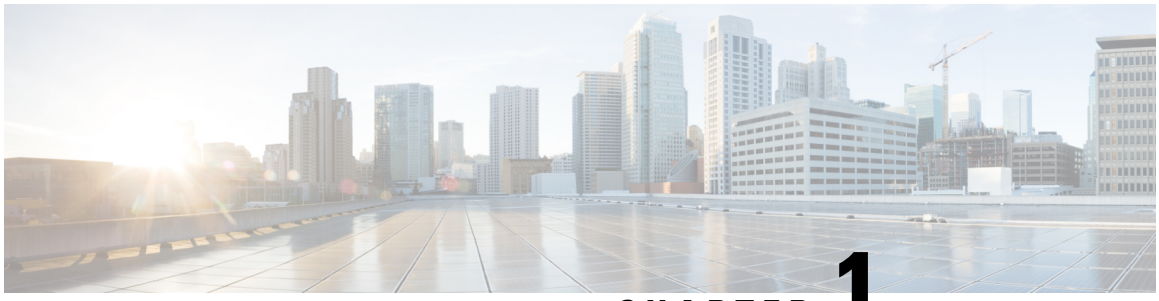
Cisco Cloud Native Broadband Router User's Guide, Release 20.2

First Published: 2020-07-23

Americas Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
http://www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883

CONTENTS



CHAPTER 1

Overview of Cisco Cloud Native Broadband Router

This chapter provides an overview of Cisco Cloud Native Broadband Router (cnBR) and its key features and benefits. It also describes the key components of the Cisco cnBR and how the router is deployed in a network.

Transformation of the Cable Network

To support the increasing needs of the customers, cable networks are undergoing major transformations. They are:

- migrating from analog to digital systems
- adding capacity and scale
- deploying new and improved service features

Replacing analog systems with digital devices, such as Remote PHY and Converged Interconnect Network (CIN) routers and switches, is preparation for what is to come: the transformation of the cable headend. With a digital access network, cable services that are reliant on headend hardware are no longer tied to physical hardware–based solutions.

The Cisco Cloud Native Broadband Router (cnBR) is a fundamental rewrite of the CCAP, virtualizing the earlier hardware-based services with a truly cloud-native design, thus offering unprecedented service velocity, highly simplified operations, and economic scalability for profitably operating your network. The Cisco cnBR is built from the ground up, taking decades of experience and expertise in networking technologies and completely rewriting the hardware-based Converged Cable Access Platform (CCAP) code to be cloud native. Instead of lifting and shifting existing code from legacy hardware and placing it in the cloud to run as a virtual machine, the Cloud Native Broadband Router is a full software rewrite for CCAP-enabled services, built as a composable set of microservices that utilize standard tools, such as Kubernetes for container orchestration and Docker for creating, deploying, and running containerized applications.

Features and Benefits of Cisco cnBR

The previous generations of Cable Modem Termination Systems (CMTS) products integrated cable modem RF connectivity, Data-over-Cable Service Interface Specifications (DOCSIS) control plane signaling, data forwarding, platform monitoring, and back office reporting into a single purpose-built hardware platform. The Cisco cnBR is a containerized, virtual CCAP solution, which is designed to take the service capabilities of physical hardware and virtualize them into a customizable, scalable, and resilient set of microservices.

The Cisco cnBR offers the following features and benefits:

- **Increased feature velocity**: The increased feature velocity is achieved by hosting the functionality on more generic hardware platforms, making it easier to develop and test features as well as leverage Open Source Software and continuous integration technologies.
- **Flexible placement of CMTS Core and PHY**: With the Cisco cnBR on general-purpose hardware and physically not containing the PHY interface, the CMTS Core can be deployed anywhere there is network connectivity to the RPDs and service provider IP network.
- **Enhanced monitoring**: With the Cisco cnBR and Operations Hub deployed on a container platform, industry leading monitoring technologies like Prometheus and ELK are readily accessible and easy to deploy.
- **Easier scaling**: Scaling up the Cisco cnBR in a datacenter is as easy as adding new cnBR service containers on existing or new clusters.
- **Rapid feature and configuration deployment**: By employing CI/CD tools in combination with a container platform, new features can be quickly tested and deployed in the service provider network.
- **DevOps support**: Increased monitoring visibility, CI/CD capabilities, use of industry-standard container platforms, and the need to keep the deployment updated, paves the way for DevOps support and tools. The product is more visible and technologically understandable by the service provider, thus allowing for a partnered support model.
- **Increased automation**: The kubernetes (K8S) platform has been designed to make automation easier, further reducing operational cost.

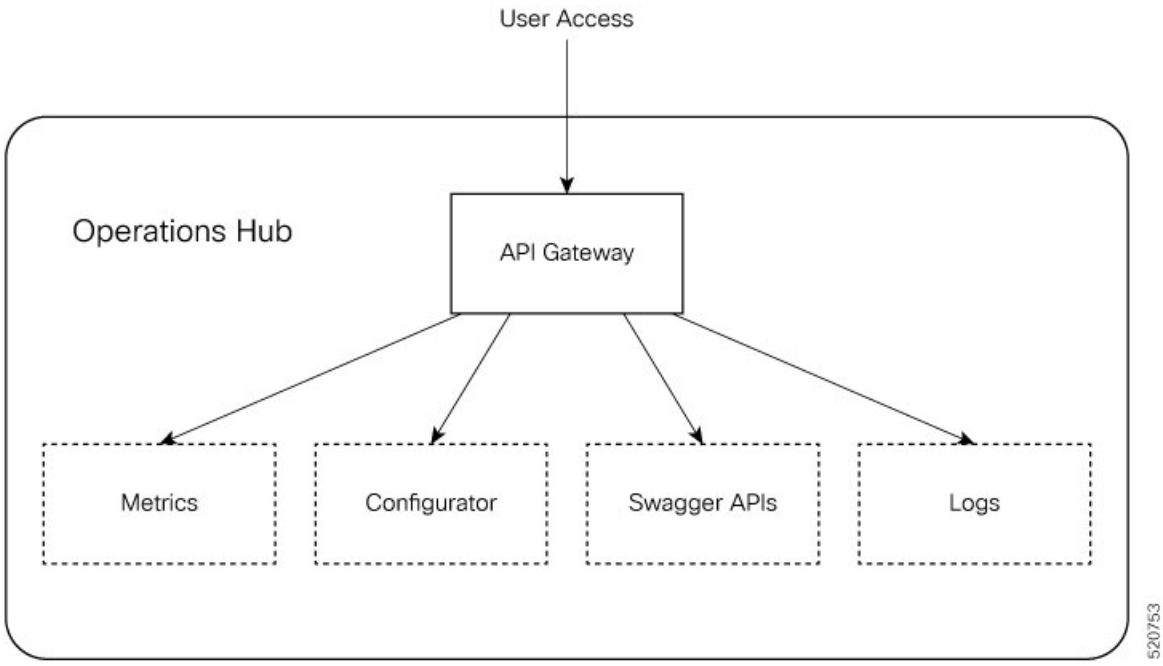
Cisco cnBR Product Components

The key components of the Cisco cnBR are:

Cisco Operations Hub

Cisco Operations Hub is the operations management tool for configuring, monitoring, and troubleshooting the Cisco cnBR. It is the tool for integrating Cisco cnBR into business and operation systems. As each Cisco cnBR operates in a datacenter, Cisco Operations Hub monitors the platform, the CMTS service health, and provides central external management access to IPDR, logging, and telemetry data.

Figure 1: Cisco Operations Hub Framework



Cisco Operations Hub provides these interfaces to manage Cisco cnBR features:

- Cisco cnBR Metrics: a GUI to view various health metrics and other information about Cisco Operations Hub, Cisco cnBR, and DOCSIS network and elements.
- Configure cnBR using Configurator: a GUI to view and change Cisco cnBR and Cisco Operations Hub configurations.
- Swagger API: a programming interface to retrieve configuration, metrics and other information about Cisco Operations Hub, Cisco cnBR, and DOCSIS network and elements.
- Logs: a GUI to view Debug Logs of Cisco cnBR and Cisco Operations Hub.

See Configure Operations Hub, on page 29 for information on accessing and configuring Cisco Operations Hub.

Besides the management of various Cisco cnBR capabilities, Cisco Operations Hub provides these interfaces for Customer OSS or Third-Party Vendor tool integrations:

- IP Detail Record Service
- Simple Network Management Protocol

cnBR Core

The cnBR Core interacts with RPDs to:

- receive cable modem (CM) data.
- process CM control plane messages to establish and maintain modem sessions.

- forward upstream and downstream data between the modem and IP network.

It also captures the KPI health of the modem and RPD network, and provides a management interface for DOCSIS features and telemetry data, including service flows.

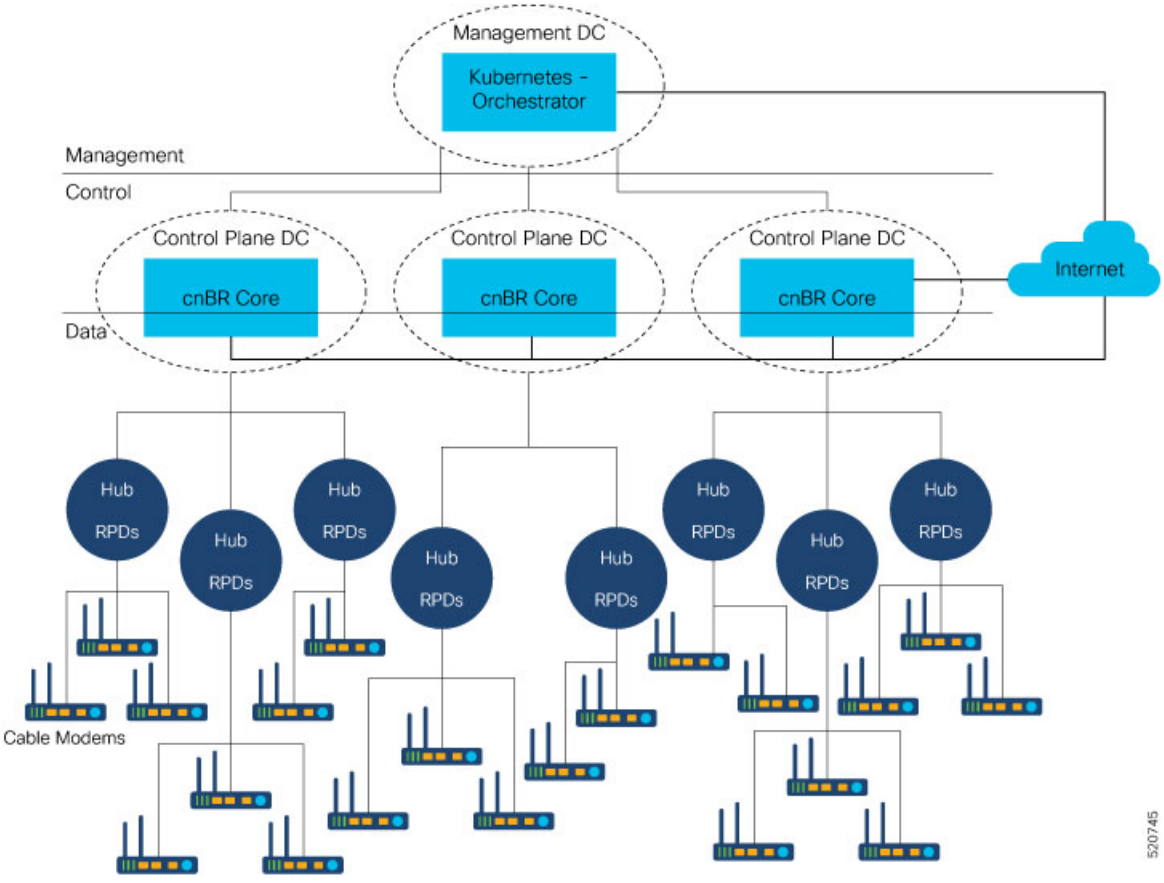
Remote PHY Device

The Remote PHY Device (RPD) provides analogue RF connectivity to the cable modems and digital connectivity to the CMTS Core (cnBR).

Cisco cnBR Deployment

The following figure depicts a typical Cisco cnBR deployment that separates management plane, control plane, and data plane components.

Figure 2: Typical cnBR Deployment



The management plane components, which include Operations Hub, are centralized within a central data center.

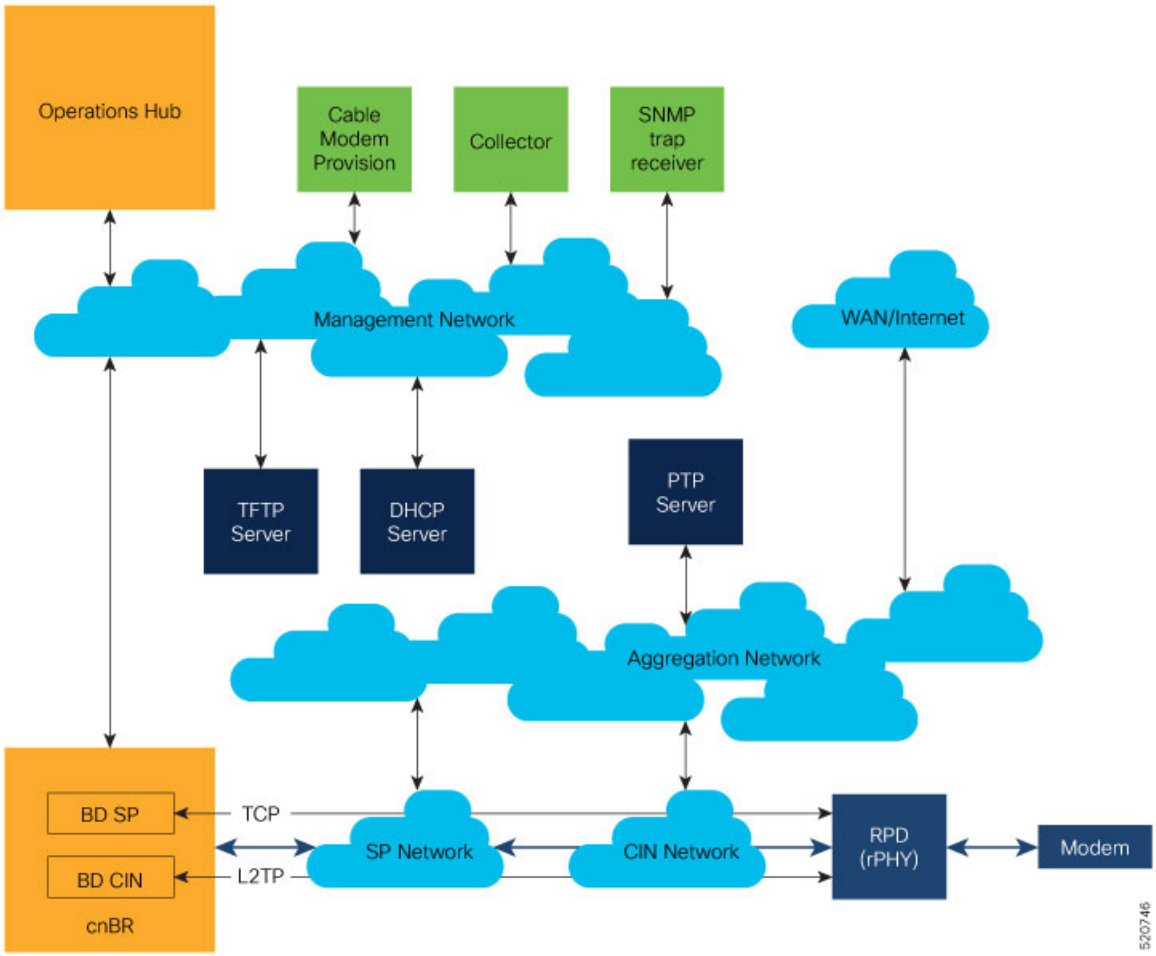
The Cisco cnBR, which contains the control plane components and routing for the data plane, is hosted within regional data centers.

The RPDs within hubs around the hub may connect to these regional data centers.

Because the entire solution has high availability, there can be no single point of failure, especially in the data plane.

The following diagram shows the components and networks that are configured when a Cisco cnBR is deployed in a typical service provider network.

Figure 3: Network Components



Network	Purpose
Aggregation	The aggregation network provides a nexus where necessary network paths converge to provide access to all necessary services.
Converged Interconnect Network (CIN)	The CIN network is well defined in the Cisco cable architecture and brings the cable modem traffic that has been converted to IP traffic into the digital DOCSIS network. The CIN network connects to the aggregation network for non-CMTS data traffic, such as PTP timing and RPD provisioning.
Management Network	The management network provides management-level interaction between the Cisco cnBR components and back-office services, such as IPDR collectors, SNMP trap, receivers, and cable modem provisioning and monitoring.

Network	Purpose
SP Network	The service provider networks provide a path for the cable modem traffic that is processed by the CMTS to reach the internet from the service provider side of the network.
WAN/Internet	The WAN/Internet network provides a path for the cable modems to send traffic to and receive traffic from the public internet.

These networks may be realized using one or more routers configured for each network.

The TFTP, DHCP, and PTP capabilities are required to be part of the solution and may be connected to different networks than those depicted in the figure. The PTP, DHCP, and TFTP address are configured within the Cisco cnBR.

The green boxes represent common service provider management features. In the past, cable modem provisioning and monitoring used information from the CMTS collected through SNMP MIBs. However, going forward, the preference is to move to REST APIs.

In the Cisco cnBR, the CIN and SP bridge domains must be configured. The CIN and SP bridge domain configurations provide first hop routing information to correspondingly named networks.

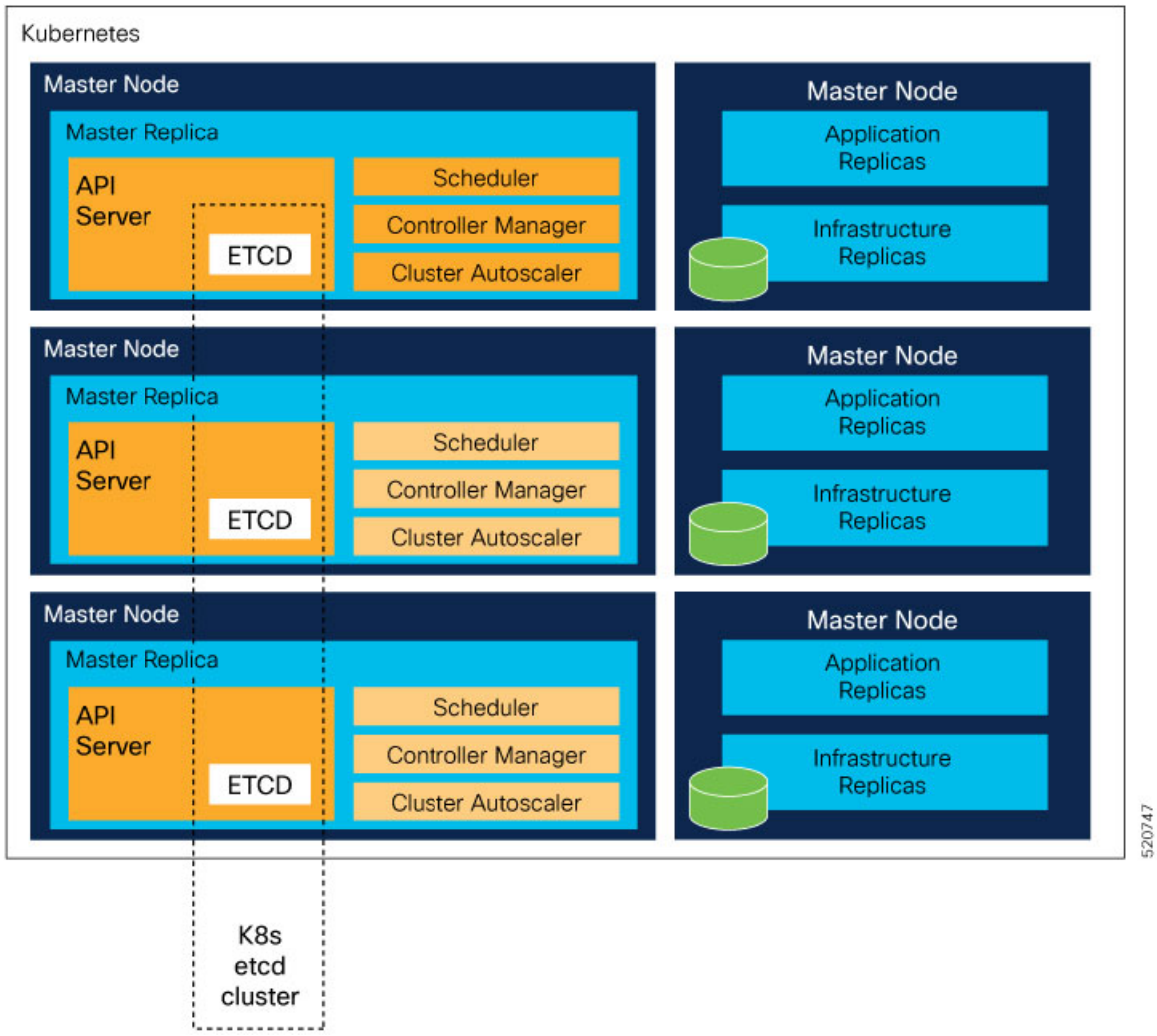
Kubernetes(K8S) Platform

The platform is deployed on VMs. In the future, bare metal deployment may be supported to maximize throughput.

The Cisco cnBR and management plane services are deployed as microservices within kubernetes (K8S) container orchestration clusters. Kubernetes platform supports deployment of replicated restartable microservices, where requests are routed and processed. Services are therefore highly available and scalable through redundancy.

To be hardware redundant, the K8S management functionality must be spread across separate nodes, either as bare metal servers or VMs hosted on separate servers as shown in the following figure.

Figure 4: Kubernetes Platform



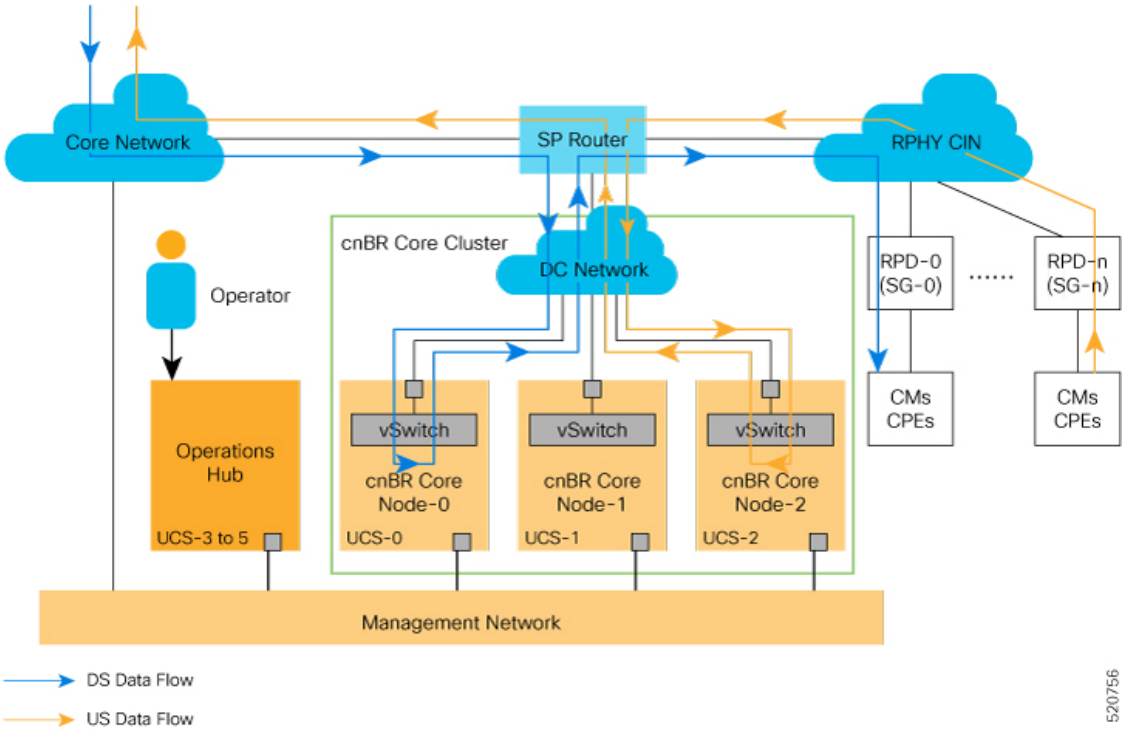
Similarly, the application load must be spread across worker nodes that are independent of the K8S control plane nodes. Separating the application workload from K8S control plane nodes protects the K8S management services from being impacted by the application workload.

The Cisco cnBR functionality and Cisco Operations Hub are hosted in a common cloud platform.

Cisco cnBR Network Topology

A typical Cisco cnBR network consists of cnBR core clusters, a SP router network, and Cisco Operations Hub. The following figure shows the core components and their inter-connections.

Figure 5: cnBR Inter-Connections and Data Flow



Core Components of Cisco cnBR Network Topology

- A highly available Cisco cnBR core cluster consists of three or more worker nodes, which provide core functionality of traditional CMTS: for example, DOCSIS control plane, data plane, and DOCSIS applications.
- SP Router forwards L3 packets between the uplink core network, RPHY CIN, and cnBR core services.
- Cisco Operations Hub is built in its own cluster and provides operation and management-related functionality in the Cisco cnBR system: for example, configuration, monitoring, and alert management.

Inter-Connections Between Core Components

- The SP router connects directly with the data center (DC) network to access multiple Cisco cnBR core nodes. The configuration is based on network virtualization technology that the UCS vSwitch uses, such as VLAN or VXLAN.
- Cisco Operations Hub communicates with the Cisco cnBR core clusters through internal RESTful messaging, which in turn is through the high-speed Management Network. The Management Network also transmits real-time telemetry data exported from Cisco cnBR core clusters to Cisco Operations Hub.

Downstream and Upstream Data Flow

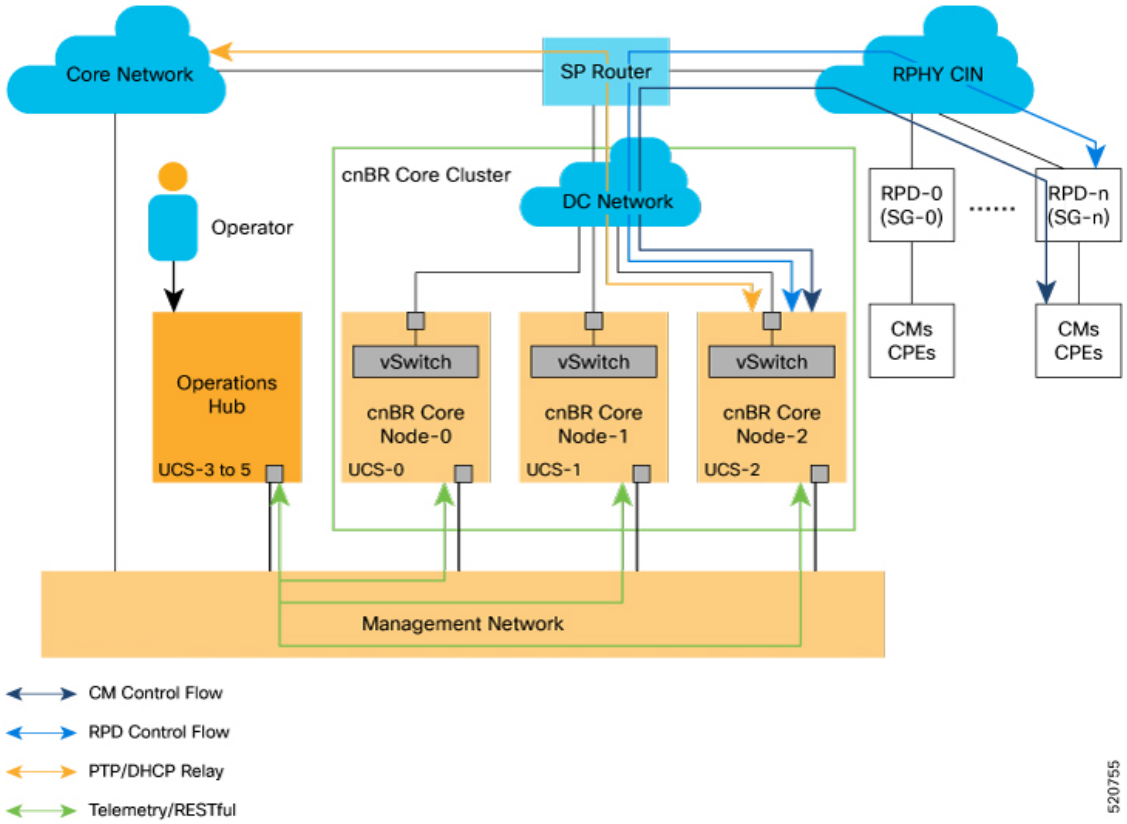
Figure 5: cnBR Inter-Connections and Data Flow, on page 8 illustrates the downstream and upstream data flows with arrows shown in different colors.

- All data traffic goes in and out of the Cisco cnBR core cluster for corresponding processing. The SP router acts as a hub.
- Different service groups (SG) are managed by different Cisco cnBR core nodes. For example, in Figure 5: cnBR Inter-Connections and Data Flow, on page 8, SG-0 is managed by cnBR-Core Node-0, while SG-n is managed by cnBR-Core Node-2.

Control Flow

As shown in the following figure, network data flows between the subscriber devices and Cisco cnBR core for control and data. It also flows between cnBR core and cnBR Cisco Operations Hub for management.

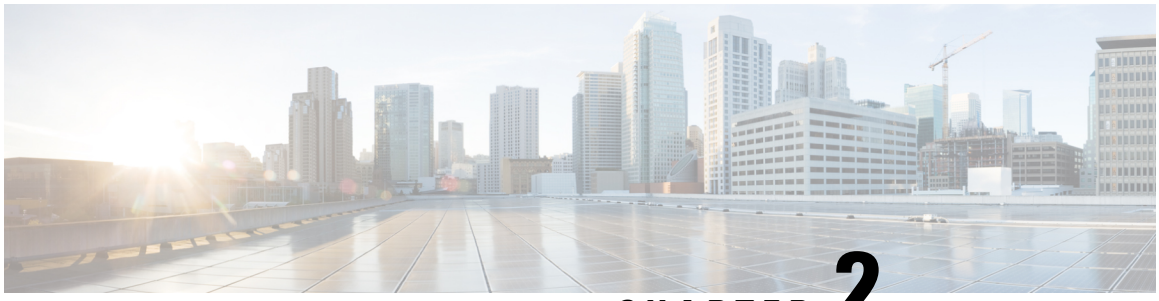
Figure 6: cnBR Control and Management Flows



The major Cisco cnBR control and management flows are:

- Cable modem control flow—between DOCSIS service and cable modems, for cable modem provisioning and management
- RPD control flow—between RPD service and RPD nodes, for RPD node provisioning and management
- Control flow—for PTP and DHCP relay service

- Cisco Operations Hub management flow—between Cisco Operations Hub and Cisco cnBR core services, for telemetry data export and RESTful interface messaging



CHAPTER 2

Set Up Cisco Cloud Native Broadband Router Components

This chapter provides information about the required prerequisite hardware and software, describes key components of Cisco cnBR, its topology, and how the router is deployed in a network. This chapter also provides information about how you can set up the Cisco cnBR core and the Cisco Operations Hub, and how you configure Cisco cnBR for service resiliency.

cnBR Prerequisites

The following prerequisite components are required to install, operate, and manage a Cisco cnBR. The prerequisites are:

- The Cisco cnBR server
- The Cisco Operations Hub server
- The Cisco cnBR topology
- The VMware deployment

Prerequisites required for the Cisco cnBR server

The Cisco cnBR runs exclusively on a Unified Computing System (UCS) server that is imaged with an VMware ESXi hypervisor.

- Cisco UCS server requirement

  Three Cisco UCS C220 M5 servers are required to run Cisco cnBR. The supported Cisco UCS servers are UCSC-C220-M5SX.

The minimum compute, storage, and networking requirements for the Cisco UCS server are listed in the following table.

Table 1: Minimum Requirements Cisco UCS Server

Component	Specification
Chassis	UCSC-C220-M5SX
Processor	Intel 6248 2.5GHz/150W 20C/27.5MB DCP DDR4 2933 MHz
Memory	384GB DDR4-2933-MHz RDIMM
Storage	4 x 800GB SSD
NIC	2 x Intel XL710-QDA2 (40G)

- VMware requirements
  - Hypervisor - VMware ESXi 6.5 Update 3 or VMware ESXi 6.7
  - Host Management - VMware vCenter Server 6.5 or VMware vCenter Server 6.7

If the VMware ESXi 6.7 is installed on host, ensure that the vCenter version is VMware vCenter Server 6.7.

Prerequisites required for the Cisco Operations Hub server

- Cisco UCS server requirement

  Three Cisco UCS C220 M5 servers are required to run Cisco cnBR. The supported Cisco UCS servers are UCSC-C220-M5SX.

  The minimum compute, storage, and networking requirements for the Cisco UCS server are listed in the following table.

Table 2: Minimum Requirements Cisco UCS Server

Component	Specification
Chassis	UCSC-C220-M5SX
Processor	Intel 6248 2.5GHz/150W 20C/27.5MB DCP DDR4 2933 MHz
Memory	384 GB DDR4-2933-MHz RDIMM
Storage	4 x 800 GB SSD
NIC	2 x Intel XL710-QDA2 (40G)

- VMware requirements
  - Hypervisor - VMware ESXi 6.5 Update 3 or VMware ESXi 6.7

- Host Management - VMware vCenter Server 6.5 or VMware vCenter Server 6.7

If the VMware ESXi 6.7 is installed on host, ensure that the vCenter version is VMware vCenter Server 6.7.

- Browser support

For the Cisco cnBR, the Cisco Operations Hub functionality is supported for the following browser versions:

- Mozilla Firefox 78.0 and later
- Google Chrome 83 and later or Google Chrome 84 and later

Prerequisites required for the Cisco cnBR topology

- Cisco cnBR Data Switch

You must use a data center switch with the requisite 40G port density between the Cisco cnBR servers and the service provider router to aggregate the Cisco cnBR data path links.

- Management Switch

A dedicated data center switch can be used for Cisco cnBR and Cisco Operations Hub management traffic. The Cisco cnBR and Cisco cnBR servers provide 1G, 10G, and 40G network interface connectivity options for the different management networks that are used in the system. The management networks can be VMware ESXi host management, Cisco cnBR and Cisco Operations Hub virtual machine cluster management, and the Cisco Integrated Management Controller (IMC) Lights-Out-Management.

- Service Provider Router

The SP Router is responsible for forwarding L3 packets between the core network, RPHY CIN, and Cisco cnBR. The SP Router and Cisco cnBR establishes connections through BGP, SG, RPHY-core for RPD session setup and traffic forwarding.

We recommend the following Cisco Network Convergence System 5500 Series models:

- NCS-55A1-36H-S
- NCS-55A1-24H

The required software version must be Cisco IOS XR 6.5.3 or later.

- DHCP Server

A standard Dynamic Host Configuration Protocol (DHCP) server is required, and typically included in an existing DOCSIS infrastructure. For example, the DHCP server included is the Cisco Network Registrar (CNR).

- PTP Server Configuration

A Precision Time Protocol (PTP) server is required and typically included in an existing DOCSIS infrastructure. For example, an OSA 5420.

- TFTP Server

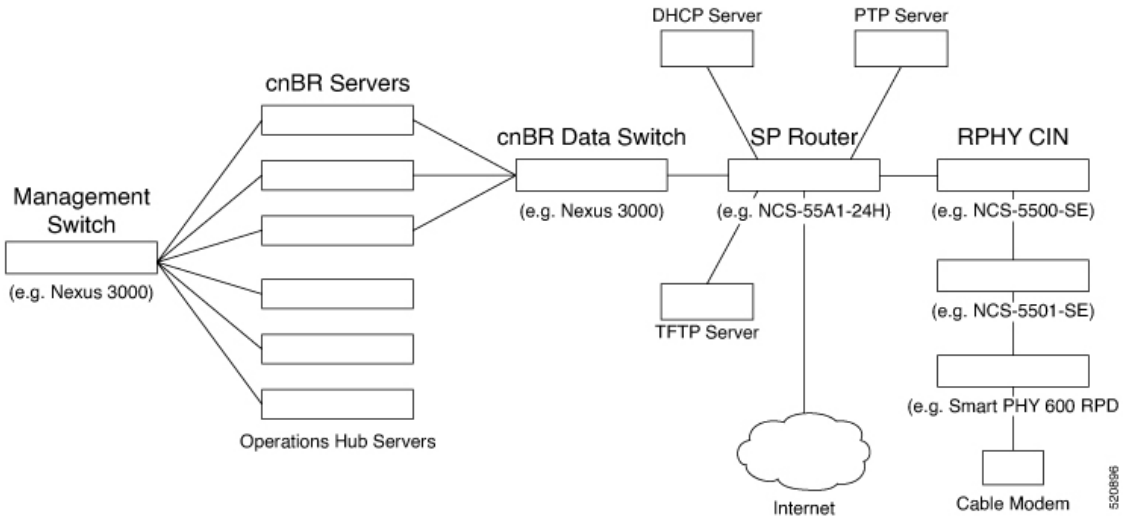
A standard Trivial File Transfer Protocol (TFTP) server is required and typically included in an existing DOCSIS infrastructure.

- RPHY CIN

  A Remote PHY Converged Interconnect Network (CIN) is required. A Remote PHY Device, and Cable Modems are also required. For example, Cisco Smart PHY 600 Shelf.

  The following image is a simplified, high-level overview of an end-to-end system and shows how these Cisco cnBR components are connected in the topology with provisioning systems and a Remote PHY CIN:

Figure 7: Simplified cnBR Topology

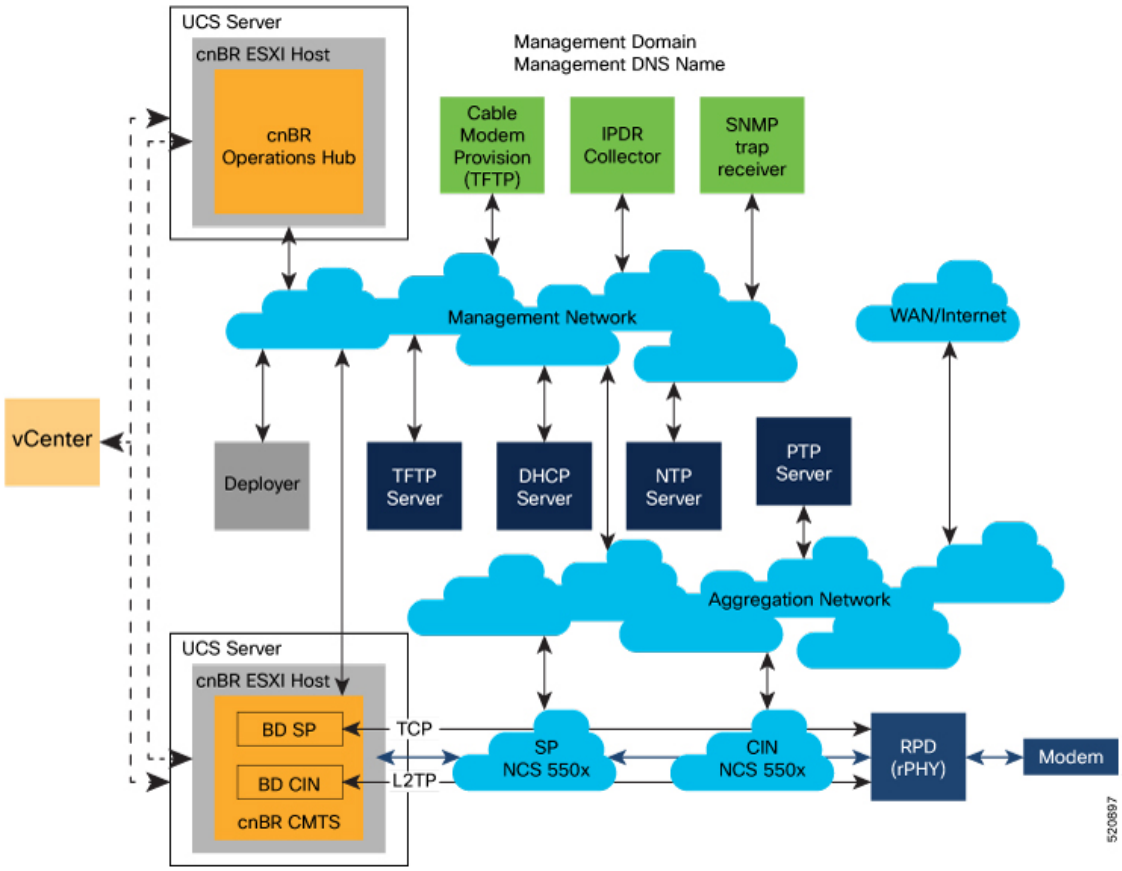


Prerequisites required for the VMware deployment

VMware is a mandatory component for the Cisco cnBR Cisco Operations Hub server, and is necessary for the deployment topology.

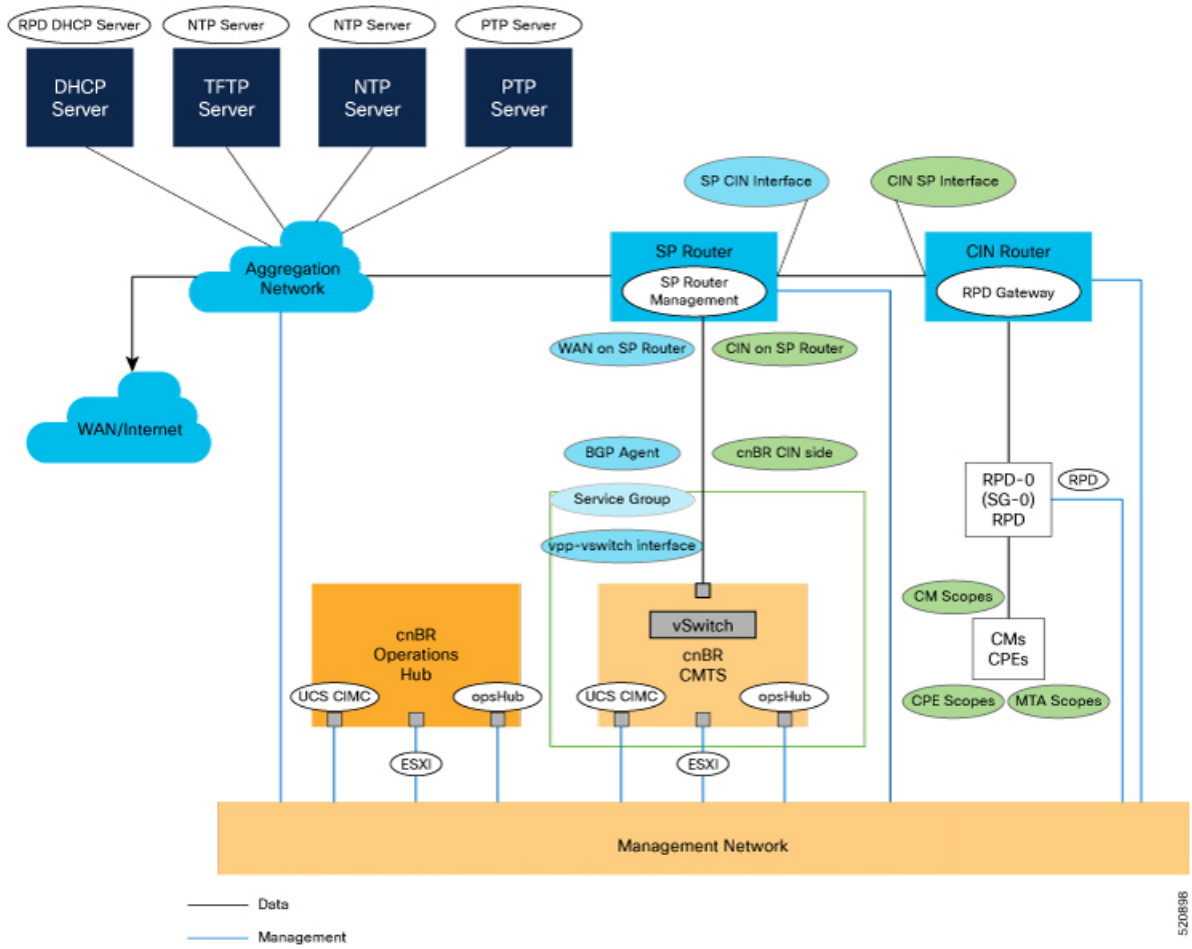
A generalized Cisco cnBR deployment with the Cisco Operations Hub and Cisco cnBR core hosted in VMware clusters is depicted in the following image:

Figure 8: cnBR Deployment in a VMware Cluster



The VMware network topology in the following image is for a VLAN configuration:

Figure 9: VLAN Configuration with VMware Network Topology



The necessary IP addresses and networks that are mapped in the diagram are described in the following sections:

- **Networks**

  The following table provides guidance for the networks that are needed in the management, WAN, and CIN routing domains:

  Table 3: Network Information for Routing Domains

Name	Subnet Mask	Function
Management	• 2 addresses for each cluster • OpsHUB/cnBR UCS • 1 for each cluster • 1 for each service device	Management
CIN	Network requirements for each customer	Connection RPD and CCAP core

Name	Subnet Mask	Function
WAN	Network requirements for each customer	Internet access for CPE
cnBR CIN side	Network requirements for each customer	-
BGP network to SP router	Network requirements for each customer	Management
Network for data	Network requirements for each customer	-
SG IP cnBR side	Network requirements for each customer	The peer IP for Service Group on cnBR
RPD address pool	Customer selected	DHCP scope for RPD sized to cover total number of RPDs
DHCP scope for CM	Customer selected	-
DHCP scope for CPE	Customer selected	-
DHCP scope for MTA	Customer selected	-

You must provide domain and DNS name for the management network.

- **Device Addresses**

  The followings tables provide information on the IP address that is needed for device and router interfaces.

  - **Management IP Address**: Each management interface that is listed in the following table requires 1 IP address:

    Table 4: Management Interface and Associated IP Addresses

Device name	Number of Addresses
CIMC cnBR	1 per cnBR UCS
ESXi cnBR	1 per cnBR UCS
CIMC OpsHub	1 per Operations Hub UCS
ESXi OpsHub	1 per Operations Hub UCS
cnBR	1 per cnBR Cluster
Operations Hub	1 per Operations Hub Cluster
Deployer	1
vCenter	1
SP router	1

Device name	Number of Addresses
CIN router	1

- **DOCSIS Network Addresses**: The following table lists the DOCSIS network-related information:

Table 5: DOCSIS Network-Related Information

Device Name	Network Name	Description	Number of Addresses
SP router to CIN	CIN	SP connection to CIN router	1
CIN router to SP	CIN	CIN connection to SP router	1
SP router to WAN	WAN	SP connection to WAN/Internet	1
RPD Gateway	CIN	RPD gateway router Address	1
cnBR CIN side	CIN	cnBR connection to CIN	Customer specific
BGP Agent	WAN	WAN router BGP Agent IP	Customer specific
Service Group	WAN	Service Group WAN IP	Customer specific
WAN on SP Router	WAN	SP connection to WAN network	Customer specific

- **Customer Provisioned Services**: The following table lists the various customer services:

Table 6: Customer Provisioned Services

Service	Notes
DHCP	Needed for both RPD and subscriber devices
TFTP	RPD only uses it during software upgrade
TOD	Time of day clock
PTP	One connection that is required for the cnBR and for each RPD
NTP	Network Time Protocol Server
DNS	Domain Name Server

Prepare Supporting Software Components

To prepare the Cisco Unified Computing System (UCS) servers for software installation, you must do the following.

- Configure the servers using Cisco Integrated Management Controller (CIMC)
- Install VMware ESXi
- Add VMware ESXi Hosts to a VMware vSphere cluster using VMware vCenter

Note Cisco UCS Servers ordered using the Cisco cnBR PID are preconfigured, imaged, and ready for installation. For Cisco cnBR PID-specific servers, execute the steps in Cisco UCS Server Installation and continue to Add Cisco cnBR ESXi Hosts to vSphere Virtual Infrastructure, on page 21.

Cisco cnBR Server Installation and Configuration

Step 1 Cisco UCS Server Installation, on page 19
Step 2 Update Firmware, on page 20
Step 3 Load Cisco cnBR Optimized BIOS Configuration, on page 20
Step 4 Configure Boot Drives, on page 20
Step 5 Configure Data Drives, on page 21
Step 6 Install VMware ESXi, on page 21
Step 7 Reboot VMware ESXi Host and Set Boot Device, on page 21

Cisco UCS Server Installation

Step 1 Rack mount the servers. See Cisco UCS C220 M5 Server Installation and Service Guide.
Step 2 Ensure both power supplies are connected on each server, and power on the servers.
Step 3 Connect the following network cables:

- For Cisco Integrated Management Controller (CIMC), use the 1Gb Ethernet dedicated management port.
- For VMware ESXi Host Management, use Ethernet port 1 of the Dual 1Gb/10Gb Intel X550T on board NIC.
- For Cisco cnBR Data, connect port 1 of the Intel XL710 40G NIC in PCIe Slot 1 to the SP Router/Leaf Switch using Cisco QSFP-40G-SR4.

Step 4 Connect the UCS Kernel-based Virtual Machine (KVM) console adapter or connect a keyboard and monitor directly to the server.

Step 5 Configure CIMC through the KVM console and update the Network Settings.

Update Firmware

Download the latest Hardware Update Utility for the UCS C220 M5 Server from Cisco's Software Download site and use it to update the CIMC, BIOS, and Device Firmware for Storage Controllers, Network Adapters, SSDs, and other components.

Load Cisco cnBR Optimized BIOS Configuration

Step 1 Create a new json file "cnbr_perf.json" and add the following structure.

Cisco cnBR Optimized BIOS profile config for C220 M5 Servers

```
{
   "name":"Perf_M5",
   "description":"",
   "tokens":{
      "EnhancedIntelSpeedStep":"Enabled",
      "IntelTurboBoostTech":"Enabled",
      "IntelHyperThread":"Disabled",
      "CPUPerformance":"Enterprise",
      "ExecuteDisable":"Enabled",
      "IntelVTD":"Enabled",
      "ProcessorC1E":"Disabled",
      "ProcessorC6Report":"Disabled",
      "PsdCoordType":"HW ALL",
      "CpuEngPerfBias":"Performance",
      "PwrPerfTuning":"BIOS",
      "CpuHWPM":"HWPM Native Mode",
      "WorkLdConfig":"IO Sensitive",
      "SelectMemoryRAS":"Maximum Performance",
      "SNC":"Disabled",
      "XPTPrefetch":"Enabled",
      "DcuIpPrefetch":"Enabled",
      "PatrolScrub":"Disabled"
   }
}
```

Step 2 Load the optimized Cisco cnBR BIOS configuration into the system using "cnbr_perf.json".

Step 3 Save a backup of the current BIOS settings.

Step 4 Select the new profile "Perf_M5" and activate it.

Configure Boot Drives

Step 1 Enable the Cisco MSTOR Boot Optimized M.2 RAID Controller.

Step 2 Create a RAID 1 virtual drive from 2 x M.2 SSD Drives.
Step 3 Set Stripe Size to 64KB

Configure Data Drives

Step 1 Enable Cisco 12G SAS Modular RAID Controller.
Step 2 Create a RAID 5 enabled virtual drive using 4 x SSDs.
Step 3 Set Stripe Size to 64KB.
Step 4 Set Write Cache Policy to *Write Back with Good BBU*.

Install VMware ESXi

Step 1 Install VMware ESXi 6.5 Update 3 on the M.2 RAID 1 Virtual Drive (Boot Drive).
Step 2 Use the Cisco Custom ISO - VMware_ESXi_6.5.0_13932383_Custom_Cisco_6.5.3.1.iso
Step 3 Set a password for the root user following the installation process.
Step 4 Reboot the VMware ESXi host following the installation process and execute the steps in Reboot VMware ESXi Host and Set Boot Device, on page 21.

Reboot VMware ESXi Host and Set Boot Device

Step 1 Interrupt the boot process with the F2 key after the host resets and boot into the BIOS.
Step 2 Under the Boot Options tab, set Boot Option #1 to the UEFI target - *VMware ESXi*.
Step 3 Disable all other boot options.
Step 4 Save changes and exit.
Step 5 Confirm the host boots directly into VMware ESXi.

Add Cisco cnBR ESXi Hosts to vSphere Virtual Infrastructure

Step 1 Configure VMware ESXi Host Management Networking, on page 22
Step 2 Add ESXi Hosts to VMware vCenter Server, on page 22
Step 3 Configure and Enable Required ESXi Host Features, on page 22
Step 4 Configure Virtual Machine Networking, on page 22

Configure VMware ESXi Host Management Networking

Step 1 Log into the VMware ESXi host through the Direct Console User Interface (DCUI) with the root account.

Note For Cisco cnBR PID Servers, use the password received from your Cisco representative as part of your Cisco cnBR order.

Step 2 Configure the management network.

a) Update IP configuration.
b) Update DNS configuration.
c) Update custom DNS suffixes.
d) Update VLAN ID if required.

Add ESXi Hosts to VMware vCenter Server

In VMware vCenter:

Step 1 Create a new, dedicated cluster for Cisco cnBR.

Note Do not enable DRS or any HA features.

Step 2 Add each new Cisco cnBR ESXi Host to the new Cisco cnBR cluster.

Configure and Enable Required ESXi Host Features

Step 1 Configure time on the host.

a) Enable NTP.

Step 2 Apply ESXi host licenses.

Step 3 Enable PCI Pass-through on all four Intel XL710 40G QSFP+ ports(requires host reboot).

Step 4 Create a new datastore on the data drive storage device.

Note By default, Cisco cnBR PID servers have a datastore created and PCI Pass-through enabled.

Configure Virtual Machine Networking

Step 1 Ensure VMware vSwitch connectivity to the physical switch.

Step 2 Create a PortGroup and a VMware vSwitch for the Kubernetes Cluster Node VM MGMT Network.

Deployment of cnBR and Operations Hub

Cisco cnBR supports offline installation of the SMI Cluster Manager, Cisco Operations Hub, and Cisco cnBR clusters.

All required installation packages are available from the SMI Cluster Deployer in an offline deployment scenario. The packages include Helm charts, Docker images used by the Cisco cnBR, and Cisco Operations Hub cluster nodes. Note that cluster nodes do not pull software or images directly from Cisco Artifactories. Product tar files containing all necessary Helm charts and container images are separate. The tar files are imported into the SMI Deployer during the deployer creation process.

The installation of the SMI Deployer Virtual Manager is from a working directory on a staging server. The staging server can be any host - physical server, virtual machine, or an administrators laptop. However, you must ensure that you can connect to the target vSphere Infrastructure, vCenter Server, and cluster nodes with the proper credentials.

The Autodeploy utility creates the deployer, and deploys the Cisco Operations Hub and Cisco cnBR clusters. The Autodeploy utility is part of the Cisco cnBR release bundle.

Prepare the Staging Server

Complete the following steps to prepare the staging server:

Before you begin

Ensure that you have a staging server setup with the following prerequisites:

- Python 3: See https://www.python.org/ for more information.
- OpenSSL: See https://www.openssl.org/ for more information.
- Docker: See https://docs.docker.com/get-docker/ for more information.
- The staging server must have network connectivity to the VMware nodes.

Step 1 Verify the image signature.

In an offline deployment scenario, you must verify the authenticity and integrity of the image before the installation and deployment. You can choose to verify the image signatures online or offline.

We recommend online verification. Offline verification can be used when where there is no network access to perform online verification.

A corrupted or tampered image can lead to an image verification failure. Discard the image and contact the Cisco Customer Support to get the authentic image.

a) Extract the Cisco cnBR release bundle. Untar the `cnbr-installer-<release-version-tag>.SPA.tgz` signed release bundle as shown:

```
~/staging$ tar xvzf cnbr-installer-<release-version-tag>.SPA.tgz
  cnbr-installer-<release-version-tag>.tgz   # cnBR release bundle
```

```
  isign/                                    # folder with image verification content
  isign/cnbr-installer-<release-version-tag>.tgz.signature
  isign/CNBR-BUNDLE_pubkey.der
  isign/cisco_x509_verify_release.py3
  isign/CNBR_IMAGE_SIGN-CCO_RELEASE.cer
  verify_signature_offline                  # script to be used to verify the image signature
offline
  verify_signature_online                   # scirpt to be used to verify the image signature
online
```

b) Verify the image by choosing either of the following methods. We recommend the online verification.

- Online image verification. Run the following script to verify the image. A successful verification is as follows:

```
~/staging$ ./verify_signature_online
Downloading CA certificate from http://www.cisco.com/security/pki/certs/crcam2.cer ...
Successfully downloaded and verified crcam2.cer.
Downloading SubCA certificate from http://www.cisco.com/security/pki/certs/innerspace.cer ...
Successfully downloaded and verified innerspace.cer.
Successfully verified root, subca and end-entity certificate chain.
Successfully fetched a public key from isign/CNBR_IMAGE_SIGN-CCO_RELEASE.cer.
Successfully verified the signature of cnbr-installer-<release-version-tag>.tgz using
isign/CNBR_IMAGE_SIGN-CCO_RELEASE.cer
```

- Offline image verification. Run the following script to verify the image. A successful verification is as follows:

```
~/staging$ ./verify_signature_offline
Verified OK
```

Step 2 Untar the Cisco cnBR release bundle:

```
> tar xvzf cnbr-installer-<release-version-tag>.tgz
> cd cnbr-installer-<release-version-tag>
```

The directory, `staging/cnbr-installer-\<release-version-tag\>`, is referred to as staging or install directory. The directory has the following content:

```
~/staging/cnbr-installer-<version-tag>$ tree
.
├── README.md
├── cluster-deployer-airgap.vmdk
├── deploy
├── docker-images
│   └── ccmts-customization_<version-tag>.tar
├── examples
│   ├── aio-opshub-config.yaml        # For Experimental, Lab/Demo purpose only
│   ├── deployer-sample-config.yaml
│   ├── multinode-cnbr-config.yaml
│   ├── day1_config_mn.yaml
│   ├── day1_config_aio.yaml
│   ├── sg_template_4x4.json
│   ├── l3_template.json
├── offline-products
│   ├── cnbr-master.tar
│   ├── cee-<version-tag>.tar
│   └── opshub-master.tar
└── utility-images
    ├── autodeploy_<version-tag>.tar
    └── cluster-manager-docker-deployer_<version-tag>.tar
```

```
4 directories, 16 files
```

Create the Configuration File

The configuration file is in the standard YAML descriptive language format.

Complete the following steps to create the configuration file:

Step 1 **Configuring the environment**: The environment configuration provides the vCenter access and network access details used to create and provision the deployers and cluster virtual machines. Deployer and clusters need environments to be defined before their creation and deployment.

The deployer contains all the defined environments that can be reused by clusters. The deployer refers to the corresponding vCenter environment by name.

```
environments:
  "<<vcenter-env>>":                                # vCenter environment name

      server: "<<XX.XX.XX.XX>>"                     # vCenter Server IP
      username: "<<user-name>>"                     # vCenter username, user will be prompted for
the password
      datacenter: "<<vmware datacenter>>"           # DataCenter Name
      cluster: "<<vcenter cluster>>"                # vCenter Cluster Name
      nics: [ "<<VM Network>>", "<<VM Network1>>" ]  # vCenter nics (port groups)
      nameservers: [ "<<YY.YY.YY.YY>"]              # DNS Servers
      search-domains: [ "<<yourdomain>>" ]          # Search domains
      ntp: "<<yourclock.domain>>"                   # NTP Server
      https-proxy: "<<http://proxyhost.domain:port>>"
      no-proxy: "<<127.0.0.1,localhost>>"
```

Step 2 **Configuring the deployer**: Ensure that you have at least one environment defined, before a deployer is created for deployment. The deployer holds all the defined environments which can be reused by clusters when referred to by name.

```
deployers:
  "<<deployer3-test>>":                             # deployer VM name
      environment: "<<vcenter-env>>"                # reference to vCenter environment
      address: "<<XX.XX.XX.XX/prefix_len>>"         # SSH-IP of the VM in CIDR format
      gateway: "<<XX.XX.XX.XX>>"                    # Gateway for the VM
      username: "<<user-name>>"                     # Deployer VM username, user will be prompted
for the password
      # SSH private-key-file with path relative to the staging directory
      # Key will be auto-generated, if one if not provided
      private-key-file: "<<cmts.pem>>"
      host: "<<XX.XX.XX.XX>>"                       # Server IP where Deployer VM is hosted
      datastore: "<<datastore1>>"                   # Datastore for the Deployer VM
```

Step 3 **Configuring the cluster**: A cluster (Cisco cnBR/Cisco Operations Hub Multi-Node) needs at least one environment and deployer to be defined before its creation and deployment. A cluster also needs references to the corresponding environment and deployer.

A cluster can be one of the following types:

- Multi-Node Cisco cnBR
- Multi-Node Cisco Operations Hub

Note
- Single-Node Cisco Operations Hub is supported for Lab or Demo purpose only.
- Single-Node Cisco cnBR clusters are not supported.

Multi-Node Configuration

- The reference configuration below distributes the Cluster Node VMs evenly across three ESXi Hosts with proper NUMA alignment and computes the resource reservation.
- 13 Management IP Addresses total = 12 for the Cluster Nodes + 1 Primary Virtual IP.
- For each of the following node, update the `k8s ssh-ip`, `VMware datastore`, and `VMware host` accordingly.
- For the DOCSIS nodes, the PCI device must be identified and available.

```
clusters:
  # Name of the cluster
  "<<cnbr-multi>>":                                 # cnBR cluster name
      type: "<<cnbr>>"                              # Cluster type 'cnbr' or 'opshub'
      environment: "<<vcenter-env>>"                # reference to vCenter environment
      # PCI Passthrough, used for docsis nodes only
      # speficy this variable only to enable PCI passthough
      pci_device: "<<0000:5e:00.0>>"
      gateway: XX.XX.XX.XX                          # Gateway for the Cluster
      username: "<<user-name>>"                     # Cluster username, User will be prompted for
the cluster password
      # SSH private-key-file with path relative to the staging directory
      # Key will be auto-generated, if not provided
      private-key-file: "<<cmts.pem>>"
      master-vip: "<<XX.XX.XX.XX/prefix_len>>"      # Master vip in CIDR format for Multi-Node only


    # For Multi-Node only
      nodes:
        -  host: "<<XX.XX.XX.182>>"                 # Server IP where Deployer VM is hosted
           # IP addresses assigned to master, etcd, infra and docsis/ops nodes respectively
         addresses: [ "<<XX.XX.XX.187>>", "<<XX.XX.XX.172>>", "<<XX.XX.XX.169>>", "<<XX.XX.XX.190>>"]

           datastore: "<<XX.XX.XX.182-datastore1>>"
        -  host: "<<XX.XX.XX.176>>"
         addresses: [ "<<XX.XX.XX.188>>", "<<XX.XX.XX.173>>", "<<XX.XX.XX.170>>", "<<XX.XX.XX.191>>"]

           datastore: "<<XX.XX.XX.176-datastore1>"
        -  host: "<<XX.XX.XX.184>>"
         addresses: [ "<<XX.XX.XX.189>>", "<<XX.XX.XX.174>>", "<<XX.XX.XX.171>>", "<<XX.XX.XX.192>>"]

           datastore: "<<XX.XX.XX.184-DataStore1>>"
           # secify pci_device id if different from the global pci_device id
           pci_device: "<<0000:5e:00.1>>"

    # For Single-Node cluster [ Only supported, for Lab/Demo purpose for OperationsHUB ]
      nodes:
        -  host: "<<XX.XX.XX.182>>"                 # Server IP where Deployer VM is hosted
           addresses: [ "<<XX.XX.XX.187/prefix_len>>"]
           datastore: "<<XX.XX.XX.182-datastore1>>"
```

Deploy the Cluster

Deploy the cluster by using the following command:

```
~/cnbr-installer-<release-version-tag>$ ./deploy -c <config_file>
```

The Cluster Manager is deployed first, before deploying any cluster. To deploy more clusters, run the command with the corresponding configuration files.

Deployment Example Configurations

Example configuration files are available in the staging or examples directory. You can copy, modify, and use the appropriate example configuration file.

Ensure that you have gone through Step 1 and Step 2 topics.

- **Sample Deployer Configuration**

  The following is a sample configuration to deploy the cluster manager. The sample has two mandatory sections for all cluster configurations.

```
environments:
  "vcenter-env":
      server: "XX.XX.XX.XX"
      username: "vCenter username"
      datacenter: "vmware datacenter"
      cluster: "vmware cluster"
      nics: [ "VM Network" ]
      nameservers: [ "DNS1", "DNS2"]
      search-domains: [ "yourdomain" ]
      ntp: "yourclock.yourdomain"
      https-proxy: "http://proxyhost.domain:port"
      no-proxy: "127.0.0.1,localhost"

deployers:
  "deployer3-test":
      environment: "vcenter-env"
      address: "XX.XX.XX.194/prefix_len"
      gateway: "XX.XX.XX.129"
      username: "cloud-user"
      private-key-file: "cmts.pem"
      host: "XX.XX.XX.184"
      datastore: "XX.XX.XX.184-DataStore1"
```

- **Multi-Node cnBR Configuration**

  Define the cluster configuration as shown:

```
clusters:
  "cnbr-mnode":
      type: "cnbr"
      environment: "vcenter-env"
      # comment out pci_devic to disable PCI
      pci_device: "0000:5e:00.0"
      master-vip: "XX.XX.XX.193/prefix_len"
      username: "cloud-user"
```

```
    private-key-file: "cmts.pem"
    gateway: XX.XX.XX.129
    nodes:
      -  host: "XX.XX.XX.182"
         datastore: "XX.XX.XX.182-datastore1"
        addresses: [ "XX.XX.XX.187", "XX.XX.XX.172", "XX.XX.XX.169", "XX.XX.XX.190"]

      -  host: "XX.XX.XX.176"
         datastore: "XX.XX.XX.176-datastore1"
        addresses: [ "XX.XX.XX.188", "XX.XX.XX.173", "XX.XX.XX.170", "XX.XX.XX.191"]

      -  host: "XX.XX.XX.184"
         datastore: "XX.XX.XX.184-DataStore1"
        addresses: [ "XX.XX.XX.189", "XX.XX.XX.174", "XX.XX.XX.171", "XX.XX.XX.192"]
```

- **Multi-Node Operations Hub Configuration**

  Define the cluster configuration as shown:

```
clusters:
  "opshub-mnode":
      type: "opshub"
      environment: "vcenter-env"
      master-vip: "XX.XX.XX.193/prefix_len"
      gateway: XX.XX.XX.129
      username: "cloud-user"
      private-key-file: "cmts.pem"
      nodes:
        -  host: "XX.XX.XX.182"
           datastore: "XX.XX.XX.182-datastore1"
          addresses: [ "XX.XX.XX.187", "XX.XX.XX.172", "XX.XX.XX.169", "XX.XX.XX.190"]

        -  host: "XX.XX.XX.176"
           datastore: "XX.XX.XX.176-datastore1"
          addresses: [ "XX.XX.XX.188", "XX.XX.XX.173", "XX.XX.XX.170", "XX.XX.XX.191"]

        -  host: "XX.XX.XX.184"
           datastore: "XX.XX.XX.184-DataStore1"
          addresses: [ "XX.XX.XX.189", "XX.XX.XX.174", "XX.XX.XX.171", "XX.XX.XX.192"]
```

- **Single-Node Operations Hub Configuration**

  The Single Node Cluster is not supported for production. It is restricted for use at the Lab.

  Define the cluster configuration as shown:

```
clusters:
  "opshub-snode":
      type: "opshub"
      environment: "vcenter-env"
      gateway: XX.XX.XX.129
      username: "cloud-user"
      private-key-file: "cmts.pem"
      nodes:
        -  host: "XX.XX.XX.139"
           datastore: "XX.XX.XX.139-datastore1"
           addresses: [ "XX.XX.XX.159/prefix_len" ]
```

Deployment Limitations

The following are the deployment limitations in this release:

- IPv6 addressing is not supported.
- The config file must comply to YAML syntax. Not conforming to the syntax might cause crash dumps.
- The configuration file must comply to all mandatory sections and attributes. You might see the autodeploy exit without warnings and errors when mandatory attributes are missing in the configuration file.
- Limited error and exception handling. When an exception or error occurs, you might see detailed crash dumps.
- Single node cluster for Cisco Operations Hub is not supported in production. Single Node Cisco Operations Hub clusters are meant for use at the Lab.

Configure Operations Hub

The Cisco Operations Hub in Cisco cnBR allows you to create and configure users.

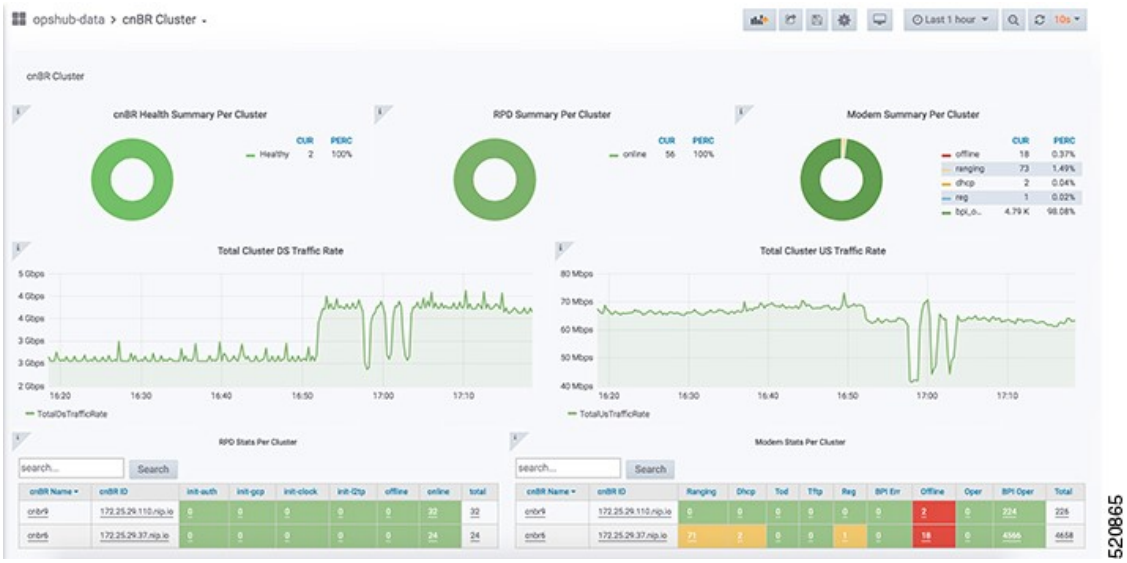
This section provides details of how to configure the Cisco Operations Hub and to use the UI and APIs.

Access Operations Hub

You can access the **Operations Hub** home page using the following URl:

`https://{Hostname}`

The format of the `Hostname` is `{vip}.nip.io`, where `vip` is the virtual IP address of the Operations Hub Cisco cnBR cluster. You can see a homepage similar to the following after you log in.



Create New Users

You can create local users and cofigure LDAP for external authentication with Active Directory (AD).

API User Roles

Operations Hub supports three user roles based on the HTTP actions:

- api-admin: Allowed http method: GET, POST, PUT, DELETE
- api-editor: Allowed http method: GET, POST, PUT
- api-viewer: Allowed http method: GET

By default, the user, `admin` is already under these three groups.

Configure Local Users

Operations Hub **ops-center** CLI allows an administrator to create new users. Use the following procedure to create a user:

Step 1 Log in to the Operations Hub **ops-center** CLI using the admin user credentials created during the Operations Hub deployment.

The Operations Hub ops-center URL is: `https://cli.opshub-data-ops-center.{Hostname}/`

```
product opshub# smiuser show-user username admin
User: admin, Group(s): admin api-admin api-editor api-viewer li-admin, Password Expiration days: 86
```

Step 2 Run the following command to define a new user:

```
smiuser add-user username <username> password <password>
```

Example:

```
product opshub# smiuser add-user username opshubuserA password Abcd123@
message User added

product opshub# smiuser show-user username opshubuserA
User: opshubuserA, Group(s): opshubuserA, Password Expiration days: -1
```

Step 3 Run the following command to add the new user to one of the API groups:

```
smiuser assign-user-group username <username> groupname <API group name>
```

Example:

```
product opshub# smiuser assign-user-group username testuser groupname api-admin
message User assigned to group successfully
product opshub
```

Configure LDAP

Operations Hub **ops-center** CLI allows the administrator to configure LDAP settings for external authentication with AD (Active Directory).

Step 1 Log into the Operations Hub ops-center CLI using the admin user credentials created during the Operations Hub deployment.

```
The Operations Hub ops-center URL is: https://cli.opshub-data-ops-center.{Hostname}/
```

Step 2 Configure the LDAP server using the following commands:

```
product opshub# config terminal
Entering configuration mode terminal
product opshub(config)# ldap-security ldap-server-url <URL>
product opshub(config)# ldap-security ldap-username-domain <domain>
product opshub(config)# ldap-security base-dn DC=<example>,DC=com
product opshub(config)# ldap-security ldap-filter userPrincipalName=%s@<domain>.com
product opshub(config)# ldap-security group-attr memberOf
product opshub(config)# end
Uncommitted changes found, commit them? [yes/no/CANCEL] yes
Commit complete.
```

Step 3 Configure the mapping between LDAP groups and API groups:

```
product opshub# config terminal
Entering configuration mode terminal
product opshub(config)# ldap-security group-mapping {ldap group} api-admin
product opshub(config-group-mapping-crdc-docsis/api-admin)# end
Uncommitted changes found, commit them? [yes/no/CANCEL] yes
Commit complete.
```

Using REST APIs

This section explains how you can use REST APIs.

Step 1 Create a user.

Use the procedure from the Create New Users, on page 29 section.

Step 2 Call auth REST API to create token.

Encode the username and password with base 64. Fill the encode output into the authentication header.

Example:

```
User: admin
Password: bell

Get the Base64 under Linux: echo -n 'admin:lab' | base64
Base64 encode output: YWRtaW46bGFi

curl -X POST "https://{hostname}/api/auth/v1/token" -H  "accept: application/json" -H  "authorization:
 Basic YWRtaW46bGFi"

Response code: 201
Response body
{
"access_token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJyb2xlIjoiYXBpLWFkbWluIiwic2FsdCI6IlViQ2daamt
IWHd6RUNzS1EiLCJleHAiOjE1NjQ2NTA2MTd9.x7ccHcOn6fLvHc_ajLJxQEY1ftvR1ZaJH9K_YZx1ues",
"refresh_token": "lYYtZqgVhnsnBJgSHbigRzeEaLnWziMpHJKVzgHA",
"refresh_token_expire": 1567221017,
"token_type": "jwt"
}
```

Step 3 With this token, call other REST APIs.

Example:

Call REST API to get the Cisco cnBR list:

```
curl -X GET "https://10.x.x.x.nip.io/api/manager/v1/cmts" -H  "accept: application/json" -H
  "Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJyb2xlIjoiYXBpLWFkbWluIiwic2FsdCI6IlV
iQ2daamtIWHd6RUNzS1EiLCJleHAiOjE1NjQ2NTA2MTd9.x7ccHcOn6fLvHc_ajLJxQEY1ftvR1ZaJH9K_YZx1ues"

Response code:200
Response body
{
  "cluster_list": [
    {
      "cluster-id": "10.124.211.141.nip.io",
      "cmts-name": "test",
      "namespace": "ccmts-infra",
      "ingress-host-name": "10.124.211.141.nip.io"
    }
  ]
}
```

Configure TLS Certificate

If a custom TLS certificate is not available, you can create and use a self-signed certificate. An authenticated certificate can be added from the Deployer CLI. Use the following commands in the example to configure a TLS certificate.

```
product opshub# config terminal
Entering configuration mode terminal
product example deployer(config)# clusters {k8s-cluster-name}

product example deployer(config-clusters-******)# secrets tls ?
Possible completions:
  Kubernetes namespaces to create the secet  range
product example deployer(config-clusters-******)# secrets tls opshub-data cert-api-ingress
 ?
Possible completions:
  certificate   Path to PEM encoded public key certificate.
  private-key   Private key associated with given certificate.
  <cr>
```

Configure cnBR using Autodeployer

You can complete the Cisco cnBR configuration using the Autodeployer.

Complete the following steps:

Step 1 Prepare Cisco cnBR configuration.

There are three categories of configuration:

- **General configuration**

  The general configuration specifies details of the Cisco cnBR and Cisco Operations Hub clusters.

```
opshub : 'xx.xx.xx.xx'           # Operations Hub IP address
cnbr   :
    name : '<name_of_cnbr>'      # Name of the cnBR cluster to be added to the Operations Hub
    type : 'MUL_NODES'           # cnBR cluster Type : 'MUL_NODES' Multi-Node cluster is only
```

```
option supported.
    ip   : 'xx.xx.xx.xx'         # cnBR IP address
```

- **Mandatory configuration**

  The mandatory configuration specifies details for the PTP, BGP, CIN, Wiring, templates (SG and L3) and rpd-list.

  Complete the following mandatory configurations:

  - PTP configuration

    ```
    ptp :
        v4 :
           domain : <clock-domain>
           master: {'ip':"xx.xx.xx.xx", 'gw':"xx.xx.xx.xx"}
    ```

  - BGP Agent configuration

    ```
    bgpagent :
        asn : <asn>
        max_hops : <max_hops>
        restart-time : <restart_time>
        stale-path-time: <stale_path_time>
        # list of neighbors ( IPv4 and IPv6 )
        neighbors :
          - {'address' :'xx.xx.xx.xx', 'asn':<asn>}
    ```

  - CIN configuration

    ```
    # Lists of IPv4 and IPv6 gateways. IPv6 is not supported in this release.
    cin :
        v4 : [ "xx.xx.xx.xx"]
    ```

  - Wiring configuration

    ```
    wiring :
        # Starting IP address for the range to be used by cnBR internal interfaces
        # Make sure the range does not crash with IP addresses of RPD, COPS, and CCAPCORE
        # IP addresses that will be carved out from this pool to assign to the below interfaces
        #     PTP, VPP-DP and other interface
        cin-start-ip:
           v4 : 'xx.xx.xx.xx'

        # SG peer IP, typically its bgp-neighbor IP address but it could be different
        #     dmic-if and relayproxy-if addresses are carved out from the same network
        sg-peer:
           v4 : 'xx.xx.xx.xx'
           v6 : 'xxxx::nnn'  #Needs dummy value even if IPv6 is not enabled. nnn is <0-255>

        # ccapcore IP, specified in the DHCP config, where RPD learn ccapcore from
        rphmgr-if:
           v4 : "xx.xx.xx.xx"

        # Packet cable interface IP
        cmts-cops-if:
           v4 : "xx.xx.xx.xx"

        # IP addresses to be used by BGP agents running in cnBR
        # AIO needs one and MultiNode needs two as that many instances of bgp agents would be
    running in the cluster
        bgp-agent-if:
           v4 : ["xx.xx.xx.xx", "xx.xx.xx.xx"]
    ```

```
      v6 : ["xxxx::xxxx", "xxxx::xxxx"]  #Needs dummy values even if IPv6 is not enabled.

   # CIN Prefix
   cin-prefix:
      v4 : <prefix_len>
      v6 : <prefix_len> #Needs dummy value even if IPv6 is not enabled due to known issue

   # DC link prefix to be used by CIMC interfaces within cnBR
   # v4 and v6 prefixes are mandatory for now due to an internal issue, even if v6 is not
enabled.
   #   will have a fix in the next release.
   dc-link-prefix:
      v4 : <prefix_len>
      v6 : <prefix_len> #Needs dummy value even if IPv6 is not enabled due to known issue

   # VLAN or VXLAN config, whichever is applicable
   vlan :
      cnbr-wan-ifname: "<name>/<bay>/<slot>"
      overlay-wan-vlan: <xxxx>
      overlay-cin-vlan: <xxxx>
      overlay-l2vpn-vlan-vlan: <xxxx>
      overlay-l2vpn-mpls-vlan: <xxxx>
   vxlan :
      sp-router-wan-ip: "xx.xx.xx.xx"
      cnbr-wan-prefix: <prefix_len>
      cnbr-wan-ip: "xx.xx.xx.xx"
      cnbr-wan-ifname: "<name>/<bay>/<slot>"
      cnbr-loopback-ip: "xx.xx.xx.xx"
      sp-router-loopback-ip: "xx.xx.xx.xx"
      overlay-cin-vni: <cin-vni>
      overlay-l2vpn-mpls-vni: <mpls-vni>
      overlay-l2vpn-vlan-vni: <vlan-vni>
      overlay-wan-vni: <wan-vni>
   # MTU used by cnBR SG
   mtu : "2450"
```

- Service Group (SG) and RPD List: Specify the list of RPDs to be loaded as rpd-list. File paths are relative to the staging directory or the directory from where you are running autodeploy. Go through Autodeployer Examples, on page 36 for examples on L3 Template and SG Template.

```
templates:
    # List of L3 templates in the {<name>:<file_path>} format
    L3 :
        'L3-1' : '<L3 template1 file>'
        'L3-2' : '<L3 template2 file>'

    # List of SG templates in the {<name>:<file_path>} format
    SG :
        '4x4_SG_Config'  : '<SG template1 file>'
        '33x8_SG_Config' : '<SG template2 file>'

# RPD location
RPD-loc1: &loc1
         region: "<region>"
         city: "<city>"
         neighborhood: "<neighborhood>"
         address: "<address>"
         latitude: <latitude>
         longitude: <longitude>

# List of RPDs
rpd-list:
   # [ 'rpd-name', 'rpd-mac', 'SG_name', 'SG_tmpl', 'L3_tmpl', 'RPD_location']
```

```
    - [ 'RPD-00', 'xx:xx:xx:xx:xx:xx', 'SG00', '33x8_SG_Config', 'L3-1', *loc1 ]
    - [ 'RPD-01', 'xx:xx:xx:xx:xx:xx', 'SG01', '33x8_SG_Config', 'L3-1', *loc1 ]
    - [ 'RPD-02', 'xx:xx:xx:xx:xx:xx', 'SG02', '4x4_SG_Config', 'L3-2', *loc1 ]
```

- **Optional configuration**

  Choose the optional configurations required. The configuration specifies details for L2VPN, L3VPN, TFTP, PacketCable, RIP, SAV, and PFG:

```
# Specify, if tftpProxy is different from CIN gateway
tftpProxy:
    v4 : ["xx.xx.xx.xx"]
    v6 : ["xx:xx:xx:xx:xx:xx:xx:xx"] #specify, if IPv6 is enabled

# cops interface in wiring config needs to be set to enable this feature.
packetcable :
    enable: 'true'
    max-gate: <value>
    t0: <value>
    t1: <value>
    subscriber: 'false'

l2vpn :
    dot1qvc :
       - {'mac':"xxxx.xxxx.xxxx", 'vlan':<vlan>, 'vpn':"<name>"}
    mplsvc :
       - {'mac':"xxxx.xxxx.xxxx", 'peerip':<peerip>, 'vc': 1, 'vpn':"<name>", 'experimental':0}
    mplsvlansg :
       - {'sg':"xxxx.xxxx.xxxx", 'vlan_max':<vlan_max>, 'vlan_min':0}
    sprstat :
       - {'id':"xxxx.xxxx.xxxx", 'asn':<asn>, 'state':'Up'}

l3vpn:
    - {"name" : "<name>", "vlan" : <vlan>, "vpn" : "<name>"}

rip :
    enable : 'false'
    update-timer : <time in seconds>
    invalid-timer : <time in seconds>
    holddown-timer : <time in seconds>
    passive-mode' : 'false'

sav:
    enable : 'true'
    entries:
       - grp-name : "testSAV"
         prefixes : [ "xx.xx.xx.xx/<prefix_len>" , "xx:xx:xx:xx:xx:xx:xx/<prefix_len>" ]

pfgactive:

{"cm_ds":-1,"cm_us":-1,"host_ds":-1,"host_us":-1,"mta_ds":-1,"mts_us":-1,"stb_ds":-1,"stb_us":-1,"ps_ds":-1,"ps_us":-1}

pfg:
     - id : 1
       rules :
          - {"isPermit":0, "isIpv6":0, "srcIp":'xx.xx.xx.xx/<prefix_len>',
"dstIp":"xx.xxx.xx.xx/<prefix_len>"}
```

Step 2 Apply the configuration.

Run the deploy command to apply the configuration and monitor the status through the Cisco Operations Hub or CLI. You can update the configuration file to add, delete or update the SGs or RPDs and rerun the command to apply the updated configuration.

```
$ ./deploy -c cnbr_config.yaml
```

The configuration file must strictly conform to YAML syntax, to avoid any crash dumps.

Autodeployer Examples

- Configuration file

```
opshub : 'xx.xx.xx.xx'
cnbr :
    name : 'cnbr001'
    type : 'MUL_NODES'
    ip   : 'xx.xx.xx.xx'
ptp :
    v4 :
       domain : 0
       master: {'ip':"xx.xx.xx.xx", 'gw':"xx.xx.xx.xx"}
bgpagent :
    asn : 65224
    max_hops : 255
    restart-time : 120
    stale-path-time: 360
    neighbors :
       - {'address' :'xx.xx.xx.xx', 'asn':65534}
cin :
    v4 : ["xx.xx.xx.xx"]
wiring :
    cin-start-ip:
       v4 : 'xx.xx.xx.xx'
    sg-peer:
       v4 : 'xx.xx.xx.xx'
    bgp-agent-if:
       v4 : ["xx.xx.xx.xx", "xx.xx.xx.xx"]
       v6 : ["xx:xx:xx:xx::1", "xx:xx:xx:xx::1"]
    rphmgr-if:
       v4 : "xx.xx.xx.xx"
    cmts-cops-if:
       v4 : "xx.xx.xx.xx"
    cin-prefix:
       v4 : 24
       v6 : 64
    dc-link-prefix:
       v4 : 24
       v6 : 64
    vlan :
       cnbr-wan-ifname: "FortyGigabitEthernetb/0/0"
       overlay-wan-vlan: 1001
       overlay-cin-vlan: 1002
       overlay-l2vpn-vlan-vlan: 1007
       overlay-l2vpn-mpls-vlan: 1008
    mtu : "2450"

templates:
    L3 :
```

```
        # {'template_name' : 'template_file_location'}
        'L3_1' : 'l3_template1.json'
    SG :
        # {'template_name' : 'template_file_location'}
        'SG_16x4' : 'sg_template1.json'

RPD-loc: &loc1
    region: "CA"
    city: "SanJose"
    neighborhood: "XXXX"
    address: "XXXXXXX"
    latitude: 0
    longitude: 0

rpd-list:
    # [ 'rpd-name', 'rpd-mac', 'SG_name', 'SG_tmpl', 'L3_tmpl', 'RPD_location']
    - [ 'RPD-00', '78:72:5D:39:26:64', 'SG00', 'SG_16x4', 'L3_1', *loc1 ]
    - [ 'RPD-01', 'F4:DB:
```

- L3 Template

```
{
  "dhcp": {
    "arpGlean": true,
    "arpProxy": true,
    "dhcpIfname": "cnr",
    "dhcpServers": [
      "xx.xx.xx.xx"
    ],
    "ipv6Lq": true,
    "mobilityScopes": [
      "xx.xx.xx.xx/<prefix_len>",
      "xx:xx:xx:xx:xx:xx:xx:xx/<prefix_len>"
    ],
    "ndProxy": true,
    # Add relayPolicies, if applicable to your setup
    "relayPolicies": [
      {
          "deviceClass": "HOST",
          "giAddr": "xx.xx.xx.xx",
          "linkAddr": "xxxx::xxxx",
          "v4ServerIp": "xx.xx.xx.xx"
      }
    ],
    "relayModeV4": 0,
    "relayModeV6": 0,
    "v4Nets": [
      "xx.xx.xx.xx/<prefix_len>"
    ],
    "v6Nets": [
      "xx:xx:xx:xx:xx:xx:xx:xx/<prefix_len>"
    ]
  },
  "spRouterName": "<SP router name>",
  "savList": {
    "prefixes": null
  },
  "sgPeerIpv4": "xx.xx.xx.xx/<prefix_len>",
  "sgPeerIpv6": "xx:xx:xx:xx:xx:xx:xx:xx/<prefix_len>"
}
```

- SG Template

```
{
  "description": "33x8 SG Config",
  "ds": [
    {
      "annex": "AnnexB",
      "attributeMask": 2147483648,
      "frequency": 255000000,
      "idInSg": 0,
      "interleaver": "fecI32J4",
      "modulation": "qam256",
      "powerAdjust": 0
    },
    {
      "annex": "AnnexB",
      "attributeMask": 2147483648,
      "frequency": 261000000,
      "idInSg": 1,
      "interleaver": "fecI32J4",
      "modulation": "qam256",
      "powerAdjust": 0
    },
    {
      "annex": "AnnexB",
      "attributeMask": 2147483648,
      "frequency": 267000000,
      "idInSg": 2,
      "interleaver": "fecI32J4",
      "modulation": "qam256",
      "powerAdjust": 0
    },
    {
      "annex": "AnnexB",
      "attributeMask": 2147483648,
      "frequency": 273000000,
      "idInSg": 3,
      "interleaver": "fecI32J4",
      "modulation": "qam256",
      "powerAdjust": 0
    },
    {
      "annex": "AnnexB",
      "attributeMask": 2147483648,
      "frequency": 279000000,
      "idInSg": 4,
      "interleaver": "fecI32J4",
      "modulation": "qam256",
      "powerAdjust": 0
    },
    {
      "annex": "AnnexB",
      "attributeMask": 2147483648,
      "frequency": 285000000,
      "idInSg": 5,
      "interleaver": "fecI32J4",
      "modulation": "qam256",
      "powerAdjust": 0
    },
    {
      "annex": "AnnexB",
      "attributeMask": 2147483648,
      "frequency": 291000000,
      "idInSg": 6,
      "interleaver": "fecI32J4",
```

```
    "modulation": "qam256",
    "powerAdjust": 0
  },
  {
    "annex": "AnnexB",
    "attributeMask": 2147483648,
    "frequency": 297000000,
    "idInSg": 7,
    "interleaver": "fecI32J4",
    "modulation": "qam256",
    "powerAdjust": 0
  },
  {
    "annex": "AnnexB",
    "attributeMask": 2147483648,
    "frequency": 303000000,
    "idInSg": 8,
    "interleaver": "fecI32J4",
    "modulation": "qam256",
    "powerAdjust": 0
  },
  {
    "annex": "AnnexB",
    "attributeMask": 2147483648,
    "frequency": 309000000,
    "idInSg": 9,
    "interleaver": "fecI32J4",
    "modulation": "qam256",
    "powerAdjust": 0
  },
  {
    "annex": "AnnexB",
    "attributeMask": 2147483648,
    "frequency": 315000000,
    "idInSg": 10,
    "interleaver": "fecI32J4",
    "modulation": "qam256",
    "powerAdjust": 0
  },
  {
    "annex": "AnnexB",
    "attributeMask": 2147483648,
    "frequency": 321000000,
    "idInSg": 11,
    "interleaver": "fecI32J4",
    "modulation": "qam256",
    "powerAdjust": 0
  },
  {
    "annex": "AnnexB",
    "attributeMask": 2147483648,
    "frequency": 327000000,
    "idInSg": 12,
    "interleaver": "fecI32J4",
    "modulation": "qam256",
    "powerAdjust": 0
  },
  {
    "annex": "AnnexB",
    "attributeMask": 2147483648,
    "frequency": 333000000,
    "idInSg": 13,
    "interleaver": "fecI32J4",
    "modulation": "qam256",
```

```
    "powerAdjust": 0
  },
  {
    "annex": "AnnexB",
    "attributeMask": 2147483648,
    "frequency": 339000000,
    "idInSg": 14,
    "interleaver": "fecI32J4",
    "modulation": "qam256",
    "powerAdjust": 0
  },
  {
    "annex": "AnnexB",
    "attributeMask": 2147483648,
    "frequency": 345000000,
    "idInSg": 15,
    "interleaver": "fecI32J4",
    "modulation": "qam256",
    "powerAdjust": 0
  },
  {
    "annex": "AnnexB",
    "attributeMask": 2147483648,
    "frequency": 351000000,
    "idInSg": 16,
    "interleaver": "fecI32J4",
    "modulation": "qam256",
    "powerAdjust": 0
  },
  {
    "annex": "AnnexB",
    "attributeMask": 2147483648,
    "frequency": 357000000,
    "idInSg": 17,
    "interleaver": "fecI32J4",
    "modulation": "qam256",
    "powerAdjust": 0
  },
  {
    "annex": "AnnexB",
    "attributeMask": 2147483648,
    "frequency": 363000000,
    "idInSg": 18,
    "interleaver": "fecI32J4",
    "modulation": "qam256",
    "powerAdjust": 0
  },
  {
    "annex": "AnnexB",
    "attributeMask": 2147483648,
    "frequency": 369000000,
    "idInSg": 19,
    "interleaver": "fecI32J4",
    "modulation": "qam256",
    "powerAdjust": 0
  },
  {
    "annex": "AnnexB",
    "attributeMask": 2147483648,
    "frequency": 375000000,
    "idInSg": 20,
    "interleaver": "fecI32J4",
    "modulation": "qam256",
    "powerAdjust": 0
```

```
},
{
  "annex": "AnnexB",
  "attributeMask": 2147483648,
  "frequency": 381000000,
  "idInSg": 21,
  "interleaver": "fecI32J4",
  "modulation": "qam256",
  "powerAdjust": 0
},
{
  "annex": "AnnexB",
  "attributeMask": 2147483648,
  "frequency": 387000000,
  "idInSg": 22,
  "interleaver": "fecI32J4",
  "modulation": "qam256",
  "powerAdjust": 0
},
{
  "annex": "AnnexB",
  "attributeMask": 2147483648,
  "frequency": 393000000,
  "idInSg": 23,
  "interleaver": "fecI32J4",
  "modulation": "qam256",
  "powerAdjust": 0
},
{
  "annex": "AnnexB",
  "attributeMask": 2147483648,
  "frequency": 399000000,
  "idInSg": 24,
  "interleaver": "fecI32J4",
  "modulation": "qam256",
  "powerAdjust": 0
},
{
  "annex": "AnnexB",
  "attributeMask": 2147483648,
  "frequency": 405000000,
  "idInSg": 25,
  "interleaver": "fecI32J4",
  "modulation": "qam256",
  "powerAdjust": 0
},
{
  "annex": "AnnexB",
  "attributeMask": 2147483648,
  "frequency": 411000000,
  "idInSg": 26,
  "interleaver": "fecI32J4",
  "modulation": "qam256",
  "powerAdjust": 0
},
{
  "annex": "AnnexB",
  "attributeMask": 2147483648,
  "frequency": 417000000,
  "idInSg": 27,
  "interleaver": "fecI32J4",
  "modulation": "qam256",
  "powerAdjust": 0
},
```

```
    {
      "annex": "AnnexB",
      "attributeMask": 2147483648,
      "frequency": 423000000,
      "idInSg": 28,
      "interleaver": "fecI32J4",
      "modulation": "qam256",
      "powerAdjust": 0
    },
    {
      "annex": "AnnexB",
      "attributeMask": 2147483648,
      "frequency": 429000000,
      "idInSg": 29,
      "interleaver": "fecI32J4",
      "modulation": "qam256",
      "powerAdjust": 0
    },
    {
      "annex": "AnnexB",
      "attributeMask": 2147483648,
      "frequency": 435000000,
      "idInSg": 30,
      "interleaver": "fecI32J4",
      "modulation": "qam256",
      "powerAdjust": 0
    },
    {
      "annex": "AnnexB",
      "attributeMask": 2147483648,
      "frequency": 441000000,
      "idInSg": 31,
      "interleaver": "fecI32J4",
      "modulation": "qam256",
      "powerAdjust": 0
    }
  ],
  "dsg": {
    "cfr": null,
    "chanList": null,
    "clientList": null,
    "tg": null,
    "timer": null,
    "tunnel": null
  },
  "dsmtu": 2100,
  "md": [
    {
      "adminState": "Up",
      "cmInitChanTimeout": 60,
      "dataBackoff": {
        "end": 5,
        "start": 3
      },
      "dsg": {
        "dcdDisable": null,
        "tg": null
      },
      "enableBalanceUs": true,
      "idInSg": 0,
      "insertionInterval": 120,
      "ipInit": "ipv4",
      "mac": "00:00:00:00:00:00",
      "mapAdvance": {
```

```
      "advanceTime": 2000,
      "mode": "static"
    },
    "primDcid": [
      0,
      8,
      16,
      24
    ],
    "rangeBackoff": {
      "end": 6,
      "start": 3
    },
    "registrationTimeout": 3,
    "syncInterval": 10,
    "ucId": [
      0,
      1,
      2,
      3
    ]
  }
],
"modProfs": [
  {
    "entries": {
      "advPhyLongData": {
        "channelType": "atdma",
        "fecCodewordLength": 232,
        "fecErrorCorrection": 9,
        "lastCodewardShortened": true,
        "modulation": "qam64",
        "preamble": "qpsk1",
        "preambleLength": 64,
        "scrambler": true,
        "scramblerSeed": 338
      },
      "advPhyShortData": {
        "channelType": "atdma",
        "fecCodewordLength": 76,
        "fecErrorCorrection": 6,
        "lastCodewardShortened": true,
        "maxBurstSize": 6,
        "modulation": "qam64",
        "preamble": "qpsk1",
        "preambleLength": 64,
        "scrambler": true,
        "scramblerSeed": 338
      },
      "initialRanging": {
        "channelType": "atdma",
        "fecCodewordLength": 34,
        "fecErrorCorrection": 5,
        "modulation": "qpsk",
        "preamble": "qpsk0",
        "preambleLength": 98,
        "scrambler": true,
        "scramblerSeed": 338
      },
      "longData": {
        "fecCodewordLength": 2,
        "fecErrorCorrection": 9,
        "lastCodewardShortened": true,
        "modulation": "qam16",
```

```
            "preambleLength": 4,
            "scrambler": true
          },
          "periodicRanging": {
            "channelType": "atdma",
            "fecCodewordLength": 34,
            "fecErrorCorrection": 5,
            "modulation": "qpsk",
            "preamble": "qpsk0",
            "preambleLength": 98,
            "scrambler": true,
            "scramblerSeed": 338
          },
          "request": {
            "channelType": "atdma",
            "fecCodewordLength": 16,
            "modulation": "qpsk",
            "preamble": "qpsk0",
            "preambleLength": 36,
            "scrambler": true,
            "scramblerSeed": 338
          },
          "shortData": {
            "fecCodewordLength": 6,
            "fecErrorCorrection": 3,
            "lastCodewardShortened": true,
            "maxBurstSize": 2,
            "modulation": "qam16",
            "scrambler": true
          },
          "ugs": {
            "channelType": "atdma",
            "fecCodewordLength": 232,
            "fecErrorCorrection": 9,
            "lastCodewardShortened": true,
            "modulation": "qam64",
            "preamble": "qpsk1",
            "preambleLength": 64,
            "scrambler": true,
            "scramblerSeed": 338
          }
        },
        "idInSg": 221
      }
    ],
    "ofdmDs": [
      {
        "cyclicPrefix": 256,
        "idInSg": 158,
        "interleaverDepth": 16,
        "pilotScaling": 48,
        "plc": 930000000,
        "profileControl": "QAM256",
        "profileNcp": "QAM16",
        "rollOff": 192,
        "startFrequency": 837000000,
        "subcarrierSpacing": "25KHZ",
        "width": 192000000
      }
    ],
    "privacy": {
      "AcceptSelfSignCert": true,
      "BpiPlusPolicy": "capable-enforcement",
      "DsxSupport": true,
```

```
  "EaePolicy": "disable-enforcement",
  "Kek": {
    "GraceTime": 300,
    "LifeTime": 86400
  },
  "Tek": {
    "GraceTime": 300,
    "LifeTime": 1800
  }
},
"punt": {
  "icpiPerCausePuntCfgList": null
},
"rpdCfg": {
    "rfTopology": {
      "dsPort": [
        {
          "adminState": "Up",
          "basePower": 21,
          "channel": [
            0,
            1,
            2,
            3,
            4,
            5,
            6,
            7,
            8,
            9,
            10,
            11,
            12,
            13,
            14,
            15,
            16,
            17,
            18,
            19,
            20,
            21,
            22,
            23,
            24,
            25,
            26,
            27,
            28,
            29,
            30,
            31,
            158
          ],
          "ofdmFreqExclBand": null
        }
      ],
      "fiberNode": [
        {
          "dsPort": [0],
          "usPort": [0]
        },
        {
          "dsPort": 0,
```

```
            "id": 1,
            "usPort": 1
          }
        ],
        "usPort": [
          {
            "channel": [
              0,
              1
            ],
            "ofdmaFreqExclBand": null,
            "ofdmaFreqUnusedBand": null
          },
          {
            "channel": [
              2,
              3
            ],
            "ofdmaFreqExclBand": null,
            "ofdmaFreqUnusedBand": null,
            "portId": 1
          }
        ]
      }
    },
    "rpdPtpCfg": {
      "domain": 0,
      "dtiMode": "SlaveDtiMode",
      "priority1": 128,
      "priority2": 255,
      "ptpClkProfileId": "00:00:00:00:00:00",
      "ptpPortCfg": [
        {
          "adminState": "Up",
          "anncReceiptTimeout": 11,
          "cos": 6,
          "dscp": 47,
          "enetPortIndex": 1,
          "gateway": "3.208.1.2",
          "localPriority": 128,
          "logDelayReqInterval": -4,
          "logSyncInterval": -4,
          "masterAddr": "3.158.185.51",
          "masterAdminState": "Up",
          "ptpPortIndex": 22,
          "unicastDuration": 300
        }
      ]
    },
    "us": [
      {
        "adminState": "Up",
        "attributeMask": 2684354560,
        "channelWidth": 6400000,
        "docsisMode": "atdma",
        "equalizationCoeffEnable": true,
        "frequency": 11400000,
        "idInSg": 0,
        "ingressNoiseCancelEnable": true,
        "modulation": 221,
        "powerLevel": 0,
        "slotSize": 1
      },
      {
```

```
        "adminState": "Up",
        "attributeMask": 2684354560,
        "channelWidth": 6400000,
        "docsisMode": "atdma",
        "equalizationCoeffEnable": true,
        "frequency": 17800000,
        "idInSg": 1,
        "ingressNoiseCancelEnable": true,
        "modulation": 221,
        "powerLevel": 0,
        "slotSize": 1
      },
      {
        "adminState": "Up",
        "attributeMask": 2684354560,
        "channelWidth": 6400000,
        "docsisMode": "atdma",
        "equalizationCoeffEnable": true,
        "frequency": 24200000,
        "idInSg": 2,
        "ingressNoiseCancelEnable": true,
        "modulation": 221,
        "powerLevel": 0,
        "slotSize": 1
      },
      {
        "adminState": "Up",
        "attributeMask": 2684354560,
        "channelWidth": 6400000,
        "docsisMode": "atdma",
        "equalizationCoeffEnable": true,
        "frequency": 30600000,
        "idInSg": 3,
        "ingressNoiseCancelEnable": true,
        "modulation": 221,
        "powerLevel": 0,
        "slotSize": 1
      }
    ],
    "usmtu": 2100
}
```

Autodeployer Limitations

In the Cisco cnBR Release 20.2, the Autodeployer has the following limitations:

- Rerunning the deploy command reapplies all configurations, except the wiring configuration. The wiring configuration update is not supported.
- When updating the SG or RPD, the existing service groups are deleted and the SG or RPD is then added back with the updated configuration.
- Placeholder values for IPv6 must be provided, even if IPv6 is not supported. Values for `sg-peer`, `bgp-agent-if`, `cin-prefix`, and `dc-link-prefix` must be as specified in the given example.
- The configuration file must specify all mandatory sections and attributes. You may see the autodeploy exit without warnings and errors when mandatory attributes are missing in the configuration file.
- Cisco cnBR has limited error and exception handling. Review the detailed crash dumps when an exception or error occurs.

Configure cnBR using Configurator

You can complete the Cisco cnBR configuration using the Configurator.

Adding cnBR to the Operations Hub

To add Cisco cnBR cores using the Cisco Operations Hub, complete the following steps:

Step 1 On the Cisco Operations Hub, click **Configurator** > **cnBR-Core Manage** > **Add cnBR Core**.

Step 2 Provide a unique name to the Cisco cnBR core, a namespace, and Core Ingress-host-name. For example:

```
cnBR-Core Name: cnbr-demo
Core Namespace: ccmts-infra
Core Ingress-host-name: 10.124.210.65.nip.io
```

Step 3 Enter the Cisco cnBR username and password.

Step 4 Click **ADD**.

Apply Global Configuration to cnBR

Complete the following steps to configure Wiring, BGP, PTP, and CIN:

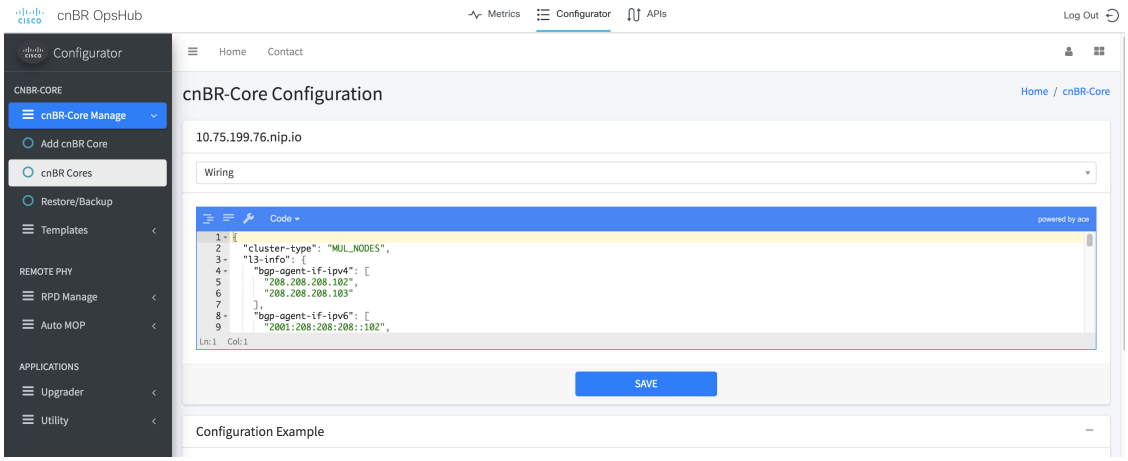
Step 1 Configure Wiring.

a) On the Cisco Operations Hub, click **Configurator** > **cnBR-Core Manage** > **cnBR Cores**.

b) Select a Cisco cnBR core cluster.

We recommend that you use the `Code` mode to configure wiring.

Figure 10: cnBR-Core Configuration Pane



c) Click **SAVE** to apply configuration to Cisco cnBR.

For example:

```
{
  "cluster-type": "MUL_NODES",
  "l3-info": {
    "bgp-agent-if-ipv4": [
      "208.208.208.102",      <---bgp address
      "208.208.208.103"
    ],
    "bgp-agent-if-ipv6": [
      "2001:208:208:208::102",
      "2001:208:208:208::103"
    ],
    "cin-ipv4-prefix": 24,
    "cin-ipv6-prefix": 64,
    "cmts-cops-if-ipv4": [
      "3.208.1.7",
      "3.208.1.8"
    ],
    "cmts-cops-if-ipv6": [],
    "dc-link-ipv4-prefix": 24,
    "dc-link-ipv6-prefix": 64,
    "dmic-if-ipv4": [
      "200.200.200.9",
      "200.200.200.10",
      "200.200.200.11"
    ],
    "dmic-if-ipv6": [
      "2008:199:1:1::9",
      "2008:199:1:1::10",
      "2008:199:1:1::11"
    ],
    "ptp-if-ipv4": [
      "3.208.1.4",      <---PTP local address
      "3.208.1.5",
      "3.208.1.6"
    ],
    "ptp-if-ipv6": [],
    "ptp-mac-addr": [
      "20:18:10:29:88:43",
      "20:18:10:29:88:44",
      "20:18:10:29:88:45"
    ],
    "relayproxy-if-ipv4": [
      "208.208.208.107",
      "208.208.208.108",
      "208.208.208.109"
    ],
    "relayproxy-if-ipv6": [
      "2001:208:208:208::107",
      "2001:208:208:208::108",
      "2001:208:208:208::109"
    ],
    "rphmgr-if-ipv4": [
      "3.208.1.3",
      "3.208.1.3"
    ],
    "rphmgr-if-ipv6": [],
    "vpp-dp-rpd-if-ipv4": [ <---15 addresses total
      "3.208.1.10",
      "3.208.1.11",
      "3.208.1.12",
      "3.208.1.13",
```

```
        "3.208.1.14",
        "3.208.1.15",
        "3.208.1.16",
        "3.208.1.17",
        "3.208.1.18",
        "3.208.1.19",
        "3.208.1.20",
        "3.208.1.21",
        "3.208.1.22",
        "3.208.1.23",
        "3.208.1.24"
      ],
      "vpp-dp-rpd-if-ipv6": []
    },
    "mtu": 2450,           <---Recommend value is 2450
    "overlay-info": {
      "overlay-type": "vlan",
      "vlan-info": {
        "cnbr-wan-ifname": "FortyGigabitEthernetb/0/0",
        "overlay-cin-vlan": 1182,          <---This vlan id should be same as vlan id in SP router

        "overlay-l2vpn-mpls-vlan": 1183,
        "overlay-l2vpn-vlan-vlan": 1184,
        "overlay-wan-vlan": 1181           <---This vlan id should be same as vlan id in SP router

      }
    }
  }
```

Step 2 Configure BGP.

a) Use the `Code` mode to configure BGP.

b) Click **SAVE** to apply configuration to Cisco cnBR.

For example:

```
{
  "asNumber": 65001,
  "ebgpMultihop": 255,
  "gracefulRestart": {
    "enable": true,
    "restartTime": 120,
    "stalePathTime": 360
  },
  "ifname": "bgp",
  "neighbors": [
    {
      "address": "208.208.208.1",   <----IP in SP Router. Same IP with SG Peer.
      "asNumber": 65000
    },
    {
      "address": "2001:208:208:208::1",
      "asNumber": 65000
    }
  ]
}
```

Step 3 Configure PTP.

a) Use the `Code` mode to configure PTP.

b) Click **SAVE** to apply configuration to Cisco cnBR.

For example:

```
PTP:
{
  "PtpDomain": 44,
  "PtpGwIp": "3.208.1.2",
  "PtpMasterIp": "3.158.185.51"
}
```

Step 4 Configure CIN.

If RPD and RPHYMAN are in different networks, you must configure CIN. Otherwise, choose to ignore this step.

a) Use the `Code` mode to configure CIN.

For example:

```
{
  "CinGwIp": "3.208.1.2"
}
```

Add Service Group Configuration to cnBR

Complete the following steps to add Service Group (SG) template and L3 template:

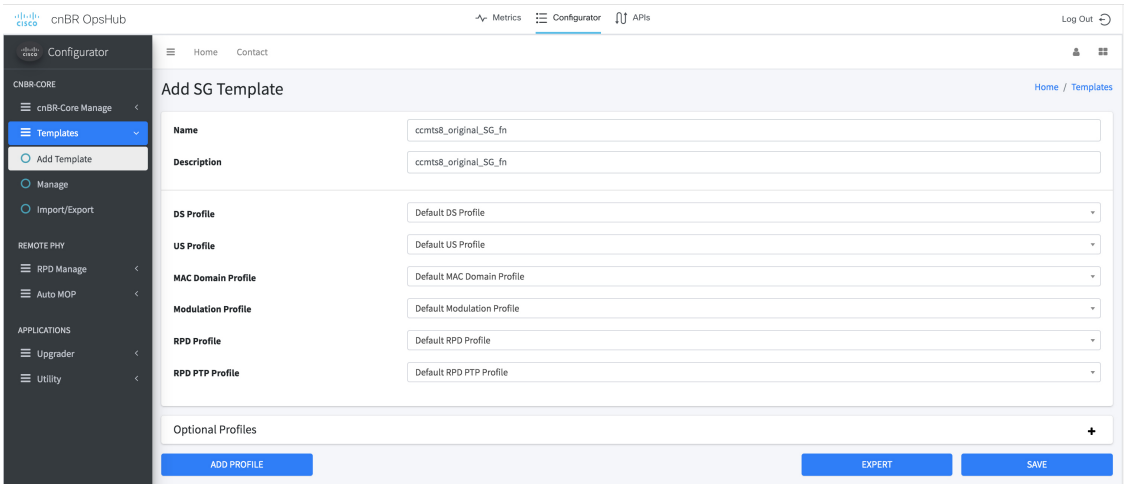
Step 1 On the Cisco Operations Hub, click **Configurator** > **Templates** > **Add Template**.

Step 2 Choose **SG Template** as the template type.

Step 3 Provide an appropriate template Name and Description. Click **Next**.

Step 4 On the `Add SG Template` pane, choose to ignore the profile changes. Click **EXPERT**.

Figure 11: Add SG Template Pane



Step 5 Provide the SG related configuration and click **SAVE**.

For example:

```
{
  "description": "33x8 SG Config",
  "ds": [
    {
      "annex": "AnnexB",
      "attributeMask": 2147483648,
      "frequency": 255000000,
      "idInSg": 0,
      "interleaver": "fecI32J4",
      "modulation": "qam256",
      "powerAdjust": 0
    },
    {
      "annex": "AnnexB",
      "attributeMask": 2147483648,
      "frequency": 261000000,
      "idInSg": 1,
      "interleaver": "fecI32J4",
      "modulation": "qam256",
      "powerAdjust": 0
    },
    {
      "annex": "AnnexB",
      "attributeMask": 2147483648,
      "frequency": 267000000,
      "idInSg": 2,
      "interleaver": "fecI32J4",
      "modulation": "qam256",
      "powerAdjust": 0
    },
    {
      "annex": "AnnexB",
      "attributeMask": 2147483648,
      "frequency": 273000000,
      "idInSg": 3,
      "interleaver": "fecI32J4",
      "modulation": "qam256",
      "powerAdjust": 0
    },
    {
      "annex": "AnnexB",
      "attributeMask": 2147483648,
      "frequency": 279000000,
      "idInSg": 4,
      "interleaver": "fecI32J4",
      "modulation": "qam256",
      "powerAdjust": 0
    },
    {
      "annex": "AnnexB",
      "attributeMask": 2147483648,
      "frequency": 285000000,
      "idInSg": 5,
      "interleaver": "fecI32J4",
      "modulation": "qam256",
      "powerAdjust": 0
    },
    {
      "annex": "AnnexB",
      "attributeMask": 2147483648,
      "frequency": 291000000,
      "idInSg": 6,
      "interleaver": "fecI32J4",
```

```
    "modulation": "qam256",
    "powerAdjust": 0
  },
  {
    "annex": "AnnexB",
    "attributeMask": 2147483648,
    "frequency": 297000000,
    "idInSg": 7,
    "interleaver": "fecI32J4",
    "modulation": "qam256",
    "powerAdjust": 0
  },
  {
    "annex": "AnnexB",
    "attributeMask": 2147483648,
    "frequency": 303000000,
    "idInSg": 8,
    "interleaver": "fecI32J4",
    "modulation": "qam256",
    "powerAdjust": 0
  },
  {
    "annex": "AnnexB",
    "attributeMask": 2147483648,
    "frequency": 309000000,
    "idInSg": 9,
    "interleaver": "fecI32J4",
    "modulation": "qam256",
    "powerAdjust": 0
  },
  {
    "annex": "AnnexB",
    "attributeMask": 2147483648,
    "frequency": 315000000,
    "idInSg": 10,
    "interleaver": "fecI32J4",
    "modulation": "qam256",
    "powerAdjust": 0
  },
  {
    "annex": "AnnexB",
    "attributeMask": 2147483648,
    "frequency": 321000000,
    "idInSg": 11,
    "interleaver": "fecI32J4",
    "modulation": "qam256",
    "powerAdjust": 0
  },
  {
    "annex": "AnnexB",
    "attributeMask": 2147483648,
    "frequency": 327000000,
    "idInSg": 12,
    "interleaver": "fecI32J4",
    "modulation": "qam256",
    "powerAdjust": 0
  },
  {
    "annex": "AnnexB",
    "attributeMask": 2147483648,
    "frequency": 333000000,
    "idInSg": 13,
    "interleaver": "fecI32J4",
    "modulation": "qam256",
```

```
    "powerAdjust": 0
  },
  {
    "annex": "AnnexB",
    "attributeMask": 2147483648,
    "frequency": 339000000,
    "idInSg": 14,
    "interleaver": "fecI32J4",
    "modulation": "qam256",
    "powerAdjust": 0
  },
  {
    "annex": "AnnexB",
    "attributeMask": 2147483648,
    "frequency": 345000000,
    "idInSg": 15,
    "interleaver": "fecI32J4",
    "modulation": "qam256",
    "powerAdjust": 0
  },
  {
    "annex": "AnnexB",
    "attributeMask": 2147483648,
    "frequency": 351000000,
    "idInSg": 16,
    "interleaver": "fecI32J4",
    "modulation": "qam256",
    "powerAdjust": 0
  },
  {
    "annex": "AnnexB",
    "attributeMask": 2147483648,
    "frequency": 357000000,
    "idInSg": 17,
    "interleaver": "fecI32J4",
    "modulation": "qam256",
    "powerAdjust": 0
  },
  {
    "annex": "AnnexB",
    "attributeMask": 2147483648,
    "frequency": 363000000,
    "idInSg": 18,
    "interleaver": "fecI32J4",
    "modulation": "qam256",
    "powerAdjust": 0
  },
  {
    "annex": "AnnexB",
    "attributeMask": 2147483648,
    "frequency": 369000000,
    "idInSg": 19,
    "interleaver": "fecI32J4",
    "modulation": "qam256",
    "powerAdjust": 0
  },
  {
    "annex": "AnnexB",
    "attributeMask": 2147483648,
    "frequency": 375000000,
    "idInSg": 20,
    "interleaver": "fecI32J4",
    "modulation": "qam256",
    "powerAdjust": 0
```

```
},
{
  "annex": "AnnexB",
  "attributeMask": 2147483648,
  "frequency": 381000000,
  "idInSg": 21,
  "interleaver": "fecI32J4",
  "modulation": "qam256",
  "powerAdjust": 0
},
{
  "annex": "AnnexB",
  "attributeMask": 2147483648,
  "frequency": 387000000,
  "idInSg": 22,
  "interleaver": "fecI32J4",
  "modulation": "qam256",
  "powerAdjust": 0
},
{
  "annex": "AnnexB",
  "attributeMask": 2147483648,
  "frequency": 393000000,
  "idInSg": 23,
  "interleaver": "fecI32J4",
  "modulation": "qam256",
  "powerAdjust": 0
},
{
  "annex": "AnnexB",
  "attributeMask": 2147483648,
  "frequency": 399000000,
  "idInSg": 24,
  "interleaver": "fecI32J4",
  "modulation": "qam256",
  "powerAdjust": 0
},
{
  "annex": "AnnexB",
  "attributeMask": 2147483648,
  "frequency": 405000000,
  "idInSg": 25,
  "interleaver": "fecI32J4",
  "modulation": "qam256",
  "powerAdjust": 0
},
{
  "annex": "AnnexB",
  "attributeMask": 2147483648,
  "frequency": 411000000,
  "idInSg": 26,
  "interleaver": "fecI32J4",
  "modulation": "qam256",
  "powerAdjust": 0
},
{
  "annex": "AnnexB",
  "attributeMask": 2147483648,
  "frequency": 417000000,
  "idInSg": 27,
  "interleaver": "fecI32J4",
  "modulation": "qam256",
  "powerAdjust": 0
},
```

```
    {
      "annex": "AnnexB",
      "attributeMask": 2147483648,
      "frequency": 423000000,
      "idInSg": 28,
      "interleaver": "fecI32J4",
      "modulation": "qam256",
      "powerAdjust": 0
    },
    {
      "annex": "AnnexB",
      "attributeMask": 2147483648,
      "frequency": 429000000,
      "idInSg": 29,
      "interleaver": "fecI32J4",
      "modulation": "qam256",
      "powerAdjust": 0
    },
    {
      "annex": "AnnexB",
      "attributeMask": 2147483648,
      "frequency": 435000000,
      "idInSg": 30,
      "interleaver": "fecI32J4",
      "modulation": "qam256",
      "powerAdjust": 0
    },
    {
      "annex": "AnnexB",
      "attributeMask": 2147483648,
      "frequency": 441000000,
      "idInSg": 31,
      "interleaver": "fecI32J4",
      "modulation": "qam256",
      "powerAdjust": 0
    }
  ],
  "dsg": {
    "cfr": null,
    "chanList": null,
    "clientList": null,
    "tg": null,
    "timer": null,
    "tunnel": null
  },
  "dsmtu": 2200,
  "md": [
    {
      "adminState": "Up",
      "cmInitChanTimeout": 60,
      "dataBackoff": {
        "end": 5,
        "start": 3
      },
      "dsg": {
        "dcdDisable": null,
        "tg": null
      },
      "enableBalanceUs": true,
      "idInSg": 0,
      "insertionInterval": 120,
      "ipInit": "ipv4",
     "mac": "00:00:00:00:00:00",   <----mark to all 0, cnBR will assign Mac domain mac automaticly
```

```
      "mapAdvance": {
        "advanceTime": 2000,
        "mode": "static"
      },
      "primDcid": [
        0,
        8,
        16,
        24
      ],
      "rangeBackoff": {
        "end": 6,
        "start": 3
      },
      "registrationTimeout": 3,
      "syncInterval": 10,
      "ucId": [
        0,
        1,
        2,
        3
      ]
    }
  ],
  "modProfs": [
    {
      "entries": {
        "advPhyLongData": {
          "channelType": "atdma",
          "fecCodewordLength": 232,
          "fecErrorCorrection": 9,
          "lastCodewardShortened": true,
          "modulation": "qam64",
          "preamble": "qpsk1",
          "preambleLength": 64,
          "scrambler": true,
          "scramblerSeed": 338
        },
        "advPhyShortData": {
          "channelType": "atdma",
          "fecCodewordLength": 76,
          "fecErrorCorrection": 6,
          "lastCodewardShortened": true,
          "maxBurstSize": 6,
          "modulation": "qam64",
          "preamble": "qpsk1",
          "preambleLength": 64,
          "scrambler": true,
          "scramblerSeed": 338
        },
        "initialRanging": {
          "channelType": "atdma",
          "fecCodewordLength": 34,
          "fecErrorCorrection": 5,
          "modulation": "qpsk",
          "preamble": "qpsk0",
          "preambleLength": 98,
          "scrambler": true,
          "scramblerSeed": 338
        },
        "longData": {
          "fecCodewordLength": 2,
          "fecErrorCorrection": 9,
          "lastCodewardShortened": true,
```

```
          "modulation": "qam16",
          "preambleLength": 4,
          "scrambler": true
        },
        "periodicRanging": {
          "channelType": "atdma",
          "fecCodewordLength": 34,
          "fecErrorCorrection": 5,
          "modulation": "qpsk",
          "preamble": "qpsk0",
          "preambleLength": 98,
          "scrambler": true,
          "scramblerSeed": 338
        },
        "request": {
          "channelType": "atdma",
          "fecCodewordLength": 16,
          "modulation": "qpsk",
          "preamble": "qpsk0",
          "preambleLength": 36,
          "scrambler": true,
          "scramblerSeed": 338
        },
        "shortData": {
          "fecCodewordLength": 6,
          "fecErrorCorrection": 3,
          "lastCodewardShortened": true,
          "maxBurstSize": 2,
          "modulation": "qam16",
          "scrambler": true
        },
        "ugs": {
          "channelType": "atdma",
          "fecCodewordLength": 232,
          "fecErrorCorrection": 9,
          "lastCodewardShortened": true,
          "modulation": "qam64",
          "preamble": "qpsk1",
          "preambleLength": 64,
          "scrambler": true,
          "scramblerSeed": 338
        }
      },
      "idInSg": 221
    }
  ],
  "ofdmDs": [
    {
      "cyclicPrefix": 256,
      "idInSg": 158,
      "interleaverDepth": 16,
      "pilotScaling": 48,
      "plc": 930000000,
      "profileControl": "QAM256",
      "profileNcp": "QAM16",
      "rollOff": 192,
      "startFrequency": 837000000,
      "subcarrierSpacing": "25KHZ",
      "width": 192000000
    }
  ],
  "privacy": {
    "AcceptSelfSignCert": true,
    "BpiPlusPolicy": "capable-enforcement",
```

```
  "DsxSupport": true,
  "EaePolicy": "disable-enforcement",
  "Kek": {
    "GraceTime": 300,
    "LifeTime": 86400
  },
  "Tek": {
    "GraceTime": 300,
    "LifeTime": 1800
  }
},
"punt": {
  "icpiPerCausePuntCfgList": null
},
"rpdCfg": [
  {
    "entries": {
      "dsPort": [
        {
          "adminState": "Up",
          "basePower": 21,
          "channel": [
            0,
            1,
            2,
            3,
            4,
            5,
            6,
            7,
            8,
            9,
            10,
            11,
            12,
            13,
            14,
            15,
            16,
            17,
            18,
            19,
            20,
            21,
            22,
            23,
            24,
            25,
            26,
            27,
            28,
            29,
            30,
            31,
            158
          ],
          "ofdmFreqExclBand": null
        }
      ],
      "fiberNode": [
        {
          "dsPort": 0,
          "usPort": 0
        },
```

```
        {
          "dsPort": 0,
          "id": 1,
          "usPort": 1
        }
      ],
      "usPort": [
        {
          "channel": [
            0,
            1
          ],
          "ofdmaFreqExclBand": null,
          "ofdmaFreqUnusedBand": null
        },
        {
          "channel": [
            2,
            3
          ],
          "ofdmaFreqExclBand": null,
          "ofdmaFreqUnusedBand": null,
          "portId": 1
        }
      ]
    },
    "rpdIp": "3.2.0.2",
    "rpdMac": "00:00:20:11:11:00"
  }
],
"rpdPtpCfg": {
  "domain": 44,
  "dtiMode": "SlaveDtiMode",
  "priority1": 128,
  "priority2": 255,
  "ptpClkProfileId": "00:00:00:00:00:00",
  "ptpPortCfg": [
    {
      "adminState": "Up",
      "anncReceiptTimeout": 11,
      "cos": 6,
      "dscp": 47,
      "enetPortIndex": 1,
      "gateway": "3.208.1.2",
      "localPriority": 128,
      "logDelayReqInterval": -4,
      "logSyncInterval": -4,
      "masterAddr": "3.158.185.51",
      "masterAdminState": "Up",
      "ptpPortIndex": 22,
      "unicastDuration": 300
    }
  ]
},
"sgName": "SG0",
"us": [
  {
    "adminState": "Up",
    "attributeMask": 2684354560,
    "channelWidth": 6400000,
    "docsisMode": "atdma",
    "equalizationCoeffEnable": true,
    "frequency": 11400000,
    "idInSg": 0,
```

```
      "ingressNoiseCancelEnable": true,
      "modulation": 221,
      "powerLevel": 0,
      "slotSize": 1
    },
    {
      "adminState": "Up",
      "attributeMask": 2684354560,
      "channelWidth": 6400000,
      "docsisMode": "atdma",
      "equalizationCoeffEnable": true,
      "frequency": 17800000,
      "idInSg": 1,
      "ingressNoiseCancelEnable": true,
      "modulation": 221,
      "powerLevel": 0,
      "slotSize": 1
    },
    {
      "adminState": "Up",
      "attributeMask": 2684354560,
      "channelWidth": 6400000,
      "docsisMode": "atdma",
      "equalizationCoeffEnable": true,
      "frequency": 24200000,
      "idInSg": 2,
      "ingressNoiseCancelEnable": true,
      "modulation": 221,
      "powerLevel": 0,
      "slotSize": 1
    },
    {
      "adminState": "Up",
      "attributeMask": 2684354560,
      "channelWidth": 6400000,
      "docsisMode": "atdma",
      "equalizationCoeffEnable": true,
      "frequency": 30600000,
      "idInSg": 3,
      "ingressNoiseCancelEnable": true,
      "modulation": 221,
      "powerLevel": 0,
      "slotSize": 1
    }
  ],
  "usmtu": 2200
}
```

Step 6 Click **Templates** > **Add Templates** and choose **L3** as the template type.

Step 7 Provide an appropriate template Name and Description. Click **Next**.

Step 8 Choose to ignore the DHCP profile. Click **NEXT**.

Step 9 Provide the L3 related configuration updates. Click **SAVE**.

For example:

```
{
  "dhcp": {
    "arpGlean": true,
    "arpProxy": true,
    "dhcpIfname": "cnr",
    "dhcpServers": [
      "20.11.0.52"
```

```
    ],
    "ipv6Lq": true,
    "mobilityScopes": [
      "10.1.1.1/24",
      "2001::a/88"
    ],
    "ndProxy": true,
    "relayModeV4": 0,
    "relayModeV6": 0,
    "v4Nets": [
      "208.1.0.2/24"
    ],
    "v6Nets": [
      "2001:100:208:1::1/64"
    ]
  },
  "spRouterName": "ccmts8-sp-router",
  "savList": {
    "prefixes": null
  },
  "sgGWMac": "20:19:03:13:19:43",
  "sgPeerIpv4": "208.208.208.1/24",            <-----IP in SP Router. SG Peer IP and BGP Peer IP
is same
  "sgPeerIpv6": "2001:208:208:208::1/64"
}
```

Step 10 Execute **RPD Add** auto-mop to add RPD one by one.

a) Click **RPD Manage** > **RPD Add**. Add the RPDs, one by one.

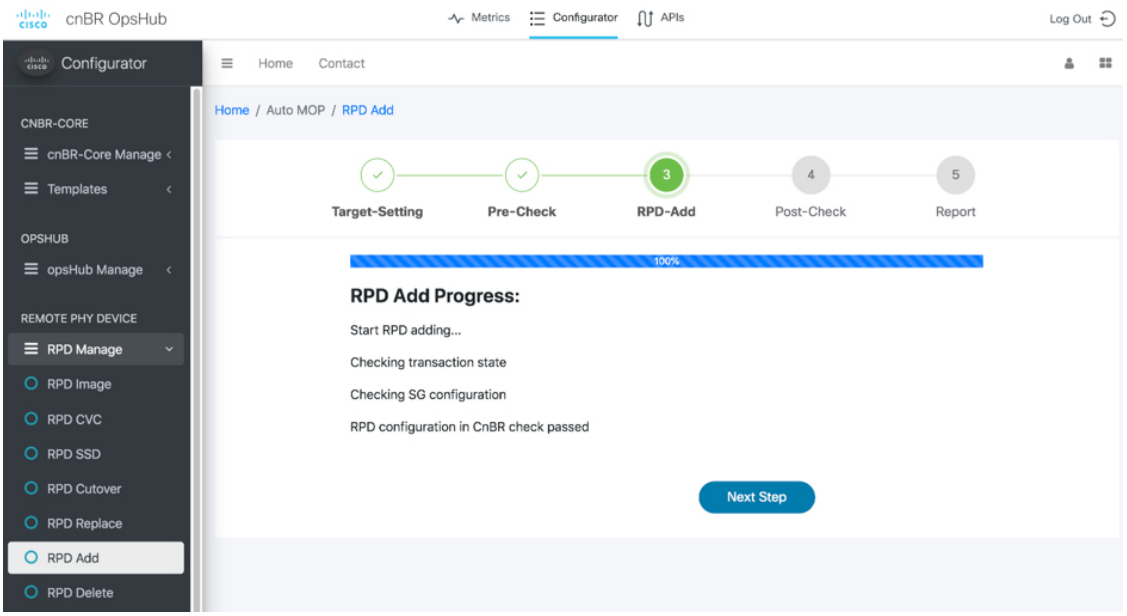
b) Set the target by providing all RPD related information.

c) Ensure that all `Pre-RPD-Add Checklist` conditions are ticked. Check the **Please confirm RPD has been connected physically and start RPD config adding** checkbox.

d) Click **Next Step**.

Wait for the `RPD Add progress` wizard to complete.

Figure 12: RPD Add Progress Wizard



e) To save time, you can alternatively choose to add another RPD during the `Post-check Progress`.

Step 11 Add consecutive RPDs to Cisco cnBR.

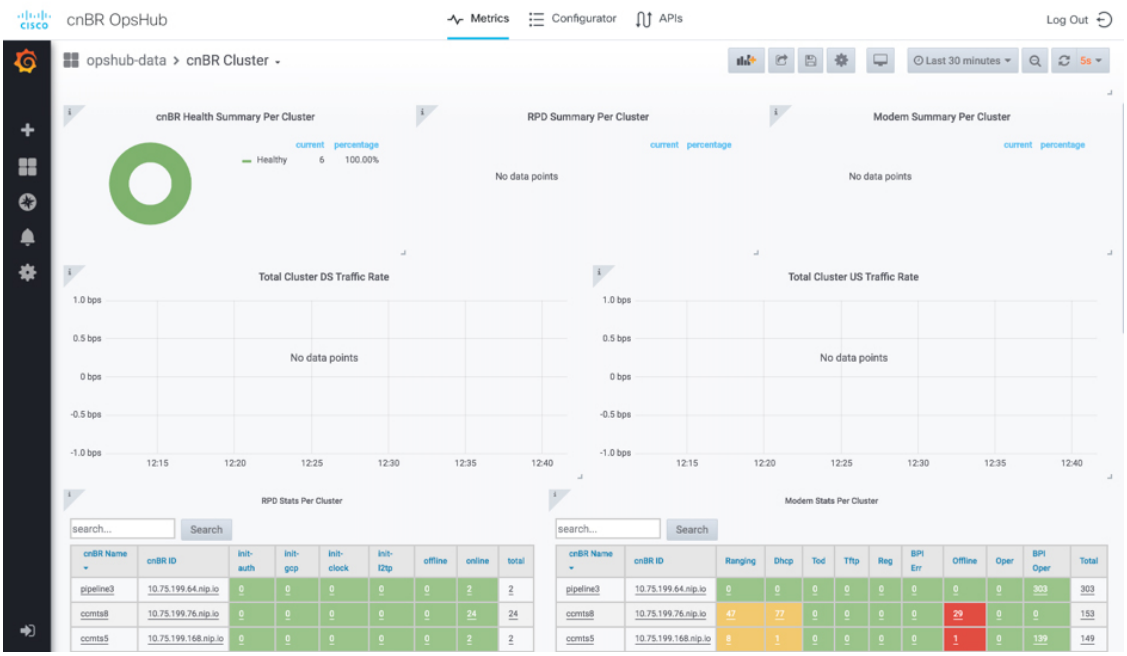
View RPD and Modem Status

You can view the RPD and modem status using Grafana.

To check the status of RPDs and CMs, complete the following step:

On the Cisco Operations Hub, click **Metrics** and search for **cnBR Cluster**.

Figure 13: RPD and Modem Status Dashboard

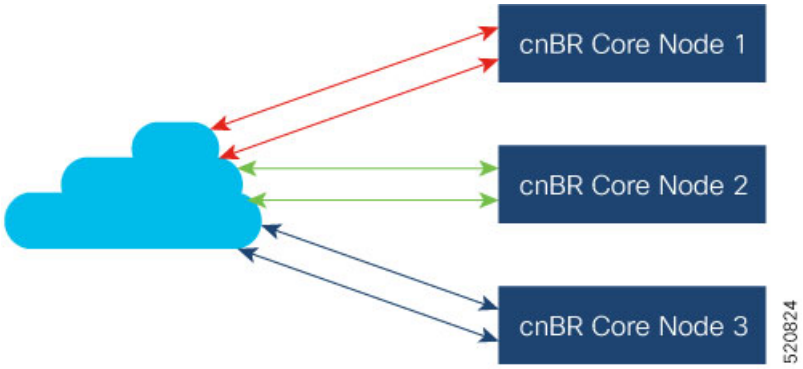


Cisco cnBR Service Resiliency

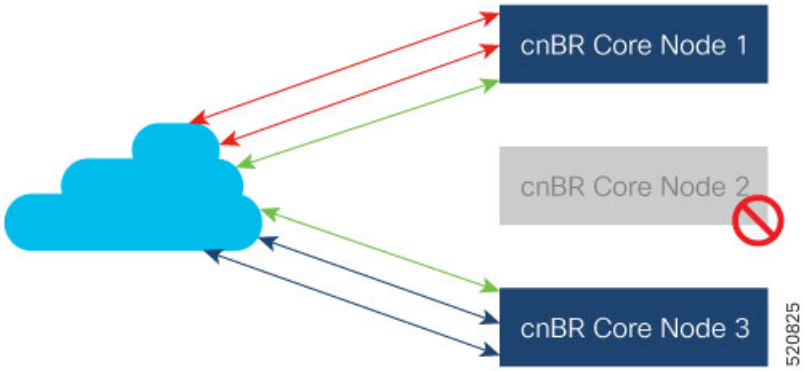
The Cisco cnBR supports service resiliency that tolerates software and hardware failures. It can dynamically balance DOCSIS service workloads among the micro service instances and DOCSIS nodes in the Cisco cnBR cluster. When a single micro service instance or node fails, to minimize service interruption, the system reassigns the affected workloads to suitable resources automatically.

Node Failure Recovery

In Cisco cnBR, all micro service instances, which provide DOCSIS services, are organized into a global resource pool. The system manages this resource pool and assigns workloads to micro service instances. When you add a new RPD into the cluster, the system chooses a proper node and assigns the newly increased workloads to the micro service instances running on the chosen node. In the following example, the system assigns the workloads of multiple RPDs to multiple nodes evenly.



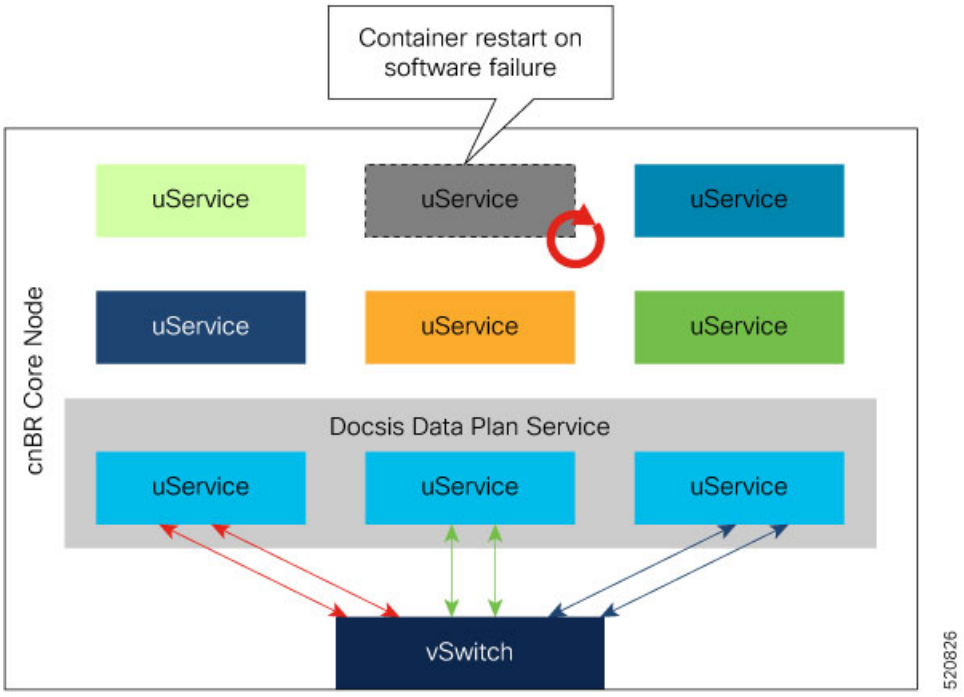
When a node fails, the system moves the workloads from the failed node to healthy nodes that have sufficient capacity to accept more workloads.



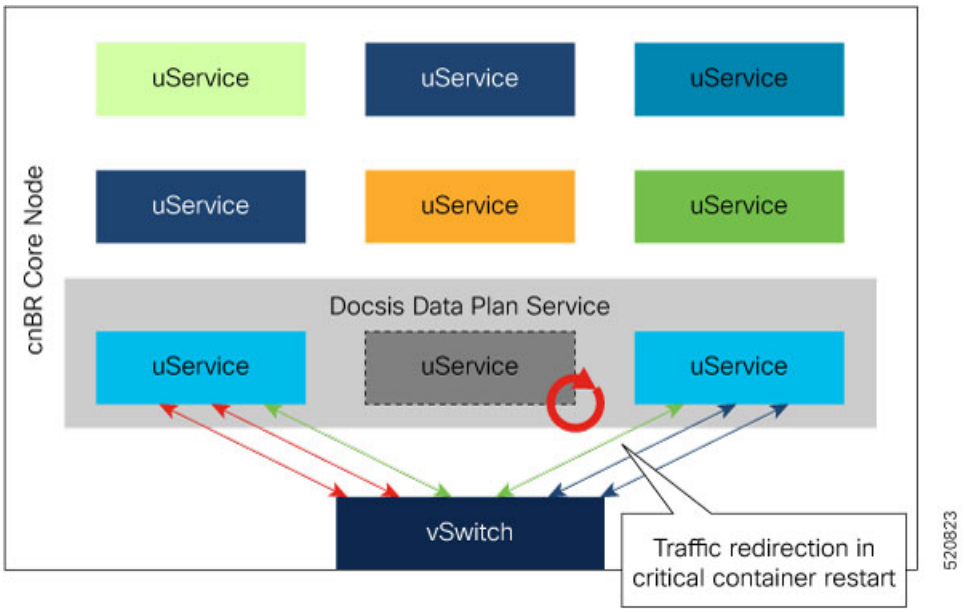
Therefore, the healthy nodes in the cluster take over the workloads from the failed node. After the failed node recovers, it returns to the resource pool and the system can assign new workloads to it. If the available capacity on the healthy node is not enough, the system moves as many workloads as possible until all resources are exhausted. The remaining workloads stay on the failed node; they are recovered after the node is recovered.

Software Failure Recovery

In addition to node resiliency, the containerized micro services are inherently tolerant to service software failures. If a micro service instance fails, it can restart itself quickly without interrupting the overall service.



Container restart may take a few seconds; it is good enough for control plane and management services. When a container in critical services such as data plane fails to minimize the traffic interruption time, the system redirects DOCSIS traffic to other instances with free service group capacity within the same node.



Configure Service Resiliency

Service resiliency is always enabled in Cisco cnBR cluster.

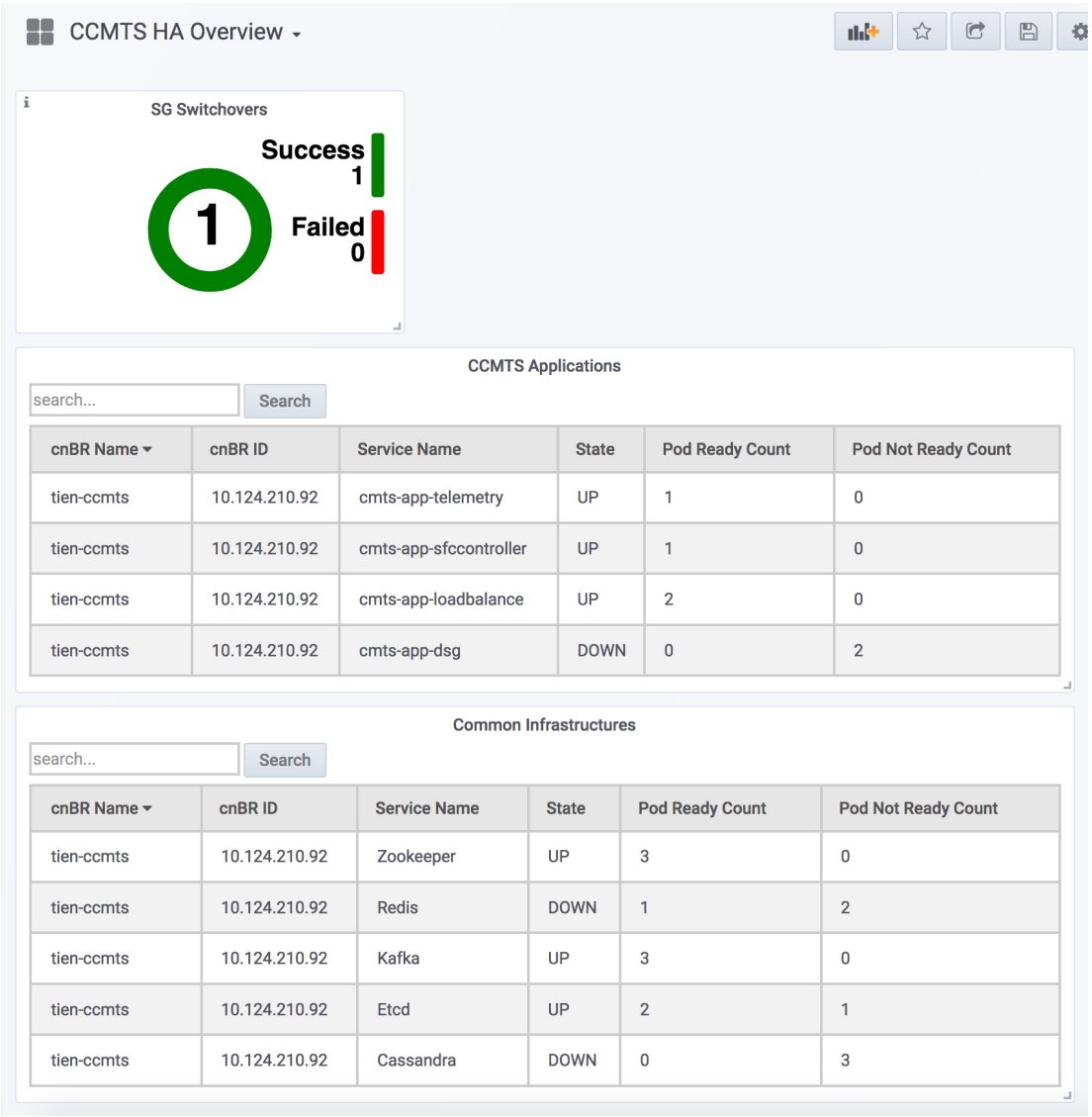
The system constantly monitors the resource (nodes and service instances) status. When there is a failure, the system automatically triggers workload reassignment. This process is transparent to the subscribers.

Workload in Cisco cnBR is measured in the unit of service group. Service groups are load balanced across DOCSIS nodes when you add them into a Cisco cnBR cluster. Make sure that there are enough capacities reserved in a Cisco cnBR cluster for resiliency.

In 20.2 release, each DOCSIS node can support up to 20 service groups. In order to tolerate one node failure without service interruption, we recommend that you do not provision more than 40 service groups for a three DOCSIS node Cisco cnBR cluster. Then, when a single DOCSIS node fails, there are enough capacities reserved for service resiliency.

Monitor and Troubleshoot

In Cisco cnBR HA Overview dashboard, you can check the overall High Availability (HA) state of the cluster.

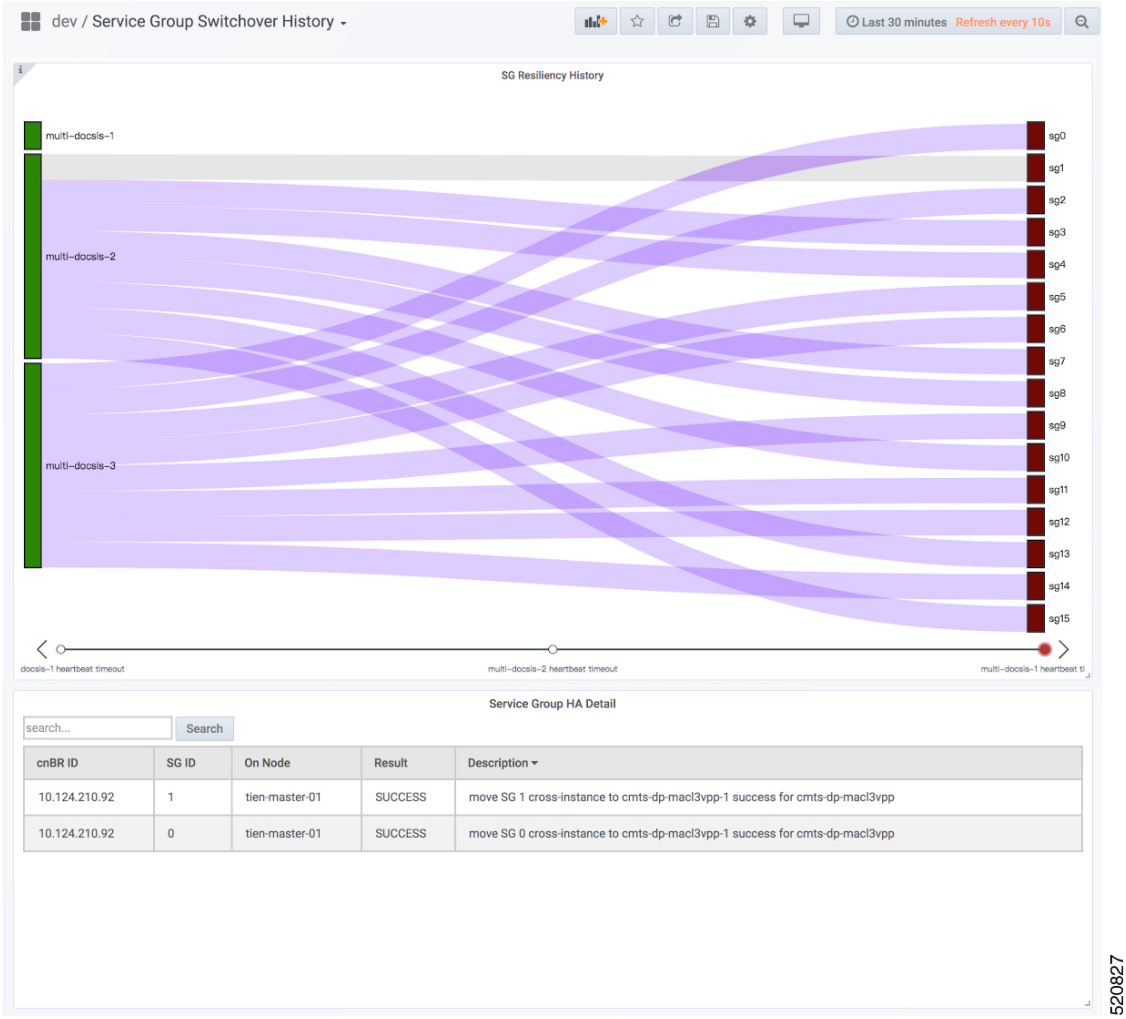


The SG Switchovers chart displays the total DOCSIS service switchover event count in the Cisco cnBR cluster. The counters increase when new service switchover occurs. In this chart:

- Success: The service switchover is complete without any issues.
- Failed: Some or all of the services failed to move workload during the service switchover. If this counter increases, click the number to check the error in the Service Group Switchover History dashboard.

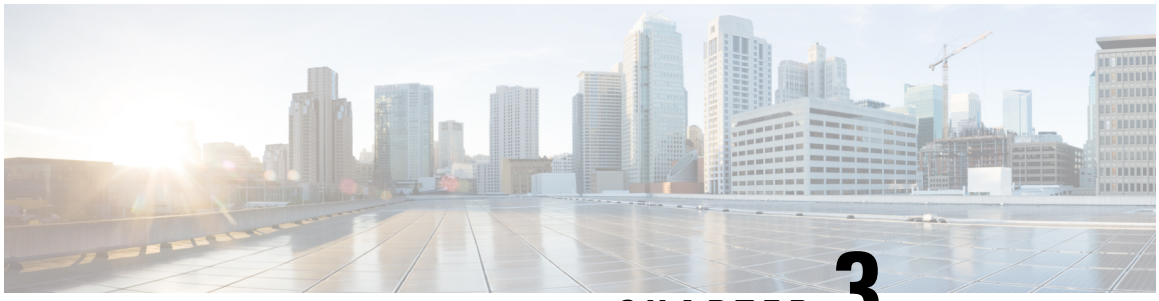
cnBR Applications table lists the HA state of all the Cisco cnBR application services.

If a new switchover event occurred, access the Service Group Switchover History dashboard to review detailed information for troubleshooting.



The SG Resiliency History diagram visualizes all historical DOCSIS service switchovers and SG mapping changes.

Click an event in the timeline to display the event details in the Service Group HA Detail panel.



CHAPTER 3

Cisco Cloud Native Broadband Router Service Configuration and Monitoring

Cisco cnBR virtualizes all of the hardware-based services, providing a cloud-native design, and offers a variety of features as microservices. You can quickly develop, test, and deploy new services or update features and functions without any downtime.

Network Services

Cisco cnBR empowers you to create a number of easily composable, scalable, and resilient network services.

DHCP Relay Service

Cisco cnBR acts as a Dynamic Host Configuration Protocol (DHCP) relay agent to implement features such as DHCP relay, Lease Query (LQ), IPv6 Prefix Delegation (PD), and to provision static IP addresses for subscribers by using source address verification (SAV).

DHCP Relay

When the Cisco cnBR acts as a relay agent, it forwards requests and replies between clients and servers when they are not on the same physical subnet. Relay agent forwarding is different with the normal IP router forwarding. In the normal IP router forwarding, IP datagrams are forwarded between networks transparently. But in relay agent forwarding, relay agent receives DHCP message and then generates a new DHCP message to send through another interface.

When a DHCP client requests an IP address from a DHCP server, for instance DHCPv4, the client sends a DHCPDISCOVER broadcast message to locate the DHCP server. Relay agent forwards the packets between DHCP client and DHCP server. DHCP server provides configuration parameters, such as IP address, MAC address, domain name, and a lease for the IP address, to the client in a DHCPOFFER unicast message.

User Guidelines:

- By default, DHCP relay is enabled on Cisco cnBR. DHCP relay depends on two Cisco cnBR services in the multiple instances environment - BGP agent and Relay proxy.
- DHCP relay agent configuration is based on service group.
- DHCP server receives DHCP request. If multiple DHCP servers are configured, all these servers receive relay packets.
- The v4Net/v6Net defines all the IP scopes for the subscriber's DHCP destination IP address. This configuration must be consistent with the configuration of the DHCP server. If multiple subscriber nets are configured, use the first scope as the default scope.
- Cisco cnBR can also assign a specific server or IP scope for a subscriber. For more information, see Policy Based Relay, on page 70.

Policy Based Relay

Policy Based Relay allows subscribers with different device classes to be classified into different IP ranges.

When the relay agent handles subscriber DHCP packets, Cisco cnBR can identify its device class based on the TLV in the DHCP packets. Then the Cisco cnBR uses a predefined relay policy to assign a specific server to get DHCP address, or notify the server to assign its DHCP address in a specific IP range.

User Guidelines:

- Define the v4serverip/v6serverip in the dhcpServers.
- Define the giaddr/linkaddr with associated v4Nets and/or v6Nets. The address is the prefix of the v4Nets/v6Nets.
- If there is no specific v4serverip/v6serverip for the device class, the subscriber requests are forwarded to all the servers defined.
- If there is no specific giaddr/linkaddr for the device class, the subscribers get the IP from the first default range.

DHCPv6 Prefix Delegation

In the IPv6 networking, you can use DHCPv6 prefix delegation (PD) to assign network address prefix, automate configuration, and provision of the public routable addresses for the network. For example, in home networks, home routers use DHCPv6 protocol to request a network prefix from the ISP's DHCPv6 server. After you assign the network prefix, the ISP routes this network prefix to your home router. Then the home router starts displaying the new addresses to hosts on the network.

Once the PD router comes online, it gets the assigned network prefix from the DHCP server.

ARP/NDP Glean and Lease Query

As a relay agent, Cisco cnBR stores all the subscriber DHCP information after DHCP is completed. Based on this information, routing is established for subscribers. However, there are several cases when subscriber information is unavailable, such as a modem reset, resulting in routing being no longer available for subscribers. When these subscribers access the network, Cisco cnBR rebuilds the data path by using ARP/NDP glean or lease query.

When using ARP/NDP Glean, Cisco cnBR can trust the packets that come from the cable side network. Once the ARP/NS is received and the source IP is updated in the configured IP ranges, Cisco cnBR rebuilds a data path for the source MAC. This method is open to MAC spoofing.

In contrast, when using Lease Query, Cisco cnBR doesn't trust the cable side network. When Cisco cnBR receives the upstream packet with no data path route, it sends a LEASEQUERY request to DHCP server. Once DHCP server gets the request and confirms that the RESPONSE, the MAC and IP are released from DHCP server, Cisco cnBR rebuilds the data path. Otherwise, Cisco cnBR drops the packets.

User guidelines:

- Enable or disable ARP/NDP Glean and Lease Query on demand.
- Lease Query checks the source IP with the v4Nets/v6Nets configuration. If the source IP of the packets isn't in the range, then Lease Query discards the packet.
- Use ARP/NDP Glean and Lease Query with Source Address Verification (SAV).

SAV

In addition to DHCP leased IP address, Cisco cnBR allows static IP address by provisioning SAV group.

A SAV group is a group of IPv4 or IPv6 prefixes. Cisco cnBR uses these prefixes to authenticate a cable modem (CM). You can configure a CM with an IPv4 or IPv6 prefix that belongs to a particular SAV group. The time, length, value (TLV) 43.7.1 specifies the group name to which a given CM belongs. If the source IP address of a packet from a CM belongs to the configured prefix in a SAV group, the Cisco CMTS considers it as an authorized packet.

You can configure a maximum of 255 SAV groups on a Cisco cnBR. Each SAV group contains up to four IPv4s, IPv6s, or a combination of both prefixes. The total number of the prefixes is no more than four.

During registration, CMs communicate their configured static prefixes to the CMTS using TLV 43.7.1 and TLV 43.7.2. The TLV 43.7.1 specifies the SAV prefix group name that the CM belongs to, and TLV 43.7.2 specifies the actual IPv4 or IPv6 prefix. Each CM can have a maximum of four prefixes configured. When the Cisco CMTS receives these TLVs, it identifies whether the specified SAV group and the prefixes are already configured on the Cisco CMTS at first. If configured, the Cisco CMTS associates them to the registering CM. However if not configured, the Cisco CMTS automatically creates the specified SAV group and prefixes before associating them to the registering CM.

The Cisco CMTS considers the SAV group name and the prefixes that are provided by these TLVs valid. The packets received from the CM with the source IP address belonging to the prefix specified by the TLV are authorized packets. For example, if a given CM has an SAV prefix of 10.10.10.0/24, and the source IP address of a packet received from this CM (or CPE behind the CM) is in the subnet 10.10.10.0/24, then it's an authorized packet.

User guidelines:

- SAV configuration is global and not per service group.
- SAV doesn’t check the MAC/IP binding. You can assign the static IP to any MAC.
- By default, SAV is disabled. You can enable it on demand.

ARP/NDP Proxy

All cable modems and subscribers are behind the HFC network. As a proxy, Cisco cnBR relays the ARP/NDP requests to the CM.

With ARP/NDP proxy enabled, Cisco cnBR can respond the ARP/NDP, and DS lease query is not supposed to be triggered.

Mobility Scopes

If the subscribers are allowed to roam between different IPv4 and IPv6 scopes, the mobility scopes contain all the IPv4 and IPv6 scopes granted to the subscribers. This configuration is optional.

Configure DHCP Relay Service

The DHCP relay service operates in a similar way as other Cisco CMTS products. You can configure it with Day 1 Deploy Script, or by importing the whole Cisco cnBR configuration yaml file to the desired Cisco cnBR using Cisco Operations Hub. The imported configuration file overwrites the existing configuration and activates the new configuration.

Update the DHCP Relay configuration using Autodeployer reconfig (Preferred)

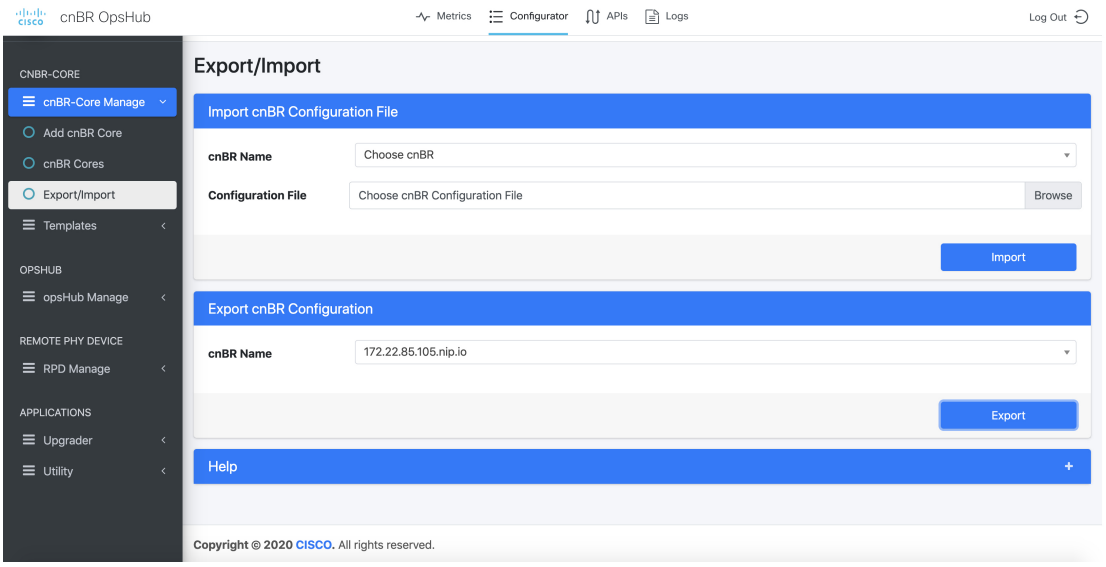
After the initial DHCP Relay configuration using Autodeployer during the deployment, you can modify the dhcp block in the L3 profile file and run the AutoDeployer configuration script again to update the configuration.

Note Rerun AutoDeployer configuration script causes all the RPDs/SGs to be deleted and added.

Update the DHCP Relay configuration using Cisco Operations Hub

After the initial DHCP Relay configuration using Autodeployer during the deployment, you can also update the configuration using the Cisco Operations Hub Configurator panel.

Figure 14: Operations Hub Configuration Export/Import



Use the following steps to update the DHCP Relay configuration:

Step 1 Select the Configurator panel, then click **Export/Import** under the cnBR-Core Manage tab to open the Export/Import page.

Step 2 In the Export cnBR Configuration section, select the Cisco cnBR to update from the drop-down list.

Step 3 Click the **Export** button to get the current SG configuration of the selected Cisco cnBR.

Step 4 Update one or more parameters in the `dhcp` section of the SG configuration.

Step 5 Save the updated configuration file on the local disk.

Step 6 In the Import cnBR Configuration File section, select the Cisco cnBR to update from the drop-down list.

Step 7 Click the **Browse** button to locate the file saved in step 5.

Step 8 Click the **Import** button to upload the updated SG configuration to the selected Cisco cnBR.

Configure DHCP Relay using Autodeployer Script

In the AutoDeployer script L3 profile file, the DHCP Relay configuration is saved in the `dhcp` section. It will be applied to all Service Groups on the Cisco cnBR. Following is an example configuration:

```
"Dhcp":
    {
        "ArpGlean":true,
        "ArpProxy":true,
        "ipv4Lq": false,
        "NdGlean":true,
        "NdProxy":true,
        "ipv6Lq":false,
        "dhcpServers":["80.80.80.3",
                       "81.81.81.3",
                       "2001:80:80:80::3",
                       "2001:81:81:81::3"
        ],
        "V4Nets":["90.90.90.1/24",
                  "91.91.91.1/24",
                  "92.92.92.1/24"
        ],
        "V6Nets":["2001:90:90:90::1/64",
                  "2001:91:91:91::1/64",
                  "2001:92:92:92::1/64"
        ],
        "RelayPolicies":[
          {"deviceClass": "HOST",
          "v4serverip": "80.80.80.3",
          "v6serverip": "2001:80:80:80::3",
          "giaddr": "90.90.90.1",
          "linkaddr": "2001:90:90:90::1"
          },
          {"deviceClass": "STB",
          "v4serverip": "81.81.81.3",
          "v6serverip": "2001:81:81:81::3",
          "giaddr": "91.91.91.1",
          "linkaddr": "2001:91:91:91::1"
          },
          {"deviceClass": "PS",
          "giaddr": "92.92.92.1",
          "linkaddr": "2001:92:92:92::1"
          },
          {"deviceClass": "EROUTER",
          "v4serverip": "80.80.80.3",
          "v6serverip": "2001:80:80:80::3",
```

```
        },
        {"deviceClass": "DVA",
        "giaddr": "90.90.90.1",
        "linkaddr": "2001:90:90:90::1"
        },
        {"deviceClass": "MTA",
        "giaddr": "91.91.91.1",
        "linkaddr": "2001:91:91:91::1"
        }
    ],
    "mobilityScopes":["90.90.90.1/24",
                      "91.91.91.1/24",
                      "92.92.92.1/24",
                      "2001:90:90:90::1/64",
                      "2001:91:91:91::1/64",
                      "2001:92:92:92::1/64"
    ]
}
```

See Configure cnBR using Autodeployer, on page 32 for additional information.

Configure DHCP Relay

Field Name	Description	Type	Enforcement
dhcpServers	DHCP server IPv4 and IPv6 addresses	IPv4 or IPv6	Required
v4Nets	IPv4 range to which the subscriber's DHCP address belongs	CIDR (Classless Inter-Domain Routing)	Required
v6Nets	IPv6 range to which the subscriber's DHCP address belongs	CIDR (Classless Inter-Domain Routing)	Required

```
"Dhcp":
{
    // all the DHCP servers IP, V4 and V6
    "dhcpServers":[
                    "81.81.81.3",
                    "24.24.24.3",
                    "2001:81:81:81::3",
                    "2001:24:24:24::3"
                  ],
    // all the V4 subnets for the subscribers in this SG
    "v4Nets":[
                "90.90.90.1/24",
                "91.91.91.1/24",
                "92.92.92.1/24",
                "93.93.93.1/24",
                "94.94.94.1/24",
                "95.95.95.1/24"
                "96.96.96.1/24",
                "97.97.97.1/24",
            ],
    // all the V6 subnets for the subscribers in this SG
    "v6Nets":[
                "2001:90:90:90::1/64",
                "2001:91:91:91::1/64",
                "2001:92:92:92::1/64",
                "2001:93:93:93::1/64",
                "2001:94:94:94::1/64",
                "2001:95:95:95::1/64",
                "2001:96:96:96::1/64",
```

```
                "2001:97:97:97::1/64"
            ],
    }
```

Configure DHCP Relay Policy

Field Name	Description	Type	Enforcement
deviceClass	The device class for each subscriber	String	Required
v4serverip	The server to which the DHCP request is forwarded	IPv4	Optional
v6serverip	The server to which the DHCPv6 request is forwarded	IPv6	Optional
giaddr	The IP range to which the DHCPv4 address belongs, the giaddr is the IP address in the v4Nets	IPv4	Optional
linkaddr	The IP range to which the DHCPv6 address belongs, the linkaddr is the IP address in the v6Nets	IPv6	Optional

```
 "Dhcp":
 {
  "RelayPolicies":[
{"deviceClass": "HOST",
 "giaddr": "92.92.92.1",
 "v4serverip": "24.24.24.3",
 "linkaddr": "2001:92:92:92::1"
},
{"deviceClass": "STB",
 "giaddr": "93.93.93.1",
 "v4serverip": "81.81.81.3",
 "linkaddr": "2001:93:93:93::1"
},
{"deviceClass": "PS",
 "giaddr": "94.94.94.1",
 "v6serverip": "2001:81:81:81::3",
 "linkaddr": "2001:94:94:94::1"
},
{"deviceClass": "EROUTER",
 "giaddr": "95.95.95.1",
 "linkaddr": "2001:95:95:95::1"
},
{"deviceClass": "DVA",
 "giaddr": "96.96.96.1",
 "v4serverip": "24.24.24.3",
 "linkaddr": "2001:96:96:96::1"
},
{"deviceClass": "MTA",
 "giaddr": "97.97.97.1",
 "v6serverip": "2001:24:24:24::3",
 "linkaddr": "2001:97:97:97::1"
}]
 }
```

Configure ARP/NDP Glean and Lease Query

Field Name	Description	Type	Enforcement
arpGlean	Enable/Disable	Boolean	Required, default is false

Field Name	Description	Type	Enforcement
ndGlean	Enable/Disable	Boolean	Required, default is false
ipv4Lq	Enable/Disable	Boolean	Required, default is false
ipv6Lq	Enable/Disable	Boolean	Required, default is false

```
"Dhcp":
{
    "arpGlean":true,
    "ipv4Lq": false,
    "ndGlean":false,
    "ipv6Lq": false,
}
```

Configure SAV

Field Name	Description	Type	Enforcement
savEnable	Enable/Disable	Boolean	Required
savEntires	SAV group structure	savGroup	Optional
grpName	SAV group name	String	Optional
prefixes	The SAV prefixes	CIDR (Classless Inter-Domain Routing) list	Optional

```
"sav"
{
    "savEnable": true,
    "savEntries": [{
                    "grpName": "testSAVV",
                    "prefixes": ["93.93.93.100/28",
                                 "2001:93:93:93100::0/72"]

                  }]
}
```

Configure ARP/NDP Proxy

Field Name	Description	Type	Enforcement
ArpProxy	Enable/Disable	Boolean	Required, default false
NdProxy	Enable/Disable	Boolean	Required, default false

```
"ArpProxy":true,
"NdProxy":true,
```

Configure Mobility Scopes

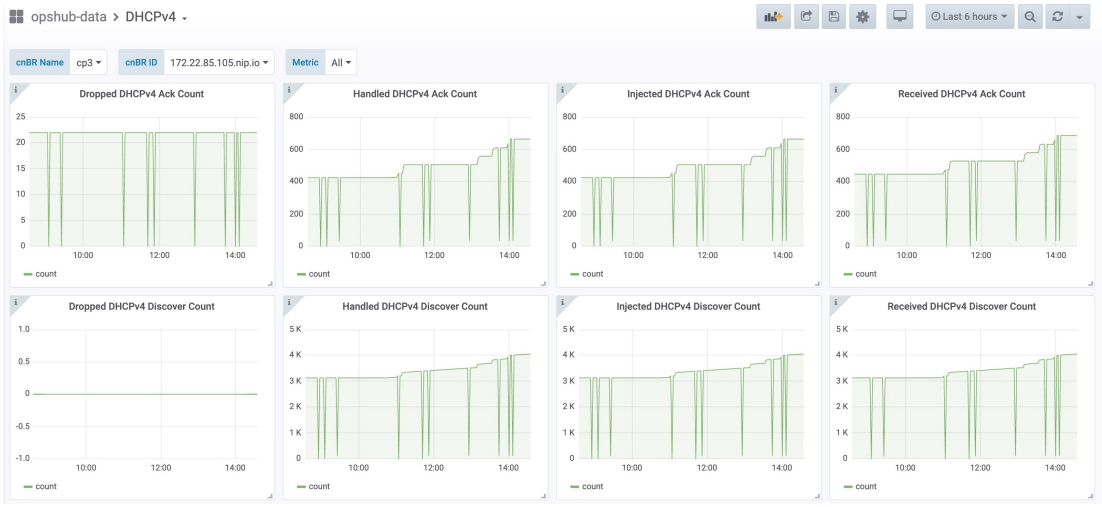
Field Name	Description	Type	Enforcement
mobilityScopes	Scopes of ipv4 and ipv6	String	Optional

```
"mobilityScopes":["90.90.90.1/24",
                  "91.91.91.1/24",
                  "92.92.92.1/24",
                  "2001:90:90:90::1/64",
                  "2001:91:91:91::1/64",
                  "2001:92:92:92::1/64"
                 ]
```

Monitor DHCP Relay Service

DHCP IPv4 Statistics

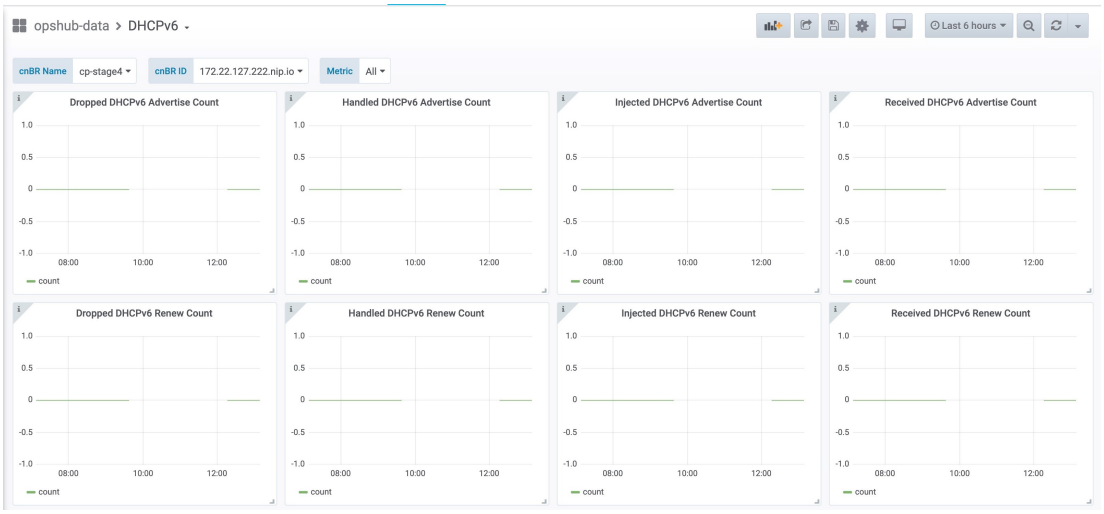
Figure 15: DHCPv4 panel in Cisco Operations Hub Metrics



This dashboard panel (DHCPv4) in Cisco Operations Hub Metrics is displaying statistics of the DHCP relay of IPv4. There are 16 dashboards in total. The preceding picture shows only half of the dashboards. Each dashboard represents the count of different states for different packet over time. There are four packet types for DHCPv4: Discover, Offer, Request, and Acknowledgment (Ack). The system processes each type of packet differently: Received, Dropped, Handled, and Injected. You can change the time span at the top-right corner. Currently they show the count in the last 6 hours.

DHCP IPv6 Statistics

Figure 16: Screenshot of DHCPv6 panel in Cisco Operations Hub Metrics



This dashboard panel (DHCPv6) in Cisco Operations Hub Metrics is displaying statistics of the DHCP relay of IPv6. There are 16 dashboards in total. The preceding picture shows only half of the dashboards. Each dashboard represents the count of different states for different packet over time. There are four packet types for DHCPv4: Renew, Advertise, Request, and Reply. The system processes each type of packet differently: Received, Dropped, Handled, and Injected. You can change the time span at the top-right corner. Currently they show the count in the last 6 hours.

PTP

Precision Time Protocol (PTP) is used to synchronize clocks throughout all cable networks. The Cisco cnBR cores and RPDs are managed by the Cisco cnBR, and runs an instance of the PTP client. To achieve time synchronization, the PTP client in Cisco cnBR and the PTP client in RPDs must synchronize their clocks to the same PTP primary clock. The Cable Modems (CMs) then synchronize their clock to the Cisco cnBR (and eventually to the PTP primary clock) through the DOCSIS timestamps provided by the RPD.

PTP allows creation of individual profiles for different scenarios. A profile is a specific selection of PTP configuration options that are selected to meet the requirements of a particular application. Cisco cnBR supports the PTP default profile.

To provide a high availability precision clock in the Cisco cnBR, two PTP primary clock sources can be configured in cnBR - a main PTP primary clock server and an alternate PTP primary clock server. Cisco cnBR synchronizes its clock to the best available PTP primary clock.

Some of the key parameters that are configured, or configurable, in the Cisco cnBR and RPD PTP client include:

- PTP Domain

  A PTP domain is a logical grouping of clocks that communicate with each other using the PTP protocol. A single computer network can have multiple PTP domains operating separately. For example, one set of clocks synchronized to one time scale and another set of clocks synchronized to another time scale. PTP can run over either Ethernet or IP, so a domain can correspond to a Local Area Network, or it can extend across a Wide Area Network.

In Cisco cnBR and RPD PTP client, the PTP domain is set during initial Cisco cnBR deployment. The PTP domain can be updated after deployment.

- PTP Transport

  In Cisco cnBR and RPD, the PTP transport is configured to use PTP over IPv4 in unicast mode. The PTP Transport mode is not configurable in Cisco cnBR PTP client. The PTP Transport mode is configurable in the RPD PTP client.

- PTP Ports

  A port can be configured to perform either fixed primary or secondary role, or can be configured to change its role dynamically. If no role is assigned to a port, it can dynamically assume a primary, passive, or secondary role, based on the Best Primary Clock Algorithm (BPCA), which is also known as Best Master Clock Algorithm (BMCA [RFCÂ 7273]).

  Cisco cnBR and RPD support the PTP port secondary role. The Cisco cnBR PTP port role is not configurable. However, the RPD PTP port role is configurable, but it must be set to secondary role.

- PTP Clock Mode

  PTP Clock Mode can be configured as either of the following modes:

  - **1-step clock mode**: The PTP primary clock includes its timestamp in the synchronization message when the synchronization message is sent by the hardware. This mode requires hardware to insert the clock timestamp right before the synchronization message is sent through the wire.
  - **2-step clock mode**: The PTP primary clock sends its timestamp in a separate message after sending the synchronization message. This mode does not require hardware support, but the timestamp messages and the synchronization messages may arrive at the PTP clients out of order in some scenarios.

  Cisco cnBR and RPD support the 1-step clock mode. The PTP Clock mode is not configurable.

Configure PTP

The PTP client in Cisco cnBR and RPD can be configured during the initial Cisco cnBR configuration using Autodeployer.

Step 1 The top-level Autodeployer configuration file used in the deployment of Cisco cnBR must include the configuration for the PTP client in the Cisco cnBR.

Table 7:

Field Name	Description	Mandatory
ptp:v4:	PTP IPv4 related parameters for the Cisco cnBR PTP container	Yes
domain	Clock domain of the PTP primary server	Yes
master:ip	IPv4 address of the PTP clock primary server	Yes

Field Name	Description	Mandatory
master:gw	IPv4 address of the Gateway to access the PTP clock primary server	Yes
alt-master:ip	IPv4 address of the PTP alternate clock primary server	No
alt-master:gw	IPv4 address of the gateway to access the PTP alternate clock primary server	No
SG_template	Go through the SG template listed in step Step 2, on page 80	Yes

Step 2 The reference Service Group template should include the configuration of the PTP client in the RPD. Go through the following table for the detailed values.

Table 8:

Field Name	Description	Mandatory
rpdPtpCfg:	< PTP related parameters for the PTP client in the RPD >	Yes
domain	Clock domain of the PTP primary server	Yes
dtiMode	DOCSIS Time Interface Mode	Yes
priority1	Priority1	No
priority2	Priority2	No
ptpClkProfileId	PTP clock profile ID in PTP primary server	Yes
ptpPortCfg: adminState	PTP port adminstration state	Yes
ptpPortCfg: anncReceiptTimeout	Annoucement Receipt Timeout interval	No
ptpPortCfg: cos	COS of 802.1Q	No
ptpPortCfg: dscp	DSCP of IP Differentiated Services	No
ptpPortCfg: enetPortIndex	Ethernet port index for the clock port	No
ptpPortCfg: gateway	IPv4 address of the gateway to access the PTP primary clock server	Yes
ptpPortCfg: masterAddr	IPv4 address of the PTP primary clock server	Yes
ptpPortCfg: localPriority	Local Priority	No

Field Name	Description	Mandatory
ptpPortCfg: logDelayReqInterval	Interval for PTP delay-req packets0-7(-7 -0)	Yes
ptpPortCfg: logSyncInterval	Interval for Sync packets	Yes
ptpPortCfg: masterAdminState	PTP Primary Administration State	Yes
ptpPortCfg: ptpPortIndex	PTP Port Index	Yes
ptpPortCfg: unicastDuration	The grant duration time in seconds for unicast	No

For more information on the listed parameters, go through the RPD documentation at https://www.cisco.com/c/en/us/td/docs/cable/cbr/configuration/guide/b-rpd-full-book-11/b-rpd-full-book-11_chapter_011.pdf.

Example

- Cisco cnBR PTP client-related parameters in Autodeployer top-level configuration file:

```
// IPv4 address of PTP Master Clock and alternate Master clock servers,
// and their respective Gateway server, in the top level config file.
ptp :
    v4 :
       domain : 0
       master: {'ip':"100.158.158.158", 'gw':"10.70.78.1"}
       alt-master: {'ip':"100.158.158.159", 'gw':"10.70.78.1"}

// Specify the "SG template" that contains the RPD PTP CLient parameters.
    SG :
        'SG_4x4': 'sg_template.json'
```

- RPD PTP client-related parameters in the SG_template:

```
"rpdPtpCfg": {
    "dtiMode": "SlaveDtiMode",
    "domain": 44,
    "priority1": 128,
    "priority2": 255,
    "ptpClkProfileId": "00:00:00:00:00:00",
    "ptpPortCfg": [
      {
        "adminState": "Up",
        "anncReceiptTimeout": 11,
        "cos": 6,
        "dscp": 47,
        "enetPortIndex": 1,
        "gateway": "10.70.78.1",
        "localPriority": 128,
        "logDelayReqInterval": -4,
        "logSyncInterval": -4,
        "masterAddr": "100.158.158.158",
        "masterAdminState": "Up",
        "ptpPortIndex": 22,
        "unicastDuration": 300
      }
```

```
    ]
}
```

Update cnBR PTP Configuration using Autodeployer

You can update the Cisco cnBR PTP configuration using the Autodeployer.

Ensure that you have configured the Cisco cnBR PTP client during deployment, and the Cisco cnBR using the Autodeployer.

See Configure cnBR using Autodeployer, on page 32 for more information.

Go through the following steps to update the PTP configuration:

Step 1 Locate the Autodeplyer configuration files used for the initial deployment and configuration of cnBR. This includes:

- Top-level Autodeployer configuration file
- SG template
- L3 template

Step 2 Update the `PTP` section of the top-level Autodeployer configuration file.

Step 3 Run the Autodeployer configuration script.

Note All RPDs or SGs (including unchanged SGs), are first deleted and added when you rerun the Autodeployer configuration.

Update cnBR PTP Configuration using Operations Hub

You can update the Cisco cnBR PTP configuration using the Cisco Operations Hub.

Ensure that you have configured the Cisco cnBR PTP client during deployment, and the Cisco cnBR using the Autodeployer. Also ensure that the Cisco cnBR is added to the Cisco Operations Hub.

To view and update the PTP configuration parameters, complete the following steps:

Step 1 Click **Configurator** > **Cores** > **Core Clusters** and select one Cisco cnBR core.

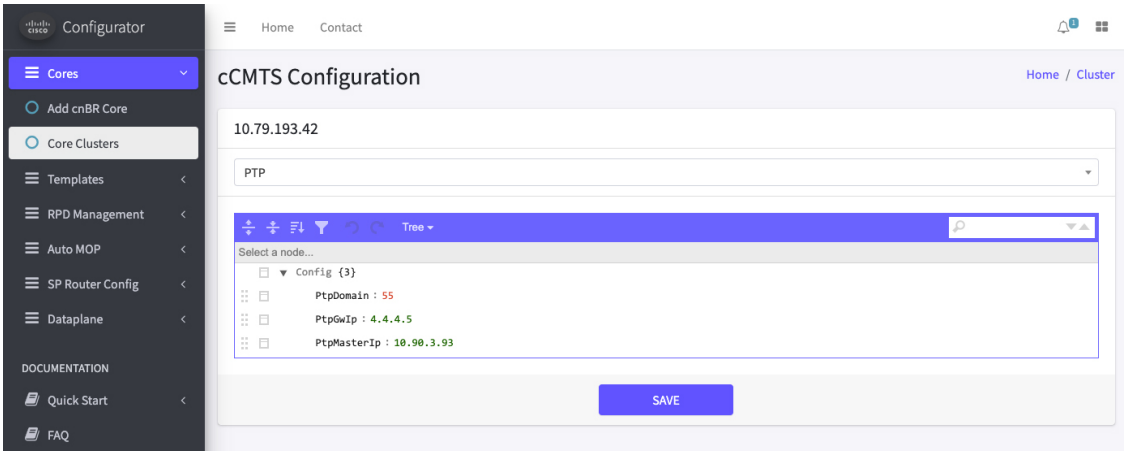
Step 2 Choose to edit the PTP configuration using one of the two modes below:

- Tree mode: Select **Tree** mode to edit each field.
- Code mode: Select **Code** mode to edit the configuration in plaintext.

Step 3 Choose to configure the Cisco cnBR PTP client with either a single primary clock or with dual primary clocks.

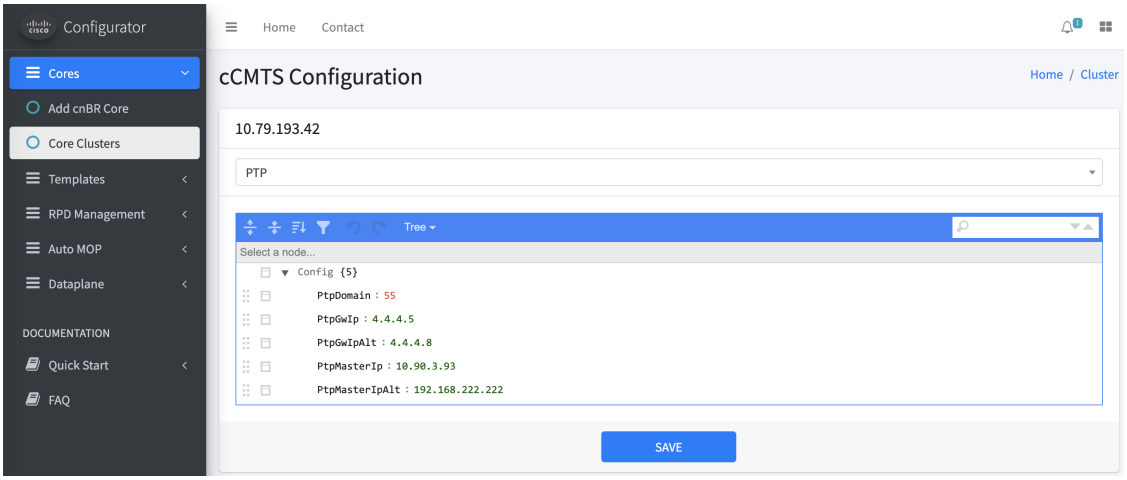
The following image shows the Cisco cnBR PTP client with a single primary clock.

Figure 17: Configuring cnBR PTP client with a single primary clock



The following image shows the Cisco cnBR PTP client with dual primary clock.

Figure 18: Configuring cnBR PTP client with a dual primary clock



Update RPD PTP Configuration using Autodeployer

You can update the RPD PTP configuration using the Autodeployer. We recommend this method of updating the RPD PTP.

Ensure that you have configured the RPD PTP client during the deployment, and have configured Cisco cnBR using the Autodeployer.

See Configure cnBR using Autodeployer, on page 32 for more information.

Step 1 Locate the complete set of Autodeplyer configuration files used in the initial deployment and configuration of cnBR. This includes:

- Top-level Autodeployer configuration file

- SG template
- L3 template

Step 2 Update the `rpdPtpCfg` section of the Service Group template.

Step 3 Run the Autodeployer configuration script.

Note Rerunning the Autodeployer configuration causes all the RPDs or SGs, including unchanged SGs, to be first deleted and added.

Update RPD PTP Configuration using Operations Hub

You can update the RPD PTP configuration using the Cisco Operations Hub.

Ensure that you have configured the RPD PTP client during deployment, and have configured Cisco cnBR using the Autodeployer.

To view and update the RPD PTP configuration parameters, complete the following steps:

Step 1 Click **Configurator** > **Export/Import**.

Step 2 Select the Cisco cnBR to update in the `Export cnBR Configuration File` pane.

Step 3 Click **Export** to retrieve the current SG configuration of the selected Cisco cnBR.

Step 4 In the <*filename*>`-configuration.txt` file, update the parameters in the `rpdPtpCfg` section of the SG configuration.

Step 5 Save the updated file to the local disk.

Step 6 Update the SG configuration.

a) In `Import cnBR Configuration File` pane, select the file that was updated through the previous step.
b) Click **Import** to update the SG configuration to the RPD.

Step 7 Delete the RPD and add the RPD again for the updated SG configuration to take effect.

Monitor and Troubleshoot PTP

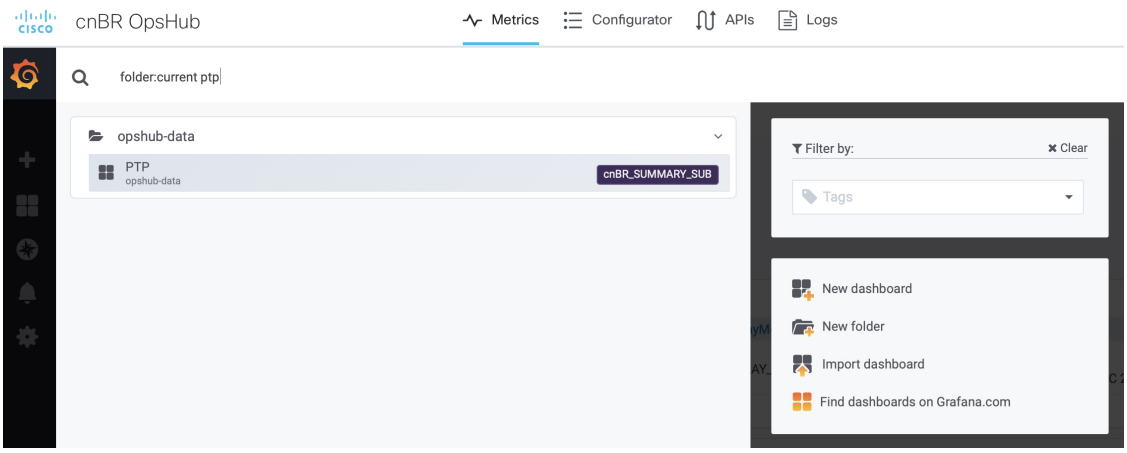
You can view the PTP status and information on the PTP panel of the Health Monitor dashboard.

To view the Health Monitor dashboard, complete the following steps:

Step 1 Enter Cisco Operations Hub URL `https://{opshub-ip}.nip.io` in the web browser.

Step 2 Click **Metrics** and search for `PTP`.

Figure 19: PTP Dashboard



The PTP dashboard appears.

Note The `OffsetFromMaster` must be within [-1ms, 1ms].

BGP Agent

The BGP Agent is a service in Cisco cnBR. It sets up BGP sessions with SP router and installs or withdraws subscribed routes on SP router while the subscribed devices (e.g. CM/CPE) are online.

The Cisco cnBR BGP Agent supports BGP version 4, includes address family IPv4 unicast, address family IPv6 unicast, and Graceful Restart, on page 87.

Configure BGP Agent

You can perform BGP Agent initial configurations through the Autodeployer Config file. See Configure cnBR using Autodeployer, on page 32 for additional information.

After initial setup, you can access BGP Agent configuration through the Cisco Operations Hub BGP Agent Configurator. See instructions for Access BGP Agent Configurator, on page 87.

Configuration Parameters

Field Name	Description	Type	Enforcement
asNumber	BGP supports 2-byte AS numbers	1 ~ 65535	Required
ebgpMultihop	The maximum number of eBGP hops allowed	0 ~ 255	Required
ifname	BGP Agent interface name	String, length 1 ~ 255	Required
neighbors	BGP peer, BGP uses TCP port 179 to create a TCP session with a peer		Required
weight	Weight of BGP peers, if you configure two BGP IPv4/IPv6 peers, the upstream routes sent from these peers are accepted in the order of weight. Default: 100	Unsigned integer	Optional
address	BGP peer IP/IPv6 address	String	Required
gateway	The gateway IP address if the BGP messages are transmitted to loopback interface on the SP router	String	Optional
gracefulRestart	BGP graceful restart parameters		Required
enable	True to enable the graceful restart BGP option and False to disable it	Bool	Required
restartTime	Determines how long the peer routers wait to delete stale routes before a BGP open message is received	1 ~ 3600 seconds	Required
stalePathTime	Determines how long a router wait before deleting stale routes after receiving an end of record (EOR) message from the restarting router	1 ~ 3600 seconds	Required

Graceful Restart

When a BGP router restarts, all of its neighbors detect that the BGP router went down, and came back up. It results in the deletion and adding back of the BGP routes in the neighbors. The unnecessary recomputation of routes, called a "routing flap", causes issues on both the BGP and neighbor routers. Graceful Restart allows the system to preserve the routes during BGP restart, which minimizes the negative effects of BGP restart.

BGP Agent Configurator

The Cisco cnBR BGP Agent Configurator allows easy modification of BGP Agent global configurations.

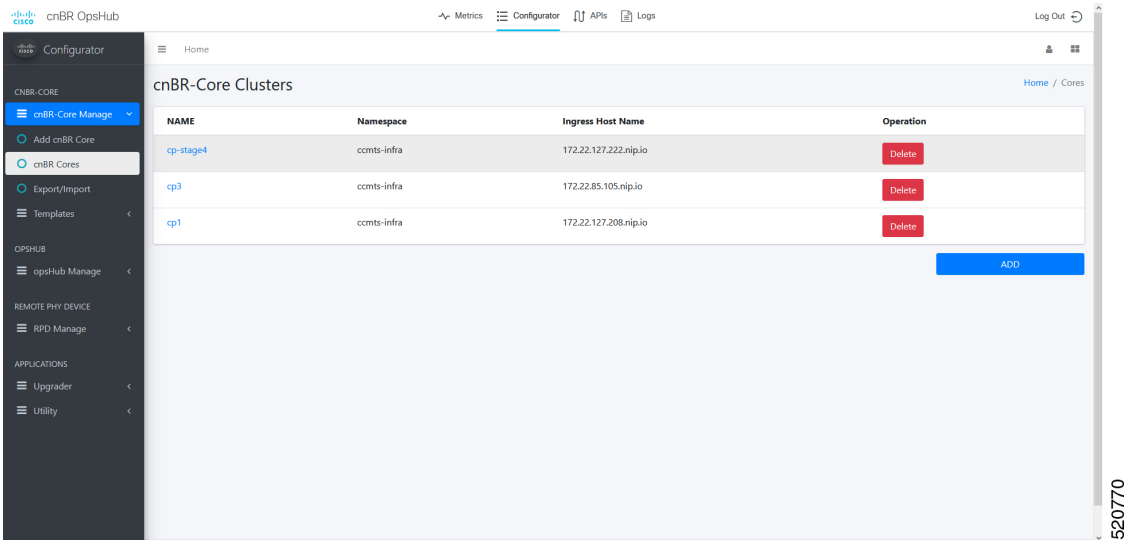
Access BGP Agent Configurator

Step 1 Log in to Cisco Operations Hub.

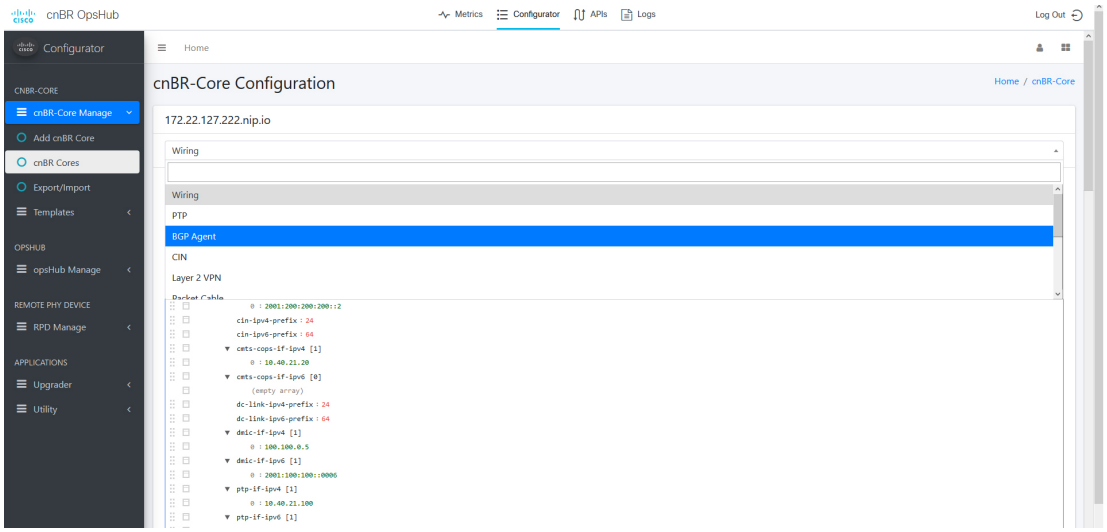
Step 2 Click **Configurator** from the horizontal navigation tab.

Step 3 Click **cnBR Cores** from the vertical navigation tab.

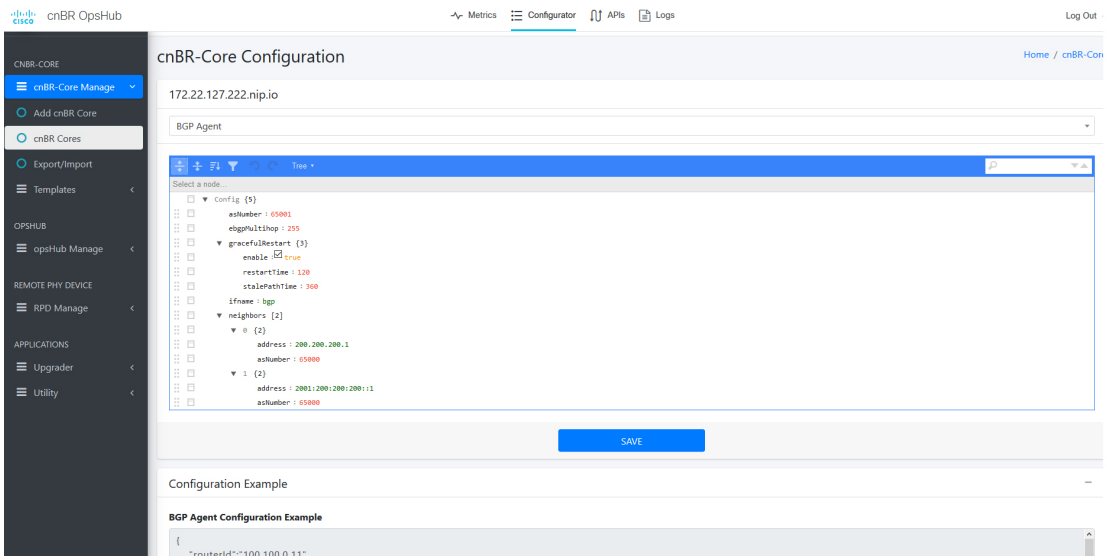
Step 4 Select desired Cisco cnBR Core.



Step 5 Select **BGP Agent** from the drop-down menu.

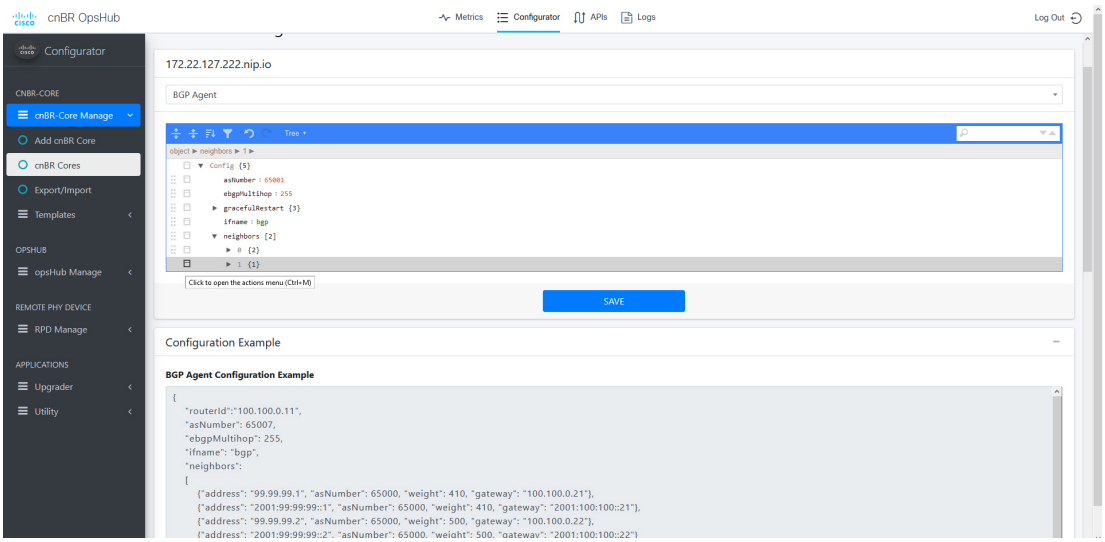


Step 6 Now you're in the BGP Agent Configurator.

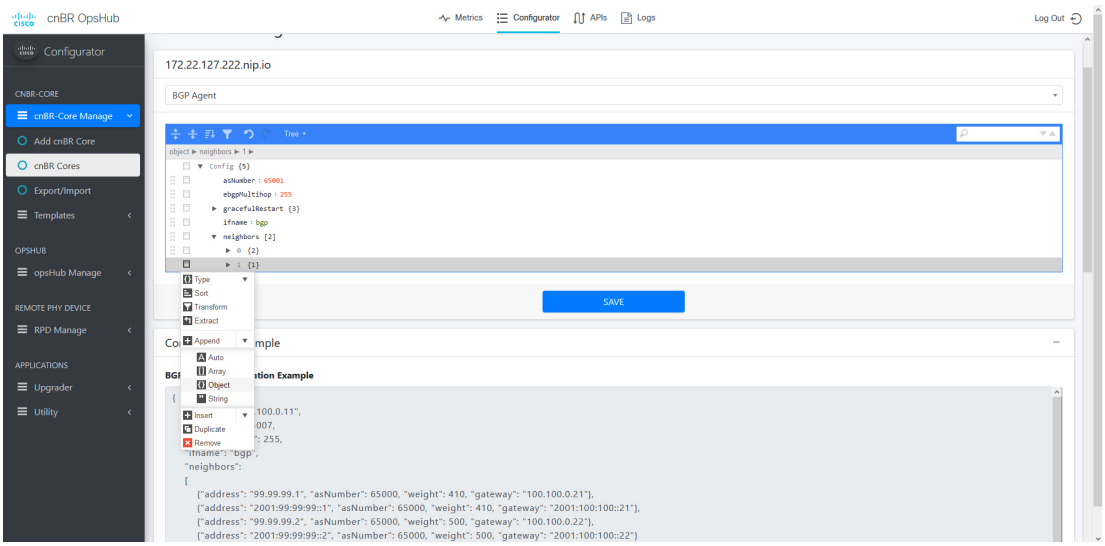


Add BGP neighbors

Step 1 In BGP Agent Configurator, expand `neighbors` field, and click on the edit box of the last element.

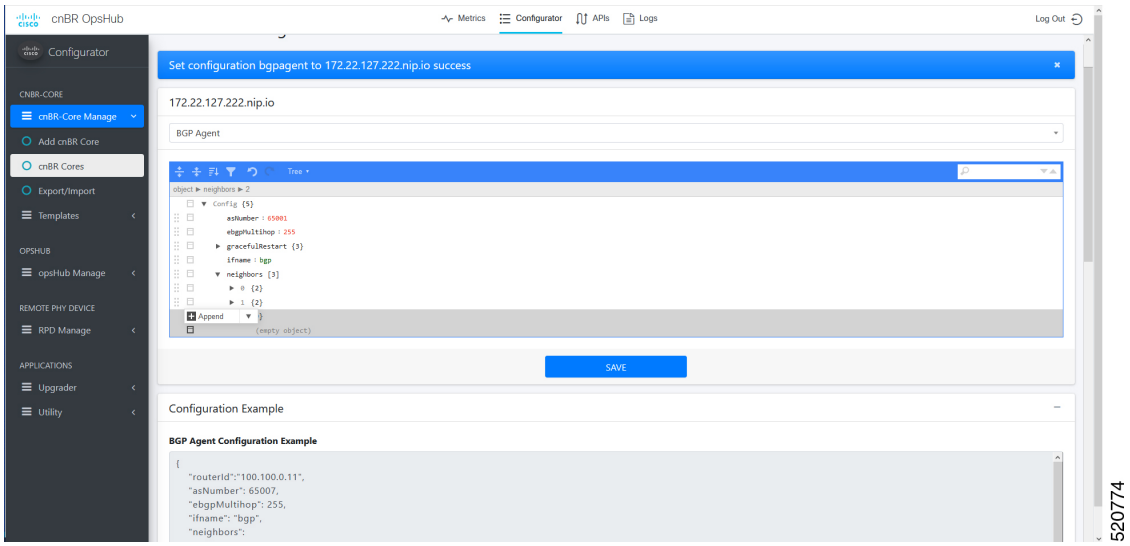


Step 2 From the drop-down menu, expand **Append**, then select **Object**.

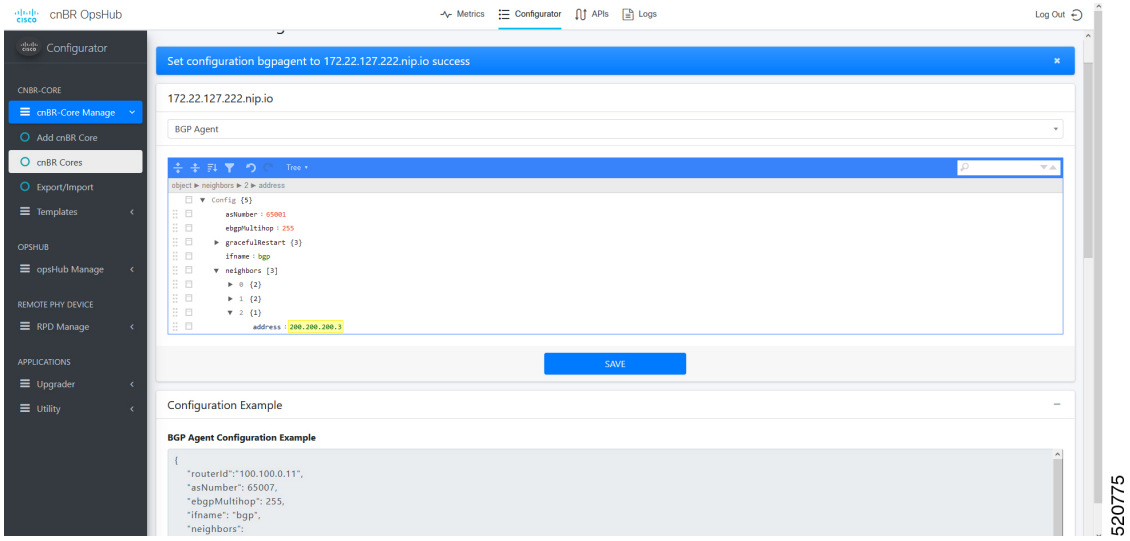


Step 3 In the new object, select the edit box of the `(empty object)` field.

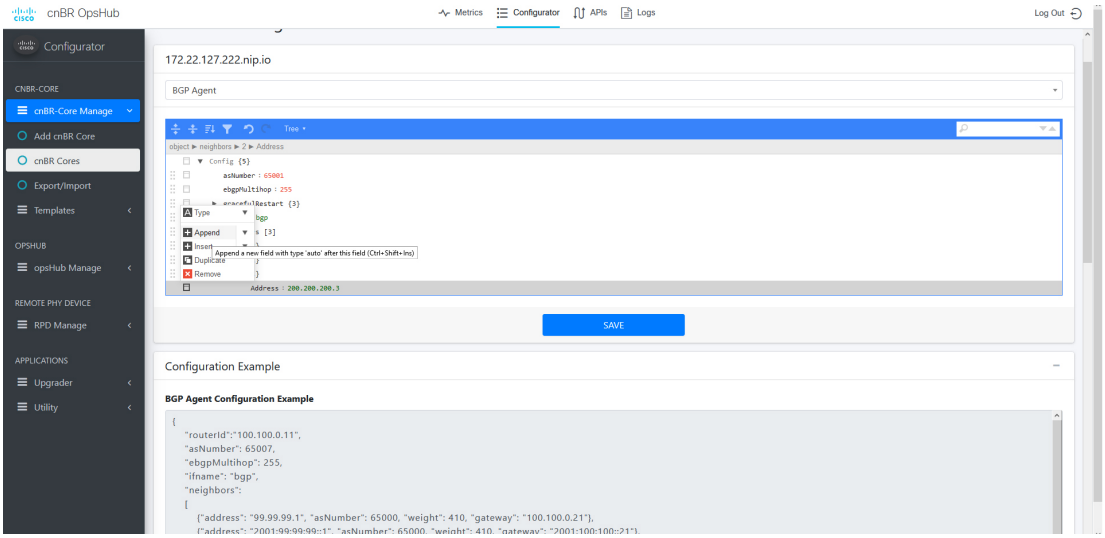
Step 4 Then, select **Append** from the drop-down menu to create an object with two fields.



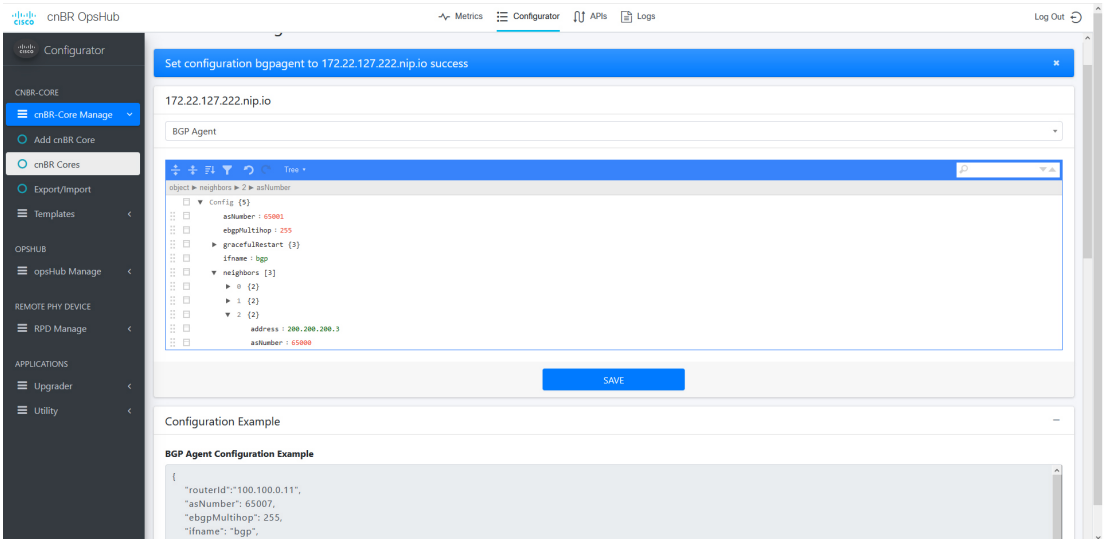
Step 5 In the first field, type `address`, and in the second field, type the IP address of the new neighbor.



Step 6 Select the edit box of the `Address` field. Then, select **Append** from the drop-down menu to create an object with two fields.



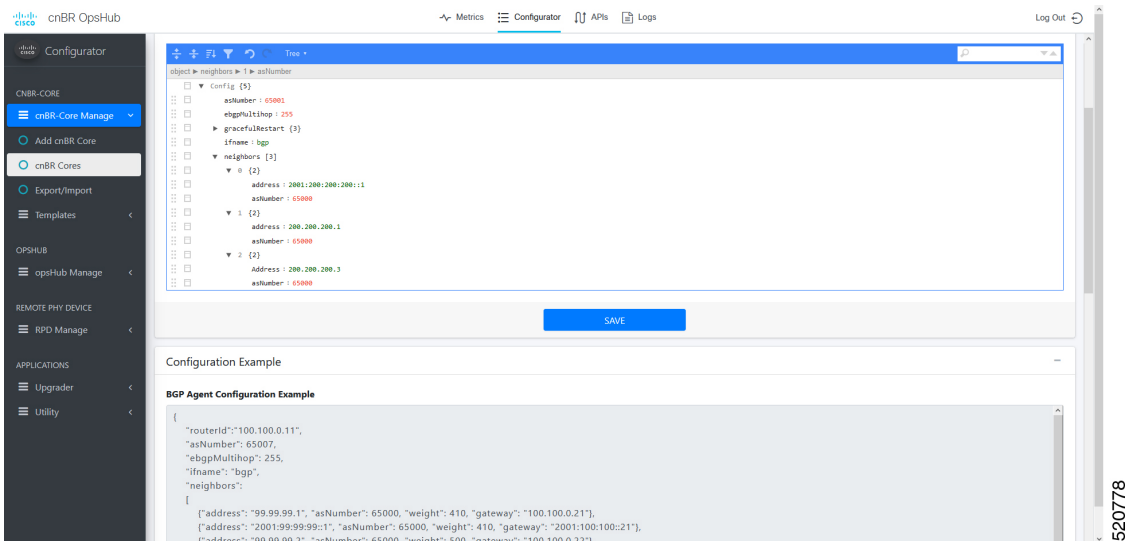
Step 7 In the first field, type `asNumber`, and in the second field, type the AS number of the new neighbor.



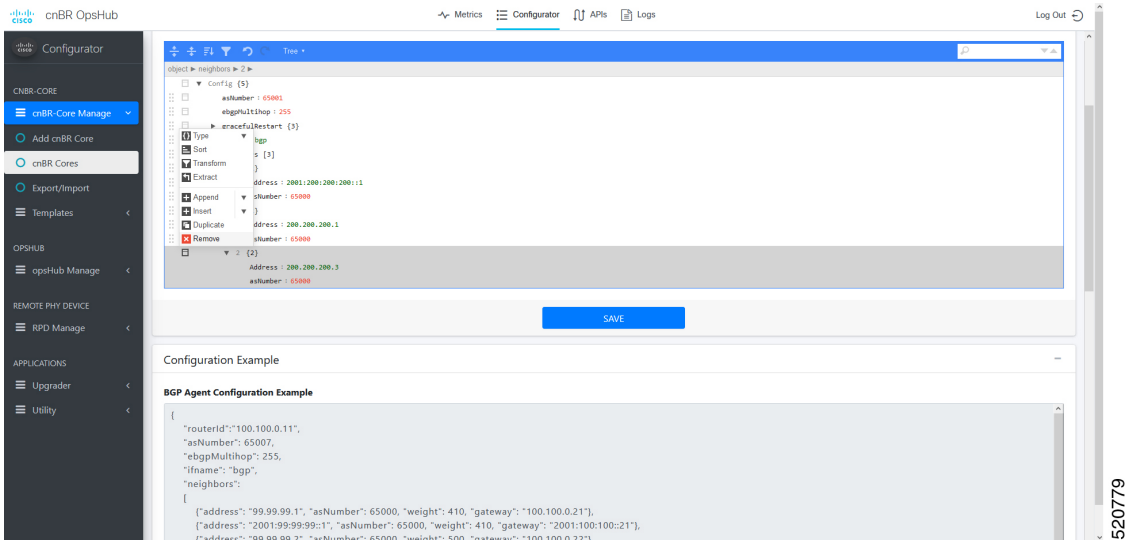
Step 8 Click **Save**.

Delete BGP Neighbors

Step 1 In BGP Agent Configurator, expand all neighbor objects to find out the neighbor to delete.



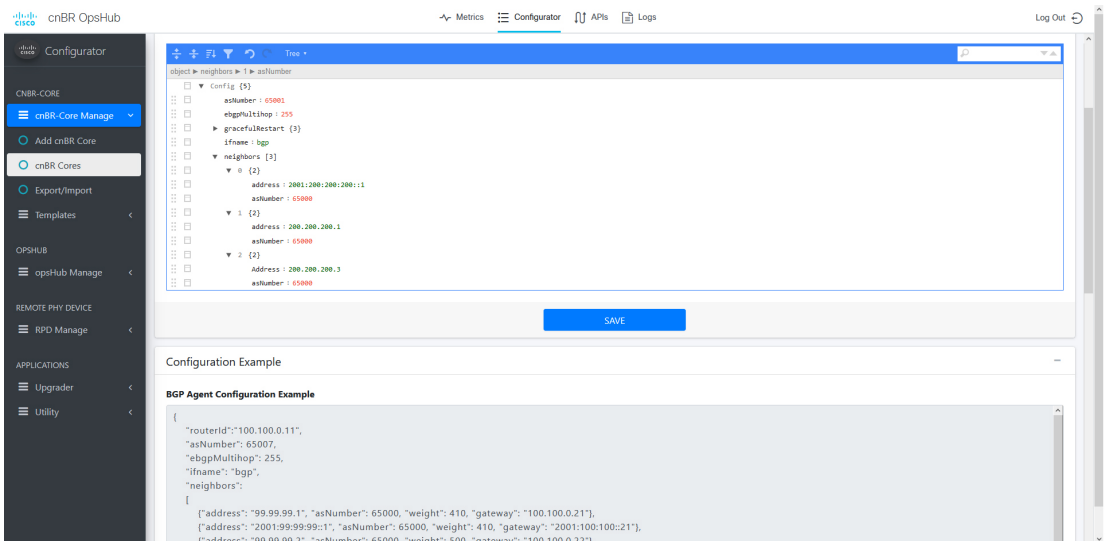
Step 2 Select the edit box of the neighbor object to delete, then select **Remove**.



Step 3 Click **Save**.

Get BGP Neighbors

BGP neighbor information is stored in the `neighbors` field in the BGP Configurator.



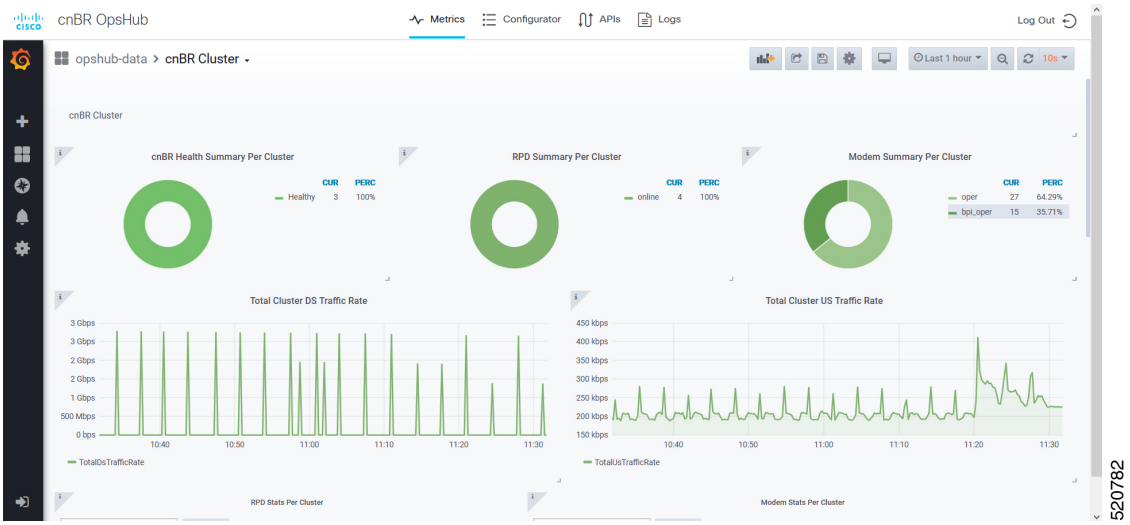
BGP Agent Dashboard

The Cisco cnBR BGP Agent Dashboard provides visibility into the BGP IPv4 and IPv6 routes and operation.

Access BGP Agent Dashboard

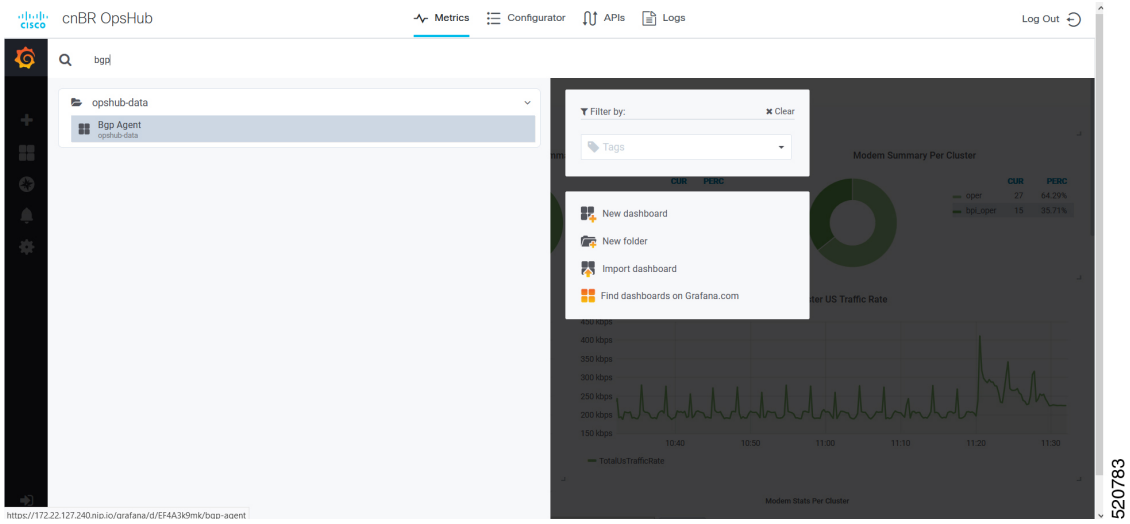
Step 1 Log in to Cisco Operations Hub.

Step 2 Click on **cnBR Cluster** to bring up the search menu.

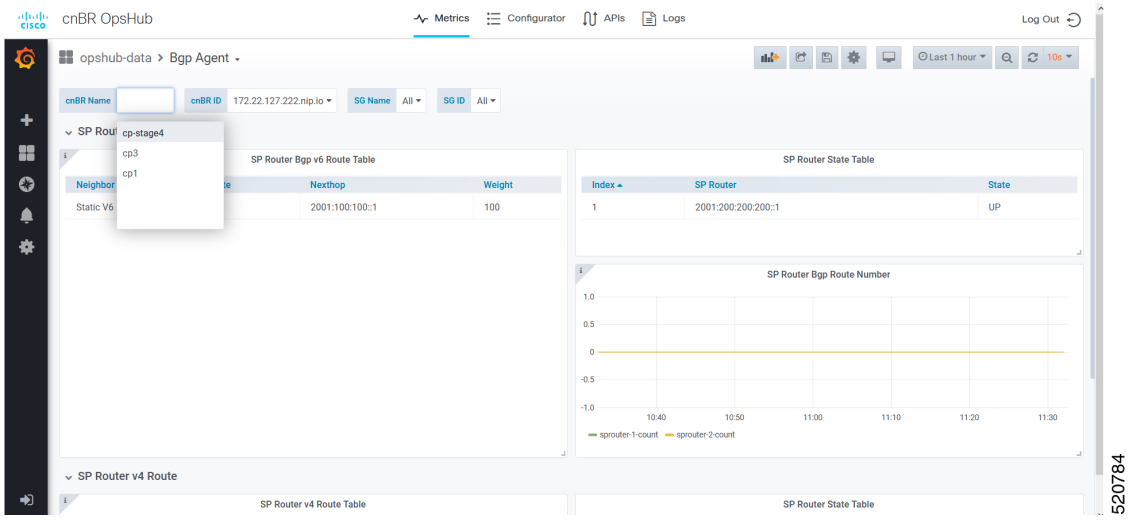


Step 3 Type `bgp` in the search bar.

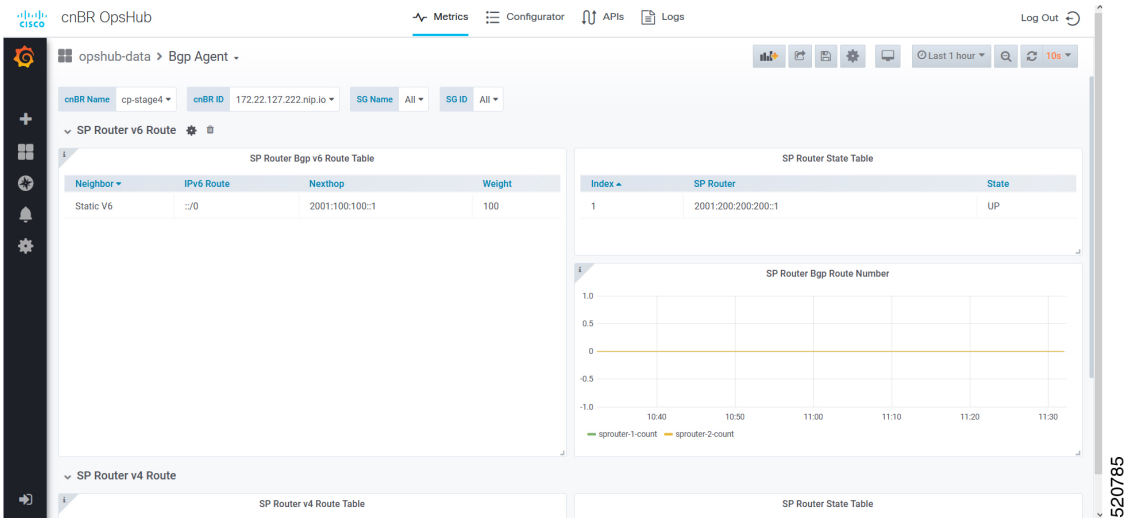
Step 4 Then click the dashboard **Bgp Agent**.



Step 5 Select the desired Cisco cnBR from the **cnBR Name** drop down menu.



Step 6 Now you can see the BGP Agent Dashboard of the desired Cisco cnBR.



WAN Route Table

WAN Route Table displays the default routes generated by BGP Agent, and BGP routes received by the SP Router.

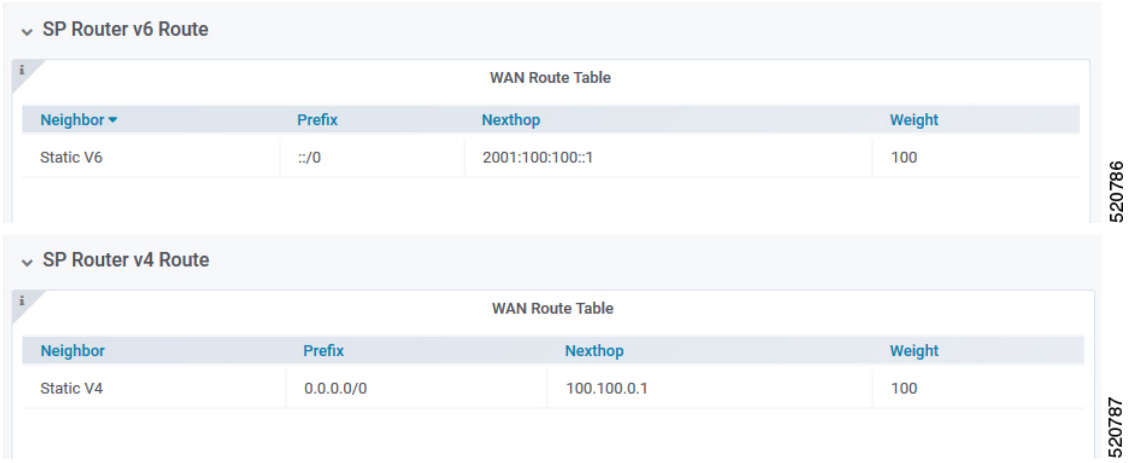
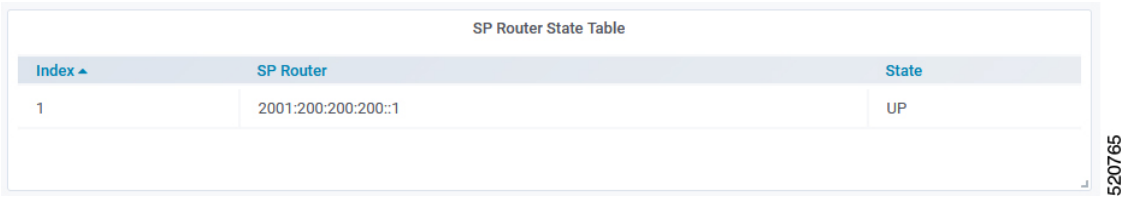


Table 9: Parameters

Name	Description
Neighbor	Neighbor IP address
Prefix	Network segment of route
Nexthop	IP address of next hop to get to destination
Weight	Weight parameter described in Configuration Parameters, on page 86

SP Router State Table

SP Router State Table displays the connection state between BGP Agent and SP Router. UP state indicates that the connection is established, and DOWN state indicates the connection is not established.



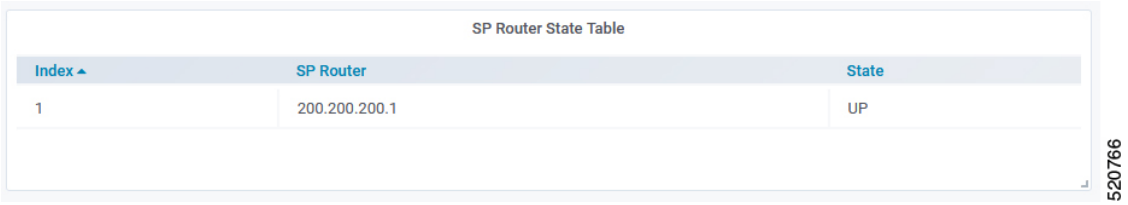


Table 10: Parameters

Name	Description
SP Router	The IP address of the SP Router
State	State of the connection between BGP Agent and SP Router

BGP Route Table

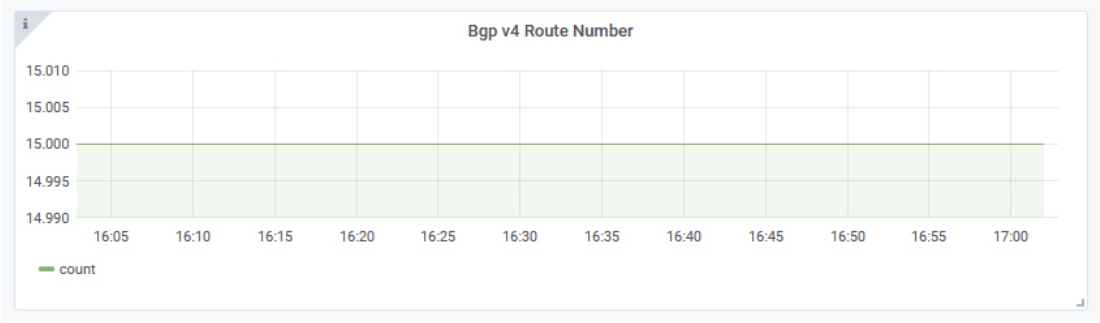
BGP Route Table displays the BGP routes that is sent to the SP Router to route packets from CM to the correct DP.

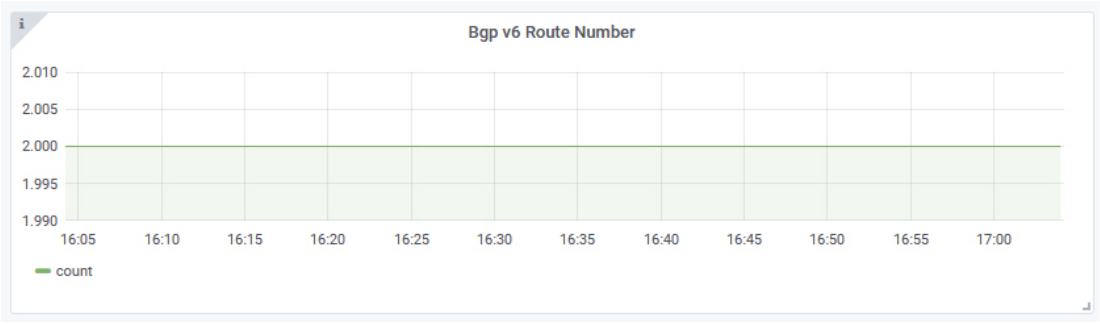
Table 11: Parameters

Name	Description
SG Name	Service Group name corresponding to the route
SG ID	Service Group ID corresponding to the route
IP Route	Destination IP address
NextHop	Next IP address hop to get to destination

BGP Route Number

BGP Route Number displays the number of BGP routes installed into SP Router over time.





- X-axis: Time
- Y-axis: Number of BGP routes

BGP Route Rate

BGP Route Rate displays the change rate of BGP routes over time.

- X-axis: Time

- Y-axis: Change rate of BGP routes

L2VPN

The Cisco cnBR application emulates the Layer 2 virtual private network (L2VPN), when L2VPN devices across shared or public networks appear as computing devices that are directly connected to a switch device. Therefore, Layer 2 packets from one device can reach the other device without changes to the Layer 2 packet header, similar to the traditional Layer 2 Forwarding method.

Several tunneling protocols are used to implement L2VPN. Cisco cnBR supports the point-to-point mode L2VPN for the IEEE 802.1Q (dot1q) protocol.

For the dot1q L2VPN, Cisco cnBR adds one layer dot1q tag for the upstream packet and removes the tag at the receiving end.

Cisco cnBR supports both cable modem (CM) based L2VPN and service flow (SF) based L2VPN.

- CM-based L2VPN: One CM can configure one L2VPN service. Primary upstream and primary downstream packets are encapsulated into a L2VPN tunnel.
- Service flow-based L2VPN: One CM can configure up to four L2VPN services using the CM configure file TLV. A maximum of eight upstream SFs and eight downstream SFs are supported for each L2VPN service. The upstream classifier on the CM and downstream classifier on the Cisco cnBR router are used to classify different packets into L2VPN service flows.

Cisco cnBR supports the following types of L2VPN tunnel:

Tunnel Type	CM-based	SF-based
dot1q	• dot1q tunnel • Configure by Rest API • One L2VPN per CM	• dot1q tunnel • Configured by CM configuration file TLV • Up to 4 L2VPN per CM

Configure L2VPN

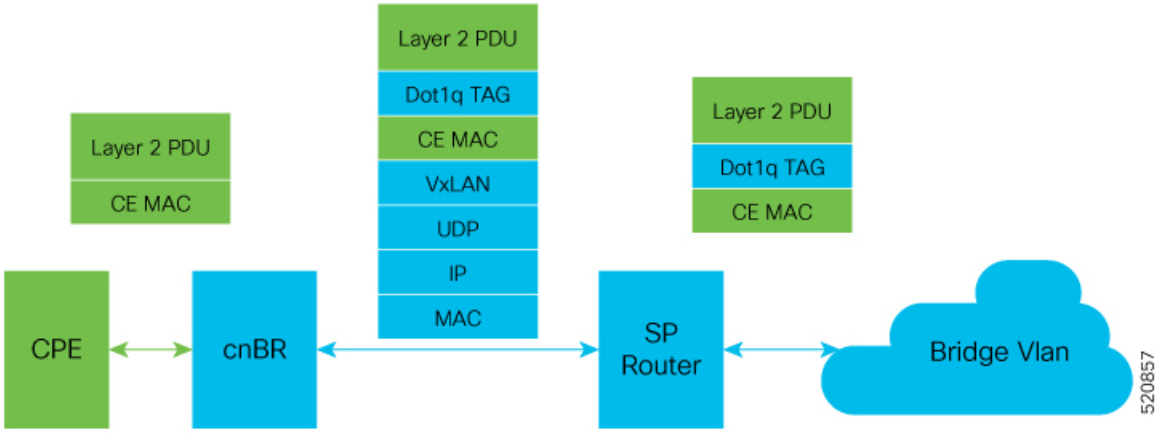
The dot1q L2VPN is implemented using the Cisco cnBR router with a Service Provider (SP) router.

SP routers are Cisco ASR 9000, Cisco ASR 1000, or Cisco Network Convergence System 5501.

The connection between the Cisco cnBR router and the SP router is supported by either the VxLAN mode or the VLAN mode.

VxLan Mode

The following image shows the dot1q L2VPN packet flow from CPE to the dot1q tunnel.

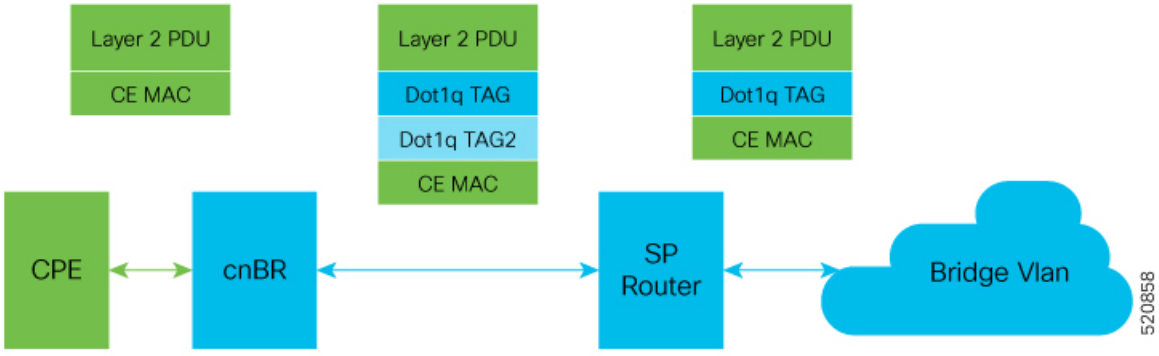


The following table summarizes the configuration that is required for the supported L2VPN types:

Tunnel Type	CM-based	SF-based
dot1q	• Cisco cnBR configuration: static dot1q L2VPN • Cisco cnBR configuration: dot1q VxLAN wiring • SP router configuration: dot1q VxLAN wiring	• CM configure file: dot1q L2VPN related TLV • Cisco cnBR configuration: dot1q VxLAN wiring • SP router configuration: dot1q VxLAN wiring

VLAN Mode

The following image shows the dot1q L2VPN packet flow from CPE to the dot1q tunnel.



The following table summarizes the configuration that is required for the supported L2VPN types:

Tunnel Type	CM-based	SF-based
dot1q	• Cisco cnBR configuration: static dot1q L2VPN • Cisco cnBR configuration: dot1q VxLan wiring • SP router configuration: dot1q VxLan wiring	• CM configure file: dot1q L2VPN related TLV • Cisco cnBR configuration: dot1q VxLan wiring • SP router configuration: dot1q VxLan wiring

Cisco cnBR L2VPN Configuration

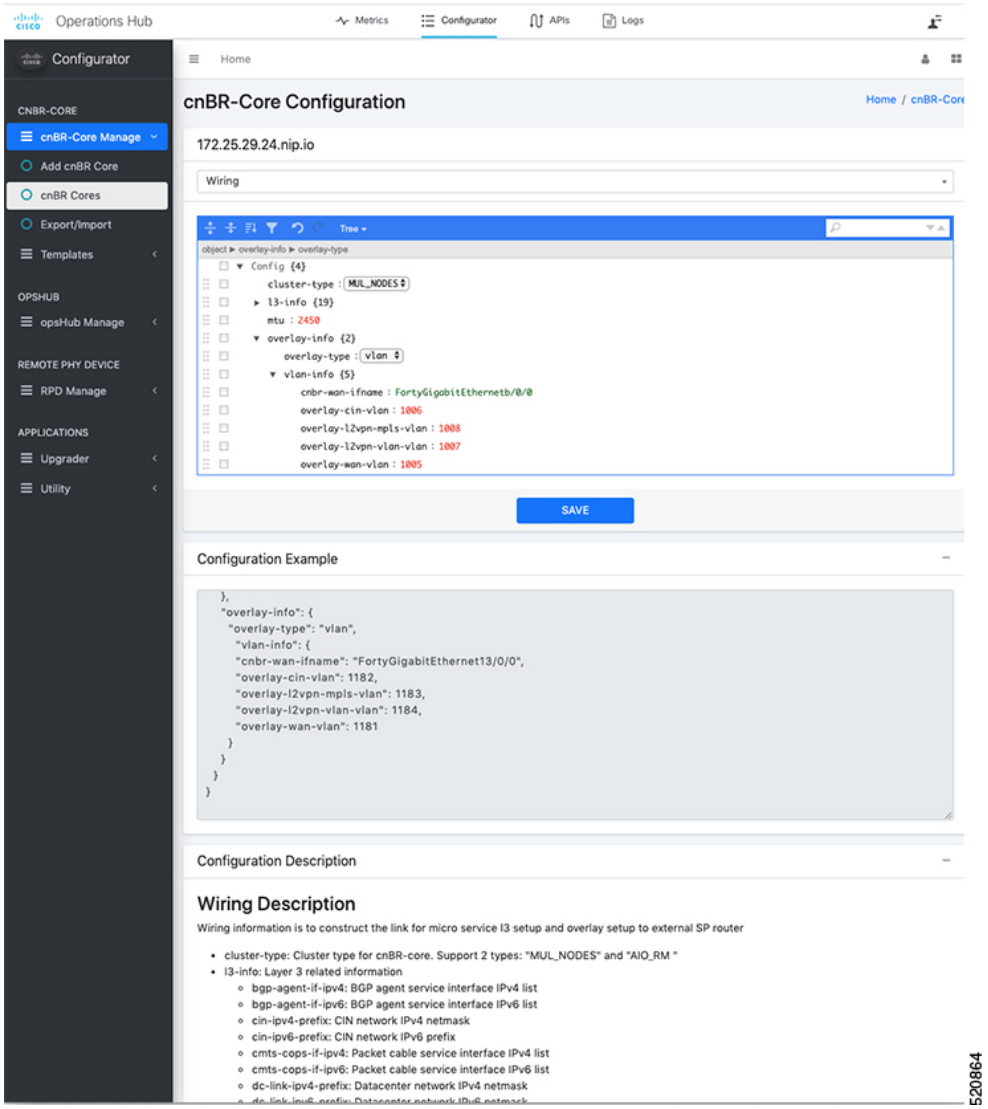
For both CM-based and SF-based L2VPN, configure the L2VPN related VLAN or VxLAN that connects to the SP router. Use the **Configurator** to configure the wiring.

For CM-based L2VPN, configure the static L2VPN map by using the REST API.

Step 1 Choose **Cisco Operations Hub** > **Configurator**.

Step 2 Choose **cnBR Cores** from the left pane and click the required cnBR Core.

Step 3 Select **Wiring** from the drop-down list.



Step 4 Update the configuration as required and click **SAVE**.

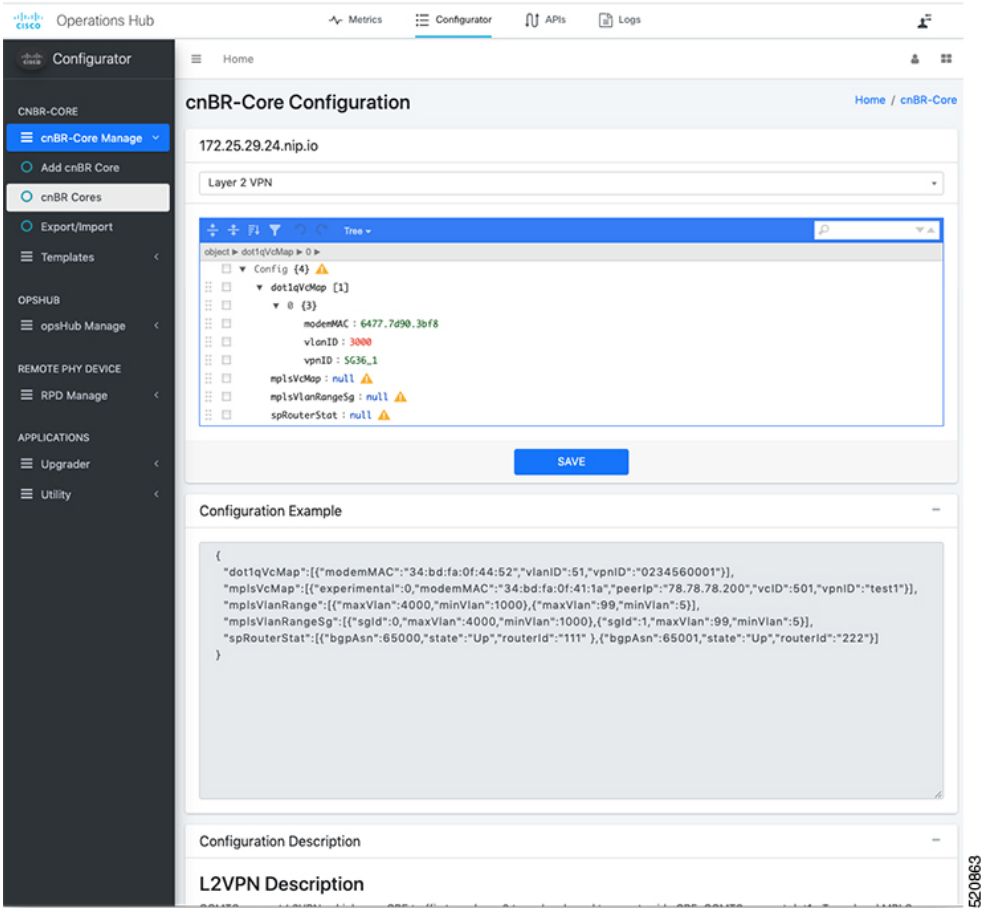
Static Dot1q L2VPN

To configure a cable modem (CM) as dot1q CM-based L2VPN, upstream traffic (primary service flow) adds one-level dot1q tag. Each L2VPN must have a different.`VLanId`

Step 1 Choose **Cisco Operations Hub** > **Configurator**.

Step 2 Choose **cnBR Cores** from the left pane and click the required cnBR Core.

Step 3 Select **Layer 2 VPN** from the drop-down list.



Step 4 Update the configuration as required and click **SAVE**.

CM Configuration File TLV Definition

SF-based L2VPN depends on the CM configuration file TLV to set up L2VPN service, L2VPN service flow, and L2VPN classifier. For more details, see the CableLabs document: *Business Services over DOCSIS Layer 2 Virtual Private Networks*.

IPv6

Cisco cnBR supports IPv6 protocol when communicating with the following network devices:

- Cable Modem (CM)
- Customer Premise Equipment (CPE)-Equipment that is connected to the CM at the customer premise.

Note Cisco cnBR supports dual-stack IPv4 and IPv6 protocols (It supports both IPv4 and IPv6 addresses at the same time).

Cisco cnBR as DHCP Relay Agent

Cisco cnBR supports CMs and CPEs operating in IPv4, IPv6, and dual-stack modes.

In a Cisco cnBR system, cable modems and some of the associated CPEs acquire IP addresses from a DHCP server in the network. These cable modems, their associated CPEs, and the DHCP server are not on the same physical network. In this scenario, Cisco cnBR acts as a DHCP relay agent to relay all requests and replies between the clients (CM and CPE) and the DHCP server. The DHCP relay agent in Cisco cnBR supports both IPv4 and IPv6 addressing.

When CMs operate in the IPv6 mode, especially only in the IPv6 mode, configure the TFTP server on the Converged Interconnect Network (CIN) network to operate in IPv6 mode. This configuration allows the CMs to connect to the TFTP server in IPv6 mode and download their CM configuration file.

Note DHCP messages from RPDs does not reach the DHCP relay agent in the Cisco cnBR router. These DHCP messages from RPDs can reach the DHCP server in the CIN without using the DHCP relay agent in Cisco cnBR.

For more details, see the DHCP Relay Service, on page 69 section.

DOCSIS

Cisco cnBR provides DOCSIS functionality, enabling next generation broadband capability for your Distributed Access Architecture.

Upstream Resiliency

A DOCSIS 3.0+ cable modem (CM) operating in upstream channel bonding mode, or Multiple Transmit Channels (MTC) mode, utilizes its assigned upstream channels, or Transmit Channel Set (TCS), to transmit data packets when Cisco cnBR grants transmission opportunities on those channels.

The Upstream (US) Resiliency feature provides the capability to automatically suspend granting transmission opportunities for a CM on one or more certain upstream channels when the Cisco cnBR determines that those upstream channels are no longer usable for the CM.

Cisco cnBR determines the usability of the upstream channel by polling the CM with Station Maintenance (SM) Ranging opportunities every 20 seconds on each of the upstream channels in the CM TCS, and waits for the Range Request from the CM on those upstream channels. If Cisco cnBR does not receive the Range Request message from the CM after granting an SM Ranging opportunity, the Cisco cnBR reduces the SM grant interval from 20 seconds to 1 second for the CM on the affected upstream channel. If the Cisco cnBR still can not receive the Ranging Request from the CM for the next 25 times, the Cisco cnBR then considers the upstream channel to be impaired for that CM.

The CM is then classified as operating in the Upstream Partial Service state. The RPTS, nRTPS service flows used by the CM, if any, will be moved to another upstream channel in the updated TCS of the CM. Once the CM is able to Range on all its TCS channels again, the CM exits the Partial Service state.

Note Other non-Best Effort Service Flows, such as UGS, UGS-AD, will not be moved away from the impaired upstream channel. Future Cisco cnBR releases will address this issue.

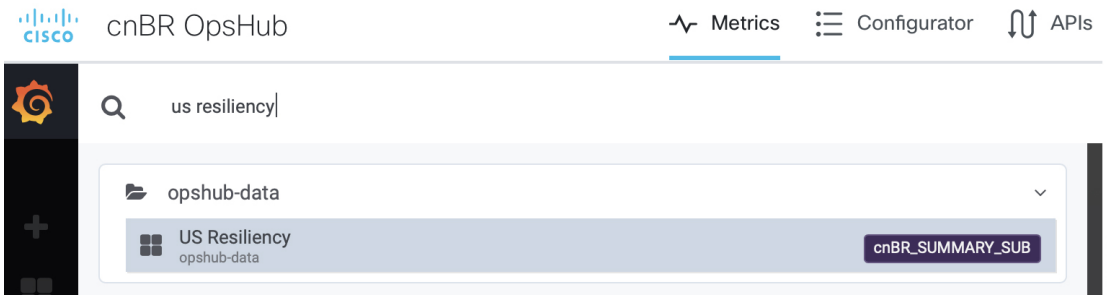
Upstream resiliency is enabled by default. And it does not require configuration, that is, you do not need to set US Resiliency parameters in the Autodeployer configuration file.

Monitor Upstream Resiliency

The Upstream Resiliency Dashboard displays the statistics of the cable modems that are in upstream partial service state, and the status of the upstream channels in the Cisco cnBR. You can use the Dashboard to identify impaired upstream channels, and help to narrow down part of the cable plant that needs servicing.

US Resiliency Cisco Operations Hub Dashboard

Enter the Cisco Operations Hub URL `https://{opshub-ip}.nip.io` in web browser. After logging in, click the horizontal navigation tab Metrics on the top of the page. Then search for the US Resiliency dashboard by entering `us resiliency`, and click on the matching result that appears in the result panel.



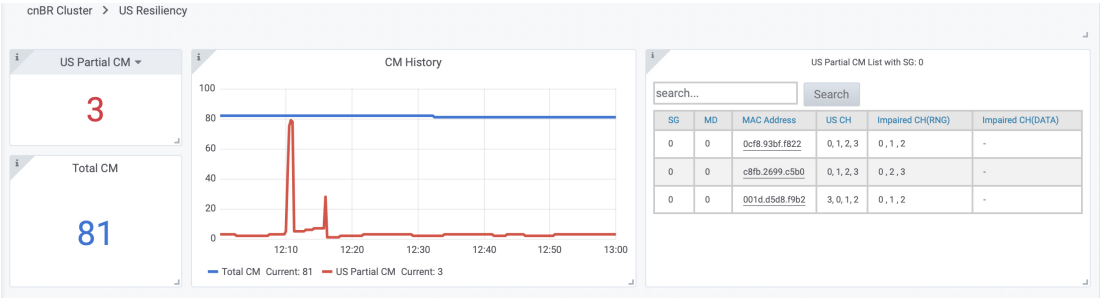
In US Resiliency Dashboard, click on the **cnBR ID** drop-down list to select the Cisco cnBR to monitor. You must add the Cisco cnBR to the Cisco Operations Hub, then you can see it in the drop-down list. After you select the Cisco cnBR, select the desired Service Group by clicking on the **SG ID** drop-down list. Similarly, you must first fill in and configure the Service Group, then you can select it in the **SG ID** drop-down list.

Cluster Statistic

The Cisco cnBR Cluster US Resiliency statistic panel provides the current and historical statistics for the selected Cisco cnBR and Service Group which includes:

- The current number of cable modems that are in partial service mode in the selected Cisco cnBR cluster.
- The current total number of cable modems detected by the selected Cisco cnBR cluster.

- The historical count of the cable modems that are in upstream partial service mode and the total number of cable modems over time.
- The current list of the cable modems in upstream partial service mode.

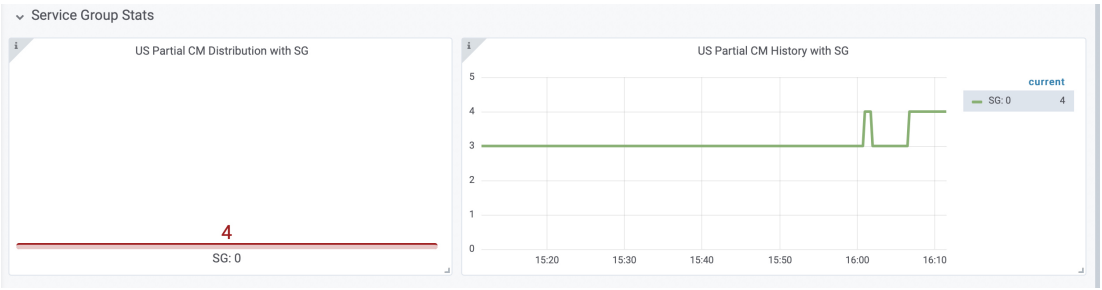


520806

Service Group Statistic

The Service Group Statistic panel provides the current and historical statistics for the selected Service Group which includes:

- The current number of cable modems that are in partial service mode in a specific Service Group.
- The historical count of the cable modems that are in upstream partial service mode in a specific service group.



520809

Upstream Channel Statistic

The US Channel Statistic panel provides the current and historical statistics for each upstream channel in the selected Service Group, which includes:

- The current number of cable modems that are in partial service mode for each upstream channel in the selected Service Group.
- The historical count of the cable modems that are in upstream partial service mode in each upstream channel in the selected Service Group.

When a significant number of CMs are having problem on a specific channel, there may be channel frequency interference in a certain segment of the cable plant or service neighborhood.

When a few CMs are having problem on all channels, it may indicate that there is loose connector or deteriorating cable on certain segment of the service neighborhood, or those CMs may be close to the boundary of the supported service area. It may also indicate cabling problem of those CMs at the customer homes.

In the preceding cases, you may need more investigation to better understand and troubleshoot the problem, and proactively implement remedies if needed (before you call the service center).

Downstream Resiliency

DOCSIS 3.0+ Cable Modems (CMs) use downstream bonding groups to receive data. In this scenario, when one or more downstream channels get impaired, it causes packet drops in that particular channel. Furthermore, as the packets need to be reordered, packet drop in one channel can cause reorder timeout and large packet delay, in a continuous manner. Therefore, detecting channel impairment and mitigating this type of condition is important for proper downstream channel bonding operation.

DOCSIS provides a mechanism that let modems detect this condition and report the issue through a CM-STATUS MAC Management Message (MMM). Therefore CMTS can stay informed about one or more channels that are impaired. However, the DOCSIS specification does not specify how the CMTS should handle the impaired channel conditions. The implementation is up to CMTS vendor.

Upon receiving a CM-STATUS MMM indicating DS channel impairment, the Cisco cnBR temporarily removes the impaired DS channel from the bonded DS Receive Channel Set (RCS). From the CM's perspective, its current RCS persists during impairment. It allows the CM to monitor all DS channels and detect when impairment is gone from the impacted DS channel. Once the Cisco cnBR has received a CM-STATUS MMM indicating DS channel impairment is gone, the previously impaired DS channel will be added back to the RCS.

Note DS resiliency applies to nonprimary DS channels only. DS impairment of a CM's primary channel is an event that cannot be mitigated and will result in a CM dropping offline.

Current DS Resiliency Feature handles three failure modes:

- MDD timeout
- QAM lock failure
- OFDM profile failure

Four types of CM-STATUS messages are handled for supporting DOCSIS 3.0 DS resilience:

- MDD timeout (Event Code 1)
- QAM lock failure (Event Code 2)
- MDD recovery (Event Code 4)
- QAM lock recovery (Event Code 5)

Two types of CM-STATUS message are handled for supporting DOCSIS 3.1 DS resilience:

- DS OFDM Profile Failure (Event Code 16)
- DS OFDM Profile Recovery (Event Code 24)

Configure DS Resiliency

The DS Resiliency configuration is a sub config of the service group configuration. To enable the DS resiliency feature, add the following sub config to all SG configurations.

```
"DsResilCfg":
    {
        "DampenTime":30,
        "ResilEn":"true"
    },
```

To disable the DS resiliency feature, change the `"ResilEn":"true"` to `"ResilEn":"false"` in all SG configurations.

Note Even with DS Resiliency disabled, logs and dashboards show all the events and impaired CMs, and don't change service flow.

Note The unit of dampen time is seconds.

Update the DS Resiliency Configuration Using Cisco Operations Hub

After the initial configuration of DS Resiliency during deployment using the Autodeployer, you can also update the configuration through the Cisco Operations Hub - Configurator Panel using the following steps:

Step 1 Select the Configurator tab, then click **Export/Import** on the side panel to get to the Export/Import page.

Step 2 Under the Export cnBR Configuration section, select the Cisco cnBR to update from the drop-down list.

Step 3 Click the **Export** button to retrieve the current SG configuration of the selected Cisco cnBR.

Step 4 Update the configuration in the `dsResilCfg` section of the SG configuration.

Step 5 Save the updated file on the local disk.

Step 6 To push the updated SG configuration, under the Import cnBR Configuration File section, select the Cisco cnBR that you want to update from the drop-down list.

Step 7 Click on the **Browse** button to locate the saved configuration file.

Step 8 Click on the **Import** button to push the updated SG configuration.

DS Resiliency Monitor Statistics

Step 1 Log in to Cisco Operations Hub.

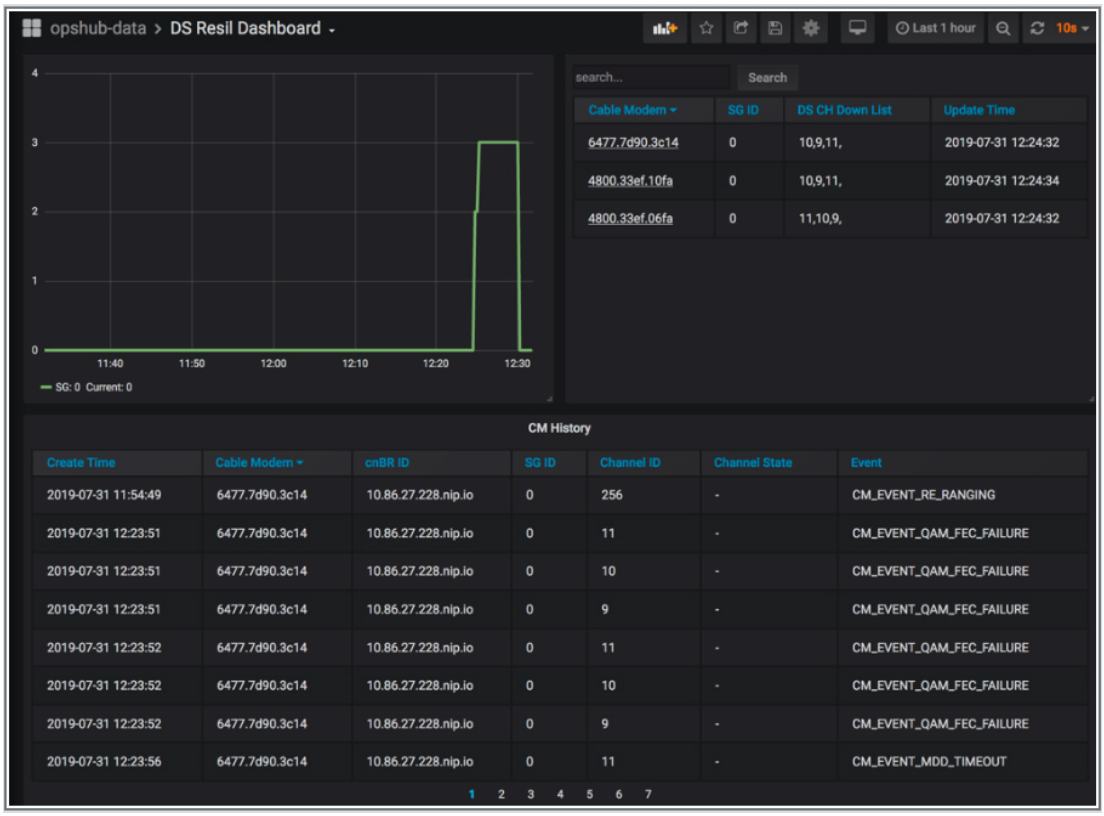
Step 2 Click on **cnBR Cluster** to bring up the search menu.

Step 3 Type `resil` in the search bar.

Step 4 Then, click on the dashboard named DS Resil Dashboard.

Step 5 Select the desired Cisco cnBR from the cnBR Name drop down menu.

Figure 20: DS Resil Dashboard



OFDM Container

Cisco cnBR provides DOCSIS 3.1 support by introducing Orthogonal Frequency-Division Multiplexing (OFDM) channels in the downstream direction, and Orthogonal Frequency-Division Multiple Access (OFDMA) channels in the upstream direction. OFDM allows for higher throughput and higher spectral efficiency, while still allowing backward compatibility to DOCSIS 3.0.

The OFDM Channel support includes 1 OFDM channel per Service Group (SG) with a channel bandwidth from 24 - 192 MHz wide. Currently, Cisco cnBR supports OFDM channel as a non-primary channel, and the OFDM container is used within a downstream bonding group with up to 32 SC QAM channels.

Each OFDM channel supports the following:

- **Control profile:** The control profile is known in CM-SP-MULTIv3.1 as Profile A, using profile ID 0. This denotes the common profile that all modems can receive and decode. A modem uses Profile A when it first initializes.
- **NCP profile**: There is a dedicated NCP profile, the Next Codeword Pointer. The NCP profile indicates which subcarriers are usable for NCP and what modulation is to be used on each subcarrier.
- **Data profile**: An OFDM channel supports a maximum of five data profiles. The data profiles are referred to as profile B, C, D, and so on, in CM-SP-MULTIv3.1.

Configure OFDM Port

Complete the following steps to configure the OFDM port:

Step 1 Configure the OFDM Frequency Exclusion band. The OFDM Frequency exclusion band configuration is supported at the DS port level. The OFDM configuration parameters are listed in the following table:

Table 12: OFDM Port Configuration Parameters

OFDM Frequency Exclusion Band Parameter	Minimum (MHz)	Maximum (MHz)	Default
Channel ID in SG	158	162	N/A
Start frequency	108	1217	N/A
Width	1	1110	N/A

Step 2 Configure OFDM channel in SG. OFDM channels are numbered from 158 to 162. An OFDM channel number must be present in the `channel` set under a `dsPort` for its configuration to take effect.

Note Only a single OFDM channel for each SG is supported.

See the following DS port configuration example:

```
"rpdCfg":
    [{
        "rpdIp":"$RPD0_IP",
        "rpdMac":"$RPD0_MAC",
        "entries":
        {
            "dsPort":
            [{
                "portId": 0,
                "basePower": 21,
                "rfMute": false,
                "adminState": "Up",
                "ofdmFreqExclBand": [{"startFreq":900000000,"width":10000000}],
                "channel": [0,1,2,3,158]
```

```
            }],
            "usPort":
            [{
                "portId": 0,
                "channel": [0,1,2,3]
            }],
            "fiberNode":
            [{
                "Id":0,
                "DsPort":0,
                "UsPort":0
            }]
        }
    }],
```

Configure OFDM Channel

Complete the following steps to configure the OFDM channel:

Go through the OFDM channel-level configuration parameters listed in the following table:

Table 13: OFDM Channel Configuration Parameters

OFDM Frequency Exclusion Band Parameter	Minimum (MHz)	Maximum (MHz)	Default
Channel ID in SG	158	162	N/A
Start frequency	108	1218	N/A
Width	24	192	N/A
PLC start frequency	108	1218	N/A
Cyclic prefix	192, 256, 512, 768, 1024		1024
Interleaver depth	1	32	16
Pilot scaling	48	120	48
Roll-off	64, 128, 192, 256		128
Subcarrier spacing	25 KHz, 50 KHz		50 KHz

Note As a Cisco cnBR convention, OFDM channels use DOCSIS Channel ID (DCID) of 158, or higher.

See the following DS channel configuration example. The OFDM channel is configured within the `ofdmDs` block at the SG level. The following block configures OFDM channel #158:

```
i
"ofdmDs":
[
    {
      "cyclicPrefix": 512,
```

```
      "idInSg": 158,
      "interleaverDepth": 16,
      "pilotScaling": 48,
      "plc": 873000000,
      "profileControl": "QAM64",
      "profileData": [
        {
          "id": 1,
          "modulationDefault": "QAM1024"
        },
        {
          "id": 2,
          "modulationDefault": "QAM2048"
        },
        {
          "id": 3,
          "modulationProfile": 9
        }
      ],
      "profileNcp": "QAM16",
      "rollOff": 128,
      "startFrequency": 867000000,
      "subcarrierSpacing": "50KHZ",
      "width": 192000000
    }
],
```

Configure Downstream Modulation Profile

A profile is a list of modulation orders that are defined for each of the subcarriers within an OFDM channel. The Cisco cnBR can define multiple profiles for use in an OFDM channel. The profiles can differ in the modulation orders that are assigned to each subcarrier.

Choose either of the supported modulation orders:

- Constant Modulation Orders

  When a profile has the same QAM modulation for all the subcarriers, it is specified by the keyword `modulationDefault`, and the modulation value (for example - `QAM256`) inside the `profileData` block for the OFDM channel configuration. See the example provided in Configure OFDM Channel, on page 111.

- Variable Modulation Orders

  When a profile has Variable QAM modulations for the subcarriers, it is specified by a separately defined block within `ofdmModProfs`. The following example defines a data profile that has modulation order of 512QAM for all subcarriers except in two segments where the modulation order is 1K QAM and 4K QAM, respectively.

  The OFDM profile configurations must be enabled within the `ofdmModProfs` block at the SG level. The following example configures a mixed profile, with a profile ID 9 and named `512-1k-4k`.

```
"ofdmModProfs":
[
    {
      "assigns": [
        {
          "modulation": "QAM512",
          "rangeSubcarriers": {
```

```
            "freqAbs": 935000000,
            "width": 7405000
          }
        },
        {
          "modulation": "QAM1024",
          "rangeSubcarriers": {
            "freqAbs": 959000000,
            "width": 6000000
          }
        }
      ],
      "description": "512-1k-4k",
      "idInSg": 9,
      "modulationDefault": "QAM4096"
    }
]
```


Configure Modulation Profile Display

The profile list that is used by an OFDM channel is displayed in the OFDM Channel Profile Data dashboard in Cisco Operations Hub.

To view the OFDM profile data, perform either of the following step:

- To load the OFDM Channel Profile Data dashboard:

  On the Cisco Operations Hub, click **Metrics** and search for `OFDM Channel Profile Data`.

- To load the OFDM Modulation Profile Data dashboard:

  On the Cisco Operations Hub, click **Metrics** and search for `OFDM Modulations Profile Data`.

  The data profile that is defined for variable modulation orders is displayed in the OFDM Modulation Profile Data page in Cisco Operations Hub.

Update Configuration Using Operations Hub

The configuration of the DS port, OFDM channel, and OFDM Modulation Profile can all be updated using the Cisco Operations Hub. After the initial configuration during deployment using the Autodeployer, the configuration can be updated through the Cisco Operations Hub.

Complete the following steps to update the configuration:

Step 1 On the Cisco Operations Hub, click **Configurator** > **Export/Import** to view the Export/Import panel.

Step 2 In the `Export cnBR Configuration` pane, choose the Cisco cnBR core that needs to be updated.

Step 3 Click **Export** to retrieve the current SG configuration of the selected Cisco cnBR. The file is downloaded in JSON format. You can choose to update the following parameters in the downloaded JSON file.

- To update the OFDM Modulation Profile, edit the values in the `ofdmModProfs` section of the SG configuration.

- To update the DS port, edit the values in the `rpdCfg` section of the SG configuration.
- To update the OFDM channel, edit the values in the `ofdmDs` section of the SG configuration.

Step 4 Save the updated file.

Step 5 On the **cnBR Export/Import** panel > **Import cnBR Configuration File** pane, choose the Cisco cnBR name from the drop-down list.

Step 6 Click **Browse** to locate the saved configuration file.

Step 7 Click **Import** to push the updated SG configuration.

Downstream Modulation Profile Selection

Cisco cnBR has the following DS modulation profiles:

- Default Data Profile

  When a CM registers, it is assigned a default data profile. The default data profile is `profile-data 1`. If `profile-data 1` is not configured, `profile-control` is assigned to the CM.

- Recommended Profile

  The Cisco cnBR chooses a profile from existing configured modulation profiles having the highest speed and sufficient Signal to Noise Ratio (SNR) margin. The profile selection is based on the Receive Modulation Error Ratio (RxMER) values collected from a modem.

  This allows optimum use of the OFDM channel while allowing the modem to receive codewords with acceptable error rate. The selected profile is the *recommended profile* for that modem.

  To compute the recommended profile, the modem's RxMER values are first mapped to desired bit loading values. The desired bit loading values are compared to those in the configured profiles. Ideally, the desired bit loading value must be higher than that in the profile for the same subcarrier.

  However, due to the error correction capabilities provided by the channel coding and interleaving, this rule allows certain exceptions. The exemptions are made a configurable value, and is called *exempt subcarrier percentage*.

Recommended Profile Age

All recommended profiles have a configurable age that is associated with it. If the recommended profile exceeds this age, it is no longer valid for that modem.

RxMER to Bit Loading Mapping

There are various methods to map the Receive Modulation Error Ratio (RxMER) values to a modem's desired bit loading values. Cisco cnBR recommends the following mapping, which is listed in CM-SP-CCAP-OSSIv3.1, as the baseline mapping:

Table 14: RxMER to Bit Loading Values

RxMER (¼ DB)	QAM	Bit Loading
60	16	4

RxMER (¼ DB)	QAM	Bit Loading
84	64	6
96	128	7
108	256	8
136	1024	10
148	2048	11
164	4096	12
184	8192	13
208	16384	14

Margin Adjustment

A margin value may be configured for each cnBR to adjust the RxMER to the Bit loading mapping listed in the table. This configured value (in quarter-DB) is added to the RxMER values collected by cnBR before using the above mapping table. This gives you more control in selecting the recommended profiles.

Exempt Subcarrier Percentage

An exempt subcarrier percentage may be configured for each cnBR. When computing the recommended profile for a modem, this threshold percentage of subcarriers may be ignored when comparing the modem's desired bit loading values to those in each configured profile.

RxMER Poll Interval

cnBR uses OPT message with bit-0 option to collect RxMER data from CMs, after the initial modem registration and periodically thereafter. The collected RxMER data is used to compute the recommended profile for each modem.

Unfit Profile

The profile indicates that the CM-STATUS message is marked as *unfit profile* if the CMTS receives CM-STATUS Event 16 (DS OFDM Profile Failure).

A configurable maximum age is associated with each unfit profile for a given modem. If the unfit profile for a modem exceeds this age, it is no longer considered *Unfit* for that modem.

Profile Selection Parameter Configuration

The following table lists the parameter range for the profile selections:

Table 15: Parameter Ranges for Profile Selections

Profile Selection Parameter	Minimum	Maximum	Default
`rxmer-poll-interval`	1 minute	1440 minutes	60 minutes

Profile Selection Parameter	Minimum	Maximum	Default
exempt-sc-pct	1	100	2
mer-margin-qdb	0 qDB	40 qDB	0
recm-prof-age	1 minute	1440 minutes	120 minutes
unfit-prof-age	1 minute	1440 minutes	120 minutes

An example of the parameter configuration is as follows:

```
"ofdmProfMgmt":
{
    "rxmer-poll-interval": 180,
    "exempt-sc-pct": 20,
    "mer-margin-qdb": 16,
    "recm-prof-age": 360,
    "unfit-prof-age": 360
}
```

View OFDM Channel and Profile Statistics

You can choose to view the OFDM channel and profile statistics information on the Cisco cnBR dashboard.

You can view the OFDM channel and profile statistics through the Metrics dashboard. You can choose to view the following:

- Downstream Channel Statistics

  The DS channel (SC QAM and OFDM channel) byte and packet counters for a given SG are displayed on the Downstream Channel Rate dashboard of the Cisco Operations Hub.

  To load this dashboard, in the Cisco Operations Hub click **Metrics** and search for `DS Channel Rate`.

  The historical data of the downstream channel (SC QAM and OFDM channel) bit and packet rates for a given SG are also displayed on the same page, along with the historical data for the downstream channel (SC QAM and OFDM channel) utilizations.

- OFDM Modulation Profile Statistics

  The OFDM modulation per-channel-per-profile byte and packet counters are displayed in the OFDM Channel Profile Stats dashboard in Cisco Operations Hub.

  To load this dashboard, in the Cisco Operations Hub click **Metrics** and search for `OFDM Channel Profile Stats`.

- OFDM OCD and DPD Information

  The OFDM channel OCD and DPD configuration sent through MAC Management Message to CMs can be viewed on the OFDM Channel OCD and DPD Information dashboard.

To load this dashboard, in the Cisco Operations Hub click **Metrics** and search for `OFDM Channel OCD DPD Info`.

View DOCSIS 3.1 Modem Data

You can view the DOCSIS 3.1 modem data through the Cisco cnBR dashboard.

You can use the dashboard to view information on the following:

- **D3.1 Modem Information display**

  On the Cisco Operations Hub, click **Metrics** and search for `Cable Modem Verbose`.

  The `Modem Other Info` and `Modem OFDM Info` tables display information specific to the D3.1.

- **OFDM Profile Stats**

  On the Cisco Operations Hub, click **Metrics** and search for `CM OFDM Profile Stats`.

  The profile stats information from each D3.1 modem is available.

DEPI Latency Measurement

DEPI Latency Measurement (DLM) measures the delay and latency of the packets traversing through the Converged Interconnect Network (CIN) from Cisco cnBR to the RPD.

DLM configuration has three parameters: staticDelay, interval, and updateMap. Without any DLM configuration, the network delay uses 500 microseconds (μs) as default value for the calculation of Map Advance Time. When you configure staticDelay with nonzero value, it replaces the default network delay in Map Advance Time. When you configure the interval with nonzero value, DLM starts to send the packets from Cisco cnBR to RPD and calculate the downstream path CIN delay. You can use the CIN delay measurements from DLM to display or debug. When you set updateMap to true, and statiDelay configuration is absent or 0, you can also use the CIN delay measurements to replace the network delay time and adjust the DOCSIS MAP Advance Time. When the DLM is disabled, the network delay restores to the default value of 500 μs.

The DLM calculated delay is valid if it falls in the range between 30 μs and 100 ms. The valid DLM delay replaces the network delay when it is enabled. Subsequent ongoing update to the network delay only happens when the difference between the old and new value is larger than 75 μs. The following table summarizes how the map advance time can be affected based on the parameters in the table.

DLM staticDelay	DLM interval	DLM updateMap	DLM measuring CIN delay	Map Advance Network Delay
Absent or zero	Absent or zero	true or false	No	500 μs (default)
Nonzero	Absent or zero	true or false	No	staticDelay (configured)

DLM staticDelay	DLM interval	DLM updateMap	DLM measuring CIN delay	Map Advance Network Delay
Nonzero	Nonzero	true or false	Yes	staticDelay (configured)
Absent or zero	Nonzero	false (display only)	Yes	500 µs (default)
Absent or zero	Nonzero	true	Yes	DLM calculated delay

Configure DLM

DLM is configured in the Service Group configuration. Since DLM measures CIN delay to RPD, it is set per RPD.

Configure DLM using AutoDeployer script

In the AutoDeployer script SG template file, you can add `netDelayCfg` block to `rpdCfg` block to enable DLM. The SG template configuration applies to all the service groups on the Cisco cnBR. See Configure cnBR using Autodeployer, on page 32 for additional information.

```
"rpdCfg": {
    "rfTopology": {
        ......
    },
    "netDelayCfg": {
        "staticDelay": 1000,
        "dlmCfg": {
            "interval": 10,
            "updateMap": true
        }
    }
}
```

Update the DLM Configuration using AutoDeployer Reconfigure (Preferred)

After the initial DLM configuration during the deployment using the AutoDeployer, you can update the configuration by modifying the `netDelayCfg` block in the SG template and running the AutoDeployer configuration script again.

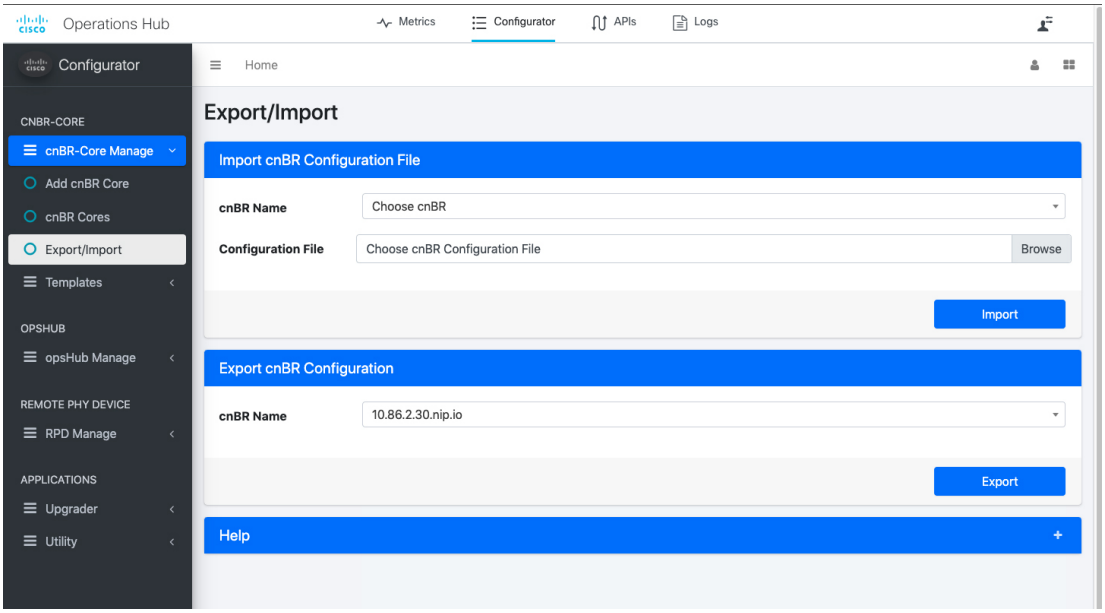
Note The system deletes all the RPDs/SGs and then adds them back when you rerun AutoDeployer configuration.

Update the DLM configuration using Cisco Operations Hub

After the initial DLM configuration during the deployment using the AutoDeployer, you can also update the configuration through the Cisco Operations Hub - Configurator Panel.

Step 1 Select the horizontal navigation tab Configurator, then click on **Export/Import** on the vertical navigation tab to get to the Export/Import page.

Step 2 Under the Export cnBR Configuration section, select the Cisco cnBR that manages the RPD to update from the drop-down list.



Step 3 Click the **Export** button to retrieve the current SG configuration of the selected Cisco cnBR.

Step 4 Update one or more parameters in the `netDelayCfg` section of the SG configuration to desired configuration.

Step 5 Save the update file on the local disk.

Step 6 To push the updated SG configuration, under the Import cnBR Configuration File section, select the Cisco cnBR that manages the RPD to update from the drop-down list.

Step 7 Click the **Browse** button to locate the file saved in step 4.

Step 8 Click the **Import** button to push the updated SG configuration to the RPD.

Step 9 Delete the RPD and add back the RPD for the updated SG configuration to take effect. See RPD Operations, on page 214 for additional information.

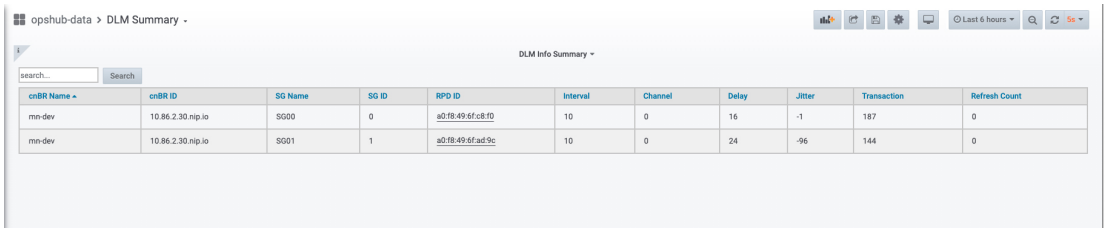
Configuration Parameters

Field Name	Description	Type	Enforcement
Interval	The interval of sending request packets to RPD and performing the delay calculation by DLM	Integer, 1 ~ 420, in second	Default is 0 and it means that DLM is disabled by default
UpdateMap	If the StaticDelay value is not set, determine if DLM calculated delay is used to update network delay portion of Map Advance.	Bool	Default is false and it means that DLM does not update Map Advance. Set to true, and clear the StaticDelay, for DLM to update Map Advance after DLM delay calculation

Field Name	Description	Type	Enforcement
StaticDelay	Use static delay to set the network delay portion of the MAP advance. If set, the dynamically calculated delay value is not used even if the UpdateMap flag is set to true.	Integer, 30 ~ 100000, in µs	Default is 0, it means no static delay to update map advance

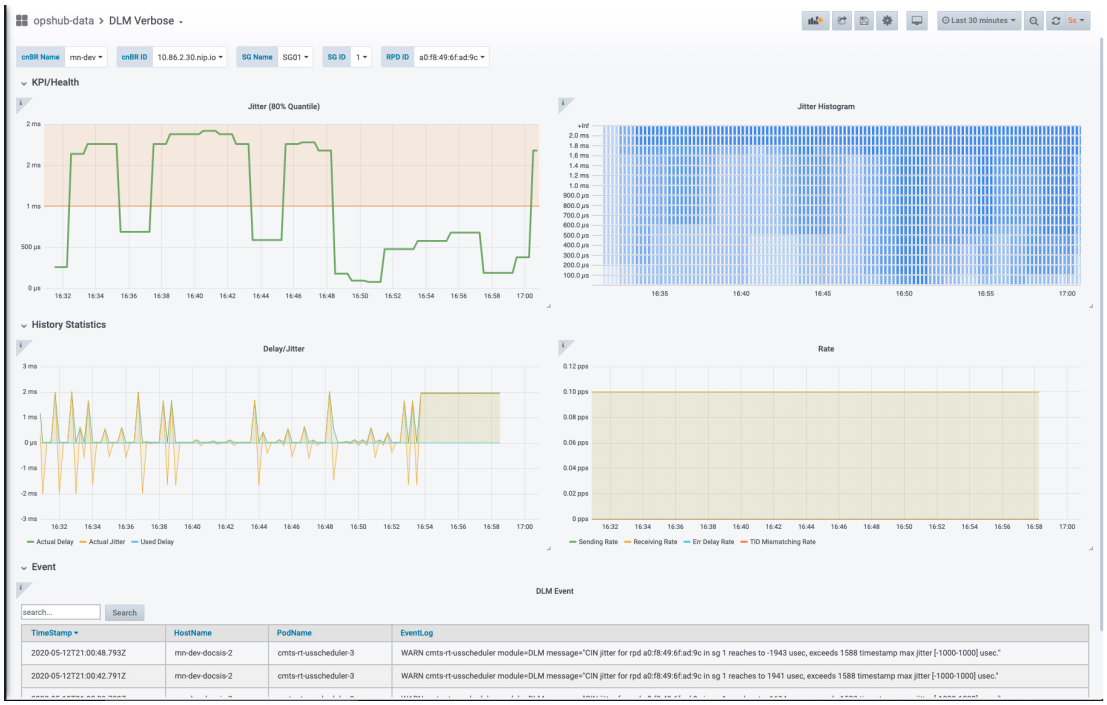
Monitor Information

You can find DLM summary and related plots in two DLM display panels in Cisco Operations Hub.



Field Name	Description	Type
cnBR Name	Cisco cnBR cluster name	Name string
cnBR ID	Cisco cnBR cluster address	IPv4/IPv6 address
SG Name	The name of the service group for the RPD	Name string
SG ID	The Service Group identifier.	Integer
RPD ID	The MAC address of the RPD. This RPD is part of the Service Group with the preceding SG ID.	MAC address
Interval	Configured DLM interval	Integer, in seconds
Channel	DS channel ID where DLM packet is sent	Integer, index
Delay	The most recent time delay calculated by DLM	Integer, in µs
Jitter	The most recent time jitter calculated by DLM	Integer, in µs
Transaction	The transaction ID of the most recent DLM request packet sent from cnBR	Integer, index
Refresh Count	The number of times the DLM updates Map Advance network delay	Integer, Counter

Click on RPD ID to enter the DLM verbose display panel.



- Jitter Health: Jitter graph and histogram are in the top of the DLM verbose display panel.
- Latency History Statistics
  - Delay/Jitter

Field Name	Description	Type
Actual Delay	The actual delay calculated by DLM over time	Integer, in µs
Actual Jitter	The actual jitter calculated by DLM over time	Integer, in µs
Used Delay	The average delay used to update map advance	Integer, in µs

  - Rate

Field Name	Description	Type
Sending Rate	Sending rate of the DLM request packets from cnBR	Rate, unit is pps.
Receiving Rate	Receiving rate of the DLM response packets from RPD	Rate, unit is pps.
Err Delay Rate	Receiving rate of the DLM response packets with abnormal timestamp from RPD	Rate, unit is pps.
TID Mismatching Rate	Receiving rate of the DLM response packets with abnormal transaction id from RPD	Rate, unit is pps.

- DLM Event: The warning events from DLM are listed in the bottom of the DLM verbose display panel.

Spectrum Surveillance

Spectrum surveillance provides radio frequency (RF) spectrum metrics for analyzing and detecting RF impairments. The Spectrum Surveillance feature includes:

- Per US CNIR: Spectrum surveillance periodically collects Fast Fourier Transformation (FFT) data from all active US to be used for the carrier-to-noise and interference ratio (CNiR) calculations.
- Per Cable Modem CNiR: Spectrum surveillance handles the cable modem (CM) burst based CNiR calculation.
- User-queried spectrum analysis: Spectrum surveillance handles requests for raw FFT spectrum analysis.

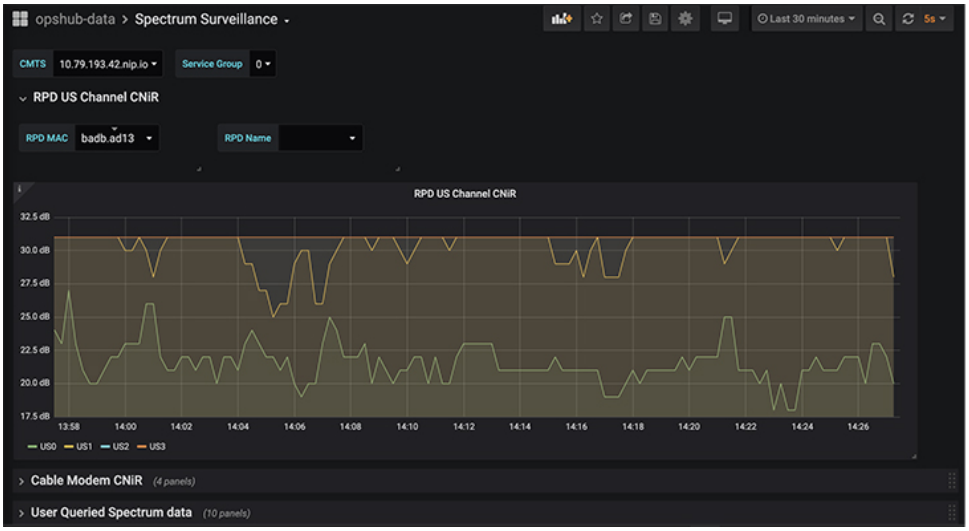
Configure Spectrum Surveillance

You can enable or disable spectrum surveillance for each service group. Spectrum surveillance is controlled by the **specsvl** option in the `models.Sg` of Operations Hub APIs.

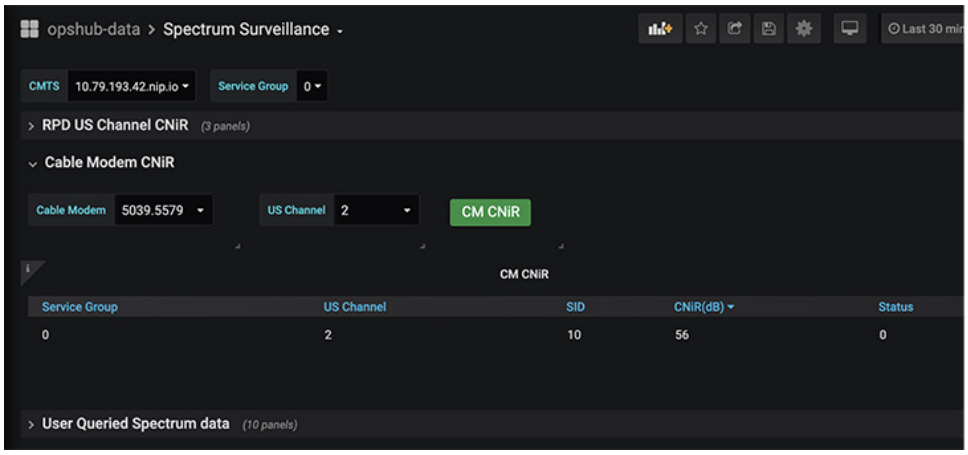
Monitor and Troubleshoot

The Dev Spectrum Surveillance panel of the Operations Hub Health Monitor Dashboard displays the spectrum surveillance metrics as shown in the following images.

View metrics on the RPD Upstream Channel CNiR:

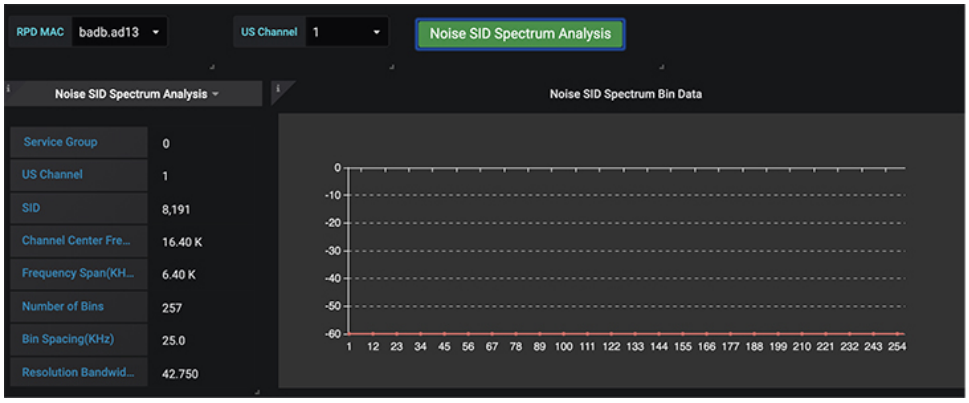


View information on the Cable Modem CNiR:



View User-Queried Spectrum Data for Primary SID:

View User-Queried Spectrum Data for Noise SID:



Spectrum Management Measurements

Carrier-to-Noise Ratio

The ratio of measured modulated power, in dB, on the upstream (before ingress noise cancellation) that compares the channel power to the noise power. The term carrier-to-noise plus interference ratio (CNiR) is part of the CableLabs nomenclature for the Carrier-to-Noise Ratio (CNR) measurement. Therefore, these two terms, CNR and CNiR, are used interchangeably.

The CNR (CNiR) measurement is usually provided only by an external spectrum analyzer, but the cable interface line cards that support intelligent and advanced hardware spectrum management features also provide CNR (CNiR) measurement.

The following two types of CNR (CNiR) measurements are supported on the Cisco CMTS:

- CNR (CNiR) measured for a particular upstream: The overall CNR (CNiR) for all the cable modems on an upstream, which is determined by measuring the RF power of the upstream receiver at the cable interface. This value is always merely a snapshot in time for a particular upstream. The cable interface measures the RF power at a time when no bursts are expected from the cable modems, but it can be skewed by a few cable modems that are experiencing or creating signal problems.
- Per-modem CNR (CNiR): The CNR (CNiR) for a particular cable modem, which is the signal strength of the burst transmissions of the modem at the upstream receiver of the cable interface. The per-modem CNR (CNiR) measurement is the most accurate measure of a particular cable modem signal. However, do not use a single modem's CNR (CNiR) to make assumptions about other cable modems on that upstream or about the upstream itself. You can analyze the upstream signal quality by polling the CNR (CNiR) for a number of cable modems over a representative time period.

Voice

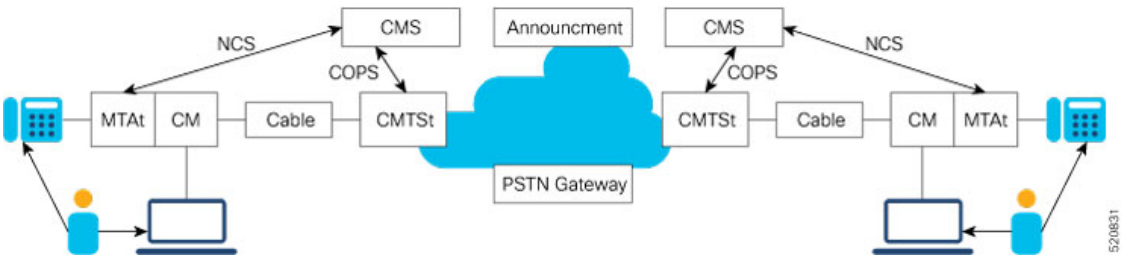
Cisco cnBR provides voice communication capabilities over cable networks.

Packetcable

Packetcable is a set of protocols developed to deliver Quality of Service (QoS) enhanced communications services using packetized data transmission technology to your home over the cable network.

Packetcable 1.5 is an enhanced version of packetcable protocols from Packetcable 1.0. The following figure shows the basic network topology.

Figure 21: Topology for Packetcable 1.5



Packetcable Configuration Parameters

Parameter	Values	Description	Default Value
pcEnable	True, False	True = Enabled, False = Disabled	True
pcMaxGate	Integer	Maximum gate number allowed in Cisco cnBR	51200
t0Timer	Integer in milliseconds	The period that an allocated gate exists without gate parameter set	30000
t1Timer	Integer in milliseconds	The period that an authorized gate exists without gate parameter set	200000
sendSubscriberEnable	True, False	If it is True, GateClose and GateSetAck message include Subscriber ID	False
copsAddrIp	IP address	IP address of CMS	None
copsGwIp	IP address	First hop gateway IP to CMS	None

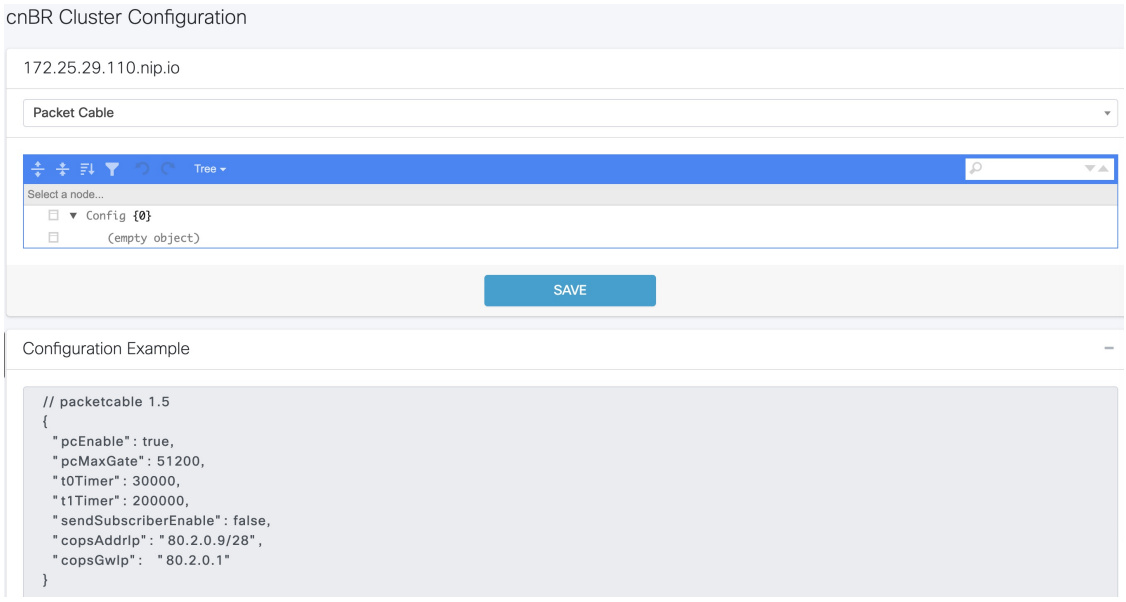
Packetcable 1.5 is enabled by default. The following configuration is used to disable the feature or change the timers. Usually the default configuration is sufficient. For further explanations of timer parameters, see DQoS1.5 SPEC.

You can configure the Packetcable 1.5 via Cisco cnBR Day 1 deployment yaml file.

```
packetcable :
    {enable:'true', 'max-gate':51200, 't0':30000, 't1':200000, 'subscriber':'false',
'ip':'5.230.205.10', 'gw':'5.230.205.1'}
```

You can also configure the Packetcable 1.5 via the Configurator as depicted in the following figure.

Figure 22: Configure Packetcable 1.5 using Cisco Operations Hub



The PC DQOS Enabled field in Cisco Operations Hub Voice Overview dashboard indicates whether the voice is enabled as shown in the following figure.

Figure 23: PC DQOS Enabled in Cisco Operations Hub

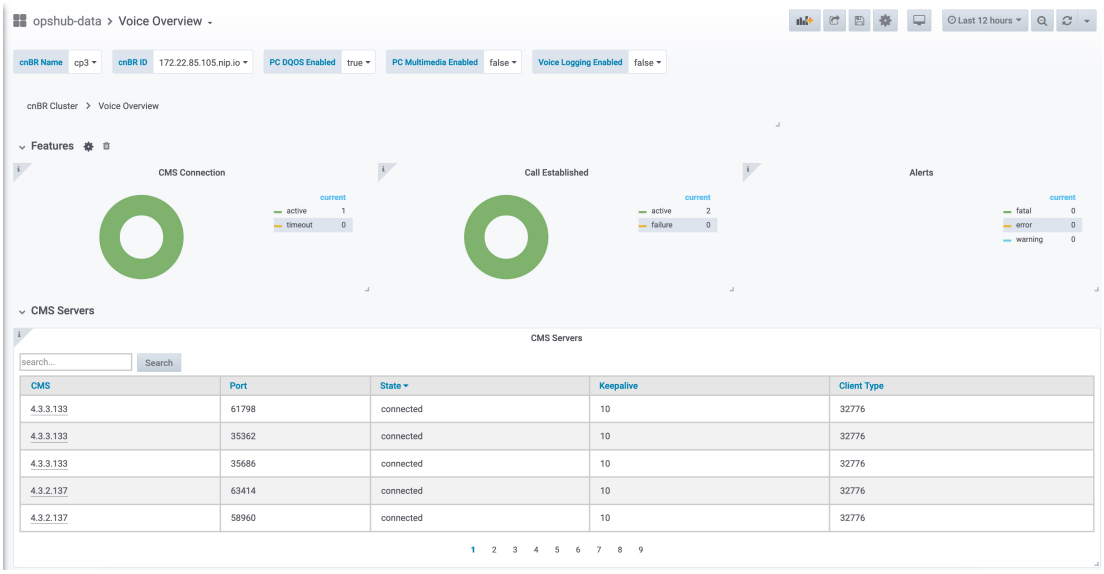
Cisco Operations Hub Voice Dashboard

The Cisco Operations Hub Voice Dashboards monitor Cisco cnBR Packetcable 1.5 voice features.

Voice Main page

As shown in the following figure, the first part of Voice Main page displays the Packetcable feature enable/disable status, COPS connection status, established call status, and the alerts that are reported by system.

Figure 24: Voice Main Dashboard Part 1



Detailed explanation for components in the preceding figure.

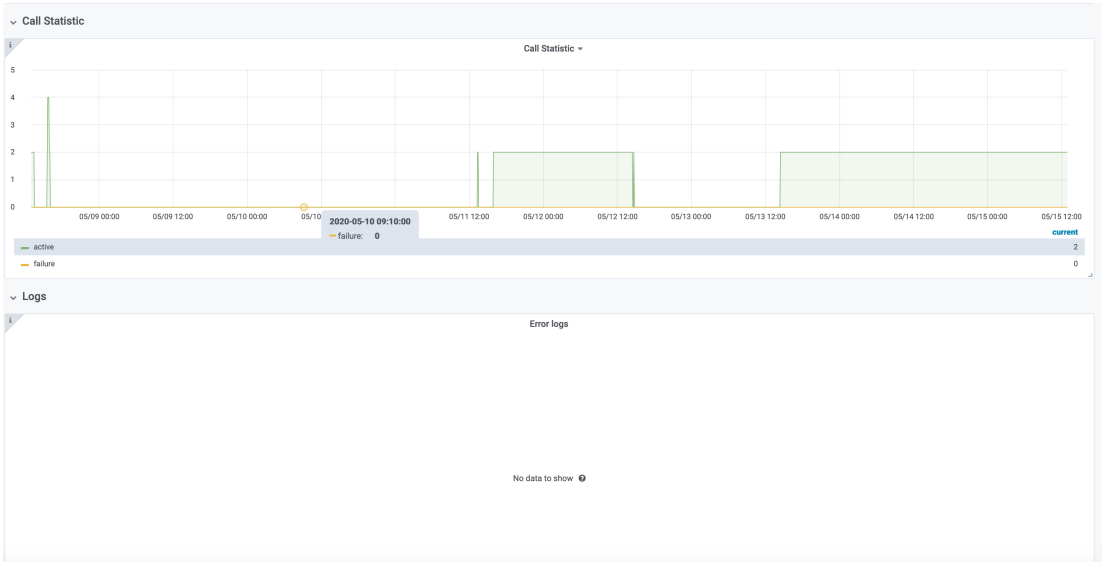
- Pie chart for CMS Connection.
  - active - The counter for CMS connections which are in active status.
  - timeout - The counter for CMS connections which are timeout.
- Pie chart for Call Established.
  - active - The counter for Established Calls which are in active status.
  - failure - The counter for Established Calls which are failure.
- Pie chart for Alert.
  - fatal - Fatal event counter.
  - error - Error event counter.

- warning - Warning event counter.

- Table for CMS Servers.
  - CMS - Server IP address.
  - Port - Server port.
  - State - Server connection states.
  - Keepalive - Keepalive timer between CMS and Cisco cnBR.
  - Client Type - The client type value (32776 for Packetcable and 32778 for Packetcable Multimedia).
  - You can use **Search...** text box to do fuzzy search in the entire table.

As shown in the following figure, the second part of Voice Main page displays overall call statistics and error logs reported by cmts-app-packetcable container in Cisco cnBR side.

Figure 25: Voice Main Dashboard Part 2



Detailed explanation for components in the preceding figure.

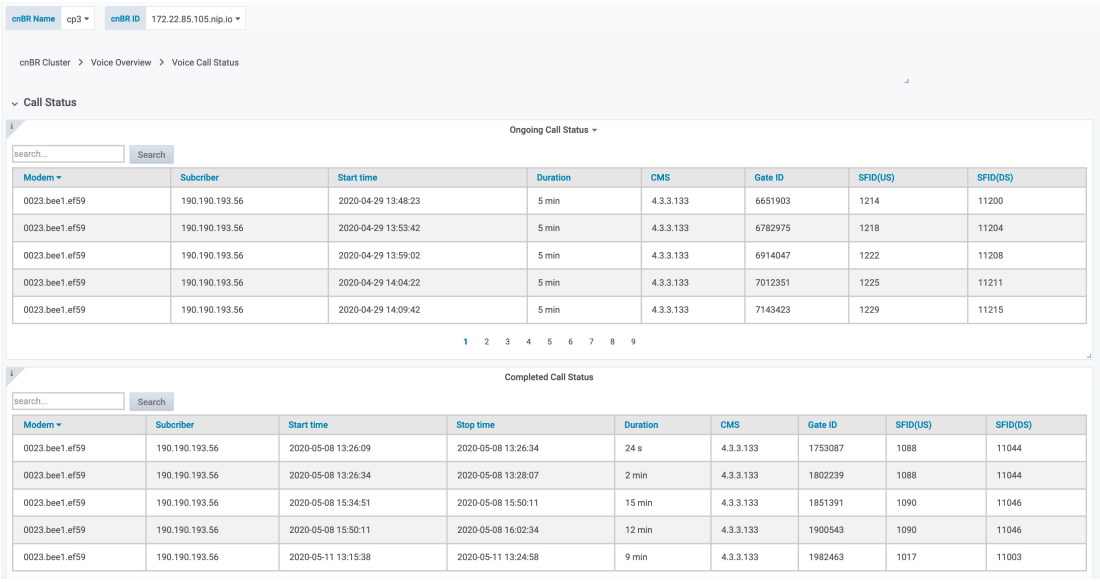
- Graph for Call Statistic
  - X-axis - Time.
  - Y-axis - Number of gates.

- Logs
  - Error messages from cmts-app-packetcable container.

Call Status Page

The Call Status page shows current and completed call status, as shown in the following figure.

Figure 26: Call Status Page



Detailed explanations for each column of tables in the preceding figure.

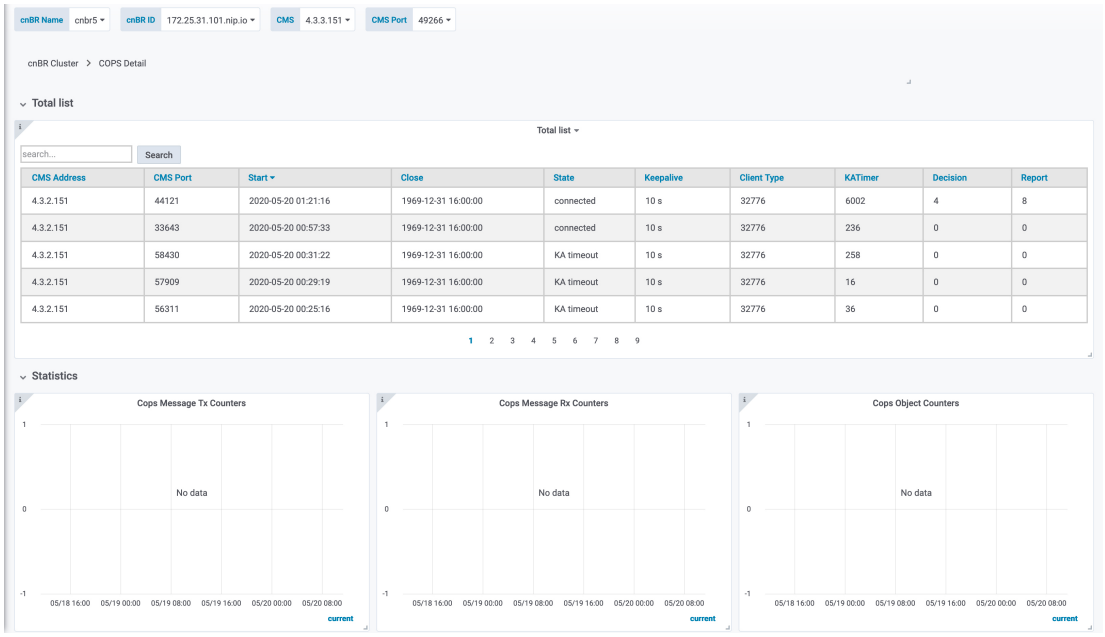
- Table for Ongoing Call Status
  - Modem - Modem MAC address.
  - Subscriber - Subscribe's MTA IP address.
  - Start time - The start time for the call.
  - Duration - Call duration.
  - CMS - Call Management Server IP address.
  - Gate ID - Gate identifier.
  - SFID(US) - Service flow ID for upstream.
  - SFID(DS) - Service flow ID for downstream.
- Table for Completed Call Status
  - Modem - Modem MAC address.
  - Subscriber - MTA IP address.
  - Start time - The start time for the call.
  - Stop time - The stop time for the call.
  - Duration - Call duration.
  - CMS - Call Management Server IP address.
  - Gate ID - Gate identifier.
  - SFID(US) - Service flow ID for upstream.

- SFID(DS) - Service flow ID for downstream.

COPS Status Page

The COPS Status page shows COPS connection status as shown in the following figure.

Figure 27: COPS Status Page



Detailed explanations for each table in the preceding figure.

- Table for Total list
  - CMS Address - Call Management Server IP address.
  - CMS Port - Port of the Call Management Server IP address.
  - Start - The start time for CMS connection.
  - Close - The close time for CMS connection.
  - State - The server connection states.
  - Keepalive - The keepalive time for CMS and Cisco cnBR.
  - Client Type - The client type (32776 for Packetcable and 32778 for Packetcable Multimedia).
  - KATimer - The counter for keepalive message.
  - Decision - The counter for COPS decision message.
  - Report - The counter for COPS report-type message.
  - You can use **Search...** text box to do fuzzy search in the entire table.
- COPS Message Tx Counters

- X-axis - Time.
- Y-axis - The counter for each type of COPS Tx Message.

- COPS Message Rx Counters
  - X-axis - Time.
  - Y-axis - The counter for each type of COPS Rx Message.

- COPS Object
  - X-axis - Time.
  - Y-axis - The counter for each type of COPS Object.

Service Flow Information

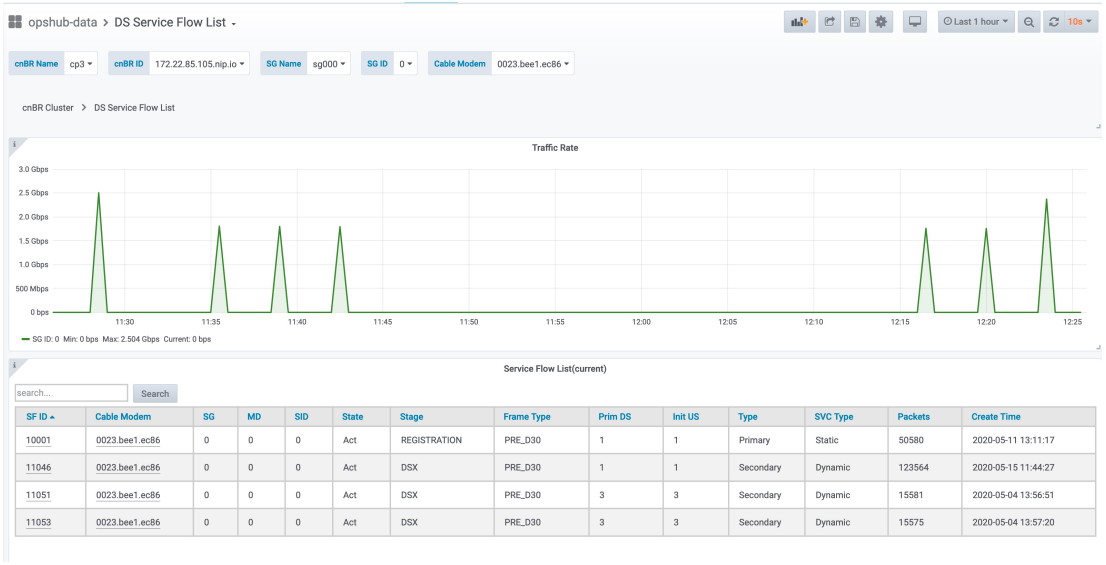
Four dynamic service flows are created to set up a voice path for each two-way call.

One upstream and one downstream service flow are created for each modem in the call.

You can find Service Flow Information for each modem in Downstream Service Flow List or Upstream Service Flow List dashboard.

Use Downstream Service Flow List as example in the following figure.

Figure 28: Service Flow List For Specific Modem



The downstream dynamic service flow created for voice call is listed under Service Flow List table.

Detailed explanations of each column in Downstream Service Flow List table in the preceding figure.

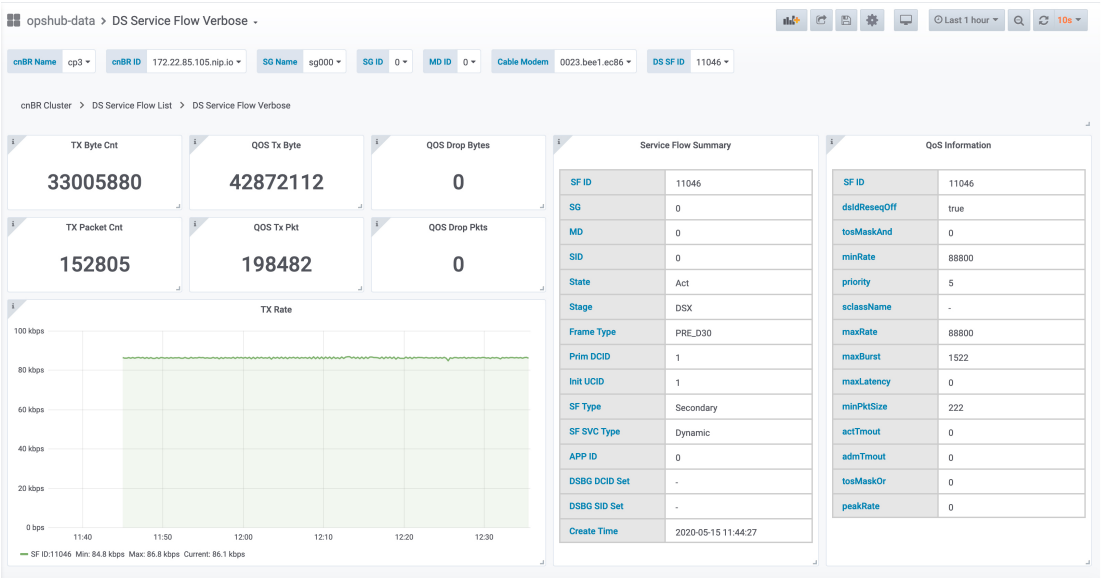
- Table for Downstream Service Flow List
  - SF ID - Service Flow ID.
  - Cable Modem - MAC Address of the modem.

- SG - Service Group of the modem.
- MD - MAC Domain of the modem.
- State - State of service flow [Prov, Adm, Act].
  - Prov - Service flow is in provision state.
  - Adm - Service flow is in admit state.
  - Active - Service flow is active state.
- Stage - Stage of service flow [PRE_REGISTRATION, REGISTRATION, DSX].
  - PRE_REGISTRATION - Service flow is provisioned before REGISTRATION.
  - REGISTRATION - Service flow is provisioned in REGISTRATION.
  - DSX - Service flow is dynamically provisioned for voice.
- Frame Type - [PRE_D30, CCF_ON, CCF_OFF].
  - PRE_D30 - Pre-3.0 DOCSIS concatenation and fragmentation.
  - CCF_ON - Continuous Concatenation and Fragmentation is enabled.
  - CCF_OFF - Continuous Concatenation and Fragmentation is disabled.
- Prim DS - Primary downstream channel ID.
- Init US - Init upstream channel ID.
- Type - [Primary, Secondary].
- SVC Type - [Dynamic, Static].
  - Dynamic - Service flow is dynamically provisioned.
  - Static - Service flow is statically provisioned.
- Packets - Number of packets.
- Create Timestamp - When the service flow created.

Clicking on the SFID of dynamic flow in above table to redirect to the Downstream Service Flow Verbose page.

The voice traffic throughput data is available in that page, as shown in the following figure.

Figure 29: Downstream Service Flow Verbose Page



520839

The TX Rate table in the preceding figure shows the downstream traffic throughout for voice.

Detailed explanations of relevant tables and counters in the preceding figure.

- Service Flow Traffic Rate
  - X-axis - Time
  - Y-axis - Throughput in kilobit per second
- TX byte cnt is the count of total bytes received by policer.
  - "TX Byte cnt" = "QOS Tx Byte" - "QOS Drop Bytes"
- TX packet cnt is the count of total packets received by policer.
  - "TX Packet Cnt" = "QOS Tx Pkt" - "QOS Drop Pkts"
- QOS TX byte is the count of total bytes sent to policer.
  - "QOS Tx Byte" = "TX Byte cnt" + "QOS Drop Bytes"
- QOS TX pkt is the count of total packets sent to policer.
  - "QOS Tx Pkt" = "TX Packet Cnt" + "QOS Drop Pkts"
- QOS drop bytes are the drop bytes count of policer, includes policer drops, queue full drops, and approximate Fair Drop drops.
  - "QOS Drop Bytes" = "QOS Tx Byte" - "TX Byte cnt"
- QOS drop pkts are the drop packets count of policer, includes policer drops, queue full drops, and approximate Fair Drop drops.
  - "QOS Drop Pkts" = "QOS Tx Pkt" - "TX Packet Cnt"

Traffic Management

Cisco cnBR provides traffic management functionalities to prevent data loss in important business applications, and to ensure that mission-critical applications take priority over other traffic.

DOCSIS Downstream QoS

DOCSIS downstream QoS consists of classifying packets into service flows for downstream and providing QoS at the service flow level.

Packet Classification

The packet classification supports the following packet header fields, as specified in the DOCSIS specification.

IPv4 fields:

- IPv4 TOS values
- IP protocol
- IP source address and mask
- IP destination address and mask

IPv6 fields:

- IPv6 traffic class values
- IPv6 flow label
- IPv6 next header type
- IPv6 source address and prefix length (bits)
- IPv6 destination address and prefix length (bits)

TCP or UDP fields:

- TCP/UDP source port start and end
- TCP/UDP destination port start and end

The packet classifiers are specified in cable modem configuration files. These configuration files are sent to Cisco cnBR either when registering the modem (for static service flows) or later through DSX messages (for dynamic service flows).

Downstream Service Flow

The basic unit of downstream QoS is the downstream service flow, which is a unidirectional sequence of packets transported across RF channels between Cisco cnBR and cable modems. The following parameters define the QoS of service flows in DOCSIS:

- Maximum sustained traffic rate
- Minimum sustained traffic rate

- Peak traffic rate
- DOCSIS traffic priority
- Maximum traffic burst size
- Maximum DS latency, used to indicate only the absolute priority

A service flow can be in one of the following three states:

- Provisioned
- Admitted
- Active

Only active flows are used to carry traffic and subject to the QoS treatment.

You can specify the service flow parameters directly in the individual modem configuration files or indirectly through the service classes on Cisco cnBR.

Service Class

Service providers can use service classes to manage QoS parameters. For example, the provider can add QoS parameters to each tier of service it offers in a service class. Use the service class names to match a modem's service flows to a service class, as defined by DOCSIS.

Downstream QoS Configuration

You can configure all packet classification parameters and the downstream service flow QoS parameters in the modem configuration files. If you want to use the service class feature, configure Cisco cnBR accordingly.

When you use a service class, the modem configuration files should have the service class names that match the ones configured in the service class.

Note QoS parameters for a service flow are decided when creating the service flow, either during modem registration or its dynamic creation.

Initial Configuration from Autodeployer Script

Configure service classes in the `svcds` block in the SG configuration `json` file. The following traffic parameters are supported.

Note The maximum values provided in the following table indicate the valid parameter range. Provide the actual parametric values that are based on the actual system capacity and traffic planning.

Parameter Name	Description	Minimum	Maximum	Unit
maxSustTrafRate	Maximum Sustained Traffic Rate	0	4G	bps

Parameter Name	Description	Minimum	Maximum	Unit
minRsvdTrafRate	Minimum Reserved Traffic Rate	0	4G	bps
peakTrafRate	Peak Traffic Rate	0	4G	bps
trafPrio	Traffic priority used to indicate traffic ratio under congestion	0	7	N/A
maxTrafBurst	Maximum traffic burst	1522	4G	Byte
maxDsLatcy	Indication for High Priority	0	>0	N/A
servClassName	Service Class Name	N/A	N/A	a string

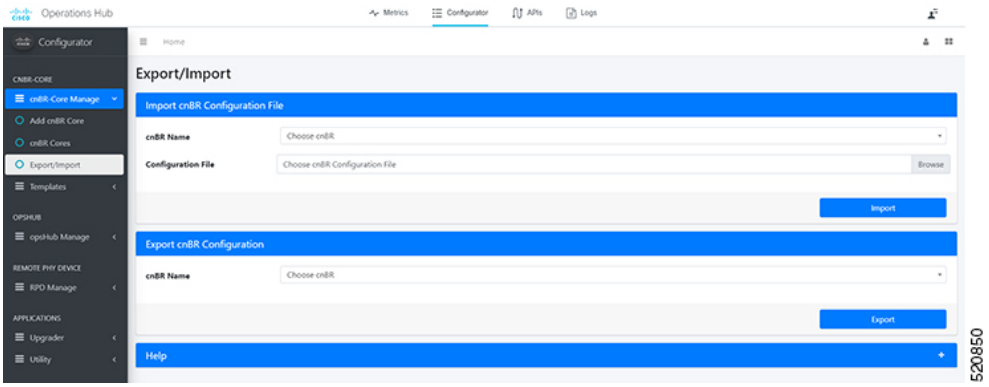
Example

```
"svcds": [
     {
         "maxSustTrafRate": 3000000,
         "servClassName": "DS_3M",
         "qoSParaSetType": 7
     },
     {
         "maxSustTrafRate": 4000000,
         "servClassName": "DS_4M",
         "qoSParaSetType": 7
     },
     {
         "maxSustTrafRate": 5000000,
         "servClassName": "DS_5M",
         "qoSParaSetType": 7
     },
     {
         "maxSustTrafRate": 10000000,
         "servClassName": "DS_MST_10M"
     },
     {
         "maxTrafBurst": 300000000,
         "servClassName": "DS_MTB_300M"
     },
     {
         "peakTrafRate": 12000000,
         "servClassName": "DS_PTR_12M"
     },
     {
         "minRsvdTrafRate": 2000000,
         "servClassName": "DS_CIR_2M"
     },
     {
         "maxSustTrafRate": 20000000,
         "maxTrafBurst": 200000000,
         "servClassName": "ds_level2_sf1"
     },
     {
```

```
        "maxSustTrafRate": 10000000,
        "peakTrafRate": 12000000,
        "servClassName": "ds_level2_sf2"
    },
    {
        "maxSustTrafRate": 15000000,
        "minRsvdTrafRate": 2000000,
        "servClassName": "ds_level2_sf3"
    },
    {
        "maxTrafBurst": 100000000,
        "peakTrafRate": 8000000,
        "servClassName": "ds_level2_sf4"
    },
    {
        "maxTrafBurst": 80000000,
        "minRsvdTrafRate": 26000000,
        "servClassName": "ds_level2_sf5"
    },
    {
        "minRsvdTrafRate": 26000000,
        "peakTrafRate": 12000000,
        "servClassName": "ds_level2_sf6"
    },
    {
        "maxSustTrafRate": 10000000,
        "maxTrafBurst": 100000000,
        "peakTrafRate": 26000000,
        "servClassName": "ds_level3_sf1"
    },
    {
        "maxSustTrafRate": 20000000,
        "maxTrafBurst": 300000000,
        "minRsvdTrafRate": 26000000,
        "servClassName": "ds_level3_sf2"
    },
    {
        "maxSustTrafRate": 25000000,
        "minRsvdTrafRate": 22000000,
        "peakTrafRate": 18000000,
        "servClassName": "ds_level3_sf3"
    },
    {
        "maxTrafBurst": 200000000,
        "minRsvdTrafRate": 3000000,
        "peakTrafRate": 26000000,
        "servClassName": "ds_level3_sf4"
    },
    {
        "maxSustTrafRate": 20000000,
        "maxTrafBurst": 300000000,
        "minRsvdTrafRate": 26000000,
        "peakTrafRate": 8000000,
        "servClassName": "ds_level4_sf"
    }
```


View Current Configuration using Cisco Operations Hub Configurator

Step 1 Choose **Cisco Operations Hub** > **Configurator** and click **Export/Import** on the left pane.



Step 2 Under the **Export cnBR Configuration** section, choose the Cisco cnBR router address from the drop-down list.

Step 3 Click **Export** to retrieve the current SG configuration of the selected Cisco cnBR.

A `.json` file containing the full configuration is saved to your machine. Service class settings are available in the `svcds` block.

Update Configuration

You can update the configuration using the following two methods:

- Operations Hub Configurator
- Autodeployer re-configuration

In both these options, the full configuration is sent to the CMTS. The existing configuration is overwritten and the new configuration is activated. For more details, see Autodeployer Limitations, on page 47.

Using Operations Hub Configurator

Step 1 Choose the **Configurator** tab, click **Export/Import** from the left pane.

Step 2 Under the **Export cnBR Configuration** section, choose the Cisco cnBR router address from the drop-down list.

Step 3 Click **Export** to retrieve the current SG configuration of the selected Cisco cnBR.

Step 4 Open the file and update the configuration in the `svcds` block of the SG configuration.

Step 5 Save the updated file on a local disk.

Step 6 Under the **Import cnBR Configuration File** section, select the Cisco cnBR address from the drop down list.

Step 7 Click **Browse** to locate the saved configuration file.

Step 8 Click **Import** to upload the updated SG configuration.

This updated file overwrites the existing configuration file and activates the new configuration.

Using Autodeployer Reconfiguration

After the initial configuration of the Source-Verify using the Autodeployer, update the configuration by modifying the appropriate blocks and rerunning the Autodeployer. This process overwrites the existing configuration and activates the new configuration.

For more details on the Autodeployer, see Configure cnBR using Autodeployer, on page 32.

Default Configuration

If the service class configuration does not exist, specify the service flow QoS parameters in the cable modem configuration file.

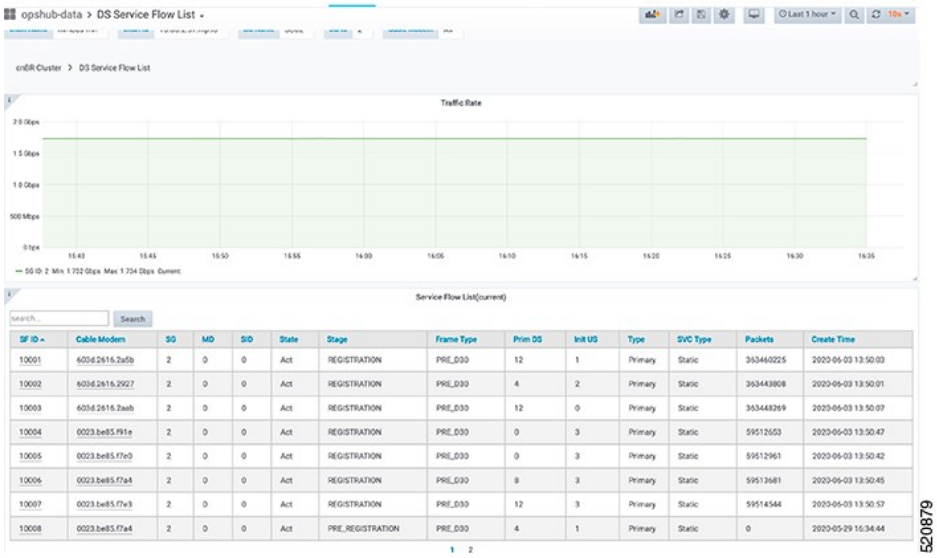
Downstream QoS Statistics

In **Cisco Operations Hub**, under **opshub-data** menu, you can see the following service flow details:

- Downstream Service Flow List
- Downstream Service Flow Verbose
- Downstream Service Flows for a Modem

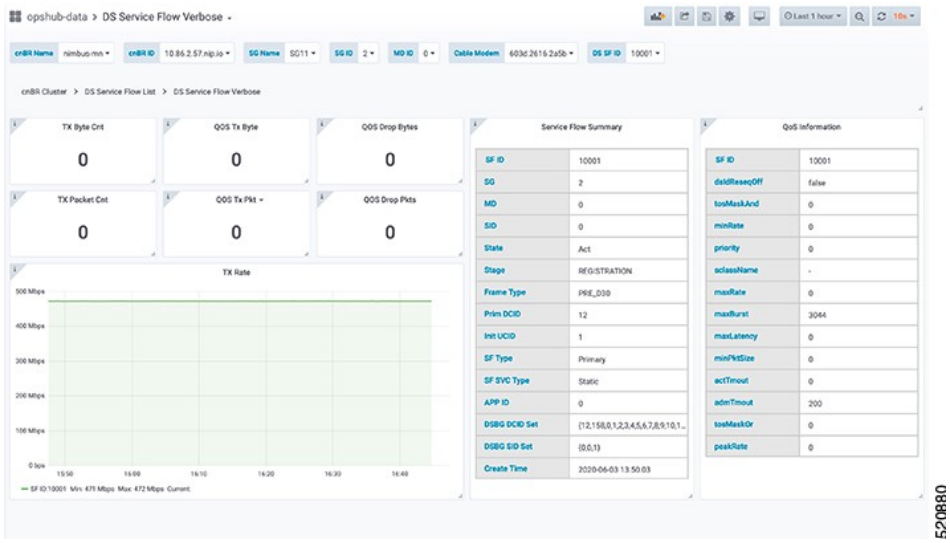
Downstream Service Flow List

The **Downstream Service Flow List** window provides the details of downstream service flows for each service group. The window displays a live graph of the traffic rate and a table listing all service flows of the selected service group.



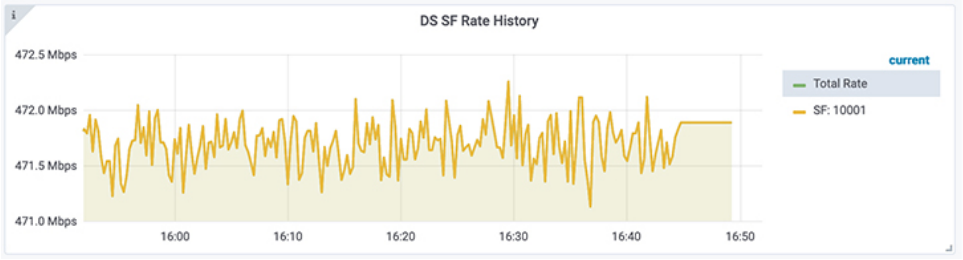
Downstream Service Flow Verbose

This window provides detailed information of an individual downstream service flow, including its transmission rate.



Downstream Service Flows for Cable Modem

The **Cable Modem Verbose** window provides the downstream service flow rate for all the flows on that modem.

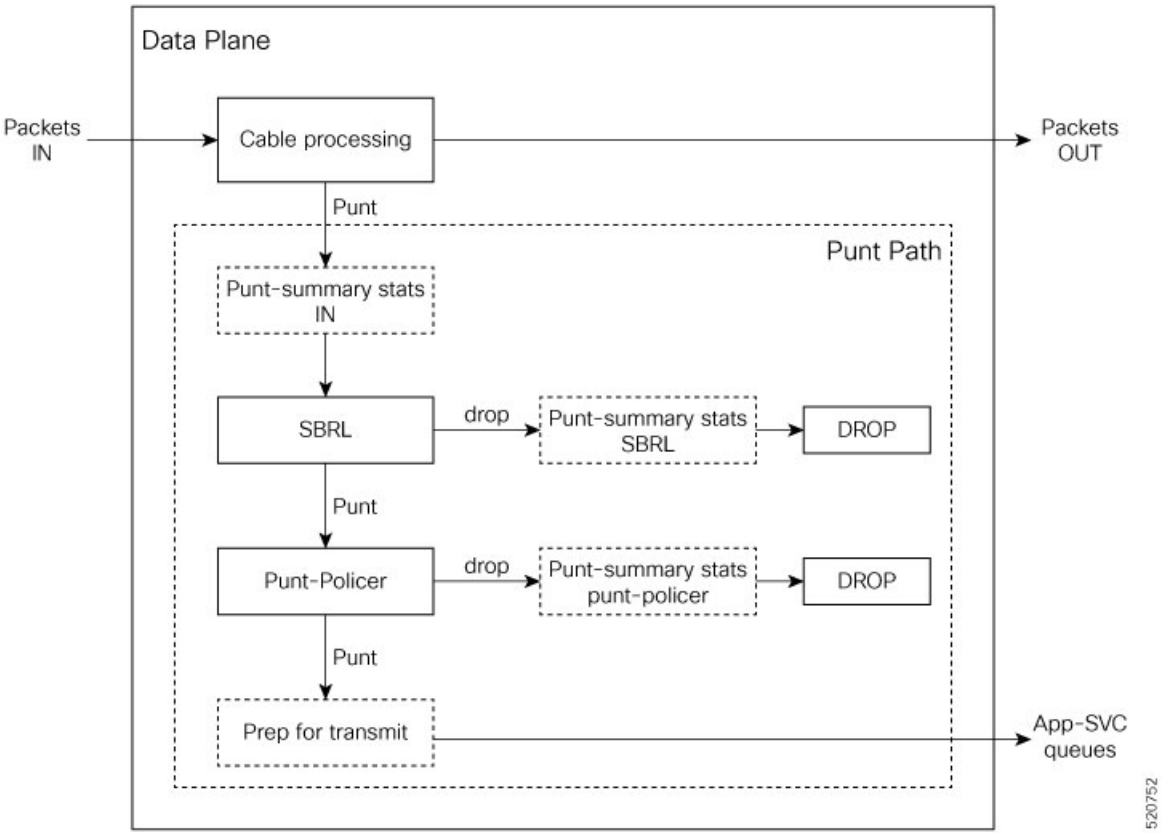


Punt Path Rate Limiting in Data Plane

The Cisco cnBR *punts* packets that the Data Plane (DP) cannot process to application services (for example, DHCP relay service) through **to-app-svc** queues. For example, ARP packets, DHCP packets, IP packets destined to unresolved adjacency, and so on.

The *DP punt-path* assigns a punt-cause to each punted packet, and prepares the packet for entry into **to-app-svc** queues.

Figure 30: Punt path rate-limiting



Denial of Service occurs when a service starts tail-dropping legitimate packets as a result of the queues becoming congested. To prevent this congestion, punt-path rate limiting (PPRL) operates in the punt-path to drop packets selectively. The Cisco cnBR identifies malicious actors and drops corresponding packets, while punting legitimate packets.

Cisco cnBR rate limiting operates on two levels:

- Source-Based Rate Limiting (SBRL) combines the subscriber MAC-address and the punt-cause to create an index for rate-limiting.
- Punt-Policer uses the punt-cause as the index for rate-limiting.

SBRL operates first. The Cisco cnBR combines MAC-address and punt-cause to create an index for rate-limiting. The Cisco cnBR rate-limits this MAC/punt stream according to the configured rate. The Cisco cnBR drops nonconforming packets. SBRL uses the source MAC address in the upstream direction and the destination MAC address in the downstream direction.

Next, the Punt-Policer aggregates packets with the same punt-cause, and rate-limits each punt-cause according to the configured rate. The Cisco cnBR drops nonconforming packets.

The following table lists the supported punt-causes:

Cause Id	Cause Name	Cause Description
6	dhcpv4_us	DHCP IPv4 upstream
14	dhcpv6_us	DHCP IPv6 upstream
10	cable_arp	ARP request and reply
11	ndp	Neighbor discovery protocol
20	svfy_v4	Source-verify IPv4
21	svfy_v6	Source-verify IPv6
22	ds_lq_v4	Lease query downstream IPv4
23	ds_lq_v6	Lease query downstream IPv6
25	mobility_v4	IPv4 CPE mobility
26	mobility_v6	IPv6 CPE mobility
7	tftp_req	TFTP request
32	ds_no_adj_v4	No adjacency downstream IPv4
33	ds_no_adj_v6	No adjacency downstream IPv6

Configuration

Both SBRL and Punt-Policer configurations are on a per-punt-cause basis.

Initial Configuration From Autodeployer Script

In the Autodeployer script SG template file, the PPRL configuration is in the *punt* block. Configure SBRL using the *subMacAddrSbrlList* block. Configure Punt-Policer using the *icpiPerCausePuntCfgList* block.

```
"sgs": [
    ...
    "sg-config": {
        ...
        "punt": {
            "subMacAddrSbrlList": [
              {
```

```
                    "PuntCause":cable_arp,
                    "RateLimitCfg": {
                   "RatePer4Sec":1000,
                 "BurstTimeMs":7000
              }
                   },
                   {
                    "PuntCause":ndp,
                    "RateLimitCfg": {
                 "RatePer4Sec":6000,
                 "BurstTimeMs":6000
              }
            }
          ]
              "icpiPerCausePuntCfgList": [
                {
                  "CauseId": 20,
                  "icpiPerCausePuntCfg": {
                      "MaxRate": 20
                  }
                },
                {
                  "CauseId": 21,
                  "icpiPerCausePuntCfg": {
                     "MaxRate": 20
                  }
                },
                {
                  "CauseId": 22,
                  "icpiPerCausePuntCfg": {
                      "MaxRate": 20
                  }
                },
                {
                  "CauseId": 23,
                  "icpiPerCausePuntCfg": {
                      "MaxRate": 20
                  }
                }
              ]
          }
          ...
      }
]
```


View Current Configuration Using Cisco Operations Hub Configurator

Step 1 Log into the Cisco Operations Hub.

Step 2 Click **Configurator** from the horizontal navigation tab.

Step 3 Click **Export/Import** under **cnBR-Core Manage** from the vertical navigation tab to access the **Export/Import** page.

Step 4 In the **Export cnBR Configuration** section, select the target Cisco cnBR from the drop down list.

Step 5 Click **Export** to retrieve the SG configuration of the selected Cisco cnBR.

A .json file containing the full configuration is saved to your machine. PPRL settings are available in the *punt* block.

Update Configuration

You can update the configuration using the following methods:

- Cisco Operations Hub Configurator
- Autodeployer reconfiguration

Both options send the full configuration to the CMTS. The Cisco cnBR overwrites the existing configuration and activates the new configuration. For more details, see Autodeployer Limitations, on page 47.

Update Configuration Using Cisco Operations Hub Configurator

Step 1 Log into the Cisco Operations Hub.
Step 2 Click **Configurator** from the horizontal navigation tab.
Step 3 Click **Export/Import** under **cnBR-Core Manage** from the vertical navigation tab to access the **Export/Import** page.
Step 4 In the **Export cnBR Configuration** section, select the target Cisco cnBR from the drop down list.
Step 5 Click **Export** to retrieve the SG configuration of the selected Cisco cnBR.
Step 6 Update the configuration in the *punt* block of the SG configuration and save the file.
Step 7 In the **Import cnBR Configuration File** section, select the target Cisco cnBR from the drop down list.
Step 8 Click **Browse** and select the saved configuration file.
Step 9 Click **Import** to push the updated SG configuration.

This import overwrites the existing configuration and activates the new configuration.

Update Configuration Using Autodeployer Reconfiguration

After the initial configuration of SBRL and Punt-Policer using the Autodeployer, update the configuration by modifying the corresponding blocks in the Autodeployer script and rerunning the Autodeployer. This process overwrites the existing configuration and activates the new configuration.

Configuration Parameters

Table 16: SBRL Configuration Parameters

Field Name	Description	Type	Units	Value	Enforcement
PuntCause	Punt cause ID to be rate limited	string	—	dhcpv4_us, dhcpv6_us, cable_arp, ndp, svfy_v4, svfy_v6, ds_lq_v4, ds_lq_v6, mobility_v4, mobility_v6, tftp_req, ds_no_adj_v4, ds_no_adj_v6	Required

Field Name	Description	Type	Units	Value	Enforcement
RatePer4Sec	Max rate in pkts-per-4-sec	integer	pkts-per-4-sec	1-255	Required
BurstTimeMs	For burst packets handling	integer	microseconds	1000-8000	Optional

Table 17: Punt-Policer Configuration Parameters

Field Name	Description	Type	Units	Value	Enforcement
CauseId	Punt cause ID to be rate limited	integer	—	6, 14, 10, 11, 20-23, 25, 26, 7, 32, 33	Required
MaxRate	Max rate in pkts-per-sec	integer	pkts-per-sec	10-300000	Required

Default Configuration

Table 18: SBRL Default Configuration

PuntCause	RatePer4Sec(pkts/4-sec)	BurstTime(msec)
dhcpv4_us	16	4000
dhcpv6_us	16	4000
cable_arp	16	4000
ndp	16	4000
svfy_v4	4	4000
svfy_v6	4	4000
ds_lq_v4	4	4000
ds_lq_v6	4	4000
mobility_v4	16	4000
mobility_v6	16	4000
tftp_req	16	4000
ds_no_adj_v4	4	4000
ds_no_adj_v6	4	4000

Table 19: Punt-Policer Default Configuration

CauseId	Cause Description	MaxRate(pkts/sec)
6	DHCP IPv4 upstream	1200
14	DHCP IPv6 upstream	1200

CauseId	Cause Description	MaxRate(pkts/sec)
10	ARP request and reply	1200
11	Neighbor Discovery Protocol	1200
20	Source-verify IPv4	1200
21	Source-verify IPv6	1200
22	Lease query downstream IPv4	400
23	Lease query downstream IPv6	400
25	IPv4 CPE mobility	1200
26	IPv6 CPE mobility	1200
7	TFTP request	1200
32	No adjacency downstream IPv4	400
33	No adjacency downstream IPv6	400

Monitoring

In the Cisco Operations Hub Metrics interface, Punt Inject Stats panel contains the PPRL statistics. Overall punt statistics are also available, along with SBRL and Punt-Policer statistics.

Figure 31: Overall Punt Statistics

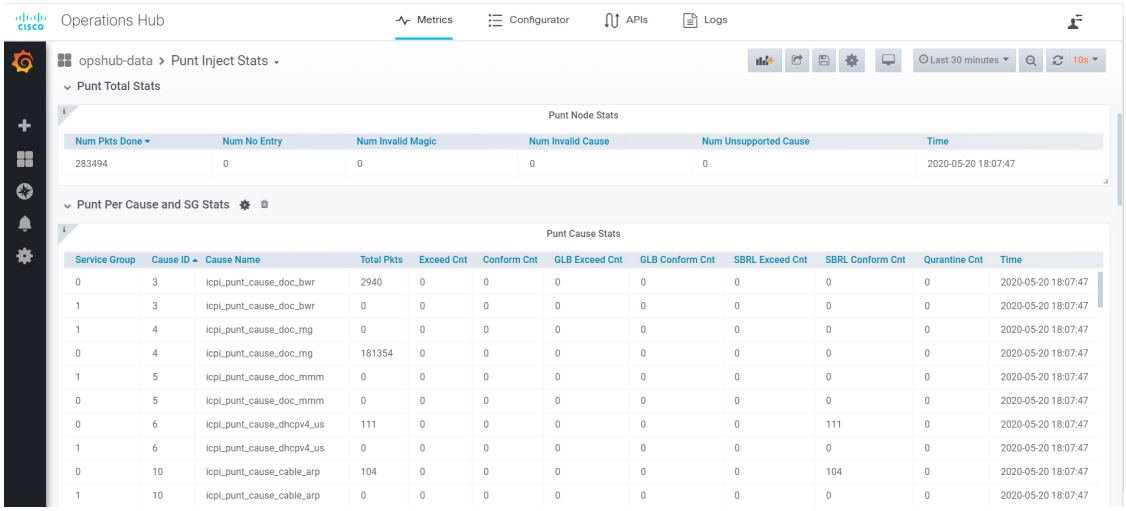


Figure 32: SBRL Statistics

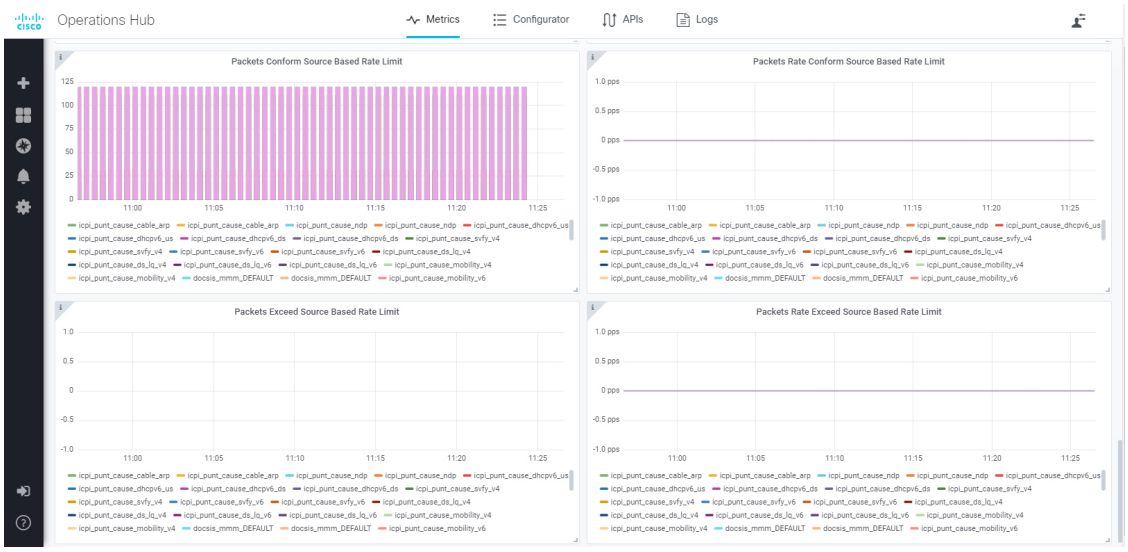
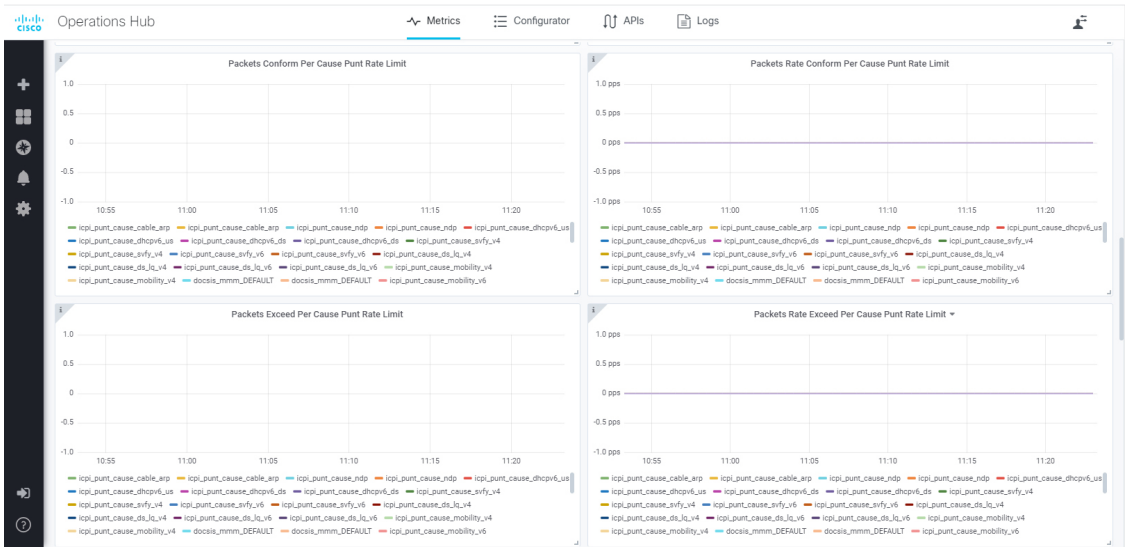


Figure 33: Punt-Policer Statistics



Upstream Type-of-Service (ToS) Overwrite

The Cisco cnBR can overwrite the DSCP/ToS field of packets associated with the DOCSIS Service Flow.

Configuration

Currently, you can configure ToS Overwrite through only the DOCSIS configuration file.

DOCSIS Configuration File

The DOCSIS service flow parameter *IP Type of Service (DSCP) Overwrite* contains two bytes, one for the **tos-and-mask** and one for the **tos-or-mask**. According to DOCSIS requirements, when you configure a

Service Flow with an *IP Type of Service (DSCP) Overwrite* parameter, the CMTS overwrites the DSCP/ToS value in the IP packets as follows:

```
new-ip-tos = ((orig-ip-tos AND tos-and-mask) OR tos-or-mask)
```

DOCSIS cable-modem configuration file uses *IP Type of Service Flow* under *Upstream Service Flow Encodings* to configure the upstream service flow parameter *IP Type of Service (DSCP) Overwrite*.

SubType	Length	Value
23	2	[and-mask, or-mask]

A configuration example is following:

```
24 (Upstream Service Flow Encoding)
   S01 (Service Flow Reference)         = 4
    S06 (QoS Parameter Set Type)         = 7
    S023 (IpTosOverwrite)                = 00 FF
```

More information on the DOCSIS parameters is available in DOCSIS 3.0 MAC and Upper Layer Protocols Interface Specification.

Default Configuration

By default ToS Overwrite is disabled; so the Cisco cnBR does not overwrite the DSCP/ToS field in the packet.

Enabling Security

Cisco cnBR provides security functionalities to defend against outside attacks.

Packet Filtering

Packet Filtering provides the ability to configure device-specific filters in the upstream and downstream directions.

- Devices are assigned with upstream and downstream filter groups via the DOCSIS configuration file.
- Different groups can be assigned for the upstream and downstream directions.
- If no filter group is specified in the DOCSIS configuration file, devices receive the default group configured on Cisco cnBR.
- If no default filter group is specified on Cisco cnBR, then no filtering is applied and the default action is FORWARD.

The rules for filter groups are configured on Cisco cnBR. Matching rules and actions (FORWARD or DROP) are specified in priority order. Rules are based on layer 2, layer 3, and layer 4 packet fields.

By default, Packet Filtering is disabled.

Configure Packet Filtering

Note Cable modems use the settings that are active during CM registration. If the default Packet Filtering groups are changed, you must reset cable modems to use the updated settings.

Initial Configuration using AutoDeployer Script

- In the Optional Configuration section of Configure cnBR using Autodeployer, on page 32, Packet Filtering configuration is in the `pfgActive` and `pfgGroup` blocks.
- Default Packet Filtering groups are specified in the `pfgActive` block.
- Rules for the groups are specified in the `pfgGroup` block.

Following is a sample configuration along with some explanation.

- The default filter group for downstream packets to a cable modem (cm_ds) is Group 10.
- Group 1 defines a filter that permits 90.90.90.2 ICMP packets, while denying other 90.90.90.0/24 ICMP packets. Groups 1 and 2 are not default groups. Therefore assign devices to these groups via the DOCSIS configuration file.

```
"global": {
    ...
    "pfgActive": {
        "cm_ds"  : 10,
        "cm_us"  : 11,
        "host_ds": 20,
        "host_us": 21,
        "mta_ds" : 30,
        "mta_us" : 31,
        "ps_ds"  : 40,
        "ps_us"  : 41,
        "stb_ds" : 50,
        "stb_us" : 51
    },
    "pfgGroup": {
        "grpList": [
          {
            "id" : 1,
            "ruleList": [
              {
                "isPermit": 1,
                "isIpv6": 0,
                "srcIp": "0.0.0.0",
                "srcIpPrefixLen": 0,
                "dstIp": "90.90.90.2",
                "dstIpPrefixLen": 32,
                "proto": 1,
                "srcportOrIcmptypeFirst": 0,
                "srcportOrIcmptypeLast": 65535,
                "dstportOrIcmptypeFirst": 0,
                "dstportOrIcmptypeLast": 65535,
                "tcpFlagsMask": 0,
                "tcpFlagsValue": 0,
                "tosMask": 0,
                "tosValue": 0
              },
```

```
                {
                  "isPermit": 0,
                  "isIpv6": 0,
                  "srcIp": "0.0.0.0",
                  "srcIpPrefixLen": 0,
                  "dstIp": "90.90.90.0",
                  "dstIpPrefixLen": 24,
                  "proto": 1,
                  "srcportOrIcmptypeFirst": 0,
                  "srcportOrIcmptypeLast": 65535,
                  "dstportOrIcmptypeFirst": 0,
                  "dstportOrIcmptypeLast": 65535,
                  "tcpFlagsMask": 0,
                  "tcpFlagsValue": 0,
                  "tosMask": 0,
                  "tosValue": 0
                }
              ],
            },
            {
              "id" : 2,
              "ruleList": [
                {
                  ...
                },
                ...
                {
                  ...
                }
              ],
            },
            {
              "id" : 10,
              "ruleList": [
                {
                  ...
                },
                ...
                {
                  ...
                }
              ],
            },
            ...

            {
              "id" : 51
              "ruleList": [
                {
                  ...
                },
                ...
                {
                  ...
                }
              ]
            }
          ]
      },

      ...

},
...
```

Display Current Configuration using Cisco Operations Hub Configurator

Step 1 Go to horizontal navigation tab Configurator in Cisco Operations Hub.

Step 2 Navigate to **cnBR-Core Manage** > **cnBR Cores**.

Step 3 Click on Cisco cnBR name in the table to open the Cisco cnBR configuration.

Step 4 Click on drop-down menu and select **PFG Active** or **PFG Group** to display the corresponding configuration.

Figure 34: OpsHub Configurator pfgActive config

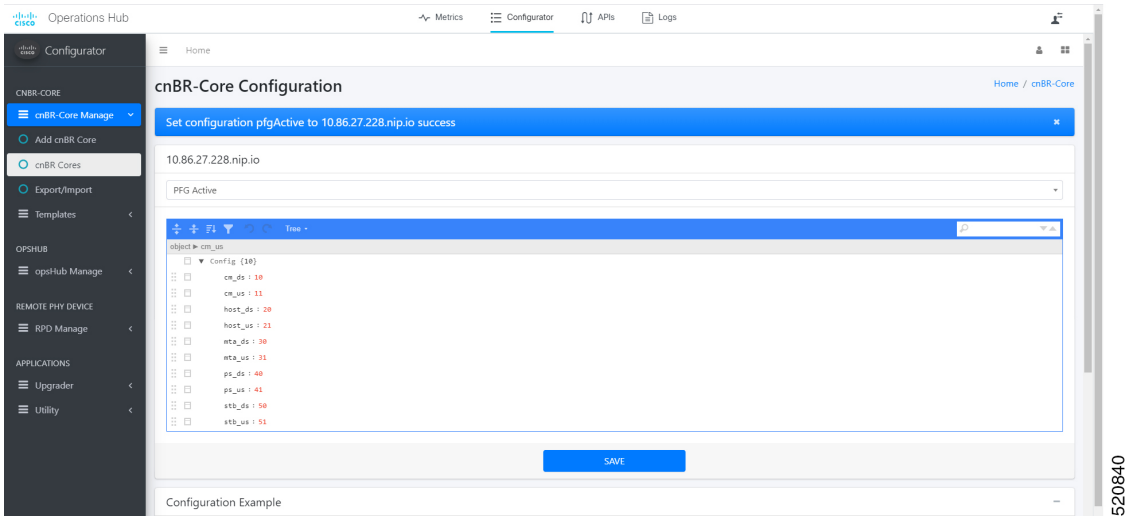
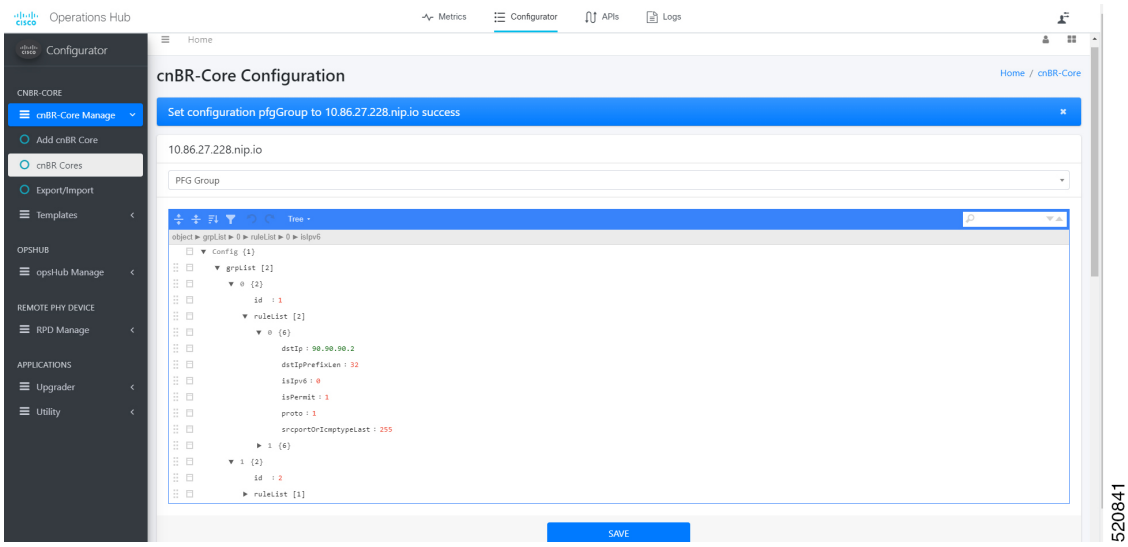


Figure 35: OpsHub Configurator pfgGroup config



Update Configuration using Cisco Operations Hub Configurator

Step 1 Go to horizontal navigation tab Configurator in Cisco Operations Hub.

Step 2 Navigate to **cnBR-Core Manage** > **cnBR Cores**.

Step 3 Click on Cisco cnBR name in the table to open the Cisco cnBR configuration.

Step 4 Click on drop-down menu and select **PFG Active** or **PFG Group** to display the corresponding configuration.

Figure 36: OpsHub Configurator pfgActive config

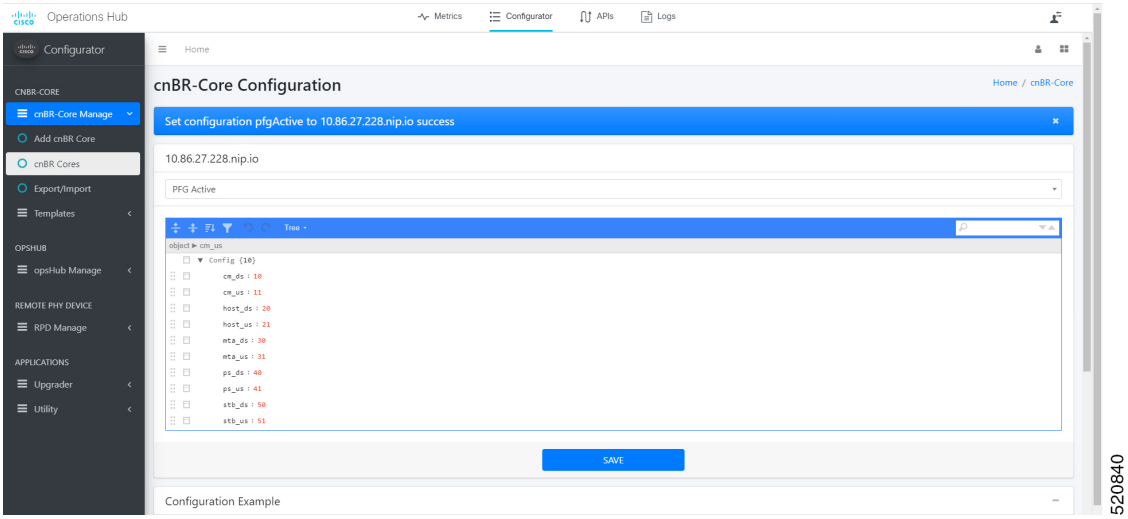
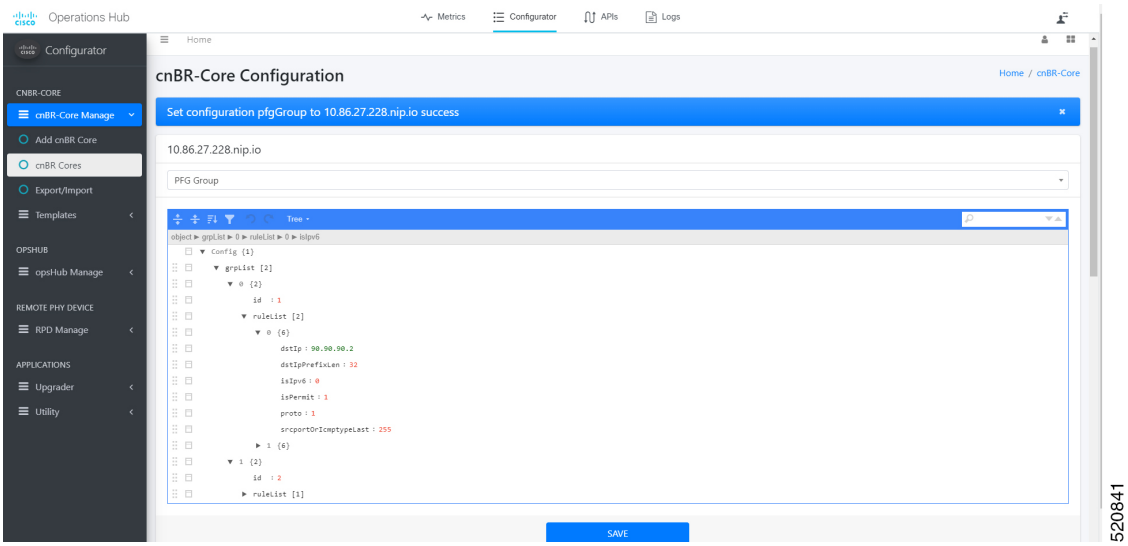


Figure 37: OpsHub Configurator pfgGroup config



Step 5 Modify the configuration.

Step 6 Click **SAVE** to push the updated configuration to the Cisco cnBR.

Update Configuration using Autodeployer Reconfiguration

After the initial configuration of Packet Filtering following the Configure cnBR using Autodeployer, on page 32, you can update the configuration by modifying the appropriate blocks and rerunning the AutoDeployer.

It fully overwrites the existing configuration and activates the new configuration. See Autodeployer Limitations, on page 47.

Configuration Parameters

- A group can have multiple rules. Rules are processed in the listed order.
- If a packet matches a rule, the specified action is performed and filtering is complete.
- If a packet does not match any rule in the group, the packet is forwarded.

Table 20: PFG Active: Default Packet Filtering Groups

Field Name	Description	Type	Range	Enforcement
cm_ds	Cable Modem downstream default group	integer	-1 means no group, otherwise [1, 254]	required
cm_us	Cable Modem upstream default group	integer	-1 means no group, otherwise [1, 254]	required
host_ds	Host (ie. CPE) downstream default group	integer	-1 means no group, otherwise [1, 254]	required
host_us	Host (ie. CPE) upstream default group	integer	-1 means no group, otherwise [1, 254]	required
mta_ds	Multimedia Terminal Adaptor downstream default group	integer	-1 means no group, otherwise [1, 254]	required
mta_us	Multimedia Terminal Adaptor upstream default group	integer	-1 means no group, otherwise [1, 254]	required
ps_ds	Portal Server downstream default group	integer	-1 means no group, otherwise [1, 254]	required
ps_us	Portal Server upstream default group	integer	-1 means no group, otherwise [1, 254]	required
stb_ds	Set-Top Box downstream default group	integer	-1 means no group, otherwise [1, 254]	required
stb_us	Set-Top Box upstream default group	integer	-1 means no group, otherwise [1, 254]	required

Table 21: PFG Group: Rule Definition

Field Name	Description	Type	Enforcement
isPermit	0 means deny, 1 means permit	Integer	required
isIpv6	0 means IPv4, 1 means IPv6	Integer	required
srcIp	Source IP value	IPv4 or IPv6	required
srcIpPrefixLen	Source IP prefix length	Integer	required

Field Name	Description	Type	Enforcement
dstIp	Destination IP value	IPv4 or IPv6	required
dstIpPrefixLen	Destination IP prefix length	Integer	required
tosValue	ToS/traffic class value	Integer	required
tosMask	ToS/traffic class mask	Integer	required
proto	Layer 4 protocol	Integer	required
srcportOrIcmptypeFirst	Start of source port or ICMP4/6 type range	Integer	required
srcportOrIcmptypeLast	End of source port or ICMP4/6 type range	Integer	required
dstportOrIcmpcodeFirst	Start of destination port or ICMP4/6 code range	Integer	required
dstportOrIcmpcodeLast	End of destination port or ICMP4/6 code range	Integer	required
tcpFlagsValue	TCP flags value	Integer	required
tcpFlagsMask	TCP flags mask	Integer	required

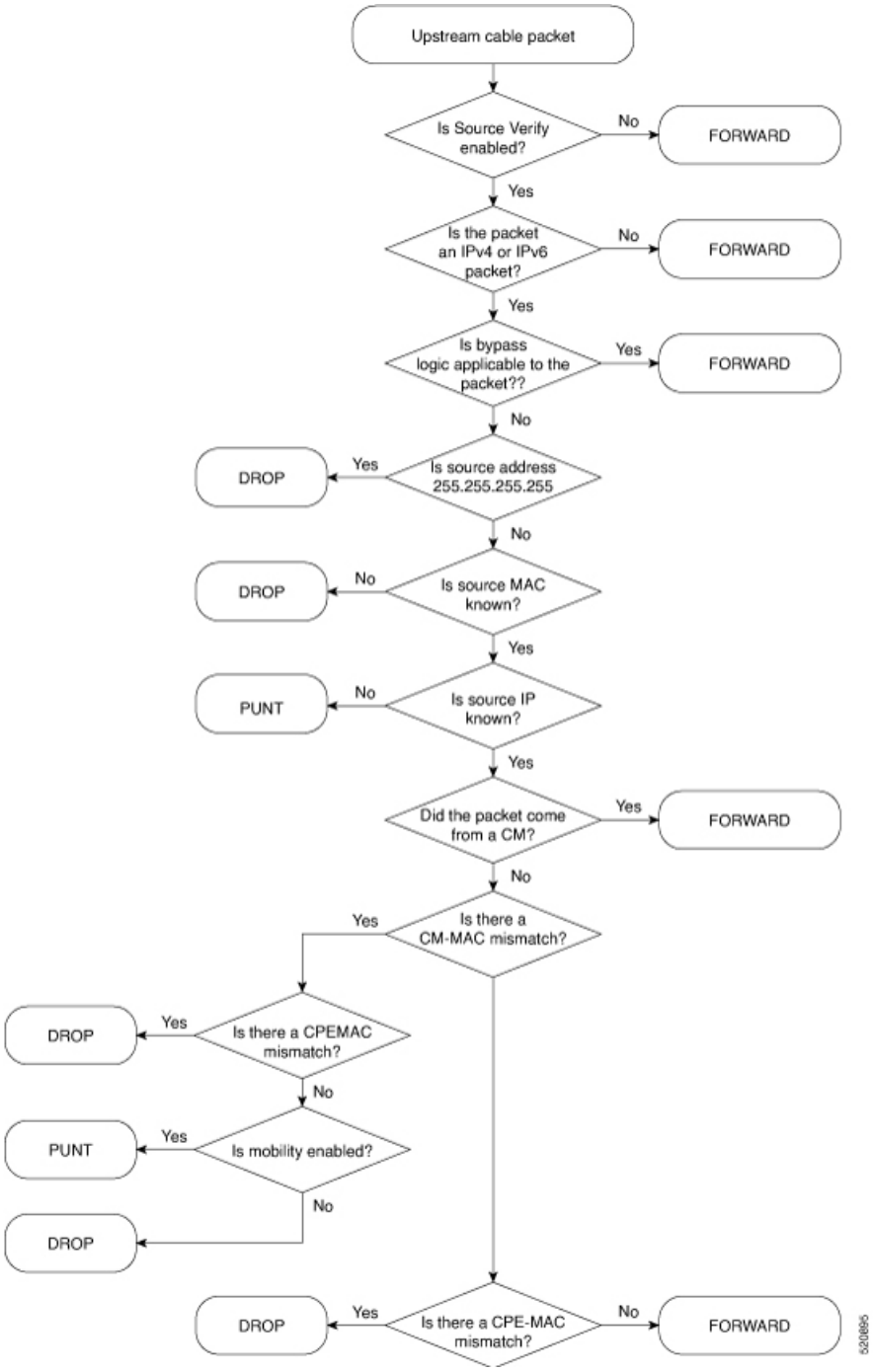
Source-Verify

Source-Verify inhibits certain types of Denial of Service attacks based on IP address spoofing and IP address theft. When you enable Source-Verify, Cisco cnBR verifies the validity of IP packets received from CMs and CPEs. This verification is based on layer 2 and layer 3 addresses known to Cisco cnBR. Cisco cnBR learns the layer 2 and layer 3 addresses when DHCP assigns IP addresses to CM and CPE clients. If Cisco cnBR cannot determine the validity of a packet, it generates a lease-query in order to verify the packet. Source-Verify supports CPE IPv6 Prefix Delegation.

Source-Verify Logic

The following flowchart describes the Source-Verify logic in Cisco cnBR.

Figure 38: Source-verify logic



Bypass Logic

Cisco cnBR forwards packets that match any of the following criteria. These packets pass Source-Verify.

- IPv4 packets with src address 0.0.0.0
- IPv6 packets with multicast link local destination address

- IPv6 packets with unicast link local source or destination address
- IPv6 packets with unspecified source address

Invalid src Logic

Cisco cnBR drops packets that match the following criteria. These packets fail Source-Verify.

- IPv4 packets with source address 255.255.255.255

Configure Source-Verify

Initial Configuration of Source Verify From Autodeployer Script

In the Autodeployer script L3 template file, the Source-Verify configuration is in the *dhcp* block. To enable IPv4 Source-Verify, set *ipv4Lq* to true. To enable IPv6 Source-Verify, set *ipv6Lq* to true. To enable mobility, align CM/CPE scope with *mobilityScopes*.

```
"sgs": [
    ...
    "sg-config": {
        ...
        "dhcp": {
            "arpGlean": true,
            "arpProxy": true,
            "dhcpIfname": "cnr",
            "dhcpServers": [
                "10.2.2.91"
            ],
            "ipv4Lq": true,
            "ipv6Lq": true,
            "mobilityScopes": [
                "10.1.1.1/24",
                "2001::a/88"
            ],
            "ndProxy": true,
            "relayModeV4": 0,
            "relayModeV6": 0,
            "relayPolicies": [
                {
                    "deviceClass": "HOST",
                    "giAddr": "24.44.9.2",
                    "linkAddr": "2010::1",
                    "v4ServerIp": "1.2.2.91"
                }
            ],
            "v4Nets": [
                "9.44.9.2/24",
                "24.44.9.2/24"
            ],
            "v6Nets": null
        },
        ...
```

View Current Configuration Using Cisco Operations Hub Configurator

Step 1 Log into the Cisco Operations Hub.

Step 2 Click **Configurator** from the horizontal navigation tab.

Step 3 Click **Export/Import** under **cnBR-Core Manage** from the vertical navigation tab to access the **Export/Import** page.

Step 4 In the **Export cnBR Configuration** section, select the target Cisco cnBR from the drop down list.

Step 5 Click **Export** to retrieve the SG configuration of the selected Cisco cnBR.

A .json file containing the full configuration is saved to your machine. Source-Verify settings are available in the *dhcp* block.

Update Configuration

You can update the configuration using the following methods:

- Cisco Operations Hub Configurator
- Autodeployer reconfiguration

Both options send the full configuration to the CMTS. Cisco cnBR overwrites the existing configuration and activates the new configuration. For more details, see Autodeployer Limitations, on page 47.

Update Configuration Using Cisco Operations Hub Configurator

Step 1 Log into the Cisco Operations Hub.

Step 2 Click **Configurator** from the horizontal navigation tab.

Step 3 Click **Export/Import** under **cnBR-Core Manage** from the vertical navigation tab to access the **Export/Import** page.

Step 4 In the **Export cnBR Configuration** section, select the target Cisco cnBR from the drop down list.

Step 5 Click **Export** to retrieve the SG configuration of the selected Cisco cnBR.

Step 6 Update the configuration in the *dhcp* block of the SG configuration and save the file.

Step 7 In the **Import cnBR Configuration File** section, select the target Cisco cnBR from the drop down list.

Step 8 Click **Browse** and select the saved configuration file.

Step 9 Click **Import** to push the updated SG configuration.

This import overwrites the existing configuration and activates the new configuration.

Update Configuration Using Autodeployer Reconfiguration

After the initial configuration of Source-Verify using the Autodeployer, update the configuration by modifying the corresponding blocks in the Autodeployer script and rerunning the Autodeployer. This process overwrites the existing configuration and activates the new configuration.

Default Configuration

By default, Source-Verify for both IPv4 and IPv6 is disabled.

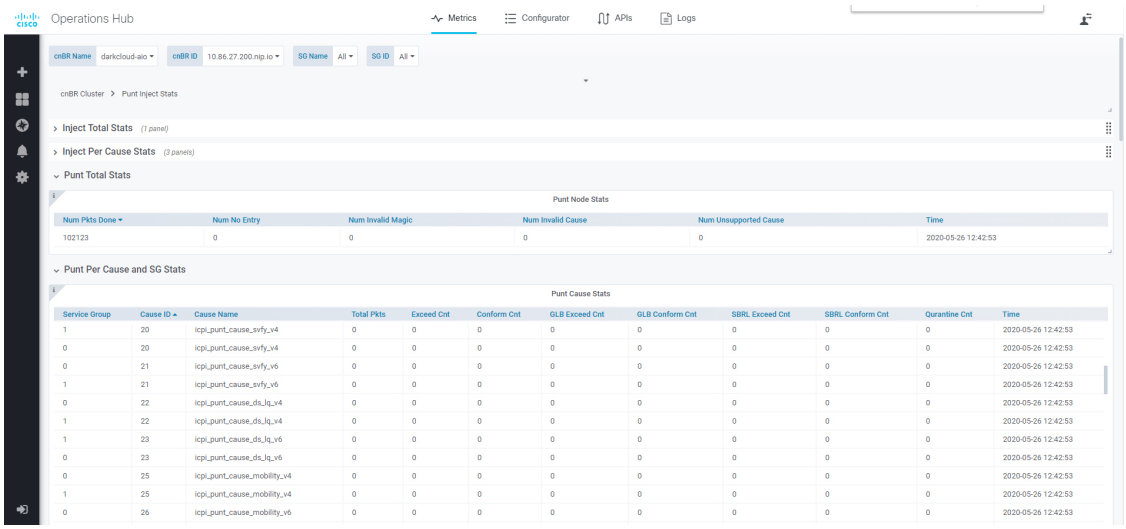
Monitoring

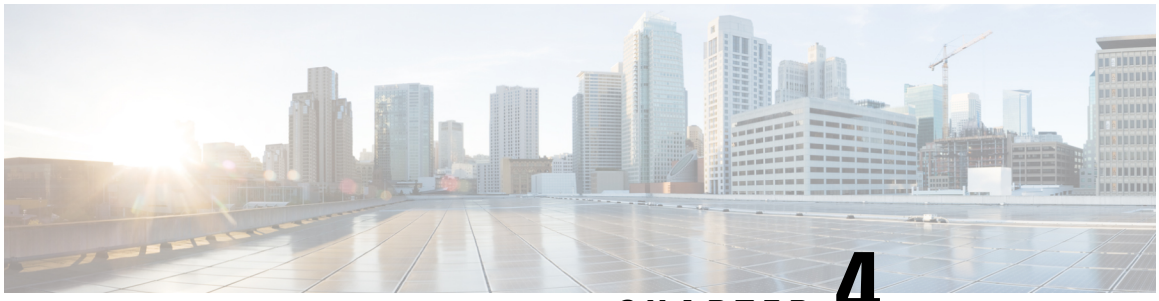
When the Cisco cnBR is unable to determine packet validity in the dataplane, it punts the packet for lease-query generation. Only punt statistics are available for Source-Verify.

- Mobility packets get the *mobility_v4* or *mobility_v6* punt-cause.
- All other Source-Verify punts get the *svfy_v4* or *svfy_v6* punt-cause.

In the Cisco Operations Hub Metrics interface, the Punt Inject Stats panel contains the punt statistics for Source-Verify and Mobility. Punted packets are subject to Punt-Rate-Limit processing. See Punt Path Rate Limiting in Data Plane, on page 140 for more information on these statistics.

Figure 39: Punt Inject Stats panel





CHAPTER 4

Cisco Cloud Native Broadband Router Maintenance

Cisco cnBR enables you to perform software upgrades seamlessly, and without disrupting any of the services. You can continuously deploy new services and features with minimal downtime.

RPD Secure Software Download

The Cisco Operations Hub provides automated ways to securely download and activate software images to RPDs.

The secure software download (SSD) feature helps you to authenticate the source of a file and verify the integrity of the downloaded code before you use it in your system. The SSD feature is applicable to Remote PHY (R-PHY) devices installed in unsecure locations.

Prerequisites

To use SSD, the following prerequisites must be met:

- A TFTP or HTTP server is available that has a network connection to the RPD.
- The RPD software image is available at the external image server. The image server is where the software image is stored from where it can be accessed by RPD.
- Code validation certificates are available. For more details, see the Add Code Validation Certificates section.

Download Software Image for RPD

Download the software image from a specified server. The software image is available on an external image server.

To download an RPD software image using SSD, perform the following tasks:

1. Manually upload the software image to the external image server.
2. Add code validation certificates.
3. Upgrade the software image.

Add Code Validation Certificates

To authenticate the source and verify the integrity of the software image, Cisco cnBR uses the following two types of RPD code validation certificates (CVC).

- M-CVC: The type of CVC released along with the Cisco RPD software image. Contact Cisco Support to get the M-CVC.
- C-CVC: The type of CVC created and signed through Manufacturer's Statement of Origin (MSO). When CVCs are available, upload them using the following procedure:

Step 1 Choose **Operations Hub** > **Configurator**.

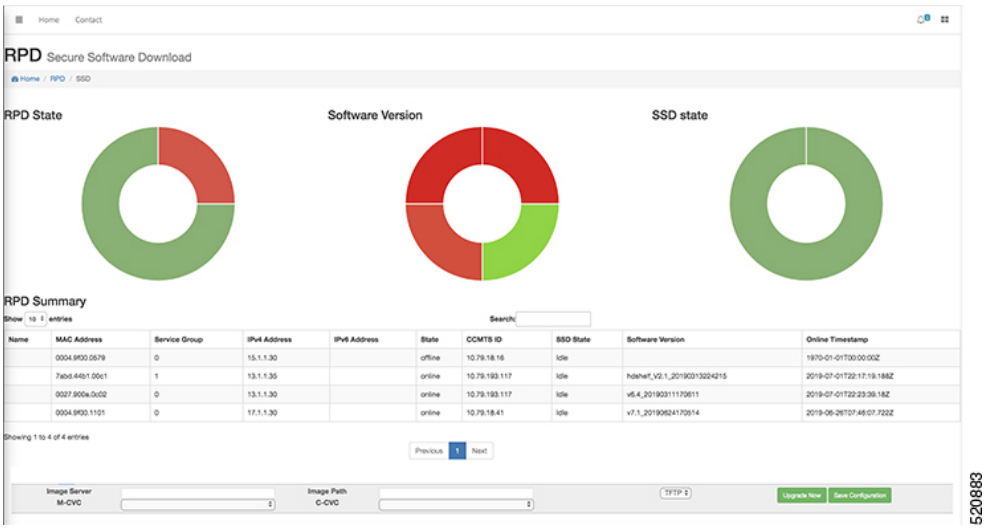
Step 2 From the left pane, choose **RPD Manage** > **RPD CVC**.

Step 3 Copy the contents from the CVC file to the appropriate text box and click **Add**.

Upgrade the Software Image

To upgrade the software, use the following procedure:

Step 1 Choose **Operations Hub** > **Configurator** > **RPD SSD**.



Step 2 Scroll down to the bottom of the page and enter the following details in the appropriate text fields:

Field	Description
Image Server	Address of the server where the software image is stored from where it can be accessed by RPDs.
Image Path	The relative path of the RPD software image on the server. The file is available in the default directory of the image server.
Method	HTTP or TFTP for RPD download SSD image.
M-CVC	Indicator showing whether the certificate is valid or not.
C-CVC	Indicator showing whether the certificate is valid or not.

Step 3 To upgrade immediately, click **Upgrade Now**.

Or

To upgrade during the next reboot, click the **Save Configuration**.

Monitor RPD and SSD State

The RPD SSD window provides options to monitor and trigger SSD operations. A dashboard, displaying three pie charts, provides details of the RPD status and metrics. Access this dashboard under the **Cisco Operations Hub** > **Configurator** > **RPD SSD**.

- RPD State: Displays the states of RPDs that are upgraded. During the upgrade process, the RPD becomes offline and then returns online.
- Software Version: Shows the number of RPDs for each RPD software version.
- SSD State: Shows various phases of the SSD progress of RPDs.

RPD Summary

The RPD Summary table provides details of RPDs which can be upgraded. You can also search for a specific RPD or set of RPDs that can be upgraded. The following table explains the fields in the RPD Summary pane.

Field	Description
Name	Name of the RPD.
MAC Address	MAC address of the RPD.
Service Group	Service group ID of the RPD.
IPv4 Address	IPv4 address of the RPD.
IPv6 Address	IPv6 address of the RPD.
State	Status of the RPD: • online • offline
CCMTS ID	Host name of the Cisco cnBR application. Example: `https://190.x.x.x.nip.io/`
SSD State	Phase of the SSD progress.
Software Version	Version of the software running on the RPD.
Online Timestamp	Time when the RPD became online.

Offline Image Upgrade

Cisco cnBR supports Offline Image Upgrade. The image upgrade workflow provides a dashboard that simplifies the image upgrade for both Cisco cnBR and Cisco Operations Hub.

Note The image upgrade workflow only supports the upgrade of the `cmts-app`, `opshub-app`, and `cloud-infra-app` charts.

Image Upgrade Preparation

Complete the following steps to prepare an image for upgrade:

Step 1 On the Cisco Operations Hub, click **Configurator** > **cnBR-Core Manage** > **Add cnBR Core**.

Step 2 Provide a unique name to the Cisco cnBR core, a namespace, and Core Ingress-host-name. See the following example:

```
cnBR-Core Name: Upgrader-demo
Core Namespace: ccmts-infra
Core Ingress-host-name: 10.124.210.65.nip.io
```

Step 3 Enter the Cisco cnBR username and password.

Step 4 Click **ADD**.

Step 5 Copy the `cnbr-installer-v20.2-06042020.tar.gz` installer bundle image to a staging server.

The installer bundle name <*06042020*> deontes the the date MMDDYYYY.

Step 6 Decompress the image into the directory.

Step 7 Setup the configuration file by following the steps as listed in the Step 1 and Step 2 section.

Step 8 Run the following autodeploy command to update the image on deployer:

```
./deploy -c <day0 config file> -u
```

The image update process takes 30–45 minutes on the deployer.

The new image URL format is as follows:

```
http://chart.<deployer's ip>.nip.io/<image name>/
```

Based on the product type, the __\<image name\>__ is either `cnbr-master` or `opshub-master`.

Image Upgrade

Complete the following steps to upgrade the image:

Step 1 Access the link: __https://<operations_hub_ip>.nip.io/configurator/upgrader/__.

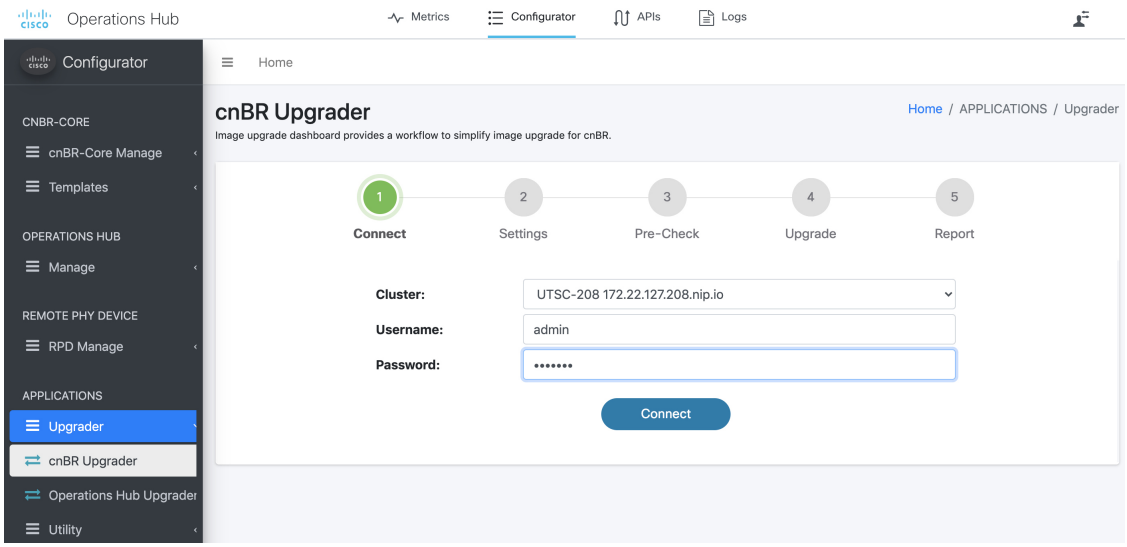
Step 2 Click **Upgrader** > **cnBR Upgrader**.

Step 3 Select the Cisco cnBR cluster that you want to upgrade.

Step 4 Enter the username and password.

Step 5 Click **Connect**.

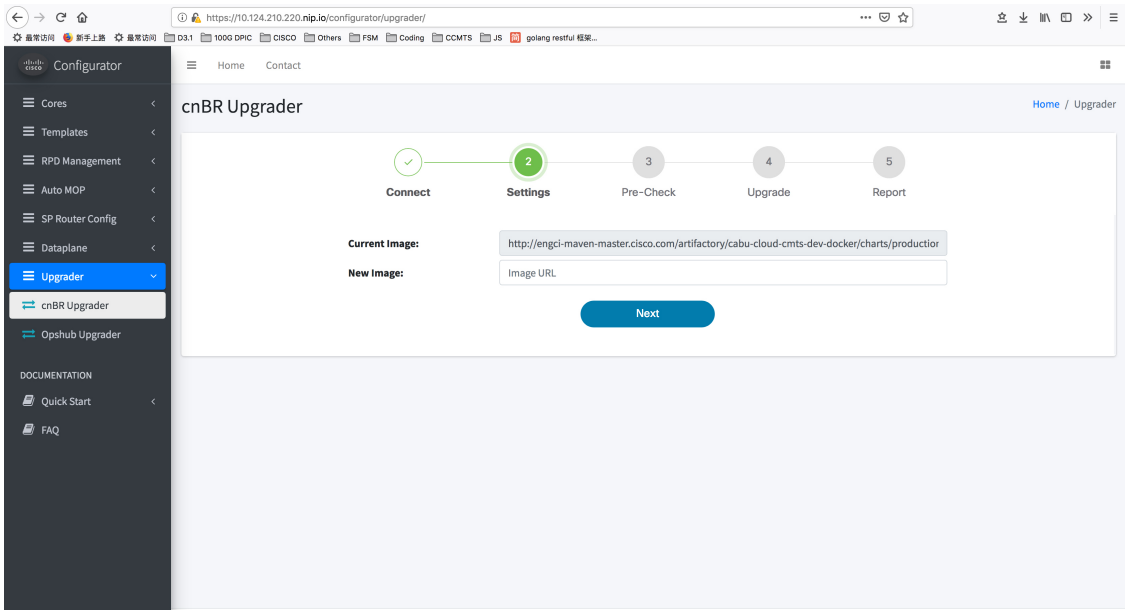
Figure 40: Connecting to the cnBR Image Upgrader



You can upgrade only the Cisco Operations Hub that is currently in use. You cannot choose a cluster when you want to upgrade the Cisco Operations Hub.

Step 6 Enter the image you want to upgrade. Provide the target URL obtained from Image Upgrade Preparation, on page 162 topic.

Figure 41: Providing the New Image to Upgrade



Step 7 Click **Next**.

Step 8 Check the following before an image upgrade:

- Helm status: Ensure that the Helm releases status is listed as DEPLOYED. To recover failed images, go through the steps that are listed in Image Recovery, on page 165.

- Updates of the new image: Lists the differences between the current and target versions.
- Target cluster pod status: Lists the status of all Pods.

Step 9 Click **Upgrade**. During the Cisco Operations Hub upgrade, the page may redirect to the Cisco Operations Hub login page. The redirect can happen due to any back-end service downtime. To resolve the issue, log in to the Cisco Operations Hub and go through step Step 1, on page 163. The workflow would jump to step Step 4, on page 163 and continue the monitoring progress.

The Image upgrade report is displayed.

Step 10 Click **SHOW** to view detailed differences of image and pod statuses before and after upgrade.

Image Recovery

To recover from an environment failure during the upgrade process, go through the following steps:

Step 1 Label all the DOCSIS worker node with the following label using deployer CLI:

```
config terminal
cluster <cluster-name>
nodes docsis-1
no k8s node-labels type_cmts no
k8s node-labels smi.cisco.com/node-type docsis
exit
exit
nodes docsis-2
no k8s node-labels type_cmts no
k8s node-labels smi.cisco.com/node-type docsis
exit
exit
nodes docsis-3
no k8s node-labels type_cmts no
k8s node-labels smi.cisco.com/node-type docsis
exit
exit
```

Note The value `<docsis-n>` denotes a number of K8s nodes. If there are more UCS servers or nodes in the system, you must repeat the steps for every worker node.

Step 2 Clean up environment. To clean up the ops-center in deployer:

```
config terminal
cluster <cluser-name>
no ops-centers cnBR infra
commit
end
clusters <cluster-name> actions sync run
```

You can check the synchronization progress by using the following CLI:

```
clusters <cluster-name> actions sync status
```

Step 3 Reconfigure the ops-centers image with the new image:

```
conf t
cluster <cluser-name>
ops-centers cnBR infra
  repository <image url>
  initial-boot-parameters use-volume-claims true
  initial-boot-parameters first-boot-password <password>
  initial-boot-parameters auto-deploy true
  initial-boot-parameters single-node false
commit
end
clusters <cluster-name> actions sync run
```

Drain Worker Node

The Cisco Cloud Native Broadband Router enables you to move Data-over-Cable Systems Interface Standard (DOCSIS) service group workloads to other Cisco cnBR nodes during maintenance and troubleshooting activities. The Drain Cisco cnBR Node feature helps you to avoid service interruptions during maintenance activities, when the workloads of Cisco cnBR nodes are brought out of service.

Drain the Node

You can drain a node by moving the DOCSIS service group workloads from the node. Draining enables the node to be safely removed from the cluster, allowing other nodes to take up workloads.

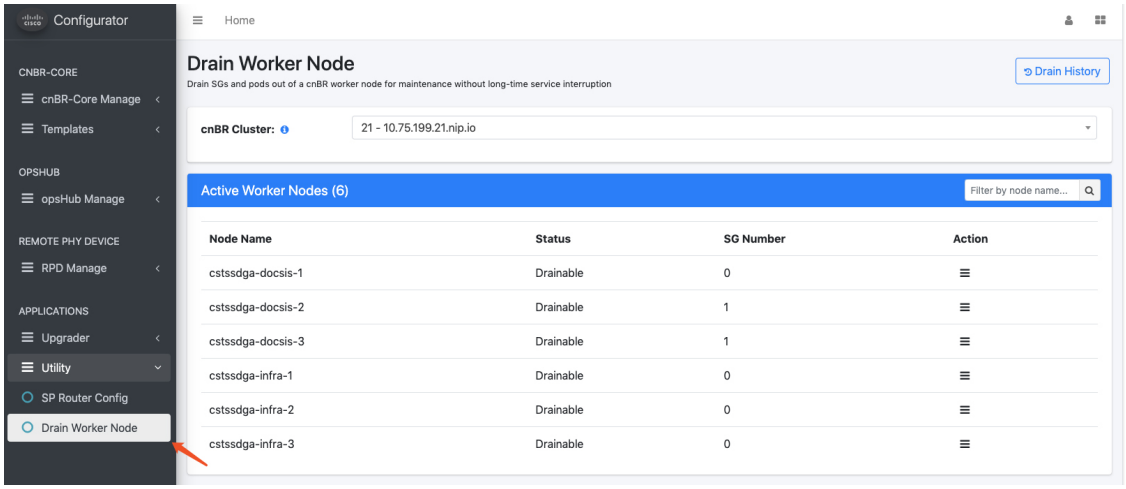
Complete the following steps to drain a node.

Step 1 On the Cisco Operations Hub, click **Configurator** > **Utility** > **Drain Worker Node**. This launches the Drain Worker Node panel.

Step 2 Select one Cisco cnBR cluster from the drop-down list at the header.

The Active Worker Nodes table displays the drainable nodes belonging to the selected cluster.

Figure 42: Listing the Active Worker Nodes



Step 3 Click **Action** > **Drain**, and choose to confirm the drain action.

The progress bar indicates progress of the draining activity. The drained node appears in the Drained Worker Nodes table after all DOCSIS workloads are moved.

Activate the Node

To move a drained node back to the working pool after maintenance, complete the following step:

Select the drained node from the Drained Worker Nodes table. Click **Action** > **Activate**.

The selected node appears in the Active Worker Nodes table, after confirmation.

Audit of the Drain History

Every drain and activate operation is recorded for audit.

To view the drain or activation history of a node, complete the following step:

Click **Drain History** to view the history.

The Operation History table lists the drain target, action, time, and status.

Drain Worker Node Errors and Warnings

The Drain Worker Node has the following errors and warnings:

Error

Error: Failed to drain node <*node-name*>, reason: job failed. Please try again later.

Diagnosis: The common cause for a draining job failure is timeout while waiting for responses from other microservices.

Solution: Attempt the operation later and see whether the issue is resolved.

Warning

Warning: Unable to drain <*node-name*>, reason: Insufficient SG capacity in other worker node.

Diagnosis: When draining a DOCSIS node, the service groups need to be moved to other DOCSIS nodes in order to keep the service running. In some cases, when other DOCSIS nodes do not have the capacity to hold all service groups, an error dialog warns of the insufficient capacity.

Solution: To resolve the issue, click **Cancel** to abort the drain operation. You can alternatively drain the node with the **Force Drain**.

Note We do not recommend the Force Drain method as it may cause several service groups to be unserved by the cluster. This can increase service donwtime.

Export and Import Configuration

The system administrator perform import and export Cisco cnBR and Cisco Operations Hub configurations using the Cisco Operations Hub UI or RESTful APIs. The system administrator can store the exported configuration at a secure location. For Disaster Recovery, the system administrator performs the import operation, to restore the Cisco cnBR, the Cisco Operations Hub, or both to their original configurations.

Export Cisco cnBR Configuration using Cisco Operations Hub

From the Configurator interface of Cisco Operations Hub, perform the following steps to export the Cisco cnBR configuration:

Step 1 From the vertical navigation tab, click **Export/Import** under **cnBR-Core Manage**.

Step 2 Select the target Cisco cnBR from the drop-down list in the **Export cnBR Configuration** section.

Step 3 Click **Export**.

Step 4 Rename the file and save it at a secure location.

Export Cisco cnBR Configuration using RESTful API

Execute the following command in a UNIX shell to export the Cisco cnBR configuration:

```
curl -k -X GET 'https://{opsHUBHost}/api/configurator/v1/cmts/config/{cmts-id}' -H 'Accept:
application/json' -H 'Authorization: Bearer <token>' | tee path/to/backup/config
```

Example

```
hostname#curl -k -X GET
'https://172.22.29.221.nip.io/api/configurator/v1/cmts/config/10.79.193.236.nip.io' -H
'Accept: application/json' -H 'Authorization: Bearer <token>' | tee
cnbr-10.79.193.236-configuration.json
```

Export Cisco Operations Hub Configuration using Cisco Operations Hub

From the Configurator interface of Cisco Operations Hub, perform the following steps to export the Cisco Operations Hub Configuration:

Step 1 From the vertical navigation tab, click **Export/Import** under **Cisco Operations Hub > Manage**.

Step 2 In the **Export Cisco Operations Hub Configuration** section, click **Export**.

Step 3 Rename the file and save it at a secure location.

Export Cisco Operations Hub Configuration using RESTful API

Execute the following command in a UNIX shell to export the Cisco Operations Hub configuration:

```
curl -k -X GET 'https://{opsHUBHost}/configurator/opshub/export' -H 'Accept: application/json' -H
'Authorization: Bearer <token>' | tee path/to/backup/config
```

Example

```
hostname#curl -k -X GET 'https://172.22.29.221.nip.io/configurator/opshub/expor' -H 'Accept:
 application/json' -H 'Authorization: Bearer <token>' | tee
opshub-172.22.29.221-configuration.json
```

Import Cisco cnBR Configuration using Cisco Operations Hub

From the Configurator interface of Cisco Operations Hub, perform the following steps to import the Cisco cnBR configuration:

Step 1 From the vertical navigation tab, click **Export/Import** under **cnBR-Core Manage**.

Step 2 Select the target Cisco cnBR from the drop-down list in the **Import cnBR Configuration File** section.

Step 3 Select the configuration file that you want to import.

Step 4 Click **Import**.

Import Cisco cnBR Configuration using RESTful API

Execute the following command in a UNIX shell to Import the Cisco cnBR configuration:

```
curl -k -X PUT 'https://{opsHUBHost}/api/configurator/v1/cmts/config/{cmts-id}' -H 'Accept:
application/json' -H 'Content-Type: application/json' -H 'Authorization: Bearer <token>' -d
'@path/to/backed/up/config
```

Example

```
hostname#curl -k -X PUT
'https://172.22.29.221.nip.io/api/configurator/v1/cmts/config/10.79.193.236.nip.io' -H
'Accept: application/json' -H 'Content-Type: application/json' -H 'Authorization: Bearer
<token>' -d '@cnbr-10.79.193.236-configuration.json
```

Import Cisco Operations Hub Configuration using Cisco Operations Hub

From the Configurator interface of Cisco Operations Hub, perform the following steps to import the Cisco Operations Hub configuration:

Step 1 From the vertical navigation tab, click **Export/Import** under **Cisco Operations Hub** > **Manage**.

Step 2 In the **Import Cisco Operations Hub Configuration** section, select the configuration file you want to import.

Step 3 Click **Import**.

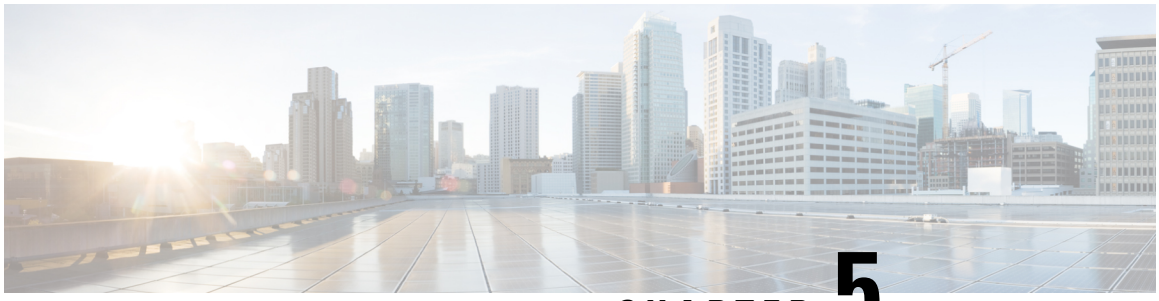
Import Cisco Operations Hub Configuration using RESTful API

Execute the following command in a UNIX shell to import the Cisco Operations Hub configuration:

```
curl -k -X PUT "https://{opsHUBHost}/configurator/opshub/import" -H "accept: application/json" -H
"Content-Type: application/json" -H 'Authorization: Bearer <token>' -d "@path/to/backed/up/config
```

Example

```
hostname#curl -k -X PUT 'https://172.22.29.221.nip.io/configurator/opshub/import' -H 'Accept:
 application/json' -H 'Content-Type: application/json' -H 'Authorization: Bearer <token>'
-d '@opshub-172.22.29.221-configuration.json
```



CHAPTER 5

Cisco Cloud Native Broadband Router Diagnosis

The Cisco cnBR provides a suite of in-built tools to diagnose and resolve common issues.

Cable Modem Diagnosis Tool

In a Data-over-Cable Systems Interface Standard (DOCSIS) environment, various elements can affect a modem's ability to maintain a connection and remain online. When a cable modem goes offline, it is difficult to diagnose the cause and identify the issues.

The Cisco cnBR includes a Cable Modem Diagnosis Tool to enable easy diagnosis of such issues. Checkpoints are created periodically for online modems, where information such as system logs, configuration details, and system statistics are saved. When a cable modem goes offline, this system information is analyzed from the saved checkpoints.

The Cable Modem Diagnosis Tool supports the following modes:

- **On-demand mode**: System logs related to a modem is collected with a single click, when needed.
- **Background mode**: Logs, health metrics and performance metrics are actively analyzed in the background to detect, diagnose, and report modem issues.

The Cable Modem Diagnosis Tool provides the following utilities:

- Detect malfunctioning modems.
- Enable debugging for malfunctioning modems and disable debugging when modems are recovered.
- Supports interactive enabling or disabling of per modem debugging.
- Display modem logs and telemetry on the Grafana dashboard.
- Download of modem logs containing modem log messages.

Configure Cable Modem Diagnosis Tool for On-Demand Diagnosis

On-Demand diagnosis allows debugging a cable modem from the Metrics dashboard. On-Demand diagnosis does not require any configuration changes. You can run the On-Demand diagnosis from the Cisco Operations Hub Metrics dashboard.

Complete the following steps to enable On-Demand Diagnosis:

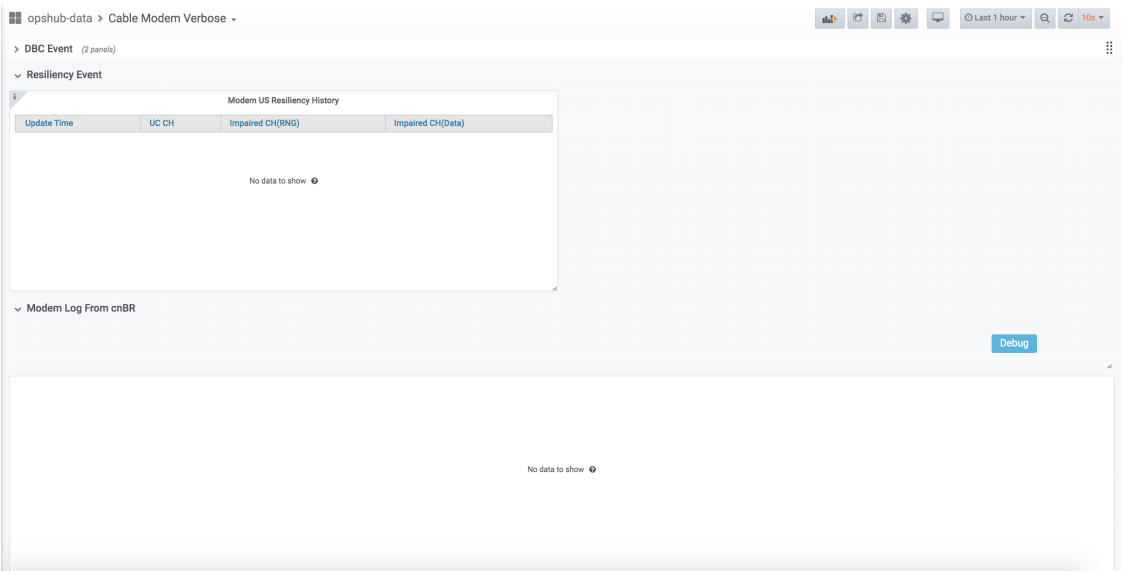
Step 1 On the Cisco cnBR Cisco Operations Hub, click **Metrics** > **opshub-data** > **Cable Modem Verbose**.

Step 2 Select the Cisco cnBR name and modem that you want to debug from the **cnBR Name** and **Cable Modem** drop-down lists.

Step 3 Click **Modem Log From cnBR**.

Step 4 Click **Debug**.

Figure 43: Modem Logs



The debug log is displayed.

Step 5 Click **Disable** to disable debugging.

Configure Cable Modem Diagnosis Tool for Background Diagnosis

The Background diagnosis utility runs periodically, and detects malfunctioning modems. The utility runs automatically in the background, and is enabled by default.

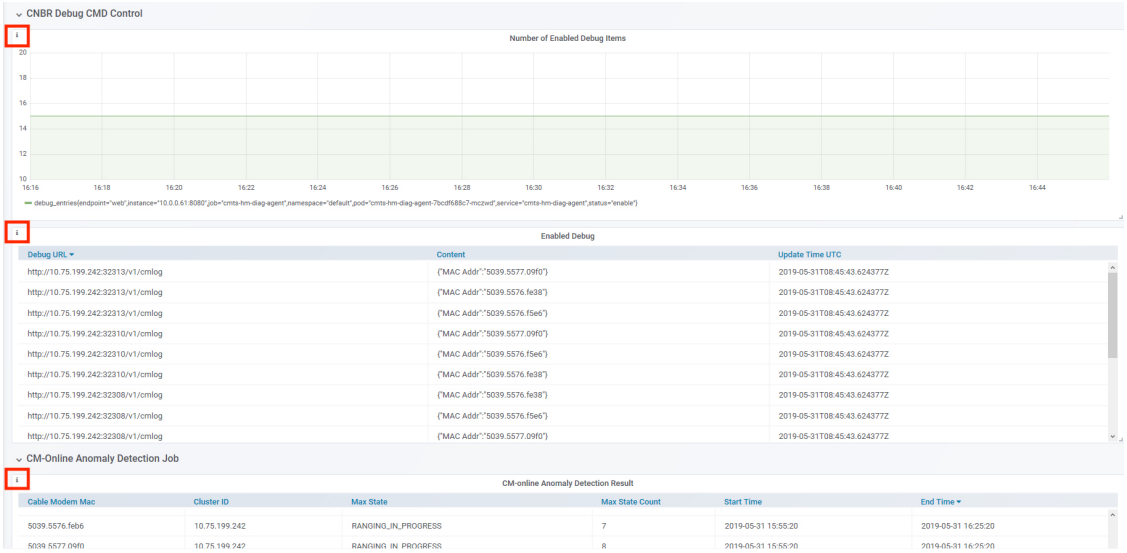
Using the Background diagnosis method, debug functions that collect modem logs are enabled. Complete the following steps to view the logs:

Step 1 On the Cisco cnBR Cisco Operations Hub, click **Metrics** > **opshub-data** > **Diagnosis**.

The debugging information is available in the `Diag Job Summary`, `CNBR Debug CMD Control`, and `CM-Online Anomaly Detection Job` tables.

Step 2 To view detailed information about these tables, expand the tables and click the **i** icon at the top-left corner.

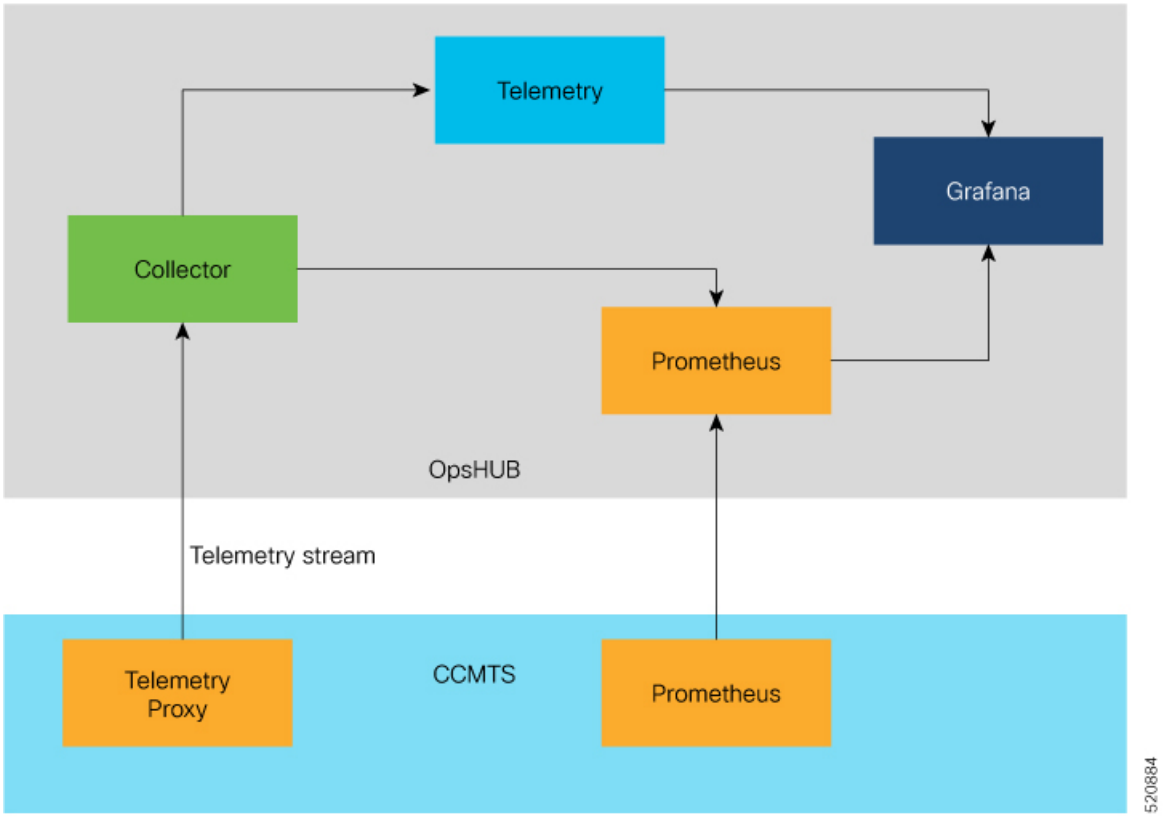
Figure 44: Listing the Diagnostic Information



Cisco cnBR Metrics

The **Metrics** tab in the Cisco cnBR application allows you to monitor the status of the Cisco cnBR router. The Cisco Operations Hub receives metrics and telemetry data from Cisco cnBR. Based on the type of data, the data is saved in the Postgres or Prometheus databases. The Metrics dashboard later retrieves the data and displays it on the dashboard.

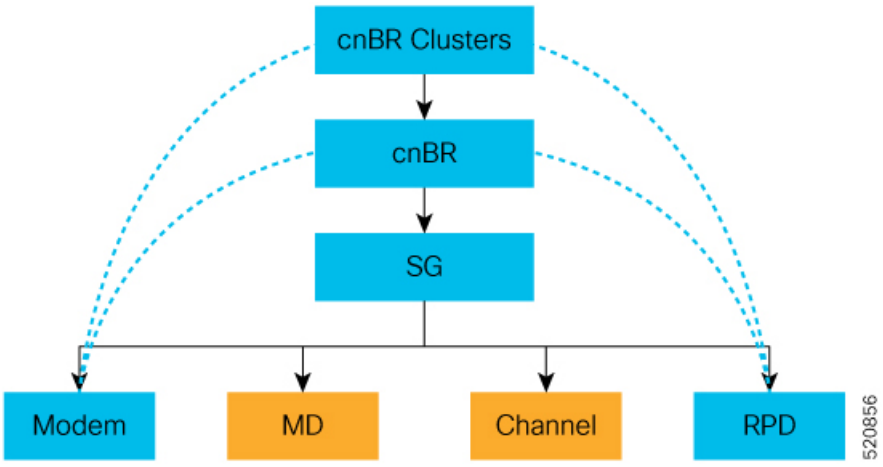
The following illustration shows the Metrics framework.



Cisco cnBR Metrics Dashboards

The Metrics Grafana dashboard displays metrics and status of the DOCSIS network, DOCSIS devices, and CMTS performance status. The Metrics dashboard is based on a hierarchical structure, which matches the Cisco cnBR system deployment exactly.

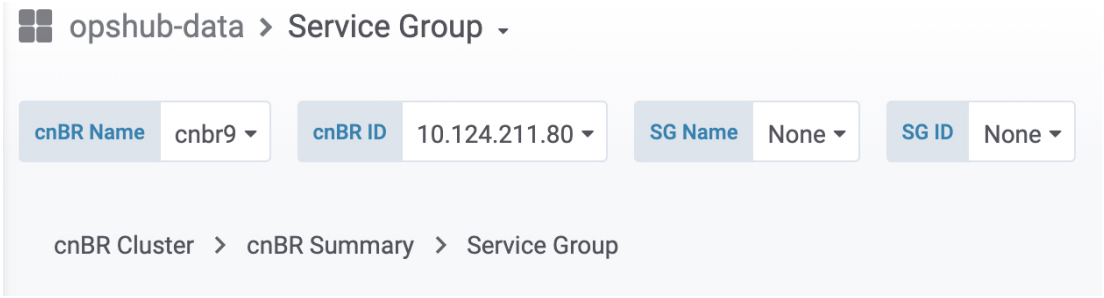
The following illustration shows the hierarchical layout:



The Metrics Dashboard user interface (UI) has the following components:

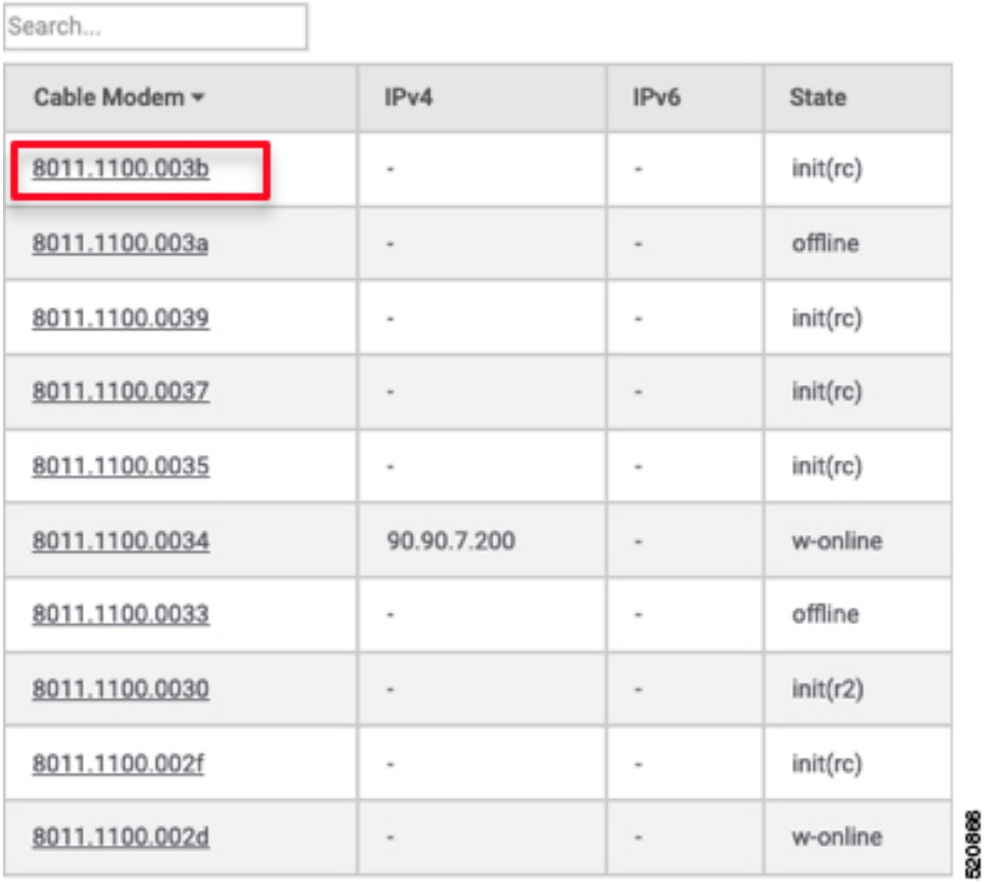
Breadcrumbs Bar

The breadcrumbs bar is available on each dashboard. It shows the dashboard pages just visited. You click each link in the breadcrumbs and go to that specific dashboard.



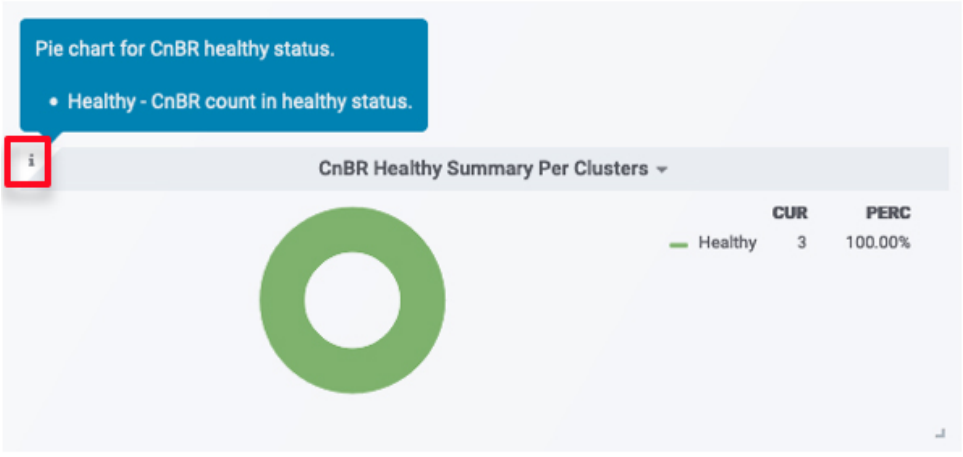
Links

Links are marked using an underline. You click the underlined text and open the related page.



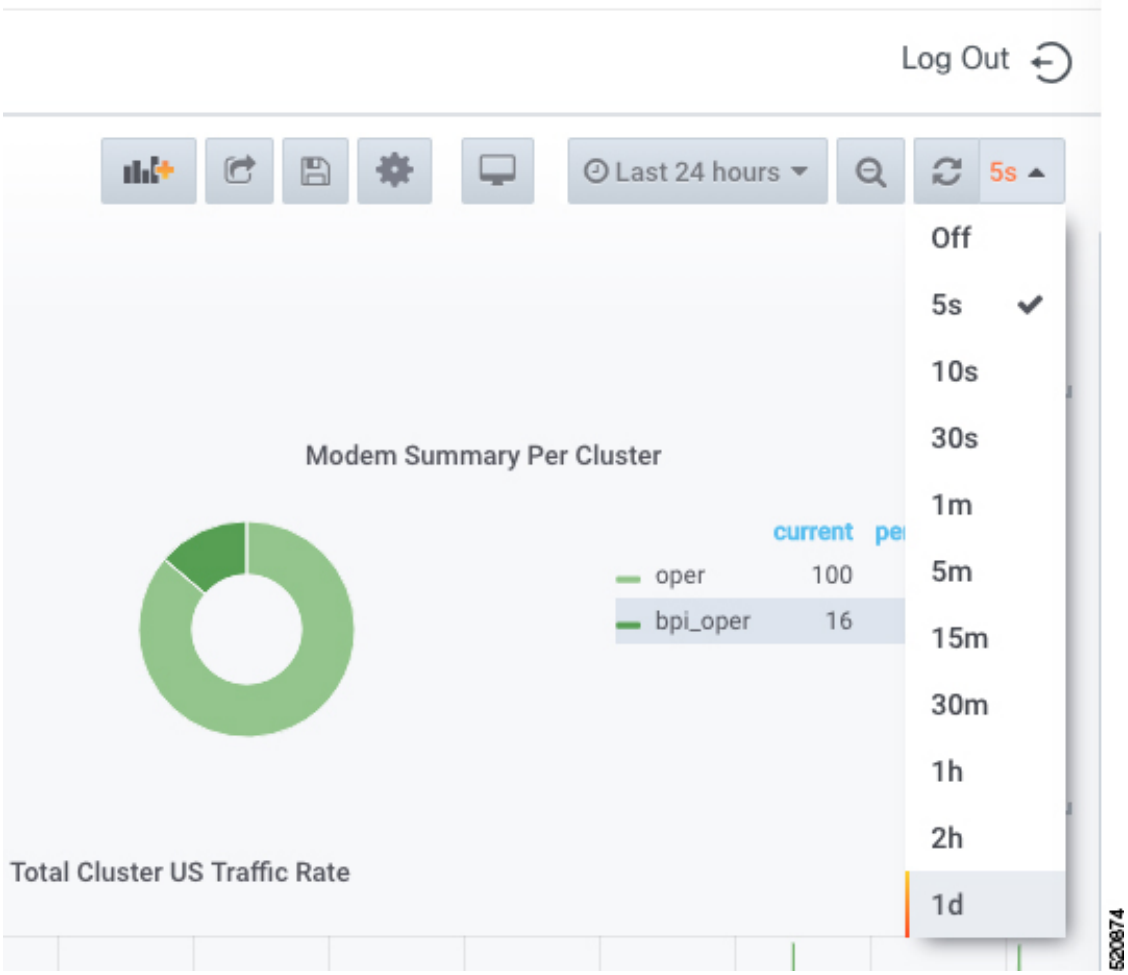
Tooltips

Tooltips are available on the dashboard to display information for each panel on the Grafana dashboard. To view a tooltip, hover your mouse over the `i` on the top-left corner of the panel.

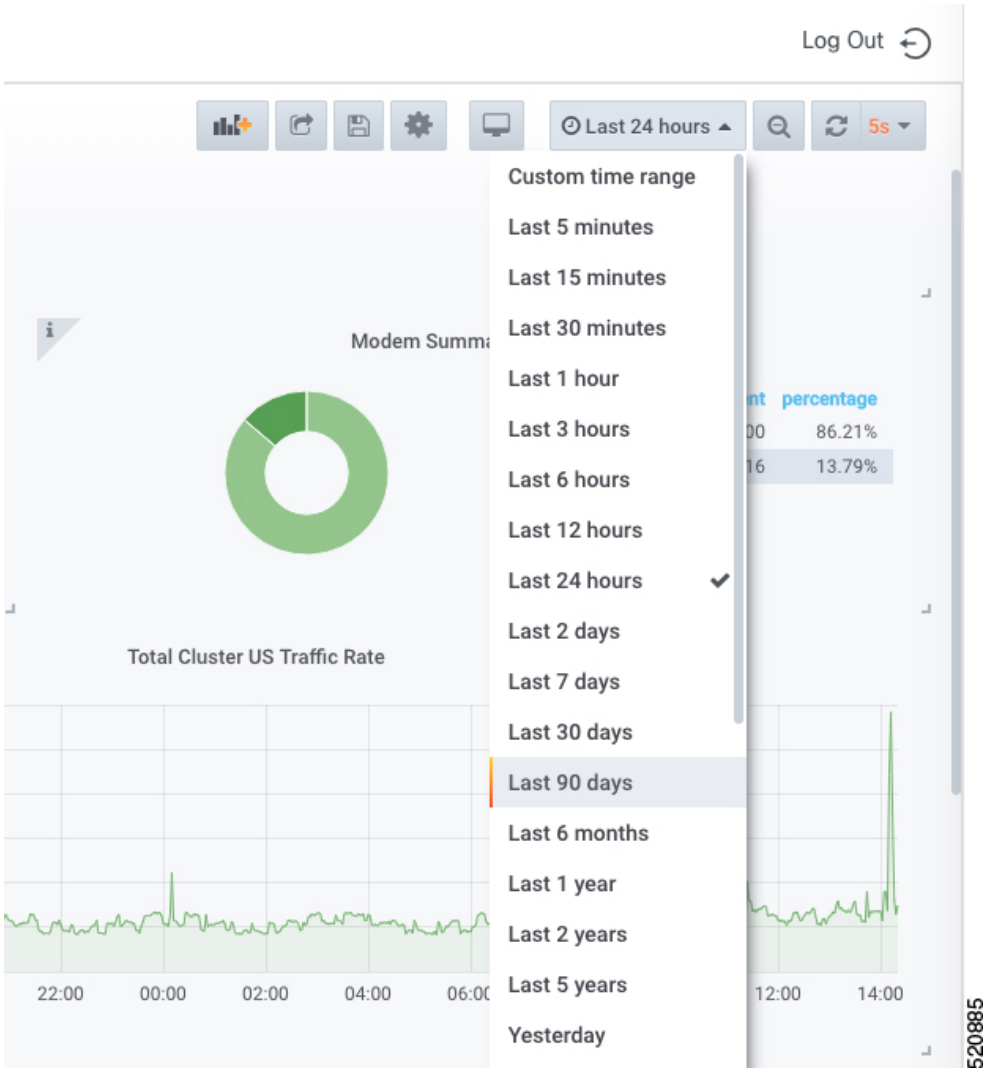


Dashboard Refresh and Time Range

To set the refresh time for each dashboard, choose the time from the drop-down list on the top-right corner of the dashboard. The default refresh time for the dashboard is 10 seconds.



If data is retrieved from the Prometheus database, choose the required value from the **Custom time range** drop-down list as shown in the following image.



Data Display on Dashboard

For all dashboards available in the Cisco cnBR application, data is represented using pie charts, tables, and live graphs.

cnBR Cluster

This Dashboard displays the following information:

Panel Name	Description
cnBR health status	The pie chart shows Cisco cnBR performance status for each Cisco cnBR cluster.

Panel Name	Description
Remote-PHY device (RPD) status	The pie chart shows the status of RPDs in each Cisco cnBR cluster.
Cable Modem status	The pie chart shows the status of cable modems in the Cisco cnBR cluster.
Summary of downstream traffic rate for all cnBR clusters	The graph shows how much download happened in the set time.
Summary of upstream traffic rate for all cnBR clusters	The graph shows how much upstream traffic happened in the set time.
Summary of RPDs in different states per cnBR cluster	The table shows how many RPDs are offline or online, and the following details: • cnBR Name: Name of the cluster • cnBR ID: The IP address to reach the Cisco cnBR router. • init-auth: Authorization status of the RPD • init-gcp: GCP provision status • init-clock: Clock synchronization status • init-l2tp: L2VPN provisioning status • total: Total number of RPDs in the cluster
Summary of modems in different states per cnBR cluster	The table shows a summary of the status of cable modems in the cluster, including the following details: • Ranging: Number of ranging requests received. • DHCP: Number of DHCP requests received. • ToD: Time-of-Day (ToD) requests received. • TFTP: Number of TFTP requests received. • Reg: Number of registration requests (REG-REQ) or multipart registration request (REG-REQ-MP) received. • BPI Err: Number of Baseline Privacy Interface (BPI) errors even if the cable modem is registered. • Offline: Number of modems which are offline. • Oper: Number of cable modems which are registered without enabling BPI. • BPI Oper: Number of cable modems with BPI.
Error log count history for all cnBR clusters	The live graph shows the history of the number of error logs generated for all Cisco cnBR clusters.

Panel Name	Description
Error logs from all cnBR clusters	The live graph shows the error logs generated for all Cisco cnBR clusters.

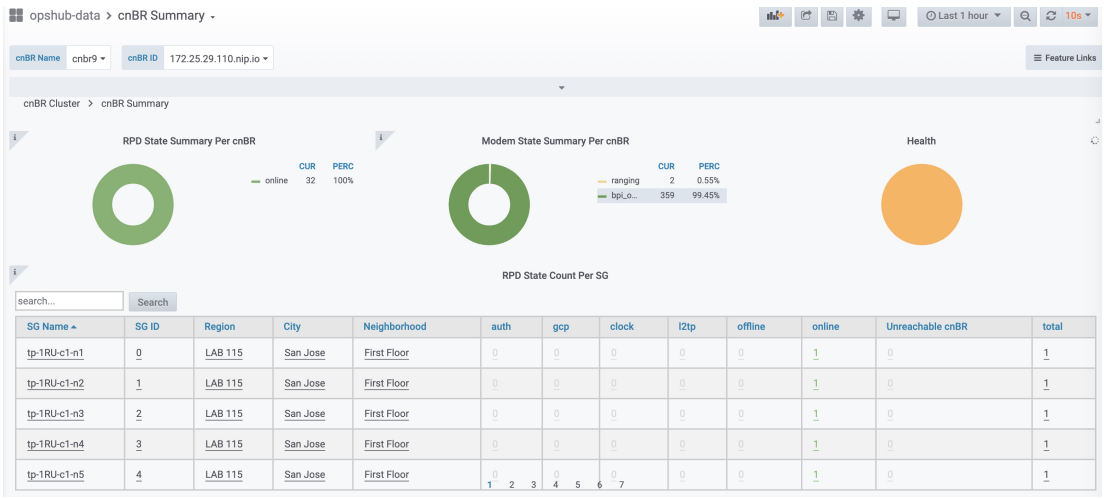
cnBR Summary

The Dashboard displays the following information:

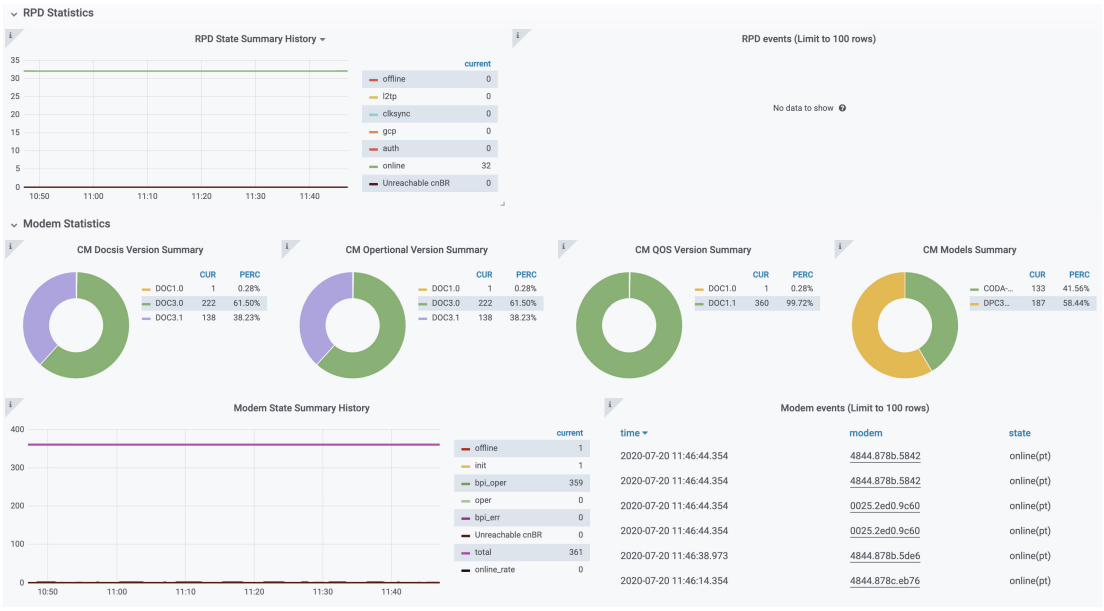
Panel Name	Description
RPD state summary per cnBR	The pie chart shows a summary of RPDs in different states under the current Cisco cnBR cluster: • online: Online state • init(l2tp): Layer Two Tunneling Protocol (L2TP) provision • init(clksync): Clock sync • init(gcp): GCP provision • init(auth): Authorization • offline: Offline state
Modem state summary per cnBR	The pie chart shows a summary of modems in different states under the current Cisco cnBR cluster: • oper: Modem that is registered without BPI enabled • bpi_oper: Modem that is registered with BPI enabled • bpi_error: Modem that is registered but BPI error • reg: REG-REQ or REG-REQ-MP was received • tod: TOD request received • tftp: Trivial File Transfer Protocol (TFTP) request received • dhcp: DHCP request received • ranging: Ranging request received • offline: Offline state
RPD state count per SG	The table provides a summary of RPDs in different states per service group: • SG_ID: Service group id • auth: init(auth) state, authorization • gcp: init(gcp) state, GCP provision • clock: init(clksync) state, clock sync • l2tp: init (L2TP) state, L2TP provision • offline: offline state • online: online state

Panel Name	Description
Modem state count per SG	Summary of modems in different states per service group: • SG_ID: Service group ID • ranging: Ranging request received • tod: TOD request received • dhcp: DHCP request received • tftp: TFTP request received • reg: REG-REQ or REG-REQ-MP was received • bpi_err: Modem that is registered but BPI error • offline: Offline state • oper: Modem that is registered without BPI enabled • bpi_oper: Modem that is registered with BPI enabled
Traffic statistics for all service Groups	Shows traffic statistics for the following: • DS Service Group Traffic Rate • US Service Group Traffic Rate
RPD statistics	History summary of RPDs in different stats
RPD Events	Latest 100 RPD state change events
Modem Statistics	• CM DOCSIS version summary • CM Operational version summary • CM QoS version summary • CM models summary
Modem state summary history	Summary history of modems in different states
Modem Events	Latest 100 modem state change events

Panel Name	Description
Modem List	A detailed modem state information list. Use the Search text box to filter modems by the MAC address, IPv4 address, or IPv6 address. The number of rows in the Modem List table is limited to 256. The total is the total number of modems that are filtered by clusterIp and the search text. • Cable Modem: MAC address of the cable modem • IPv4: IPv4 address of the cable modem • IPv6: IPv6 address of the cable modem • State: State of the cable modem • SG: Service group ID of the cable modem • MD: MAC domain ID of the cable modem • Online Time: Last time when the cable modem transitioned to online • Offline Time: Last time when the cable modem transitioned to offline • Last Update: Last time when the entry was updated.



520854



520855

Service Group

The Dashboard displays the following information:

Panel Name	Description
Cable modem status information	This section shows a summary of the status of cable modems in the specified service group by using a pie chart, a live graph, and the number of modems available with each service: • Ranging: Number of ranging requests received • DHCP: Number of DHCP requests received • TFTP: Number of TFTP requests received • ToD: Time-of-Day (ToD) requests received • Registration: Number of registration requests (REG-REQ) or multipart registration request (REG-REQ-MP) received • Oper: Number of cable modems that are registered without enabling BPI • BPI Oper: Number of cable modems with BPI • BPI Error: Number of Baseline Privacy Interface (BPI) errors even if cable modem is registered • Offline: Number of modems that are offline • Unreachable: Number of modems that are unreachable
Traffic throughput information	Shows the traffic throughput for a selected service group. Provides two live graphs: • DS Traffic Throughput • US Traffic Throughput

Panel Name	Description

RPD information	The table shows a list of RPDs of this service group.
	• MAC Address: MAC address of the RPD, link to RPD Verbose page.
	• Name: Name of the RPD.
	• SG Name: Service group name of the RPD.
	• Service Group: Service group ID of the RPD.
	• IPv4 Address: IPv4 address of the RPD.
	• IPv6 Address: IPv6 address of the RPD.
	• State: State of the RPD.
	  • online
	  • offline
	• Role: Role of the RPD.
	  • principal
	  • auxiliary
	• cnBR ID: cnBR cluster ID of the RPD.
	• Online Timestamp: Timestamp when the RPD is online.

Modem Statistics	This section contains pie charts for the following summaries: • CM DOCSIS version summary • CM QoS version summary • CM OperVer summary • CM models summary • Online CM Summary on Primary DS Chan • Online CM Summary per TCS and US Chan

Panel Name	Description
Modem Table	Cable modem table for selected service group. • Cable Modem: MAC address of the cable modem. • IPv4: IPv4 address of the cable modem. • IPv6: IPv6 address of the cable modem. • State: State of the cable modem. • SG: Service group of the cable modem. • MD: MAC domain of the cable modem. • SID: Service ID of the cable modem. • DS Count: Downstream channel count of the cable modem. • US Count: Upstream channel count of the cable modem • CPE Count: CPE count of the cable modem • Online Time: Timestamp when the modem online • Offline Time: Timestamp when the cable modem offline. You can do the following from this window: • Reset: Reset the modems in the list. • Delete: Delete the modems in the list.
CM Reachability	Displays a graph and a table for the cable modems which are not reachable.

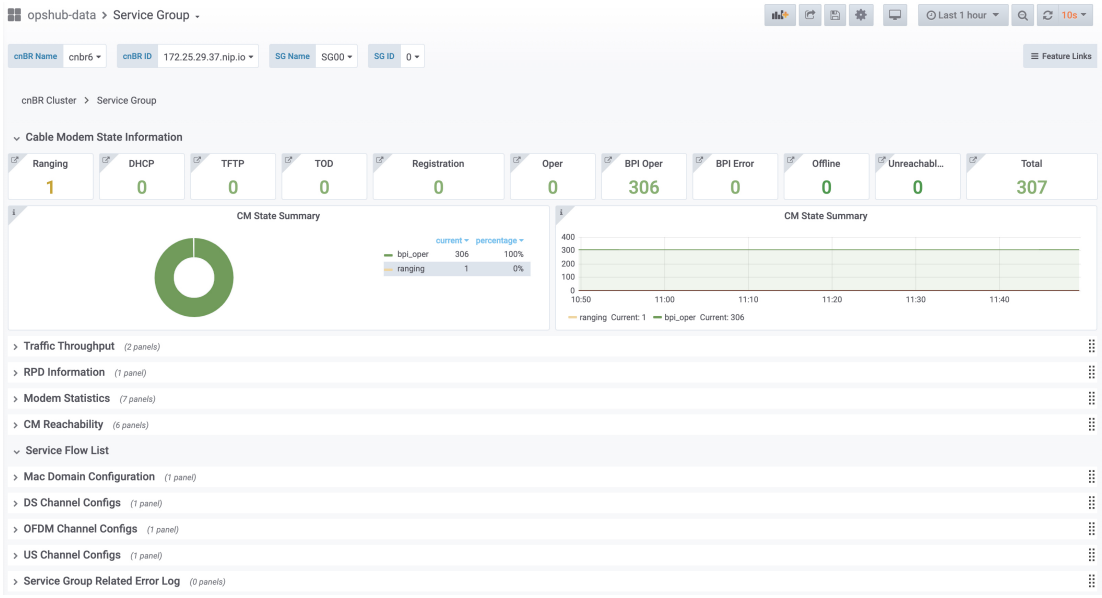
Panel Name	Description

Service Flow List	The **Downstream Service Flow List** and **Upstream Service Flow List** tables provide the following details:
	• SF ID: Service Flow ID.
	• CableModem: MAC address of the modem.
	• Stage: Stage of service flow:
	  • PRE_REGISTRATION: Service flow is provisioned before REGISTRATION.
	  • REGISTRATION: Service flow is provisioned in REGISTRATION.
	• Frame Type:
	  • PRE_D30: Pre-3.0 DOCSIS concatenation and fragmentation.
	  • CCF_ON: Continous Concatenation and Fragmentation is enabled.
	  • CCF_OFF: Continous Concatenation and Fragmentation is disabled.
	• State: State of service flow
	  • Prov: Service flow is in provision.
	  • Adm: Service flow is in admit.
	  • Active: Service flow is active.
	  • Type: Primary, Secondary
	  • MdID: MAC Domain ID of the modem.
	  • SgId: Service group ID of the modem.

MAC domain configuration	MAC domain configuration. • MAC Domain: MAC domain ID. The link opens the CMTS Mac Domain page. • cnBR: cnBR cluster. • Service Group ID: Service group ID. • Primary DS channels: Primary downstream channels for this MAC domain. • US channels: Upstream channels for this MAC domain.

Panel Name	Description
DS channel configs	Downstream channel configuration. • downstream channel ID: Downstream channel ID • annex: Annex of the downstream channel • AnnexA • AnnexB • frequency: Frequency of the downstream channel • modulation: Modulation of this downstream channel • qam64 • qam256 • interlevel: Interlever of the downstream channel • poweradjust: Power adjustment of the downstream channel.
OFDM channel configs	OFDM channel configuration. • ofdm chan id: OFDM channel ID • startfrequency: Start frequency • width: Width of the OFDM channel • plc: PHY Link Channel. • rolloff: Rolloff of the OFDM channel • profilencp: Profile Next Codeword Pointer • cyclicprefix: Cyclicprefix of the OFDM channel. • pilotScaling: Pilot Scaling • profiilecontrol: Profile control • interleaverdepth: Interlever depth • subcarrierspacing: Subcarrier Spacing • profiles: Link to OFDM Channel Profile Data page.

Panel Name	Description
US channel configs	Upstream channel configuration. • upstream channel id: Upstream channel ID • slotsize: Minislot size • frequency: Frequency • docsismode: DOCSIS Mode • modulation: Modulation profile • powerlevel: Power level • channelwidth: Channel width • sgid: Service group ID

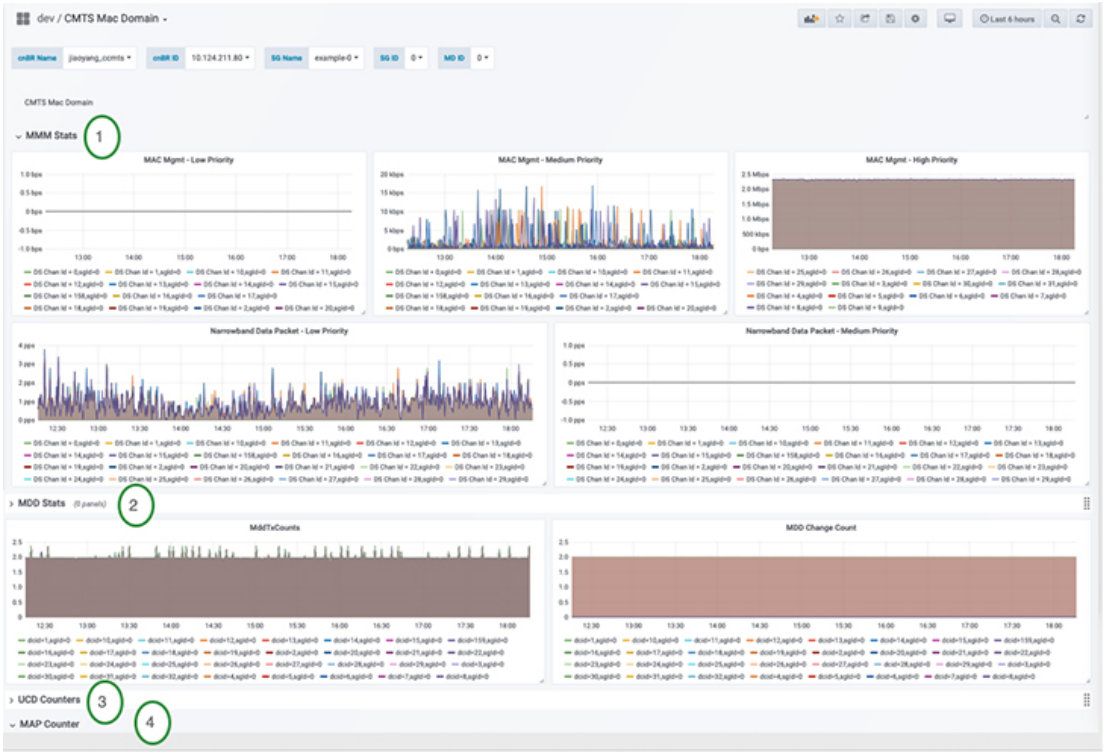


520877

CMTS MAC Domain

The Dashboard displays the following information:

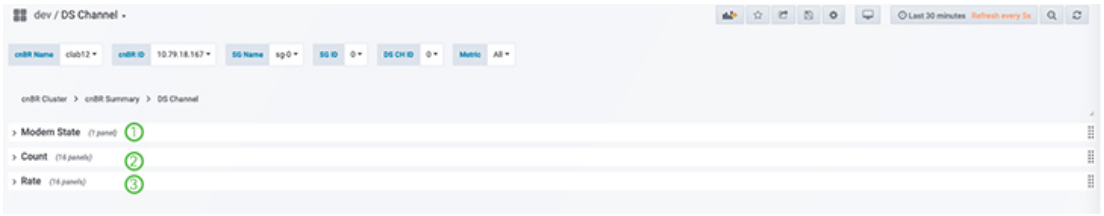
Panel Name	Description
MMM stats	Rate history of MAC management messages.
MDD stats	Rate history of MDD.
UCD counters	Number of upstream channel descriptors.
MAP counter	Number of Media Access Protocol (MAP) messages.



DS Channel

The Downstream (DS) Channel dashboard displays the following information:

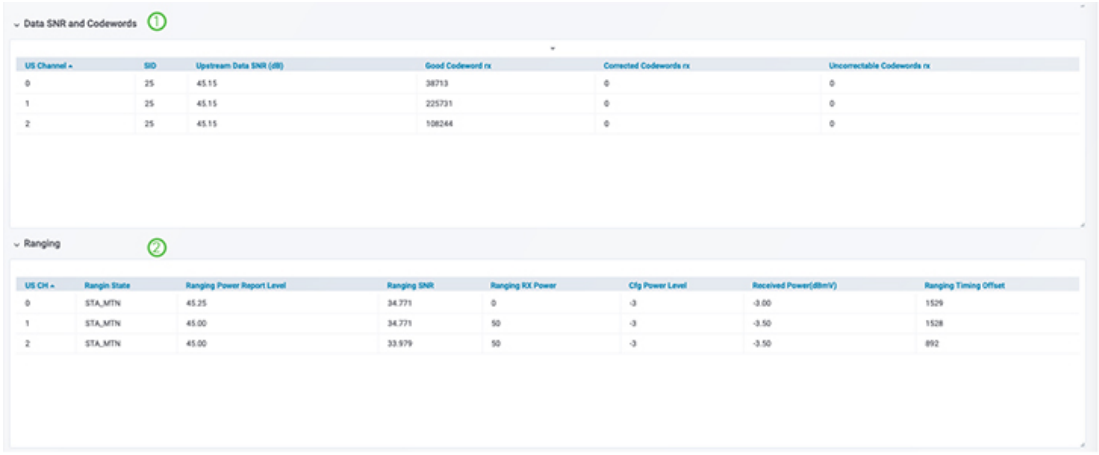
Panel Name	Description
Modem state	Summary of modems in different states for the specific Cisco cnBR cluster.
Count	Downstream channel count of the cable modem.
Rate	Downstream traffic rate for the Cisco cnBR cluster.

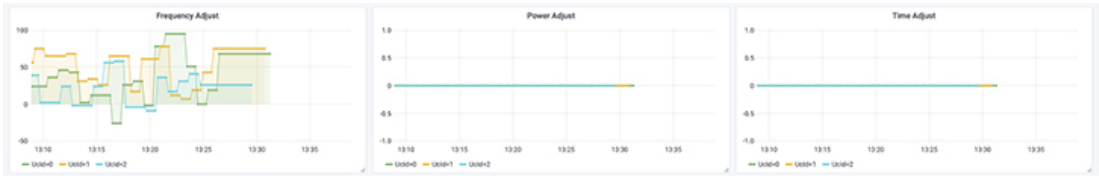


Modem US Channel

The Dashboard displays the following information:

Panel Name	Description
Data SNR and Codewords	SNR and Codeword information of the upstream channel for the cable modem.
Ranging	Ranging information of cable modems.



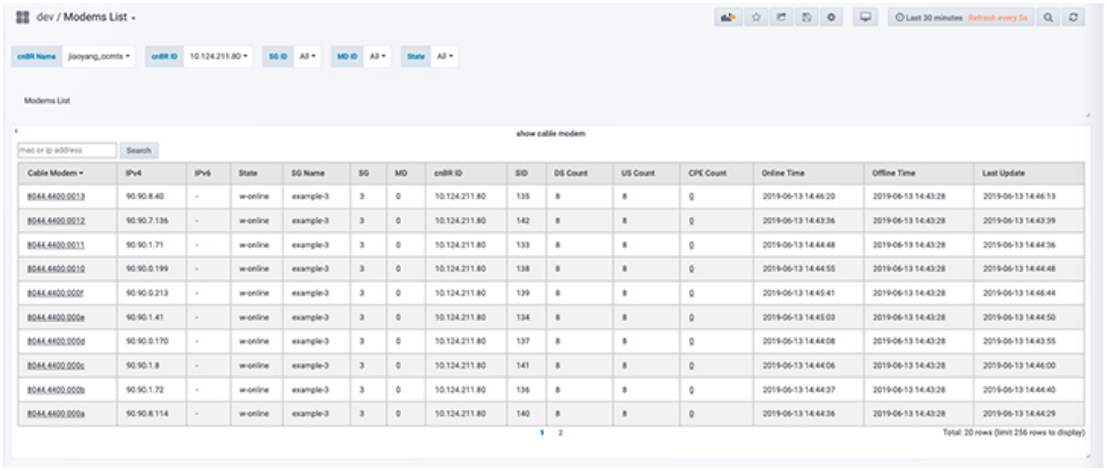


Modems List

The Dashboard displays all CMs in a list, based on the cluster, SG ID, MD, and the status:

Panel Name	Description
Download CSV	Download CSV for online and offline modems. You can download the CSV for one Cisco cnBR cluster at a time. A single CSV file for all clusters is not available for downloading.

Panel Name	Description
Modem List	Detailed modem state information list. Use the Search text box to filter modems by MAC address, IPv4 address, or IPv6 address. The number of rows in the Modem List table is limited to 256. Total is the total number of modems that are filtered by clusterIp and the search text. • Cable Modem: MAC address of the cable modem • IPv4: IPv4 address of the cable modem • IPv6: IPv6 address of the cable modem • State: State of the cable modem • SG Name: Service group name of the cable modem • SG: Service group ID of the cable modem • MD: MAC domain ID of the cable modem • cnBR ID: Cloud CMTS ID • SID: Service ID of the cable modem • DS Count: Downstream channel count of the cable modem • US Count: Upstream channel count of the cable modem • CPE Count: CPE count of the cable modem • Online Time: Time stamp when the modem became online • Offline Time: Timestamp when the cable modem transitioned to offline • Last Update: Last time when the entry was updated

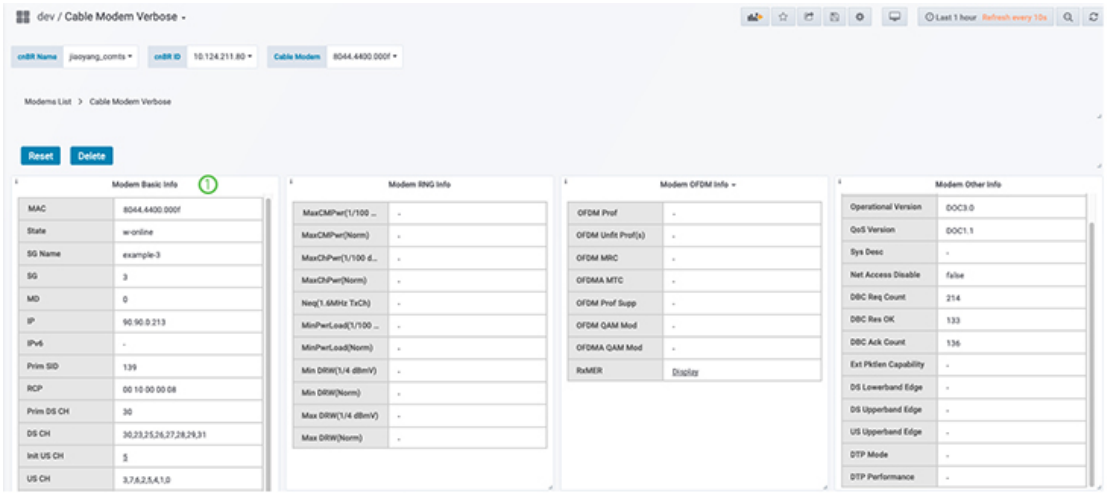


Cable Modem Verbose

The Dashboard displays the following information:

Panel Name	Description
Modem Basic Info	Basic information about the cable modem.
Modem RNG Info	Ranging information of the cable modem.
Modem OFDM Info	OFDM information of the cable modem.
Modem Other Info	Additional information of the cable modem: • DOCSIS Version: DOC1.0, DOC1.1, DOC2.0, DOC3.0, DOC3.1 • Operational Version: DOC1.0, DOC1.1, DOC2.0, DOC3.0, DOC3.1 • QoS Version: DOC1.0, DOC1.1 • Sys Desc: System description • DBC Req Count: Count of DBC(Dynamic Bonding Change) request • DBC Res OK: Count of DBC response with OK • DBC Ack Count: Count of DBC ack • Ext Pktlen Capability: External packet length capability • DS Lowerband Edge: Downstream lower band edge • DS Upperband Edge: Downstream upper band edge • US Upperband Edge: Upstream upper band edge • DTP Mode: DOCSIS time protocol mode • DTP Performance: DOCSIS time protocol performance
Modem State History	History of the cable modem status.
Modem CPE List	CPE list of cable modem. • MAC: CPE MAC address • IP: IP address of the modem • Device Class: CPE device class CPE device class: • EROUTER • EMTA • SMTA • ESTB • EDVA • ECM • EPS

Panel Name	Description
Modem Ping Stats	History of cable modem IP pings.
Service Flow Stats	Shows the details of upstream and downstream service flows.
Channel Stats	You can view the following details: • US CH RNG SNR History: History of upstream channel ranging SNR(Signal Noise Ratio). • US CH RNG RX Power History: History of upstream channel ranging RX power(dBm). • DS CH (RxPwr&SNR) History: History of downstream channel RX power and SNR by remote query. • US CH (TxPwr&TxTimingOffset) History: History of upstream channel TX power and timing offset by remote query. • Modem Timing Offset History: History of upstream channel timing offset of Cisco cnBR side. • Modem Ranging State History: History of upstream channel ranging state.
DBC Event	Shows the details of the Dynamic bonding change (DBC) events.
Resiliency Event	History of upstream resiliency state.

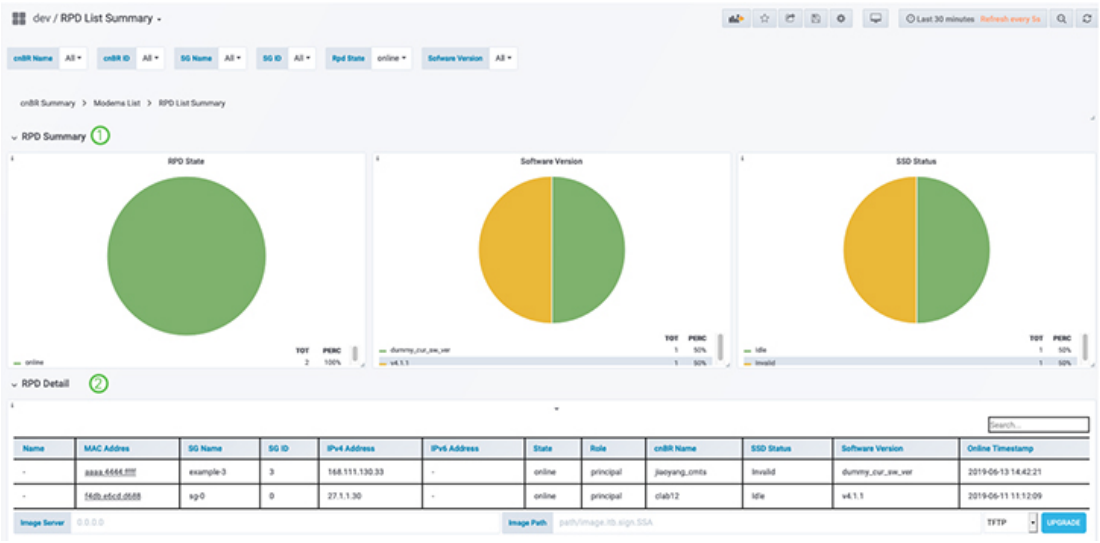


RPD List Summary

The Dashboard displays the following information:

Panel Name	Description
RPD Summary	Shows the following details using pie charts: • RPD state: States of the RPDs in the Cisco cnBR cluster • Software version: Software version running on the RPDs • SSD state: RPD secure software download status

RPD Detail	The table shows details of the RPDs:
	• Name: Name of the RPD
	• MAC: MAC address of the RPD
	• SG: Service group
	• SG Name: Service group name of the RPD.
	• SG ID: Service group ID of the RPD.
	• IPv4 Address: IPv4 address of the RPD.
	• IPv6 Address: IPv6 address of the RPD.
	• State: State of the RPD.
	  • online
	  • offline
	• Role: Role of the RPD.
	  • principal
	  • auxiliary
	• cnBR Name: Name of the cluster
	• SSD state: RPD secure software download status
	• Software version: Software version running on the RPDs
	• Online Time: RPD online timestamp



RPD Verbose

The Dashboard displays the following information:

Panel Name	Description
Basic information	Shows basic information about the RPD. • MAC: MAC address of the RPD • Name: Name of the RPD • SG: Service group • IPv4: IPv4 address • IPv6: IPv6 address • State: init(auth), init(gcp), init(clksync), init(l2tp), online, offline • GCP State: Generic control plane state • Role: principle, auxiliary • cnBR ID: Cloud CMTS ID • Last State: The previous status of RPD • Last GCP State: The previous generic control plane state • Auth Time: RPD authentication timestamp • Online Time: RPD online timestamp

Panel Name	Description
RPD identification	Shows the basic RPD identification detials. • RPD ID: RPD MAC address • Vendor Name: Vendor name • Vendor ID: Vendor ID • Model Number: Model number of the RPD • Sw Version: Current software version running on the RPD • Boot Rom Sw Version: Boot read-only memory software version • Device Description: Device description • Device Alias: Device alias • Serial Number: Serial number • Rcp Protocol Ver: R-PHY control protocol version • Rpd Rcp Protocol Ver: R-PHY control protocol version • Rpd Rcp Schema Version: R-PHY control protocol schema version • Hw Revision: Hardware revision • Asset Id: Asset ID of the RPD • Vsp Selector: Vendor-Specific Pre-configuration. • Us Burst Receiver Vendor Id: Upstream burst receiver vendor ID • Us Burst Receiver Driver Version: Upstream burst receiver driver version

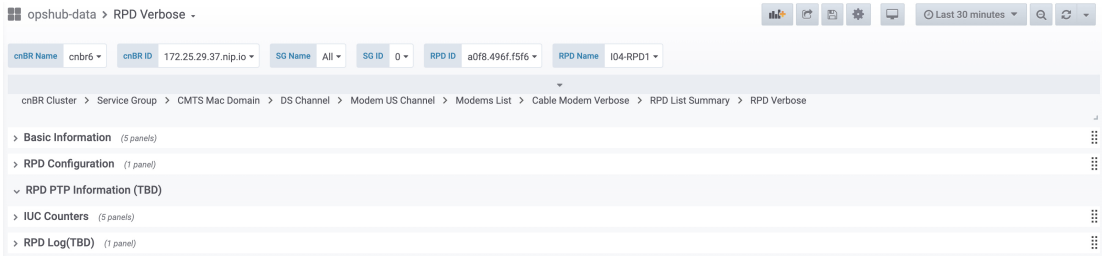
Panel Name	Description
RPD Capability	

Panel Name	Description
	Shows the basic capabilities. • RPD ID: RPD MAC address • Bi-direction RF Ports: Bi-directional radio frequency ports • DS RF Ports: Downstream radio frequency ports • US RF Ports: Upstream radio frequency ports • 10G Eth Ports: 10 Gigabit Ethernet port number • 1G Eth Ports: 1 Gigabit Ethernet port number • DS SC-QAM Channels Per Port: Downstream single carrier quadrature amplitude modulation (qam) channels per port • DS OFDM Channels Per Port: Downstream orthogonal frequency division multiplexing (OFDM) channels per port • US SC-QAM Channels Per Port: Upstream single carrier QAM channels per port • US OFDMA Channels Per Port: Upstream OFDM channels per port • DS SCTE-55-1 Channels Per Port: Downstream SCTE-55-1 channels per port • US SCTE-55-1 Channels Per Port: Upstream SCTE-55-1 channels per port • SCTE-55-2 Modules: SCTE-55-2 Modules • US SCTE-55-2 Demodulator Num: Upstream SCTE-55-2 demodulator numbers • NDF Channels Per Port: Remote-PHY narrowband digital forward channels per port • NDR Channels Per Port: Remote-PHY narrowband digital return channels per port • UDP Encapsulation On L2TPv3: User datagram protocol (UDP) encapsulation on layer 2 tunneling protocol version 3 • DS Distinct PSP Flows: Downstream distinct packet streaming (DPS) protocol flows • US Distinct PSP Flows: Upstream DPS protocol flows • Asyn MPEG Video Channels Per Port: Asynchronous MPEG video channels per port • Flow Tags support capability: Shows whether flow tags support is available or not.

Panel Name	Description
	• Freq Tilt support: Shows whether frequency tilt is supported on the RPD • Range of tilt setting: Range of tilt setting • ucd processing time: RPD upstream channel descriptor processing time • ucd change null grant time: RPD upstream channel descriptor change null grant time • Buffer depth monitor alert support: Buffer depth monitor alert support • Buffer depth config support: Buffer depth monitor configuration support • Multi section timing mer reporting support: Multiple section timing mer reporting support • Max DS Psp Seg Count: Max Downstream packet streaming protocol seg count • Direct DS Flow Queue Mapping: Direct downstream flow queue mapping • DS scheduler PhbId list: Downstream scheduler per hop behavior ID list • Pending EvRep Queue Size: RPD pending event report queue size • Local Event Log Size: RPD local event log size • Supported Optical Node RF: Shows whether optical node radio frequency is supported on the RPD • MAX DS Freq: RPD maximum downstream frequency • MIN DS Freq: RPD minimum downstream frequency • MAX Base Power: RPD maximum base power • MIN Tilt Value: RPD minimum tilt value • MIN Power Adjust for ScQam Chan: RPD minimum power adjust for single carrier quadrature amplitude modulation channels • MAX Power Adjust for ScQam Chan: RPD maximum power adjust for single carrier quadrature amplitude modulation channels • MIN Power Adjust for OFDM Chan: RPD minimum power adjust for orthogonal frequency division multiplexing channels • MAX Power Adjust for OFDM Chan: RPD maximum power adjust for orthogonal frequency division multiplexing channels

Panel Name	Description
Update history	Update history of the previous 50 RPDs. • MAC Address: MAC address of the RPD • State: Shows in which state the RPD is functioning: • init(auth) • init(gcp) • init(clksync) • init(l2tp) • online • offline • GCP State: offline, c1, c2 ready • TimeStamp: TimeStamp
show cable modem	The table provides basic details of the cable modem. • Cable Modem: Cable modem MAC address • IPv4: IPv4 address of the modem • IPv6: IPv6 address of the modem • State: Shows in which state the modem is functioning: • init(auth) • init(gcp) • init(clksync) • init(l2tp) • online • offline • SG: Service group • MD: MAC domain • cnBR ID: Cloud CMTS ID • Online Time: RPD online time • Offline Time: RPD offline time • Last Update: Last update time

Panel Name	Description
RPD configuration	The table shows the basic configuration of RPD. • Ds Channel: Downstream channel configuration • DsChan Base Power: Base power of downstream channel • DsChan Admin State: Admin state • Us Channel: Upstream channel configuration • Fiber Node: Fiber node configuration



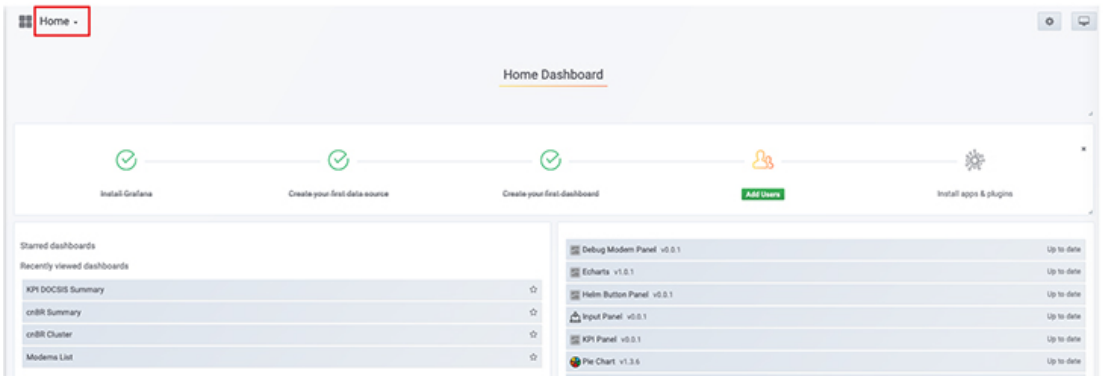
520876

Search for Dashboards

Follow this procedure to search for dashboards:

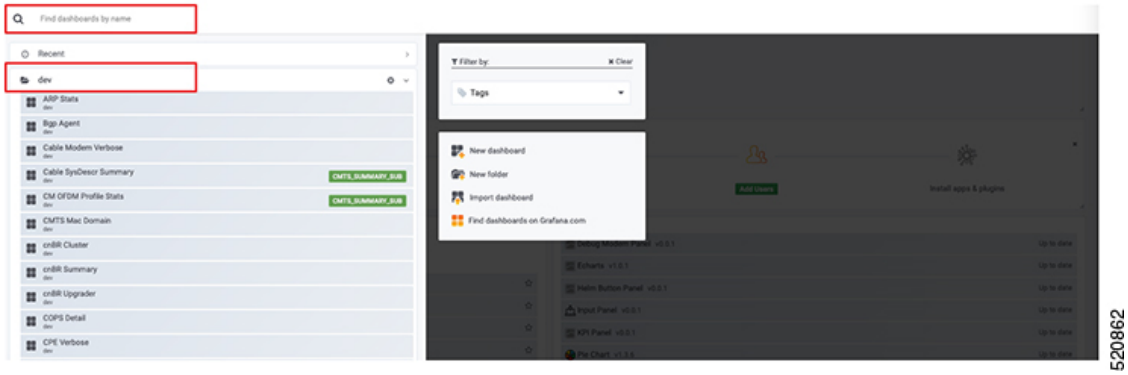
Step 1 In the **Metrics** tab, choose the Dashboard icon from the left pane.

This dashboard is the landing page.



520861

Step 2 Click **Home** on the top left corner.



Step 3 Enter the dashboard name in the **Find dashboard by name** text field.

Or click the `opshub-data` folder or `cee-data` folder and browse through it.

KPI Alert Management

KPIs (Key Performance Indicator) of Cisco cnBR clusters help in getting information on the overall system stability and on the components that are not functioning normally and impact the system stability.

The Cisco Operations Hub supports the following KPI Alert categories:

- Subscriber
- RF Plant
- Infra

Subscriber

This KPI Alert category provides an overview of the subscriber health status of the Cisco cnBR cluster. The following parameters are available in this KPI:

- CM state: Summary of the CM online status.
- CM ping: Summary of the reachability of cable modems and latency.
- US partial service: Summary of the CM upstream partial service state.
- DS partial service: Summary of the CM downstream partial service state.
- RPD online state: Summary of the RPD online status.

RF Plant

This KPI Alert category provides an overview of the RF plant health status of the Cisco cnBR cluster. The following parameters are available in this KPI:

- CM DS SNR (Signal to Noise Ratio): If 5 percent of modem's downstream primary channel's SNR is less than 30 dB, the downstream channel is unhealthy.
- CM US ranging SNR (Signal to Noise Ratio): If 5 percent of modem's upstream channel's SNR is less than 20 dB, the upstream channel is unhealthy.
- CM US data SNR (Signal to Noise Ratio): If 5 percent of modem's upstream channel's SNR is less than 30 dB, the upstream channel is unhealthy.

Infra

This KPI Alert category provides an overview of the pod CPU and memory health status of the Cisco cnBR cluster and the Cisco Operations Hub cluster.

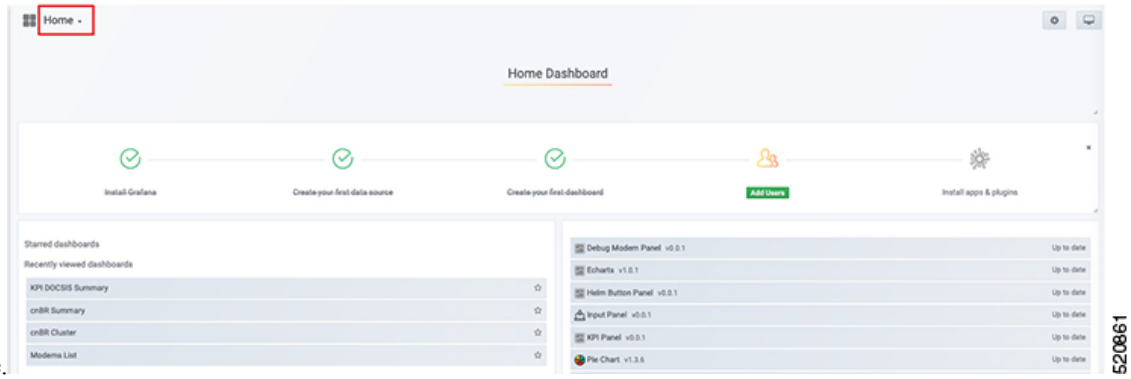
Alert Management

The Cisco Operations Hub Alert Management is based on the KPIs. Cisco cnBR records all alerts for users to view. You can configure the alert-manager to manage alerts.

Follow this procedure to view the KPI Alert Management page:

Step 1 In the **Metrics** tab, choose the **Dashboard** icon from the left pane.

This dashoboard is the landing page.



Step 2 Click **Home** on the top left corner.

Step 3 Click the **opshub-data** folder and select **KPI Alert Management**.

Alert Definition

All alerts are built based on the KPI metrics and divided into several alert groups. Each KPI metric generates one alert that belongs to a predefined alert group. For example, KPI metrics: CM state (Summary of CM online state) generates one alert that is named CMNotHealthy, which is part of the Subscriber alert group. The Alert Management supports the following alerts and alert groups.

Group	Alert
Subscriber	• CMNotHealthy • RPDNotHealthy
RF	• USCHNotHealthy • DSCHNotHealthy
Infra	• NodeNotHealthy • PodNotHealthy

Alert Record

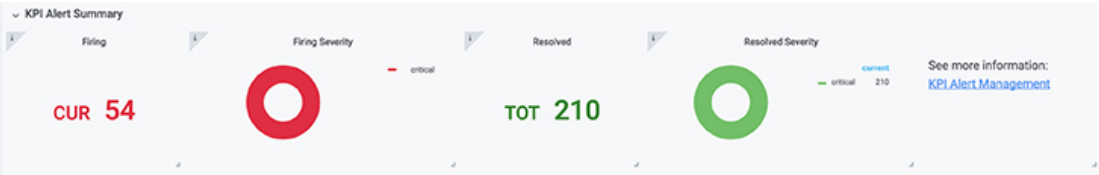
Alert Management records all alerts that are generated in the Cisco cnBR router. The dashboards display an alert summary and detailed information about those alerts.

KPI Alert Summary

Alert summary dashboards show the number of current alerts and total resolved alerts. In addition, the dashboards display the distribution of alerts based on severity. Cisco cnBR supports two levels of alerts:

- critical
- warning

You can view this pane on the Cisco cnBR cluster dashboard under **Dashboards** > **Manage** > **opshub-data**.



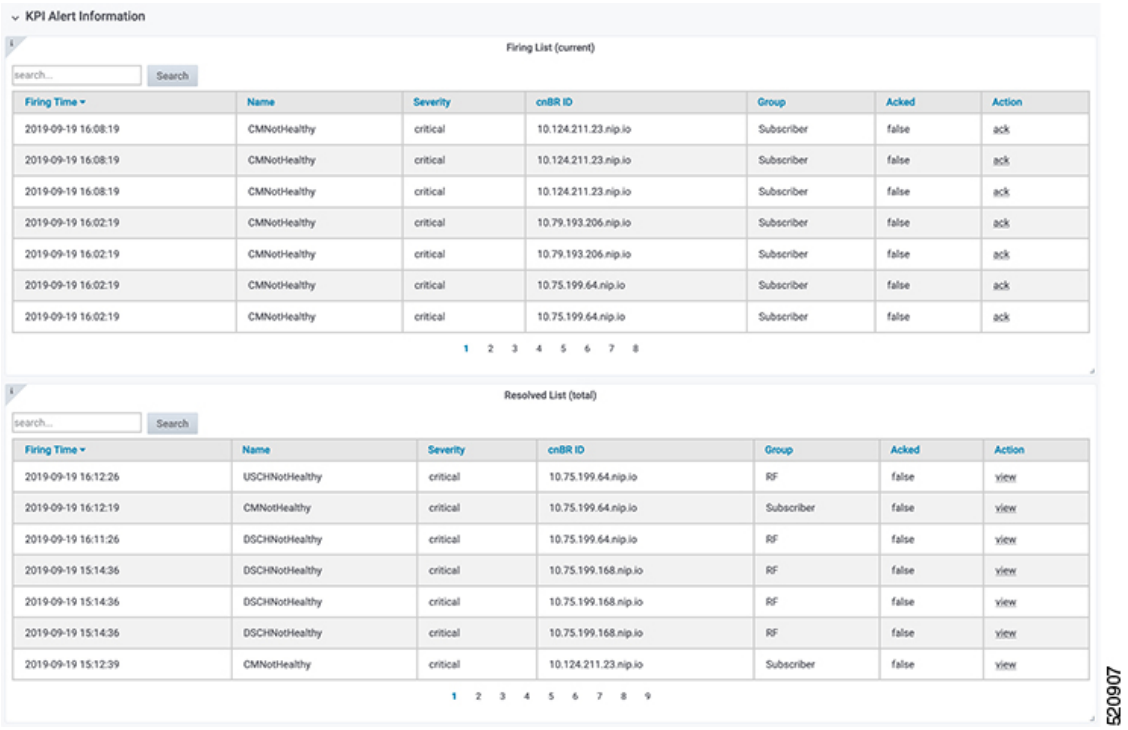
KPI Alert Information

The dashboard shows two lists of Firing Alerts and Resolved Alerts. The following details are available in these tables:

Panel Name	Description
Firing Time	Alert fired time.
Name	Alert name.
Severity	Critical or warning.
cnBR ID	Cisco cnBR where the alert is fired.
Alert Group	Category of the KPI alert.

Panel Name	Description
Acknowledge status	Shows whether acknowledged or not.
Action	Acknowledge or view an alert. Click the **View** link. The **Alert Action** pane on the right side shows the details of the alert.

You can view details and acknowledge firing alerts. For the resolved alerts, you can view the details of each alert.



Acknowledge KPI Alert

You can acknowledge the firing alerts. By default, every three hours, you are notified about the firing alerts by email. You can stop receiving the alert emails by setting the silence time, creator, and comments.

Configure Alerts

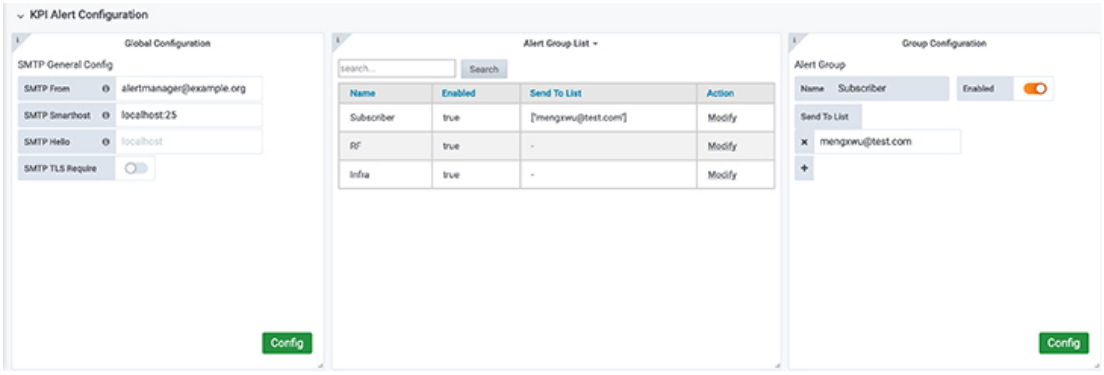
You can configure global alerts through Alert Management. For global configuration, update the SMTP (Simple Mail Transfer Protocol) settings. By default, this option is disabled.

On the **Global Configuration** pane, configure the notification channel. The **SMTP General Config** pane is available under the **KPI Global Configuration** pane of the **KPI Alert Configuration** dashboard.

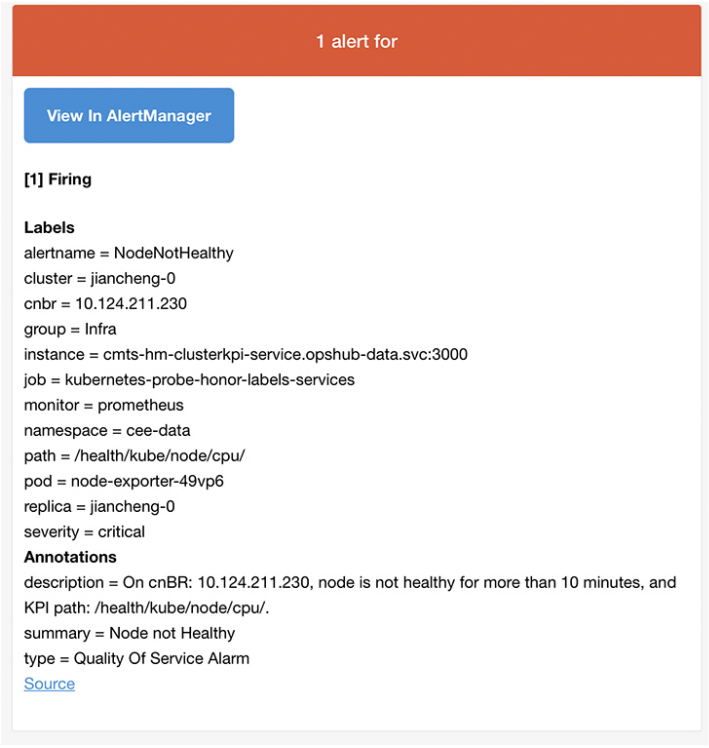
Field	Description
SMTP From	The default **SMTP From** header field.
SMTP Smarthost	The default SMTP smarthost used for sending emails, including the port number. The port number is 25 or 587 for SMTP over TLS (STARTTLS). Example: `smtp.example.org:587`
SMTP Hello	The default hostname to identify to the SMTP server.
SMTP TLS Require	The default SMTP TLS requirement (Default: false).

KPI Alert Configuration

You can enable or disable an alert group and add or delete email addresses of receivers for each alert group. When you enable an alert group and add email addresses, those users are notified when an alert is generated in the respective group.



KPI Alert Email

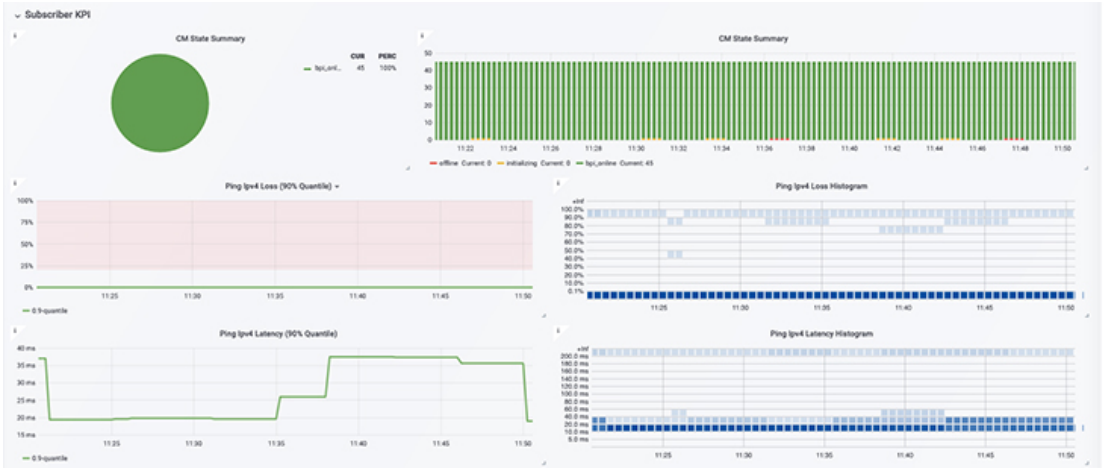


Monitor and Troubleshoot

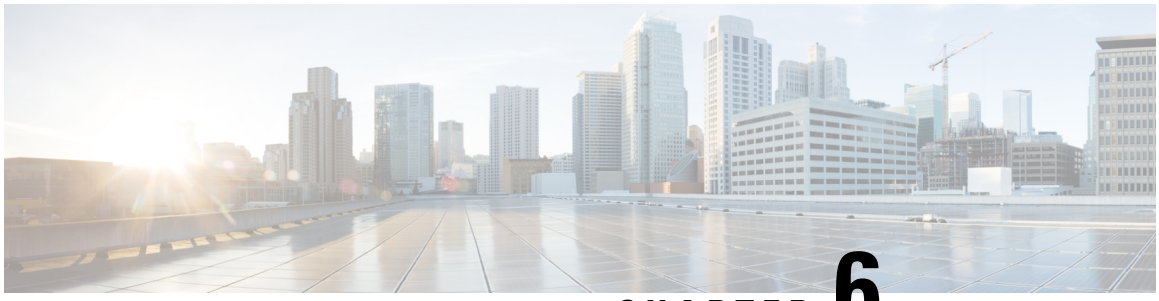
The KPI of one Cisco cnBR-Core is displayed on this dashboard.

The state of each category is displayed on this dashboard.

This dashboard shows the details of the subscriber.



This dashboard shows the details of the RF.



CHAPTER 6

Operations of Cisco Cloud Native Broadband Router

Cisco cnBR supports day-to-day operations of the Data-over-Cable Service Interface Specifications (DOCSIS) system from the deployment to the monitoring for optimal operations. This chapter describes utilities to facilitate typical workflows during such operations.

RPD Cutover

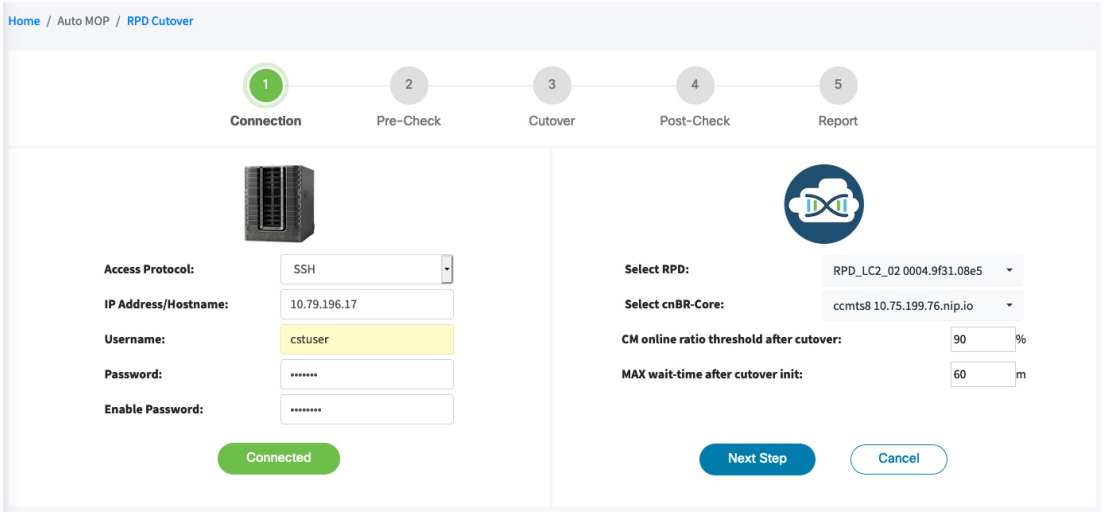
Cisco Operations Hub supports Remote PHY Device (RPD) Cutover process through a GUI-based step-by-step wizard to facilitate the moving of RPDs from an existing cBR-8 system to Cisco cnBR. RPD cutover assumes Cisco cnBR is operational and the related service group (SG) configuration is ready before running this wizard. The wizard focuses on step-by-step instructions, preparation, and post verification including CM, CPE, and RF signal.

Prerequisites for RPD Cutover

- Target RPD is online and connected to cBR-8 system.
- Cisco cnBR is operational and healthy.
- Target SG is configured and verified on Cisco cnBR.
- SG configuration has the correct Secure Software Download (SSD) configuration.
- The SSD Image is in the TFTP or HTTP server.
- Network connectivity between Cisco cnBR and target cutover RPD is available.

Perform RPD Cutover from cBR-8 to Cisco cnBR

Go to **RPD Manage** > **RPD Cutover** menu of the horizontal navigation tab Configurator to start RPD cutover. The wizard presents fields for entry in sets on a progressive series of interface pages.



Step 1 Set target details at the Connection page.

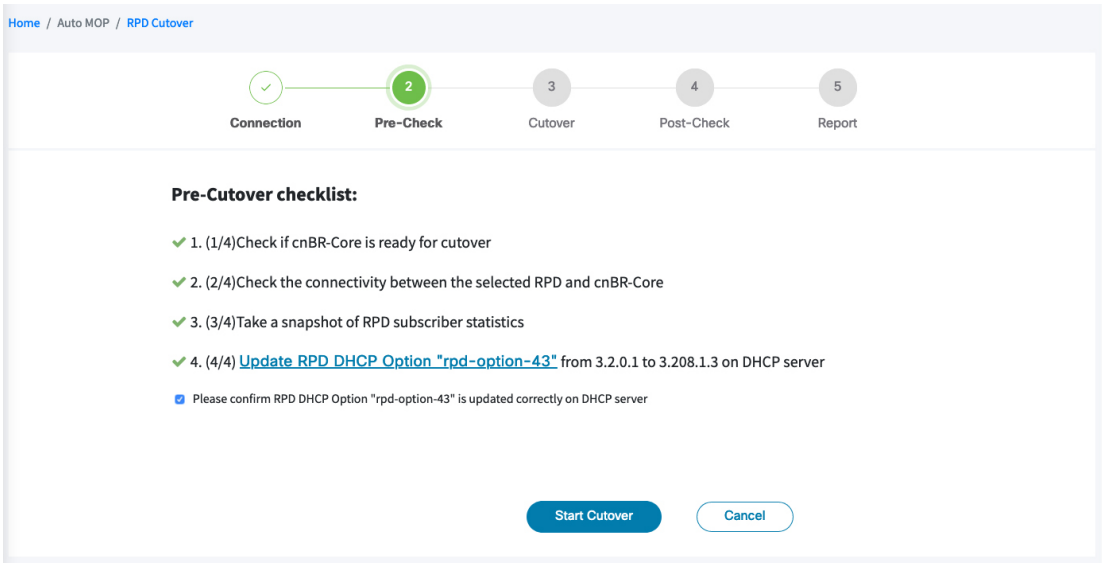
a) Select or input original cBR-8 Access Protocol, IP Address, Username, Password, and Enable Password.

b) Click **Connect to cBR-8** to see all RPDs on the cBR-8 available for selection.

c) Select the target RPD to cutover and target Cisco cnBR.

d) Fill in expected values for the `CM online ratio threshold after cutover:` (e.g. 95) and `Max wait-time after cutover init:` (e.g. 40) fields.

e) Click **Next Step**.



Step 2 Get prechecks done at the Pre-Check page.

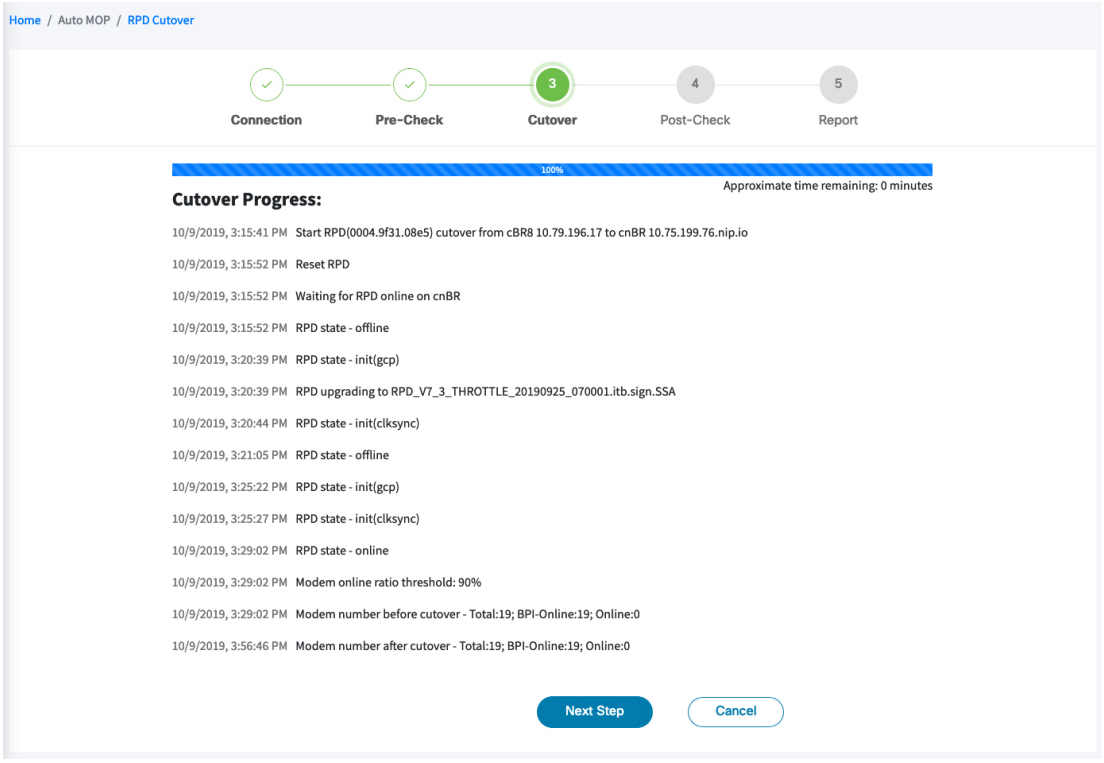
a) This step automatically performs prechecks. If a check fails, correct the failure, and click the rerun icon to rerun the precheck.

b) After all prechecks are complete, manuall update the RPD CCAP-CORE-List from cBR-8 to Cisco cnBR at the DHCP server.

c) If the same CM/CPE IP scopes need to move from the original cBR-8 to Cisco cnBR on DHCP server, move it.

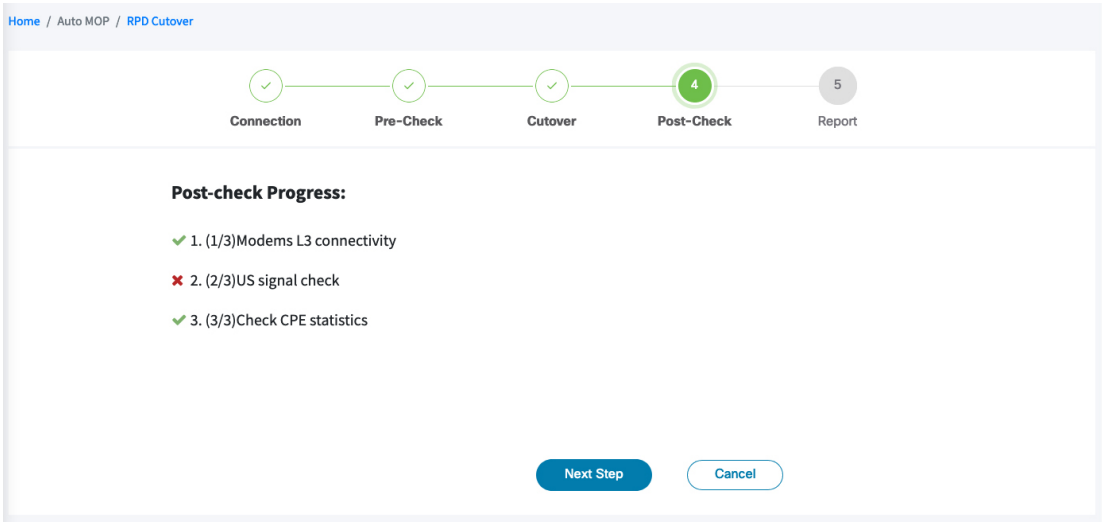
d) Check the confirm checkbox when it is complete.

e) Click **Start Cutover** to start the RPD cutover process.



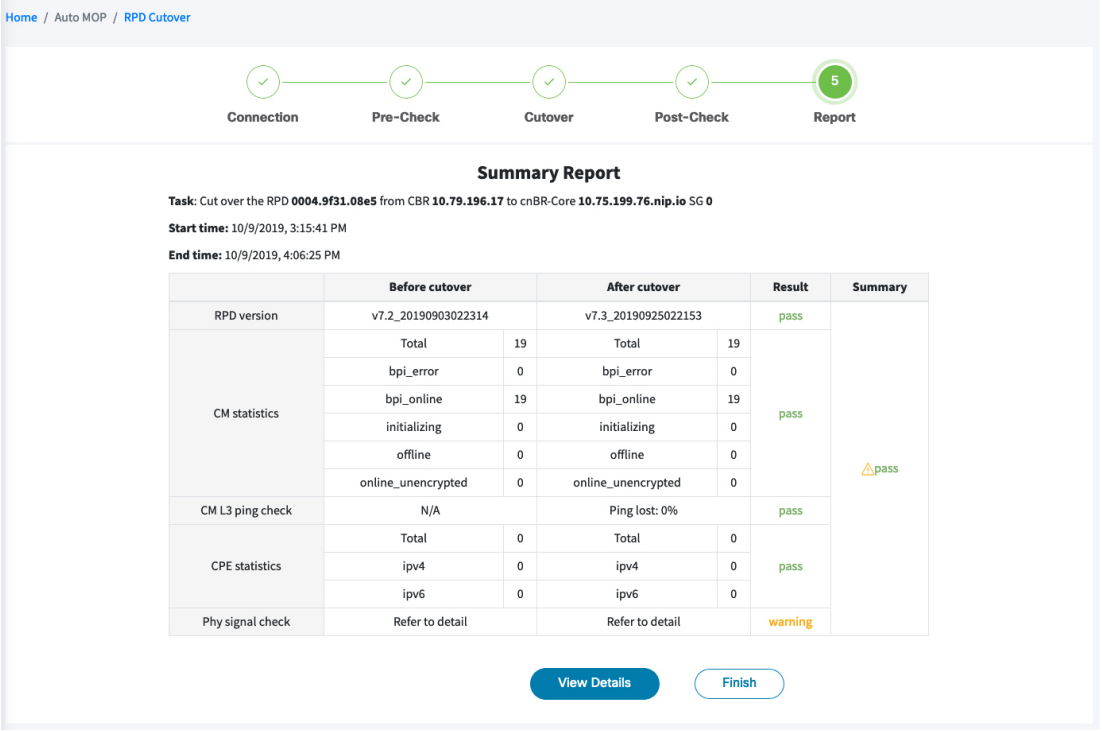
Step 3 View progress and summary information at the Cutover page.

a) Click **Next Step** when the RPD successfully comes online on Cisco cnBR and the target Cable Modem ratio comes online within the time specified in the MAX wait-time field.



Step 4 Get postchecks done at the Post-Check page.

a) After these checks are complete, click **Next Step**.



Step 5 View the Summary Report at the Report page. If there is any issue during this task, click **View Details** to get more information. Rollback tips are available in case you want to rollback when the cutover is not successful.

RPD Operations

Cisco Operations Hub allows you to add, delete, and replace RPDs serviced by a Cisco cnBR using the Configurator interface. This section provides step-by-step instructions to add, delete, and replace RPDs.

Add RPDs

Before you begin

- Use an existing template or create new SG and L3 templates. See Configure cnBR using Configurator, on page 48 for steps to create new SG templates and L3 templates.
- Have the MAC address of the target RPD at hand.

Step 1 Log in to the Cisco Operations Hub.

Step 2 Click **Configurator** from the horizontal navigation tab.

Step 3 Click **RPD Add** under **RPD Manage** from the vertical navigation tab.

Step 4 Select the Cisco cnBR and enter the target RPD information. cnBR, RPD MAC, RPD Name, SG Name, SG Template, and L3 Template are required fields. Click **Next Step**.

Step 5 Wait for the Cisco Operations Hub to complete the automated pre-checks. Check the box at the bottom of the checklist to confirm that RPD is physically connected. Click **Next Step**.
The Cisco Operations Hub starts to add the RPD. The progress is displayed in this page.

Step 6 If the operation is successful, click **Next Step**. If it fails, the Cisco Operations Hub displays a message to let you know that this step has failed; skip to Step 8.

Step 7 Wait for the Cisco Operations Hub to perform the automated post-checks. Click **Next Step**. (If Step 6 fails, the Cisco Operations Hub skips this step.)

Step 8 The Cisco Operations Hub displays a summary report that has the results of the operation.

What to do next

If adding the RPD fails at Steps 6 or 7, the Cisco Operations Hub displays the error information. Use this information to diagnose and correct the problem. After you fix the problem, restart the **RPD Add** operation from Step 1. If the problem resolution requires a change in the configuration, delete and add the RPD again with the updated configuration.

Delete RPDs

Before you begin

- Have the MAC address of the target RPD at hand.

Step 1 Log in to the Cisco Operations Hub.

Step 2 Click **Configurator** from the horizontal navigation tab.

Step 3 Click **RPD Delete** under **RPD Manage** from the vertical navigation tab.

Step 4 Select the Cisco cnBR and the target RPD and click **Next Step**.

Step 5 Wait for the Cisco Operations Hub to check if the target cnBR is ready. Check the box to confirm you want to delete the RPD and click **Start Delete**.
The Cisco Operations Hub starts to delete RPD. The progress is displayed in this page.

Step 6 If the operation is successful, click **Next Step**. If it fails, the Cisco Operations Hub displays a message to inform you that this step has failed; skip to Step 8.

Step 7 Wait for the Cisco Operations Hub to perform the automated post-checks. Click **Next Step**. (If Step 6 fails, the Cisco Operations Hub skips this step.)

Step 8 The Operations Hub displays a summary report that has the results of the operation.

What to do next

If deleting the RPD fails at Steps 6 or 7, the Cisco Operations Hub displays the error information. Use this information to diagnose and correct the problem. After you fix the problem, restart the **RPD Delete** operation from Step 1.

Replace RPDs

Before you begin

- Have the MAC addresses of the target RPD and the new RPD at hand.

Step 1 Log in to the Cisco Operations Hub.

Step 2 Click **Configurator** from the horizontal navigation tab.

Step 3 Click **RPD Replace** under **RPD Manage** from the vertical navigation tab.

Step 4 Select the Cisco cnBR and the existing RPD to be replaced. Enter the MAC address of the new RPD and the maximum wait time for the new RPD to be online. Click **Next Step**.

Note New RPDs need at least 12 minutes to come online.

Step 5 Wait for the Cisco Operations Hub to complete automated pre-checks. Check the box at the bottom of the checklist to confirm that the RPD is physically connected. Click **Start Replace**.
The Cisco Operations Hub starts to replace RPD. The progress is displayed in this page.

Step 6 If the operation is successful, click **Next Step**. If it fails, the Cisco Operations Hub displays a message to inform you that this step has failed; skip to Step 8.

Step 7 Wait for the Cisco Operations Hub to perform the automated post-checks. Click **Next Step**. (If Step 6 fails, the Cisco Operations Hub skips this step.)

Step 8 The Cisco Operations Hub displays a summary report that has the results of the operation.

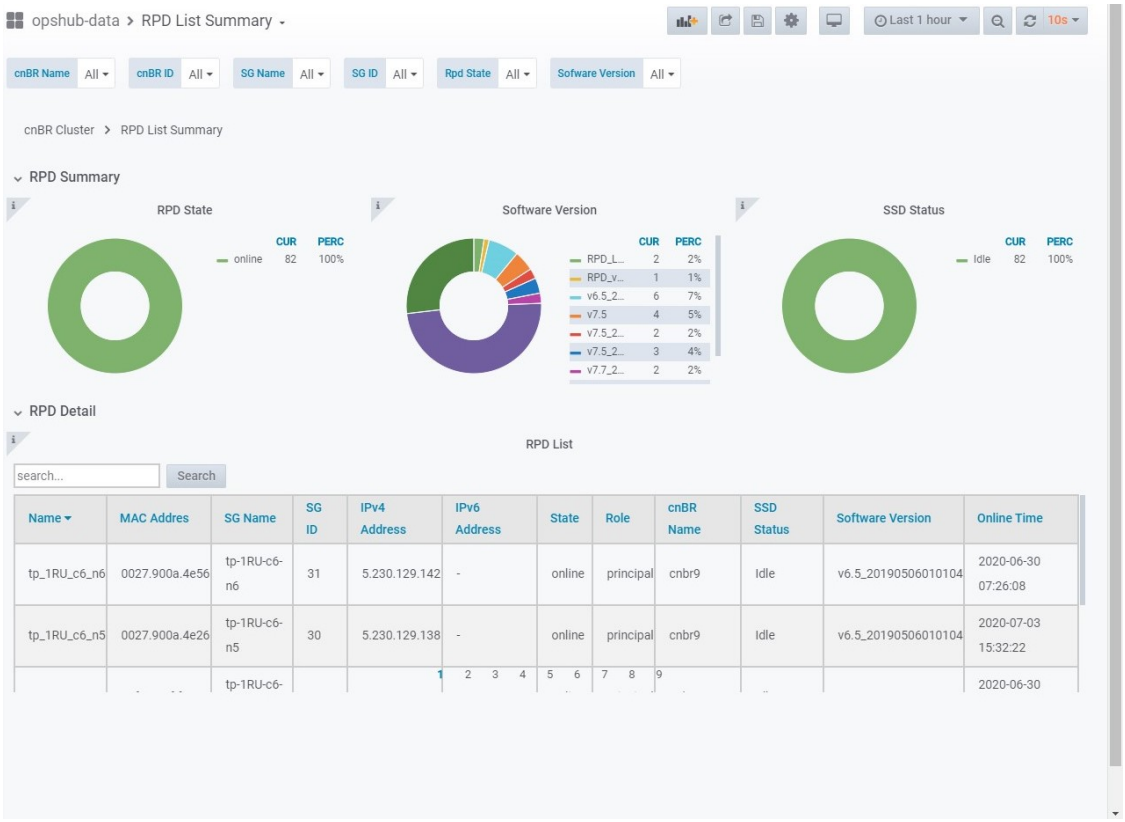
What to do next

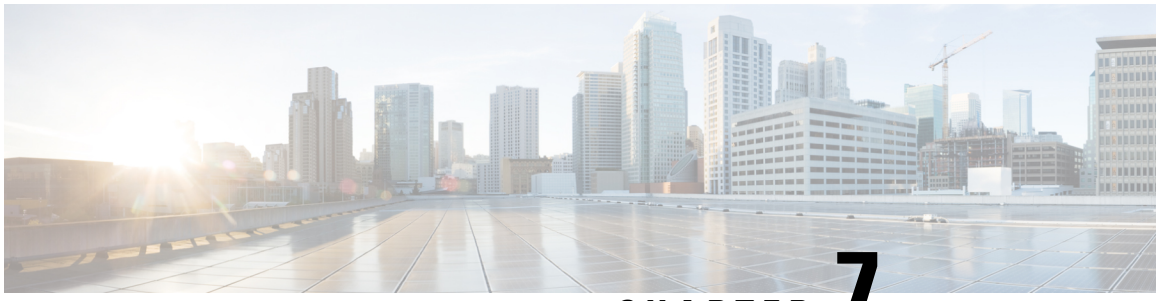
If replacing the RPD failed at Steps 6 or 7, the Cisco Operations Hub displays the error information. Use this information to diagnose and correct the problem. After you fix the problem, restart the **RPD Replace** operation from Step 1.

Monitor RPDs

After you add or replace an RPD serviced by a Cisco cnBR, you can use the RPD List Summary Dashboard of the Cisco Operations Hub to monitor the status of the RPD. The following snapshot shows this dashboard with a sample RPD List Summary.

Figure 45: RPD List Summary Dashboard





CHAPTER 7

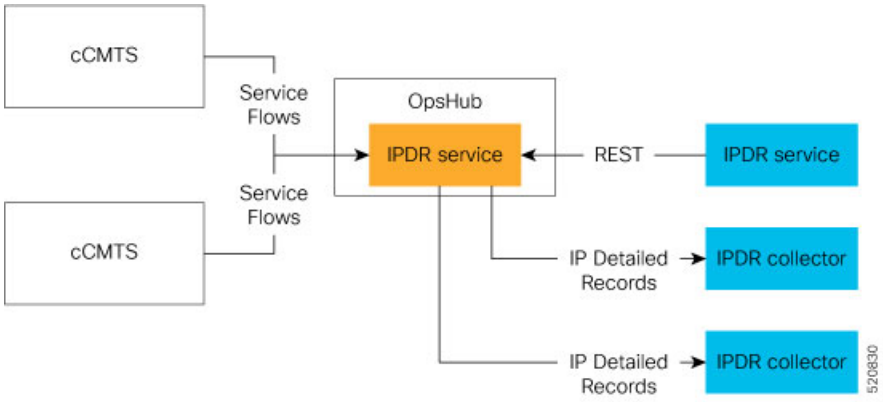
External Interfaces Support for Cisco Cloud Native Broadband Router

Cisco cnBR supports legacy interface translation, general network management, and monitoring information.

Cisco cnBR supports the following external interfaces:

IP Detail Record Service

The Cisco Operations Hub hosted IP Detail Record (IPDR) service provides the mechanisms to export IP detailed records to IPDR collectors and the ability to configure the IPDR service.



The Cisco Operations Hub IPDR service operates in a similar way as other Cisco Cable Modem Termination Systems (CMTS) products. You can configure it through the REST interface. See IPDR Streaming Protocol on the Cisco CMTS Routers for reference.

The URL `https://{Hostname}/api/ipdr` is created for the IPDR service, which is used for the REST configuration and status requests. The collector connects to the IPDR service on default port 4737 to establish a TCP session. Then, IPDR records are streamed from the IPDR service to the collector over this TCP session.

For the IPDR service to deliver records, the IP address of the collector that receives the records is required. An ordered list of collectors is contained in a session. Only one collector in a session receives the records, the

others are available as backup. The session describes the delivery mechanism and record format. You can define multiple sessions so that more than one collector can receive IPDR records from Operations Hub.

Terminology

Term	Description
Collector	The host that receives (collects) the IPDR records.
Exporter	The IPDR service includes an exporter service that generates the IPDR records.
Session	Describes the set of collectors and templates that are used to send IPDR records. At a time, only one collector in a session gets IPDR data at a time based on a priority order. If a collector is unavailable, the collector with the next highest priority gets the records.
Template	Identifies the record format for sending the records.

Configure IPDR Service

To configure the IPDR service, use a single command to set all configuration parameters in JSON format in one single action. This configuration method overwrites the existing configuration and activates the new configuration.

```
/v1/config
```

Note `/ipdr/config` is deprecated but usable.

To set the configuration, use the **PUT** HTTP method as shown in the following example.

```
curl -k -X PUT -H "Content-Type: application/json" -d @- << EOF
https://{Hostname}/api/ipdr/v1/config
{json_string}
EOF
```

Note Parameter -k allows insecure server connections when using SSL.

Example: Add or change IPDR configuration.

```
curl -k -X PUT -H "Content-Type: application/json" -d @- << EOF
https://192.168.0.1.nip.io/api/ipdr/v1/config
{
    "sessions": [
        {
            "id": 1,
            "name": "session_1",
            "description": "IPDR Session 1",
            "type": {
                "type": "time-interval",
                "interval": 15
            },
            "templates": [
                {
                    "template-type": "SAMIS-TYPE1"
```

```
                }
            ],
            "associated-collectors": [
                {
                    "collector-name": "Collector1",
                    "priority": 1
                }
            ]
        },
        {
            "id": 2,
            "name": "session_2",
            "description": "IPDR Session 2",
            "type": {
                "type": "event",
                "interval": 0
            },
            "templates": [
                {
                    "template-type": "DS-UTIL"
                }
            ],
            "associated-collectors": [
                {
                    "collector-name": "Collector1",
                    "priority": 1
                }
            ]
        },
        {
            "id": 3,
            "name": "session_3",
            "description": "IPDR Session 3",
            "type": {
                "type": "event",
                "interval": 0
            },
            "templates": [
                {
                    "template-type": "US-UTIL"
                }
            ],
            "associated-collectors": [
                {
                    "collector-name": "Collector1",
                    "priority": 1
                }
            ]
        }
    ],
    "collectors": [
        {
            "name": "Collector1",
            "address": "10.0.0.1",
            "nat-address": "0.0.0.0",
            "port": 0
        }
    ],
    "exporter": {
        "ack-timeout": 60,
        "keep-alive": 300,
        "max-unacked": 200,
        "started": true
    },
```

```
    "utilization": {
        "interval": 300
    }
}
EOF
```

After setting the configuration, you can use the **GET** HTTP method as shown in the following example to display the consolidated configuration:

```
curl -H 'Content-Type: application/json' -X GET https://{Hostname}/api/ipdr/v1/config
```

Example: Display the existing IPDR configuration

```
curl -k -H 'Content-Type: application/json' -X GET
https://192.168.0.1.nip.io/api/ipdr/v1/config
{
    "sessions": [
        {
            "id": 1,
            "name": "session_1",
            "description": "IPDR Session 1",
            "type": {
                "type": "time-interval",
                "interval": 2
            },
            "templates": [
                {
                    "template-type": "SAMIS-TYPE1"
                }
            ],
            "associated-collectors": [
                {
                    "collector-name": "Collector1",
                    "priority": 1
                }
            ]
        },
        {
            "id": 2,
            "name": "session_2",
            "description": "IPDR Session 2",
            "type": {
                "type": "event",
                "interval": 0
            },
            "templates": [
                {
                    "template-type": "DS-UTIL"
                }
            ],
            "associated-collectors": [
                {
                    "collector-name": "Collector1",
                    "priority": 1
                }
            ]
        },
        {
            "id": 3,
            "name": "session_3",
            "description": "IPDR Session 3",
            "type": {
                "type": "event",
                "interval": 0
            },
```

```
            "templates": [
                {
                    "template-type": "US-UTIL"
                }
            ],
            "associated-collectors": [
                {
                    "collector-name": "Collector1",
                    "priority": 1
                }
            ]
        }
    ],
    "collectors": [
        {
            "name": "Collector1",
            "address": "10.0.0.1",
            "nat-address": "0.0.0.0",
            "port": 0
        }
    ],
    "exporter": {
        "ack-timeout": 60,
        "keep-alive": 300,
        "max-unacked": 200,
        "started": true
    },
    "utilization": {
        "interval": 240
    }
}
```

Example: Remove IPDR configuration

```
curl -X PUT -H "Content-Type: application/json" https://192.168.0.1.nip.io/api/ipdr/v1/config
```

Note The IP address 192.168.0.1 is only for illustrative purposes. Use the IP address of the Cisco Operations Hub deployed at your site.

Fields In JSON

This table lists the fields used in JSON and their description.

Field Name	Description	Type	Enforcement
ack-timeout	Exporter timeout, after which an acknowledgement is received from the collector before retry.	Number. 5–60 seconds; the default value is 60.	Optional
address	The IP address of the collector, which is used to receive the IPDR records.	IP Address	Required
collector-name	A specific collector definition for collectors.	String	Required
collectors:name	Unique name used to identify a collector.	String	Required
description	Long descriptive text.	String	Required

Field Name	Description	Type	Enforcement
id	A unique session number for the purpose of reference.	Number	Required
interval	The interval used to send DS-UTIL and US-UTIL data.	Number in seconds, **0** means disabled.	Optional
keep-alive	The keepalive time after which the collector is considered unavailable.	Number. 5–300 seconds; the default value is 300.	Optional
max-unacked	The maximum number of unacknowledged records.	Number. 5–200; the default value is 200.	Optional
name	Descriptive name for reference purposes.	String	Required
nat-address	The NAT IP address of the collector.	IP Address	Optional
port	The port of the collector.	Number	Optional
priority	The order to use the collector. Use the collector with the lowest priority number first.	> 0	Required
started	Start the IPDR service or not.	Boolean	Required
type:type	The method used to request data from the service.	String. Possible values: adhoc, event, time-interval	Required
type:interval	The frequency of sending the data for a session.	2–1440 minutes.	Required only if "type:type" field is set to "time-interval".
template-type	Identifies the records format.	String. Possible values: SAMIS-TYPE1, US-UTIL, DS-UTIL, TOPOLOGY	Required

REST Return Codes

You can use the status codes listed in the following table to convey the results of a request.

Code	Short Description	Response Text	Actions
400	HTTP_BAD_REQUEST	• Failure: request format error. • Failed to configure session when exporter starts, stop it at first.	Confirm that the format of the request is valid or restart the IPDR service to apply new sessions.
404	HTTP_NOT_FOUND	• Failure: collector doesn't exist.	Return this code when adding a session referring to a collector that does not exist. If it is a consolidated configuration request, correct the request to include a valid collector.

Code	Short Description	Response Text	Actions
500	HTTP_BAD_REQUEST	• Failed to add new session to cache. • Failed to apply IPDR configuration. • Failed to config ipdr session to exporter. • Failed to get ipdr sessions with internal error. • Failed to recover configurations. • Failed to remove session in cache. • Failed to revert session in cache when db failed. • Failed to update IPDR configuration. • Failure: allocate JSON object error. • Failure: get ipdr config information error. • Failure: not enough memory. • Failure: save global cfg error. • IPDR configuration not updated, restored to original.	Internal error that requires engineering team engagement.
503	HTTP_SER_UNAVAIL	• not ready	Use this code only in response to readiness check. If the service is not ready, confirm that the Cassandra database is ready. Otherwise, get the database ready. If the Cassandra database is ready and operational, ask for customer support.

Monitor

Use the **GET** HTTP method of the following REST APIs to monitor the status of the IPDR session, collector, and exporter.

Monitor Session Status

- Get all sessions status.

  ```
  /v1/sessions
  ```

- Get specific session status.

  ```
  /v1/sessions/{id}
  ```

Note `/ipdr/session/status` is deprecated but usable.

Example:

```
curl -k -H 'Content-Type: application/json' -X GET
https://192.168.0.1.nip.io/api/ipdr/v1/sessions
Session ID: 1, Name: samis, Descr: samis, Started: True
Session Type: Time Interval (15 minutes).
Expires in 81 seconds.
Exporting not started.
2019-05-29T05:08:14 Statistics:
 Transmitted 0 Acknowledged 0 Enqueued 0 Lost 0
 queuedOutstanding 0 queuedUnacknowledged 0
1 Collectors in the session:
 Name: collector1, IPAddr: 10.0.0.1, Port: N/A, Priority: 1[DISCONNECTED]
Templates in the session:
 Template ID: 2, Name:
http://www.cablelabs.com/namespaces/DOCSIS/3.0/xsd/ipdr/DOCSIS-SAMIS-TYPE-1/DOCSIS-SAMIS-TYPE-1_3.5.1-A.1.xsd,
 Type: SAMIS-TYPE-1, KeyNumber: 28
Session 1 has a total of 1 templates.
Session ID: 2, Name: cmts-ds-util-stats, Descr: cmts-ds-util-stats, Started: True
Session Type: Event Based.
2019-05-29T05:08:14 Statistics:
 Transmitted 0 Acknowledged 0 Enqueued 0 Lost 0
 queuedOutstanding 0 queuedUnacknowledged 0
1 Collectors in the session:
 Name: collector1, IPAddr: 10.0.0.1, Port: N/A, Priority: 0[DISCONNECTED]
Templates in the session:
 Template ID: 13, Name:
http://www.cablelabs.com/namespaces/DOCSIS/3.0/xsd/ipdr/DOCSIS-CMIS-DS-UTIL-STATS-TYPE/DOCSIS-CMIS-DS-UTIL-STATS-TYPE_3.5.1-A.3.xsd,
 Type:
http://www.cablelabs.com/namespaces/DOCSIS/3.0/xsd/ipdr/DOCSIS-CMIS-DS-UTIL-STATS-TYPE/DOCSIS-CMIS-DS-UTIL-STATS-TYPE_3.5.1-A.3.xsd,
 KeyNumber: 11
Session 2 has a total of 1 templates.
Session ID: 3, Name: cm-status, Descr: cm-status, Started: True
Session Type: Ad-hoc.
Exporting not started.
2019-05-29T05:08:14 Statistics:
 Transmitted 0 Acknowledged 0 Enqueued 0 Lost 0
 queuedOutstanding 0 queuedUnacknowledged 0
1 Collectors in the session:
 Name: collector1, IPAddr: 10.0.0.1, Port: N/A, Priority: 1[DISCONNECTED]
Templates in the session:
 Template ID: 8, Name:
http://www.cablelabs.com/namespaces/DOCSIS/3.0/xsd/ipdr/DOCSIS-CMIS-CM-REG-STATUS-TYPE/DOCSIS-CMIS-CM-REG-STATUS-TYPE_3.5.1-A.1.xsd,
 Type:
http://www.cablelabs.com/namespaces/DOCSIS/3.0/xsd/ipdr/DOCSIS-CMIS-CM-REG-STATUS-TYPE/DOCSIS-CMIS-CM-REG-STATUS-TYPE_3.5.1-A.1.xsd,
 KeyNumber: 18
Session 3 has a total of 1 templates.
```

```
curl -k -H 'Content-Type: application/json' -X GET
https://192.168.0.1.nip.io/api/ipdr/v1/sessions/1
Session ID: 1, Name: samis, Descr: samis, Started: True
Session Type: Time Interval (15 minutes).
Expires in 81 seconds.
Exporting not started.
2019-05-29T05:08:14 Statistics:
 Transmitted 0 Acknowledged 0 Enqueued 0 Lost 0
 queuedOutstanding 0 queuedUnacknowledged 0
1 Collectors in the session:
 Name: collector1, IPAddr: 10.0.0.1, Port: N/A, Priority: 1[DISCONNECTED]
```

```
Templates in the session:
 Template ID: 2, Name:
http://www.cablelabs.com/namespaces/DOCSIS/3.0/xsd/ipdr/DOCSIS-SAMIS-TYPE-1/DOCSIS-SAMIS-TYPE-1_3.5.1-A.1.xsd,
 Type: SAMIS-TYPE-1, KeyNumber: 28
Session 1 has a total of 1 templates.
```

Monitor Collector Status

```
/v1/collectors
```

Note `/ipdr/collectors/status` is deprecated but usable.

Example:

```
curl -k -H 'Content-Type: application/json' -X GET
https://192.168.0.1.nip.io/api/ipdr/v1/collectors
Collector name collector1, ip addr 10.0.0.1, port 0
```

Monitor Exporter Status

```
/v1/exporter
```

Note `/ipdr/exporter/status` is deprecated but usable.

Example:

```
curl -k -H 'Content-Type: application/json' -X GET
https://192.168.0.1.nip.io/api/ipdr/v1/exporter
IPDR exporter is started.
Current parameters:
    KeepAliveInterval:  300
      AckTimeInterval:   60
  AckSequenceInterval:  200
```

Simple Network Management Protocol

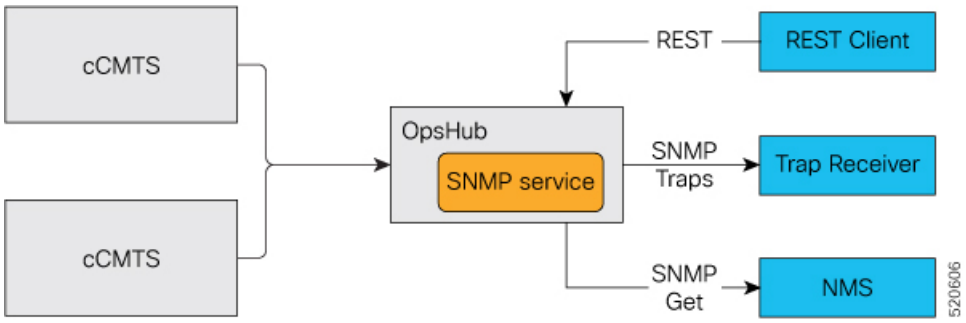
The Simple Network Management Protocol (SNMP) allows you to monitor the DOCSIS elements of Cisco cnBR.

The REST API is the recommended method to configure and operate Cisco cnBR. However, partial SNMP functionality is provided for compatibility with legacy SNMP applications. The Cisco cnBR SNMP Agent is located on the Cisco Operations Hub, and not on individual Cisco cnBRs.

SNMP aggregates information from multiple Cisco cnBR cores that are managed by Cisco Operations Hub.

From an application perspective, you must consider the Cisco Operations Hub as a *large* Cisco cnBR.

The following image provides you an overview of how the SNMP works in the Cisco cnBR.



Configure SNMP

Follow these steps to configure SNMP for Cisco cnBR:

Use the REST API to configure the SNMPv2 community string or Trap Receivers.

```
curl -X {GET|PUT|DELETE} https://{hostname}/api/snmp/v1/config
```

Use one of the following options:

- **SNMPv2 Community**

  To configure SNMPv2 Community, replace `<opshub-ip>` with the Cisco Operations Hub IP. The following example is only indicative. See the *Cisco Cloud Native Broadband Router Operations Hub REST API Guide* for the authentication and encryption format.

```
curl -X GET https://{hostname}/api/snmp/v1/config
{"community-list":[],"v3user-list":[],"trap-receivers":[],"trap-enabled-list":[]}

curl -X PUT -d @- << EOF https://{hostname}/api/snmp/v1/config
{
  "community-list": [
    {
      "community": "public",
      "access": "ro",
      "source": "",
      "oid": ""
    }
  ]
}
EOF

curl -X GET https://{hostname}/api/snmp/v1/config
{"community-list":[{"community":"public","access":"ro","source":"","oid":""}],
"v3user-list":[],"trap-receivers":[],"trap-enabled-list":[]}

curl -X DELETE -d @- << EOF https://{hostname}/api/snmp/v1/config
{
  "community-list": [
    {
      "community": "public",
      "access": "ro",
      "source": "",
      "oid": ""
    }
  ]
```

```
}
EOF
```

- **SNMPv1/v2 Trap**

  The Trap Receiver is a server listening to a specific UDP port for SNMP Trap events. Use the following REST API to configure the Trap Receiver IP address, port, and other information in the Cisco cnBR SNMP agent. The REST API enables the agent to send traps to the trap receiver.

```
curl -X PUT -d @- << EOF https://{hostname}/api/snmp/v1/config
{
  "trap-receivers": [
    {
      "host": "10.1.1.2",
      "port": 12348,
      "version": 2,
      "community": "private"
    }
  ]
}
EOF

curl -X GET https://{hostname}/api/snmp/v1/config
{"community-list":[],"v3user-list":[],"trap-receivers":
[{"host":"1.1.2.2","port":12345,"version":1,"community":"public"},
{"host":"10.1.1.2","port":12348,"version":2,"community":"private"}]}

curl -X DELETE -d @- << EOF https://{hostname}/api/snmp/v1/config
{
  "trap-receivers": [
    {
      "host": "1.1.2.2"
    }
  ]
}
EOF
```

Note

- `host`: Trap Receiver's IP address. For DELETE action, *host* is the key, and the other fields are not necessary.
- `port`: Trap Receiver listens on this port. The Trap Receiver uses the default port **162**, if the *port* is not specified.
- `version`: 1 for SNMPv1, 2 for SNMPv2.
- `community`: Specify the community string to send or receive trap. At the receiver side, there is a configuration file to specify the *community*.

SNMP Support Scope

MIB

The following tables are supported.

```
docsIf31CmtsDsOfdmChanTable
docsIf31DocsisBaseCapability
```

```
docsIf3CmtsCmRegStatusTable
docsIf3CmtsCmUsStatusTable
docsIf3DsChSetTable
docsIf3MdChCfgTable
docsIf3MdDsSgStatusTable
docsIf3MdNodeStatusTable
docsIf3MdUsSgStatusTable
docsIf3UsChSetTable
docsIfCmtsChannelUtilizationInterval
docsIfCmtsChannelUtilizationTable
docsIfCmtsCmStatusTable
docsIfCmtsDownChannelCounterTable
docsIfCmtsModulationTable
docsIfCmtsUpChannelCounterTable
docsIfDocsisBaseCapability
docsIfDownstreamChannelTable
docsIfUpstreamChannelTable
docsPnmBulkDestIpAddr
docsPnmBulkDestIpAddrType
docsPnmCmtsUtscCfgTable
docsPnmCmtsUtscCtrlTable
docsQos3CmtsMacToSrvFlowTable
docsQos3ServiceFlowStatsTable
docsQos3ServiceFlowTable
docsRphyCmtsCmRegStatusTable
docsRphyRpdDevIdentificationTable
docsRphyRpdDevNdfCfgTable
docsRphyRpdDevNdrCfgTable
docsRphyRpdIfCoreToRpdMapTable
docsRphyRpdIfRpdToCoreMapTable
docsRphyStatsRpdUsOfdmaChanPerfStatsTable
docsRphyStatsRpdUsScQamChanPerfStatsTable
ifTable
```

Note

- Only a subset of OIDs required for the third-party tools integration is supported.
- Only the following MIBs supports SNMP Write:

  ```
  docsPnmBulkDestIpAddr
  docsPnmBulkDestIpAddrType
  docsPnmCmtsUtscCfgTable
  docsPnmCmtsUtscCtrlTable
  ```

- Cisco cnBR does not support NDF/NDR. The following MIBs only conform to prerequisites of third-party tools to capture upstream spectrum:

  ```
  docsRphyRpdDevNdfCfgTable
  docsRphyRpdDevNdrCfgTable
  ```

- For the following MIB, the table returns value zero (0) for all rows until OFDMA is supported by Cisco cnBR.

  ```
  docsRphyStatsRpdUsOfdmaChanPerfStatsTable
  ```

Trap

Only CM online and offline events are supported.

Reference

DOCSIS MIBs

SNMP Limitations

Cisco cnBR SNMP has the following limitations:

- SNMP write is supported only for the MIB object or table that is listed in the MIB section. For more information, see SNMP Support Scope, on page 229.
- Only a limited set of DOCSIS MIB OIDs and traps is supported. For more information, see SNMP Support Scope, on page 229.