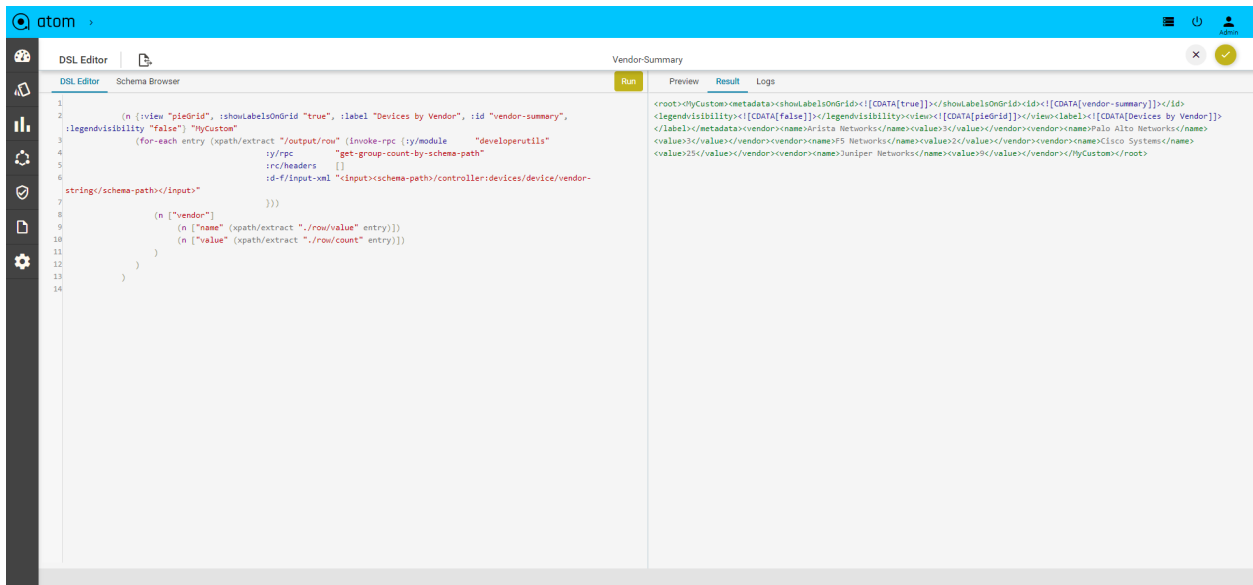
Name	Version	Id	Description	Value
Alerts-history-trend-in-last-7-days	1	Alerts-history-trend-in-last-7-days,1		(n (view 'trendChart', type 'trendChart', title 'Alerts History Trend in Last 7 Days'))
BGP-Neighbor-Shutdown-With-Approval	1	BGP-Neighbor-Shutdown-With-Approval,1	BGP Neighbor Shutdown With Approval	(do (let (alert (from-context 'tick-message')) (do (let (summary (invoke-rgc 'y/module 'tasks') 'y/rp (let (summary (invoke-rgc 'y/module 'tasks') 'y/rp (let (summary (invoke-rgc 'y/module 'tasks') 'y/rp
BGP-Neighbor-Shutdown-With-Out-Approval	1	BGP-Neighbor-Shutdown-With-Out-Approval,1	BGP Neighbor Shutdown With Out Approval	(let (alert (from-context 'tick-message')) (do (let (summary (invoke-rgc 'y/module 'tasks') 'y/rp (let (summary (invoke-rgc 'y/module 'tasks') 'y/rp
Compliant	1	Compliant,1		(n (view 'number-card', label 'Compliant', id 'compliant'))
Device	1	Device,1		(n (view 'number-card', label 'Devices by Status', id 'device'))
Device-And-Collection-Profiles	1	Device-And-Collection-Profiles,1		(n (view 'number-card', label 'Devices And Collection Profiles', id 'device-and-collection-profiles'))
Device-LastComplianceTime-Report	1	Device-LastComplianceTime-Report,1	No. of Reconciliation Devices	(n (view 'grid', title 'Device Last Compliance Time Report', id 'device-last-compliance-time-report'))
Device-Type-Summary	1	Device-Type-Summary,1		(n (view 'pieChart', type 'pieChart', title 'Device Type Summary', id 'device-type-summary'))
Devices-With-Max-Alerts	1	Devices-With-Max-Alerts,1		(n (view 'grid', title 'Top-N Devices - Alerts', id 'devices-with-max-alerts'))
Interface-Shutdown-With-Approval	1	Interface-Shutdown-With-Approval,1	Interface Shutdown With Approval	(do (let (alert (from-context 'tick-message')) (do (let (summary (invoke-rgc 'y/module 'tasks') 'y/rp (let (summary (invoke-rgc 'y/module 'tasks') 'y/rp
Interface-Shutdown-With-Out-Approval	1	Interface-Shutdown-With-Out-Approval,1	Interface Shutdown With Out Approval	(let (alert (from-context 'tick-message')) (do (let (summary (invoke-rgc 'y/module 'tasks') 'y/rp (let (summary (invoke-rgc 'y/module 'tasks') 'y/rp
NOC-B	1	NOC-B,1		(n (view 'vBarChart', label 'NOC A', id 'filter-noc-b'))
Non-Compliance	1	Non-Compliance,1		(n (view 'number-card', label 'Non Compliance', id 'non-compliance'))
Number-Of-Service-Tenants	1	Number-Of-Service-Tenants,1		(n (view 'pieGrid', label 'No. of services per tenant', id 'number-of-service-tenants'))
Recent-Alerts	1	Recent-Alerts,1		(n (view 'grid', title 'Recent Alerts', id 'recent-alerts'))
Reconciliation-Report	1	Reconciliation-Report,1	No. of Reconciliation Devices	(n (view 'number-card', label 'Device Compliance', id 'reconciliation-report'))
SSLcertificate-report	1	SSLcertificate-report,1	SSL Certificates	(n (view 'grid', title 'SSL Certificates Report', id 'ssl-certificate-report'))
Task-Summary	1	Task-Summary,1		(let (summary (invoke-rgc 'y/module 'tasks') 'y/rp (let (summary (invoke-rgc 'y/module 'tasks') 'y/rp (let (summary (invoke-rgc 'y/module 'tasks') 'y/rp
Telemetry-Alerts-And-Actions	1	Telemetry-Alerts-And-Actions,1		(n (view 'number-card', label 'Telemetry Alerts And Actions', id 'telemetry-alerts-and-actions'))
Vendor-Summary	1	Vendor-Summary,1		(n (view 'pieChart', type 'pieChart', title 'Device Vendor Summary', id 'vendor-summary'))
acl-firewall-report	1	acl-firewall-report,1	ZoneBased Firewall ACL Report	(n (view 'grid', title 'ZoneBased FW ACL Report', id 'acl-firewall-report'))
actions-summary	1	actions-summary,1	Actions Summary	(n (view 'grid', title 'Triggered Action Based Report', id 'actions-summary'))
alert-summary	1	alert-summary,1		(n (view 'vBarChart', label 'Active Alerts', id 'alert-summary'))
assurance-summary	1	assurance-summary,1	Assurance Summary	(n (view 'grid', title 'Assurance Summary', id 'assurance-summary'))
bgp-neighbor-report	1	bgp-neighbor-report,1	To listout BGP Neighbor details across devices	(n (view 'grid', title 'BGP Neighbor Report', id 'bgp-neighbor-report'))

Click on view/download option in dsl query, it can be shown as xml/json/csv/form template.

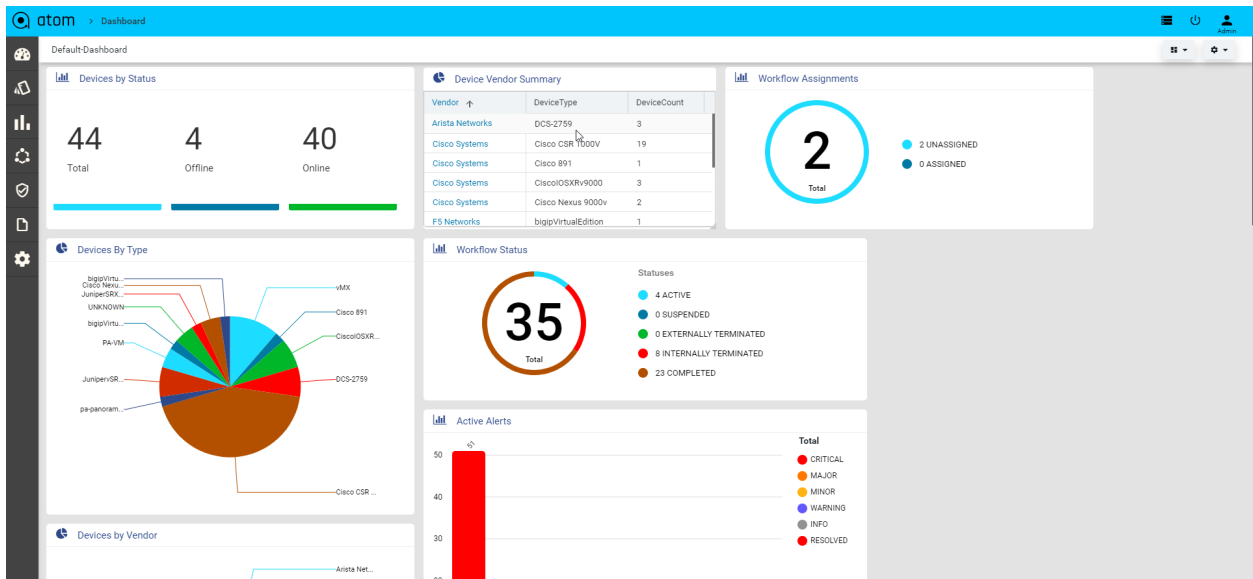
The screenshot shows the ATOM DSL Queries interface. The main panel displays a table of queries with columns: Name, Version, Id, and Description. The 'Device' query is selected. The right panel shows the details for the selected query, including category, description, id, is-batch-mode, name, no-of-past-reports-to-show, owner, resource path, and shared with system.

Name	Version	Id	Description
3.36_test	1	3.36_test	
Alerts-history-trend-in-last-7-days	1	Alerts-history-trend-in-last-7-days,1	
BGP-Neighbor-Shutdown-With-Approval	1	BGP-Neighbor-Shutdown-With-Approval,1	BGP Neighbor Shutdown With Approval
BGP-Neighbor-Shutdown-With-Out-Approval	1	BGP-Neighbor-Shutdown-With-Out-Approval,1	BGP Neighbor Shutdown With Out Approval
Compliant	1	Compliant,1	
Device	1	Device,1	
Device-And-Collection-Profiles	1	Device-And-Collection-Profiles,1	
Device-LastComplianceTime-Report	1	Device-LastComplianceTime-Report,1	No. of Reconciliation Devices
Device-Type-Summary	1	Device-Type-Summary,1	
Devices-With-Max-Alerts	1	Devices-With-Max-Alerts,1	
Interface-Shutdown-With-Approval	1	Interface-Shutdown-With-Approval,1	Interface Shutdown With Approval
Interface-Shutdown-With-Out-Approval	1	Interface-Shutdown-With-Out-Approval,1	Interface Shutdown With Out Approval
NOC-B	1	NOC-B,1	
Non-Compliance	1	Non-Compliance,1	
Number-Of-Service-Tenants	1	Number-Of-Service-Tenants,1	
Recent-Alerts	1	Recent-Alerts,1	
Reconciliation-Report	1	Reconciliation-Report,1	No. of Reconciliation Devices
SSLcertificate-report	1	SSLcertificate-report,1	SSL Certificates

Include your new DSL and click the **Save** button on the right side panel of the Editor.



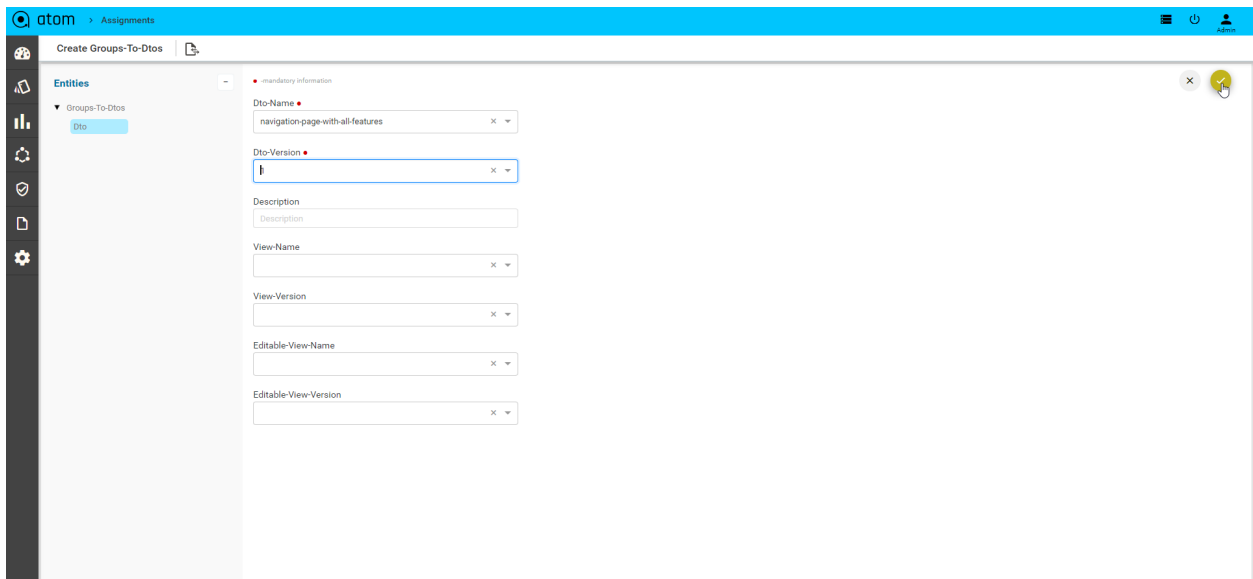
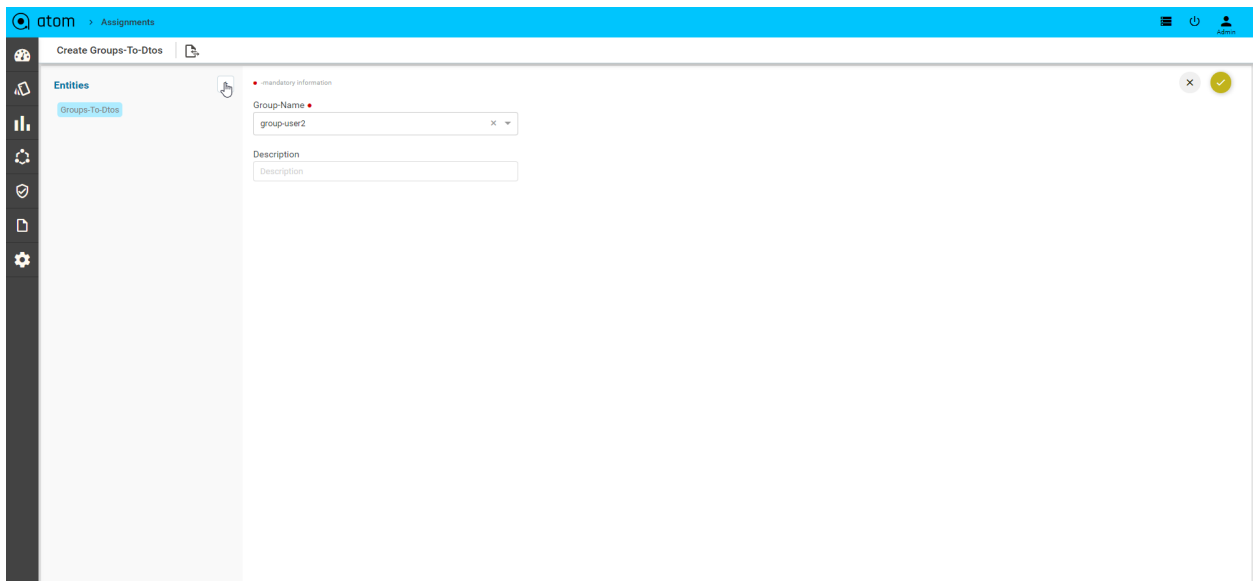
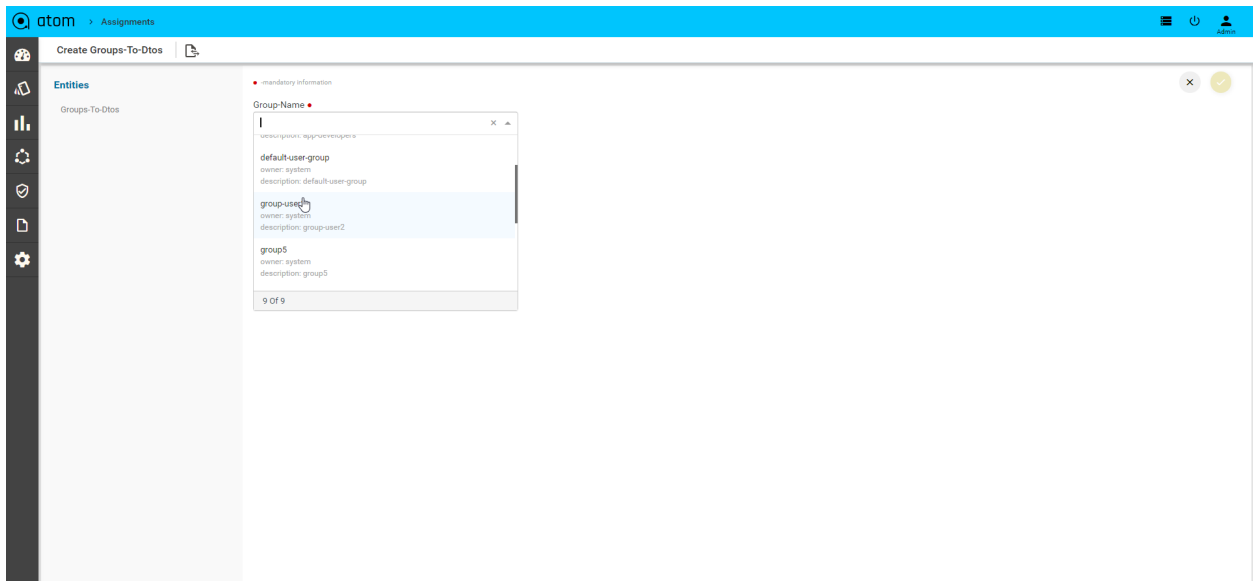
Now you can browse to Dashboard to view the report that has been included (due to the new DSL query that was added) in the Dashboard DSL.



DSL Assignment

The define the dsl assignment is associated with the group and dto is a transfer of the object(to get the navigation menu items after login user).

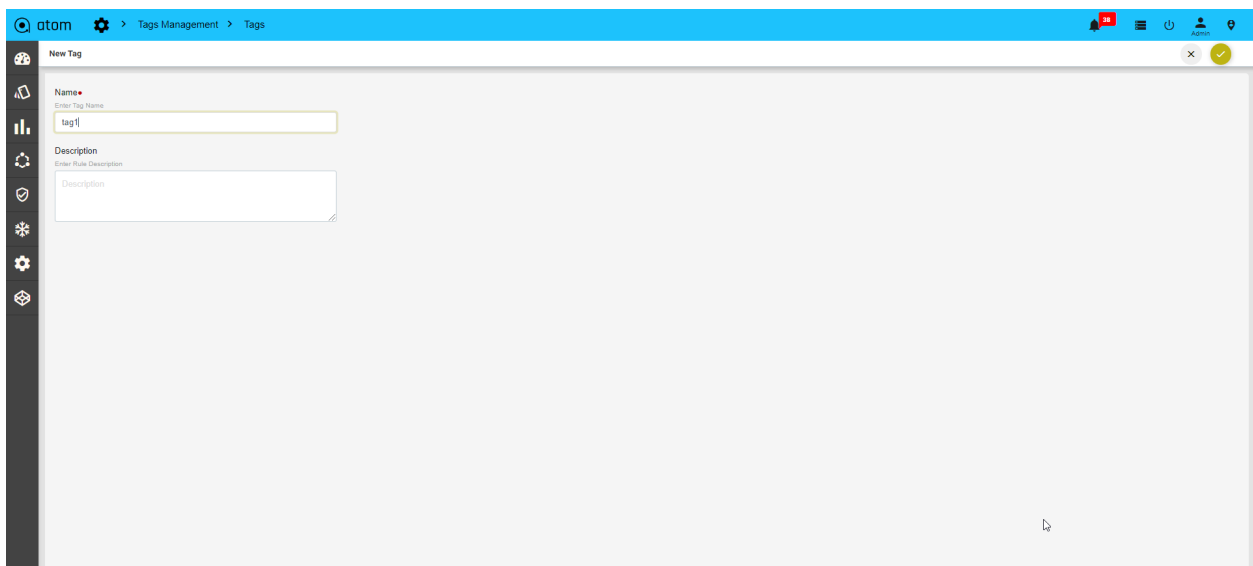
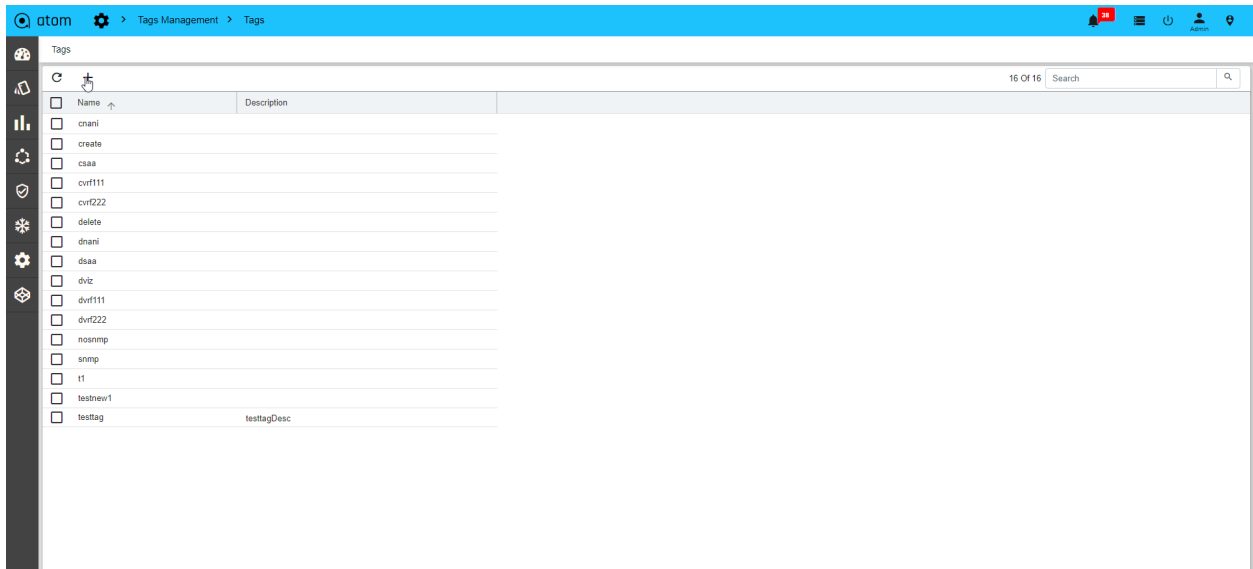
Navigate > Developer Tools > DSL > DSL Assignment > Add



Tag Management

Navigate > Administration > Tag Management

By creating the tags in global, it should show in create alert rule definition

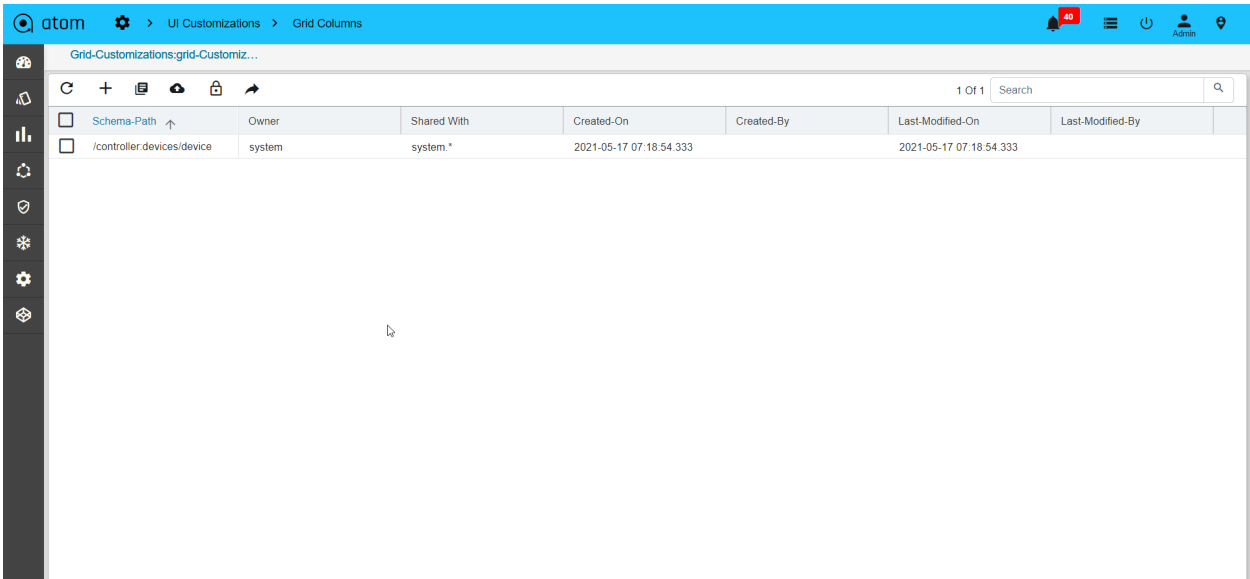
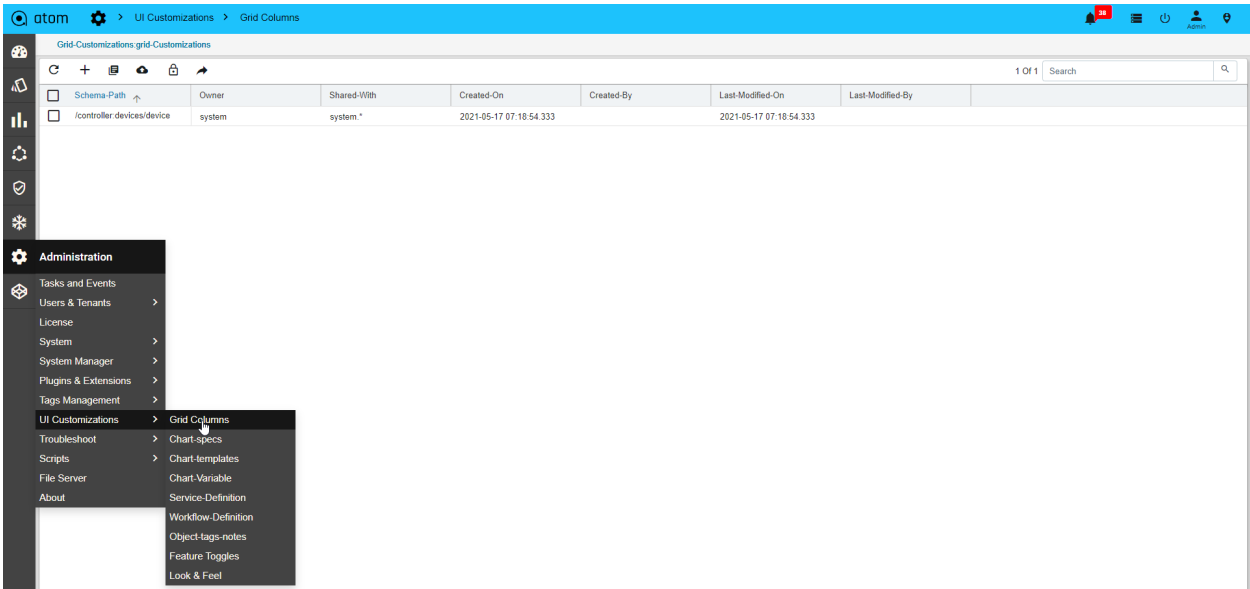


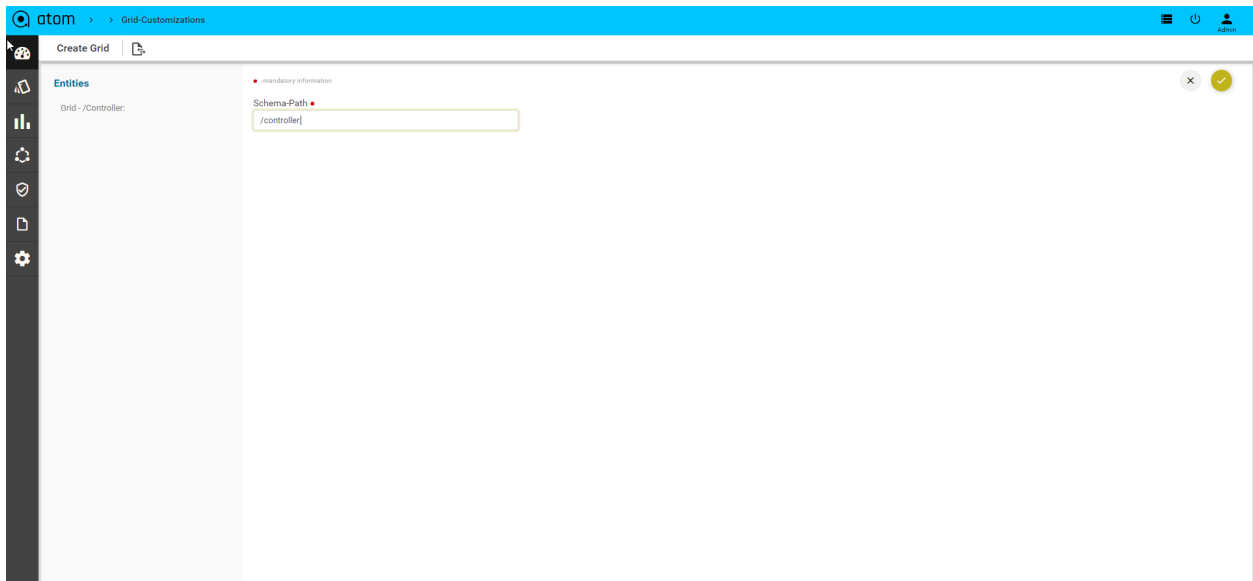
UI Customizations

Customization give control to the user. Customization may involve moving items around an interface to reflect the users' priorities

1. Grid column: Grid columns allows us to select which grid columns can be shown for each entity. Current by default we have one for devices now in grid column.

Navigate > Administration > UI customization > Grid column > Add





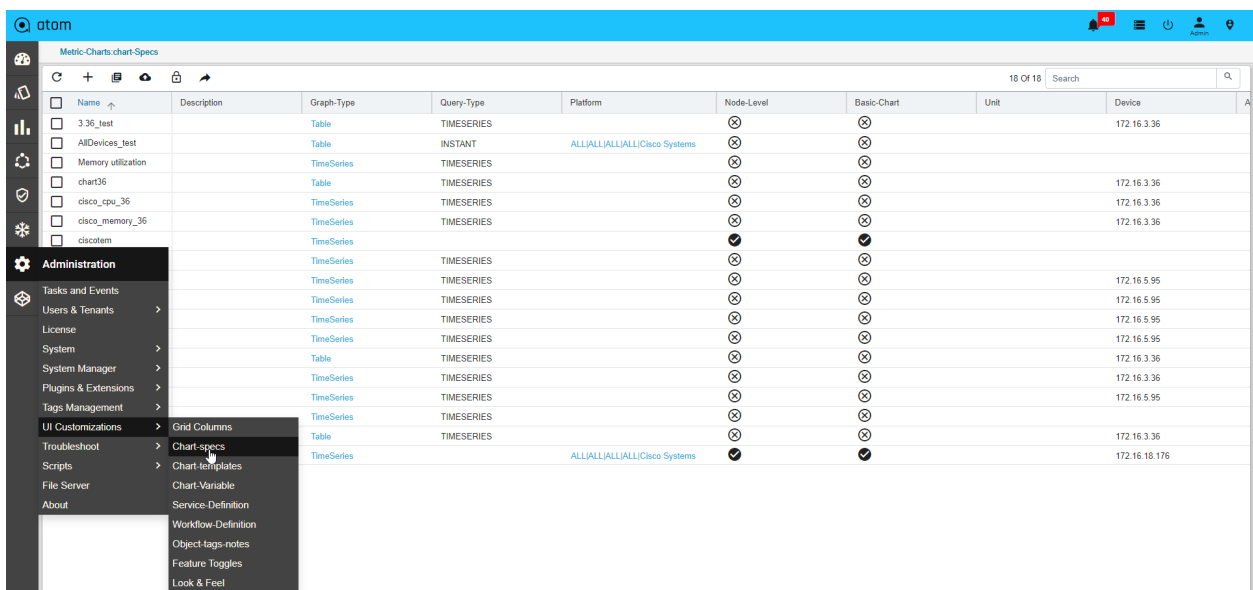
2.Chart-specs:

A Chart Spec defines the query to be used, variables used in the query and the intended graph type. For example, a chart spec can be written for a TopNUtilizedInterfaces report. The query would use variables for device and the N.

An admin may decide to create two invocations of this chart spec and make them available readily; one for top 5 and one for top 10. Admin can do so by creating 2 chart-invocation payloads and give 25 and 50 for the N value. An end user can run these two different charts out of the box. In addition, end user can tweak the parameter values and explore the graphs.

If a user decides to save the changed chart invocations it is a simple matter of changing the corresponding chart invocation objects.

Navigate > Administration > UI customization > chart -specs > Add



atom

Metric-Charts: chart-Specs

18 Of 18 Search

Name	Description	Graph-Type	Query-Type	Platform	Node-Level	Basic-Chart	Unit	Device
<input type="checkbox"/> 3_36_test		Table	TIMESERIES		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		172.16.3.36
<input type="checkbox"/> AllDevices_test		Table	INSTANT	ALL/ALL/ALL/Cisco Systems	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
<input type="checkbox"/> Memory utilization		TimeSeries	TIMESERIES		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
<input type="checkbox"/> chart36		Table	TIMESERIES		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		172.16.3.36
<input type="checkbox"/> cisco_cpu_36		TimeSeries	TIMESERIES		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		172.16.3.36
<input type="checkbox"/> cisco_memory_36		TimeSeries	TIMESERIES		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		172.16.3.36
<input type="checkbox"/> ciscotem		TimeSeries			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
<input type="checkbox"/> cpu utilization		TimeSeries	TIMESERIES		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
<input type="checkbox"/> cpu_95		TimeSeries	TIMESERIES		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		172.16.5.95
<input type="checkbox"/> disc_95		TimeSeries	TIMESERIES		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		172.16.5.95
<input type="checkbox"/> interface_95		TimeSeries	TIMESERIES		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		172.16.5.95
<input type="checkbox"/> memory_95		TimeSeries	TIMESERIES		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		172.16.5.95
<input type="checkbox"/> new1		Table	TIMESERIES		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		172.16.3.36
<input type="checkbox"/> temp		TimeSeries	TIMESERIES		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		172.16.3.36
<input type="checkbox"/> temp_95		TimeSeries	TIMESERIES		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		172.16.5.95
<input type="checkbox"/> test1		TimeSeries	TIMESERIES		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
<input type="checkbox"/> test36cpu		Table	TIMESERIES		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		172.16.3.36
<input type="checkbox"/> test_Table		TimeSeries		ALL/ALL/ALL/Cisco Systems	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		172.16.18.176

atom > Chart-Specs

Create Chart-Spec

Entities

Chart-Spec - Basic Chart-Spec

Name • Basic-chart-spec

Description Description

Graph-Type Data.Tables

Platform ALL/ALL/ALL/Cisco Systems

Node-Level

Basic-Chart ☒

Unit unit

Device-Group

20 Of 20 search

☐ Device-Group

☐ AllDevices

☐ Firewall

☐ Host

☐ Layer 2 switch

☐ Layer 2/3 switch

atom > Chart-Specs

Create Chart-Spec

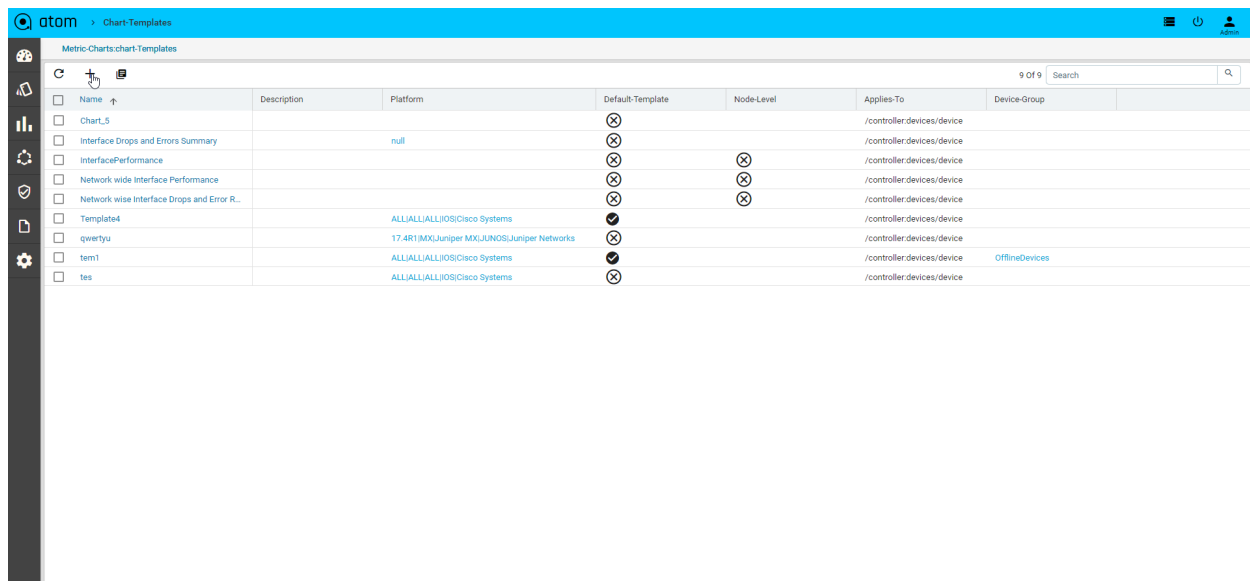
Entities

Chart-Spec - Basic Chart-Spec

3.Chart-Template:

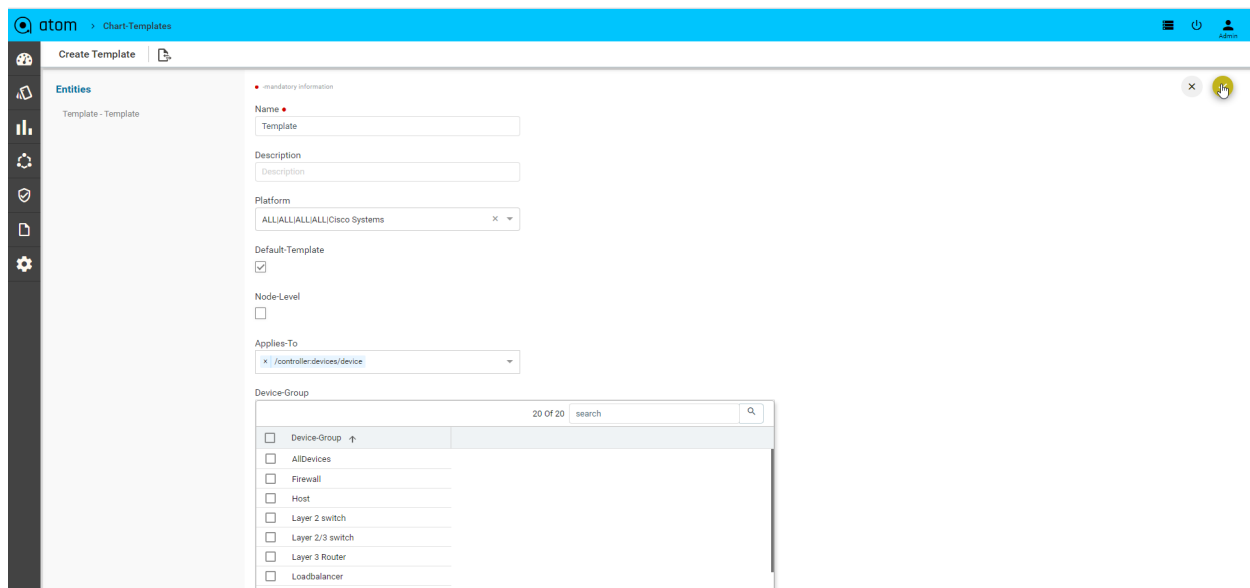
To define the template is a collection of custom charts.create the multiple basic and advanced custom charts can be grouped into one template.it should show the chart graph in monitoring

Navigate > Administration > UI customization > chart -template >Add



The screenshot shows the 'Chart-Templates' page in the ATOM interface. It features a table with columns: Name, Description, Platform, Default-Template, Node-Level, Applies-To, and Device-Group. The table lists several templates, including 'Chart_5', 'Interface Drops and Errors Summary', 'InterfacePerformance', 'Network wide Interface Performance', 'Network wise Interface Drops and Error R...', 'Template4', 'qwertyu', 'tsm1', and 'tes'. Each row has checkboxes for 'Default-Template' and 'Node-Level', and a dropdown for 'Applies-To'. The 'Device-Group' column shows 'OfflineDevices' for the 'tsm1' template.

Name	Description	Platform	Default-Template	Node-Level	Applies-To	Device-Group
Chart_5			<input checked="" type="checkbox"/>		/controller.devices/device	
Interface Drops and Errors Summary		null	<input checked="" type="checkbox"/>		/controller.devices/device	
InterfacePerformance			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	/controller.devices/device	
Network wide Interface Performance			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	/controller.devices/device	
Network wise Interface Drops and Error R...			<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	/controller.devices/device	
Template4		ALLJALLJALLIOS/Cisco Systems	<input checked="" type="checkbox"/>		/controller.devices/device	
qwertyu		17.4R1(MX)Juniper MXJUNOS/Juniper Networks	<input checked="" type="checkbox"/>		/controller.devices/device	
tsm1		ALLJALLJALLIOS/Cisco Systems	<input checked="" type="checkbox"/>		/controller.devices/device	OfflineDevices
tes		ALLJALLJALLIOS/Cisco Systems	<input checked="" type="checkbox"/>		/controller.devices/device	



The screenshot shows the 'Create Template' page in the ATOM interface. It includes a sidebar with 'Entities' and a main form area. The form has fields for Name, Description, Platform (a dropdown menu), Default-Template (a checkbox), Node-Level (a checkbox), Applies-To (a dropdown menu), and Device-Group (a dropdown menu). The 'Device-Group' dropdown is open, showing a list of options: AllDevices, Firewall, Host, Layer 2 switch, Layer 2/3 switch, Layer 3 Router, and Loadbalancer.

4.Chart-variables: To define a Variables are useful when the user is writing the queries to build a custom chart.Global variable which can be reusable across multiple charts

Here Two types of variables are Constant and Query,Query type is useful to fetch a list of entities like devices, interfaces etc.and constant is for any static values like bucket interval etc.

Navigate > Administration > UI customization > chart -variables >Add

atom

Metric-Charts: variables

2 Of 2

Search

<input type="checkbox"/>	Name	Description	Default-Value	System-Defined	Type	Metric	Filter	Tsdb-Query	Owner	Shared
<input type="checkbox"/>	device			✓		device	\$device		system	system
<input type="checkbox"/>	ifDescr			✓		ifDescr	{'device': '\$device'}		system	system

Administration

Tasks and Events

Users & Tenants

License

System

System Manager

Plugins & Extensions

Tags Management

UI Customizations

Troubleshoot

Scripts

File Server

About

Grid Columns

Chart-specs

Chart-templates

Chart-Variable

Service-Definition

Workflow-Definition

Object-tags: notes

Feature Toggles

Look & Feel

atom

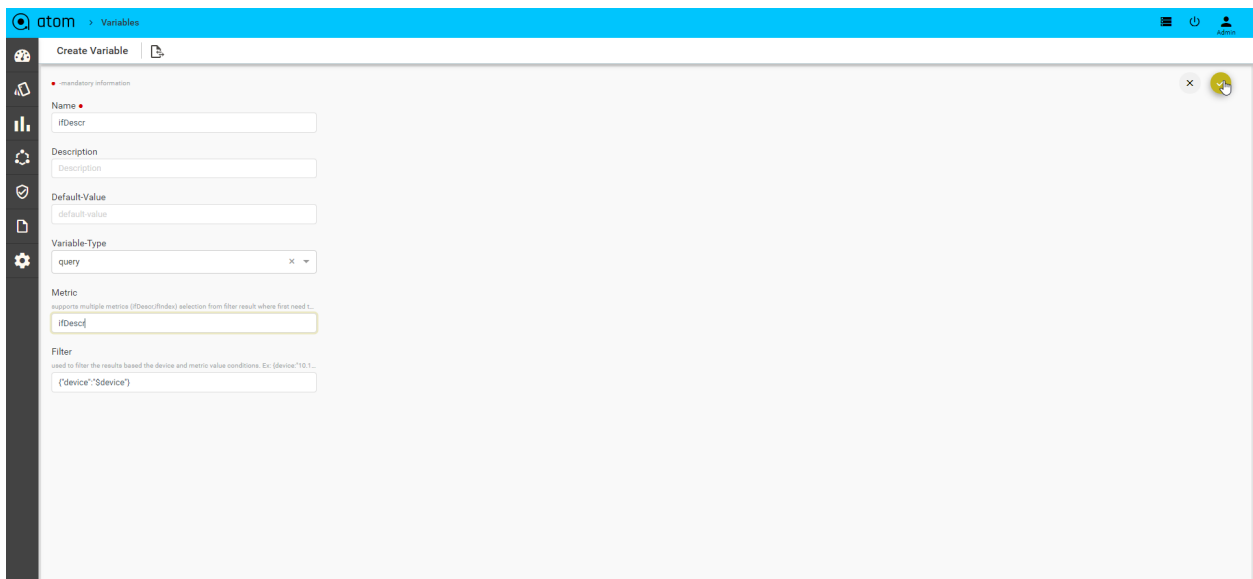
Variables

Metric-Charts: variables

6 Of 6

Search

<input type="checkbox"/>	Name	Description	Default-Value	System-Defined	Type	Metric	Filter
<input type="checkbox"/>	device			✓		device	\$device
<input type="checkbox"/>	deviceA			✗		device	\$device
<input type="checkbox"/>	deviceB			✗		device	\$device
<input type="checkbox"/>	ifDescr			✓		ifDescr	{'device': '\$device'}
<input type="checkbox"/>	ifDescrA			✗		ifDescr	{'device': '\$deviceA'}
<input type="checkbox"/>	ifDescrB			✗		ifDescr	{'device': '\$deviceB'}



Troubleshoot

From the Troubleshoot tab, the system administrator can perform the following tasks:

- ["Services & Metrics"](#)
- ["Queue Statistics"](#)
- ["Device Comm and Inv"](#)

Services & Metrics :

The administrator can view a summary of the health of the server and the associated agents. All the services running on the ATOM server are also displayed here.

1. Go to **Administration > Troubleshoot > Services and Metrics**
2. In the left pane, click **Servers > Components** to view the different categories of the servers running in ATOM.
3. Click **Servers > System Health** in the right pane to view the health of the components of the server.

The health of the associated services is also displayed in the lower pane.

4. Click the **Servers** icon to view the Services in the right pane.
5. Select a service that is running and view the Statistics and Events associated with the service in the bottom panel.

Queue Statistics :

ATOM uses a bus to communicate various events between the different ATOM modules or components, thereby providing a view of the activity of the Naas Bus.

- Broker Statistics
- Naas Bus Monitor Statistics

Each row represents a particular event that ATOM components publish or subscribe to, thereby help in monitoring the Bus.

Device Comm and Inv

Ping

As an administrator, you can check the reachability of a device from ATOM using Ping Test.

1. Navigate to **Administration > Troubleshoot > Device Comm & In > Ping Test**
2. To create a **Ping Test**, fill the fields described below:
 - **IP Address:** Enter the IP address of a device that needs to be verified for its reachability.
 - **Packet Count:** Enter the number of ICMP Echo request messages to be sent.
 - **Time Out (sec):** Specify a value for the time for which the ping command should wait for each reply.

SNMP

SNMP Tests You can test if SNMP devices are responding correctly to SNMP queries. By default, SNMP v2c is supported.

1. Navigate to **Administration > Troubleshoot > Device Comm & In > SNMP**
2. To create an SNMP test, fill the following fields:
 - **Device Name:** Enter the name of the device
 - **SNMP Operation:** Choose an appropriate SNMP operation from the drop- down menu:
 - GET
 - GET NEXT
 - GET BULK
 - WALK
 - **SNMP OID:** Enter the OID of the SNMP device

SNMP TEST

Mandatory information

Device Name

Select a device name

csr3.30.anutacorp.com/172.16.3.30

SNMP Operation

Select a SNMP Operation

Get

SNMP OID

Enter a Oid value

1.3.6.1.2.1.1.0

SNMP Result

Agent :	default_agent
Status :	Success
Error Code :	0
Error Index :	0
Error Message :	
Result :	1.3.6.1.2.1.1.0 -> Cisco IOS Software, CSR1000V Software (X86_64_LINUX_IOSD-UNIVERSALK9-M), Version 15.6(1)S, RELEASE SOFTWARE (fc4) Technical Support: http://www.cisco.com/techsupport Copyright (c) 1986-2015 by Cisco Systems, Inc. Compiled Wed 25-Nov-15 15:02 by mcpre

Config Parser

This utility helps you check the parsed output of any running configuration of any device. By doing so, you can verify the extent to which ATOM supports the config parsing for a given running configuration. All the parsed configuration can also be visualized in the supported data models in ATOM.

1. Navigate to **Administration > Troubleshoot > Device Comm & Inv > Config Parse**

atom

Configuration

Config Parsing Report

Vendor

Cisco Systems

OS Type

IOSXE

Device Family

Cisco CSR 1000V

Device Type

Cisco CSR 1000V

OS Version

ALL

Running Config

```

service timestamps debug datetime msec
service timestamps log datetime msec
!
hostname CSR10001
!
bootstart-marker
bootend-marker
!
!
vrf definition 12
!
vrf definition east
!
address-family ipv4
ext-address-family
!
vrf definition 101
!
address-family ipv4
ext-address-family
!
vrf definition 102
!
address-family ipv4
ext-address-family
!
vrf definition 103
!
address-family ipv4
ext-address-family
!
vrf definition test
!
address-family ipv4

```

Parsed Data

Parsed Config

Parsing Errors

Unsupported Config

```

{
  "subject": {
    "parent-output": {
      "result": {
        "agent": "controller/devices/device",
        "parsed-data": {
          "device": {
            "basicDeviceConfig": {
              "http://anutanetworks.com/basicDeviceConfig",
              "device": {
                "basicDeviceConfigShell-option": {
                  "shell-option": "enable"
                }
              }
            }
          }
        }
      }
    }
  },
  "agent": "controller/devices/device",
  "parent-data": {
    "device": {
      "basicDeviceConfig": {
        "http://anutanetworks.com/basicDeviceConfig",
        "device": {
          "basicDeviceConfigAgent": {
            "agent-community-new": false
          }
        }
      }
    }
  },
  "agent": "controller/devices/device",
  "parent-data": {
    "device": {
      "basicDeviceConfig": {
        "http://anutanetworks.com/basicDeviceConfig",
        "device": {
          "basicDeviceConfigHostname": {
            "hostname": "CSR10001"
          }
        }
      }
    }
  },
  "agent": "controller/devices/device/3Features/vrf",
  "parent-data": {
    "vrf": {

```

Parsing Report Generated Successfully

2. Select the device family details for which the config parsing support needs to be verified in ATOM:

- i. **Vendor:** Select the vendor from the supported vendor list in ATOM
- ii. **OS Type:** Select the OS type for the device vendor for which the config parsing needs to be checked
- iii. **Device Family:** Select the device family that the device belongs to
- iv. **Device Type:** Select the type of the device belonging to the selected device family
- v. **OS Version:** Select the version of the OS
- vi. In the Running Config pane, paste the running configuration of the device for which config parsing needs to be verified
- vii. Click **Submit** to generate the Config Parsing Report

The results of the Report can be viewed in the right pane as

- **Parsed Data:** The configuration, which is parsed in ATOM, for which the data model is available can be viewed in this tab.
- **Parsed Configurations:** The running configuration that is parsed into blocks by ATOM can be viewed in this tab.
- **Parsing Errors:** The parsed configuration derived in ATOM but with errors can be viewed in this tab.
- **Unsupported Configurations:** The running configuration for which there is no parsing support available in ATOM can be viewed in this tab

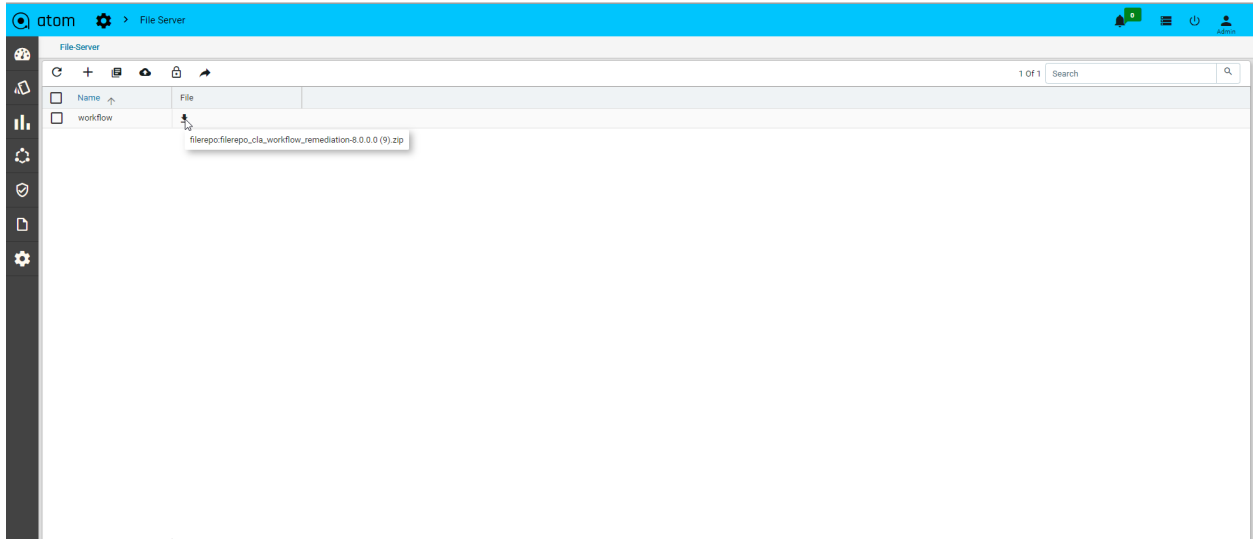
File Server

File server helps users to upload and can also down the packages related to the services or any workflow if any user wants to save a package in the repository.

Navigate to **Administration > File Server** and click on the add button.

Name: Enter a name of the package

Choose File: Upload a package or a file which needs to be saved.



About

Navigate > Administration > About

