

Cloud Native BNG Control Plane Configuration Guide, Release 2026.04.0

First Published: 2026-11-16

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

THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS IN THIS MANUAL ARE SUBJECT TO CHANGE WITHOUT NOTICE. ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS MANUAL ARE BELIEVED TO BE ACCURATE BUT ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. USERS MUST TAKE FULL RESPONSIBILITY FOR THEIR APPLICATION OF ANY PRODUCTS.

THE SOFTWARE LICENSE AND LIMITED WARRANTY FOR THE ACCOMPANYING PRODUCT ARE SET FORTH IN THE INFORMATION PACKET THAT SHIPPED WITH THE PRODUCT AND ARE INCORPORATED HEREIN BY THIS REFERENCE. IF YOU ARE UNABLE TO LOCATE THE SOFTWARE LICENSE OR LIMITED WARRANTY, CONTACT YOUR CISCO REPRESENTATIVE FOR A COPY.

The Cisco implementation of TCP header compression is an adaptation of a program developed by the University of California, Berkeley (UCB) as part of UCB's public domain version of the UNIX operating system. All rights reserved. Copyright © 1981, Regents of the University of California.

NOTWITHSTANDING ANY OTHER WARRANTY HEREIN, ALL DOCUMENT FILES AND SOFTWARE OF THESE SUPPLIERS ARE PROVIDED “AS IS" WITH ALL FAULTS. CISCO AND THE ABOVE-NAMED SUPPLIERS DISCLAIM ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, THOSE OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE.

IN NO EVENT SHALL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS MANUAL, EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.

All printed copies and duplicate soft copies of this document are considered uncontrolled. See the current online version for the latest version.

Cisco has more than 200 offices worldwide. Addresses and phone numbers are listed on the Cisco website at www.cisco.com/go/offices.

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: https://www.cisco.com/c/en/us/about/legal/trademarks.html. Third-party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1721R)

© 2026 Cisco Systems, Inc. All rights reserved.



CONTENTS

PREFACE

About this Guide xxiii

Conventions Used xxiii

CHAPTER 1

cnBNG Overview 1

Overview 1

Evolution of cnBNG 2

cnBNG Architecture 2

cnBNG Components 4

Subscriber Microservices Infrastructure 4

cnBNG Control Plane 5

cnBNG User Plane 6

License Information 6

Standard Compliance 7

Limitations and Restrictions 7

CHAPTER 2

Deployment Options For Integrating cnBNG with Data Center Fabrics 9

Flexible deployment models 9

cnBNG on VXLAN BGP EVPN Fabric 12

cnBNG on Cisco ACI Fabric 12

cnBNG on Traditional vPC and LACP Networks 13

CHAPTER 3

cnBNG Installation and Configuration 15

Feature Summary and Revision History 15

Summary Data 15

Revision History 16

Feature Description 16

BNG Ops Center 16
Installing cnBNG and Accessing BNG Ops Center 17
Prerequisites 17
Installing cnBNG in an Offline Environment 18
Accessing BNG Ops Center 22
CP and UP Service Configuration 24
Configuring the CP 24
Configuring the UP 30
Loading Day1 Configuration 30
Mapping Pods with Node Labels 30
High Availability Support on BareMetal Server 31
cnBNG CNF Deployment on AIO BareMetal Server 31
BareMetal CNDP AIO Bring-Up Procedure 33
Implementing cnBNG CP Validation with CNDP 33
Prerequisites 34
Instantiating and Provisioning Inception Server Instance 34
Installing Cluster Manager Node 34
Installing the All-in-One Cluster 35
Integrating RADIUS and UP with the AIO BareMetal Server 39
cnBNG Cluster Deployment Using Inception Server 39
Installing Inception Server on Baremetal 40
Clear a Boot Drive 40
Create Virtual Drive from Unused Physical Drives 40
Install Base ISO image 42
Configure User and Network Parameters 43
Install Inception Server 43
Deploy SMI Cluster 44
Add Images to Inception Server 44
Generate SSH Keys 45
Add SMI Cluster Deployer Configuration 45
cnBNG Cluster Deployment Support on Red Hat OpenShift 46
Maintenance Operation Procedure (MOP) for cnBNG Node Scaling 48
cnBNG Health-Check pods 60
How cnBNG Health-Check pod works 61

Health-Check validators 63
Restrictions for the cnBNG Health-Check pod 63
Configure the cnBNG Health-Check pod 64

CHAPTER 4 Pods and Services Reference 67
Feature Summary and Revision History 67
Summary Data 67
Revision History 67
Feature Description 68
Pods 70
Services 73
Open Ports and Services 74
Associating Pods to the Nodes 74
Viewing the Pod Details and Status 75
States 75
Configuring Pod Layout 76
Configuring Event Trace 77

CHAPTER 5 NSO Subscriber Microservices Infrastructure Core Function Pack 79
Feature Summary and Revision History 79
Summary Data 79
Revision History 80
Feature Description 80
Benefits of NSO SMI CFP 80
Supported Scenarios 80
Prerequisites for NSO SMI CFP 81
Initial Configuration 81
Add the SMI Cluster Manager as a device to NSO 81
Deployment Services 83
Deploy the Kubernetes Cluster 84
Functions Services 92
Deploy a CNF 92
Upgrade a CNF 95
Applications Services 96

Apply Day-1 Configuration to BNG Ops-center 96
Disable the auto-sync Feature 97
Trigger the Sync Action 99
Delete the SMI Deployment 100

CHAPTER 6 Smart Licensing 101
Feature Summary and Revision History 101
Summary Data 101
Revision History 101
Feature Description 102
Cisco Software Central 102
Smart Accounts and Virtual Accounts 102
Requesting a Cisco Smart Account 102
cnBNG Control Plane Smart Licensing 103
Essential License 104
Advantage License 105
Configuring Smart Software Licensing for cnBNG CP 106
Users with Access to CSC 106
Users without Access to CSC 110
Monitoring and Troubleshooting Smart Licensing 116

CHAPTER 7 Alarm Support 117
Feature Summary and Revision History 117
Summary Data 117
Revision History 117
Feature Description 117
Supported Alarm Categories 118
Alert Configuration Recommendations 119
Application-based Alerts 119
Use-Case Based Alerts 121
Alert Routing to SNMP Trapper 125
Alert Routing to Alert Logger 125
Alarm Severity Levels 125
Configuring Alarm Support 125

Configuring Alert Rules 126
Configuring SNMP Traps 127

CHAPTER 8 Authentication, Authorization, and Accounting Functions 129
Feature Summary and Revision History 129
Summary Data 129
Revision History 130
Feature Description 132
AAA Overview 132
Using RADIUS Server Group 133
Specifying Method Order 134
Defining AAA Attributes 134
Creating Attributes of Specific Format 135
Making RADIUS Server Settings 137
Message-Authenticator validation for RADIUS servers 137
Configure Message-Authenticator validation for RADIUS servers 139
Balancing Transaction Load on the RADIUS Server 141
RADIUS Change of Authorization Overview 142
Multi-Action Change of Authorization 144
Generating Accounting Records 145
Enhanced CoA with Conditional Retry Logic 146
User Authentication and Authorization in the Local Network 146
Service Accounting 147
Utilizing Session accounting AAA profiles for Service Accounting and Accounting Send-Stop Setup-Failure 149
Configure Session Accounting AAA Profiles for Service Accounting 149
Configure Accounting Send-Stop Setup-Failure 151
RADIUS Accounting Message Handling 152
Configure Asynchronous RADIUS Accounting 152
Standard Compliance 153
RADIUS Automated Testing 153
Configure RADIUS Automated Testing 154
Clear RADIUS operational statistics 156
Clear RADIUS server statistics 156

Clear dynamic authorization client statistics 157
RADIUS Attributes Filtering 159
Restrictions and Guidelines for RADIUS Attributes Filtering 159
Configure RADIUS Attributes Filtering 160
Framed Route with tag and preference support 163
Restrictions on Framed Route with tag and preference support 165
Configure framed route preference support 165
Configuring AAA Functions 167
Configuring AAA Attributes 168
Configuring the CoA-NAS Interface 170
Configuring Method Order for AAA 171
Configuring RADIUS Accounting Options 173
Configuring RADIUS Accounting Server Group 175
Configuring RADIUS Attributes 176
Configuring RADIUS Attribute Format 177
Configuring RADIUS Dead Time 178
Configuring RADIUS Detect Dead Server 179
Configuring RADIUS NAS-IP 179
Configuring RADIUS Pod 180
Configuring RADIUS Retries 180
Configuring RADIUS Server 181
Configuring RADIUS Server Group 182
Configuring RADIUS Server Selection Logic 182
Configuring RADIUS Timeout 183
Configuring Radius PID-Quarantine 183

CHAPTER 9 **Cisco Common Data Layer** 185
Feature Summary and Revision History 185
Summary Data 185
Revision History 185
Feature Description 186
Limitations 186

CHAPTER 10 **Control Plane and User Plane Association** 187

Feature Summary and Revision History 187
Summary Data 187
Revision History 187
Feature Description 188
Manually remove CP-UP association 188
Enabling Control Plane and User Plane Association 189
Associating the User Plane 189
Multiple Control Plane associations 189
Restrictions for multiple Control Plane associations 190
Configure and verify multiple CP associations for a UP 190
Configure multiple CP associations for a UP 190
Verify multiple CP associations to a UP 193
Adding a new CP-UP association without disrupting existing connections 212
Behavior during CP–UP connectivity failure 212
UP Session Disconnect Notification 213
Feature Description 213
UP Rollback Failure Handling Notification 213
Feature Description 213

CHAPTER 11 **DHCP and IPoE Subscriber Management** 215
Feature Summary and Revision History 215
Summary Data 215
Revision History 216
Feature Description 216
DHCP and IPoE Functionalities 217
How it Works 222
Call Flows 222
Standard Compliance 224
Limitations and Restrictions 224
Configuring the DHCP and IPoE Subscriber Management Feature 225
Configuring the IPv4 DHCP Server Profile 225
Configuring the IPv4 DHCP Class 227
Configuring the IPv6 DHCP Server Profile 228
Configuring the IPv6 DHCP Class 229

DHCPv6 Raw Option Support 231
Configure DHCPv6 Raw Option Support 231
IPv6 Class Configuration and Static IP Allocation Support 234
Configure the IPv6 DHCP Class 236
Configure Static IP Allocation 238
Leased IP Hold Time 240
Feature Summary and Revision History 240
Summary Data 240
Revision History 240
Feature Description 240
How it Works 240
Limitations and Restrictions 243
Configuring the Leased IP Hold Time Feature 244
DHCP IP Lease Reservation 244
Feature Summary 244
Revision History 244
Feature Description 245
How it Works 245
Limitations and Restrictions 245
Configuring DHCP IP Lease Reservation 245
Reserving IP Address using CLI (Action Command/REST API) 245
Enhanced support for RADIUS attributes 246
Stateful-IPv6-Address-Pool support 246
Configure Stateful-IPv6-Address-Pool attribute 247
Delegated-IPv6-Prefix-Pool support 248
Configure Delegated-IPv6-Prefix-Pool attribute 248

CHAPTER 12 **End-to-End Flow Control** 251
Feature Summary and Revision History 251
Summary Data 251
Revision History 251
Feature Description 252
Simplified Control Plane flow control configurations 252
Configure user-plane and subscriber CPS values 254

Migrate CP flow control models 255
Upgrade from manual to automated method on a GR setup 255
Upgrade from manual to automated method on a non-GR setup 255
Downgrade from automated to manual method on GR and non-GR setups 255
FSOL Token Mechanism 256
Configuring Token for FSOL Pod 257
Limitations and Restrictions 257
How it Works (Manual Configuration) 258
Dispatcher 258
Overload Control 259
Configuring End-to-End Flow Control (Manual Configuration) 259
Configuring Dispatcher for GTPu Interface 260
Configuring Dispatcher for N4 Interface 261
Configuring Overload Control for Message Types 263
Associating Flow Control Group in User Plane 264

CHAPTER 13 High Availability and CP Reconciliation 265
Feature Summary and Revision History 265
Summary Data 265
Revision History 265
Feature Description 266
How it Works 267
Automatic Session Mismatch Detection 268
Synchronizing Sessions Across CP Pods and UP 269
Limitations and Restrictions 269
Configuring High Availability and CP Reconciliation 270
Reconciling Sessions Across CP Pods and UP 270
Configuring CDL Bulk Notifications 271
Automatic CP-UP reconciliation for CP events 272
Operational restrictions for automatic CP-UP reconciliation 273
Configure automatic CP-UP reconciliation for CP events 276
Monitor reconciliation with alerts 278

CHAPTER 14 IP Address Management 285

Feature Summary and Revision History 285
Summary Data 285
Revision History 286
Feature Description 286
IPAM Components 287
IPAM Sub-Modules 287
IPAM Integration in cnBNG 288
Configuration Recommendations 288
How it Works 290
Call Flows 290
Limitations 294
Configuring IPAM Feature 294
Configuring IPAM Source 294
Configuring Global Threshold 295
Disable DP threshold monitoring 296
Configuring IPAM Address Pool 296
Configuring IPAM Quarantine Timer 296
Configuring IP Address Reservation 297
Configuring IPv4 Address Ranges 298
Configuring IPv6 Address Ranges 298
Configuring IPv6 Prefix Ranges 299
Configuring IPv4 Threshold 299
Configuring IPv6 Prefix-Range Threshold 300
Configuring IPv4 Address Range Split 300
Configuring IPv6 Address and Prefix Address-Range-Spilt 301
Configuring Variable Chunk Size Support for an IPAM Data Plane 302
IPAM Enhancements 303
IPAM Chunk-Optimization with Node Manager Pair for SRG 305
Dynamic Chunk Allocation for CP-GR 307
Configuration Example 307
RADIUS-based address assignment and route-tagging 308
How RADIUS-based address assignment and route-tagging works 309
Limitations for RADIUS-based address assignment and route-tagging 310
Configure RADIUS-based address assignment and route-tagging 310

Reserved IPv4 address controls for PPPoE pools 314
Limitations for reserved IPv4 address controls for PPPoE pools 314
Configure reserved IPv4 address controls for PPPoE pools 315
Periodic reconciliation 316
Configure periodic reconciliation 317
Periodic reconciliations for ID Manager 318
Best practice for scheduling periodic reconciliation 319
Configure periodic reconciliation for ID Manager 319
IPAM Route Programming Enhancements 320
Pre-Allocation of Gateway IP and Address Chunks 321
Configure Pre-Allocation of Gateway IP and Address Chunks 322
IANA and IAPD Allocation from Same IP Range 324
Restrictions for IANA and IAPD Allocation from Same IP Range 325
Configure IANA and IAPD Allocation from Same IP Range 325
Static IP Support 327
Feature Summary and Revision History 327
Summary Data 327
Revision History 328
Feature Description 328
Limitations 329
Configuring Static IP Support 329
Configuring Static IP Address Ranges 330
Configuring a Static IP-pool to Support Multiple UPFs 330
Support for circuit-id as ipam-dp-key in static IPAM pools 331
Configure associated-srg-peer-id when using circuit-id as ipam-dp-key in a static IPAM pool 332
Static IP Mapping Database 333
Feature Summary and Revision History 333
Summary Data 333
Revision History 333
Feature Description 333
How it Works 334
Configuring the Static IP Mapping Database 335
Configuring Static IP Mapping Database in IPAM 335
Configuring Static Database Key in DHCP Profile 338

IP, ID, and Chunk event logging 339
Configure event logging 341

CHAPTER 15 L2TP Subscriber Management 343
Feature Summary and Revision History 343
Summary Data 343
Revision History 343
Feature Description 344
L2TP Overview 344
L2TP Features 345
L2TP tunnel failover support 346
How L2TP tunnel failover works 347
Configure tunnel busy timeout for LNS servers 348
Seamless L2TP session continuity during server reboot 349
IETF Tagged Attributes on LAC 350
Tunnel-Preference support 351
Configure Tunnel-Preference attribute 352
Tunnel-Client-Auth-Id support 353
Configure Tunnel-Client-Auth-ID attribute 353
How it Works 355
L2TP Handling 355
AAA Attributes for L2TP 355
Handling L2TP Sessions during CP-GR Switchover 356
Call Flows 356
LAC Session Bringup Call Flow 356
LAC Session Bringdown Call Flow 359
LNS Session Bringup Call Flow 360
LNS Session Bringdown Call Flow 363
Standard Compliance 364
Limitations 364
Configuring the L2TP Subscriber Management Feature 365
Creating the L2TP Profile 365
Monitoring and troubleshooting 368
View L2TP tunnel instance details 368

View detailed LNS session information using the sublabel identifier 369
View L2TP tunnel disconnect history 371
Clear L2TP-tunnel disconnect history records 373

CHAPTER 16 Log Generation Support 375
Feature Summary and Revision History 375
Summary Data 375
Revision History 375
Feature Description 375

CHAPTER 17 Monitor Protocol and Subscriber 377
Feature Summary and Revision History 377
Summary Data 377
Revision History 377
Feature Description 378
Configuring Monitor Subscriber and Protocol 378
Configuring Monitor Subscriber 378
Configuring Monitor Protocol 389
Copying Log Files 390
Viewing Log Files 393

CHAPTER 18 Multiple Replica Support for cnBNG Services 395
Feature Summary and Revision History 395
Summary Data 395
Revision History 395
Feature Description 396
How it Works 396
Configuring Multiple Replica Support for cnBNG Services 397
Replicating Multiple cnBNG Service Instances 397

CHAPTER 19 PPPoE Subscriber Management 399
Feature Summary and Revision History 399
Summary Data 399
Revision History 399

Feature Description 400
PPPoE Overview 400
PPPoE Features 401
PPP Overview 403
PPP Features 403
Address Assignment Strategies 404
How it Works 404
PPPoE Handling 404
PPP Handling 405
Call Flows 406
Standard Compliance 408
Limitations 408
Configuring the PPPoE Subscriber Management Feature 408
Creating PPPoE Profile 409
Creating the PPP Feature Template 410
View PPPoE or LNS session event history 411
Stateless Address Autoconfiguration (SLAAC) 414
Configure SLAAC for PPPOE 415

CHAPTER 20 **Rolling Software Update** 419
Feature Summary and Revision History 419
Summary Data 419
Revision History 419
Feature Description 420
How it Works 421
Rolling Software Update Using SMI Cluster Manager 421
Installing the Rolling Software Update 422
Prerequisites 422
Performing Rolling Software Update 425
Monitoring the Rolling Software Update 426
Viewing the Pod Details 427

CHAPTER 21 **Subscriber Manager** 437
Feature Summary and Revision History 437

Summary Data 437
Revision History 438
Feature Description 438
How it Works 440
Configuring Subscriber Manager Features 440
Configuring the HTTPR Policy Name 440
Configuring IPv4 Options 441
Configuring IPv6 Options 441
Configuring QoS Parameters 442
Configuring the VRF Name 442
Configuring a Subscriber Profile 443
Automatic Session Reconciliation 445
Feature Description 445
How it Works 445
Framed Route Support 445
Feature Description 445
Session disconnect history 446
Restrictions for session disconnect history 447
Configure the disconnect history record limit per User Plane 447
Clear session disconnect history records 448
Monitor disconnect history metrics 448
Verify session disconnect history 449
Subscriber Accounting Functions 461
Feature Description 461
Limitations and Restrictions 462
Configuring Subscriber Accounting Functions 462
Configuring Service Accounting 462
Configuring Session Accounting 463
Subscriber QoS Policy 463
Feature Description 463
RADIUS-Based Policing - QoS Shape-Rate parameterization 464
Limitations of configuring RADIUS-Based Policy 465
Configure QoS shape-rate parameterization 465
Shared Policy Instance 468

Limitations of configuring Shared Policy Instance 469
Configure a policy with SPI using feature template 469
Configure a Policy with SPI using RADIUS 471

CHAPTER 22 **Routed packet trigger subscriber sessions 475**

Feature Summary 475
Revision History 475
cnBNG routed packet trigger subscriber sessions 476
Restrictions for routed packet trigger subscriber sessions 477
Configure routed packet trigger subscriber sessions 477
Monitor subscriber sessions 481

CHAPTER 23 **CP Geographical Redundancy 483**

Feature Summary 483
Revision History 484
Feature Description 484
Prerequisites for CP-GR Cluster Bring Up 484
CP-GR Network Slicing Requirements 486
Architecture 490
Active-Active GR Deployment 491
MED Value 493
Geo Redundancy Support for AIO Control Plane Cluster 494
Geographical redundancy for L2TP sessions 496
Limitations of configuring geographical redundancy for L2TP sessions 496
Configure CP-GR support for L2TP tunnels and sessions 497
Synchronize L2TP tunnels between CP and UP for CP-GR 499
GR-Replication Pod 499
ETCD and Cache Pod Replication 500
Pod Monitoring 501
Traffic Monitoring 501
Configure Traffic Monitoring 502
Instance Roles 503
Automated standby-state recovery 504
IPAM 505

Limitations and Restrictions 505
Configuring CP Geo-Redundancy 506
Configuring NF Instance 506
Configuring Endpoints 507
Configuring Geo Replication 508
Configuring IPAM 509
Configuring RADIUS 509
Configuring a Subscriber Profile 510
Configuring the IPv4 DHCP Server Profile 510
Configuring the IPv6 DHCP Server Profile 511
Creating PPPoE Profile 512
Creating the PPP Feature Template 512
Configuring Dynamic Routing using BGP 513
AS-Path Prepending for BGP VIP Routes 516
Configuring BGP Speaker 516
Configuring BFD 517
Configuring POD Monitoring 518
Configuring CDL Instance Awareness and Replication 519
Configuring CDL Instance Awareness 520
Configuring CDL Replication 520
CP-GR for AIO - Configuration Example 523
Cluster Maintenance Mode 526
Manual CLI Switchover 526
Geo Switch Role 527
Geo Reset Role 527
Key Performance Indicators (KPIs) 527
Monitoring and Troubleshooting 530
show bgp kernel route 530
show bgp global 531
show bgp neighbors 531
show bgp route summary 532
show bgp routes 533
show bfd neighbor 533
show bgp-learned-routes 534

show bgp-advertised-routes 534

CHAPTER 24 **UP Geo Redundancy** 537

Feature Summary 537

Revision History 538

Feature Description 538

UP Geo Redundancy Architecture 538

Subscriber Redundancy Group 539

SRG warm-standby mode 540

Configure SRG warm-standby mode 541

SRG cold standby mode 543

Configure SRG cold standby mode 544

Session Distribution Across SRG 547

Benefits of UP Geo Redundancy 550

Supported Features in UP Geo Redundancy 550

UP Geo Redundancy Configuration Guidelines 551

Configuring UP Geo Redundancy 551

Configuration Example 552

Configuration Verification 553

Configuring IPAM 554

Range-based interface list 556

L3 Routed Subscriber Sessions with Subscriber Redundancy Group 557

Restrictions for L3 Routed Subscriber Sessions with SRG 563

Configure L3 Routed Subscriber Sessions with SRG 563

Session Synchronization between UPs 569

Route Synchronization between CP and UP 572

Order of Reconciliation 572

Monitoring Support 572

clear subscriber sessmgr 572

clear subscriber pppoe 573

show subscriber redundancy 573

show subscriber redundancy-sync 574

show subscriber dhcp 574

show subscriber pppoe 575

show subscriber session 576
show subscriber synchronize 577
show ipam dp 578

CHAPTER 25 CnBNG Control Plane Quality of Service 581
Feature Summary and Revision History 582
Summary Data 582
Revision History 582
DSCP marking for replication traffic 582
How DSCP marking for replication traffic works 584
Configure DSCP marking for replication traffic 584
DSCP marking for CP-UP control packets 585
How DSCP marking for CP-UP control packets works 586
Configure DSCP marking for CP-UP control packets 587
DSCP marking for RADIUS traffic 588
How RADIUS DSCP marking works 588
Configure RADIUS DSCP marking 589
CP-UP liveliness optimization 589
How CP-UP liveliness optimization works 590
Configure the CP-UP heartbeat timer 591

CHAPTER 26 Lawful Intercept for cnBNG 593
Lawful Intercept for cnBNG 593
How cnBNG LI works 594
Configure LI for cnBNG subscriber session 596
Redundancy Handling 596
Call flow for LI provisioning 597
Packet format 598

APPENDIX A RADIUS Attributes 601
RADIUS IETF Attributes 601
RADIUS Vendor-Specific Attributes 603
Vendor-Specific Attributes for Account Operations 607
RADIUS ADSL Attributes 607

RADIUS ASCEND Attributes 607

RADIUS Disconnect-Cause Attributes 607



About this Guide

Note The documentation set for this product strives to use bias-free language. For purposes of this documentation set, bias-free is defined as language that does not imply discrimination based on age, disability, gender, racial identity, ethnic identity, sexual orientation, socioeconomic status, and intersectionality. While any existing biased terms are being substituted, exceptions may be present in the documentation due to language that is hardcoded in the user interfaces of the product software, language used based on RFP documentation, or language that is used by a referenced third-party product.

This preface describes the Cloud Native Broadband Network Gateway (cnBNG) Control Plane (CP) Configutration Guide, how it is organized, and its document conventions.

This guide describes the Cloud Native BNG solution and includes feature descriptions, specification compliance, session flows, configuration instructions, CLI commands and so on.

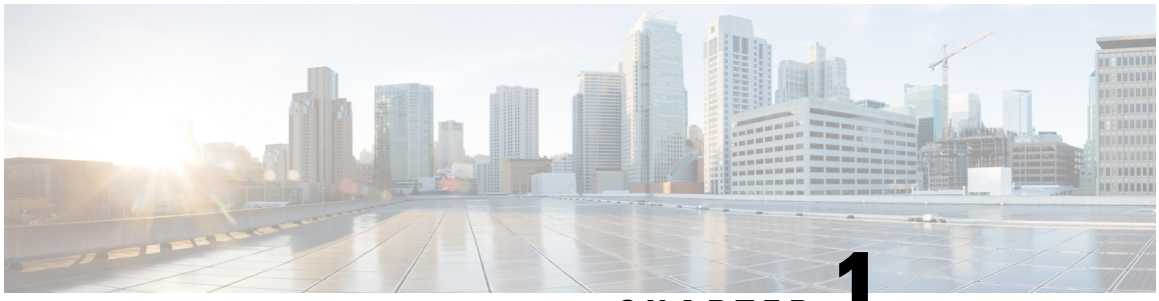
• Conventions Used, on page xxiii

Conventions Used

The following tables describe the conventions used throughout this documentation.

Notice Type	Description
Information Note	Provides information about important features or instructions.
Caution	Alerts you of potential damage to a program, device, or system.
Warning	Alerts you of potential personal injury or fatality. May also alert you of potential electrical hazards.

Typeface Conventions	Description
Text represented as a `screen display`	This typeface represents displays that appear on your terminal screen, for example: `Login:`
Text represented as **commands**	This typeface represents commands that you enter, for example: **show ip access-list** This document always gives the full form of a command in lowercase letters. Commands are not case sensitive.
Text represented as a **command** *variable*	This typeface represents a variable that is part of a command, for example: **show card** *slot_number* *slot_number* is a variable representing the desired chassis slot number.
Text represented as menu or sub-menu names	This typeface represents menus and sub-menus that you access within a software application, for example: Click the **File** menu, then click **New**



CHAPTER 1

cnBNG Overview

- Overview, on page 1
- License Information, on page 6
- Standard Compliance, on page 7
- Limitations and Restrictions, on page 7

Overview

The Broadband Network Gateway (BNG) is the access point for subscribers, through which they connect to the broadband network. When a connection is established between BNG and Customer Premise Equipment (CPE), the subscriber can access the broadband services provided by the Network Service Provider (NSP) or Internet Service Provider (ISP).

BNG establishes and manages subscriber sessions. When a session is active, BNG aggregates traffic from various subscriber sessions from an access network, and routes it to the network of the service provider.

BNG is deployed by the service provider and is present at the first aggregation point in the network, such as the edge router. An edge router, like the Cisco ASR 9000 Series Router, needs to be configured to act as the BNG. Because the subscriber directly connects to the edge router, BNG effectively manages subscriber access, and subscriber management functions such as:

- Authentication, Authorization, and Accounting (AAA) of subscriber sessions
- Address assignment
- Security
- Policy management
- Quality of Service (QoS)

Implementing the BNG provides the following benefits:

- Communicates with authentication, authorization, and accounting (AAA) server to perform session management and billing functions besides the routing function. This feature makes the BNG solution more comprehensive.
- Provides different network services to the subscriber. This enables the service provider to customize the broadband package for each customer based on their needs.

Cisco provides two BNG solutions:

- **Physical BNG** where the BNG Control Plane (CP) and the User Plane (UP) are tightly coupled inside a Cisco IOS XR platform where the CP runs on an x86 CPU and the UP runs on a physical NPU or ASIC.

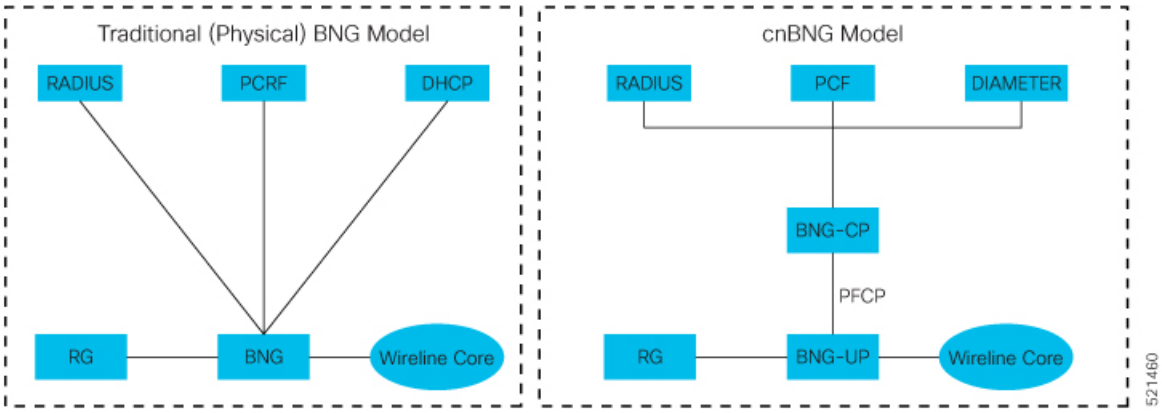
  For more information about the physical BNG, refer to the latest version of the *Broadband Network Gateway Configuration Guide* for Cisco ASR 9000 Series Routers.

- **Virtual BNG (vBNG)** where the BNG CP and UP run in separate VM-based Cisco IOS XR software on general purpose x86 UCS servers.

Evolution of cnBNG

The Cisco Cloud Native Broadband Network Gateway (cnBNG) provides a new dimension to the Control Plane and User Plane Separation (CUPS) architecture of the Broadband Network Gateway (BNG), enabling flexibility and rapid scaling for Internet Service Providers (ISPs).

Figure 1: Evolution of BNG to cnBNG



The architectural change is an evolution from an integrated traditional BNG running on a single router to a disaggregated solution, where the centralized subscriber management runs on an elastic and scalable Cloud Native Control Plane (CP) and the User Plane (UP) delivers the forwarding functionality.

cnBNG Architecture

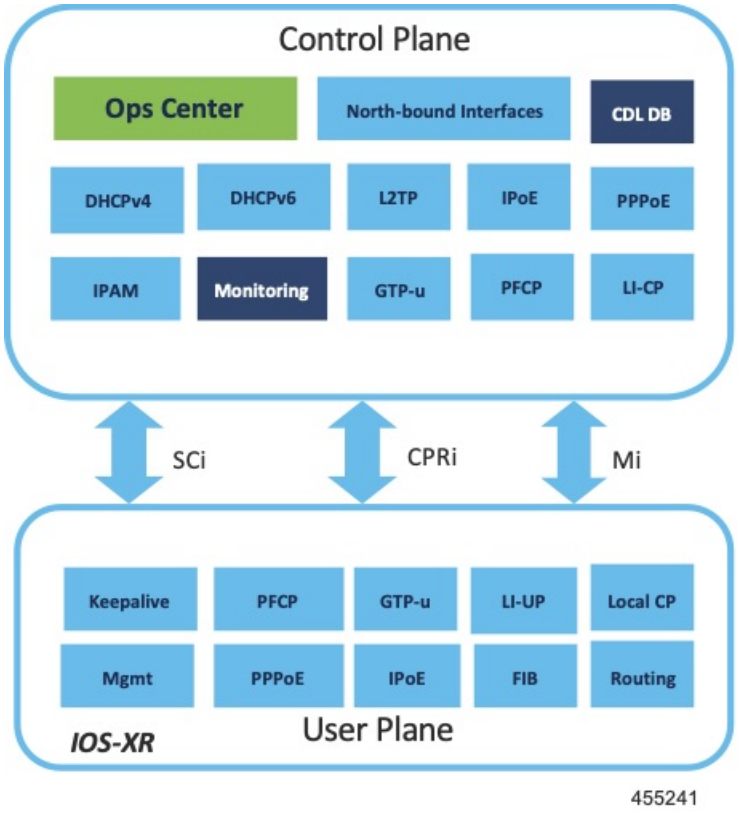
In the cnBNG architecture, the CPs and UPs are clearly and cleanly separated from each other and run in completely distinct and independent environments.

The BNG CP is moved out to a container-based microservice cloud environment.

The UP can be on any of the physical platforms that supports the BNG UP, like Cisco ASR 9000 Series Routers.

The following figure illustrates the overall cnBNG architecture.

Figure 2: cnBNG Architecture



Features and Benefits

The cnBNG supports the following features:

- **Path to convergence**: With shared Subscriber Management infrastructure, common microservices across the policy layer and shared UPs for BNG and Mobile back-haul, cnBNG paves the way for real Fixed Mobile Convergence (FMC).
- **Flexibility of scaling**: cnBNG architecture provides flexibility by decoupling the required scalability dimensions. The CP can be scaled with requirement of number of subscribers to be managed and UPs can be augmented based on the bandwidth requirements. Instead of building the CP for peak usage, the orchestrator can be triggered to deploy the relevant microservices as needed to handle the increased rate of transactions.
- **Distributed UPs**: With reduced operational complexity and minimal integration efforts with centralize CP, UPs can be distributed, closer to end-users to offload traffic to nearest peering points and CDNs. This feature reduces the core transport costs.
- **Cost effective and Leaner User planes**: With the subscriber management functions moved to cloud, you can choose cost-effective UP models for optimized deployment requirements.

The benefits of the cnBNG architecture are:

- Simplified and unified BNG CP
- Platform independent and Network Operation System (NOS) agnostic BNG CP

- Unified Policy interface across both BNG and mobility
- Common infrastructure across wireline and mobility
- Seamless migration from existing deployments
- Leverage the common infrastructure across access technologies
- Standardized model driven interface with the UP
- Data externalization for North-bound interfaces (NBI)
- Highly available and fault tolerant
- Simplified Subscriber Geo redundancy
- Horizontally scalable CP
- Independent CP and UP upgrades
- Feature agility with CI and CD
- Manageability and Operational Simplification

cnBNG Components

The cnBNG solution comprises of the following components:

Subscriber Microservices Infrastructure

The Cisco Ultra Cloud Core Subscriber Microservices Infrastructure (SMI) is a layered stack of cloud technologies that enable the rapid deployment, and seamless life-cycle operations for microservices-based applications.

The SMI stack consists of the following:

- SMI Cluster Manager—Creates the Kubernetes (K8s) cluster, creates the software repository, and provides ongoing LCM for the cluster including deployment, upgrades, and expansion.
- Kubernetes Management—Includes the K8s master and etcd functions, which provide LCM for the NF applications deployed in the cluster. This component also provides cluster health monitoring and resources scheduling.
- Common Execution Environment (CEE)—Provides common utilities and OAM functionalities for Cisco cloud native NFs and applications, including licensing and entitlement functions, configuration management, telemetry and alarm visualization, logging management, and troubleshooting utilities. Additionally, it provides consistent interaction and experience for all customer touch points and integration points in relation to these tools and deployed applications.
- Common Data Layer (CDL)—Provides a high performance, low latency, stateful data store, designed specifically for 5G and subscriber applications. This next generation data store offers HA in local or geo-redundant deployments.
- Service Mesh—Provides sophisticated message routing between application containers, enabling managed interconnectivity, additional security, and the ability to deploy new code and new configurations in low risk manner.

- NB Streaming—Provides Northbound Data Streaming service for billing and charging systems.
- NF/Application Worker nodes—The containers that comprise an NF application pod.
- NF/Application Endpoints (EPs)—The NF's/application's interfaces to other entities on the network.
- Application Programming Interfaces (APIs)—SMI provides various APIs for deployment, configuration, and management automation.

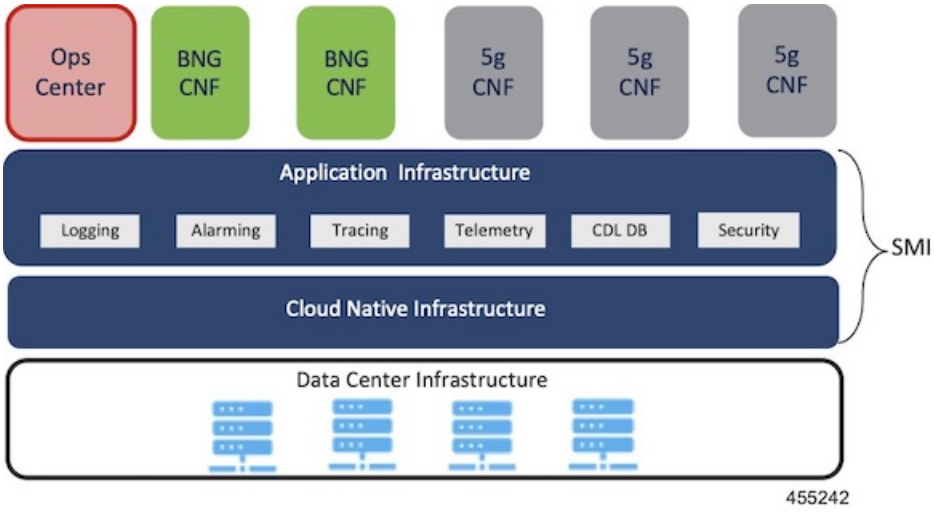
For more information on SMI components, refer to the "Overview" chapter of the *Ultra Cloud Core Subscriber Microservices Infrastructure* documentation—*Deployment Guide*.

For information on the Cisco Ultra Cloud Core, see https://www.cisco.com/c/en/us/products/collateral/wireless/packet-core/datasheet-c78-744630.html.

cnBNG Control Plane

The Cisco cnBNG CP is built on Cisco® Cloud Native Infrastructure, which is a Kubernetes-based platform that provides a common execution environment for container-based applications. This CP is built on principles of stateless microservices, to scale at-ease, introduce services much faster and more cost-effective.

Figure 3: cnBNG Control Plane Architecture



The CP runs as a Virtual Machine (VM) to adapt to existing service provider-deployed virtual infrastructure. It is built ground-up on a clean-slate architecture with a view on ‘Converged Subscriber Services’ and is aligned to 3gpp and BBF standards.

The cnBNG CP effectively manages the subscriber management functions such as:

- Authentication, authorization, and accounting of subscriber sessions
- IP Address assignment
- In-built DHCP Server
- Security
- Policy management
- Quality of Service (QoS)

Service providers can choose from wide choice of available ASR 9000 form factors, based on exact deployment requirements. The CUPS architecture allows to run these UPs in a distributed mode, to the edge of network, for early traffic offloads.

For more information about the cnBNG control plane, refer to the *Cloud Native Broadband Network Gateway Control Plane Configuration Guide.*

cnBNG User Plane

The UP delivers the forwarding functionality of the entire cnBNG solution. With the CP handling the subscriber management functionality, the cnBNG architecture enables the UP to be more distributed and interoperable with cnBNG CP with minimal integration efforts. The cnBNG Subscriber Provisioning Agent (SPA), which is the common interface between UP and CP, is bundled with the existing Cisco IOS XR image to transform an integrated physical BNG router to a cnBNG user plane.

For more information about the cnBNG UP, see the *Cloud Native BNG User Plane Overview* chapter.

License Information

cnBNG supports the following licenses:

License	Description
Application Base	Per cluster
Session (Increments)	Network-wide

These are the software license PIDs for cnBNG:

cnBNG CP Essential Tier Licenses

Product IDs	Description
CN-BNG-100k-L	Session scale for 100,000 subscribers (network-wide) base licenses
CN-BNG-400k-L	Session scale for 400,000 subscribers (network-wide) base licenses
CN-BNG-1M-L	Session scale for 1,000,000 subscribers (network-wide) base licenses
CN-BNG-2M-L	Session scale for 2,000,000 subscribers (network-wide) base licenses

cnBNG CP Advantage Tier Licenses

Product IDs	Description
CN-BNG-ADV-100k-L	Session scale for 100,000 subscribers (network-wide) advantage licenses

Product IDs	Description
CN-BNG-ADV-400k-L	Session scale for 400,000 subscribers (network-wide) advantage licenses
CN-BNG-ADV-1M-L	Session scale for 1,000,000 subscribers (network-wide) badvantagease licenses
CN-BNG-ADV-2M-L	Session scale for 2,000,000 subscribers (network-wide) advantage licenses

Cisco cnBNG User Planes:

Refer the ASR9000 data sheet for ordering information:
https://www.cisco.com/c/en/us/products/routers/asr-9000-series-aggregation-services-routers/datasheet-listing.html

Standard Compliance

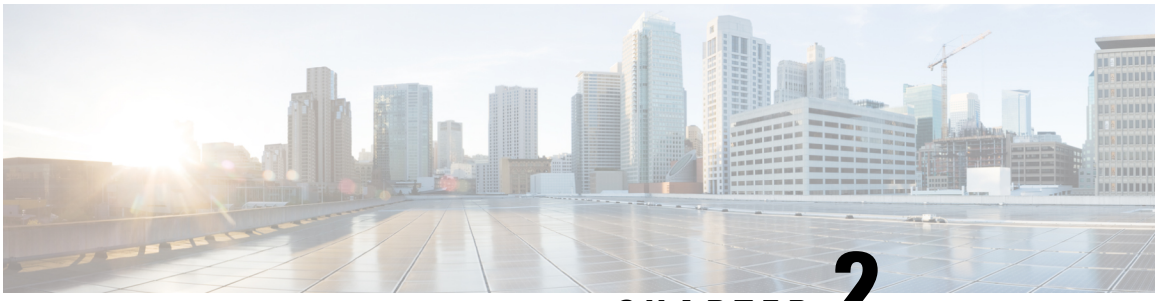
cnBNG solution is aligned with the following standard:

TR-459 Control and User Plane Separation for a disaggregated BNG

Limitations and Restrictions

The cnBNG has the following limitations and restrictions in this release:

- Only one subnet is supported per VRF.
- QoS provisioning is supported only through service.



CHAPTER 2

Deployment Options For Integrating cnBNG with Data Center Fabrics

This chapter describes how cnBNG integrates with modern data center fabrics to provide high availability, flexibility, and reliable services. To integrate cnBNG into the network, you need a few Bridge Domains (BDs) and Border Gateway Protocol (BGP). The underlying infrastructure can be Layer 2 only—with routing managed by Data Center Routers (DCRs)—or a mix of Layer 2 and Layer 3. The main goal is to enable cnBNG to communicate with both the northbound and southbound networks, as well as with other cnBNG clusters.

You can choose from several integration options based on your data center infrastructure, such as VXLAN EVPN, Cisco ACI, or traditional vPC. Use this information to select the solution that best meets your network and business requirements.

- Flexible deployment models, on page 9
- cnBNG on VXLAN BGP EVPN Fabric, on page 12
- cnBNG on Cisco ACI Fabric, on page 12
- cnBNG on Traditional vPC and LACP Networks, on page 13

Flexible deployment models

Cisco cnBNG is inherently fabric-agnostic and performance-sensitive, making it suitable for a variety of deployment scenarios:

Supported deployment models:

- Modern VXLAN-based fabrics
- Cisco ACI policy-driven fabrics
- Traditional vPC + LACP designs

cnBNG offers deployment flexibility. It operates on a Kubernetes cluster of servers, which can be as simple as a single node (All-in-One) or scaled across multiple nodes for higher availability. Each node is dual-homed to the data center fabric, ensuring both link and device redundancy for uninterrupted service.

Figure 4: Topology: cnBNG with VXLAN-based fabrics or ACI policy-driven fabrics

Figure 5: Topology: cnBNG with L2 domain or vPC LACP networks

cnBNG on VXLAN BGP EVPN Fabric

For organizations leveraging VXLAN BGP EVPN, cnBNG integrates with a spine-leaf architecture built on Cisco Nexus switches.

Architecture overview

- **Spines** provide Layer-3 routing, ECMP transport, and act as BGP Route Reflectors.
- **Leafs** serve as VXLAN Tunnel Endpoints (VTEPs) and connect directly to cnBNG servers.

Fabric structure

- The underlay uses OSPF or eBGP to ensure IP reachability across all fabric nodes.
- The overlay leverages VXLAN encapsulation and MP-BGP EVPN to manage tunnel endpoints and advertise service routes.

Resiliency and convergence

- EVPN multihoming (ESI-based) ensures active-active connections between cnBNG servers and leaf pairs.
- In the event of a link or leaf failure, traffic immediately converges to available paths without service interruption.

Control and data plane efficiency

- EVPN BGP broadcasts MAC/IP bindings, VTEP membership, and routed prefixes.
- VXLAN encapsulation provides a fully routed Layer-3 underlay and eliminates spanning-tree complexity.
- Distributed Anycast Gateway (DAG) maintains a consistent gateway presence across the fabric.

cnBNG on Cisco ACI Fabric

Key benefits of deploying cnBNG on Cisco ACI

- **High Availability and resiliency:** Supports multi-pod and multi-site architectures with active-active configurations to ensure seamless failover and business continuity.
- **Enhanced application performance:** Optimizes application flow and network performance with a policy-driven fabric and real-time health monitoring.
- **Redundancy:** Cisco ACI supports multi-site and multi-pod deployments with active-active border leaf nodes, ensuring high availability and seamless failover for cnBNG services.
- **Bridge Domains (BDs):** BDs in ACI provide flexible Layer 2 domains that can be stretched or localized, enabling efficient segmentation and endpoint isolation for cnBNG clusters.

- **L3Out:** ACI's L3Out feature offers scalable Layer 3 connectivity to external networks with support for routing protocols and host route advertisement, facilitating optimized north-south traffic flow for cnBNG.
- **Fabric-wide mobility:** The solution uses a consistent VXLAN overlay across the entire fabric.
- **Cost Efficiency:** Reduces operational costs by automating routine tasks, minimizing configuration errors, and enabling faster issue resolution.

Suitability

This solution is ideal for enterprise or multi-tenant environments that require strong compliance, segmentation, and centralized policy management.

cnBNG on Traditional vPC and LACP Networks

For smaller sites or brownfield environments, cnBNG can operate over vPC-based Layer-2 domains. In this scenario, each host bonds its NICs using LACP and connects to a pair of vPC peer switches.

Key benefits of deploying cnBNG on traditional vPC and LACP networks

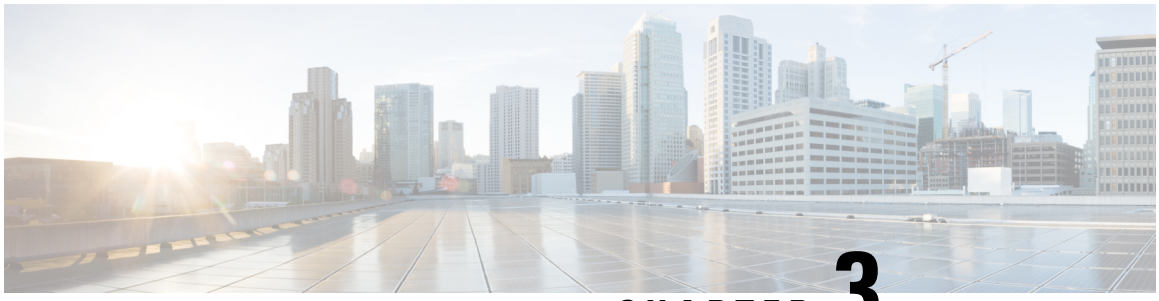
- **Redundancy:** Provides active-active operation facilitated by vPC peer-link coordination.
- **Compatibility:** Hosts require no overlay configuration, simplifying network integration.
- **Simplicity:** Deploys cnBNG nodes as a VLAN subnet over the vPC fabric, with gateways available at routers.

Advantages

- Straightforward deployment with immediate compatibility for legacy infrastructures.
- Reduces operational complexity by leveraging existing L2 technologies.

Suitability

This approach is optimal for legacy environments where rapid deployment and compatibility are priorities, but may be unsuitable for large-scale, highly scalable network fabrics requiring advanced routing capabilities.



CHAPTER 3

cnBNG Installation and Configuration

- Feature Summary and Revision History, on page 15
- Feature Description, on page 16
- Installing cnBNG and Accessing BNG Ops Center, on page 17
- Mapping Pods with Node Labels, on page 30
- High Availability Support on BareMetal Server, on page 31
- cnBNG CNF Deployment on AIO BareMetal Server, on page 31
- cnBNG Cluster Deployment Using Inception Server, on page 39
- cnBNG Cluster Deployment Support on Red Hat OpenShift, on page 46
- Maintenance Operation Procedure (MOP) for cnBNG Node Scaling, on page 48
- cnBNG Health-Check pods, on page 60

Feature Summary and Revision History

Summary Data

Table 1: Summary Data

Applicable Product(s) or Functional Area	cnBNG
Applicable Platform(s)	SMI
Feature Default Setting	Disabled - Configuration Required
Related Changes in this Release	Not Applicable
Related Documentation	Not Applicable

Revision History

Table 2: Revision History

Revision Details	Release
Introduced support for cnBNG cluster deployment on Red Hat OpenShift.	2025.01.0
Introduced Maintenance Operation Procedure (MOP) for cnBNG Node Scaling.	2024.03.0
Introduced support for cnBNG Cluster Deployment Using Inception Server.	2023.04.0
Introduced support for the cnBNG CNF Deployment on AIO BareMetal Server.	2022.02.0
cnBNG CP deployment on bare metal server is supported (with support for IPoE, PPPoE, LAC and LNS call models and High Availability) and fully qualified in this release.	2022.01.0
First introduced.	2021.01.0

Feature Description

This chapter describes cnBNG installation and configuration using the Ultra Cloud Core Subscriber Microservices Infrastructure (SMI) Cluster Manager and the BNG Operations (Ops) Center. The BNG Ops Center is based on the ConfD command line interface (CLI).

To install the SMI Cluster Manager, refer to the "Deploying the SMI Cluster Manager on VMware vCenter" section in the *Ultra Cloud Core Subscriber Microservices Infrastructure - Deployment Guide*.

The SMI Ops Center is the platform to install the cnBNG cluster with the offline or online repository. It is mandatory to install the SMI Ops Center to set up and access the BNG Ops Center.

Note To access the offline or online repository, contact your Cisco Account Manager or representative to get access to the offline or online repository.

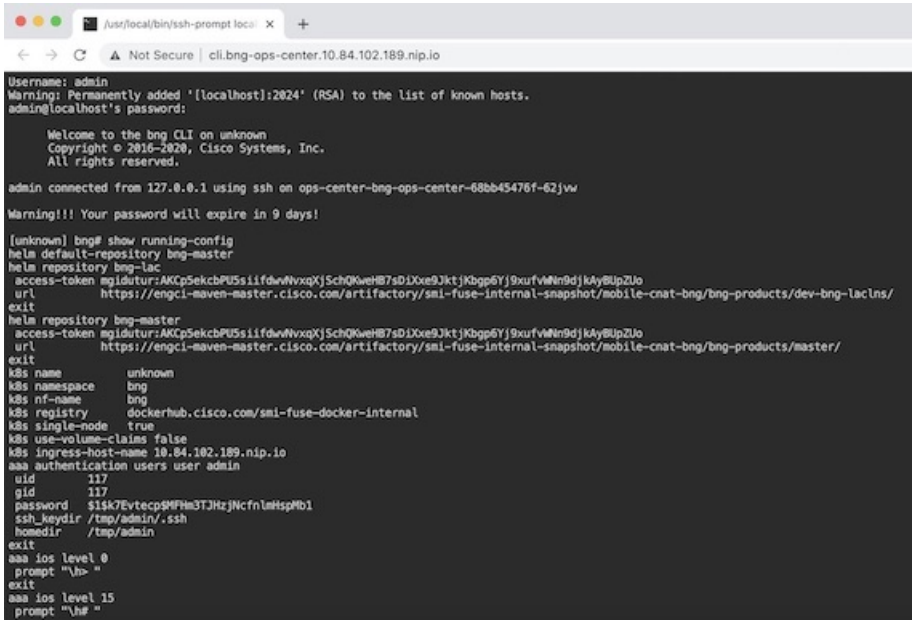
BNG Ops Center

The BNG Ops Center is a system-level infrastructure that provides the following functionality:

- A user interface to trigger a deployment of microservices with the flexibility of providing variable helm chart parameters to control the scale and properties of Kubernetes objects (deployment, pod, services, and so on) associated with the deployment.
- A user interface to push application-specific configuration to one or more microservices through Kubernetes configuration maps.

- A user interface to issue application-specific execution commands (such as show and clear commands). These commands:
  - Invoke some APIs in application-specific pods
  - Display the information returned on the user interface application

The following figure shows a sample of the web-based CLI presented to the user.



The BNG Ops Center allows you to configure features such as licensing, REST endpoint, and CDL.

For information on how to deploy BNG Ops Center on bare metal servers (currently Cisco UCS-C servers) environment, see "Operating the SMI Cluster Manager on Bare Metal" section in the *Ultra Cloud Core Subscriber Microservices Infrastructure — Operations Guide*.

Installing cnBNG and Accessing BNG Ops Center

This section describes how to install cnBNG and access the BNG Ops Center.

The Ultra Cloud Core SMI platform is responsible for setting up and managing the Cloud Native Broadband Network Gateway application.

Prerequisites

Before installing cnBNG on the SMI layer in an offline environment:

- Ensure that the SMI Cluster Manager all-in-one (AIO) is installed. This helps orchestrate the K8s Cluster and load the image.
- Ensure that all SMI K8s cluster nodes are in Ready state.

- Run the SMI synchronization operation for the BNG Ops Center and Cloud Native Common Execution Environment (CN-CEE).

  For CEE installation, refer to the *Ultra Cloud Core Common Execution Environment- Configuration and Administration Guide*.

- Ensure that the local repositories, which host the product offline TAR ball version, is installed.

System Requirements

Feature	Description
Disk Space	2 x 800 GB SSD (RAID 1) or equivalent input/output operations per second (IOPS) and redundancy.
Hardware	• High-performance x86 64-bit chipset • CPU performance Passmark benchmark of 13K rating per chip and 1,365 rating per thread, or better • VMware ESXi-compatible **Note** The following is recommended: • All the host servers should be UCSC-C220-M6SX, UCSC-C220-M7SX, or UCSC-C245-M8SX. • All the UCS systems should have SSD storge type.
Platform	VMware ESXi and VMware vCenter versions 6.5 and 6.7 **Note** SMI Cluster Manger support is qualified on the preceding platforms.
Memory	• At least DDR3-1600 or better than 1600 MT/s • ECC
Deployment Requirement	Hardware oversubscription, network saturation, or CPU oversubscription reduces application performance and productivity. The Cisco Ultra Cloud Core Subscriber Microservices Infrastructure detects and takes action when infrastructure requirements are not met.

Installing cnBNG in an Offline Environment

Using the SMI Cluster Manager, download the offline TAR ball of the cnBNG, the host and its charts, and corresponding images in the local registries. The SMI Cluster Manager supports the deployment of the BNG Ops Center and all the applications and services associated with it. This section describes the procedures involved in installing cnBNG in an offline environment using the SMI Cluster Manager.

To install cnBNG, complete the following steps:

1. Download the TAR ball from the URL.

software-packages download *URL*

Example:

```
SMI Cluster Manager# software-packages download
http://<ipv4address>:<port_number>/packages/bng-2021-02-1.tar
```

2. Verify whether the TAR balls are loaded.

   software-packages list

   Example:

```
BNG Cluster Manager# software-packages list
[ bng-2021-02-1 ]
[ sample ]
```

3. Configure the necessary SMI Ops Center parameters in the cluster to install cnBNG.

```
config
   cluster cluster_name
      ops-centers app_name instance_name
         repository url
         netconf-ip ipv4_address
         netconf-port port
         ssh-ip ipv4_address
         ssh-port port
         ingress-hostname <ipv4_address>.<customer_specific_domain_name>
         initial-boot-parameters use-volume-claims true/false
         initial-boot-parameters first-boot-password password
         initial-boot-parameters auto-deploy true/false
         initial-boot-parameters single-node true/false
         initial-boot-parameters image-pull-secrets
         exit
   exit
```

   Example:

```
SMI Cluster Manager# config
Entering configuration mode terminal
SMI Cluster Manager(config)# clusters cnbng-smi-cluster-01
SMI Cluster Manager(config-clusters-cnbng-smi-cluster-01)# ops-centers bng bng
SMI Cluster Manager(config-ops-centers-bng/bng)# repository
https://charts.10.10.105.50.nip.io/bng-2021.02.1
SMI Cluster Manager(config-ops-centers-bng/bng)# ingress-hostname 10.10.105.34.nip.io
SMI Cluster Manager(config-ops-centers-bng/bng)# initial-boot-parameters use-volume-claims
 true
SMI Cluster Manager(config-ops-centers-bng/bng)# initial-boot-parameters
first-boot-password test123
SMI Cluster Manager(config-ops-centers-bng/bng)# initial-boot-parameters auto-deploy
false
SMI Cluster Manager(config-ops-centers-bng/bng)# initial-boot-parameters single-node
false
SMI Cluster Manager(config-ops-centers-bng/bng)# exit
SMI Cluster Manager(config-clusters-cnbng-smi-cluster-01)# exit
SMI Cluster Manager(config)#
```

4. Configure the secrets, if your local registry contains secrets.

```
config
   cluster cluster_name
      secrets docker-registry secret_name
```

```
        docker-server server_name
        docker-username username
        docker-password password
        docker-email email
        namespace k8s namespace
        commit
        exit
    exit
```

Example:

```
SMI Cluster Manager# config
SMI Cluster Manager(config)# clusters test2
SMI Cluster Manager(config-clusters-test2)# secrets docker-registry sec1
SMI Cluster Manager(config-docker-registry-sec1)# docker-server serv1
SMI Cluster Manager(config-docker-registry-sec1)# docker-username user1
SMI Cluster Manager(config-docker-registry-sec1)# docker-password Cisco@123
SMI Cluster Manager(config-docker-registry-sec1)# docker-email reg@cisco.com
SMI Cluster Manager(config-docker-registry-sec1)# bng bng
SMI Cluster Manager(config-docker-registry-sec1)# exit
SMI Cluster Manager(config-clusters-test2)# exit
SMI Cluster Manager(config)#
```

5. Run the cluster synchronization.

   clusters *cluster_name* **actions sync run**

   Example:

   ```
   SMI Cluster Manager# clusters cnbng-smi-cluster-01 actions sync run
   ```

Notes:

- **software-packages download** *url*–Specifies the software packages to be downloaded through HTTP/HTTPS.
- **software-packages list**–Specifies the list of available software packages.
- **ops-centers** *app_name instance_name*–Specifies the BNG Ops Center and instance. *app_name* is the application name. *instance_name* is the name of the instance.
- **repository** *url*-Specifies the local registry URL for downloading the charts.
- **netconf-ip** *ipv4_address*–Specifies the BNG Ops Center netconf IPv4 address.
- **netconf-port** *port*–Specifies the BNG Ops Center netconf port number.
- **ssh-ip** *ipv4_address*–Specifies the SSH IPv4 address for the BNG Ops Center.
- **ssh-port** *port*–Specifies the SSH port number for the BNG Ops Center.
- **ingress-hostname** <*ipv4_address*>.<*customer_specific_domain_name*>–Specifies the ingress hostname to be set to the BNG Ops Center. <*customer_specific_domain_name*> specifies the domain name of the customer.
- **initial-boot-parameters**–Specifies the initial boot parameters for deploying the helm charts.
  - **use-volume-claims** *true/false*–Specifies the usage of persistent volumes. Set this option to True to use persistent volumes. The default value is true.
  - **first-boot-password** *password*–Specifies the first boot password for the product's Ops Center.

- **auto-deploy** *true/false*–Auto deploys all the services of the product. Set this option to false to deploy only the product's Ops Center.
- **single-node** *true/false*– Specifies the product deployment on a single node. Set this option to false for multi node deployments.
- **image-pull-secrets**–Specifies the docker registry secret name to be used.

- **secrets docker-registry** *secret_name*–Specifies the secret name for your docker registry.
  - **docker-server** *server_name*–Specifies the docker server name.
  - **docker-username** *username*–Specifies the docker registry user name.
  - **docker-password** *password*–Specifies the docker registry password.
  - **docker-email** *email*–Specifies the docker registry email.
  - **namespace** *namespace*–Specifies the docker registry namespace.

Verifying the cnBNG Installation

Verify the status of the cnBNG installation deployment through the cnBNG CLI. To verify, use the following commands:

1. Log in to the cnBNG product CLI.
2. Verify whether the charts are loaded in the specific instance (verify the namespace).

   `show helm charts`

   Example:

   ```
   bng# show helm charts
   CHART    INSTANCE  STATUS    VERSION  REVISION  RELEASE   NAMESPACE
   ------------------------------------------------------------------------
   infra-charts - DEPLOYED 0.0.6-rel-2021-01-0073-210208130850-fac5207 1 bng-bng-infra-charts
    bng-bng
   oam-pod - DEPLOYED 0.1.2-rel-2021-01-0144-210122165946-fcb74ed 1 bng-bng-oam-pod bng-bng
   bng-dashboard - DEPLOYED 0.0.1-rel-2021-01-0039-210122165311-0d542be 1
   bng-bng-bng-dashboard bng-bng
   etcd-cluster - DEPLOYED 0.7.0-0-7-0060-210203074532-f118407 1 bng-bng-etcd-cluster bng-bng
   ngn-datastore - DEPLOYED 1.3.0-1-3-0782-210125161812-f50a892 1 bng-bng-ngn-datastore
   bng-bng
   ```

3. Verify the status of the system.

   `show system status`

   Example:

   ```
   bng# show system status
   system status deployed true
   system status percent-ready 100.0
   ```

Notes:

- **show helm charts**–Displays the helm release details.
- **show system status**–Displays the status of the system.

Accessing BNG Ops Center

You can connect to the BNG Ops Center through SSH.

ssh admin@ops_center_pod_ip **-p 2024**

Day 0 Configuration

To view the Day 0 configuration, run the following command.

show running-config

The following is a sample Day 0 configuration:

```
deployment
 app-name     BNG
 cluster-name Local
 dc-name      DC
exit
k8 bng
 etcd-endpoint      etcd:2379
 datastore-endpoint datastore-ep-session:8882
 tracing
  enable
  enable-trace-percent 30
  append-messages      true
  endpoint             jaeger-collector:9411
 exit
exit
k8 label protocol-layer key smi.cisco.com/node-type value protocol
exit
k8 label service-layer key smi.cisco.com/node-type value service
exit
k8 label cdl-layer key smi.cisco.com/node-type value session
exit
k8 label oam-layer key smi.cisco.com/node-type value oam
exit
instances instance 1
 system-id  DC
 cluster-id Local
 slice-name 1
exit
local-instance instance 1
system mode shutdown
helm default-repository base-repos
helm repository base-repos
 url
https://engci-maven-master.cisco.com/artifactory/smi-fuse-internal-snapshot/mobile-cnat-bng/bng-products/master/

 username smf-deployer.gen
 password ***
exit
k8s name          svi-cn-bng-tb3
k8s namespace     bng-bng
k8s nf-name       bng
k8s registry      dockerhub.cisco.com/smi-fuse-docker-internal
k8s single-node   false
k8s use-volume-claims true
k8s ingress-host-name 10.81.103.86.nip.io
aaa authentication users user admin
 uid        1117
 gid        1117
 password   $1$vDWeJvJm$v46wiBWqdOj7eWgoPoZZE/
```

```
 ssh_keydir /tmp/admin/.ssh
 homedir    /tmp/admin
exit
aaa ios level 0
 prompt "\h> "
exit
aaa ios level 15
 prompt "\h# "
exit
aaa ios privilege exec
 level 0
  command action
  exit
  command autowizard
  exit
  command enable
  exit
  command exit
  exit
  command help
  exit
  command startup
  exit
 exit
 level 15
  command configure
  exit
 exit
exit
nacm write-default deny
nacm groups group admin
 user-name [ admin ]
exit
nacm rule-list admin
 group [ admin ]
 rule any-access
  action permit
 exit
exit
nacm rule-list confd-api-manager
 group [ confd-api-manager ]
 rule any-access
  action permit
 exit
exit
nacm rule-list ops-center-security
 group [ * ]
 rule change-self-password
  module-name       ops-center-security
  path              /smiuser/change-self-password
  access-operations exec
  action            permit
 exit
 rule smiuser
  module-name       ops-center-security
  path              /smiuser
  access-operations exec
  action            deny
 exit
exit
```

CP and UP Service Configuration

The CP service requires the basic configuration to process the API calls.

Note For information about the User Plane service configuration, refer to the *Cloud Native BNG User Plane Configuration Guide for Cisco ASR 9000 Series Routers, IOS XR Release 7.3.x*

Configuring the CP

The CP configuration is provided using the Ops Center infrastructure.

The following is a sample CP configuration:

```
ipam
 instance 1
  source local
  address-pool <dummy-pool-1>
   vrf-name default
   ipv4
    split-size
     per-cache 8192
     per-dp    8192
    exit
    address-range 10.0.0.1 10.10.255.255
   exit
   ipv6
    address-ranges
     split-size
      per-cache 8192
      per-dp    8192
     exit
     address-range 2001:DB8::1 2001:DB8::ff00
    exit
    prefix-ranges
     split-size
      per-cache 8192
      per-dp    1024
     exit
     prefix-length 56
     prefix-range 2001:db0:: length 40
    exit
   exit
  exit
  address-pool dummy-pool2
   vrf-name default
   ipv6
    prefix-ranges
     split-size
      per-cache 8192
      per-dp    1024
     exit
     prefix-range 2001:DB8:: length 48
    exit
   exit
  exit
  address-pool slaac-radius
   vrf-name default
   ipv6
    prefix-ranges
```

```
      split-size
       per-cache 8192
       per-dp    1024
      exit
      prefix-range 2001:DB8:: length 48
     exit
    exit
   exit
   address-pool static-dummy-pool2
    vrf-name default
    static enable user-plane dummy-asr9k-1
    ipv6
     prefix-ranges
      split-size
       no-split
      exit
      prefix-range 2001:DB8:: length 48
     exit
    exit
   exit
  exit
 exit
cdl node-type session
cdl logging default-log-level error
cdl datastore session
 slice-names [ 1 ]
 endpoint replica 2
 endpoint settings slot-timeout-ms 750
 index replica 2
 index map 1
 slot replica 2
 slot map 2
 slot notification limit 300
exit
cdl kafka replica 1
!
profile dhcp dummy-dhcp-server1
 ipv4
  mode server
  server
   pool-name   dummy-pool-1
   dns-servers [ network-dns ]
   lease days 1
   lease hours 2
   lease minutes 3
  exit
 exit
 ipv6
  mode server
  server
   iana-pool-name dummy-pool-1
   iapd-pool-name dummy-pool-1
   lease days 10
   lease hours 0
   lease minutes 4
  exit
 exit
exit
profile pppoe pppoe-prof
 mtu                        1500
 ctrl-pkt-priority          7
 service-selection-disable true
 max-payload minimum 1492 maximum 1540
 session-limit max 64000 threshold 60000
```

```
exit
profile pppoe ppp1
 mtu 1492
exit
profile aaa dummy-aaa-prof
 authentication
  method-order [ dummy-ser-grp ]
 exit
 authorization
  type subscriber method-order [ dummy-ser-grp ]
  username identifier client-mac-address
  password <any-password>
 exit
 accounting
  method-order [ dummy-ser-grp ]
 exit
exit
profile server-group dummy-ser-grp
 radius-group dummy-ser-grp
exit
profile attribute-format attr1
 format-order [ client-mac-address ]
exit
profile attribute-format attr2
 format-order [ addr ]
exit
profile attribute-format attr3
 format-order [ username ]
exit
profile subscriber dummy-subs-prof
 dhcp-profile               dummy-dhcp-server1
 pppoe-profile              ppp1
 session-type               ipv4v6
 activate-feature-templates [ svc1 VOICE TV ]
 aaa authorize dummy-aaa-prof
exit
profile feature-template VOICE
 qos
  in-policy   VOICE_INGRESS
  out-policy  VOICE_EGRESS
  merge-level 40
 exit
exit
profile feature-template APPS
 httpr-policy ACCESS-PBR
exit
profile feature-template TV
 qos
  in-policy   TV_INGRESS
  out-policy  TV_EGRESS
  merge-level 50
 exit
exit
profile feature-template svc1
 vrf-name default
 ipv4
  mtu                  1492
  disable-unreachables
  verify-unicast-source reachable-via-rx
 exit
 ipv6
  mtu                  1492
  ingress-acl          ipv6-acl-in-1
  egress-acl           ipv6-acl-out-1
```

```
  disable-unreachables
  verify-unicast-source reachable-via-rx
 exit
 session-accounting
  enable
  aaa-profile       dummy-aaa-prof
  periodic-interval 1800
 exit
exit
!
profile feature-template HSI_100M_5MQ
 qos
  in-policy   HSI_UPLOAD_RATE_100MB_IN
  out-policy  HSI_DOWNLOAD_RATE_100MB_OUT
  merge-level 30
 exit
 service-accounting
  enable
  aaa-profile       aaa-prof1
  periodic-interval 120
 exit
exit
profile feature-template HSI_100M_30MQ
 qos
  in-policy   HSI_UPLOAD_RATE_100MB_IN
  out-policy  HSI_DOWNLOAD_RATE_100MB_OUT
  merge-level 30
 exit
 service-accounting
  enable
  aaa-profile       dummy-aaa-prof
  periodic-interval 1800
 exit
exit
profile radius
 algorithm first-server
 deadtime  2
 detect-dead-server response-timeout 30
 max-retry 3
 timeout   5
 server 10.1.2.3 1812
  type     auth
  secret   <password>
  priority 1
 exit

 server 2001::10:1:36:121 1812
   type     auth
   secret   cisco
   priority 1
 exit
 server 10.1.2.3 1813
  type     acct
  secret   <password>
  priority 1
 exit
 server 2001::10:1:36:121 1813
   type     acct
   secret   cisco
   priority 1
 exit
attribute
  nas-identifier CISCO-BNG
  nas-ip        <209.165.201.1>
```

```
  nas-ipv6 2001::250:56ff:fe95:658
  nas-ip user-plane-ip
  instance 1
  exit
 exit
 accounting
  attribute
   nas-ip <209.165.201.1>
   nas-ipv6 2001::10:1:7:95
  exit
 exit
 server-group dummy-ser-grp
  server auth 10.1.2.3 1812
  exit
  server acct 10.1.2.3 1813
  exit
  server auth 2001::10:1:36:121 1812
  exit
  server acct 2001::10:1:36:121 1813
  exit
 exit
exit
profile coa
 client 10.1.2.3
  server-key <password>
 exit
 client 2001::10:1:36:111
  server-key cisco
 exit
user-plane
 instance 1
  user-plane dummy-asr9k-1
   peer-address ipv4 209.165.201.3
   subscriber-profile dummy-subs-prof
  exit
  user-plane dummy-asr9k-2
   peer-address ipv4 209.165.201.2
   subscriber-profile dummy-subs-prof
  exit
 exit
exit
instance instance-id 1
 endpoint sm
  replicas 1
 exit
 endpoint l2tp-tunnel
  replicas 1
 exit
 endpoint nodemgr
  replicas 1
 exit
 endpoint n4-protocol
  replicas 1
  retransmission max-retry 1
 exit
 endpoint dhcp
  replicas 1
 exit
 endpoint pppoe
  replicas 1
 exit
 endpoint radius
  interface coa-nas
   vip-ip <209.165.201.1> vip-port 2000
```

```
   vip-ipv6 2001::10:1:39:191 vip-ipv6-port 2000
  exit
 exit
 endpoint udp-proxy
  vip-ip <209.165.201.1>
  vip-ipv6 2001::10:1:39.192
  interface n4
   sla response 180000
  exit
  interface gtpu
   sla response 180000
  exit
 exit
exit
logging level application error
logging level transaction error
logging level tracing error
logging name infra.affinity_cache.core level application off
logging name infra.application.core level application off
logging name infra.bgipcstream.core level application off
logging name infra.cache_client.core level application off
logging name infra.cdl_update_queue.core level application off
logging name infra.config.core level application off
logging name infra.diagnostic.core level application off
logging name infra.diagnostics.core level application off
logging name infra.dpd.core level application off
logging name infra.ds_client.core level application off
logging name infra.edr.core level application off
logging name infra.heap_dump.core level application off
logging name infra.ipc_action.core level application off
logging name infra.ipcstream.core level application off
logging name infra.memory_cache.core level application off
logging name infra.message_trace.core level application off
logging name infra.resource_monitor.core level application off
logging name infra.resource_monitor_load_factor.core level application off
logging name infra.rest_server.core level application off
logging name infra.session_cache.core level application off
logging name infra.topology.core level application off
logging name infra.topology_lease.core level application off
logging name infra.transaction.core level application off
logging name infra.virtual_msg_queue.core level application off
logging name infra.vrf_etcd_update.core level application off
deployment
 app-name     BNG
 cluster-name Local
 dc-name      DC
exit
k8 bng
 etcd-endpoint      etcd:2379
 datastore-endpoint datastore-ep-session:8882
 tracing
  enable
  enable-trace-percent 30
  append-messages      true
  endpoint             jaeger-collector:9411
 exit
exit
k8 label protocol-layer key smi.cisco.com/proto-type value protocol
exit
k8 label service-layer key smi.cisco.com/svc-type value service
exit
k8 label cdl-layer key smi.cisco.com/sess-type value cdl-node
exit
k8 label oam-layer key smi.cisco.com/node-type value oam
```

```
exit
resource cache-pod
 gomaxproc 8
exit
instances instance 1
 system-id  DC
 cluster-id Local
 slice-name 1
exit
local-instance instance 1
system mode <shutdown or running>
commit
end
```

Configuring the UP

The following is a sample UP configuration:

```
user-plane
 instance 1
  user-plane dummy-asr9k-1
   peer-address ipv4 209.165.201.3
   subscriber-profile dummy-subs-prof
  exit
  user-plane dummy-asr9k-2
   peer-address ipv4 209.165.201.2
   subscriber-profile dummy-subs-prof
  exit
 exit
exit
```

Loading Day1 Configuration

To load the Day 1 configuration for cnBNG, run the following command:

ssh admin*@ops_center_pod_ip* **-p 2024** *< Day1config.cli*

Note The **day1config.cli file** contains the necessary parameters required for the Day 1 configuration.

Alternatively, you can copy the configuration and paste it in the BNG Ops Center CLI to load the Day 1 configuration.

```
config
  <Paste the Day 1 configuration here>
  commit
  exit
```

Day1config.cli

The **day1config.cli file** file contains the Day 1 configuration for cnBNG. For a sample day1 configuration, see Configuring the CP, on page 24.

Mapping Pods with Node Labels

Prerequisites

- Ensure that the node labels are according to the pod deployment layout.
- Ensure that the external VIPs are according to the requirement of NF.
- Enable Istio for pod to pod traffic load balancing.

Node Labels are key and value pairs that are attached to nodes at cluster synchronization. Each node can have a set of key and value labels defined. Each key must be unique for a node. With labels, users can map their NF pods onto nodes in a loosely coupled manner.



Important

- The pod-level labeling configuration is applicable only when the cnBNG CP is deployed on a bare metal server.
- Ensure to configure the node label on the SMI cluster deployer before mapping the pods. Following is the sample command for master-1 labeling:

```
[cndp-clpnc-cm-cm-primary] SMI Cluster Deployer (config-nodes-master-1)# k8s node-labels
smi.cisco.com/svc-type bng-node
```

To map the pods with node labels, use the following sample configuration:

```
config
   k8 label protocol-layer key label_key value label_value
   k8 label service-layer key label_key value label_value
   k8 label cdl-layer key label_key value label_value
   k8 label oam-layer key label_key value label_value
   end
```

Following is an example configuration of pod to node-label mapping:

```
k8 label protocol-layer key smi.cisco.com/node-type value bng-proto
exit
k8 label service-layer key vm-type value bng-svc
exit
k8 label cdl-layer key smi.cisco.com/node-type value bng-cdl
exit
k8 label oam-layer key smi.cisco.com/node-type value oam
exit
```

High Availability Support on BareMetal Server

High Availablity on cnBNG CP is validated on BareMetal server deployment. For more informaton about High Availablity, see High Availability and CP Reconciliation, on page 265.

cnBNG CNF Deployment on AIO BareMetal Server

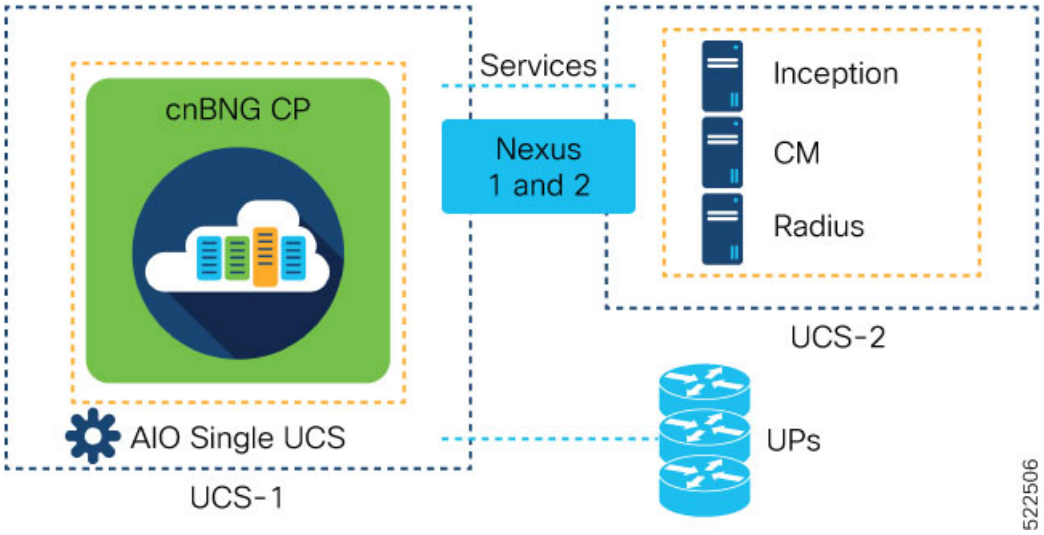
The cnBNG CNF Deployment on AIO BareMetal Server explains the process of onboarding a cnBNG Cloud Native Function (CNF) on the Cloud Native Deployment Platform (CNDP) on the BareMetal all-in-one (AIO) Kubernetes (K8s) cluster.

In the AIO deployment, all the management VMs are hosted on a different UCS server, however, this depends on the deployment strategy.

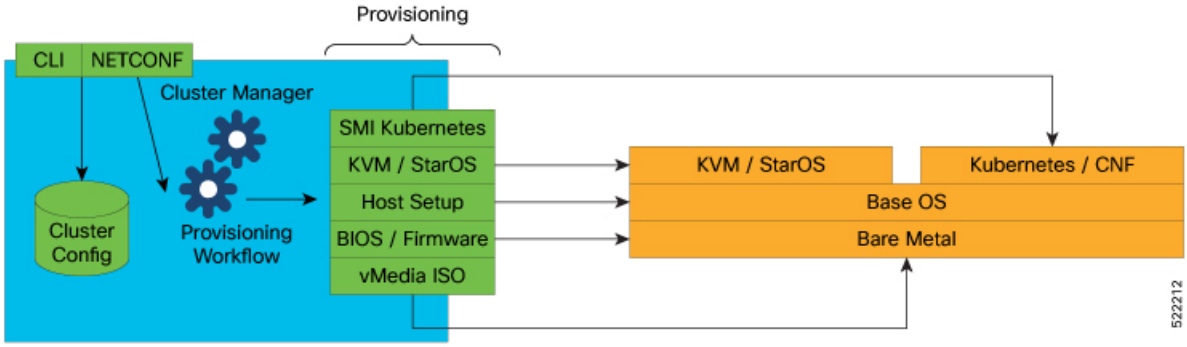
The cnBNG CNF is hosted on another UCS server referred as AIO server. During installation, the Cluster Manager (CM) accesses the AIO via the Cisco Integrated Management Controller (IMC) interface and adds the respective image and SMI packages to complete the installation.

Note The management VMs are the Inception, Cluster Manager, and RADIUS servers.

Figure 6: Logical Topology for cnBNG CNF Deployment on AIO BareMetal Server



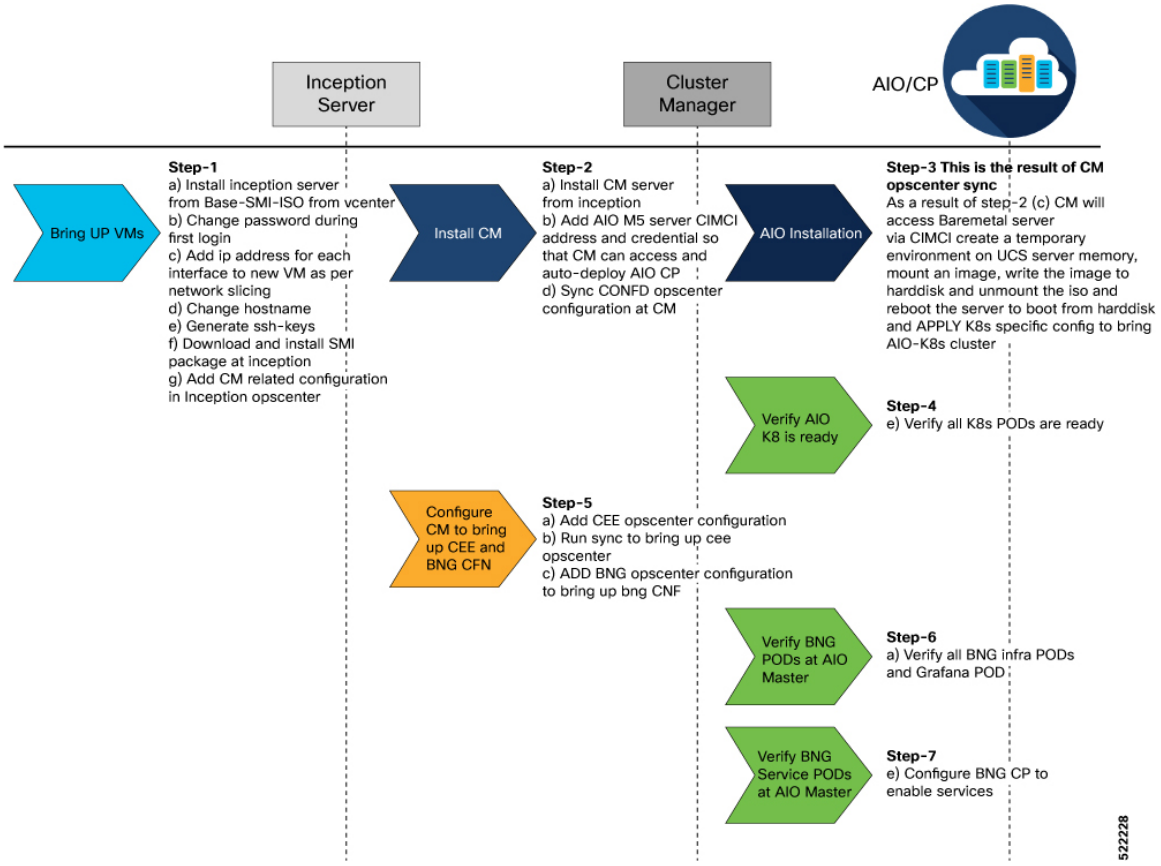
The CNDP is a 'SMI Bare-Metal'. The Cluster Manager uses REST APIs (instead of VIM APIs) of the management cards, which are on the servers, to create a set of Linux servers and then loads the K8s software.



Before installing network functions (NF) on the Cluster Manager, the common containerized software from SMI is installed. For example, monitoring and logging. The SMI NFs include their own common containerized software.

BareMetal CNDP AIO Bring-Up Procedure

The following figure illustrates the step-by-step process that is required to bring up the cnBNG CNF on K8s AIO server.



For more information about the Inception, Cluster Manager, and All-in-One server installation, see the "SMI Cluster Manager - Deployment" chapter of the *Ultra Cloud Core Subscriber Microservices Infrastructure - Deployment Guide*.

Implementing cnBNG CP Validation with CNDP

Implementing cnBNG CP Validation with CNDP involves the following procedures.

- Prerequisites
- Instantiating and Provisioning Inception Server Instance
- Installing the Cluster Manager Node
- Deploying the All-in-One Cluster
- Integrating RADIUS and UP with the AIO BareMetal Server

Prerequisites

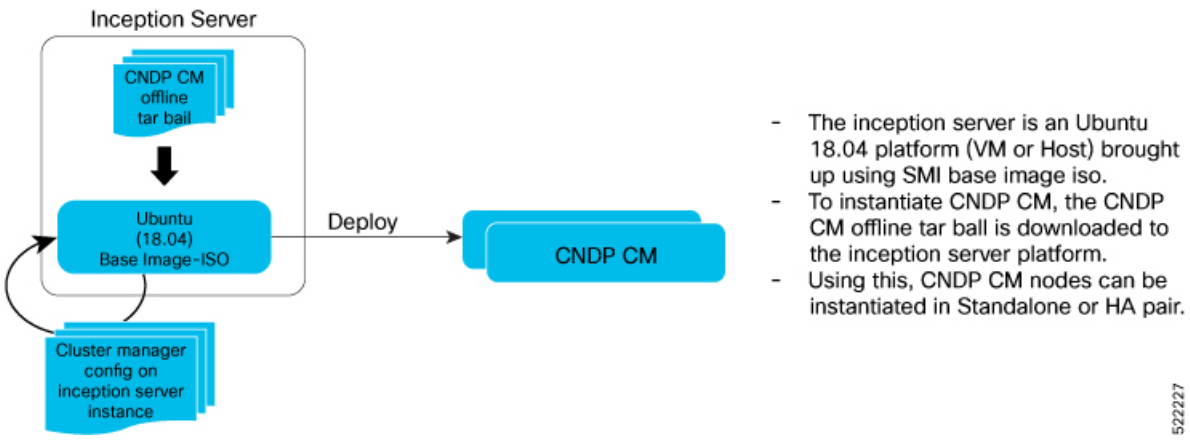
The following sections detail the prerequisites for deploying the cnBNG CNF on the AIO BareMetal Server.

Instantiating and Provisioning Inception Server Instance

The Inception server is used to deploy the CM for CNDP deployment. It is an Ubuntu 18.04 based VM installed with additional packages such as, docker, docker-compose, and its dependencies. The offline tar ball for the CNDP CM is installed on this instance and configured to deploy the CNDP CM nodes.

The following section presents the procedure on how to bring up an Inception server instance on a VM.

Figure 7: CNDP Inception VM



Before beginning the configuration of the Inception server, verify that all the dependent packages such as docker, docker-compose are installed on the VM. This is a prerequisite before loading the tar ball to configure and deploy the CM. If the SMI Base-ISO is used for installing Inception server, the packages are preinstalled.

Installing Cluster Manager Node

The Cluster Manager (CM) handles the installation and upgrade of the Kubernetes (K8s) cluster and associated infrastructure. In this deployment, the Inception server launches two machines that use DRBD to replicate the state to provide High Availability (HA) of the CM. This section covers the bring up procedure of the CM in standalone mode.

For High Availability CM deployment, see the "SMI Cluster Manager - Deployment" chapter of the *Ultra Cloud Core Subscriber Microservices Infrastructure - Deployment Guide*.

Configuring the Cluster Manager - Single Instance

Configure the single instance of the CM:

1. Login to the Ops Center CLI of the Inception server. Use the following steps to install and configure the CM.

   Use the IP address of the Inception server to login.

   ```
    ssh admin@<ip_address> -p 2022   (or)
   https://cli.smi-deployer.<ip_address>.nip.io
   ```

2. Configure the Inception server cluster deployer to install the CM.

Note Ensure the following before configuring the CM.

- All the passwords must be typed manually because copying and pasting the encrypted passwords from the configuration throws an error during validation
- Private key and public key must be generated in the Inception VM and copied. Both the keys must match the keys present in the Inception VM. Use the **ssh-keygen** command in the Inception VM and follow the prompts to generate the keys.
- Private key and public key are multiline commands. For instance, after typing **node-defaults ssh-connection- private-key**, press Enter to paste the keys.

For the Inception configuration, see the "SMI Cluster Manager - Deployment" chapter of the *Ultra Cloud Core Subscriber Microservices Infrastructure - Deployment Guide*.

Note Modify IPs, password, keys , username , cluster-name, and CNF name based on the specified configurations.

Installing the All-in-One Cluster

This section provides the detailed steps to deploy the CNDP AIO (K8s) cluster from the CM node. It also specifies the CLI that is used on the CM to configure and perform a cluster synchronization operation.

Configuring the AIO Cluster

This section describes the procedure to configure and deploy the CNDP AIO cluster from the CM using the CLI method.

1. Login to the CM Ops Center and load the SMI cluster, cnBNG, CEE, and Ops Center node configurations. Multiline configuration for private key must be pasted separately.

   ```
   ssh admin@<ip_address> -p 2022
   ```

2. Update the Sha256 value, which was generated for the software from the previous step, in the configuration for the respective software under the sha256 section.

Note

- All the passwords must be typed manually because copying and pasting the encrypted passwords from the configuration throws an error during validation
- Private key and public key must be generated in the CM and copied in. Both the keys must match the keys present in the CM. Use the **ssh-keygen** command in the Inception VM and follow the prompts to generate the keys.

3. Before running cluster synchronization, enable detail logging using the following configuration.

   ```
   clusters <cluster_name>
    configuration restrict-logging false exit
   ```

4. From the SMI cluster configuration, configure the Software CNF repository to use the cnBNG image, CEE, and include the sha256 checksum as generated previously and provide the path of the image.

   For more information, see the "SMI Cluster Manager - Deployment" chapter of the *Ultra Cloud Core Subscriber Microservices Infrastructure - Deployment Guide*.

```
[inception] SMI Cluster Deployer# show running-config
software cnf <cnf_software_version>c
 url      <repo_url>
 user     <user_name>
 password <password>
 sha256   <SHA256_hash_key>
exit
```

   Example:

```
Cluster Manager# config
Cluster Manager(config)# software cnf <example=cm-2020-02-0-i06>
Cluster Manager(config)# url <repo_url>
Cluster Manager(config)#user <username>
Cluster Manager(config)#password "<password>"
Cluster Manager(config)#sha256 <sha256_key>
Cluster Manager(config)#exit
```

   In this deployment model, a single AIO node is deployed.

   From a CM configuration perspective, the AIO node definition, corresponding Ops Center CEE and cnBNG instances are defined as part of a single AIO cluster.

   The following configuration snippet shows the sample configuration for a cluster from the cluster manager

```
config
  software cnf <cnf_software_version>
  url <repo_url>
  user <user_name>
  password <password>
  sha256 <SHA256_hash_key>
  exit
environments bare-metal
 ucs-server
exit
clusters <cluster_name> #For example, cndp-testbed
 environment bare-metal
 addons ingress bind-ip-address <IPv4address>
 addons cpu-partitioner enabled
 configuration allow-insecure-registry true
 node-defaults ssh-username <username>
 node-defaults ssh-connection-private-key
  "-----BEGIN OPENSSH PRIVATE KEY-----\n
 <SSH_private_key>
  -----END OPENSSH PRIVATE KEY-----\n"
  node-defaults initial-boot default-user <username>
  node-defaults initial-boot default-user-ssh-public-key
   "<SSH_Public_Key>"
  node-defaults initial-boot default-user-password #For example, Csco123#
  node-defaults os proxy https-proxy <proxy_server_url>
  node-defaults os proxy no-proxy <proxy_server_url/IPv4address>
  node-defaults os ntp enabled
  node-defaults os ntp servers <ntp_server>
  exit
 node-defaults initial-boot netplan ethernets <interface_name> #For example, eno1
  dhcp4 false
  dhcp6 false
  gateway4 <IPv4address>
  nameservers search <nameserver>
```

```
 nameservers addresses <IPv4addresses>
exit
node-defaults initial-boot netplan ethernets eno2 # same like eno1 other interfaces to
be configured
 dhcp4 false                                      # without any ip address
 dhcp6 false
exit
node-defaults initial-boot netplan ethernets eno5
 dhcp4 false
 dhcp6 false
exit
node-defaults initial-boot netplan ethernets eno6
 dhcp4 false
 dhcp6 false
exit
node-defaults initial-boot netplan ethernets enp216s0f0
 dhcp4 false
 dhcp6 false
exit
node-defaults initial-boot netplan ethernets enp216s0f1
 dhcp4 false
 dhcp6 false
exit
node-defaults initial-boot netplan ethernets enp94s0f0
 dhcp4 false
 dhcp6 false
exit
node-defaults initial-boot netplan ethernets enp94s0f1
 dhcp4 false
 dhcp6 false
exit
node-defaults initial-boot netplan vlans <vlan_name> #For example, vlan309
 dhcp4 false
 dhcp6 false
 id    <vlan_id> #For example, 309
 link  eno6
exit
node-defaults initial-boot netplan vlans <vlan_name> #For example, vlan310
 dhcp4 false
 dhcp6 false
 id    <vlan_id> #For example, 310
 link  eno6
exit
node-defaults initial-boot netplan vlans <vlan_name> #For example, vlan311
 dhcp4 false
 dhcp6 false
 id    <vlan_id> #For example, 311
 link  enp94s0f0
exit
node-defaults ucs-server cimc user admin
node-defaults ucs-server cimc storage-adaptor create-virtual-drive true
node-defaults ucs-server cimc remote-management sol enabled
node-defaults ucs-server cimc remote-management sol baud-rate 115200
node-defaults ucs-server cimc remote-management sol comport com0
node-defaults ucs-server cimc remote-management sol ssh-port 2400
node-defaults ucs-server cimc networking ntp enabled
node-defaults ucs-server cimc networking ntp servers <example: ntp.server1.com>
exit
node-defaults ucs-server cimc networking ntp servers <example: ntp.server2.com>
exit
node-defaults os ntp enabled
node-defaults os ntp servers <example: ntp.server1.com>
exit
node-defaults os ntp servers <example: ntp.server1.com>
```

```
exit

 nodes <aio>  #For example it can be master or aio
 k8s node-type master
 k8s ssh-ip <IPv4address>
 k8s node-ip <IPv4address>
 k8s node-labels disktype ssd
 exit
 k8s node-labels smi.cisco.com/node-type oam
 exit
 ucs-server cimc user admin
 ucs-server cimc password <IPv4address> #this CIMCI address of the AIO UCS SERVER
 ucs-server cimc ip-address 10.81.103.117
 initial-boot netplan ethernets eno1
 addresses [ <IPv4address-mgmt>/24 ]
 gateway4    <gateway-address>
 exit
 initial-boot netplan vlans vlan309
 addresses [ <IPv4address-k8s>/24 ]
 exit
 initial-boot netplan vlans vlan310
 addresses [ <IPv4address-SMI>/24 ]
 exit
 initial-boot netplan vlans vlan311
 addresses [ <IPv4address-services>/24 ]
 exit
exit
```

Each CNF provides a ConfD based Ops Center CLI to configure and manage the CNF pods. There is a separate Ops Center required for each CNF deployed on the AIO node.

The following is the Ops Center configuration for the AIO node, which has the Ops Center configuration for CEE and CNF.

```
ops-centers bng bng
  repository       <url> or offline-tarball
  username         <username>
  password         <password>
  ingress-hostname <ip-address>.nip.io
  initial-boot-parameters use-volume-claims false
  initial-boot-parameters first-boot-password <password>
  initial-boot-parameters auto-deploy false
  initial-boot-parameters single-node true
 exit
  ops-centers cee cee
  repository-local        cee-release-build
  sync-default-repository true
  netconf-ip            <ip-address>
  netconf-port            2024
  ssh-ip                   <ip-address>
  ssh-port                2022
  ingress-hostname        <ip-address>.nip.io
  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 true
 exit
exit
```

Note To bring the network function NF at the AIO K8 cluster, always use the “initial-boot-parameters single-node true" option.

5. Run the cluster synchronization to deploy the cluster, cnBNG, and CEE Ops Centers

   ```
   clusters cndp-cm actions sync run debug true
   ```

   The cluster synchronization operation takes approximately 45 minutes to complete.

6. Monitor the cluster synchronization operation using the following command.

   ```
   monitor sync-logs cndp-cm
   ```

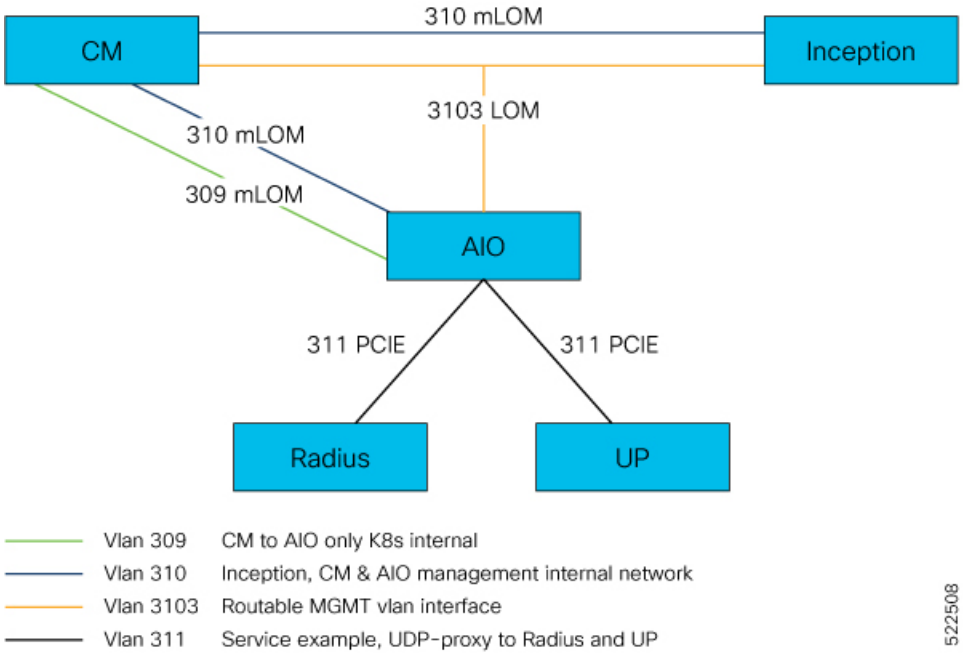
   After cluster synchronization is completed, a message is shown indicating a successful cluster synchronization.

Integrating RADIUS and UP with the AIO BareMetal Server

The RADIUS and UP are part of the services network and therefore should be part of the same network. If they are not in the same VLAN, then the necessary routing should be available to have reachability between them.

The AIO services interface is also part of the services VLAN, which has routable reachability between AIO UDP proxy interface, RADIUS, and the User Plane function (UPF).

Figure 8: Logical Network Connectivity



cnBNG Cluster Deployment Using Inception Server

You can now deploy the cnBNG cluster using the Inception server alone. You do not require the SMI Cluster Manager, which was previously used along with the Inception server to deploy the cnBNG cluster. This enhancement can help you save on hardware resources (servers).

Installing Inception Server on Baremetal

The procedure to install the the Inception server on baremetal is as follows:

1. Clear a Boot drive.
2. Create a virtual drive from unused physical drives.
3. Install Base ISO image.
4. Configure User and Network Parameters.
5. Install Inception Server.
6. Deploy SMI Cluster.
7. Add images to Inception Server.
8. Create SSH keys.
9. Add SMI Cluster Deployer configuration.

Clear a Boot Drive

You must clean up the server storage before installing the Base ISO image. To clear the boot drive configurations on the Cisco Integrated Management Controller (CIMC) server, perform the following steps:

1. Log in to the CIMC Web UI using admin privileges.
2. In the **Navigation** pane, click the **Storage** menu.
3. On the **Storage** menu, click the appropriate LSI MegaRAID or Host Bus Adapter (HBA) controller.

   In the **RAID Controller** area, the **Controller Info** tab is displayed by default.
4. In the **Actions** area, click **Clear Boot Drive**.



Note The **Clear Boot Drive** option is enabled only if the server was used previously. If the server is new, this option is disabled.

5. Click **OK** to confirm.

Create Virtual Drive from Unused Physical Drives

1. Log in to the CIMC Web UI using admin privileges.
2. On the **Storage** menu, click the appropriate LSI MegaRAID or HBA controller.
3. In the **Actions** area, click **Create Virtual Drive from Unused Physical Drives**.

   The **Create Virtual Drive from Unused Physical Drives** dialog box appears.
4. In the **Create Virtual Drive from Unused Physical Drives** dialog box, select **1** from the **RAID Level** drop-down list.
5. In the **Create Drive Groups** area, choose one or more physical drives to include in the group.

Use the >> button to add the drives to the **Drive Groups** table. Use the << button to remove physical drives from the drive group.



Note

- The size of the smallest physical drive in the drive group defines the maximum size used for all the physical drives. To ensure maximum use of space for all physical drives, it is recommended that the size of all the drives in the drive group are similar.
- CIMC manages only RAID controllers and not Host Bus Adapters (HBAs) attached to the server.
- You must have multiple drive groups available to create virtual drives (VDs) for certain RAID levels. While creating drives for these RAID levels, the create drive option is available only if the required number of drives are selected.

6. In the **Virtual Drive Properties** area, update the following properties:

Name	Description
Virtual Drive Name field	The name of the new virtual drive you want to create.
Read Policy drop-down list	The read-ahead cache mode.
Cache Policy drop-down list	The cache policy used for buffering reads.
Strip Size drop-down list	The size of each strip, in MB.
Write Policy drop-down list	This can be one of the following • **Write Through**— Data is written through the cache and to the physical drives. Performance is improved, because subsequent reads of that data can be satisfied from the cache. • **Write Back**— Data is stored in the cache, and is only written to the physical drives when space in the cache is needed. Virtual drives requesting this policy fall back to **Write Through** caching when the BBU cannot guarantee the safety of the cache in the event of a power failure. • **Write Back Bad BBU**—With this policy, write caching remains **Write Back** even if the battery backup unit is defective or discharged.
Disk Cache Policy drop-down list	This can be one of the following • **Unchanged**— The disk cache policy is unchanged. • **Enabled**— Allows IO caching on the disk. • **Disabled**— Disallows disk caching.
Access Policy drop-down list	This can be one of the following • **Read Write**— Enables host to perform read-write on the VD. • **Read Only**— Host can only read from the VD. • **Blocked**— Host can neither read nor write to the VD.

Name	Description
Size field	The size of the virtual drive (VD) you want to create. Enter a value and select one of the following units: • MB • GB • TB

7. Click **Create Virtual Drive**.
8. After the virtual drive is created, click the **Virtual Drive Info** tab.

   In the **Virtual Drives** area, choose the drive from which the controller must boot, and then click **Set as Boot Drive**.
9. Click **OK** to confirm.

Install Base ISO image

Perform the following steps to install the Base ISO image.

1. Log in to the CIMC Web UI using admin privileges.
2. Click the **Virtual KVM** tab.
3. In the **Actions** area, click **Launch KVM Console**.
4. Click the **Virtual Media** menu, and click **Activate Virtual Devices**. A virtual media session is activated, and that allows you to attach a drive or image file from your local computer or network.
5. Click the **Virtual Media** menu again, and select **Map CD/DVD**. You can map a CD or a DVD image from your local machine and map the drive to the image.
6. Browse, and select the Base ISO image from the image folder.
7. From the tool bar, click the **Host Power** link. Select **Power Cycle** from the drop-down list. The chosen server is powered on with the mapped Base ISO image.
8. From the tool bar, click **Launch KVM**. The **KVM Console** opens in a separate window.
9. Once you see the login prompt and other options, press the **F6** function key.
10. On the boot menu, select the CIMC based vKVM-Mapped option to which the Base ISO image is mapped.

    The server boots with the required ISO image.
11. Using CIMC, you can also configure the order in which the server attempts to boot from the available boot device types.
    a. Click the **Compute** menu.
    b. In the **BIOS** tab, click the **Configure Boot Order** tab. **Configure Boot Order** dialog box appears.
    c. In the **Configure Boot Order** dialog box, you can select the order of boot, and click **Save Changes**.

12. Power cycle the server to trigger the Base ISO installation. You can view the installation status in the KVM console.

Configure User and Network Parameters

Before installing the Inception server you must configure the user and network parameters.

To configure the network and static IP addressing:

1. Login with the default *cloud-init* credentials.



Note You must change the password immediately after logging in.

2. Update the network configuration in `/etc/netplan/50-cloud-init.yaml` file.

   The following is a sample network configuration:

```
network:
   ethernets:
     eno1:
       addresses:
      - 10.0.0.20/8
      dhcp4: false
      gateway4: 10.197.209.193
      nameservers:
          addresses:
        - 10.104.128.236
        - 10.163.128.140
        - 10.37.142.73
version: 2
```

3. Run the following command to apply the configuration:

```
sudo netplan apply
```

4. To preserve hostname or make the hostname persistent after reboot, you must edit the cloud configuration file:

```
sudo vi /etc/cloud/cloud.cfg
```

   Set **preserve_hostname: true**

5. Modify the hostname.

```
sudo hostnamectl set-hostname hostname
sudo vi /etc/hosts
127.0.0.1 hostname
```

Install Inception Server

Perform the following steps to install the Inception server:

1. Copy the SMI offline package to the images folder.

   Example:

```
cloud-user@cnbng4-cxbgl:/data/software/images$ pwd /data/software/images
cloud-user@cnbng4-cxbgl:/data/software/images$
```

2. Untar the the downloaded file.

   Example:

   ```
   cloud-user@cnbng4-cxbgl:/data/software/images$ tar -xvf
   cluster-deployer-2023.01.1.i18.tar
   ```

3. Navigate to the `deployer-inception` folder which has the required charts and docker files.

   Example:

   ```
   cd /data/deployer-inception/
   ```

4. Run the following command to install the Inception server.

   Example:

   sudo ./deploy --external-ip *external_ipaddress* **--first-boot-password** *first_boot_password*

Note The *external_ipaddress* is the management IP address of the inception server.

Deploy SMI Cluster

Perform the following steps to deploy the SMI cluster:

1. Log in to the cluster using the following:

   ```
   ssh admin@<ip_address> -p 2022
   ```

2. Add the cluster level configurations for one or more K8s clusters.

Add Images to Inception Server

1. Fetch the offline tarball for SMI, cnBNG, & CEE and save it to the `/data/software/images` folder. You can fetch the tarball either from the artifactory or copy it securely through the **scp** or **winscp** command.
2. Untar the offline tarball, and copy the tar file to the respective path.

   Example:

   ```
   root@cnbng-inception:/data/downloads# tar -xvzf bng.2020.04.m0.i37.SSA.tgz ./
   ./bng.2020.04.m0.i37.tar.SSA.README
   ./CNBNG_DEV_KEY-CCO_DEV.cer
   ./trca.cer
   ./Innerspace_DEV.cer ./cisco_x509_verify_dev.py ./bng.2020.04.m0.i37.tar.signature.SSA
   ./bng.2020.04.m0.i37.tar
   ```

3. Generate **sha256** checksum for the images and verify them with artifactory checksum.

   Example:

   ```
   cloud-user@cnbng4-cxbgl:/data/software/images$ sudo sha256sum bng.2020.04.m0.i37.tar
   2e4fe956daf4afa13909d6fa89be5e727b9e4c03619436ecd04805045b780c0b bng.2020.04.m0.i37.tar
   cloud-user@cnbng4-cxbgl:/data/software/images$ sudo sha256sum cee-2023.01.1.i18.tar
   320e61f56976a2c107fa489a2a12d16301671f28212ec5b7d902b608d2e6ab80 cee-2023.01.1.i18.tar
   ```

```
cloud-user@cnbng4-cxbgl:/data/software/images$ sudo sha256sum cluster-deployer-
2023.01.1.i18.tar
929dd80a840483f65a9f4afa314144f0f544e3bee23703214c03c831465ae707 cluster-deployer-
2023.01.1.i18.tar
```

4. Add the images to Inception deployer cluster configuration. The inception deployer uses cnBNG & CEE images from the provided file path to bring up cnBNG control plane & CEE ops-center.

   Example:

```
software cnf cnbng
url file:///data/software/images/bng.2020.04.m0.i37.tar
sha256 2e4fe956daf4afa13909d6fa89be5e727b9e4c03619436ecd04805045b780c0b
exit
software cnf cee-2023.01.1.i18
url file:///data/software/images/cee-2023.01.1.i18.tar
sha256 320e61f56976a2c107fa489a2a12d16301671f28212ec5b7d902b608d2e6ab80
exit
software host-profile bios-ht-25
url file:///data/software/images/ht.tgz
sha256 aa7e240f2b785a8c8d6b7cd6f79fe162584dc01b7e9d32a068be7f6e5055f664
exit
environments bare-metal
ucs-server
exit
!
```

Generate SSH Keys

Generate SSH public and private keys.

Example:

```
cloud-user@inception-28:~$ ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
Created directory '/root/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_rsa.
Your public key has been saved in /root/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:YQ3OfuEinrurnkkFlLW/vBDu5Zkti8zMt8dfIxpgYIE ubuntu@nfs-server-cn-svi
The key's randomart image is:
+---[RSA 2048]----+
| .+o . |
| .E = o |
| . + = o |
| o = o . |
| + S o |
| + * = |
| . = +.. . o |
| . O *o=oo o . |
| .=.@+B=o.. |
+----[SHA256]-----+
cloud-user@inception-28:~/.ssh$ ls -ltr ~/.ssh/*id*
-rw------- 1 cloud-user cloud-user 577 Jan 19 2023 /home/cloud-user/.ssh/id_rsa.pub
-rw------- 1 cloud-user cloud-user 2484 Jan 19 2023 /home/cloud-user/.ssh/id_rsa
```

Add SMI Cluster Deployer Configuration

Perform the following steps to add the SMI Cluster Deployer configurations:

1. Log in to the SMI cluster deployer.

   Example:

   ```
   cloud-user@incep.on-28:~$ ssh admin@127.0.0.1 -p 2022
   admin@127.0.0.1's password:
   Welcome to the Cisco SMI Cluster Deployer on inception-28
   Copyright Â© 2016-2020, Cisco Systems, Inc.
   All rights reserved.
   admin connected from 10.0.1.1 using ssh on c0993db6451a
   ```

2. Add SSH-keys to the Inception Manager.

   Example:

   ```
   clusters <cluster_name>
   environment bare-metal
   node-defaults initial-boot default-user-ssh-public-key
   "<SSH_Public_key>"
   node-defaults ssh-connection-private-key
   "-----BEGIN OPENSSH PRIVATE KEY-----
   <SSH_Private_Key>
   -----END OPENSSH PRIVATE KEY-----\n"
   CNTRL+D
   ```

3. Add CNF to clusters.

   You can either download CNF images from online repositories, web servers, or local servers, or copy the tarball images to the Inception server folder and configure the folder path.

4. You can verify the configuration using the following command:

   ```
   SMI Cluster Deployer# show running-config
   software cnf cee-2023.01.1.i18
   url file:///data/software/images/cee-2023.01.1.i18.tar
   sha256 320e61f56976a2c107fa489a2a12d16301671f28212ec5b7d902b608d2e6ab80
   exit
   software cnf cnbng
   url file:///data/software/images/bng-dev-private.tar
   sha256 2e4fe956daf4afa13909d6fa89be5e727b9e4c03619436ecd04805045b780c0b
   exit
   software host-profile bios-ht-25
   url file:///data/software/images/ht.tgz
   sha256 aa7e240f2b785a8c8d6b7cd6f79fe162584dc01b7e9d32a068be7f6e5055f664
   exit
   ```

cnBNG Cluster Deployment Support on Red Hat OpenShift

Starting with Release 2025.01, the cnBNG control-plane can also be deployed on a Red Hat OpenShift cluster.

Supported Functionalities:

The following functionalities are supported:

- OpenShift Container Platform (OCP) versions 4.16 and 4.17 are supported.
- Single Node OpenShift (SNO) is supported.
- Both cnBNG Standalone and CP-GR are supported.
- Only IPOE is supported.

- Compatible with IPv4-only and dual-stack configurations.

Requirements:

The following software versions are required:

- CEE version: 2025.01 or above
- cnBNG-CP version: 2025.01.0 or above

Monitoring and Management:

The following monitoring and management functionalities are available:

- Monitoring dashboards are available via CEE.
- Local storage support is provided through CEE.
- Cisco TAC debug support is available through CEE.
- Log and metric forwarding capabilities are provided via CEE.

ISTIO:

ISTIO is not supported.



Note This guide does not cover the installation of the OpenShift cluster or the setup of Helm and the Docker registry.

Set Up the cnBNG Application on an OCP Cluster

After installing the OpenShift Container Platform (OCP) cluster, follow these steps to set up the cnBNG application:

1. Add the hostname to the `/etc/hosts` file.
2. Assign the necessary node labels.
3. Execute the following command to enable support for external IP addresses:

```
oc patch network cluster --type merge -p '{ "spec": { "externalIP": { "policy":
{"allowedCIDRs": ["0.0.0.0/0", "::/0"]}}}}'
```

4. Add Helm and Docker repositories for the BNG and CEE charts.
5. Install the **cee-ops-center** and **bng-ops-center** charts using the specified inputs.

   Ensure that your configuration includes the following global settings:

```
global:
  registry: <docker-registry>
  istio: false
  singleNode: true
  useVolumeClaims: true
  caasModel: "openshift"
  pathBasedIngress: false
  smiNfMonitoring: true
  smiPlatformMonitoring: true
  ingressHostname: <ingress>
  imagePullPolicy: IfNotPresent
```

```
imagePullSecrets:
  - name: <secret-name>
networkPolicy:
  enabled: false
```

The CLI commands for `bng-ops-center` and `cee-ops-center` are consistent with CNDP, allowing you to enable additional functionalities using the same setup.

Maintenance Operation Procedure (MOP) for cnBNG Node Scaling

The current solution for a cnBNG Control Plane consists of two racks working together to provide high availability. Each rack contains four Kubernetes (K8s) clusters and is in the process of scaling up to a fifth Kubernetes cluster.

This MOP outlines the steps to scale a cnBNG Control Plane deployment from Half-Rack to Full-Rack.

The procedure includes

- adding new servers
- scaling up Kubernetes (K8s) clusters
- scaling up CDL, and
- scaling up services.

Before you begin

Perform these steps prior to the actual maintenance window.

- Generate configuration files for K8s node scale up.
  - Prepare and review the config change payload for scaling up each cluster.
  - Verify that label configuration is in line with the current pod.
  - Load and merge change payload files into the Deployer configuration and perform a commit dry-run.
  - Verify that all changes are as per the expectation.
  - Exit Deployer config mode without issuing a commit.

Note Actual commits are done during the maintenance window.

Estimated Time for Procedure

Time for	Time taken
Adding new servers Wiring up the new servers Configure the switches	1 day
Scale up – Rack1 – CDL+Service	90m
Scale up – Rack2 – CDL+Service	90m
Scale up – CDL	60 ~ 90m

Software Used

Software For	Software Version
cnBNG	CNBNG-2024.03.0

Adding and Wiring New Servers

1. Update the firmware.

   Ensure that new servers have the same firmware version as existing nodes.



Note This step should be done prior to the actual maintenance window before racking the servers in the pod.

2. Install and wire servers.
   - Install new servers in the same rack as existing servers.
   - Wire servers using available ports in current switches.
3. Power up servers.
4. Verify that there are no existing boot or virtual disks on these new servers.
5. Ensure that disks appear as Unconfigured-good.
6. Check the CIMC IP addresses of these new servers for connectivity.
7. Configure switch ports.
   - Configure leaf switch ports for new servers.

     Use the same configuration as other leaf ports connected to existing worker nodes.
   - Verify from the ClusterManager Linux shell that you can ping CIMC IP addresses of new nodes.

Scale Up Kubernetes Cluster

You must scale up the Kubernetes cluster in each rack of the pod.

Follow these steps to scale up the Kubernetes cluster:

1. Perform a planned GR switchover to move all cnBNG instances to the RackInService as primary.



Note RackInService refers to the rack hosting all primary GR instances of all cnBNGs in the pod.

2. Reset states of cnBNG GR instances in STANDBY_ERROR state to STANDBY state in RackBeingScaled.



Note RackBeingScaled refers to the rack undergoing K8s node scaling.

3. Verify that all GR instances in RackInService are in PRIMARY state and those in RackBeingScaled are in STANDBY state.
4. Check the output of **show idmgr-reconcile status** command to confirm a successful RoleSwitchover status in RackInService. If the status is `InProgress`, wait until it changes to `Success`.
5. Set geo-maintenance mode in all cnBNGs in both RackBeingScaled and RackInService.
6. Shut down all cnBNGs in RackBeingScaled.
7. Add the configuration in Deployer via CLI or NSO load merge config change payload files to scale up RackBeingScaled.

   See the section "K8s Cluster scale-up config example" for example configuration.
8. Manually add networks in the netplan on the new server.
   - Command: **sudo vi /etc/netplan/50-cloud-init.yaml**
   - Apply: **sudo netplan apply**



Note If the server is new, you don't have to add netplan manually. If you are reusing a server, write a netplan manually.

9. [**Optional**] If you are reusing an existing server, reset it using the CLI command:
   - **kubeadm reset --force**
   - Clean up (delete) the virtual drive by logging into the CIMC of that server.
10. Issue a cluster sync to scale up RackBeingScaled.
    - Command: **clusters svi-cnbng-gr-tb1 actions sync run debug true**
11. Monitor the cluster sync progress by logging into the Cluster Manager.
12. Verify if all nodes in the K8s cluster in RackBeingScaled are in ready state.
13. Verify if all pods in the K8s cluster in RackBeingScaled are in good running state.

14. Start up all cnBNGs in RackBeingScaled.
15. Verify if the added nodes appear in the cnBNG Ops Center configurations.
16. Verify if both GR instances of all cnBNGs in RackBeingScaled are in STANDBY state.
17. Check the output of **show idmgr-reconcile status** command to confirm a successful PodRestart status on the RackBeingScaled. If the status is `InProgress`, wait until it changes to `Success`.
18. Unset geo-maintenance mode in all cnBNGs in both RackBeingScaled and RackInService.

Post-Requirements:

Repeat this procedure to scale up K8s cluster in each rack of the Pod.

Scale Up CDL

Follow these steps to scale up the CDL:

1. Verify that all cnBNGs in the pod are in good state.
2. Update CDL Grafana dashboard to monitor the total number of index keys that have been rebalanced.
   - **Query**: sum(index_rebalanced_keys_total{namespace="$namespace"}) by (cdl_slice, shardId, pod)
   - **Legend**: irkt-{{cdl_slice}}-{{shardId}}-{{pod}}



Note CDL rebalancing happens once the CDL rebalance query is issued.

3. Make rack-1 as ACTIVE and rack-2 as STANDBY.
   a. Unset geo maintenance mode in all cnBNGs in both racks (if previously set).
   b. Make both GR instances of all cnBNGs in Rack1 PRIMARY.
   c. Make both GR instances of all cnBNGs in Rack2 STANDBY.
   d. Check the output of **show idmgr-reconcile status** command to confirm a successful RoleSwitchover status on the RackInService. If the status is `InProgress`, wait until it changes to `Success`.
   e. Set geo maintenance mode in all cnBNGs in both racks.
4. Add configuration changes to Rack2 (STANDBY)
   a. Shut down all cnBNGs in Rack2.
   b. Add configuration changes to all cnBNGs in Rack2 (see "*CDL Configuration changes*" section).
   c. Start up all cnBNGs in Rack2.
5. Verify the cnBNG State
   - Ensure that all cnBNGs are in good state.
   - Both GR instances in all cnBNGs in Rack2 should be in STANDBY state.

- Both GR instances in all cnBNGs in Rack1 should be in PRIMARY state.

6. Check the output of **show idmgr-reconcile status** command to confirm a successful PodRestart status on the RackBeingScaled. If the status is `InProgress`, wait until it changes to `Success`.
7. Verify CDL indices and slots sync
   a. Verify if CDL indices and slots in Rack2 can sync with remote peers (may take ~15 mins).
8. Switch rack states
   - Unset geo maintenance mode in all cnBNGs in both racks.
   - Make both GR instances of all cnBNGs in Rack2 PRIMARY.
   - Make both GR instances of all cnBNGs in Rack1 STANDBY.
   - Check the output of **show idmgr-reconcile status** command to confirm a successful RoleSwitchover status on the RackInService. If the status is `InProgress`, wait until it changes to `Success`.
   - Set geo maintenance mode in all cnBNGs in both racks.
9. Add configuration changes to Rack1 (STANDBY)
   a. Shut down all cnBNGs in Rack1.
   b. Add configuration changes to all cnBNGs in Rack1 (see "*CDL Configuration changes*" section).
   c. Start up all cnBNGs in Rack1.
10. Verify the state of all cnBNGs.
    - Ensure that all cnBNGs are in good state.
    - Both GR instances in all cnBNGs in Rack1 should be in STANDBY state.
    - Both GR instances in all cnBNGs in Rack2 should be in PRIMARY state.
11. Check the output of **show idmgr-reconcile status** command to confirm a successful PodRestart status on the RackBeingScaled. If the status is `InProgress`, wait until it changes to `Success`.
12. Verify if CDL indices and slots in Rack1 can sync with remote peers (may take ~15 mins).
13. Trigger CDL index rebalancing.
    - Issue the following command in all cnBNGs in the current STANDBY rack:

      cdl rebalance-index run
14. Verify CDL index rebalancing completion.
    a. Monitor progress with the following command in the STANDBY rack cnBNG:

       cdl rebalance-index status
    b. Validate rebalancing with the following command in the STANDBY rack cnBNG:

       cdl rebalance-index validate

15. Remove CDL scale up mode configuration:

```
config
  cdl datastore session
  no mode
  commit
  end
```

16. Unset Geo maintenance mode in all cnBNGs in both racks.
17. Initiate GR switchovers

    Switch all cnBNGs in the pod to ACTIVE-ACTIVE mode.
18. Verify that all cnBNGs are operating in sunny day mode.

Scale Up Services

You can now scale up cnBNG Services.

Note There is no restriction to always scale up Cluster1 before Cluster2. You can scale up any cluster first, depending on the existing role set on the cluster.

Follow these steps to scale up cnBNG services:

1. Make Rack-1 Active and Rack-2 Standby
   - Unset geo maintenance mode in all cnBNGs in both racks (if previously set).
   - Make both GR instances of all cnBNGs in Rack1 PRIMARY.
   - Make both GR instances of all cnBNGs in Rack2 STANDBY.
   - Check the output of **show idmgr-reconcile status** command to confirm a successful RoleSwitchover status on the RackInService. If the status is `InProgress`, wait until it changes to `Success`.
   - Set geo maintenance mode in all cnBNGs in both racks.
2. Add configuration changes in Rack-2 (STANDBY).
   a. Shut down all cnBNGs in Rack2.
   b. Add the configuration changes to all cnBNGs in Rack2 (see the "Service scaleup config changes" section).
   c. Start up all cnBNGs in Rack2.
3. Verify the cnBNG state.
   - Ensure that all cnBNGs are in good state.
   - Both GR instances in all cnBNGs in Rack2 should be in STANDBY state.
   - Both GR instances in all cnBNGs in Rack1 should be in PRIMARY state.
4. Verify that the scale-up configuration changes are applied (may take approximately 15 mins).

5. Check the output of **show idmgr-reconcile status** command to confirm a successful PodRestart status on the RackBeingScaled. If the status is `InProgress`, wait until it changes to `Success`.
6. Make Rack-2 active and Rack-1 standby
   a. Unset geo maintenance mode in all cnBNGs in both racks.
   b. Make both GR instances of all cnBNGs in Rack2 PRIMARY.
   c. Make both GR instances of all cnBNGs in Rack1 STANDBY.
   d. Check the output of **show idmgr-reconcile status** command to confirm a successful RoleSwitchover status on the RackInService. If the status is `InProgress`, wait until it changes to `Success`.
   e. Set Geo maintenance mode in all cnBNGs in both racks.
7. Add configuration changes in Rack-1 (STANDBY)
   a. Shutdown all cnBNGs in Rack1.
   b. Add the configuration changes to all cnBNGs in Rack1 (see *"Service Scale up Configuration"* section).
   c. Start up all cnBNGs in Rack1.
8. Verify the cnBNG State.
   - Ensure that all cnBNGs are in good state.
   - Both GR instances in all cnBNGs in Rack1 should be in STANDBY state.
   - Both GR instances in all cnBNGs in Rack2 should be in PRIMARY state.
9. Verify that the scale-up configuration changes are applied (may take ~15 mins).
10. Check the output of **show idmgr-reconcile status** command to confirm a successful PodRestart status on the RackBeingScaled. If the status is `InProgress`, wait until it changes to `Success`.
11. Remove the geo maintenance mode in both clusters.
12. Move both CP-GR clusters to their original state.

Configuration Examples

K8s Cluster Scale Up:

Scale up configurations for both the servers are available in the following combinations:

- Node Scaling for Service only
- Node Scaling for CDL only
- Node Scaling for both CDL and Service

Node Scaling for Service Only

Cluster1 Node5 Deployer Configuration	Cluster2 Node5 Deployer Configuration

clusters svi-cnbng-gr-tb1
 nodes server-5
  host-profile bng-ht-sysctl-enable
  k8s node-type worker
  k8s ssh-ip 1.1.111.15
  k8s ssh-ipv6 2002:4888:1:1::111:15
  k8s node-ip 1.1.111.15
  k8s node-ipv6 2002:4888:1:1::111:15
  k8s node-labels smi.cisco.com/svc-type
service
  exit
  ucs-server cimc ip-address 10.81.103.78
  initial-boot netplan vlans bd0.mgmt.3103
   addresses [10.81.103.119/24]
   gateway4 10.81.103.1
  exit
  initial-boot netplan vlans bd1.k8s.111
   addresses [1.1.111.15/24
2002:4888:1:1::111:15/112]
   routes 203.203.203.50/32 1.1.111.1
   exit
   routes 2002:4888:203:203::203:50/128
2002:4888:1:1::111:1
   exit
  exit
  initial-boot netplan vlans bd1.inttcp.104
   dhcp4   false
   dhcp6   false
   addresses [1.1.104.15/24
2002:4888:1:1::104:15/112]
   id   104
   link   bd1
   routes 2.2.104.0/24 1.1.104.1
   exit
   routes 2002:4888:2:2::104:0/112
2002:4888:1:1::104:1
   exit
  exit
  os tuned enabled
 exit
exit
	clusters svi-cnbng-gr-tb2
	 nodes server-5
	  host-profile bng-ht-sysctl-enable
	  k8s node-type worker
	  k8s ssh-ip 2.2.112.15
	  k8s ssh-ipv6 2002:4888:2:2::112:15
	  k8s node-ip 2.2.112.15
	  k8s node-ipv6 2002:4888:2:2::112:15
	  k8s node-labels smi.cisco.com/svc-type
	service
	  exit
	  ucs-server cimc ip-address 10.81.103.58
	  initial-boot netplan vlans bd0.mgmt.3103
	   addresses [10.81.103.68/24]
	   gateway4 10.81.103.1
	  exit
	  initial-boot netplan vlans bd1.inttcp.104
	   dhcp4 false
	   dhcp6 false
	   addresses [2.2.104.15/24
	2002:4888:2:2::104:15/112]
	   id 104
	   link bd1
	   routes 1.1.104.0/24 2.2.104.1
	   exit
	   routes 2002:4888:1:1::104:0/112
	2002:4888:2:2::104:1
	   exit
	  exit
	  initial-boot netplan vlans bd1.k8s.112
	   addresses [2.2.112.15/24
	2002:4888:2:2::112:15/112]
	   routes 203.203.203.50/32 2.2.112.1
	   exit
	   routes 2002:4888:203:203::203:50/112
	2002:4888:2:2::112:1
	   exit
	  exit
	  os tuned enabled
	 exit
	exit

Node Scaling for CDL Only

Cluster1 Node5 Deployer Configuration	Cluster2 Node5 Deployer Configuration

clusters svi-cnbng-gr-tb1
 nodes server-5
  host-profile bng-ht-sysctl-enable
  k8s node-type worker
  k8s ssh-ip 1.1.111.15
  k8s ssh-ipv6 2002:4888:1:1::111:15
  k8s node-ip 1.1.111.15
  k8s node-ipv6 2002:4888:1:1::111:15
  k8s node-labels smi.cisco.com/sess-type cdl-node
  exit
  ucs-server cimc ip-address 10.81.103.78
  initial-boot netplan vlans bd0.mgmt.3103
  addresses [10.81.103.119/24]
  gateway4 10.81.103.1
  exit
  initial-boot netplan vlans bd1.cdl.103
  dhcp4 false
  dhcp6 false
  addresses [1.1.103.15/24 2002:4888:1:1::103:15/112]
  id 103
  link bd1
  routes 2.2.103.0/24 1.1.103.1
  exit
  routes 2002:4888:2:2::103:0/112 2002:4888:1:1::103:1
  exit
  exit
  initial-boot netplan vlans bd1.k8s.111
  addresses [1.1.111.15/24 2002:4888:1:1::111:15/112]
  routes 203.203.203.50/32 1.1.111.1
  exit
  routes 2002:4888:203:203::203:50/128 2002:4888:1:1::111:1
  exit
  exit
  os tuned enabled
 exit
exit
	clusters svi-cnbng-gr-tb2
	 nodes server-5
	  host-profile bng-ht-sysctl-enable
	  k8s node-type worker
	  k8s ssh-ip 2.2.112.15
	  k8s ssh-ipv6 2002:4888:2:2::112:15
	  k8s node-ip 2.2.112.15
	  k8s node-ipv6 2002:4888:2:2::112:15
	  k8s node-labels smi.cisco.com/sess-type cdl-node
	  exit
	  ucs-server cimc ip-address 10.81.103.58
	  initial-boot netplan vlans bd0.mgmt.3103
	  addresses [10.81.103.85/24]
	  gateway4 10.81.103.1
	  exit
	  initial-boot netplan vlans bd1.cdl.103
	  dhcp4 false
	  dhcp6 false
	  addresses [2.2.103.15/24 2002:4888:2:2::103:15/112]
	  id 103
	  link bd1
	  routes 1.1.103.0/24 2.2.103.1
	  exit
	  routes 2002:4888:1:1::103:0/112 2002:4888:2:2::103:1
	  exit
	  exit
	  initial-boot netplan vlans bd1.k8s.112
	  addresses [2.2.112.15/24 2002:4888:2:2::112:15/112]
	  routes 203.203.203.50/32 2.2.112.1
	  exit
	  routes 2002:4888:203:203::203:50/112 2002:4888:2:2::112:1
	  exit
	  exit
	  os tuned enabled
	 exit
	exit

Node Scaling for both CDL and Services

Cluster1 Node5 Deployer configuration	Cluster2 Node5 Deployer configuration

```
clusters svi-cnbng-gr-tb1
 nodes server-5
  host-profile bng-ht-enable
  k8s node-type worker
  k8s ssh-ip 1.1.111.15
  k8s ssh-ipv6 2002:4888:1:1::111:15
  k8s node-ip 1.1.111.15
  k8s node-ipv6 2002:4888:1:1::111:15
  k8s node-labels smi.cisco.com/sess-type
cdl-node
  exit
  k8s node-labels smi.cisco.com/svc-type
service
  exit
  ucs-server cimc ip-address 10.81.103.78
  initial-boot netplan vlans bd0.mgmt.3103
   addresses [ 10.81.103.119/24 ]
   gateway4  10.81.103.1
  exit
  initial-boot netplan vlans bd1.inttcp.104
   dhcp4     false
   dhcp6     false
   addresses [ 1.1.104.15/24
2002:4888:1:1::104:15/112 ]
   id        104
   link      bd1
   routes 2.2.104.0/24 1.1.104.1
   exit
   routes 2002:4888:2:2::104:0/112
2002:4888:1:1::104:1
   exit
  exit
  initial-boot netplan vlans bd1.cdl.103
   dhcp4     false
   dhcp6     false
   addresses [ 1.1.103.15/24
2002:4888:1:1::103:15/112 ]
   id        103
   link      bd1
   routes 2.2.103.0/24 1.1.103.1
   exit
   routes 2002:4888:2:2::103:0/112
2002:4888:1:1::103:1
   exit
  exit
  initial-boot netplan vlans bd1.k8s.111
   addresses [ 1.1.111.15/24
2002:4888:1:1::111:15/112 ]
   routes 203.203.203.50/32 1.1.111.1
   exit
   routes 2002:4888:203:203::203:50/128
2002:4888:1:1::111:1
   exit
  exit
  os tuned enabled
 exit
exit
```

```
clusters svi-cnbng-gr-tb2
 nodes server-5
  host-profile bng-ht-enable
  k8s node-type worker
  k8s ssh-ip 2.2.112.15
  k8s ssh-ipv6 2002:4888:2:2::112:15
  k8s node-ip 2.2.112.15
  k8s node-ipv6 2002:4888:2:2::112:15
  k8s node-labels smi.cisco.com/sess-type
cdl-node
  exit
  k8s node-labels smi.cisco.com/svc-type
service
  exit
  ucs-server cimc ip-address 10.81.103.58
  initial-boot netplan vlans bd0.mgmt.3103
   addresses [ 10.81.103.85/24 ]
   gateway4  10.81.103.1
  exit
  initial-boot netplan vlans bd1.inttcp.104
   dhcp4     false
   dhcp6     false
   addresses [ 2.2.104.15/24
2002:4888:2:2::104:15/112 ]
   id        104
   link      bd1
   routes 1.1.104.0/24 2.2.104.1
   exit
   routes 2002:4888:1:1::104:0/112
2002:4888:2:2::104:1
   exit
  exit
  initial-boot netplan vlans bd1.cdl.103
   dhcp4     false
   dhcp6     false
   addresses [ 2.2.103.15/24
2002:4888:2:2::103:15/112 ]
   id        103
   link      bd1
   routes 1.1.103.0/24 2.2.103.1
   exit
   routes 2002:4888:1:1::103:0/112
2002:4888:2:2::103:1
   exit
  exit
  initial-boot netplan vlans bd1.k8s.112
   addresses [ 2.2.112.15/24
2002:4888:2:2::112:15/112 ]
   routes 203.203.203.50/32 2.2.112.1
   exit
   routes 2002:4888:203:203::203:50/112
2002:4888:2:2::112:1
   exit
  exit
  os tuned enabled
 exit
exit
```

CDL Configuration Changes

CDL Scale-up Configuration on Cluster1	CDL Scale-up Configuration on Cluster2

CDL Scale-up Configuration on Cluster1	CDL Scale-up Configuration on Cluster2
cdl datastore session label-config session geo-remote-site [2] **mode scale-up** slice-names [1 2] overload-protection disable true endpoint go-max-procs 16 **endpoint replica 3** endpoint copies-per-node 2 endpoint settings slot-timeout-ms 750 endpoint external-ip 1.1.103.100 endpoint external-ipv6 2002:4888:1:1::103:100 endpoint external-port 8882 index go-max-procs 8 index replica 2 **index prev-map-count 2** **index map 3** features instance-aware-notification enable true features instance-aware-notification system-id 1 slice-names [1] exit features instance-aware-notification system-id 2 slice-names [2] exit slot go-max-procs 8 slot replica 2 **slot map 6** slot notification limit 1500 slot notification max-concurrent-bulk-notifications 20 exit cdl label-config session endpoint key smi.cisco.com/sess-type endpoint value cdl-node slot map 1 key smi.cisco.com/sess-type value cdl-node exit slot map 2 key smi.cisco.com/sess-type value cdl-node exit slot map 3 key smi.cisco.com/sess-type value cdl-node exit slot map 4 key smi.cisco.com/sess-type value cdl-node exit **slot map 5** **key smi.cisco.com/sess-type** **value cdl-node** **exit** **slot map 6** **key smi.cisco.com/sess-type**	cdl datastore session label-config session geo-remote-site [1] **mode scale-up** slice-names [1 2] overload-protection disable true endpoint go-max-procs 16 **endpoint replica 3** endpoint copies-per-node 2 endpoint settings slot-timeout-ms 750 endpoint external-ip 2.2.103.100 endpoint external-ipv6 2002:4888:2:2::103:100 endpoint external-port 8882 index go-max-procs 8 index replica 2 **index prev-map-count 2** **index map 3** features instance-aware-notification enable true features instance-aware-notification system-id 1 slice-names [1] exit features instance-aware-notification system-id 2 slice-names [2] exit slot go-max-procs 8 slot replica 2 **slot map 6** slot notification limit 1500 slot notification max-concurrent-bulk-notifications 20 exit cdl label-config session endpoint key smi.cisco.com/sess-type endpoint value cdl-node slot map 1 key smi.cisco.com/sess-type value cdl-node exit slot map 2 key smi.cisco.com/sess-type value cdl-node exit slot map 3 key smi.cisco.com/sess-type value cdl-node exit slot map 4 key smi.cisco.com/sess-type value cdl-node exit **slot map 5** **key smi.cisco.com/sess-type** **value cdl-node** **exit** **slot map 6** **key smi.cisco.com/sess-type**

CDL Scale-up Configuration on Cluster1	CDL Scale-up Configuration on Cluster2
value cdl-node **exit** index map 1 key smi.cisco.com/sess-type value cdl-node exit index map 2 key smi.cisco.com/sess-type value cdl-node exit **index map 3** **key smi.cisco.com/sess-type** **value cdl-node** **exit** **exit**	**value cdl-node** **exit** index map 1 key smi.cisco.com/sess-type value cdl-node exit index map 2 key smi.cisco.com/sess-type value cdl-node exit **index map 3** **key smi.cisco.com/sess-type** **value cdl-node** **exit** exit

The commands that are required for node scaling are highlighted.

Service Scale up Configuration

Scaling up services by increasing replica counts and nodes is supported. Reducing replicas and nodes is not supported and it will impact the system.

Existing Configuration	Service Scale Up Configuration in Both Clusters
instance instance-id 1 endpoint dhcp replicas 4 nodes 2 exit exit endpoint sm replicas 6 nodes 2 exit exit instance instance-id 2 endpoint dhcp replicas 4 nodes 2 exit exit endpoint sm replicas 6 nodes 2 exit exit	instance instance-id 1 endpoint dhcp replicas 4 **nodes 3** exit exit endpoint sm replicas 6 **nodes 3** exit exit instance instance-id 2 endpoint dhcp replicas 4 **nodes 3** exit exit endpoint sm replicas 6 **nodes 3** exit exit

cnBNG Health-Check pods

cnBNG Health-Check pod is a health validation feature that

- performs deterministic health checks for cluster resources, configuration, workload states, and connectivity
- aggregates results from built-in validators to provide a system health verdict with drill-down detail, and

- offers centralized, operator-friendly reporting surfaces for health status and historical trends.

The Health-Check pod operates in three modes: startup (initial readiness), periodic (ongoing health monitoring), and on-demand (manual or API-triggered validation). It is deployed as a DaemonSet with a single leader pod orchestrating health checks. High availability is achieved by leader election, allowing other pods to assume leadership if needed. Platform support targets Kubernetes-based Linux environments, and the pod requires appropriate network and Role-based access control (RBAC) access.

Table 3: Feature History Table

Feature Name	Release Information	Feature Description
cnBNG Health-Check pods	Release 2026.03.0	You can now enhance operational reliability with centralized, automated health checks for your cnBNG Control Plane clusters. The Health-Check pod continuously or on demand validates configuration, workloads, network, and connectivity, presenting actionable status via CLI, Web UI, REST API, and Prometheus metrics to reduce manual inspection and operational risk.

Benefits

cnBNG Health-Check pods provide these benefits:

- **Deterministic validation**: Replaces ad-hoc scripts with repeatable, code-defined checks, ensuring consistent pass/fail criteria across all cluster deployments.
- **Centralized visibility**: Provides a single source of truth for operators, TAC engineers, and observability tools, reducing the time required for troubleshooting and hand-offs.
- **Flexible execution modes**: Supports automated startup warm-up for self-healing during bring-up, periodic monitoring for steady-state operations, and on-demand manual triggers for immediate verification.
- **Actionable insights**: Delivers detailed reports with clear remediation hints, allowing operators to identify and resolve misconfigurations or stalled workloads quickly.

How cnBNG Health-Check pod works

Summary

The key components involved in the cnBNG Health-Check process are:

- DaemonSet Pods: Each node runs a pod, with one elected leader to coordinate validation operations.
- Leader Election Mechanism: Maintains high availability by ensuring there is always an active validation authority.
- Validator Engine: Executes category-based health checks in parallel.
- Aggregator: Collects results and computes the overall cluster health verdict.

- Operator Interfaces: CLI, REST API, Web UI, and Prometheus endpoints deliver health status and allow for detailed analysis and resolution steps.

This process automates repeatable health checks during startup, on a scheduled basis, or when initiated by an operator, ensuring reliable cluster operations and faster troubleshooting.

Workflow

These stages describe how cnBNG Health-Check pod performs cluster health validation and publishes results to operators.

1. Deployment and leader election: The DaemonSet creates one Health-Check Pod per node, and a leader is elected to coordinate validation activities.
   - Pods are deployed across all cluster nodes via DaemonSet.
   - Pods participate in leader election to determine the validation leader.

   If the leader becomes unavailable, another pod automatically takes over to ensure continuous validation.
2. Trigger selection: The leader pod determines when to launch a health check cycle according to execution mode settings.
   - Health checks can be triggered at startup, on a schedule, or by operator requests through CLI, REST API, or Web UI.
3. Validation sequencing: The leader runs validators in a defined order (netplan, config, workload, connectivity) to respect prerequisite dependencies.
   - Validators run in parallel, while the individual checks within each validator are performed sequentially.
4. Parallel validator execution: Inside each category, enabled validators run concurrently to evaluate resource states.
   - Examples include netplan-validation, configdeps-validation, workload-validation, and udpproxy-validation to inspect configuration, Control Plane objects, endpoints, and connectivity.

   The validator engine allows enabling or disabling specific validators per category for flexibility.
5. Issue detection and status mapping: Validators report issues if checks fail and normalize results to standard status codes.
   - Status codes include: PENDING, RUNNING, SUCCESS, WARNING, ERROR, SKIPPED, and DISABLED.
6. Aggregation and verdict calculation: The aggregator consolidates validator statuses to determine the overall cluster health verdict.
   - Verdicts are HEALTHY (all required checks succeed), DEGRADED (warnings or optional failures), or CRITICAL (mandatory checks fail).

   The verdict communicates the cluster's health and whether urgent remediation is required.
7. Status publication and operator access: The leader pod publishes health results to operator interfaces.
   - Status appears in the show system-health-check CLI, Web UI dashboard, REST API endpoints (such as /api/v1/status and /api/v1/reports), and Prometheus metrics (/metrics).

Operators can examine results, resolve issues, or request new validation cycles as needed.

Health-Check validators

This table details the available health-check validators, their categories, and the specific aspects each validator evaluates:

Validator	Category	Description
configdeps	config	Validates mandatory, conditional, and optional BNG ConfigMaps plus cross-ConfigMap consistency and reference checks.
netplan	netplan	Inspects per-node `/etc/netplan` configuration for YAML, interface, route, DNS, and hardware-related abnormalities via leader fan-out.
workload	workload	Sweeps Pods and Kubernetes workloads across Deployment, StatefulSet, DaemonSet, ReplicaSet, and Pod to assess health and rollout state.
podreplicas	workload	Enforces named-component HA and replica-count rules, with single-node hostNetwork-aware handling to avoid misleading warnings.
udpproxy	connectivity	Verifies UDP-proxy connectivity and runtime coverage for UPF, PFCP, ping, RADIUS, VIP listener, route-interface, and N4 peer checks.
cdl	connectivity	Verifies CDL local-VIP binding, Kafka and mirror-maker probe origin, CPGR consistency, slice configuration, and remote CDL endpoint checks.
georep	connectivity	Verifies geo-replication pod health, local VIP binding, internal and remote VIP reachability, OAM role, and replication status.
bgpbfd	connectivity	Validates BGP/BFD-related configuration and runtime reachability including neighbor probes, session state, bond-member presence, and policy interface or gateway checks.
rpcstatus	workload	Parses `show rpc all` output from oam-pod and flags any RPC whose status is not `Started`.
endpointstatus	workload	Parses `show endpoint all` output from oam-pod and flags any endpoint whose status is not `Started`.
diagnostics	workload	Parses `show_diag` output from oam-pod and flags any diagnostic whose status is not `Success`.

Restrictions for the cnBNG Health-Check pod

These operational restrictions apply to the cnBNG Health-Check pod:

- OpenShift is not supported.
- Only supported on Kubernetes-based Linux clusters for cnBNG control-plane deployments.
- Netplan and VIP binding validation are available only on Linux clusters. These checks are skipped on unsupported operating systems.
- Multi-cluster and cross-cluster verdict aggregation are not supported. Validation occurs independently on each cluster.

- Clusters must expose the Kubernetes API and endpoints with appropriate network access and permissions.

Configure the cnBNG Health-Check pod

Configure the cnBNG Health-Check pod to automate and centralize validation of the cnBNG Control Plane cluster health, readiness, and connectivity. The Health-Check pod provides consistent platform validation by executing configured checks for configuration data, system workload status, and network connectivity.

Before you begin

Ensure these prerequisites are met before configuring the cnBNG Health-Check pod:

- The deployment uses a supported Kubernetes-based Linux cluster for cnBNG Control Plane.
- Network connectivity between Control Plane nodes and the Kubernetes API as well as oam-pod endpoints is verified.
- Your user account has sufficient RBAC privileges to apply configuration changes and access health-check status.

Follow these steps to configure the cnBNG Health-Check pod.

Procedure

Step 1 Enable the Health-Check pod from configuration mode.

Example:

```
configure
 k8 system-health-check enable
 exit
```

Enabling the pod starts deployment of the DaemonSet, which schedules pods across all nodes in the cluster and elects a leader to perform health validation.

- Enabling does not run validation until at least one execution trigger is configured (startup or periodic).

Step 2 Configure the execution mode (startup or periodic) and set validation interval.

Example:

```
configure
  k8 system-health-check mode periodic 10
  exit
```

- Periodic mode triggers health-check cycles at the specified interval (minimum 5 minutes, maximum 10080 minutes).
- Startup mode performs checks during cluster bring-up until healthy.

Choose the mode that fits your operational monitoring requirements.

Step 3 Enable or disable validator categories as needed.

Example:

```
configure
  k8 system-health-check config-validation disable
  k8 system-health-check netplan-validation disable
  k8 system-health-check workload-validation disable
  k8 system-health-check connectivity-validation disable
  exit
```

Validator categories can be selectively enabled or disabled depending on operational needs. Omitting a category enables it by default.

- Disabling a category prevents its validators from running and reporting.

Step 4 Manually trigger health validation for all or a specific category.

Example:

```
bng# system-health-check start type all
```

To run a specific category, specify its type (for example, **system-health-check start type { all | config | connectivity | netplan | workload }**).

- Manual triggers provide immediate feedback outside scheduled cycles.

Step 5 View overall health-check status and validator results using CLI.

Example:

```
bng# show system-health-check
bng# show system-health-check detail
```

Use **show system-health-check** for a cluster summary and **show system-health-check detail** for per-validator status and drill-down.

- Review the verdict (HEALTHY, DEGRADED, CRITICAL) and any outstanding issues.

If...	Then...
The summary states HEALTHY and all validators show SUCCESS	Cluster health is validated and operational.
The summary states DEGRADED or CRITICAL	Investigate per-validator detail for warning or error conditions.
Some validators show WARNING or ERROR	Review recommended actions in the detail output and resolve issues as needed.

Step 6 (Optional) Access health-check results via Web UI, REST API, or Prometheus metrics endpoints .

- Example Web UI usage:

  The Web UI URL is provided in the output of the **show system-health-check** command.

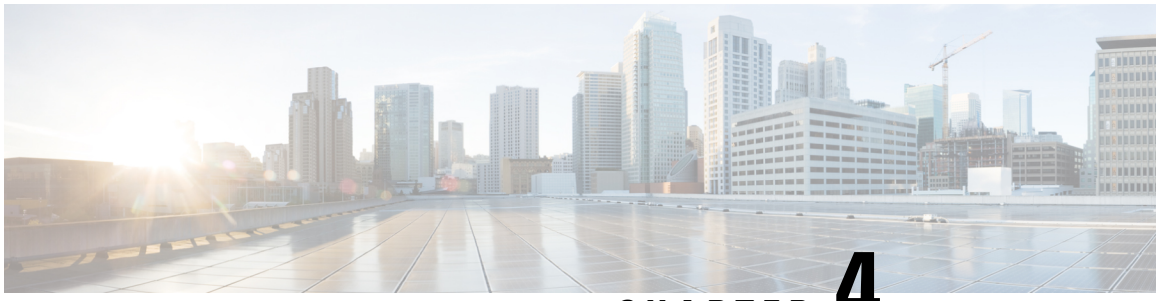
  Navigate to dashboard URL, authenticate, review cluster health and per-validator detail.

- Example REST API usage:

Access `/api/v1/status` or `/api/v1/reports` endpoints.

- Example Prometheus metrics:

  Query `bng_health_check_cluster_status` gauge for latest verdict.



CHAPTER 4

Pods and Services Reference

- Feature Summary and Revision History, on page 67
- Feature Description, on page 68
- Associating Pods to the Nodes, on page 74
- Configuring Pod Layout, on page 76
- Configuring Event Trace, on page 77

Feature Summary and Revision History

Summary Data

Table 4: Summary Data

Applicable Product(s) or Functional Area	cnBNG
Applicable Platform(s)	SMI
Feature Default Setting	Disabled - Configuration Required
Related Changes in this Release	Subscriber Manager, on page 437
Related Documentation	Not Applicable

Revision History

Table 5: Revision History

Revision Details	Release
Enhancement Introduced: cnBNG supports the pod layout configuration and event tracing configuration.	2022.03.0

Revision Details	Release
Enhancement Introduced: The Subscriber Manager pod supports the charging functionality, that is, subscriber access and subscriber management.	2021.03.0
First introduced.	2021.01.0

Feature Description

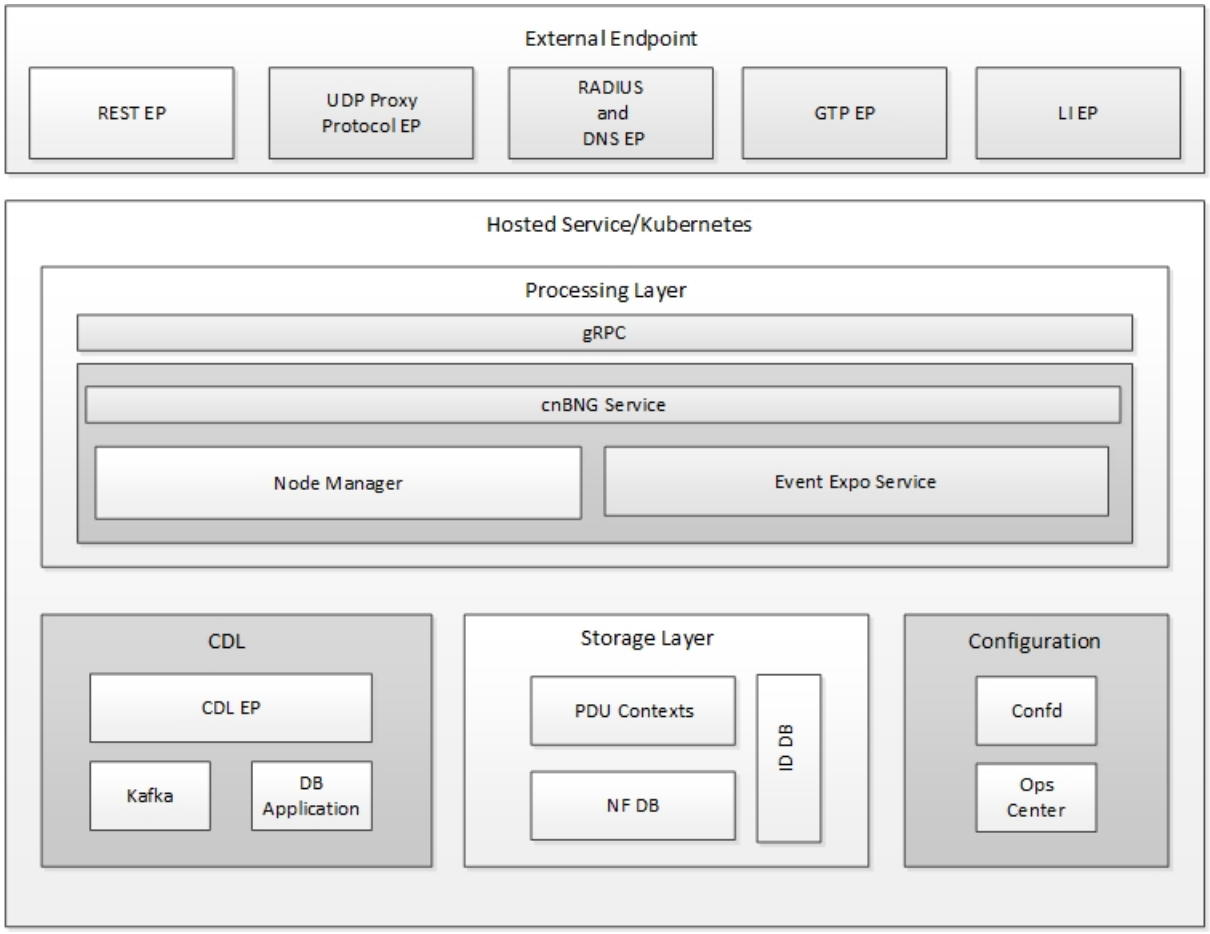
The SMI Ops Center is the platform to deploy cnBNG cluster with the offline or online repository. It is mandatory to deploy the SMI Ops Center to install the BNG Ops Center.

The cnBNG is built on the Kubernetes cluster strategy, which implies that it has adopted the native concepts of containerization, high availability, scalability, modularity, and ease of deployment. To achieve the benefits offered by Kubernetes, cnBNG uses the construct that includes the components such as pods and services.

Depending on the deployment environment, the cnBNG deploys the pods on the virtual machines that you have configured. Pods operate through the services that are responsible for the intra-pod communications. If the machine hosting the pods fail or experiences network disruption, the pods are terminated or deleted. However, this situation is transient and BNG spins new pods to replace the invalid pods.

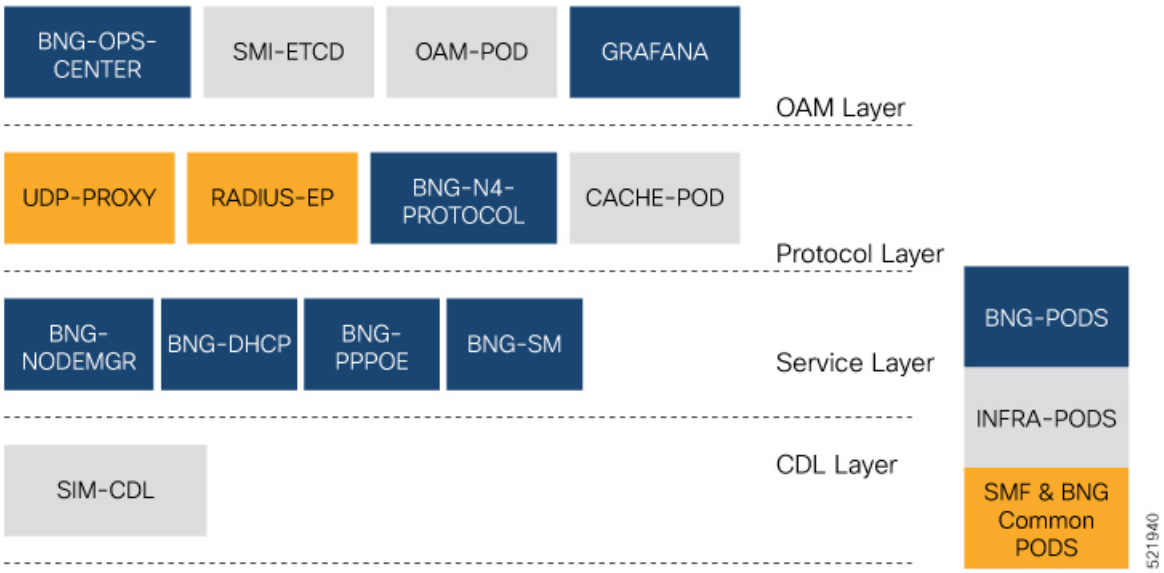
The following workflow provides a high-level visibility into the host machines, and the associated pods and services. It also represents how the pods interact with each other. The representation might defer based on your deployment infrastructure.

Figure 9: Communication Workflow of Pods



The following figure shows the cnBNG cluster pod layout.

Figure 10: cnBNG Cluster Pod Layout



Kubernetes deployment includes the **kubectl** command-line tool to manage the Kubernetes resources in the cluster. You can manage the pods, nodes, and services.

For generic information on the Kubernetes concepts, see the Kubernetes documentation.

The following sections provide more information on the Kubernetes components in cnBNG.

Pods

A pod is a process that runs on your Kubernetes cluster. It encapsulates a granular unit known as a container. A pod contains one or multiple containers.

Kubernetes deploys one or multiple pods on a single node which can be a physical or virtual machine. Each pod has a discrete identity with an internal IP address and port space. However, the containers within a pod can share the storage and network resources.

The following tables list the cnBNG and Common Execution Environment (CEE) pod names and the hosts on which they are deployed depending on the labels that you assign. For information on how to assign the labels, see Associating Pods to the Nodes, on page 74.

Table 6: cnBNG Pods

Pod Name	Description	Host Name
api-bng-bng-ops-center	Functions as the *confD* API pod for the BNG Ops Center.	OAM
bng-dhcp-n0	Operates as the DHCP server and handles all DHCP related control messages.	Service
bng-n4-protocol-n0	Operates as encoder and decoder of application protocols (PFCP, GTP, RADIUS, and so on) whose underlying transport protocol is UDP.	Protocol

Pod Name	Description	Host Name
bng-nodemgr-n0	Performs node level interactions Service such as N4 link establishment, management (heart-beat), and so on.	Service
bng-pppoe-n0	Runs the combined Control Plane (CP) for PPPoE and PPP.	Service
bng-sm-n0	Manages subscriber access and subscriber management functions such as Authentication, Authorization, Accounting, and policy management..	Service
cache-pod-0	Operates as the pod to cache any sort of system information that will be used by other pods as applicable.	Protocol
cdl-ep-session-c1-d0	Provides an interface to the CDL.	Session
cdl-index-session-c1-m1-0	Preserves the mapping of keys to the session pods.	Session
cdl-slot-session-c1-m1-0	Operates as the CDL Session pod Session to store the session data.	Session
documentation	Contains the documentation.	OAM
etcd-bng-bng-etcd-cluster-0	Hosts the etcd for the BNG application to store information such as pod instances, leader information, NF-UUID, endpoints, and so on.	OAM
grafana-dashboard-app-infra	Contains the default dashboard of app-infra metrics in Grafana.	OAM
grafana-dashboard-bng	Contains the default dashboard of the cnBNG-service metrics in Grafana.	OAM
grafana-dashboard-cdl	Contains the default dashboard of CDL metrics in Grafana.	OAM
kafka	Hosts theKafka details for the CDL replication.	Protocol
oam-pod	Operates as the pod to facilitate Ops Center actions like show commands, configuration commands, monitor protocol monitor subscriber, and so on.	OAM
ops-center-bng-bng-ops-center	Acts as the BNG Ops Center.	OAM
prometheus-rules-cdl	Contains the default alerting rules and recording rules for Prometheus CDL.	OAM
radius-ep-n0-0	Operates as RADIUS endpoint of cnBNG.	
smart-agent-bng-bng-ops-center	Operates as the utility pod for the BNG Ops Center.	OAM
bng-udp-proxy-0	Operates as proxy for all UDP messages. Owns UDP client and server functionalities.	Protocol
swift-bng-bng-ops-center	Operates as the utility pod for the BNG Ops Center.	OAM

Pod Name	Description	Host Name
zookeeper	Assists Kafka for topology management.	OAM

Table 7: CEE Pods

Pod Name	Description	Host Name
alert-logger	Stores the history of active and resolved alerts.	OAM
alertmanager	Duplicates alerts and sends out resolution of alerts when they are resolved in Prometheus.	OAM
api-cee-global-ops-center	Functions as the confD API pod for the CEE Ops Center.	OAM
bulk-stats	Assists to retrieve bulkstats saved by Prometheus containers.	OAM
cee-global-product-documentation	Contains the product documentation (API, CLI, and so on).	OAM
core-retriever	Assists in retrieving the core dumps.	All the nodes except ETCD nodes.
documentation	Contains the documentation (metrics and usage).	OAM
grafana-dashboard-metrics	Assists in collating Grafana metrics on the dashboard.	OAM
grafana	Contains the Grafana metrics for CEE.	OAM
kube-state-metrics	Assists in generating metrics about the state of Kubernetes objects: node status, node capacity (CPU and memory), and so on.	OAM
logs-retriever	Assists in retrieving Kernel, Kubelet, and Container level logs through output to JournalD driver.	All the nodes except ETCD nodes.
node-exporter	Exports the node metrics.	All the nodes.
ops-center-cee-global-ops-center	Provides NETCONF and CLI interface to the application.	OAM
path-provisioner	Provisions the local storage volume.	All the nodes except ETCD nodes.
pgpool	*Pgpool* is a middleware that works between *PostgreSQL* servers and a *PostgreSQL* database.	OAM
postgres	Storage of alerts and Grafana dashboards.	OAM

Pod Name	Description	Host Name
prometheus-hi-res	Stores all metrics and generates alerts by alerting rules.	OAM
prometheus-rules	Contains the default alerting rules and recording rules for Prometheus.	OAM
prometheus-scrapeconfigs-synch	Synchronizes the Prometheus scrape configuration.	OAM
pv-manager	Provisions the local storage volume.	OAM
pv-provisioner	Provisions the local storage volume.	OAM
show-tac-manager	Assists in creating and deleting debug package.	OAM
smart-agent-cee-global-ops-center	Operates as the utility pod for the CEE Ops Center.	OAM
snmp-trapper	Sends the SNMP traps based on triggered alerts.	OAM
swift-cee-global-ops-center	Operates as the utility pod for the CEE Ops Center.	OAM
thanos-query-hi-res	Implements the Thanos query for Prometheus HA.	OAM
fluentbit	Assists in log forwarding to the external logs collector.	All the nodes except ETCD nodes.

Services

The cnBNG configuration is composed of several microservices that run on a set of discrete pods. Microservices are deployed during the cnBNG deployment. The cnBNG uses these services to enable communication between the pods. When interacting with another pod, the service identifies the IP address of the pod to initiate the transaction and acts as an endpoint for the pod.

The following table describes the BNG services and the pod on which they run.

Table 8: BNG Services and Pods

Service Name	Pod Name	Description
bng-nodemgr	bng-nodemgr-n0	Responsible for node level interactions Serv such as N4 link establishment, management (heart-beat), and so on.
bng-dhcp	bng-dhcp-n0	Functions as the DHCP server and handles a DHCP related control messages.
bng-pppoe	bng-pppoe-n0	Functions as the combined Control Plane (C PPPoE and PPP.

Service Name	Pod Name	Description
bng-sm	bng-sm-n0	Manages subscriber access and subscriber management functions such as Authentication, Authorization, Accounting, and policy management..

Open Ports and Services

cnBNG uses different ports for communication purposes. The following table describes the default open ports and the associated services in an SMI based cnBNG system.

Application Infrastructure (App-infra)

Port	Service
8850	Golang net/HTTP server TCP Golang net/HTTP server
8879	Golang net/HTTP server TCP Golang net/HTTP server
8850	DefaultPProfPort
8879	DefaultAdminEndPointPort

UDP

Port	Service	CP to UP Interfaces
2152	GTPU	CPRi
8805	PFCP	SCi

Associating Pods to the Nodes

This section describes how to associate a pod to the node based on their labels.

After you have configured a cluster, you can associate pods to the nodes through labels. This association enables the pods to get deployed on the appropriate node based on the key-value pair.

Labels are required for the pods to identify the nodes where they must get deployed and to run the services. For example, when you configure the protocol-layer label with the required key-value pair, the pods are deployed on the nodes that match the key-value pair.

To associate pods to the nodes through the labels, use the following configuration:

1. To associate pods to the nodes through the labels, use the following configuration:

```
config
k8 label protocol-layer key key_value vm-type value protocol
exit
k8 label service-layer key key_value vm-type value service
exit
```

```
k8 label cdl-layer key key_value vm-type value cdl
exit
k8 label oam-layer key key_value vm-type value oam
exit
```

NOTES:

- If you opt not to configure the labels, then BNG assumes the labels with the default key-value pair.
  - **k8 label protocol-layer key** *key_value* **vm-type** *value* **protocol**: Configures the key value pair for protocol layer.
  - **k8 label service-layer key** *key_value* **vm-type** *value* **service**: Configures the key value pair for the service layer.
  - **k8 label cdl-layer key** *key_value* **vm-type** *value* **cdl**: Configures the key value pair for CDL.
  - **k8 label oam-layer key** *key_value* **vm-type** *value* **oam**: Configures the key value pair for OAM layer.

Viewing the Pod Details and Status

If the service requires additional pods, BNG creates and deploys the pods. You can view the list of pods that are participating in your deployment through the BNG Ops Center.

You can run the **kubectl** command from the master node to manage the Kubernetes resources.

1. To view the comprehensive pod details, use the following command.

   ```
   kubectl get pods -n bng_namespace pod_name -o yaml
   ```

   The pod details are available in YAML format. The output of this command results in the following information:

   - The IP address of the host where the pod is deployed.
   - The service and application that is running on the pod.
   - The ID and name of the container within the pod.
   - The IP address of the pod.
   - The current state and phase in which the pod is.
   - The start time from which pod is in the current state.

2. Use the following command to view the summary of the pod details.

   ```
   kubectl get pods -n bng_namespace -o wide
   ```

States

Understanding the pod's state lets you determine the current health and prevent the potential risks. The following table describes the pod's states.

Table 9: Pod States

State	Description
Running	The pod is healthy and deployed on a node. It contains one or more containers.
Pending	The application is in the process of creating the container images for the pod.
Succeeded	Indicates that all the containers in the pod are successfully terminated. These pods cannot be restarted.
Failed	One ore more containers in the pod have failed the termination process. The failure occurred as the container either exited with non zero status or the system terminated the container.
Unknown	The state of the pod could not be determined. Typically, this could be observed because the node where the pod resides was not reachable.

Configuring Pod Layout

To configure the cnBNG pod layout when the virtual machine is short of CPU and memory resources, use the following sample configuration:

```
config
   instance instance-id instance_id
     endpoint sm
        cpu { max-process max_os_threads | request cpu_resource_request }
        memory { limit max_memory_resource | request memory_resource_request }
        end
```

NOTES:

- **cpu { max-process** *max_os_threads* **| request** *cpu_resource_request* **}**: Enable the K8s pod CPU configuration.
  - **max-process** *max_os_threads*: Specify the maximum number of parallel OS threads to use. *max_os_threads* must be an integer in the range of 1 to 32.
  - **request** *cpu_resource_request*: Specify the CPU resource request in millicores. *cpu_resource_request* must be an integer in the range of 100 to 1000000.
- **memory { limit** *max_memory_resource* **| request** *memory_resource_request* **}**: Enable the K8s pod memory configuration.
  - **limit** *max_memory_resource*: Specify the maximum number of used memory resources in megabytes. *max_memory_resource* must be an integer in the range of 100 to 200000.
  - **request** *memory_resource_request*: Specify the memory resource request in megabytes. *memory_resource_request* must be an integer in the range of 100 to 200000.

Configuring Event Trace

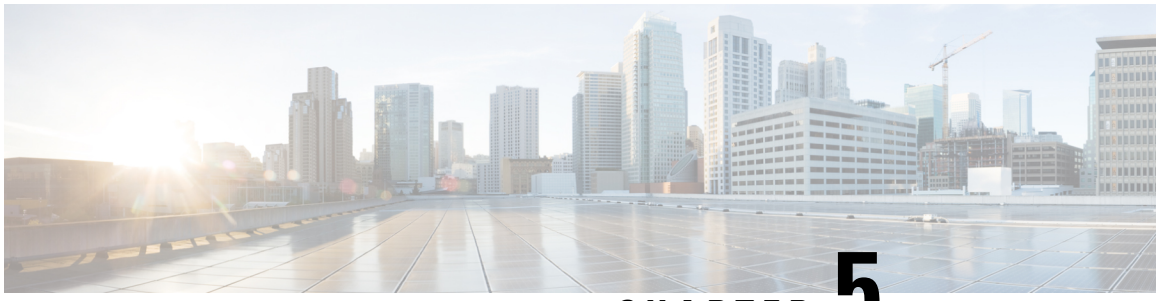
To configure event tracing, use the following sample configuration:

```
config
  subscriber [ event-trace-disable | event-trace-max-count event_trace_count
 ]
  end
```

NOTES:

- **event-trace-disable**: Disable subscriber event tracing. cnBNG uses event traces for session level event history in CDL records.
- **event-trace-max-count** *event_trace*: Specify the number of entries in event tracing.

  event_trace must be an integer in the range of 1 to 8192. Default value: 100.



CHAPTER 5

NSO Subscriber Microservices Infrastructure Core Function Pack

- Feature Summary and Revision History, on page 79
- Feature Description, on page 80
- Benefits of NSO SMI CFP, on page 80
- Supported Scenarios, on page 80
- Prerequisites for NSO SMI CFP, on page 81
- Initial Configuration, on page 81
- Deployment Services, on page 83
- Functions Services, on page 92
- Applications Services, on page 96
- Disable the auto-sync Feature, on page 97
- Trigger the Sync Action , on page 99
- Delete the SMI Deployment, on page 100

Feature Summary and Revision History

Summary Data

Table 10: Summary Data

Applicable Product(s) or Functional Area	cnBNG
Applicable Platform(s)	SMI
Feature Default Setting	Disabled - Configuration Required
Related Changes in this Release	Not Applicable
Related Documentation	Not Applicable

Revision History

Table 11: Revision History

Revision Details	Release
First introduced.	2024.04.0

Feature Description

The NSO Subscriber Microservices Infrastructure Core Function Pack (NSO SMI CFP) leverages the Network Service Orchestrator (NSO) to automate the deployment and configuration of functions by integrating with the Subscriber Microservices Infrastructure (SMI). This feature is essential for efficiently managing and orchestrating microservices-based applications, particularly in cloud-native environments.

NSO SMI CFP uses NSO to streamline the deployment of clusters and functions via the SMI Cluster Manager. This core function pack automates the setup of Kubernetes (K8s) clusters, enabling the deployment of Containerized Network Functions (CNFs) and other necessary microservices. It is designed to operate on Cisco UCS® bare metal infrastructure and integrates with the Cisco Container Platform for cloud management.

For detailed information about NSO, see NSO User Guide.

Key Components:

- **Cluster Manager (Cisco Unified Communications Manager)**: Manages the creation and deployment of K8s clusters.
- **Ops-Center**: Provides operational support.
- **Common Execution Environment (CEE)**: Standardized environment for running microservices.
- **Common Data Layer (CDL)**: Centralized data management layer.

Benefits of NSO SMI CFP

These are the benefits of using NSO SMI CFP:

- **Automated Deployment**: Simplifies the setup and configuration of clusters and microservices.
- **Scalability**: Supports deployment across multiple clusters, tailored to customer requirements.
- **Efficiency**: Reduces delays and rework during configuration, ensuring faster time-to-market.
- **Integrated Management**: Provides a unified interface for managing both infrastructure and applications.

Supported Scenarios

NSO SMI CFP is applicable in various scenarios, particularly where there is a need for automated and scalable deployment of microservices in cloud-native environments. This includes:

- **Enterprise Data Centers**: For large-scale deployment and management of microservices.
- **Service Providers**: To dynamically adopt orchestration solutions with changes in service portfolio.
- **Cloud Environments**: Leveraging Kubernetes clusters to manage containerized network functions.

Prerequisites for NSO SMI CFP

These are the prerequisites for using NSO SMI CFP:

1. **NSO Minimum Version**: 6.1.11
2. **Python Version**: Python 3.8 or above
3. **SMI Inception/Deployer Minimum Version**: 2024.04.1
4. **Dependent NEDs**:
   - ncs-6.1.11.2-cisco-smi-nc-2024.04.1
   - ncs-6.1.11.2-cisco-cee-nc-2024.04.1
   - ncs-6.1.11.2-cisco-bng-nc-1.1

Initial Configuration

To begin using NSO SMI CFP, you must add the SMI Cluster Manager as a device in the NSO device tree. This allows NSO to instruct the SMI Cluster Manager to create a cluster and then onboard the cluster into the NSO device tree.

Add the SMI Cluster Manager as a device to NSO

Procedure

Step 1 Log in to NSO.

Example:

```
<nacm xmlns="urn:ietf:params:xml:ns:yang:ietf-netconf-acm">
    <exec-default>permit</exec-default>
    <groups>
        <group>
            <name>ncsadmin</name>
            <user-name>admin</user-name>
        </group>
    </groups>
</nacm>
<smi xmlns="http://cisco.com/ns/nso/cfp/cisco-smi">
    <settings>
        <deployment xmlns="http://cisco.com/ns/nso/cfp/cisco-smi-deployment">
            <local-user>admin</local-user>
        </deployment>
```

```
    </settings>
</smi>
```

Step 2 Configure the global settings on NSO. Set the **<out-of-sync-commit-behaviour>** parameter to accept so that NSO does not track any transactions on the device from multiple users.

Example:

```
<config xmlns="http://tail-f.com/ns/config/1.0">
    <devices xmlns="http://tail-f.com/ns/ncs">
        <global-settings>
            <trace>pretty</trace>
            <out-of-sync-commit-behaviour>accept</out-of-sync-commit-behaviour>
            <trace-dir>/var/log/ncs</trace-dir>
        </global-settings>
    </devices>
</config>
</config>
```

Step 3 Configure the device authgroup.

Example:

```
<config xmlns="http://tail-f.com/ns/config/1.0">
    <devices xmlns="http://tail-f.com/ns/ncs">
        <authgroups>
            <group>
                <name>smi-auth</name>
                <default-map>
                    <remote-name>admin</remote-name>
                    <remote-password>xyz</remote-password>
                </default-map>
            </group>
        </authgroups>
    </devices>
</config>
```

Step 4 Add the SMI Cluster Manager to the NSO device tree.

Example:

```
<config xmlns="http://tail-f.com/ns/config/1.0">
    <devices xmlns="http://tail-f.com/ns/ncs">
        <device>
            <name>smi</name>
            <address>5.5.5.5</address>
            <port>830</port>
            <authgroup>smi-auth</authgroup>
            <device-type>
                <netconf>
                    <ned-id xmlns:cisco-smi-nc-1.1="http://tail-f.com/ns/ned-id/cisco-smi-nc-1.1">
                        cisco-smi-nc-1.1:cisco-smi-nc-1.1
                    </ned-id>
                </netconf>
            </device-type>
            <trace>pretty</trace>
            <state>
                <admin-state>unlocked</admin-state>
            </state>
        </device>
    </devices>
</config>
```

Step 5 Perform SMI Device Connect and sync.

Example:

```
admin@ncs# devices fetch-ssh-host-keys
fetch-result {
device smi
result updated
fingerprint {
algorithm ssh-rsa
value e6:9d:6d:f8:bc:50:46:9a:00:c0:23:1e:bd:5e:c4:9a
}
admin@ncs# devices device smi connect
result true
admin@ncs# devices device smi sync-from
result true
```

Step 6 Set up notification subscriptions.

Example:

```
<config xmlns="http://tail-f.com/ns/config/1.0">
    <devices xmlns="http://tail-f.com/ns/ncs">
        <device>
            <name>cm-1</name>
            <netconf-notifications>
                <subscription>
                    <name>download-status</name>
                    <stream>download-status</stream>
                    <local-user>admin</local-user>
                </subscription>
                <subscription>
                    <name>node-state</name>
                    <stream>node-state</stream>
                    <local-user>admin</local-user>
                </subscription>
                <subscription>
                    <name>node-status</name>
                    <stream>node-status</stream>
                    <local-user>admin</local-user>
                </subscription>
                <subscription>
                    <name>sync-state</name>
                    <stream>sync-state</stream>
                    <local-user>admin</local-user>
                </subscription>
                <subscription>
                    <name>sync-status</name>
                    <stream>sync-status</stream>
                    <local-user>admin</local-user>
                </subscription>
            </netconf-notifications>
        </device>
    </devices>
</config>
```

Deployment Services

Deployment services help you effectively deploy CNDP clusters by reducing delays, rework, and other problems during configuration. These services use the SMI Cluster Manager to set up the infrastructure on the nodes defined in the cluster.

Deploying Clusters

Clusters are the resources that applications need, and they include the worker nodes that run the applications. You can deploy multiple clusters based on your requirements.

NSO communicates with the SMI deployer, instructing it to create a cluster and then onboard the cluster into the NSO device tree.

Deploy the Kubernetes Cluster

The Kubernetes (K8s) cluster is used to install a new Cluster Manager or to install CNF functions services. You can deploy a cluster, such as a control-plane node, using the SMI Cluster Manager.



Note The valid K8s cluster configurations are either an All-in-One configuration (a single control-plane node) or a cluster with three control-plane nodes, with or without worker nodes (zero to N worker nodes).

Procedure

Step 1 The following is a sample payload to deploy a K8s cluster with a single control-plane node.

Example:

```
<config xmlns="http://tail-f.com/ns/config/1.0">
  <smi xmlns="http://cisco.com/ns/nso/cfp/cisco-smi">
    <deployment xmlns="http://cisco.com/ns/nso/cfp/cisco-smi-deployment">
      <name>bng-cfp-deployment</name>
      <local-user>admin</local-user>
      <cluster-manager>
        <manager-device>smi-cm-109</manager-device>
      </cluster-manager>
      <clusters>
        <cluster>
          <name>bng-cfp-cluster-53</name>
          <type>k8s</type>
          <environment>ucs-server</environment>
          <ipv6-mode>dual-stack</ipv6-mode>
          <bind-ip-address>10.81.103.86</bind-ip-address>
          <bind-ip-address-internal>203.203.203.101</bind-ip-address-internal>
          <bind-ipv6-address>2001:420:27c1:903::86</bind-ipv6-address>
          <bind-ipv6-address-internal>2002:4888:203:203::203:101</bind-ipv6-address-internal>
          <istio>true</istio>
          <master-vip>203.203.203.101</master-vip>
          <master-vip-cidr>24</master-vip-cidr>
          <master-vip-ipv6>2002:4888:203:203::203:101</master-vip-ipv6>
          <master-vip-interface>bd0.k8s.303</master-vip-interface>
          <additional-master-vip>10.81.103.86</additional-master-vip>
          <additional-master-vip-ipv6>2001:420:27c1:903::86</additional-master-vip-ipv6>
          <additional-master-vip-interface>bd0.mgmt.3103</additional-master-vip-interface>
          <virtual-ip-vrrp-router-id>60</virtual-ip-vrrp-router-id>
          <pod-subnet>192.203.0.0/16</pod-subnet>
          <pod-subnet-ipv6>2002:4888:192:203::/96</pod-subnet-ipv6>
          <node-pod-subnet-ipv6-mask>112</node-pod-subnet-ipv6-mask>
          <allow-insecure-registry>true</allow-insecure-registry>
          <restrict-logging>false</restrict-logging>
          <enable-network-policy>true</enable-network-policy>
```

```
<enable-ssh-firewall-rules>false</enable-ssh-firewall-rules>
<tuned>true</tuned>
<ntp-address>2001:420:27c1:903::109</ntp-address>
<initial-boot>
  <default-user>cloud-user</default-user>
  <default-user-password>Starent@123</default-user-password>
  <netplan>
    <ethernet>
      <device-id>eno1</device-id>
      <dhcp4>false</dhcp4>
      <dhcp6>false</dhcp6>
    </ethernet>
    <ethernet>
      <device-id>eno2</device-id>
      <dhcp4>false</dhcp4>
      <dhcp6>false</dhcp6>
    </ethernet>
    <ethernet>
      <device-id>eno5</device-id>
      <dhcp4>false</dhcp4>
      <dhcp6>false</dhcp6>
    </ethernet>
    <ethernet>
      <device-id>eno6</device-id>
      <dhcp4>false</dhcp4>
      <dhcp6>false</dhcp6>
    </ethernet>
    <ethernet>
      <device-id>enp216s0f0</device-id>
      <dhcp4>false</dhcp4>
      <dhcp6>false</dhcp6>
    </ethernet>
    <ethernet>
      <device-id>enp216s0f1</device-id>
      <dhcp4>false</dhcp4>
      <dhcp6>false</dhcp6>
    </ethernet>
    <ethernet>
      <device-id>enp94s0f0</device-id>
      <dhcp4>false</dhcp4>
      <dhcp6>false</dhcp6>
    </ethernet>
    <ethernet>
      <device-id>enp94s0f1</device-id>
      <dhcp4>false</dhcp4>
      <dhcp6>false</dhcp6>
    </ethernet>
    <bond>
      <device-id>bd0</device-id>
      <dhcp4>false</dhcp4>
      <dhcp6>false</dhcp6>
      <optional>true</optional>
      <interface>eno5</interface>
      <interface>eno6</interface>
      <parameters>
        <mode>active-backup</mode>
        <mii-monitor-interval>100</mii-monitor-interval>
        <fail-over-mac-policy>active</fail-over-mac-policy>
      </parameters>
    </bond>
    <bond>
      <device-id>bd1</device-id>
      <dhcp4>false</dhcp4>
      <dhcp6>false</dhcp6>
```

```
      <optional>true</optional>
      <interface>enp216s0f1</interface>
      <interface>enp94s0f0</interface>
      <parameters>
        <mode>active-backup</mode>
        <mii-monitor-interval>100</mii-monitor-interval>
        <fail-over-mac-policy>active</fail-over-mac-policy>
      </parameters>
    </bond>
    <bond>
      <device-id>bd0</device-id>
      <dhcp4>false</dhcp4>
      <dhcp6>false</dhcp6>
      <optional>true</optional>
      <interface>enp216s0f0</interface>
      <interface>enp94s0f1</interface>
      <parameters>
        <mode>active-backup</mode>
        <mii-monitor-interval>100</mii-monitor-interval>
        <fail-over-mac-policy>active</fail-over-mac-policy>
      </parameters>
    </bond>
    <vlan>
      <device-id>bd0.k8s.303</device-id>
      <dhcp4>false</dhcp4>
      <dhcp6>false</dhcp6>
      <id>303</id>
      <link>bd0</link>
    </vlan>
    <vlan>
      <device-id>bd0.mgmt.3103</device-id>
      <dhcp4>false</dhcp4>
      <dhcp6>false</dhcp6>
      <id>3103</id>
      <link>bd0</link>
      <nameservers>
        <search>cisco.com mitg-bxb300.cisco.com</search>
        <address>2001:420:200:1::a</address>
        <address>2001:420:210d::a</address>
      </nameservers>
    </vlan>
  </netplan>
</initial-boot>
<cimc>
  <user>admin</user>
  <password>Starent@123</password>
    <ntp-address>2001:420:27c1:903::109</ntp-address>
    <storage-adaptor>
      <create-virtual-drive>true</create-virtual-drive>
    </storage-adaptor>
    <bios>
      <configured-boot-mode>Uefi</configured-boot-mode>
      <uefi-secure-boot>yes</uefi-secure-boot>
    </bios>
</cimc>
<node>
  <name>server-1</name>
  <host-profile>
    <repository>bng-ht-sysctl-enable</repository>
  </host-profile>
  <type>control-plane</type>
  <ssh-ip>203.203.203.11</ssh-ip>
  <ssh-ipv6>2002:4888:203:203::203:11</ssh-ipv6>
  <node-ip>203.203.203.11</node-ip>
```

```
<node-ipv6>2002:4888:203:203::203:11</node-ipv6>
<node-label>
  <key>smi.cisco.com/node-type</key>
  <value>oam</value>
</node-label>
<node-label>
  <key>smi.cisco.com/proto-type</key>
  <value>protocol</value>
</node-label>
<node-label>
  <key>comp-type</key>
  <value>mcollector-ssd</value>
</node-label>
<cimc>
  <ip-address>2001:420:27c1:903::53</ip-address>
</cimc>
<initial-boot>
  <netplan>
    <vlan>
      <device-id>bd0.k8s.303</device-id>
      <address>203.203.203.11/24</address>
      <address>2002:4888:203:203::203:11/112</address>
    </vlan>
    <vlan>
      <device-id>bd0.mgmt.3103</device-id>
      <address>10.81.103.72/24</address>
      <address>2001:420:27c1:903::72/64</address>
      <gateway4>10.81.103.1</gateway4>
      <gateway6>2001:420:27c1:903::1</gateway6>
    </vlan>
    <vlan>
      <device-id>bd1.n4.319</device-id>
      <address>219.219.219.11/24</address>
      <address>2002:4888:219:219::219:11/112</address>
      <id>319</id>
      <link>bd1</link>
    </vlan>
    <vlan>
      <device-id>bd1.radius.320</device-id>
      <address>220.220.220.11/24</address>
      <address>2002:4888:220:220::220:11/64</address>
      <id>320</id>
      <link>bd1</link>
    </vlan>
  </netplan>
</initial-boot>
<netplan-additions>
  <vlan>
    <device-id>bd1.n4.319</device-id>
    <route>
      <to>101.101.101.0/24</to>
      <via>219.219.219.1</via>
    </route>
    <route>
      <to>192.69.0.0/16</to>
      <via>219.219.219.1</via>
    </route>
    <route>
      <to>2002:4888:101:101::101:0/64</to>
      <via>2002:4888:219:219::219:1</via>
    </route>
    <route>
      <to>2002:4888:192:69::/64</to>
      <via>2002:4888:219:219::219:1</via>
```

```
      </route>
    </vlan>
    <vlan>
      <device-id>bd1.radius.320</device-id>
      <route>
        <to>192.71.0.0/16</to>
        <via>220.220.220.1</via>
      </route>
      <route>
        <to>2002:4888:192:71::100:0/64</to>
        <via>2002:4888:220:220::220:1</via>
      </route>
    </vlan>
  </netplan-additions>
</node>
<node>
  <name>server-2</name>
  <host-profile>
    <repository>bng-ht-enable</repository>
  </host-profile>
  <type>control-plane</type>
  <ssh-ip>203.203.203.12</ssh-ip>
  <ssh-ipv6>2002:4888:203:203::203:12</ssh-ipv6>
  <node-ip>203.203.203.12</node-ip>
  <node-ipv6>2002:4888:203:203::203:12</node-ipv6>
  <node-label>
    <key>smi.cisco.com/node-type</key>
    <value>oam</value>
  </node-label>
  <node-label>
    <key>smi.cisco.com/sess-type</key>
    <value>cdl-node</value>
  </node-label>
  <node-label>
    <key>smi.cisco.com/svc-type</key>
    <value>service</value>
  </node-label>
  <node-label>
    <key>comp-type</key>
    <value>prom-ssd</value>
  </node-label>
  <cimc>
    <ip-address>2001:420:27c1:903::54</ip-address>
  </cimc>
  <initial-boot>
    <netplan>
      <vlan>
        <device-id>bd0.k8s.303</device-id>
        <address>203.203.203.12/24</address>
        <address>2002:4888:203:203::203:12/112</address>
      </vlan>
      <vlan>
        <device-id>bd0.mgmt.3103</device-id>
        <address>10.81.103.73/24</address>
        <address>2001:420:27c1:903::73/64</address>
        <gateway4>10.81.103.1</gateway4>
        <gateway6>2001:420:27c1:903::1</gateway6>
      </vlan>
    </netplan>
  </initial-boot>
</node>
<node>
  <name>server-3</name>
  <host-profile>
```

```
    <repository>bng-ht-enable</repository>
  </host-profile>
  <type>control-plane</type>
  <ssh-ip>203.203.203.13</ssh-ip>
  <ssh-ipv6>2002:4888:203:203::203:13</ssh-ipv6>
  <node-ip>203.203.203.13</node-ip>
  <node-ipv6>2002:4888:203:203::203:13</node-ipv6>
  <node-label>
    <key>smi.cisco.com/node-type</key>
    <value>oam</value>
  </node-label>
  <node-label>
    <key>smi.cisco.com/sess-type</key>
    <value>cdl-node</value>
  </node-label>
  <node-label>
    <key>smi.cisco.com/svc-type</key>
    <value>service</value>
  </node-label>
  <node-label>
    <key>comp-type</key>
    <value>loki-ssd</value>
  </node-label>
  <cimc>
    <ip-address>2001:420:27c1:903::55</ip-address>
  </cimc>
  <initial-boot>
    <netplan>
      <vlan>
        <device-id>bd0.k8s.303</device-id>
        <address>203.203.203.13/24</address>
        <address>2002:4888:203:203::203:13/112</address>
      </vlan>
      <vlan>
        <device-id>bd0.mgmt.3103</device-id>
        <address>10.81.103.74/24</address>
        <address>2001:420:27c1:903::74/64</address>
        <gateway4>10.81.103.1</gateway4>
        <gateway6>2001:420:27c1:903::1</gateway6>
      </vlan>
    </netplan>
  </initial-boot>
</node>
<node>
  <name>server-4</name>
  <host-profile>
    <repository>bng-ht-sysctl-enable</repository>
  </host-profile>
  <type>worker</type>
  <host-profile>
    <repository>bng-ht-enable</repository>
  </host-profile>
  <ssh-ip>203.203.203.14</ssh-ip>
  <ssh-ipv6>2002:4888:203:203::203:14</ssh-ipv6>
  <node-ip>203.203.203.14</node-ip>
  <node-ipv6>2002:4888:203:203::203:14</node-ipv6>
  <node-label>
    <key>smi.cisco.com/proto-type</key>
    <value>protocol</value>
  </node-label>
  <cimc>
    <ip-address>2001:420:27c1:903::56</ip-address>
  </cimc>
  <initial-boot>
```

```
      <netplan>
        <vlan>
          <device-id>bd0.k8s.303</device-id>
          <address>203.203.203.14/24</address>
          <address>2002:4888:203:203::203:14/112</address>
        </vlan>
        <vlan>
          <device-id>bd0.mgmt.3103</device-id>
          <address>10.81.103.75/24</address>
          <address>2001:420:27c1:903::75/64</address>
          <gateway4>10.81.103.1</gateway4>
          <gateway6>2001:420:27c1:903::1</gateway6>
        </vlan>
        <vlan>
          <device-id>bd1.n4.319</device-id>
          <address>219.219.219.12/24</address>
          <address>2002:4888:219:219::219:12/112</address>
          <id>319</id>
          <link>bd1</link>
        </vlan>
        <vlan>
          <device-id>bd1.radius.320</device-id>
          <address>220.220.220.12/24</address>
          <address>2002:4888:220:220::220:12/64</address>
          <id>320</id>
          <link>bd1</link>
        </vlan>
      </netplan>
    </initial-boot>
    <netplan-additions>
      <vlan>
        <device-id>bd1.n4.319</device-id>
        <route>
          <to>101.101.101.0/24</to>
          <via>219.219.219.1</via>
        </route>
        <route>
          <to>192.69.0.0/16</to>
          <via>219.219.219.1</via>
        </route>
        <route>
          <to>2002:4888:101:101::101:0/64</to>
          <via>2002:4888:219:219::219:1</via>
        </route>
        <route>
          <to>2002:4888:192:69::/64</to>
          <via>2002:4888:219:219::219:1</via>
        </route>
      </vlan>
      <vlan>
        <device-id>bd1.radius.320</device-id>
        <route>
          <to>192.71.0.0/16</to>
          <via>220.220.220.1</via>
        </route>
        <route>
          <to>2002:4888:192:71::100:0/64</to>
          <via>2002:4888:220:220::220:1</via>
        </route>
      </vlan>
    </netplan-additions>
  </node>
  <virtual-ip>
    <name>udpvip</name>
```

```
            <check-ports>28000</check-ports>
            <interface>bd1.n4.319</interface>
            <router-id>222</router-id>
            <check-interface>bd0.k8s.303</check-interface>
            <check-interface>bd1.n4.319</check-interface>
            <check-interface>bd1.radius.320</check-interface>
            <address>
              <address>203.203.203.51</address>
              <device>bd0.k8s.303</device>
              <prefix-length>24</prefix-length>
            </address>
            <address>
              <address>219.219.219.51</address>
              <device>bd1.n4.319</device>
              <prefix-length>24</prefix-length>
            </address>
            <address>
              <address>220.220.220.51</address>
              <device>bd1.radius.320</device>
              <prefix-length>24</prefix-length>
            </address>
            <address>
              <address>2002:4888:203:203::203:51</address>
              <device>bd0.k8s.303</device>
              <prefix-length>112</prefix-length>
            </address>
            <address>
              <address>2002:4888:219:219::219:51</address>
              <device>bd1.n4.319</device>
              <prefix-length>64</prefix-length>
            </address>
            <address>
              <address>2002:4888:220:220::220:51</address>
              <device>bd1.radius.320</device>
              <prefix-length>64</prefix-length>
            </address>
            <node>
              <name>server-1</name>
              <priority>100</priority>
            </node>
            <node>
              <name>server-4</name>
              <priority>100</priority>
            </node>
          </virtual-ip>
        </cluster>
      </clusters>
    </deployment>
  </smi>
</config>
```

Step 2 View the SMI deployment plan and verify the deployment status of the K8s cluster.

Example:

```
admin@ncs# show smi deployment-plan bng-cfp-l3-deployment plan | tab
                                                BACK
                    POST ACTION
TYPE           NAME                             TRACK  GOAL  STATE                STATUS   WHEN
               ref  STATUS          MESSAGE
-------------------------------------------------------------------------------------------------
self           self                             false  -     init                 reached
2024-09-30T05:00:47  -    -               -
                                                             ready                reached
```

```
2024-09-30T05:42:15  -    -               -
deployment-bm  bng-cfp-l3-deployment           false  -     init                  reached
2024-09-30T05:00:47  -    -               -
                                                           smid-ns:config-apply  reached
2024-09-30T05:00:47  -    create-reached  -
                                                           ready                 reached
2024-09-30T05:00:47  -    -               -
ucs-cluster    bng-cfp-l3-tb02-151             false  -     init                  reached
2024-09-30T05:00:47  -    -               -
                                                           smid-ns:config-apply  reached
2024-09-30T05:00:47  -    -               -
                                                           ready                 reached
2024-09-30T05:42:15  -    -               -
cluster-node   bng-cfp-l3-tb02-151-s1-r1-svr-1  false  -     init                  reached
2024-09-30T05:00:47  -    -               -
                                                           smid-ns:config-apply  reached
2024-09-30T05:00:47  -    -               -
                                                           ready                 reached
2024-09-30T05:42:15  -    -               -
software       cee-smi-cfp-repo                false  -     init                  reached
2024-09-30T05:01:30  -    -               -
                                                           smid-ns:config-apply  reached
2024-09-30T05:01:30  -    -               -
                                                           ready                 reached
2024-09-30T05:01:30  -    -               -
software       cnbng-smi-cfp-repo              false  -     init                  reached
2024-09-30T05:01:30  -    -               -
                                                           smid-ns:config-apply  reached
2024-09-30T05:01:30  -    -               -
                                                           ready                 reached
2024-09-30T05:01:30  -    -               -
```

Functions Services

The functions services deploy functions on the clusters using the Cluster Manager. These services manage various network functions, allowing you to deploy the following:

- Containerized Network Functions (CNFs)
  - Common Execution Environment (CEE)
  - Broadband Network Gateway Function (BNG)

Deploy a CNF

Containerized Network Functions (CNFs) are managed using Kubernetes-style orchestration, ensuring consistent lifecycle management across all containers.

To deploy CNFs, you need to have the CNF images on your system.

These are the steps to deploy a CNF:

Procedure

Step 1 Log in to NSO and configure the SMI deployment software repository auth to deploy the CNF.

Example:

```
<config xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <smi xmlns="http://cisco.com/ns/nso/cfp/cisco-smi">
    <deployment xmlns="http://cisco.com/ns/nso/cfp/cisco-smi-deployment">
      <name>bare-metal</name>
      <software>
        <repository-auth>
          <name>imgcread</name>
          <user>admin</user>
          <password>admin</password>
        </repository-auth>
      </software>
    </deployment>
  </smi>
</config>
```



Note If you are using a simple HTTP server, authentication credentials are not required.

Step 2 Add the software repository configuration to NSO and commit the changes to deploy the service. The following is a sample payload:

Example:

```
<config xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <smi xmlns="http://cisco.com/ns/nso/cfp/cisco-smi">
    <deployment xmlns="http://cisco.com/ns/nso/cfp/cisco-smi-deployment">
      <name>bng-cfp-deployment</name>
      <local-user>admin</local-user>
      <cluster-manager>
        <manager-device>smi-cm-109</manager-device>
      </cluster-manager>
      <software>
        <repository>
          <name>cee-smi-cfp-repo</name>
          <type>cnf</type>

<url>https://engci-maven-master.cisco.com/artifactory/smi-fuse-internal-group/releases/smi-apps/smi-cee-products/2024.03.1.i12-offline/cee-2024.03.1.i12.tar</url>

          <sha256>5d87baee367bbc9fb8184699d73735873b8b7999f347d59f3d01dd99d55ff91e</sha256>
          <repository-auth>imgcread</repository-auth>
         <ned-id xmlns:id="http://tail-f.com/ns/ned-id/cisco-cee-nc-1.1">id:cisco-cee-nc-1.1</ned-id>

        </repository>
        <repository>
          <name>cnbng-smi-cfp-repo</name>
          <type>cnf</type>

<url>https://engci-maven-master.cisco.com/artifactory/mobile-cnat-charts-release/releng-builds/2024.03.0/bng/2024.03.0.i46/bng.2024.03.0.i46-offline/bng.2024.03.0.i46.SSA.tgz</url>

          <sha256>152c21e2d6c12393d6d557439c438fffae7cc1c623a8426992766acf5b93afec</sha256>
          <repository-auth>imgcread</repository-auth>
         <ned-id xmlns:id="http://tail-f.com/ns/ned-id/cisco-bng-nc-1.1">id:cisco-bng-nc-1.1</ned-id>
```

```
        </repository>
        <repository-auth>
          <name>imgcread</name>
          <user>madhs</user>
          <password>...</password>
          <accept-self-signed-certificate>true</accept-self-signed-certificate>
          <allow-dev-image>true</allow-dev-image>
        </repository-auth>
      </software>
    </deployment>
  </smi>
</config>
```

Step 3 Configure the functions on the CNF using the deployed SMI Cluster Manager and perform a load merge.

Example:

```
<smi xmlns="http://cisco.com/ns/nso/cfp/cisco-smi">
  <functions xmlns="http://cisco.com/ns/nso/cfp/cisco-smi-functions">
    <function>
      <app-name>cee</app-name>
      <name>cee-smi-cfp-109</name>
      <local-user>admin</local-user>
      <first-boot-username>admin</first-boot-username>
      <first-boot-password>Starent@123</first-boot-password>
      <authentication-group>smi-cm-auth-109</authentication-group>
      <deployer>bng-cfp-deployment</deployer>
      <repository>cee-smi-cfp-repo</repository>
      <managed-cluster>bng-cfp-cluster-53</managed-cluster>
      <netconf-port>2027</netconf-port>
      <ssh-ipv6>2001:420:27c1:903::86</ssh-ipv6>
      <netconf-ipv6>2001:420:27c1:903::86</netconf-ipv6>
      <ssh-port>2028</ssh-port>
      <use-volume-claims>true</use-volume-claims>
      <auto-deploy>true</auto-deploy>
      <single-node>false</single-node>
      <ingress-hostname>cnbng-tb3.nip.io</ingress-hostname>
    </function>
  </functions>
</smi>
```

Step 4 Poll the plan component for each function and view the status of the service. The following example shows how to view the status of the service.

Example:

```
admin@ncs# show smi functions function-plan cee cee-smi-cfp-151 plan | tab
                                    BACK
          POST ACTION
TYPE             NAME               TRACK  GOAL  STATE                  STATUS   WHEN
     ref  STATUS          MESSAGE
-------------------------------------------------------------------------------------------------------------
self             self               false  -     init                   reached  2024-09-30T05:01:28
  -    -               -
                                                 ready                  reached  2024-10-03T02:49:13
  -    -               -
function         cee                false  -     init                   reached  2024-09-30T05:01:28
  -    -               -
                                                smif-ns:cluster-ready   reached  2024-09-30T05:42:14
  -    create-reached  -
                                                smif-ns:onboarded       reached  2024-09-30T05:43:03
  -    create-reached  -
                                                validate               reached  2024-10-03T02:49:13
  -    -               -
                                                ready                  reached  2024-10-03T02:49:13
```

```
  -    -               -
managed-cluster  bng-cfp-l3-tb02-151  false  -     init                  reached  2024-09-30T05:01:28
  -    -               -
                                                  deployed              reached  2024-09-30T05:01:28
  -    create-reached  -
                                                  ready                 reached  2024-09-30T05:42:14
  -    -               -
```

Upgrade a CNF

Before you begin

- Ensure that you have installed the SMI software using the deployment services.

Procedure

Step 1 Create a repository with the newer version of CNF to upgrade the existing CNF. The following is a sample payload:

Example:

```
<config xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <smi xmlns="http://cisco.com/ns/nso/cfp/cisco-smi">
    <deployment xmlns="http://cisco.com/ns/nso/cfp/cisco-smi-deployment">
      <name>bng-cfp-deployment</name>
      <local-user>admin</local-user>
      <cluster-manager>
        <manager-device>smi-cm-109</manager-device>
      </cluster-manager>
      <software>
        <repository>
          <name>cee-smi-cfp-repo-04</name>
          <type>cnf</type>

<url>https://engci-maven-master.cisco.com/artifactory/smi-fuse-internal-group/releases/smi-apps/smi-cee-products/2024.04.1.i12-offline/cee-2024.04.1.i12.tar</url>

          <sha256>6d87baee367bbc9fb8184699d73735873b8b7999f347d59f3d01dd99d55ff91e</sha256>
          <repository-auth>imgcread</repository-auth>
         <ned-id xmlns:id="http://tail-f.com/ns/ned-id/cisco-cee-nc-1.1">id:cisco-cee-nc-1.1</ned-id>

        </repository>
      </software>
    </deployment>
  </smi>
</config>
```

Step 2 Set the SMI function to use the new repository.

Example:

```
<smi xmlns="http://cisco.com/ns/nso/cfp/cisco-smi">
  <functions xmlns="http://cisco.com/ns/nso/cfp/cisco-smi-functions">
    <function>
      <app-name>cee</app-name>
      <name>cee-smi-cfp-109</name>
      <local-user>admin</local-user>
      <first-boot-username>admin</first-boot-username>
```

```
      <first-boot-password>Starent@123</first-boot-password>
      <authentication-group>smi-cm-auth-109</authentication-group>
      <deployer>bng-cfp-deployment</deployer>
      <repository>cee-smi-cfp-repo-04</repository>
      <managed-cluster>bng-cfp-cluster-53</managed-cluster>
      <netconf-port>2027</netconf-port>
      <ssh-ipv6>2001:420:27c1:903::86</ssh-ipv6>
      <netconf-ipv6>2001:420:27c1:903::86</netconf-ipv6>
      <ssh-port>2028</ssh-port>
      <use-volume-claims>true</use-volume-claims>
      <auto-deploy>true</auto-deploy>
      <single-node>false</single-node>
      <ingress-hostname>cnbng-tb3.nip.io</ingress-hostname>
    </function>
  </functions>
</smi>
```

Step 3 [Optional] If the auto-sync functionality is disabled, manually trigger a cluster-sync.

Applications Services

The applications layer sits on top of the function services and provides additional functionality once the function is ready. It is used to apply day-1 configuration to the BNG Ops-Center via NSO device templates. This layer ensures that the necessary configurations are in place for the application to operate effectively from the start.

Apply Day-1 Configuration to BNG Ops-center

Procedure

Step 1 Create a device template with day-1 BNG configuration.

Example:

```
<config xmlns="http://tail-f.com/ns/config/1.0">
  <devices xmlns="http://tail-f.com/ns/ncs">
    <template>
      <name>template-bng-ops-109</name>
      <ned-id>
        <id xmlns:id="http://tail-f.com/ns/ned-id/cisco-bng-nc-1.1">id:cisco-bng-nc-1.1</id>
        <config>
          <ipam xmlns="http://cisco.com/cisco-cn-ipam">
            <instance>
              <instance-id>1</instance-id>
...
        </config>
      </ned-id>
    </template>
  </devices>
</config>
```

Step 2 Apply the device-template to bng-ops-center.

Example:

```
<smi xmlns="http://cisco.com/ns/nso/cfp/cisco-smi">
    <applications xmlns="http://cisco.com/ns/nso/cfp/cisco-smi-applications">
      <application>
        <bng>
          <name>bng-smi-cfp-109</name>
          <local-user>admin</local-user>
          <deployment>
            <authentication-group>smi-cm-auth-109</authentication-group>
            <deployer>bng-cfp-deployment</deployer>
            <managed-cluster>bng-cfp-cluster-53</managed-cluster>
            <repository>cnbng-smi-cfp-repo</repository>
            <first-boot-password>Starent@123</first-boot-password>
          </deployment>
          <device-template>
            <name>template-bng-ops-109-02</name>
          </device-template>
        </bng>
      </application>
    </applications>
</smi>
```

Step 3 View the application status.

Example:

```
admin@ncs# show smi applications application-plan bng plan | tab

                                POST
                                 LOG                        BACK
                                ACTION
NAME             FAILED  MESSAGE  ENTRY  TYPE         NAME  TRACK  GOAL  STATE
STATUS   WHEN                 ref  STATUS  ID
-------------------------------------------------------------------------------------------
bng-smi-cfp-151  -       -        -      self         self  false  -     init
reached  2024-09-30T05:04:56  -    -
                                                                        ready
reached  2024-09-30T05:43:05  -    -
                                       application  bng   false  -     init
reached  2024-09-30T05:04:56  -    -
                                                                        smia-ns:onboarded
reached  2024-09-30T05:43:03  -    -
                                                                        application-configured
reached  2024-09-30T05:43:03  -    -
                                                                        system-configured
reached  2024-09-30T05:43:05  -    -
                                                                        ready
reached  2024-09-30T05:43:05  -    -
```

Disable the auto-sync Feature

The auto-sync feature is enabled by default to automatically trigger cluster-sync on the deployer. You can choose to disable the auto-sync feature and manually trigger cluster-sync by setting the **sync-disabled** flag to true, either at the global level or at the cluster level (for managed clusters).

When auto-sync is disabled, the SMI deployment plan and the functions plan are updated based on the notifications received.

Procedure

Step 1 At the global level, set the **sync-disabled flag** to true in SMI settings deployment.

Example:

```
ncs_cli -u admin -C
admin@ncs# configure
admin@ncs(config)# load merge cluster-payload-example.xml
admin@ncs(config)# smi deployment settings sync-disabled true
admin@ncs(config)# commit
```

Step 2 At the cluster level, set the **sync-disabled flag** flag to true in the **smi deployment** *depl_name* **clusters** *cluster_name* command.

Example:

```
ncs_cli -u admin -C
admin@ncs# configure
admin@ncs(config)# load merge cluster-payload-example.xml
admin@ncs(config)# smi deployment test clusters cluster c1 sync-disabled
true
admin@ncs(config)# commit
```

In this example, *cluster c1* is deployed under the SMI deployment *test*.

Step 3 Once the cluster is up, view the plan for the deployment, and observe that the clusters are not synced.

Example:

```
admin@ncs# show smi deployment-plan test
smi deployment-plan bng-cfp-l3-deployment
 result smi-device smi-cm-31
 result local-user admin
 result cluster-result bng-cfp-l3-tb02-151
  ha-cluster false
  sync-state state DEPLOYED
  sync-status status DONE
  sync-status event-time 2024-10-03T00:08:53.784+00:00
  node-result s1-r1-svr-1
   drain-status NONE
   node-state state JOINED
   node-state event-time 2024-10-03T00:08:53.785+00:00
   node-state node-sync false
  functions  [ "/smi:smi/smif:functions/function{bng bng-smi-cfp-151}"
"/smi:smi/smif:functions/function{cee cee-smi-cfp-151}" ]
 plan component self self
  back-track false
  state init
   status reached
   when   2024-09-30T05:00:47
  state ready
   status reached
   when   2024-09-30T05:42:15
 plan component deployment-bm bng-cfp-l3-deployment
  back-track false
  state init
   status reached
   when   2024-09-30T05:00:47
  state smid-ns:config-apply
   status             reached
   when               2024-09-30T05:00:47
```

```
  post-action-status create-reached
 state ready
  status reached
  when   2024-09-30T05:00:47
plan component ucs-cluster bng-cfp-l3-tb02-151
 back-track false
 state init
  status reached
  when   2024-09-30T05:00:47
 state smid-ns:config-apply
  status reached
  when   2024-09-30T05:00:47
 state ready
  status reached
  when   2024-09-30T05:42:15
plan component cluster-node bng-cfp-l3-tb02-151-s1-r1-svr-1
 back-track false
 state init
  status reached
  when   2024-09-30T05:00:47
 state smid-ns:config-apply
  status reached
  when   2024-09-30T05:00:47
 state ready
  status reached
  when   2024-09-30T05:42:15
plan component software cee-smi-cfp-repo
 back-track false
 state init
  status reached
  when   2024-09-30T05:01:30
 state smid-ns:config-apply
  status reached
  when   2024-09-30T05:01:30
 state ready
  status reached
  when   2024-09-30T05:01:30
plan component software cnbng-smi-cfp-repo
 back-track false
 state init
  status reached
  when   2024-09-30T05:01:30
 state smid-ns:config-apply
  status reached
  when   2024-09-30T05:01:30
 state ready
  status reached
  when   2024-09-30T05:01:30
```

Trigger the Sync Action

The sync action initiates synchronization on the Cluster Manager and re-deploys the associated functions and clusters. This action also restarts the monitoring services and components accordingly. In managed clusters, NSO triggers a sync-from action once the sync action on the Cluster Manager is complete. If the sync-to flag in NSO is set to true, NSO will also trigger a sync-to action.

To trigger the sync action through the deployment service, use the following command:

```
admin@ncs(config)# smi deployment deployment-name clusters sync [ cluster-name-1
cluster-name-2 ... ]
```

When a sync action is triggered, the plan backtracks and reaches the desired state only after the sync action is complete, and both the functions and the Cluster Manager are reachable.

The sync action supports the following options, which are set to false by default:

- sync-to
- force-vm-redeploy
- force-partition-redeploy
- reset-k8s-nodes
- purge-data-disks

You can use one or all of these options as needed. For example, the following command demonstrates the usage of these options:

```
admin@ncs(config)# smi deployment test clusters sync cluster [ c3 ]
force-partition-redeploy true force-vm-redeploy true purge-data-disks
true reset-k8s-nodes true sync-to true
```

Delete the SMI Deployment

You must first delete the CNF before deleting the SMI deployment. After each deletion, verify that there are no zombie processes.

To delete the SMI deployment, follow these steps:

1. Delete the cnBNG application, if it is installed.

   ```
   admin@ncs(config)# no smi applications application bng bng-1
   ```

2. Delete all applicable functions.

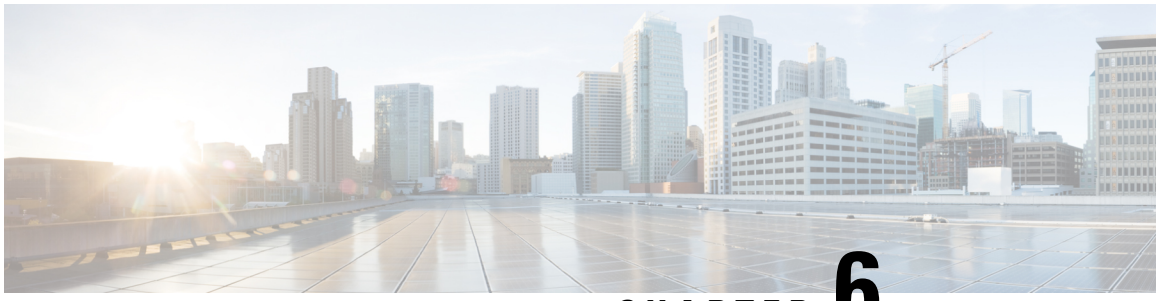
   ```
   admin@ncs(config)# no smi functions function cee cee-test1
   admin@ncs(config)# no smi functions function bng bng-1
   admin@ncs(config)# commit
   ```

3. Delete the software repository associated with the bare-metal deployment.

   ```
   admin@ncs(config)# no smi deployment bare-metal software repository
   admin@ncs(config)# commit
   ```

4. Delete the bare-metal deployment.

   ```
   admin@ncs(config)# no smi deployment bare-metal
   admin@ncs(config)# commit
   ```



CHAPTER 6

Smart Licensing

- Feature Summary and Revision History, on page 101
- Feature Description, on page 102
- cnBNG Control Plane Smart Licensing, on page 103
- Configuring Smart Software Licensing for cnBNG CP, on page 106
- Monitoring and Troubleshooting Smart Licensing, on page 116

Feature Summary and Revision History

Summary Data

Table 12: Summary Data

Applicable Products or Functional Area	cnBNG
Applicable Platform(s)	SMI
Feature Default Setting	Enabled – Always-on
Related Changes in this Release	Not Applicable
Related Documentation	Not Applicable

Revision History

Table 13: Revision History

Revision Details	Release
Support for Advantage license was added.	2026.03.0
First introduced.	2022.01.0

Feature Description

Cisco employs two types of license models - Legacy Licensing and Smart Software Licensing. Legacy Licensing consists of software activation by installing Product Activation Keys (PAK) on to the Cisco product. A Product Activation Key is a purchasable item, ordered in the same manner as other Cisco equipment and used to obtain license files for feature set on Cisco Products. This traditional licensing does not need any online communication with the Cisco licensing server.

Smart Software Licensing is a cloud-based licensing of the end-to-end platform through the use of a few tools that authorize and deliver license reporting. Smart Software Licensing functionality incorporated into the NFs complete the product registration and authorization. The cnBNG Control Plane (CP) supports the Smart Software Licensing model.

Smart Licensing simplifies the purchase, deployment, and management of Cisco software assets. Entitlements are purchased through your Cisco account through Cisco Commerce Workspace (CCW) and immediately available in your Virtual Account for usage. This approach eliminates the need to install license files on every device. Smart-enabled products communicate directly to Cisco to report consumption. A single location—Cisco Software Central—is available for customers to manage Cisco software licenses. License ownership and consumption are readily available to help make a better purchase decision that is based on consumption or business need.

For more information on Cisco Smart Licensing, see https://www.cisco.com/c/en/us/buy/smart-accounts/software-licensing.html.

Cisco Software Central

Cisco Software Central (CSC) enables the management of software licenses and the smart account from a single portal. The CSC interface allows you to enable your product, manage entitlements, renew, and upgrade software. You need a functioning smart account to complete the registration process.

To access Cisco Software Central, see https://software.cisco.com.

Smart Accounts and Virtual Accounts

A Smart Account provides a single location for all smart-enabled products and entitlements. It helps in procurement, deployment, and maintenance of Cisco Software. When creating a smart account, you must have the authority to represent the requesting organization. After submission, the request goes through approval process.

A Virtual Account exists as a sub-account within the smart account. Virtual Accounts are customer-defined based on the organizational layout, business function, geography, or any defined hierarchy. Smart account administrator creates and maintains the virtual accounts.

For information on setting up or managing the Smart Accounts, see https://software.cisco.com.

Requesting a Cisco Smart Account

A Cisco Smart Account is an account where smart licensing-enabled products are available. A Cisco smart account allows you to manage and activate your licenses to devices, monitor license use, and track Cisco license purchases. Through transparent access, you have a real-time view into your smart licensing products.

IT administrators can manage licenses and account users within the organization's smart account through Cisco Software Central. To create a Cisco Smart Account, perform the following steps:

Procedure

Step 1 Visit the following URL:

`https://software.cisco.com`

Step 2 Log in using your credentials, and click **Request a Smart Account** in the **Administration** area.

The **Smart Account Request** window appears.

Step 3 Under **Create Account**, select one of the following options:

- **Yes, I have authority to represent my company and want to create the Smart Account**. If you select this option, you agree to authorize to create and manage product and service entitlements, users, and roles, on behalf of the organization.
- **No, the person specified below will create the account**. If you select this option, you must enter the email address of the person who creates the smart account.

Step 4 Under **Account Information**,

a) Click **Edit** beside **Account Domain Identifier**.
b) In the **Edit Account Identifier** dialog box, enter the domain, and click **OK**. By default, the domain is based on the email address of the person creating the account, and must belong to the company that will own this account.
c) Enter the **Account Name** (typically, the company name).

Step 5 Click **Continue**.

The Smart Account request will be in pending status until it is approved by the Account Domain Identifier. After the approval, you will receive an email confirmation with instructions for completing the setup process.

cnBNG Control Plane Smart Licensing

Smart Licensing (SL) is a cloud-based software licensing model that allows you to

- activate and manage Cisco software licenses across your organization
- easily track the status of your licenses and software usage trends, and
- pool licenses or entitlements across the entire organization in a flexible and automated manner.

Table 14: Feature History Table

Feature Name	Release Information	Feature Description
Support for Advantage license	2026.03.0	Added support for the Advantage license. The Advantage license is the subscriber-session licensing tier for active sessions with redundancy, such as SRG or CPGR, and for LNS sessions.

The cnBNG Control Plane (CP) Smart Licensing supports application entitlement for online and offline licensing. Application usage is unrestricted during all stages of licensing, including Out of Compliance, or OOC, and expired stages.

Note A 90-day evaluation period is granted for all licenses in use. The functionality and operation of the applications is unrestricted even after the evaluation period ends.

Smart Licensing reports cnBNG CP license consumption by using tags. These tags identify subscriber-session licenses that are consumed based on deployment type and session characteristics.

The cnBNG calculates license information every hour. During the subsequent hour, if the smart agent queries license consumption, cnBNG reports the same value. After one hour, the license information is recalculated to provide updated data.

License types

License types are license categories that define the level of entitlement for cnBNG CP.

A license type defines the type or level of license entitlement. A tag is the Smart Licensing identifier used to report and track license usage for that tier.

cnBNG CP Smart Licensing supports these license types:

Essential License

The Essentials license is the base and subscriber-session licensing tier for active sessions that meet all essentials conditions. It applies to active subscriber sessions that do not use redundancy, such as SRG or CPGR, and are not LNS sessions.

The Essentials tier uses entitlement tags to report subscriber-session-based license consumption to Smart Licensing. Per 1000 active subscriber sessions, one Essentials license is consumed.

Entitlement Tags

Entitlement tag is a part of the software that identifies the features in an image that are being used. These tags underlay the communication on usage and entitlements of software products that are installed on devices. The entitlement tag maps to both the product IDs (PID) license and the software image. Every Smart-enabled PID contains one or more entitlement tags.

These entitlement tags exist for the cnBNG CP.

Tag Name	Smart Tag	Condition for License Calculation
CNBNG_CP_ESS_RTU	`regid.2021-07.com.cisco.CNBNG_CP_ESS_RTU,1.0_cb8f62c6-1acf-4e65-9d8c-35b12acfbaaf`	Per 1000 session only one (1) license is consumed for each entitlement tag. The sessions are rounded off when calculating the number of licenses required: For Example: If num of session count is <= 1000, one license is consumed. If number of sessions are: 1001 to 2000, two licenses are consumed; similarly, from 2001 to 3000, three licenses are consumed and so on. **Note** The number of sessions (session count) is derived from CDL.
CNBNG_CP_ESS_SIA	`regid.2021-07.com.cisco.CNBNG_CP_ESS_SIA,1.0_c40ac644-7a76-41b8-a3dc-b66159c0f1ae`	

Note

- The license information is retained during software upgrades and rollback.
- If the entitlements service has no entitlements information, the BNG Ops Center does not send any request to the Smart License server and does not install any license.
- If the entitlement is not reported to the BNG Ops Center, then that entitlement is not listed under the **show license *** commands.
- If the given entitlement is reported at least once (license consumed for an entitlement) to the BNG Ops Center, then that entitlement is shown under the **show license *** commands with count zero (0) even though the license is not used in the future.

Advantage License

The Advantage license is the subscriber-session licensing tier for active sessions with redundancy, such as SRG or CPGR, and to LNS sessions.

Advantage Tags

These advantage tags exist for the cnBNG CP.

Tag Name	Smart Tag	Condition for License Calculation
CNBNG_CP_ADV_RTU	regid.2022-03.com.cisco.CNBNG_CP_ADN_RTU, 1.0_977ca1d3-6e87-4598-9960-779dbff12b53	Per 1000 active subscriber sessions, one Advantage license is consumed when the session includes CPGR deployments, SRG sessions, and LNS sessions.
CNBNG_CP_ADV_SIA	regid.2022-03.com.cisco.CNBNG_CP_ADN_SIA, 1.0_dee6cc1f-1a4e-49a5-87f9-37d0974cc7ba	

Configuring Smart Software Licensing for cnBNG CP

Smart Software Licensing for cnBNG CP can be configured after a new cnBNG CP installation.

Users with Access to CSC, on page 106

Users without Access to CSC, on page 110

Users with Access to CSC

This section describes how to configure Smart Licensing if you have access to CSC portal from your environment.

Setting Up the Product and Entitlement in CSC

Before you begin, you need to set up your product and entitlement in the CSC. To set up your product and entitlement:

1. Log on to your CSC account.
2. Click **Add Product** and enter the following details:
   - **Product name**–Specify the name of the deployed product. For example, cnBNG CP.
   - **Primary PM CEC ID**–Specify the primary Project Manager's CEC ID for the deployed product.
   - **Dev Manager CEC ID**–Specify the Development Manager's CEC ID for the deployed product.
   - **Description** (Optional)–Specify a brief description of the deployed product.
   - **Product Type**–Specify the product type.
   - **Software ID Tag**–Specify the software ID Tag provided by the Cisco Accounts team.
3. Click **Create**.
4. Select your product from the **Product/Entitlement Setup** grid.
5. From the **Entitlement** drop-down list, select **Create New Entitlement**.
6. Select **New Entitlement** in **Add Entitlement** and enter the following details:
   - **Entitlement Name**–Specify the license entitlement name. For example, CNBNG_CP_SYS_RTU.
   - **Description** (Optional)–Enter a brief description about the license entitlement.

- **Entitlement Tag**–Specify the entitlement tag provided by the Cisco Accounts team.
- **Entitlement Type**–Specify the type of license entitlement.
- **Vendor String**–Specify the vendor name.

7. Click **Entitlement Allocation**.
8. Click **Add Entitlement Allocation**.
9. In **New License Allocation**, enter the following details:
   - **Product** – Select your product from the drop-down list.
   - **Entitlement** – Select your entitlement from the drop-down list.
10. Click **Continue**.
11. In **New License Allocation** window, enter the following details:
    - **Quantity**–Specify the number of licenses.
    - **License Type**–Specify the type of license.
    - **Expiring Date**–Specify the date of expiry for the license purchased.
12. Click **Create**.
13. Verify the status of Smart Licensing by using the following command.

    `show license status`

    Example:

```
bng# show license status
Smart Licensing is ENABLED

Registration:
  Status: UNREGISTERED
  Export-Controlled Functionality: Not Allowed

License Authorization:
  Status: No licenses in use
  Last Communication Attempt: NONE

License Conversion:
 Automatic Conversion Enabled: true
 Status: NOT STARTED

Utility:
  Status: DISABLED

Transport:
  Type: Smart Transport
  Registration URL: null
  Utility URL: null
bng#
```

Registering Smart Licensing

You must register the product entitled to the license with CSC. To register, you must generate an ID token from CSC.

1. Log on to your CSC account.
2. Click **General > New Token** and enter the following details:
   - **Description**–Provide a brief description about the ID token.
   - **Expires After**–Specify the number of days for the token to expire.
   - **Max. Number of Uses**–Specify the maximum number of uses.
3. Click **Create Token**.
4. Select **New ID token** in **Product Instance Registration Token**.
5. Click **Actions > Copy**.
6. Log on to BNG Ops Center CLI and paste the **ID token** by using the following command.

   `license smart register` *`idtoken`*

   Example:

   ```
   bng# license smart register
   Value for 'idtoken' (<string>): MTI2Y2FlNTAtOThkMi00YTAxLWE4M2QtOTNhNzNjNjY4ZmFiLTE2MTc4N
   Tky%0AMTA5MDh8ck1jUHNwc3k1ZC9nWFFCSnVEcUp4QU1jTFoxOGxDTU5kQ3lpa25E%0Ab04wST0%3D%0A
   bng#
   ```

7. Verify the Smart Licensing status by using the following command.

   `show license status`

   Example:

   ```
   show license status
   Wed Jan  12 14:31:16.858 UTC+00:00

   Smart Licensing is ENABLED
   License Reservation is ENABLED

   Registration:
     Status: REGISTERED
     Smart Account: BU Production Test 1
     Virtual Account: CNBNG_QA_PROD_TEST
     Export-Controlled Functionality: Allowed
     Initial Registration: SUCCEEDED on Jan 12 14:31:35 2022 GMT
     Last Renewal Attempt: SUCCEEDED on Jan 12 14:31:35 2022 GMT
     Next Renewal Attempt: Jul 11 14:31:35 2022 GMT
     Registration Expires: Jan 12 14:26:55 2023 GMT

   License Authorization:
     Status: AUTHORIZED on Jan 12 14:31:40 2022 GMT
     Last Communication Attempt: SUCCEEDED on Jan 12 14:31:40 2022 GMT
     Next Communication Attempt: Feb 11 14:31:40 2022 GMT
     Communication Deadline: Apr 12 14:27:00 2022 GMT

   License Conversion:
    Automatic Conversion Enabled: true
    Status: NOT STARTED
   ```

```
Utility:
  Status: DISABLED

Transport:
  Type: Smart Transport
  Registration URL: null
  Utility URL: null
```

NOTES:

- **license smart register** : Register Smart Licensing with CSC.
- *idtoken* : Specify the ID token generated from CSC.

Deregistering Smart Licensing

To deregister Smart Licensing:

1. Log on to BNG Ops Center CLI and use the following command.

   `license smart deregister`

2. Verify the Smart Licensing status by using the following command:

   `show license all`

   Example:

```
bng# show license all
Mon Dec  13 05:29:03.370 UTC+00:00

Smart Licensing Status
======================
Smart Licensing is ENABLED
License Reservation is ENABLED

Registration:
  Status: UNREGISTERED
  Export-Controlled Functionality: Not Allowed

License Authorization:
  Status: EVAL MODE
  Evaluation Period Remaining: 86 days, 13 hr, 52 min, 39 sec
  Last Communication Attempt: NONE

License Conversion:
 Automatic Conversion Enabled: true
 Status: NOT STARTED

Utility:
  Status: DISABLED
Transport:
   Type: Smart Transport
   Registration URL: null
   Utility URL: null

Evaluation Period:
  Evaluation Mode: In Use
  Evaluation Period Remaining: 86 days, 13 hr, 52 min, 39 sec

License Usage
=============
License Authorization Status: EVALUATION MODE
```

```
  Evaluation Period Remaining: 86 days, 13 hr, 52 min, 39 sec

 (CNBNG_CP_SYS_RTU)
  Description: <empty>
  Count: 1
  Version: 1.0
  Status: EVAL MODE
  Export status: NOT RESTRICTED
  Feature Name: <empty>
  Feature Description: <empty>

 (CNBNG_CP_SYS_SIA)
  Description: <empty>
  Count: 1
  Version: 1.0
  Status: EVAL MODE
  Export status: NOT RESTRICTED
  Feature Name: <empty>
  Feature Description: <empty>

Product Information
===================
UDI: PID:cnBNG,SN:QHJZRXY-IRNN3JA

Agent Version
=============
Smart Agent for Licensing: 3.1.11
bng#
```

NOTES:

- **license smart deregister** : Deregisters Smart Licensing from CSC.

Users without Access to CSC

The Smart License Reservation feature – Perpetual Reservation – is reserved for customers without access to CSC from their internal environments. With this feature, Cisco allows customers to reserve licenses from their virtual account and tie them to their devices Unique Device Identifier (UDI). Smart License Reservation enables customers to use their devices with reserved licenses in a disconnected mode.

The subsequent sections describe the procedure involved in reserving Smart License for users without access to CSC from their internal environment.

Enabling Smart License Reservation

To enable Smart License reservation through BNG Ops Center CLI:

Log on to BNG Ops Center CLI and use the following configuration.

```
config
   license smart reservation
   commit
   exit
```

NOTES:

license smart reservation : Enable license reservation.

Generating Smart License Reservation Request Code

Note Before generating the Smart License reservation request code, complete the "Enabling Smart License Reservation" procedure.

To generate the Smart License reservation request code:

1. Enter the following command.

   `license smart reservation request`

   Example:

   ```
   bng# license smart reservation request
   reservation-request-code CJ-ZBNG:6GKJ2OA-NMUWA7Y-Ai75GxtBs-3B
   ```

   NOTES:

   - **license smart reservation request** : Generate the license reservation request code.

Important You must copy the generated license request code from the BNG Ops Center CLI.

Generating an Authorization Code from CSC

To generate an authorization code from CSC using the license reservation request code:

1. Log on to your CSC account.
2. Click **License Reservation** from the Licenses tab.
3. Copy the request code from the BNG Ops Center CLI and paste the request code in the **Reservation Request Code** text-box.
4. Click **Reserve a Specific License** option and select the required licenses and counts to be reserved.
5. Review your selection.
6. Click **Generate Authorization Code**.
7. The authorization code is generated and displayed on-screen. Either click **Copy to Clipboard** or **Download as File** to download the authorization code.
8. Click **Close**.

Reserving Smart Licensing

There are two methods available to reserve the Smart License:

- Key-based: Using the copied clipboard content of the authorization code directly from the CSC.
- URL-based: Using the downloaded file containing the authorization code from CSC, saved on the local server.

To reserve Smart License for the deployed product:

1. Log on to BNG Ops Center CLI and enter the following command.

   Key-based:

   `license smart reservation install key` *`authorization_code`*

   Example:

   ```
   bng# license smart reservation install key
   Value for 'key' (<string>): <specificPLR><authorizationCode><flag>A</flag><version>
   C</version><piid>73ce7376-4631-45db-bccc-f8b4d31acd33</piid><timestamp>1642000062377
   </timestamp><entitlements><entitlement><tag>regid.2021-07.com.cisco.CNBNG_CP_ESS_RTU,
   1.0_cb8f62c6-1acf-4e65-9d8c-35b12acfbaaf</tag><count>1</count><startDate>
   2021-Dec-01 UTC</startDate><endDate>2022-May-30
   UTC</endDate><licenseType>TERM</licenseType><
   displayName>CNBNG Control Plane - Essentials - RTU</displayName><tagDescription>
   CNBNG Control Plane - Essentials Tier - RTU</tagDescription><subscriptionID>
   </subscriptionID></entitlement><entitlement><tag>regid.2021-07.com.cisco.
   CNBNG_CP_ESS_SIA,1.0_c40ac644-7a76-41b8-a3dc-b66159c0f1ae</tag><count>1</count>
   <startDate>2021-Dec-01 UTC</startDate><endDate>2022-May-30 UTC</endDate>
   <licenseType>TERM</licenseType><displayName>CNBNG Control Plane - Essentials -
   SIA</displayName><tagDescription>CNBNG Control Plane - Essentials Subscription -
   SIA</tagDescription><subscriptionID></subscriptionID></entitlement><entitlement>
   <tag>regid.2021-07.com.cisco.CNBNG_CP_SYS_RTU,1.0_2e15ed3f-929d-47b5-8495-c96531c416b8
   </tag><count>1</count><startDate>2021-Dec-01 UTC</startDate><endDate>2022-May-30 UTC
   </endDate><licenseType>TERM</licenseType><displayName>CNBNG Control Plane - System -
   RTU</displayName><tagDescription>CNBNG Control Plane - System - RTU</tagDescription>
   <subscriptionID></subscriptionID></entitlement><entitlement><tag>regid.2021-07.com.cisco.
   CNBNG_CP_SYS_SIA,1.0_f6cab505-581f-41ec-8170-3d8c325f7f73</tag><count>1</count><startDate>
   2021-Dec-01 UTC</startDate><endDate>2022-May-30
   UTC</endDate><licenseType>TERM</licenseType>
   <displayName>CNBNG Control Plane - System - SIA</displayName><tagDescription>
   CNBNG Control Plane - System Subscription - SIA</tagDescription><subscriptionID>
   </subscriptionID></entitlement></entitlements></authorizationCode><signature>
   MEYCIQC3VB12XNo+gi00fE23Pqd1GZ67MOxzkl+DCVPhS0LonQIhANP27J3skpMnd3O
   Qwzm82knoUBFM8Fk0yf2llqQvuByi</signature><udi>P:cnBNG,
   S:6OUP5ZY-LMXHB2A</udi></specificPLR>
   ```

   URL-based:

   `license smart reservation install url { path` *`httpPath`*
   `[ username` *`username`* **`| password`** *`password`* **`] }`**

   Example:

   ```
   bng# license smart reservation install url { username cnbng password **** path http://
   10.105.254.55:8000/AuthorizationCode_SN_6OUP5ZY-LMXHB2A.txt }
   ```

2. Verify the smart licensing status by using the following command.

   `show license all`

   Example:

   ```
   bng# show license all


   Smart Licensing Status
   ======================
   Smart Licensing is ENABLED
   License Reservation is ENABLED

   Registration:
     Status: REGISTERED - SPECIFIC LICENSE RESERVATION
   ```

```
  Export-Controlled Functionality: Allowed
  Initial Registration: SUCCEEDED on Thu Jan 20 10:02:40 GMT 2022
  Last Renewal Attempt: None

License Authorization:
  Status: AUTHORIZED - RESERVED on Thu Jan 20 10:02:40 GMT 2022

Utility:
  Status: DISABLED

Transport:
  Type: Smart Transport
  Registration URL: null
  Utility URL: null

Evaluation Period:
  Evaluation Mode: Not In Use
  Evaluation Period Remaining: 83 days, 6 hr, 14 min, 43 sec

License Usage
=============
License Authorization Status:
  Status: AUTHORIZED - RESERVED on Thu Jan 20 10:02:40 GMT 2022
  Last Communication Attempt: SUCCEEDED on Jan 20 10:02:40 2022 GMT
  Next Communication Attempt: NONE
  Communication Deadline: NONE

CNBNG Control Plane - System - RTU (CNBNG_CP_SYS_RTU)
  Description: CNBNG Control Plane - System - RTU
  Count: 1
  Version: 1.0
  Status: AUTHORIZED
  Export status: NOT RESTRICTED
  Feature Name: <empty>
  Feature Description: <empty>
  Reservation:
    Reservation Status: SPECIFIC INSTALLED
    Total Reserved Count: 1
    Term expiration: 2022-May-30 GMT

CNBNG Control Plane - System - SIA (CNBNG_CP_SYS_SIA)
  Description: CNBNG Control Plane - System Subscription - SIA
  Count: 1
  Version: 1.0
  Status: AUTHORIZED
  Export status: NOT RESTRICTED
  Feature Name: <empty>
  Feature Description: <empty>
  Reservation:
    Reservation Status: SPECIFIC INSTALLED
    Total Reserved Count: 1
    Term expiration: 2022-May-30 GMT

CNBNG Control Plane - Essentials - RTU (CNBNG_CP_ESS_RTU)
  Description: CNBNG Control Plane - Essentials Tier - RTU
  Count: 1
  Version: 1.0
  Status: AUTHORIZED
  Export status: NOT RESTRICTED
  Feature Name: <empty>
  Feature Description: <empty>
  Reservation:
    Reservation Status: SPECIFIC INSTALLED
    Total Reserved Count: 10
```

```
    Term expiration: 2022-May-30 GMT

CNBNG Control Plane - Essentials - SIA (CNBNG_CP_ESS_SIA)
  Description: CNBNG Control Plane - Essentials Subscription - SIA
  Count: 1
  Version: 1.0
  Status: AUTHORIZED
  Export status: NOT RESTRICTED
  Feature Name: <empty>
  Feature Description: <empty>
  Reservation:
    Reservation Status: SPECIFIC INSTALLED
    Total Reserved Count: 10
    Term expiration: 2022-May-30 GMT

Product Information
===================
UDI: PID:cnBNG,SN:3WI5UXQ-5XPKM6A

Agent Version
=============
Smart Agent for Licensing: 3.1.11
```

Returning the Reserved License

To return the reserved license to CSC if required, generate a return code at the BNG Ops Center.

Note If there is an issue with the return code generation, open a case with the Cisco Technical Assistance Center.

To return the reserved license:

1. When the license reservation authorization code is installed in the BNG Ops Center:
   a. Log on to the BNG Ops Center CLI and use the following command.

      `license smart reservation return`

      Example:

      ```
      bng# license smart reservation return
      reservation-return-code CJ6m3k-RAvu6b-hMNmwf-mrdcko-NoSwKL-tF7orz-9aNtEu-yVjGAm-D6j
      bng#
      ```

   b. Copy the license reservation return code generated in BNG Ops Center CLI.
   c. Log on to your CSC account.
   d. Select your product instance from the list in the Product Instances tab.
   e. Click **Actions** > **Remove**.
   f. Paste the license reservation return code in **Return Code** text-box.
   g. Select **Remove Product Instance**.

   NOTES:

   - **license smart reservation return** : Return a reserved Smart License.

2. When the license reservation authorization code is not installed in the BNG Ops Center.
   a. Log on to the BNG Ops Center CLI and use the following command to generate the return code.

      license smart reservation return authorization
      authorization_code

      Paste the license reservation authorization code generated in CSC to generate the return code.
   b. Log on to your CSC account.
   c. Select your product instance from the list in the Product Instances tab.
   d. Click **Actions > Remove**.
   e. Paste the license reservation return code in **Return Code** text-box.
   f. Select **Remove Product Instance**.
3. Verify the smart licensing status by using the following command.

   show license all

   Example:

```
bng# show license all
Mon Dec  13 05:29:03.370 UTC+00:00

Smart Licensing Status
======================
Smart Licensing is ENABLED
License Reservation is ENABLED

Registration:
  Status: UNREGISTERED
  Export-Controlled Functionality: Not Allowed

License Authorization:
  Status: EVAL MODE
  Evaluation Period Remaining: 86 days, 13 hr, 52 min, 39 sec
  Last Communication Attempt: NONE

License Conversion:
 Automatic Conversion Enabled: true
 Status: NOT STARTED

Utility:
  Status: DISABLED

Transport:
  Type: CALLHOME

Evaluation Period:
  Evaluation Mode: In Use
  Evaluation Period Remaining: 86 days, 13 hr, 52 min, 39 sec

License Usage
=============
License Authorization Status: EVALUATION MODE
  Evaluation Period Remaining: 86 days, 13 hr, 52 min, 39 sec

 (CNBNG_CP_SYS_RTU)
  Description: <empty>
  Count: 1
```

```
  Version: 1.0
  Status: EVAL MODE
  Export status: NOT RESTRICTED
  Feature Name: <empty>
  Feature Description: <empty>

 (CNBNG_CP_SYS_SIA)
  Description: <empty>
  Count: 1
  Version: 1.0
  Status: EVAL MODE
  Export status: NOT RESTRICTED
  Feature Name: <empty>
  Feature Description: <empty>

Product Information
===================
UDI: PID:cnBNG,SN:QHJZRXY-IRNN3JA

Agent Version
=============
Smart Agent for Licensing: 3.1.11
```

Canceling License Reservation Request

To cancel a license reservation request through the BNG Ops Center CLI:

Log on to the BNG Ops Center CLI and use the following command.

`license smart reservation cancel`

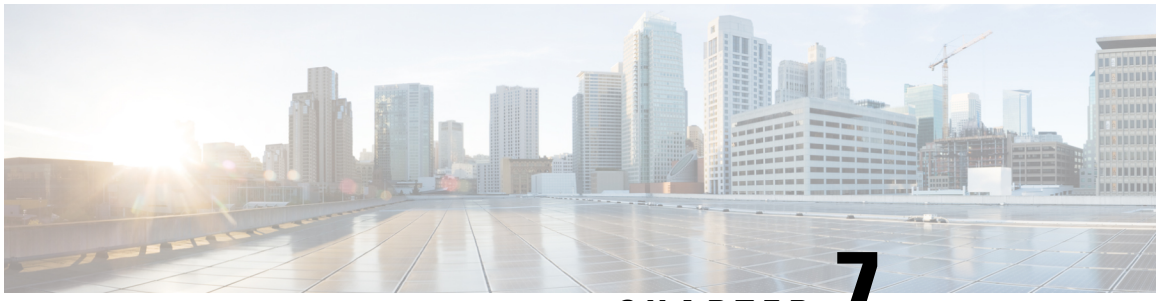
Monitoring and Troubleshooting Smart Licensing

You can use the following show commands to view Smart Licensing related information in the BNG Ops Center.

`show license [ all | UDI | displaylevel | reservation | smart | status | summary | tech-support | usage ]`

NOTES:

- **all** –Displays an overview of Smart Licensing information that includes license status, usage, product information and Smart Agent version.
- **UDI** –Displays Unique Device Identifiers (UDI) details.
- **displaylevel** –Depth to display information.
- **reservation** –Displays Smart Licensing reservation information.
- **smart** –Displays Smart Licensing information.
- **status** –Displays the overall status of Smart Licensing.
- **summary** –Displays a summary of Smart Licensing.
- **tech-support** –Displays Smart Licensing debugging information.
- **usage** –Displays the license usage information for all the entitlements that are currently in use.



CHAPTER 7

Alarm Support

- Feature Summary and Revision History, on page 117
- Feature Description, on page 117
- Configuring Alarm Support, on page 125

Feature Summary and Revision History

Summary Data

Table 15: Summary Data

Applicable Product(s) or Functional Area	cnBNG
Applicable Platform(s)	SMI
Feature Default Setting	Disabled - Configuration Required
Related Documentation	Not Applicable

Revision History

Table 16: Revision History

Revision Details	Release
First introduced.	2022.02.0

Feature Description

When an anomaly is detected, the system generates a notification called an alarm or alert. The system triggers an alarm or alert when the statistics crosses the specified threshold. The Cloud Native BNG Control Plane uses the Common Execution Environment (CEE) infrastructure to generate alarms and SNMP traps.

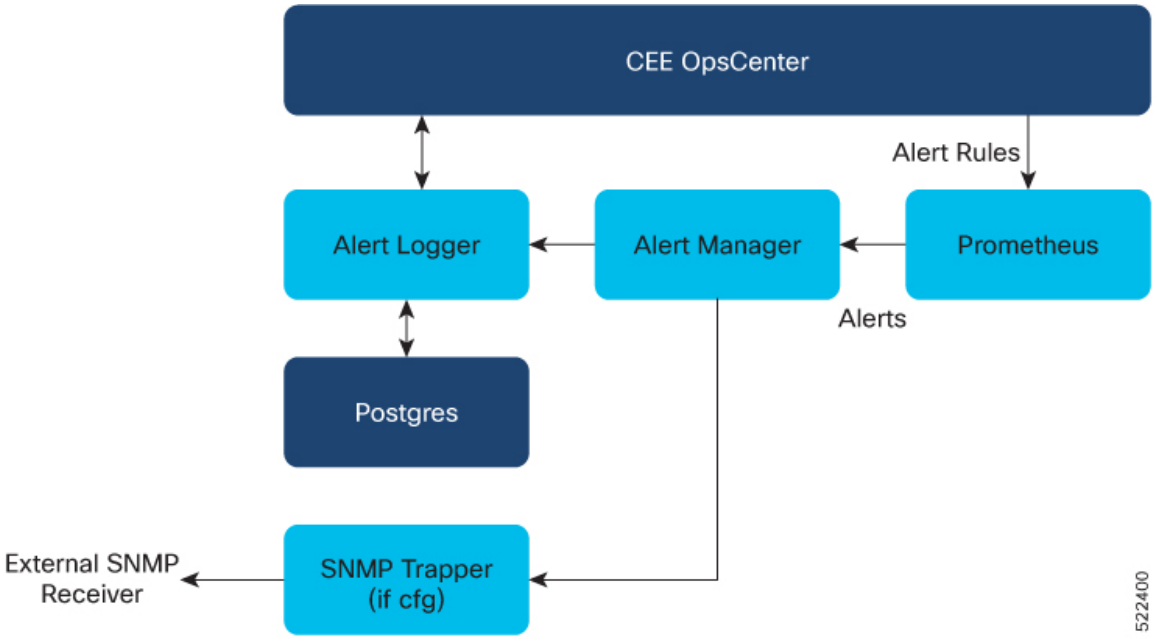
The Common Execution Environment (CEE) uses the Prometheus Alert Manager for alerting operations. The CEE YANG model - either through CLI or API - allows users to view the active alerts, silenced alerts, and alert history. Also, the applications can call the alert API directly to add or clear alerts. The Prometheus Alert Manager API (v2) is the standard API used.

The Prometheus Alerts Manager includes the following options:

- **Defining Alert Rules**: This option defines the types of alerts that the Alert Manager should trigger. Use the Prometheus Query Language (PromQL) to define the alerts.
- **Defining Alert Routing**: This option defines the action the Alert Manager should take after receiving the alerts. At present, the SNMP Trapper is supported as the outbound alerting. Also, the CEE provides an Alert Logger for storing the generated alerts.

The CNEE provides a set of predefined alerting rules regarding system health and Ops Center monitoring. For more details, see the "Alerts Reference" chapter in the *Ultra Cloud Core Subscriber Microservices Infrastructure Operations Guide*

The following figure depicts the components involved in the alerting mechanism.



Supported Alarm Categories

The cnBNG CP supports the following alarm categories.

- **CP-UP Connectivity**—Alarms are generated if Control Plane (CP) - User Plane (UP) association fails, active, or inactive. Expressions can be formulated on CP defined UPF and Packet Forwarding Control Protocol (PFCP) metrics.
- **cnBNG Session**—Alarms can be generated if session bring-up or bring-down success rate is less than the specificied threshold, drop rate, and if subscriber limit is crossed. Expressions can be formulated on Session Manager (SM) and First Sign of Life (FSOL) metrics.

- **Accounting**—Alarms can be generated if accounting start, interim, and stop success rate is less than the specified certain threshold and so on. Expressions can be formulated on accounting metrics.
- **Radius** —Alarms can be generated for RADIUS server for the following events:
  - Active or Inactive server state
  - Statistics success, failure, or reject rates for Authorization and Accounting based on threshold
  - Change of Authorization (CoA) success or failure rates.

  Expressions can be formulated on RADIUS metrics.
- **IP Pool** —Alarms can be generated if IP pool allocation reaches the specified threshold. Expressions can be formulated on IPAM metrics.

Alert Configuration Recommendations

Based on alarm categories, the following alert configurations are recommended.



Note
- The Threshold field is configurable as per requirement.
- The *interval-seconds* and *duration* can vary based on requirements.

Application-based Alerts

Configure the following alerts to detect an application anomaly and trigger the alert or alarm.



Note These alerts are critical, therefore, it is recommended that these alerts are configured.

RADIUS Authorization or Accounting Status

Use the following commands to configure an alert when the RADIUS authorization or accounting server is down.

```
alerts rules group RadiusEP
       interval-seconds 300
       rule Auth_Radius_Server_Down
       expression “sum by (namespace,
rad_svr_ip,rad_svr_port)(radius_server_status{rad_svr_port_type=\"Auth\"} < 1)”
       duration   5m
       severity   major
       type       “Processing Error Alarm”
       annotation summary
       value “Auth Radius Server: {{ $labels.rad_svr_ip}}, Port: {{ $labels.rad_svr_port
}} in namespace: {{ $labels.namespace }} is DOWN for more than 5min.”
       exit
       rule Acct_Radius_Server_Down
       expression “sum by (namespace,
radSvrIP,rad_svr_port)(radius_server_status{rad_svr_portType=\"Acct\"} < 1)”
       duration   5m
```

```
       severity   major
       type       "Processing Error Alarm"
       annotation summary
       value "Acct Radius Server: {{ $labels.rad_svr_ip}}, Port: {{ $labels.rad_svr_port
}} in namespace: {{ $labels.namespace }} is DOWN for more than 5min."
       exit
 exit
```

User Plane Function Status

Use the following commands to configure an alert related to User Plane (UP) to Control Plane (CP) connectivity.

```
alerts rules group CpUpAssociation
       interval-seconds 300
       rule CpUpConnectionStatus
       expression "sum by (namespace, upf)(upf_status{status=~\"InActive\"}) > 0 )"
       duration   1m
       severity   major
       type       "Processing Error Alarm"
       annotation summary
       value "Upf {{ $labels.namespace }}/{{ $labels.upf }} is inactive for 1m"
       exit
 exit
```

Subscriber Limit Threshold

Use the following commands to configure an alert when the session count crosses the specified threshold.

```
alerts rules group BngSession
       interval-seconds 300
       rule BngSubscriberLimit
       expression "sum by (namespace) ((avg(db_records_total{session_type=\"SM:PPPOE\"})
OR on() vector(0)) + (avg(db_records_total{session_type=\"SM:DHCP\"})  OR on() vector(0))
+ (avg(db_records_total{session_type=\"SM:LNS\"})  OR on() vector(0)) +
(avg(db_records_total{session_type=\"SM:LAC\"})  OR on() vector(0))) > THRESHOLD"
       severity   critical
       type       "Communications Alarm"
       annotation summary
       value "This alert is fired when session count rises above threshold."
exit
 exit
```

System Overload Status

Use following commands to configure an alert if or when the system overloads.

```
alerts rules group BngSystemStatus
   interval-seconds 300
   rule BngOverload
   expression "sum by(component,level) (system_overload_status{level=~\"Critical|Crash\"})
 == true"
   duration   5m
   severity critical
   type "Communications Alarm"
   annotation summary
  value "This alert is fired when there is system overload as component {{ $labels.component
 }} has health level is {{ $labels.level }}."
   exit
exit
```

IP Pool Consumption

Use the following commands to configure IP pool consumption alerts.

```
alerts rules group IPPool
   interval-seconds 300
   rule IPPoolConsumption
   expression "sum by (namespace,pool,addressType)(ipam_address_allocations_current)/sum
by (namespace,pool,address_type)(IPAM_address_pool_total) > THRESHOLD"
   duration 1m
   severity major
   type "Processing Error Alarm"
   annotation summary
   value "Pool: {{ $labels.pool }} AddressType: {{$labels.address_type}} in Namespace: {{
$labels.namespace }} has reached  THRESHOLD % of utilization"
   exit
exit
```

PPPoE Session Limit Threshold

Use the following commands to configure an alert when the PPPoE session limit crosses the specified threshold.

```
alerts rules group PPPoESessionLimit
 rule PPPoESessionLimit
  expression
"((pppoe_session_limit_total{session_limit_count=\"SessionRejected\",session_limit_type=\"SessionMaxLimit\"}
 unless
pppoe_session_limit_total{session_limit_count=\"SessionRejected\",session_limit_type=\"SessionMaxLimit\"}
 offset 1m) OR
(increase(pppoe_session_limit_total{session_limit_count=\"SessionRejected\",session_limit_type=\"SessionMaxLimit\"}[1m])
 )) > 0"
  severity   critical
  type       "Communications Alarm"
  annotation summary
   value "PPPoE session limit crossed in last 1min."
  exit
 exit
exit
```

L2TP Session Limit Threshold

Use the following commands to configure an alert when the L2TP session limit crosses the specified threshold.

```
alerts rules group L2TPSession
 rule SessionLimit
  expression "sum by (remote_hostname,upf_name)((l2tp_session_limit_total unless
l2tp_session_limit_total offset 1m) OR (increase(l2tp_session_limit_total[1m]) )) > 0"
  severity   critical
  type       "Communications Alarm"
  annotation summary
   value "Session Limit crossed for Tunnel: Routername {{$labels.upf_name}} Remote
{{$labels.remote_hostname}} !!!"
  exit
 exit
exit
```

Use-Case Based Alerts

Configure the following alerts based on requirements.

RADIUS Authorization Success Rate

Use the following commands to configure RADIUS authorization success rate alerts.

```
alerts rules group RadiusEP
       interval-seconds 300
       rule RadiusAuthSuccessRate
```

```
       expression "sum by (namespace)(increase(radius_requests_statistics{
rad_msg_code=\"AaaAuthReq\",rad_pkt_type=\"Rx\",rad_result=\"Success\"}[5m]))/sum by
(namespace)(increase(radius_requests_statistics{
rad_msg_code=\"AaaAuthReq\",rad_pkt_type=\"Tx\"}[5m])) < THRESHOLD"
       severity   major
       type       "Communications Alarm"
       annotation summary
       value "This alert is fired when the percentage of successful Radius Authentication
responses received is lesser than threshold"
       exit
 exit
```

RADIUS Accounting Success Rate

Use the following commands to configure RADIUS accounting response success rate alerts.

```
alerts rules group RadiusEP
       interval-seconds 300
       rule RadiusAcctSuccessRate
       expression "sum by (namespace)(increase(radius_requests_statistics{
rad_msg_code=\"AaaAcctReq\",rad_pkt_type=\"Rx\",rad_result=\"Success\"}[5m]))/sum by
(namespace)(increase(radius_requests_statistics{
rad_msg_code=\"AaaAcctReq\",rad_pkt_type=\"Tx\"}[5m])) < THRESHOLD"
       severity   major
       type       "Communications Alarm"
       annotation summary
       value "This alert is fired when the percentage of successful Radius Accounting
responses received is lesser than threshold"
       exit
 exit
```

Radius CoA Success Rate

Use the following commands to configure RADIUS Change of Authorization (CoA) success rate alerts.

```
alerts rules group RadiusEP
       interval-seconds 300
       rule RadiusCoaSuccessRate
       expression "sum by (namespace)(increase(radius_coa_dm_requests_statistics{
rad_msg_code=\"CoAACK\",rad_pkt_type=\"Tx\",rad_result=\"Success\"}[5m]))/sum by
(namespace)(increase(radius_coa_dm_requests_statistics{
rad_msg_code=\"CoARequest\",rad_pkt_type=\"Rx\"}[5m])) < THRESHOLD"
       severity   major
       type       "Communications Alarm"
       annotation summary
       value "This alert is fired when the percentage of successful Coa Ack received is
lesser than threshold"
       exit
 exit
```

Accounting Start Success Rate

Use the following commands to configure accounting start success rate alerts.

```
alerts rules group Accounting
       interval-seconds 300
       rule AcctStartSuccessRate
       expression "sum by (namespace)(increase(accounting_message_total{
acct_type=\"Start\",status=\"Success\"}[5m]))/sum by
(namespace)(increase(accounting_message_total{ acct_type=\"Start\",status=\"Attempt\"}[5m]))
 < THRESHOLD"
       severity   major
       type       "Processing Error Alarm"
```

```
       annotation summary
      value "This alert is fired when the percentage of successful Accounting Start Responses
 received is lesser than threshold"
       exit
 exit
```

Accounting Interim Success Rate

Use the following commands to configure accounting interim success rate alerts.

```
alerts rules group Accounting
       interval-seconds 300
       rule AcctInterimSuccessRate
       expression "sum by (namespace)(increase(accounting_message_total{
acct_type=\"Interim\",status=\"Success\"}[5m]))/sum by
(namespace)(increase(accounting_message_total{ acct_type=\"Interim\",status=\"Attempt\"}[5m]))
 < THRESHOLD"
       severity   major
       type       "Processing Error Alarm"
       annotation summary
       value "This alert is fired when the percentage of successful Accounting Interim
Responses received is lesser than threshold"
       exit
 exit
```

Accounting Stop Success Rate

Use the following commands to configure accounting stop success rate alerts.

```
alerts rules group Accounting
       interval-seconds 300
       rule AcctStopSuccessRate
       expression "sum by (namespace)(increase(accounting_message_total{
acct_type=\"Stop\",status=\"Success\"}[5m]))/sum by
(namespace)(increase(accounting_message_total{ acct_type=\"Stop\",status=\"Attempt\"}[5m]))
 < THRESHOLD"
       severity   major
       type       "Processing Error Alarm"
       annotation summary
      value "This alert is fired when the percentage of successful Accounting Stop Responses
 received is lesser than threshold"
       exit
       exit
```

N4 Session Creation Success Rate

Use the following commands to configure N4 session creation success rate alerts.

```
alerts rules group BngSession
       interval-seconds 300
       rule SessionCreateSuccessRate
       expression "sum by(namespace,upf_name)
(increase(bng_proto_udp_total{message_name=\"n4_session_establishment_res\",message_direction=\"inbound\",
 status=\"accepted\"}[5m]))/sum by (namespace,upf_name) (increase(bng_proto_udp_total{
message_name=\"n4_session_establishment_req\", message_direction=\"outbound\",
transport_type=\"origin\",status=\"accepted\"}[5m])) < THRESHOLD"
       severity   major
       type       "Communications Alarm"
       annotation summary
      value "This alert is fired when the percentage of successful Session Creation Responses
 received is lesser than expected threshold for upf: {{$labels.upf}}"
        exit
exit
```

N4 Session Update Success Rate

Use the following commands to configure N4 session update success rate alerts.

```
alerts rules group BngSession
       interval-seconds 300
       rule SessionUpdateSuccessRate
       expression "sum by(namespace,upf)
(increase(bng_proto_udp_total{message_name=\"n4_session_modification_res\",message_direction=\"inbound\",
 status=\"accepted\"}[5m]))/sum by (namespace,upf) (increase(bng_proto_udp_total{
message_name=\"n4_session_modification_req\",  message_direction=\"outbound\",
transport_type=\"origin\",status=\"accepted\"}[5m])) < THRESHOLD"
       severity   major
       type       "Communications Alarm"
       annotation summary
      value "This alert is fired when the percentage of successful Session Update Responses
 received is lesser than expected threshold for upf: {{$labels.upf_name}}"
        exit
exit
```

N4 Session Release Success Rate

Use the following commands to configure N4 session release success rate alerts.

```
alerts rules group BngSession
       interval-seconds 300
       rule SessionReleaseSuccessRate
       expression "sum by (namespace,upf_name)
(increase(bng_proto_udp_total{message_name=\"n4_session_deletion_res\",message_direction=\"inbound\",
 status=\"accepted\"}[5m]))/sum by (namespace,upf_name)(increase(bng_proto_udp_total{
message_name=\"n4_session_deletion_req\",  message_direction=\"outbound\",
transport_type=\"origin\",status=\"accepted\"}[5m])) < THRESHOLD"
       severity   major
       type       "Communications Alarm"
       annotation summary
      value "This alert is fired when the percentage of successful Session Release Responses
 received is lesser than expected threshold for upf: {{$labels.upf_name}}"
        exit
exit
```

N4 Session Request Timeouts

Use the following commands to configure alerts to calculate the rate of N4 session requests that timeout awaiting response from the UP.

```
alerts rules group BngSession
       interval-seconds 300
       rule N4SessionReqTimeouts
       expression "sum by (namespace,
upf_name)(increase(bng_proto_udp_total{message_name=~\"n4_session_establishment_req|n4_session_modification_req|n4_session_deletion_req\",message_direction=\"outbound\",
 status=\"Timeout\"}[15m]))/sum by (namespace, upf_name)(increase(bng_proto_udp_total{
message_name=~\"n4_session_establishment_req|n4_session_modification_req|n4_session_deletion_req\",
  message_direction=\"outbound\", transport_type=\"origin\"}[15m])) > THRESHOLD"
       severity   major
       type       "Communications Alarm"
       annotation summary
       value "This alert is fired for upf {{$labels.upf_name}} as n4 session requests are
getting timeout for last 15mins"
        exit
exit
```

Alert Routing to SNMP Trapper

The CNEE SNMP Trapper supports alert or alarm routing. Login to the CNEE Ops center to enable the SNMP Trapper because it is disabled by default. To enable SNMP traps, see Configuring SNMP Traps, on page 127.

The SNMP agent uses the Management Information Base (MIB) to handle SNMP trap notifications in the CISCO-CNEE-MIB.my. The CNEE generates two types of notifications with the following trap object identifiers (OID):

- **cneeFaultClearNotif**—The CNEE generates this notification when fault or alert gets cleared.
- **cneeFaultActiveNotif**—The CNEE generates this notification when fault or alert gets triggered.

For more details, see the "SMI MIB Reference" chapter in the *Ultra Cloud Core Subscriber Microservices Infrastructure Operations Guide*.

Alert Routing to Alert Logger

The Alert Logger allows to view active and silenced alerts or history of alerts triggered from CNEE Ops Center using show commands. Alert routing is enabled by default.

For more details, see the "Viewing Alert Logger" section in the "Common Execution Environment" chapter of the *Ultra Cloud Core Common Execution Environment Configuration and Administration Guide*.

Alarm Severity Levels

The alert or alarm severity levels are as follows:

- Critical
- Major
- Minor
- Warning

All severity level alerts are routed to the SNMP Trapper. The CNEE does not have a mechanism to route only critical or major alerts or alarms to the SNMP Trapper while configuring alerts rules. To address this requirement, configure the following CLI command to avoid routing minor or warning alerts or alarms to the SNMP Trapper.

```
cee# alerts silence add matchers { name severity isRegex true value
\"warning\|minor\" }
```

Use the following command to view silenced alerts or alarms.

```
show alerts silenced { summary | detail }
```

Configuring Alarm Support

This section describes how to configure Alarm Support on cnBNG CP.

Configuring Alarm Support involves the following procedures:

Configuring Alert Rules

Use the following commands to configure alert rules:

```
config
   alerts rules group alert_group_name
   interval-seconds seconds
   rule rule_name
      expression promql_expression
      duration duration
      severity severity_level
      type alert-type
      annotation annotation_name
      value annotation_value
      exit
   exit
```

NOTES:

- **alerts rules**: Specifies the Prometheus alerting rules.
- **group** *alert_group_name*: Specifies the Prometheus alerting rule group. One alert group can have multiple lists of rules. *alert-group-name* is the name of the alert group. The alert-group-name must be a string in the range of 0 to 64 characters.
- **interval-seconds** *seconds*: Specifies the evaluation interval of the rule group in seconds.
- **rule** *rule_name*: Specifies the alerting rule definition. *rule_name* is the name of the rule.
- **expression** *promql_expression*: Specifies the PromQL alerting rule expression. *promql_expression* is the alert rule query expressed in PromQL syntax.
- **duration** *duration*: Specifies the duration of a true condition before it is considered true. *duration* is the time interval before the alert is triggered.
- **severity** *severity_level*: Specifies the severity of the alert. *severity-level* is the severity level of the alert. The severity levels are critical, major, minor, and warning.
- **type** *alert_type*: Specifies the type of the alert. *alert_type* is the user-defined alert type. For example, Communications Alarm, Environmental Alarm, Equipment Alarm, Indeterminate Integrity Violation Alarm, Operational Violation Alarm, Physical Violation Alarm, Processing Error Alarm, Quality of Service Alarm, Security Service Alarm, Mechanism Violation Alarm, or Time Domain Violation Alarm.
- **annotation** *annotation_name*: Specifies the annotation to attach to the alerts. *annotation_name* is the name of the annotation.
- **value** *annotation_value*: Specifies the annotation value. *annotation_value* is the value of the annotation.

The following example configures an alert, which is triggered when the percentage of Unified Data Management (UDM) responses is less than the specified threshold limit.

Example:

```
config terminal
 alerts rules group BNGUDMchk_incr
 interval-seconds 300
 rule BNGUDMchk_incr
 expression "sum(increase(bng_restep_http_msg_total{nf_type=\"udm\",
```

```
message_direction=\"outbound\", response_status=~\"2..\"}[3m])) /
sum(increase(smf_restep_http_msg_total{nf_type=\"udm\", message_direction=\"outbound\"}[3m]))
 < 0.95"
  severity major
  type "Communications Alarm"
  annotation summary
  value "This alert is fired when the percentage of UDM responses is less than threshold"
  exit
 exit
exit
```

You can view the configured alert using the **show running-config alerts** command.

Example:

The following example displays the alerts configured in the running configuration:

```
 show running-config alerts
  interval-seconds 300
  rule SMFUDMchk_incr
  expression "sum(increase(smf_restep_http_msg_total{nf_type=\"udm\",
message_direction=\"outbound\", response_status=~\"2..\"}[3m])) /
sum(increase(smf_restep_http_msg_total{nf_type=\"udm\", message_direction=\"outbound\"}[3m]))
 < 0.95"
  severity major
  type "Communications Alarm"
  annotation summary
  value "This alert is fired when the percentage of UDM responses is less than threshold"

  exit
 exit
exit
```

Configuring SNMP Traps

Use the following commands to configure or enable SNMP Traps.

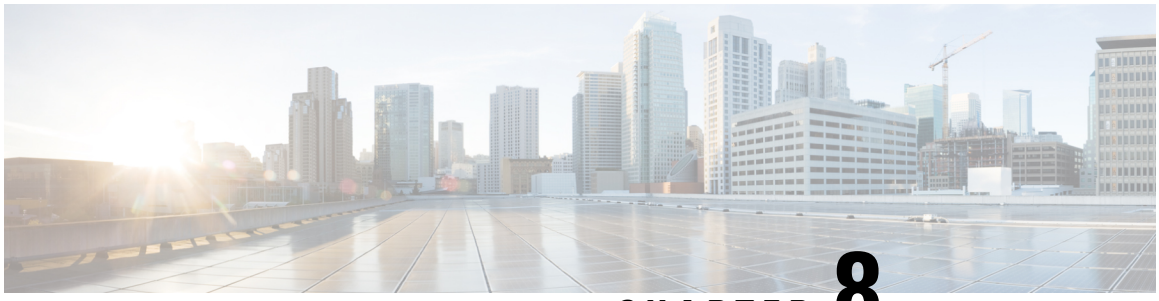
```
config
   snmp-trapper enable true
   snmp-trapper { v2c-target target | v3-target target |
   v3-engine-id source_engine_id }
      community [ community_string ]
      port [ port ]
      exit
   snmp-trapper source-ip-routes [ vip_options ]
   exit
```

```
config
  no snmp-trapper enable
  exit
```

NOTES:

- **snmp-trapper enable true**: Enables the SNMP trapper parameters.
- **v2c-target|v3-target [** *target* **]**: Specifies the list of SNMP v2c and v3 trap receivers.
- **community [** *community_string* **]**: Specifies the SNMP trap receiver community.
- **v3-engine-id** *source_engine_id*: Specifies the source engine ID for the v3 traps. *source_engine_id* must be an hexagonal string. For instance, 80004f.

- **port [** *port* **]**: Specifies the SNMP trap receiver port. *port* must be an integer in the range of 0 through 65535. The default value is 162.
- **source-ip-routes [** *vip_options* **]**: Enables binding to source IP for SNMP routing. *vip_options* specifies the virtual IP (VIP) address. The different options for virtual IP addresses include:
  - **default-external-vip**: Specifies the default external VIP for source IP routing.
  - **internal-vip**: Specifies the internal VIP for source IP routing.
  - **source-external-vips**: Specifies the external VIP per namespace.
- **no snmp-trapper enable**: Disables SNMP traps.



CHAPTER 8

Authentication, Authorization, and Accounting Functions

- Feature Summary and Revision History, on page 129
- Feature Description, on page 132
- AAA Overview, on page 132
- Using RADIUS Server Group, on page 133
- Specifying Method Order, on page 134
- Defining AAA Attributes, on page 134
- Making RADIUS Server Settings, on page 137
- Message-Authenticator validation for RADIUS servers, on page 137
- Balancing Transaction Load on the RADIUS Server, on page 141
- RADIUS Change of Authorization Overview, on page 142
- User Authentication and Authorization in the Local Network, on page 146
- Service Accounting, on page 147
- Utilizing Session accounting AAA profiles for Service Accounting and Accounting Send-Stop Setup-Failure, on page 149
- RADIUS Accounting Message Handling, on page 152
- Standard Compliance, on page 153
- RADIUS Automated Testing, on page 153
- Clear RADIUS operational statistics, on page 156
- RADIUS Attributes Filtering, on page 159
- Framed Route with tag and preference support, on page 163
- Configuring AAA Functions, on page 167

Feature Summary and Revision History

Summary Data

Table 17: Summary Data

Applicable Product(s) or Functional Area	cnBNG

Applicable Platform(s)	SMI
Feature Default Setting	Disabled - Configuration Required
Related Changes in this Release	Not Applicable
Related Documentation	*Cloud Native BNG Control Plane Command Reference Guide*

Revision History

Table 18: Revision History

Revision Details	Release
Introduced support to balance transaction load on the RADIUS server based on least outstanding transactions.	2026.04.0
Introduced Message-Authenticator validation support for RADIUS servers.	2026.01.0
Introduced support for clearing RADIUS operational statistics.	2026.01.0
The **user-plane-ip** keyword is removed from **nas-ip** and **nas-ipv6** commands under **radius accounting**, and **radius attribute** configurations. Introduced an optional command, **encode-user-plane-ip**, to allow explicit control over the encoding of user-plane IPv4/IPv6 addresses.	2025.04.0
Introduced support for framed routes.	2025.03.0
Enhancement introduced to session accounting and failure management with AAA profiles.	2025.01.0
Introduced support to balance transaction load on the RADIUS server based on least outstanding transactions.	2024.03.0
Introduced Asynchronous mode for sending RADIUS Accounting messages.	2024.03.0
Introduced support for different interim intervals for service and session accounting.	2024.02.0
Introduced a conditional approach to Change of Authorization (CoA) retries based on the Error-Cause AVPs.	2024.02.0

Revision Details	Release
Support is added for IPv6 transport between cnBNG and RADIUS AAA server, and also between cnBNG endpoint and CoA client.	2024.01.0
The RADIUS Automated Testing feature is introduced.	2024.01.0
The **user-plane-ip** keyword is added to the **nas-ip** attribute under **radius accounting**, and **radius attribute** configurations.	2023.04.0
Enhancement Introduced: The AAA feature is NSO-integrated.	2021.04.0

• This release introduces support for the Multi-Action Change of Authorization.
• Updated the following sections:
  • Configuring AAA Attributes: The **user-plane** is added as the one of the **format-order** identifiers while formatting AAA attributes.
  The **format-string** keyword was added to the AAframe format of the attribute.
  • Configuring RADIUS Accounting Options: The *nas_port* variable is added to explicitly specify the **nas-port** value.
  The **{ format-e** *format_e* **{ nas-port-type** *nas_port_type* **} }** options are added to the **nas-port** keyword.
  • Configuring RADIUS Attribute Format: The *nas_port* variable is added to explicitly specify the **nas-port** value.
  The **{ format-e** *format_e* **{ nas-port-type** *nas_port_type* **} }** options are added to the **nas-port** keyword.
	2021.03.0

First introduced.	2021.01.0

Feature Description



Note This feature is Network Services Orchestrator (NSO) integrated.

Note: All references to BNG in this chapter refer to the Cloud-Native Broadband Network Gateway (cnBNG).

This chapter provides information about configuring authentication, authorization, and accounting (AAA) functions on the BNG. BNG interacts with the RADIUS server to perform AAA functions. A group of RADIUS servers form a server group that is assigned specific AAA tasks. A method list defined on a server or server group lists methods by which authorization is performed. Some of the RADIUS features include creating specific AAA attribute formats, load balancing of RADIUS servers, throttling of RADIUS records, Change of Authorization (CoA), Session Accounting, and Service Accounting for QoS.

AAA Overview

AAA acts as a framework for effective network management and security. It helps in managing network resources, enforcing policies, auditing network usage, and providing bill-related information. BNG connects to an external RADIUS server that provides the AAA functions.

The RADIUS server performs the three independent security functions (authentication, authorization, and accounting) to secure networks against unauthorized access. The RADIUS server runs the Remote Authentication Dial-In User Service (RADIUS) protocol. (For details about RADIUS protocol, refer to RFC 2865). The RADIUS server manages the AAA process by interacting with BNG, and databases and directories containing user information.

The RADIUS protocol runs on a distributed client-server system. The RADIUS client runs on BNG (Cisco ASR 9000 Series Router) that sends authentication requests to a central RADIUS server. The RADIUS server contains all user authentication and network service access information.

The AAA processes, the role of RADIUS server during these processes, and some BNG restrictions, are explained in these sections:

Authentication

The authentication process identifies a subscriber on the network, before granting access to the network and network services. The process of authentication works on a unique set of criteria that each subscriber has for gaining access to the network. Typically, the RADIUS server performs authentication by matching the credentials (user name and password) the subscriber enters with those present in the database for that subscriber. If the credentials match, the subscriber is granted access to the network. Otherwise, the authentication process fails, and network access is denied.

Authorization

After the authentication process, the subscriber is authorized for performing certain activity. Authorization is the process that determines what type of activities, resources, or services a subscriber is permitted to use. For example, after logging into the network, the subscriber may try to access a database, or a restricted website. The authorization process determines whether the subscriber has the authority to access these network resources.

AAA authorization works by assembling a set of attributes based on the authentication credentials provided by the subscriber. The RADIUS server compares these attributes, for a given username, with information contained in a database. The result is returned to BNG to determine the actual capabilities and restrictions that are to be applied for that subscriber.

Accounting

The accounting keeps track of resources used by the subscriber during network access. Accounting is used for billing, trend analysis, tracking resource utilization, and capacity planning activities. During the accounting process, a log is maintained for network usage statistics. The information monitored include, but are not limited to - subscriber identities, applied configurations on the subscriber, the start and stop times of network connections, and the number of packets and bytes transferred to, and from, the network.

BNG reports subscriber activity to the RADIUS server in the form of accounting records. Each accounting record comprises of an accounting attribute value. This value is analyzed and used by the RADIUS server for network management, client billing, auditing, etc.

The accounting records of the subscriber sessions may timeout if the BNG does not receive acknowledgments from the RADIUS server. This timeout can be due to RADIUS server being unreachable or due to network connectivity issues leading to slow performance of the RADIUS server. It is therefore recommended that a RADIUS server **deadtime** be configured on the BNG, to avoid loss of sessions. Once this value is configured, and if a particular session is not receiving an accounting response even after retries, then that particular RADIUS server is considered to be non-working and further requests are not sent to that server.

Restrictions

- On session disconnect, transmission of the Accounting-Stop request to RADIUS may be delayed for a few seconds while the system waits for the "final" session statistics to be collected from the hardware. The Event-Timestamp attribute in that Accounting-Stop request should, however, reflect the time the client disconnects, and not the transmission time.

Using RADIUS Server Group

A RADIUS server group is a named group of one or more RADIUS servers. Each server group is used for a particular service. For example, in an AAA network configuration having two RADIUS server groups, the first server group can be assigned the authentication and authorization task, while the second group can be assigned the accounting task.

Server groups can include multiple host entries for the same server. Each entry, however, must have a unique identifier. This unique identifier is created by combining an IP address and a UDP port number. Different ports of the server, therefore, can be separately defined as individual RADIUS hosts providing a specific AAA service. In other words, this unique identifier enables RADIUS requests to be sent to different UDP ports on the same server. Further, if two different host entries on the same RADIUS server are configured for the same service (like the authentication process), then the second host entry acts as a fail-over backup for the first one. That is, if the first host entry fails to provide authentication services, BNG tries with the second host entry. (The RADIUS host entries are tried in the order in which they are created.)

For assigning specific actions to the server group, see Configuring RADIUS Server Group, on page 182.

Specifying Method Order

Method order for AAA defines the methods using which authorization is performed, and the sequence in which these methods are executed. Before any defined authentication method is performed, the method order must be applied to the configuration mechanism responsible for validating user-access credentials.

On BNG, you have to specify the method order and the server group that will be used for AAA services. For specifying method order, see Configuring Method Order for AAA, on page 171.

Defining AAA Attributes

The AAA attribute is an element of RADIUS packet. A RADIUS packet transfers data between a RADIUS server and a RADIUS client. The AAA attribute parameter, and its value - form a Attribute Value Pair (AVP). The AVP carries data for both requests and responses for the AAA transaction.

The AAA attributes either can be predefined as in Internet Engineering Task Force (IETF) attributes or vendor defined as in vendor-specific attributes (VSAs). For more information about the list of BNG supported attributes, see RADIUS Attributes, on page 601.

The RADIUS server provides configuration updates to BNG in the form of attributes in RADIUS messages. The configuration updates can be applied on a subscriber during session setup through two typical methods—per-user attributes, which applies configuration on a subscriber as part of the subscriber's authentication Access Accept, or through explicit domain, port, or service authorization Access Accepts. This is all controlled by the Policy Rule Engine's configuration on the subscriber.

When BNG sends an authentication or an authorization request to an external RADIUS server as an Access Request, the server sends back configuration updates to BNG as part of the Access Accept. In addition to RADIUS configuring a subscriber during setup, the server can send a change of authorization (CoA) message autonomously to the BNG during the subscriber's active session life cycle, even when the BNG did not send a request. These RADIUS CoA updates act as dynamic updates, referencing configured elements in the BNG and instructing the BNG to update a particular control policy or service policy.

BNG supports the concept of a "service", which is a group of configured features acting together to represent that service. Services can be represented as either features configured on dynamic-templates through CLI, or as features configured as RADIUS attributes inside Radius Servers. Services are activated either directly from CLI or RADIUS through configured "activate" actions on the Policy Rule Engine, or through CoA "activate-service" requests. Services can also be deactivated directly (removing all the involved features within the named service) through configured "deactivate" action on the Policy Rule Engine or through CoA "deactivate-service" requests.

The attribute values received from RADIUS interact with the subscriber session in this way:

- BNG merges the values received in the RADIUS update with the existing values that were provisioned statically by means of CLI commands, or from prior RADIUS updates.
- In all cases, values received in a RADIUS update take precedence over any corresponding CLI provisioned values or prior RADIUS updates. Even if you reconfigured the CLI provisioned values, the system does not override session attributes or features that were received in a RADIUS update.
- Changes made to CLI provision values on the dynamic template take effect immediately on all sessions using that template, assuming the template features have not already been overridden by RADIUS. Same applies to service updates made through CoA "service-update" requests.

AAA Attribute List

An attribute list is named list that contains a set of attributes. You can configure the RADIUS server to use a particular attribute list to perform the AAA function.

To create an attribute list, see Configuring RADIUS Attributes, on page 176.

AAA Attribute Format

It is possible to define a customized format for some attributes. For the configuration syntax for creating a new format, see Configuring AAA Attributes, on page 168.

Once the format is defined, the FORMAT-NAME can be applied to various AAA attributes such as username, nas-port-ID, calling-station-ID, and called-station-ID. The configurable AAA attributes that use the format capability are explained in the section Creating Attributes of Specific Format, on page 135.

To create a customized nas-port attribute and apply a predefined format to nas-port-ID attribute , see Configuring RADIUS Attribute Format, on page 177.

Specific functions can be defined for an attribute format for specific purposes. For example, if the input username is "text@abc.com", and only the portion after "@" is required as the username, a function can be defined to retain only the portion after "@" as the username. Then, "text" is dropped from the input, and the new username is "abc.com". To apply username truncation function to a named-attribute format, see Configuring AAA Attributes, on page 168.

Creating Attributes of Specific Format

BNG supports the use of configurable AAA attributes. The configurable AAA attributes have specific user-defined formats. The following sections list some of the configurable AAA attributes used by BNG.

Username

BNG has the ability to construct AAA username and other format-supported attributes for subscribers using MAC address, circuit-ID, remote-ID, and DHCP Option-60 (and a larger set of values available in CLI). The DHCP option-60 is one of the newer options that is communicated by the DHCP client to the DHCP server in its requests; it carries Vendor Class Identifier (VCI) of the DHCP client's hardware.

The MAC address attribute is specified in the CLI format in either of these forms:

- mac-address: for example, 0000.4096.3e4a
- mac-address-ietf: for example, 00-00-40-96-3E-4A
- mac-address-raw: for example, 000040963e4a
- mac-address-custom1: for example, 01.23.45.67.89.AB

  (This particular MAC address format is available only from Release 6.2.1 and later).

NAS-Port-ID

The NAS-Port-ID is constructed by combining BNG port information and access-node information. The BNG port information consists of a string in this form:

```
"eth phy_slot/phy_subslot/phy_port:XPI.XCI"
```

For 802.1Q tunneling (QinQ), XPI is the outer VLAN tag and XCI is the inner VLAN tag.

If the interface is QinQ, the default format of nas-port-ID includes both the VLAN tags; if the interface is single tag, it includes a single VLAN tag.

In the case of a single VLAN, only the outer VLAN is configured, using this syntax:

```
<slot>/<subslot>/<port>/<outer_vlan>
```

In the case of QinQ, the VLAN is configured using this syntax:

```
<slot>/<subslot>/<port>/<inner_vlan>.<outer_vlan>
```

In the case of a bundle-interface, the phy_slot and the phy_subslot are set to zero (0); whereas the phy_port number is the bundle number. For example, 0/0/10/30 is the NAS-Port-ID for a Bundle-Ether10.41 with an outer VLAN value 30.

The nas-port-ID command is extended to use the 'nas-port-type' option so that the customized format (configured with the command shown above) can be used on a specific interface type (nas-port-type).

If 'type' option is not specified, then the nas-port-ID for all interface types is constructed according to the format name specified in the command.

By default, the same NAS-Port-ID value is sent in **Accounting Start**, **Interim**, and **Stop** records for the entire subscriber session.

If the **service-name** attribute is included in the NAS-Port-ID format (see Configuring AAA Attributes), that format is not used for session accounting, because session accounting does not include a service-name value.

Calling-Station-ID and Called-Station-ID

BNG supports the use of configurable calling-station-ID and called-station-ID. The calling-station-ID is a RADIUS attribute that uses Automatic Number Identification (ANI), or similar technology. It allows the network access server (NAS) to send to the Access-Request packet, the phone number from which the call came from. The called-station-ID is a RADIUS attribute that uses Dialed Number Identification (DNIS), or similar technology. It allows the NAS to send to the Access-Request packet, the phone number that the user called from.

NAS-Port Format

NAS-Port is a 4-byte value that has the physical port information of the Broadband Remote Access Server (BRAS), which connects the Access Aggregation network to BNG. It is used both by Access-Request packets and Accounting-Request packets. To uniquely identify a physical port on BRAS, multiple pieces of information such as shelf, slot, adapter, and so on is used along with the port number. A configurable format called format-e is defined to allow individual bits or group of bits in 32 bits of NAS-Port to represent or encode various pieces that constitute port information.

Individual bits in NAS-Port can be encoded with these characters:

- Zero: 0
- One: 1
- PPPoX slot: S
- PPPoX adapter: A
- PPPoX port: P
- PPPoX VLAN Id: V
- PPPoX VPI: I

- PPPoX VCI: C
- Session-Id: U
- PPPoX Inner VLAN ID: Q

The permissible nas-port type values are:

Nas-port-types	Values	Whether value can be derived from associated interface
VIRTUAL_PPPOEOVLAN	36	Yes
VIRTUAL_PPPOEOQINQ	37	Yes
VIRTUAL_IPOEOVLAN	43	Yes
VIRTUAL_IPOEOQINQ	44	Yes



Note If a NAS-Port format is not configured for a NAS-Port-Type, the system looks for a default CLI configuration for the NAS-Port format. In the absence of both these configurations, for sessions with that particular NAS-Port-Type, the NAS-Port attribute is not sent to the RADIUS server.

Making RADIUS Server Settings

In order to make BNG interact with the RADIUS server, certain server specific settings must be made on the BNG router.

For more making RADIUS server settings, see Configuring RADIUS Server, on page 181.

Restriction

The service profile push or asynchronously pushing a profile to the system is not supported. To download a profile from Radius, the profile must be requested initially as part of the subscriber request. Only service-update is supported and can be used to change a service that was previously downloaded.

Message-Authenticator validation for RADIUS servers

Message-Authenticator validation for RADIUS servers is a security enhancement that

- ensures the integrity and authenticity of RADIUS authentication and accounting packets
- uses the Message-Authenticator attribute (Type 80) to protect against tampering and forgery, and
- helps mitigate protocol-level vulnerabilities such as Blast-RADIUS (CVE-2024-3596).

Blast-RADIUS (CVE-2024-3596) is a protocol-level security vulnerability that affects RADIUS servers and clients using non-EAP authentication methods over UDP or TCP. It enables a man-in-the-middle attacker to forge valid Access-Accept responses to failed authentication requests without knowing user passwords or

shared secrets, due to weaknesses in how the RADIUS protocol uses MD5 for packet authentication. Although Blast-RADIUS does not reveal user credentials, it can allow unauthorized access to network resources by exploiting the authentication process. To mitigate this vulnerability, it is recommended to enforce Message-Authenticator validation on all RADIUS packets or to migrate to secure transports such as RADIUS over TLS (RadSec).

Starting from Release 2026.03.0, Message-Authenticator validation supports both **strict** and **permissive** modes. Strict mode requires supported inbound packets to include a valid Message-Authenticator attribute. Permissive mode supports phased rollout by continuing to insert the attribute on outbound packets while accepting supported inbound packets that omit it. Packets with an invalid Message-Authenticator are silently dropped in both modes.

Table 19: Feature history

Feature Name	Release Information	Description
Enhanced Message-Authenticator validation for RADIUS servers	2026.03.0	You can now configure Message-Authenticator validation in **strict** or **permissive** mode to support phased rollout across cnBNG and AAA peers.
Message-Authenticator validation for RADIUS servers	2026.01.0	You can now protect your RADIUS authentication and accounting traffic from protocol-level attacks such as Blast-RADIUS by enabling Message-Authenticator validation. This feature ensures that all RADIUS packets include a cryptographic Message-Authenticator attribute, which both the client and server validate using a shared secret. If a packet fails validation, the system silently drops it, preventing forged or tampered responses from being accepted.

How Message-Authenticator validation works

When enabled, Message-Authenticator validation adds or verifies a cryptographic Message-Authenticator attribute on all RADIUS requests and responses. Both the RADIUS client (such as cnBNG) and server use a shared secret to generate and validate this attribute using HMAC-MD5, ensuring that only legitimate, untampered packets are accepted.

Validation behavior depends on the configured mode:

- **strict**: Access-Accept, Access-Reject, Access-Challenge, Accounting-Response, CoA-Request, and Disconnect-Request packets must include a valid Message-Authenticator attribute. Packets that omit the attribute or fail validation are silently dropped.
- **permissive**: outbound packets still include the Message-Authenticator attribute. Supported inbound packets that omit the attribute are accepted to support phased migration. Packets with an invalid Message-Authenticator are silently dropped.

For CoA and disconnect, if a request is received without the Message-Authenticator attribute in permissive mode, the request is processed and the corresponding ACK or NAK is sent without the attribute for that exchange.

Configuration considerations

You can configure Message-Authenticator validation globally, per RADIUS server, globally for CoA, or per CoA client by using **enable-msg-authenticator { strict | permissive }** command.

- The command has no default value.
- If the command is absent at both the global and entity scope, Message-Authenticator processing is off for that entity.
- A per-server or per-CoA-client setting overrides the corresponding global setting.
- If no per-entity setting is configured, the entity inherits the global value.

Note

- The previous form of **enable-msg-authenticator** is no longer used. After upgrade, existing configurations that used **enable-msg-authenticator** without a value must be re-applied as **enable-msg-authenticator strict**.
- To move from permissive mode back to full enforcement, explicitly configure **strict** keyword. Removing the command disables Message-Authenticator processing.
- Configuration changes take effect on subsequent packet transmissions. In-flight packets may continue to follow the previous behavior until the next retry.

Supported platforms

This feature is supported across all cnBNG deployment models, including AIO, multi-server, and OpenShift.

Configure Message-Authenticator validation for RADIUS servers

Enable Message-Authenticator validation to protect your cnBNG deployment from protocol-level vulnerabilities such as Blast-RADIUS by ensuring the authenticity and integrity of RADIUS packets.

Procedure

Step 1 Use the **enable-msg-authenticator { strict | permissive }** command to enable Message-Authenticator validation.

- Use this sample configuration to enable validation globally for all RADIUS servers:

```
Configure
  profile radius
   enable-msg-authenticator strict
  exit
```

- Use this sample configuration to enable validation for a specific RADIUS server. Global configuration takes effect if not set.

```
Configure
  profile radius
   server 10.105.254.43 1812
    type   auth
    enable-msg-authenticator strict
```

```
 exit
 server 10.105.254.43 1813
  type   acct
  enable-msg-authenticator strict
 exit
exit
```

- Use this sample configuration to enable validation for all CoA clients:

```
Configure
  profile coa
   enable-msg-authenticator strict
  exit
```

- Use this sample configuration to enable validation for a specific CoA client:

```
Configure
  profile coa
   client 10.105.254.43
    enable-msg-authenticator strict
   exit
  exit
```



Note For phased rollout, replace the **strict** keyword with the **permissive** keyword.

Configuration is resolved at the entity scope first. If a RADIUS server or CoA client has its own **enable-msg-authenticator** setting, that value takes precedence over the corresponding global setting. If the entity does not have its own value, it inherits the global setting. If the command is absent at both scopes, Message-Authenticator processing is off for that entity.

Use permissive mode to support phased rollout across cnBNG and AAA peers. After all peers have been updated to always include Message-Authenticator, explicitly change the configuration to **strict**.

- Use this sample configuration to enable permissive mode globally while keeping one server in strict mode:

```
Configure
  profile radius
    enable-msg-authenticator permissive
    server 10.105.254.43 1812
     type auth
     enable-msg-authenticator strict
    exit
  exit
```

In this example, the per-server **strict** setting takes precedence over the global **permissive** setting.

Step 2 Use the **show radius** command to monitor for failed authentications (for example, **bad authenticators**) and confirm feature operation.

Example:

```
bng# show radius
------------------------------------------------------------
Server: 10.105.254.105, port: 1812, status: instance 1: up  , port-type: Auth
2 requests, 0 pending, 0 retransmits
0 accepts, 0 rejects, 2 timeouts
0 bad responses, 0 bad authenticators
0 unknown types, 0 dropped, 0 ms latest rtt
Auto Test Stats:
```

```
instance 1:
  Requests:0 Pending:0 Retransmits:0
  Rejects:0 Timeouts:0 Responses:0
  BadAuth:0 Dropped:0 BadResp:0
```

Access-Accept, Access-Reject, Access-Challenge, and Accounting-Response packets that fail the Message-Authenticator check are silently dropped, and the **bad authenticator** counter is incremented. In strict mode, packets that omit the attribute are treated as failures. In permissive mode, packets that omit the attribute are accepted, but packets with an invalid Message-Authenticator are still dropped.

Example:

```
bng# show radius-dyn-auth
-------------------------------------------------------
 IP: 10.105.254.105
 ------------------
 COA:
 0 total-requests       0 inprocess-requests
     0 retry-request-drops  0 invalid-requests
     0 bad-authenticators   0 internal-errors
 0 ack-sent             0 nak-sent
 ------------------
 DISCONNECT:
 0 total-requests       0 inprocess-requests
     0 retry-request-drops  0 invalid-requests
     0 bad-authenticators   0 internal-errors
 0 ack-sent             0 nak-sent
```

CoARequest and DisconnectRequest packets that fail the Message-Authenticator check are silently dropped, and the **bad authenticator** counter is incremented. In strict mode, requests that omit the attribute are treated as failures. In permissive mode, requests that omit the attribute are accepted and processed, and the corresponding ACK or NAK is sent without the attribute for that exchange. Requests with an invalid Message-Authenticator are dropped in both modes.

Balancing Transaction Load on the RADIUS Server

Table 20: Feature History

Feature Name	Release Information	Description
Balancing Transaction Load on the RADIUS Server	2024.03.0	This feature enhances performance by distributing AAA messages across servers, ensuring faster response times. It selects the RADIUS server with the fewest outstanding transactions, rather than using the previous First server or Round Robin methods, which did not account for server load. This results in a more efficient handling of authentication, authorization, and accounting tasks.

The Balancing Transaction Load on the RADIUS Server feature provides a mechanism to share the load of RADIUS access and accounting transactions, across a set of RADIUS servers. Each AAA request processing is considered to be a transaction. cnBNG distributes batches of transactions to servers within a server group.

When the first transaction for a new batch is received, cnBNG determines the server with the lowest number of outstanding transactions in its queue. This server is assigned that batch of transactions. cnBNG keeps

repeating this determination process to ensure that the server with the least-outstanding transactions always gets a new batch. This method is known as the least-outstanding method of load balancing.

The size of each batch is configurable. Changes in the batch size may impact the CPU load and network throughput. There is no standard size for batches, but generally, more than 50 transactions is considered large, and fewer than 25 is considered small.

Retransmitted messages go to the same server. The number of outstanding RADIUS messages is tracked per server at the pod level. Load balancing applies to the named RADIUS server groups or a global server group specified in the subscriber profile. In AAA method lists, this group must be referred to as "radius." All servers in the RADIUS group are subject to load balancing.

RADIUS Server Status and Automated Testing

The Balancing Transaction Load on the RADIUS Server feature checks the status of servers before sending transaction batches. Only live servers receive transaction batches.

Transactions are not sent to servers marked as dead. A server remains marked as dead until its timer expires. The server is included again only when the RADIUS automated tester verifies it as alive. Otherwise, the server is excluded from the selection algorithm.

The RADIUS automated tester periodically sends a request to the server. If the server returns an **Access-Reject** message, it is alive. If not, it remains marked as dead until detected as alive.

If a server is unresponsive, it is marked as dead, and transactions fail over to the next available server.

When using the RADIUS automated tester, verify that the authentication, authorization, and accounting (AAA) servers respond to test packets sent by the network access server (NAS). Incorrect configurations may cause packet drops and servers to be erroneously marked as dead.

For configuring the load balancing on the RADIUS server, see Configuring RADIUS Server Selection Logic, on page 182.

RADIUS Change of Authorization Overview

The RADIUS Change of Authorization (CoA) function allows the RADIUS server to change the authorization settings for a subscriber who is already authorized. CoA is an extension to the RADIUS standard that allows sending asynchronous messages from RADIUS servers to a RADIUS client, like BNG.



Note A CoA server can be a different from the RADIUS server.

To identify the subscriber whose configuration needs to be changed, a RADIUS CoA server supports and uses a variety of keys (RADIUS attributes) such as Accounting-Session-ID, Username, IP-Address, and ipv4:vrf-id.

The RADIUS CoA supports:

- account-update — BNG parses and applies the attributes received as part of the CoA profile. Only subscriber-specific attributes are supported and applied on the user profile.
- activate-service — BNG starts a predefined service on a subscriber. The service settings can either be defined locally by a dynamic template, or downloaded from the RADIUS server.

- deactivate-service — BNG stops a previously started service on the subscriber, which is equivalent to deactivating a dynamic-template.

For a list of supported Vendor-Specific Attributes for account operations, see Vendor-Specific Attributes for Account Operations, on page 607.

Note In order for BNG to enable interim accounting, it is mandatory for the CoA request to have both accounting method list from the dynamic-template and Acct-Interim-Interval attribute from the user profile. This behavior is applicable for accounting enabled through dynamic-template. Whereas, from Cisco IOS XR Software Release 5.3.0 and later, the CoA request needs to have only the Acct-Interim-Interval attribute in the user profile.

Service Activate from CoA

BNG supports activating services through CoA requests. The CoA **service-activate** command is used for activating services. The CoA request for the service activate should contain these attributes:

- "subscriber:command=activate-service" Cisco VSA
- "subscriber:service-name=<service name>" Cisco VSA
- Other attributes that are part of the service profile

The "<subscriber:sa=<service-name>" can also be used to activate services from CoA and through RADIUS.

Duplicate service activate requests can be sent to BNG from the CoA server. BNG does not take any action on services that are already activated. BNG sends a CoA ACK message to the CoA server under these scenarios:

- When a duplicate request with identical parameters comes from the CoA for a service that is already active.
- When a duplicate request with identical parameters comes from the CoA to apply a parameterized service.

BNG sends a CoA NACK message to the CoA server with an error code as an invalid attribute under these scenarios:

- When a request comes from the CoA to deactivate a non-parameterized service that is not applied to the session.
- When a request comes from the CoA to deactivate a parameterized service that is not applied to the session.
- When a duplicate request to apply a parameterized service is made with non-identical parameters from the CoA.
- When a request with non-identical parameters comes from CoA to deactivate a parameterized service.

Service Update from CoA

The service update feature allows an existing service-profile to be updated with a new RADIUS attribute list representing the updated service. This impacts any subscriber who is already activated with the service and new subscriber who activate the service in the future. The new CoA **service-update** command is used for activating this feature. The CoA request for the service update should have these attributes:

- "subscriber:command=service-update" Cisco VSA
- "subscriber:service-name=<service name>" Cisco VSA
- Other attributes that are part of the service profile

A service update CoA should have a minimum of these attributes:

- vsa cisco generic 1 string "subscriber:command=service-update"
- vsa cisco generic 1 string "subscriber:service-name=<service name>"

Web Logon with RADIUS Based CoA

To support Web Logon, a set of Policy Rule Events need to be configured in an ordered manner. These events are as follows:

- session-start:
  - On the start of a session, a subscriber is setup to get internet connectivity. The service is activated to redirect HTTP traffic to a Web portal for web-based logon.
  - Start the timer with duration for the maximum waiting period for authentication.
- account-logon—The Web portal collects the user credentials such as username and password and triggers a CoA account-logon command. When this event is triggered, subscriber username and password are authenticated by the RADIUS server. Once the authentication is successful, the HTTP redirect service is deactivated, granting user access to already connected internet setup. Also, the timer established in session-start must be stopped. However, if the authentication fails during account-logon, BNG sends a NAK CoA request, allowing for further authentication attempts to take place.
- timer expiry—When the timer expires, the subscriber session is disconnected based on the configuration.

Multi-Action Change of Authorization

BNG supports multi-action Change of Authorization (CoA) wherein service providers can activate and deactivate multiple services using a single CoA request. Multi-action CoA is supported for **Service-Activate** and **Service-Deactivate** commands.

During the multi-action CoA request, if any of the COA requests fail to activate or deactivate, then any of the services which have been activated or deactivated as part of that CoA request is rolled back to its previous state. The session restores back to the its pre-MA-CoA state upon failure to activation or deactivation.

An Example of a Multi-Action Change of Authorization Use Case

The following example lists the sequence of events that occur in the case of a PTA session initiation.

1. PTA session's web traffic redirected to a service portal (HTTP Redirect)
2. The user activates the first level of service through the service portal. A multi-action COA request is initiated in the following sequence.
   a. Deactivate redirection
   b. Activate Turbo Button 1

c. Activate VoIP with two channels

3. The user activates the second level of service through the service portal. A multi-action COA request is initiated in the following sequence.
   a. Deactivate Turbo Button 1
   b. Activate Turbo Button 2
   c. Deactivate VoIP with two channels
   d. Activate VoIP with 4 channels

Interworking with Service-Level Accounting

BNG supports Service-Level Accounting, where a service is a collection of features that are activated and deactivated as a group. Service-Level Accounting and MA-CoA features are independent, that is, they can be applied separately. However, MA-CoA accounts for services that are activated or deactivated that have Service-Level Accounting enabled through the dynamic template configuration.

Generating Accounting Records

The following cases describes how the multi-action CoA records are generated for accounting purposes.

MA-CoA ACK Case

- If MA-CoA request contains only service activate commands, then START accounting record for those services are generated after the CoA Ack is sent out.
- If MA-CoA request contains only deactivate services or combination of activate and deactivate services, then for those services START or STOP accounting records are generated after the CoA Ack is sent out.

MA-CoA NAK Case (Rollback scenario)

- If MA-CoA request fails due to presence of invalid command formats or due to internal software failure or due to presence of invalid service names, that are not defined in the box, in such cases the accounting START or STOP messages are not generated upon rollback.
- If MA-CoA request fails due to internal feature programming failure, then the Service-START or Service-STOP accounting records may be generated for the services that were activated or deactivated before the failure. After the failure, the rollback is initiated and appropriate Service-START or Service-STOP records are generated for these services.

Sample MA-COA Request

```
exec /bin/echo
"Cisco-AVPair='subscriber:sd=svcQoSAcct1',Cisco-AVPair='subscriber:sd=svcQoSAcct2',
Cisco-AVPair='subscriber:sd=svcQoSAcct3',Cisco-AVPair='subscriber:sa=qosin_coa',
Cisco-AVPair='subscriber:sa=qosout_coa',
Acct-Session-Id=00000001" | /usr/local/bin/radclient  -r 1 -x 5.11.17.31:1700 coa coa
```

Enhanced CoA with Conditional Retry Logic

Table 21: Feature History

Feature Name	Release Information	Description
Enhanced CoA with Conditional Retry Logic	2024.02.0	We have introduced a conditional approach to Change of Authorization (CoA) retries based on the Error-Cause AVPs carried in CoA response messages. The CoA client uses the error cause reason and determines whether to initiate a CoA retry. This enhancement can reduce unnecessary traffic and processing overhead, resulting in more efficient network operations and better allocation of resources.

If a CoA request is erroneous, the cnBNG rejects it with a CoA Negative Acknowledgment (NAK) response. The CoA client treats all NAK responses equally, and attempts to send CoA requests automatically. This logic of sending unconditional retries for all error types strains network resources, leading to potential performance degradation for both the policy plane and the cn-BNG.

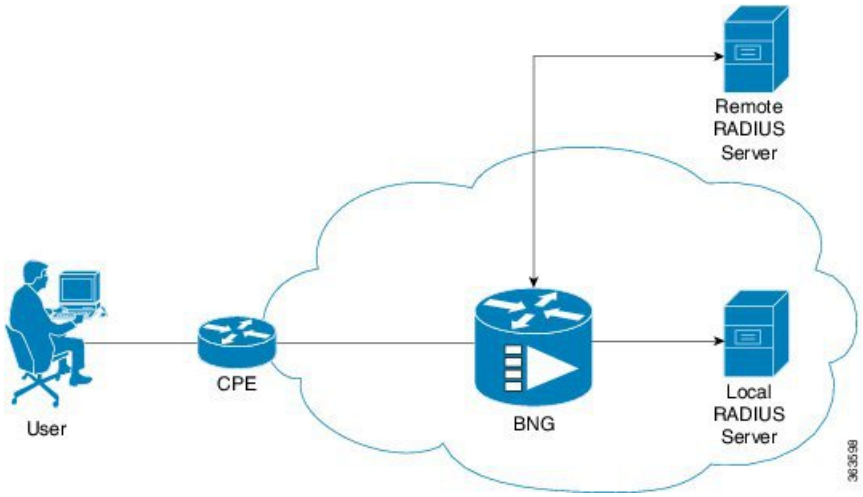
The Enhanced CoA with Conditional Retry Logic feature enables the CoA client to use the error cause carried in a CoA NAK response message and determine whether to send the CoA request again. For example, if the error cause is “503 Session Context Not found,” indicating that the subscriber session is nonexistent or already terminated, the CoA client would recognize that retrying the request would be futile as the cnBNG does not recognize the session context.

By adopting a conditional approach to CoA retries based on the Error-Cause AVPs, the system can avoid unnecessary traffic and processing overhead, resulting in a more efficient network performance.

User Authentication and Authorization in the Local Network

The user authentication and authorization in the local network feature in BNG provides the option to perform subscriber authorization locally (in a subscriber's network), instead of both remote authentication and authorization that occurs in RADIUS servers. With the User Authentication and Authorization in the Local Network feature, you can run the RADIUS server locally in your network, manage, and configure the RADIUS server locally in your network to the profile that is required for the environment. In the case of a remote RADIUS server, the RADIUS server is maintained by an external regulatory body (not within the subscriber's network) and subscriber will not be able to manage or configure the server.

Figure 11: User Authentication and Authorization in the Local Network



User Authentication and Authorization in the Local Network feature is used in a case when a user wants to perform a two-level authentication or authorization, first, a remote authentication (or authorization) followed by a local authorization (or authentication).

Note All the debug commands applicable to AAA server are applicable on User Authentication and Authorization in the Local Network feature.

Service Accounting

Accounting records for each service enabled on a subscriber can be sent to the configured RADIUS server. These records can include service-start, service-stop, and service-interim records containing the current state of the service and any associated counters. This feature is the Service Accounting feature. Service accounting records are consolidated accounting records that represent the collection of features that make up a service as part of a subscriber session.

Service accounting starts when a subscriber session comes up with a service enabled on it. This can happen through a dynamic template applied through a control policy, through access-accept (AA) messages when the session is authorized, or through a change of authorization (CoA), when a new service is applied on a subscriber session. Service accounting stops either when the session is terminated, or a service is removed from the session through CoA, or some other event that deactivates the service. Start records have no counters; interim and stop records with QoS counters are generated when service accounting is enabled for QoS. Interim accounting records can be generated, in between start and stop accounting, as an option with a pre-defined periodic interval. When the interim period is zero, interim accounting records are not created. Different interim intervals are based on every service for each session. Service accounting is enabled on each template, based on the configuration.

From Release 2024.02.0 onwards, different interim intervals for service and session accounting is supported. We recommend that you set the session interim accounting interval as a multiple of the service interim interval. For example, if the service accounting interim is set to 5 minutes, the session interim accounting interval must be set to a multiple of 5 minutes (such as 5, 10, 15 minutes, and so on).

Service Accounting is supported on bundle subscriber interfaces and line card subscriber interfaces.

For more information on service accounting for QoS, refer to Authentication, Authorization, and Accounting Functions, on page 129. For more information on commands to configure service accounting, refer to the Configuring Service Accounting, on page 462.

Note The policy-map associated to a dynamic template can be edited to change the service parameters. However, this does not update the accounting records. Therefore, to generate all the accounting records accurately, it is recommended that a new service with all the required service parameters be created and associated to the new service, through a CoA.

For service accounting, statistics for ingress and egress QoS policies, which are applied under each service for a given subscriber, may need to be reported as part of the accounting interim and stop records. For each service, these QoS counters can be reported as part of the accounting records:

- BytesIn — Aggregate of bytes matching all classes of the ingress QoS policy for the service minus the policer drops.
- PacketsIn — Aggregate of packets matching all classes of the ingress QoS policy for the service minus the policer drops.
- BytesOut — Aggregate of bytes matching all classes of the egress QoS policy for the service minus the queuing drops.
- PacketsOut — Aggregate of packets matching all classes of the egress QoS policy for the service minus the queuing drops

Dynamic template features that support accounting statistic collection and require that their statistics be reported in the AAA service accounting records can enable accounting statistics on their features using the newly-introduced optional **acct-stats** configuration option. This option is not available for the features that do not support statistic collection. By default, QoS accounting statistics are disabled to optimize performance.

Note The QoS counters for each direction is reported only if a QoS policy is applied for that service in the given direction. For example, if a service does not have an ingress policy applied, BytesIn and PacketsIn counters are reported as being 0.

Pre-requisites

- Subscriber accounting, the parent accounting record for service accounting, must be configured to enable the service accounting feature to work.
- The keyword **acct-stats** must be configured in service-policy configuration to enable the service accounting feature to report feature counter information as part of the records.

Restriction

- IPv4 and IPv6 subscriber sessions has a single set of service accounting records. They are merged into one set of bytes_in, bytes_out, packets_in, packets_out counters.

- Service accounting is not supported for static sessions.

Utilizing Session accounting AAA profiles for Service Accounting and Accounting Send-Stop Setup-Failure

Table 22: Feature History

Feature Name	Release Information	Description
Utilizing Session accounting AAA profiles for Service Accounting and Accounting Send-Stop Setup-Failure	2025.01.0	This feature enhances session accounting by allowing the use of a single AAA profile for both session and service accounting, reducing the need for multiple feature templates. This simplifies the configuration process when connecting to multiple RADIUS servers. Accounting Send-Stop Setup-Failure improves session management by sending an accounting stop record in case of session setup failures, ensuring that the RADIUS server clears stale session entries.

Utilizing Session accounting AAA profiles for Service Accounting

This feature allows you to efficiently manage Authentication, Authorization, and Accounting (AAA) profiles when a Control Plane (CP) is connected to multiple RADIUS servers. Normally, setting up multiple AAA profiles for authorization, session, and service accounting requires an increase in feature templates, which can complicate configuration and management. This feature aims to streamline the process by allowing the use of session accounting AAA profiles for service accounting, thereby reducing the number of necessary feature templates.

Accounting Send-Stop Setup-Failure

The RADIUS server creates a session context when authorization and authentication are successful. If the session fails to establish, a stale session context may remain on the RADIUS server. To clear this, the cnBNG Control Plane sends an accounting stop record in the event of a session setup failure. This stop record is sent if the subscriber profile is configured with the **send-stop** option.

Configure Session Accounting AAA Profiles for Service Accounting

Procedure

Step 1 Configure AAA profiles to be used for service accounting.

Example:

```
config
  profile aaa aaa_name
    authentication
```

```
      method-order custom_server_group
      exit
    authorization
      password password
      type subscriber method-order custom_server_group
      username value value
      password password
      exit
    accounting
      method-order custom_server_group
      exit
    exit
```

Here is a sample configuration for creating AAA profiles named **east-aaaprofile** and **west-aaaprofile.**

```
profile aaa east-aaaprofile
 authentication
  method-order server-group1
 exit
 authorization
  type subscriber method-order server-group1
  username value cnbng
  password cisco
 exit
 accounting
  method-order server-group1
 exit
exit
```

```
profile aaa west-aaaprofile
 authentication
  method-order server-group2
 exit
 authorization
  type subscriber method-order server-group2
  username value cnbng
  password cisco
 exit
 accounting
  method-order server-group2
 exit
exit
```

Step 2 Configure session accounting AAA profiles in feature templates.

Example:

```
Config
  profile feature-template basetemplate1
   session-accounting
     enable
     aaa-profile east-aaaprofile
    periodic-interval 2000
  exit

  profile feature-template basetemplate2
   session-accounting
     enable
     aaa-profile west-aaaprofile
```

```
   periodic-interval 2000
  exit

  profile feature-template plan-100Mbps
   qos
    in-policy inpolicy
    out-policy outpolicy
   exit
    service-accounting
     enable
     aaa-profile use-session-aaaprofile
     periodic-interval 2000
  exit
 exit
```

According to this sample configuration, subscribers using **basetemplate1** with the **plan-100Mbps** feature template will use **east-aaaprofile** for service accounting. Conversely, subscribers using **basetemplate2** with the **plan-100Mbps** feature template will use **west-aaaprofile** for service accounting.

Configure Accounting Send-Stop Setup-Failure

Procedure

Configure the subscriber profile to enable the sending of accounting stop records upon session bring-up failures.

Example:

```
configure
  profile subscriber subsprofile1
   accounting send-stop setup-failure aaa-profile aaa_profile_name
  exit
```

If the AAA_Profile_Name is	then
use-author-profile	the AAA profile configured for authorization is used for sending the accounting stop record in case of session bringup failure. This configuration is used for IPoE sessions.
use-authen-profile	the AAA profile configured for authentication is used for sending the accounting stop record in case of session bringup failure. This configuration is used for PPPoE, LAC or LNS sessions.

This configuration is crucial for managing IPoE sessions, as well as PPPoE, LAC, or LNS sessions, ensuring that session contexts are appropriately cleared from the RADIUS server.

RADIUS Accounting Message Handling

Table 23: Feature History

Feature Name	Release Information	Description
RADIUS Accounting Message Handling	2024.03.0	We have enhanced the system performance by preventing packet identifier (PID) exhaustion with the new Asynchronous mode for sending RADIUS Accounting messages. The cnBNG now releases PIDs after dispatching messages, rather than waiting for a long time to receive server responses. This change allows for PID reuse, mitigating scale issues and improving KPIs.

The cnBNG currently sends RADIUS Accounting messages to servers in synchronous mode. It reserves a packet identifier (PID) for each message. The PID is only released when a response arrives from the server. If the server fails to respond promptly, the cnBNG attempts to resend the message. The PID remains reserved until all timeouts and retransmissions are complete. During periods of high traffic, this can lead to PID exhaustion. As a result, cnBNG may drop new incoming RADIUS messages due to the lack of available PIDs, leading to a decrease in Key Performance Indicators (KPIs).

Asynchronous RADIUS Accounting Messages

To improve performance, cnBNG can switch to asynchronous mode for sending RADIUS Accounting messages. In this mode, cnBNG does not wait for a server response. After sending a message, the cnBNG waits for a timeout of one second, and then releases the PID. This allows the PID to be reused for subsequent messages. However, a brief delay is implemented before reusing a PID. This change can prevent PID exhaustion during high traffic and enhance KPIs. Note that the asynchronous mode applies only to RADIUS Accounting messages.

Configure Asynchronous RADIUS Accounting

Configure Asynchronous mode for Session Accounting

Use this configuration to enable asynchronous mode for session accounting.

```
config
  profile aaa aaa_name
    accounting
      session { acct-interim-async true/false | acct-start-async true/false |
acct-stop-async true/false }
        commit
```

Configure Asynchronous mode for Service Accounting

Use this configuration to enable asynchronous mode for service accounting.

```
config
  profile aaa aaa_name
    accounting
      service { acct-interim-async true/false | acct-start-async true/false |
```

```
acct-stop-async true/false }
       commit
```



Note If asynchronous RADIUS accounting is configured, message retransmission functionality will be unavailable. Lost messages due to network issues or server errors will not be retried by the Control Plane.

NOTES:

- **profile aaa** *aaa_name*: Specifies the AAA profile name and enters the AAA configuration mode.
- **accounting**: Enters the accounting sub-mode.
- **session [service] acct-interim-async** *true/false* : Specifies the option to enable or disable the asynchronous mode when interim updates are sent.
- **session [service] acct-start-async** *true/false*: Specifies the option to enable or disable the asynchronous mode when an Accounting-Start request is sent to AAA.
- **session [service] acct-stop-async** *true/false*: Specifies the option to enable or disable the asynchronous mode when an Accounting-Stop request is sent.

Standard Compliance

The AAA features are aligned with the following standards:

- RFC 2865 - Remote Authentication Dial In User Service (RADIUS)
- RFC 2866 - RADIUS Accounting
- RFC 5176 - Dynamic Authorization Extensions to Remote Authentication Dial In User Service (RADIUS)

RADIUS Automated Testing

For subscriber authentication and accounting, the cnBNG-CP chooses a RADIUS server from the list of configured servers, and sends RADIUS messages. If the RADIUS server is unreachable, it is considered dead, and is excluded from the selection algorithm until a 'dead-timer' expires. After this timer expires, the dead server is re-included in the selection list without checking if it is now reachable. This causes the cnBNB-CP to retransmit messages to the still-unreachable server, causing retransmissions and delays, which negatively impact Key Performance Indicators (KPIs).

The RADIUS Automated Testing feature allows the cnBNG-CP to periodically check the status of the RADIUS server until the server is considered dead or the dead-timer expires. With this feature, if the dead-timer expires, the cnBNG-CP attempts to send authentication and accounting TEST messages to the RADIUS server that is currently unreachable. If the server does not respond to these messages, it is marked as dead, and this process continues until the server is reachable. If the RADIUS server responds within the set number of retransmissions and timeouts, it is marked as available, and is then included in the selection algorithm list that is used to choose RADIUS servers.

There are two configuration scenarios:

- The following happens when RADIUS Automated Testing is enabled without the Idle-timer functionality:

  When the dead-detection criteria is met, the dead-timer starts with the value configured for **deadtime** CLI. Whenever the dead-timer expires, TEST messages are sent to the RADIUS server to check if the server is reachable. If the RADIUS server responds, the dead-timer stops, and the RADIUS server is marked UP. If the RADIUS server doesn't respond, the dead-timer is restarted.

- The following happens when the **idle-timer** CLI along with **auto-test enable** is configured:

  When the RADIUS server is UP, the status of the server is checked periodically by sending TEST RADIUS messages to the selected RADIUS server as per the configured idle-timer value. If the RADIUS server doesn't respond to the TEST messages, then the RADIUS server is marked as dead. The dead-timer starts as per the value configured in the **deadtime** CLI, and the periodic TEST message is stopped. Once the dead-timer expires, TEST RADIUS messages are sent to the RADIUS server, and the dead-timer is restarted. If the server responds to the TEST messages, the server is marked as UP, the dead-timer is stopped, and periodic TEST messages are restarted. If the server does not respond to TEST messages, on every dead-timer expiry, TEST RADIUS messages are sent to check reachability.

Restrictions

- RADIUS server availability based on VRF is not supported.
- **Show peers** command is not instance-aware.
- Round-trip time (RTT) is based on the server level, not the instance level.
- Server status is maintained per instance. All other statuses are maintained at a global level.
- Presently, only two VIPs are supported.

Software Upgrades

When you upgrade your current software (release version prior to cnBNG 2024.01) to cnBNG 2024.01 and later releases, perform the following steps:

- Fetch the keys of the RADIUS server from ETCD using the following command:

  ```
  kubectl exec -it -n bng <etcd pod name> -- etcdctl get --prefix "" --keys-only | grep serv
  ```

- Delete the key from ETCD using the following command:

  ```
  kubectl exec -it -n bng <etcd pod name> -- etcdctl del key “serverkey”
  ```



Note Rolling upgrade is not supported for this software upgrade scenario.

Configure RADIUS Automated Testing

This section describes how to enable RADIUS Automated Testing feature for accounting and authorization.

```
config
  profile radius
    server ipv4_address port_number
```

```
type { acct | auth }
auto-test enable
auto-test enable idle-timer number
commit
```

To disable the RADIUS Automated Testing feature, use the **no auto-test enable** command, and to disable the idle-timer functionality, use the **no auto-test idle-timer** command.

Note When the RADIUS Automated Testing feature is disabled, the status of the RADIUS server is reset to UP (reachable).

NOTES:

- **profile radius**: Enters the RADIUS configuration mode.
- **server** *ipv4_address port_number*: Specifies the IPv4 address and port of the RADIUS server.
- **type { acct | auth }**: Specifies the type of the RADIUS server. It can be one of the following:
  - **acct**: RADIUS server used for the accounting requests
  - **auth**: RADIUS server used for the authentication requests
- **auto-test enable**: Enables RADIUS automated testing.
- **auto-test enable idle-timer** *minutes*: Enables the idle-timer functionality. *minutes* value ranges from 1 through 30.
- **commit**: Commits the configuration.

Verifying RADIUS Automated Testing

Use the following **show** commands to verify the RADIUS Automated Testing feature.

```
bng# show radius auth-server 10.1.45.112:1812
Mon Nov  6  11:01:25.394 UTC+00:00
-------------------------------------------------------
 Server: 10.1.45.112, port: 1812, status: instance 1: up , port-type: Auth
 75 requests, 0 pending, 0 retransmits
 0 accepts, 75 rejects, 0 timeouts
 0 bad responses, 0 bad authenticators
 0 unknown types, 0 dropped, 1004 ms latest rtt
 Auto Test Stats:
 instance 1:
  Requests:74 Pending:0 Retransmits:0
  Rejects:74 Timeouts:0 Responses:0
  BadAuth:0 Dropped:0 BadResp:0
-------------------------------------------------------

bng# show radius acct-server 10.1.45.112:1813
Mon Nov  6  11:22:58.330 UTC+00:00
-------------------------------------------------------
 Server: 10.1.45.112, port: 1813, status: instance 1: up , port-type: Acct
 338 requests, 0 pending, 0 retransmits
 338 responses, 0 timeouts
 0 bad responses, 0 bad authenticators
 0 unknown types, 0 dropped, 3 ms latest rtt
```

```
Auto Test Stats:
instance 1:
 Requests:337 Pending:0 Retransmits:0
 Rejects:0 Timeouts:0 Responses:337
 BadAuth:0 Dropped:0 BadResp:0
```

Clear RADIUS operational statistics

Clearing RADIUS operational statistics is a management function that

- resets selected operational counters for RADIUS authentication, accounting, and dynamic authorization (CoA or Disconnect-Message) servers or clients
- enables administrators to perform targeted or comprehensive resets based on server role, IP address, and port, and
- requires explicit user confirmation to ensure intentional execution.

Table 24: Feature history

Feature Name	Release Information	Description
Clear RADIUS operational statistics	2026.01.0	You can simplify RADIUS server monitoring and troubleshooting by quickly resetting operational statistics as needed. This feature allows you to clear statistics for specific or all servers and clients, with a confirmation prompt to prevent accidental resets.

This feature allows you to selectively clear operational statistics for all RADIUS servers, all authentication servers, all accounting servers, specific authentication or accounting servers by IP and port, all dynamic authorization clients, or specific clients by IP address. Clearing statistics does not affect the UP/DOWN status of any RADIUS server.

Note that some counters, such as pending or in-process requests, are not reset by these commands.

Clear RADIUS server statistics

Reset authentication and accounting statistics for RADIUS servers to aid in operational troubleshooting.

Clearing counters can help operators monitor current activity and isolate issues.

Procedure

Step 1 Use the appropriate CLI command to clear statistics. These options are available:

Command	Description
clear radius all	Clear statistics for all RADIUS servers.
clear radius authentication all	Clear all authentication servers.

Command	Description
clear radius authentication server *ip* port *port*	Clear a specific authentication server. For example, `clear radius authentication server 192.168.1.100 port 1812`
clear radius accounting all	Clear all accounting servers.
clear radius accounting server *ip* port *port*	Clear a specific accounting server. For example, `clear radius accounting server 192.168.1.100 port 1812`

Step 2 When prompted with **This will clear RADIUS server statistics. Continue? [yes/no]:** type **`yes`** to confirm the operation. If you do not confirm, the operation is canceled.

Step 3 Use the **show radius** command to verify the server's operational status before and after clearing statistics.

Example:

Before clearing statistics

```
bng# show radius
------------------------------------------------------------
 Server: 10.105.254.43, port: 1842, status: instance 1: up  , port-type: Auth
 10 requests, 0 pending, 0 retransmits
 5 accepts, 5 rejects, 0 timeouts
 0 bad responses, 0 bad authenticators
 0 unknown types, 0 dropped, 39 ms latest rtt
 Auto Test Stats:
 instance 1:
  Requests:0 Pending:0 Retransmits:0
  Rejects:0 Timeouts:0 Responses:0
  BadAuth:0 Dropped:0 BadResp:0
```

Example:

After clearing statistics

```
bng# show radius
------------------------------------------------------------
 Server: 10.105.254.43, port: 1842, status: instance 1: up  , port-type: Auth
 0 requests, 0 pending, 0 retransmits
 0 accepts, 0 rejects, 0 timeouts
 0 bad responses, 0 bad authenticators
 0 unknown types, 0 dropped, 0 ms latest rtt
 Auto Test Stats:
 instance 1:
  Requests:0 Pending:0 Retransmits:0
  Rejects:0 Timeouts:0 Responses:0
  BadAuth:0 Dropped:0 BadResp:0
```

RADIUS statistics for the selected servers are reset to zero, except for `pending` or `inProcess` counters.

Clear dynamic authorization client statistics

Reset statistics for dynamic authorization (CoA or Disconnect-Message) clients.

Procedure

Step 1 Use the **clear radius-dyn-auth all** command to clear all clients.

Step 2 Use the **clear radius-dyn-auth client** *ip-address* command to clear a specific client.

Step 3 When prompted with **This will clear RADIUS server statistics. Continue? [yes/no]:**, type **`yes`** to confirm the operation. If you do not confirm, the operation is canceled.

Step 4 Use the **show radius-dyn-auth** command to verify the operational status of the client before and after clearing statistics.

Example:

Before clearing statistics

```
bng# show radius-dyn-auth
------------------------------------------------------------
------------------------------------------------------------
 IP: 10.1.2.105
 ------------------
 COA:
 10 total-requests       0 inprocess-requests
     0 retry-request-drops  0 invalid-requests
     0 bad-authenticators   0 internal-errors
 5 ack-sent             5 nak-sent
 ------------------
 DISCONNECT:
 20 total-requests       0 inprocess-requests
     0 retry-request-drops  0 invalid-requests
     0 bad-authenticators   0 internal-errors
 10 ack-sent             10 nak-sent
 ------------------
 UnknownTypesRcvd: 0
```

Example:

After clearing statistics

```
bng# show radius-dyn-auth
------------------------------------------------------------
IP: 10.1.2.105
 ------------------
 COA:
 0 total-requests       0 inprocess-requests
     0 retry-request-drops  0 invalid-requests
     0 bad-authenticators   0 internal-errors
 0 ack-sent             0 nak-sent
 ------------------
 DISCONNECT:
 0 total-requests       0 inprocess-requests
     0 retry-request-drops  0 invalid-requests
     0 bad-authenticators   0 internal-errors
 0 ack-sent             0 nak-sent
 ------------------
 UnknownTypesRcvd: 0
```

Dynamic authorization statistics are reset to zero for the selected clients, except for `inprocess-requests`.

RADIUS Attributes Filtering

Table 25: Feature History

Feature Name	Release Information	Description
RADIUS Attributes Filtering	2025.01.0	You can now customize RADIUS messages by including or excluding specific attributes to meet your server's requirements. This feature enhances your control over shared data between RADIUS clients and servers and optimizes message performance by filtering unnecessary attributes, thus reducing server load.

The RADIUS Attributes Filtering feature allows you to manage which RADIUS attributes are included or excluded from RADIUS messages. This is achieved through CLI commands that specify which IETF and vendor-specific attributes should be filtered. This feature applies only to RADIUS Authentication and Accounting Request and Reply message types, and does not affect CoA and Disconnect messages originated by the AAA Client.

This feature does not control the behavior of the AAA server. Instead, it manages how attributes are included or excluded in outbound messages based on specific actions:

- **Request Processing**: During the **Request** phase, the system processes Accept or Reject actions to determine the inclusion or exclusion of specific attributes in outbound Access-Request and Accounting-Request messages (such as START, INTERIM, or STOP messages) as necessary.
- **Reply Processing**: In the **Reply** phase, even if the AAA Server sends specific attributes, the cnBNG Control Plane (CP) uses its predefined configurations to determine whether to accept or reject these attributes. These configurations guide the cnBNG CP in deciding whether to apply or ignore the attribute values for sessions. This process ensures that attributes are correctly applied without modifying the original messages from the AAA Server.

Restrictions and Guidelines for RADIUS Attributes Filtering

These restrictions apply to the RADIUS Attributes Filtering feature:

- This feature is not supported for CoA and Disconnect messages originated by the AAA Client.
- This feature is only applicable to Authentication and Accounting Request and Reply message types.
- Attribute filtering is not supported for global RADIUS configurations.
- Multiple attribute lists can be configured, but only one attribute list per request and reply can be attached. Configuring multiple lists per request or reply for authentication and accounting is not possible.
- If there is a mismatch between the server-group list name attached to request/reply and the one configured in RADIUS server attribute configuration, the list is considered invalid, and attribute filtering will function as PASS. Similarly, if the list is not present, the feature will work as PASS.
- There is no automatic validation for the number of attributes or the names of Vendor Specific Attributes (VSAs). You must ensure that the IETF or numbered attributes are configured correctly and VSA names are spelled accurately. The Ops Center does not check the attribute list values for correctness.

- No change is introduced by this feature in how monitor protocol and monitor subscriber work.
- No change is introduced by this feature in other profile radius CLIs.

Configure RADIUS Attributes Filtering

Procedure

Step 1 Define the IETF and vendor-specific attributes in lists using the **profile radius** configuration.

Example:

```
configure
 profile radius
  radius-server
   attribute attr-list list_name
   ietf-attributes ietf_attributes
   vendor-attribute vendor-id vendor_id vendor-type vendor_type_value name [ addrv6
client-mac-address ]
   exit
```

The following is a sample configuration:

```
profile radius
 radius-server
  attribute attr-list list1
   ietf-attributes [ 88 100 30 2 ]
   vendor-attribute vendor-id 9 vendor-type 1 name [ addrv6 client-mac-address ]
   vendor-attribute vendor-id 9 vendor-type 56
   vendor-attribute vendor-id 9 vendor-type 60
   vendor-attribute vendor-id 3561 vendor-type 1
  exit
  attribute attr-list list2
   ietf-attributes [ 40 42 47 61 88 100 ]
   vendor-attribute vendor-id 9 vendor-type 56
   vendor-attribute vendor-id 3561 vendor-type 1
  exit
 exit
```

NOTES:

- **profile radius**: Enters the RADIUS configuration mode.
- **radius-server attribute attr-list** *list_name*:Specifies the RADIUS server attributes list.
- **ietf-attributes** *ietf_attributes*: Configures the list of Internet Engineering Task Force (IETF) attributes. For example, **ietf-attributes [88 8 30 2]** indicates that IETF attributes such as Framed-IP-Pool (88), Framed-IP-Address (8), Called-Station-Id(30), User-Password(2) are configured.
- **vendor-attribute vendor-id** *vendor_id*: Configures RADIUS attribute filtering for Vendor Specific Attributes (VSAs) by specifying vendor information such as the vendor-id in the RADIUS attribute list.
- **vendor-type** *value*: Configures the vendor specific information such as the vendor-type in the RADIUS attribute list.

Note For **vendor-type** *1* and **vendor-type** *9*, attribute values are specified in the name list.

For example, **vendor-attribute vendor-id 9 vendor-type 56** configures Cisco vendor attributes `cisco-vsa-sub-qos-pol-out`, and `cisco-vsa-sub-activate-service` with vendor-id `9` and vendor-type `56`.

vendor-attribute vendor-id 3561 vendor-type 56 configures the non-Cisco vendor attribute `Agent-Circuit-Id` with vendor-id `3561` and vendor type `1`.

- **name**: [Optional] Specifies the attribute name for a Cisco generic Vendor Specific Attribute (VSA).

  The **name** option is only available for vendor-id 9 and vendor-type 1.

Note For vendor-type 1, if names are not configured, the accept or reject action applies to all vendor-type attributes of the configured vendor ID.

- **addrv6 client-mac-address**: [Optional] Specifies the client MAC address in AABB.CCDD.EEFF format.

Step 2 Configure the accept attributes list for both request and reply messages in RADIUS authentication and accounting.

Example:

```
config
  profile radius
   server-group server_group_name
   accounting
    request accept list_1_name
    reply accept list_1_name
   exit
   authentication
    request accept list_2_name
    reply accept list_2_name
   exit
   server auth ip_address port_number
   exit
   server acct ip_address port_number
   commit
```

Step 3 Configure the reject attributes list for both request and reply messages in RADIUS authentication and accounting.

Example:

```
config
  profile radius
   server-group server_group_name
   accounting
    request reject list_1_name
    reply reject list_1_name
   exit
   authentication
```

```
 request reject list_2_name
 reply reject list_2_name
exit
server auth ip_address port_number
exit
server acct ip_address port_number
commit
```

NOTES:

- **server-group** *server_group_name*: Specifies the profile server group name to enter the Profile Server Group configuration mode.
- **accounting**: Enters the accounting sub-mode.
  - **request accept** *list_1_name*: Configures the accept attributes list for request messages in RADIUS accounting.
  - **reply accept** *list_1_name*: Configures the accept attributes list for reply messages in RADIUS accounting.
  - **request reject** *list_1_name*: Configures the reject attributes list for request messages in RADIUS accounting.
  - **reply reject** *list_1_name*: Configures the reject attributes list for reply messages in RADIUS accounting.
- **authentication**: Enters the authentication sub-mode.
  - **request accept** *list_1_name*: Configures the accept attributes list for request messages in RADIUS authentication.
  - **reply accept** *list_1_name*: Configures the accept attributes list for reply messages in RADIUS authentication.
  - **request reject** *list_1_name*: Configures the reject attributes list for request messages in RADIUS authentication.
  - **reply reject** *list_1_name*: Configures the reject attributes list for reply messages in RADIUS authentication.
- **server auth** *ip_address port_number*: Configures the authentication server.
- **server acct** *ip_address port_number*: Configures the accounting server.

Step 4 Use the **monitor protocol interface radius pcap yes** command to display and capture data packets from RADIUS interface messages.

NOTES:

- **pcap yes**: Enables PCAP file generation. By default, it is set to **no**.

Step 5 Use the **monitor subscriber supi** *supi_id* **capture-duration** *duration* **internal-messages yes** command to enable subscriber monitoring based on subscriber identity (SUPI).

NOTES:

- **supi** *supi_id*: Specifies the subscriber identity. For example, *aa11.0000.0001*, where *aa11.0000.0001* is the mac-id.
- **capture-duration** *duration*: [Optional] Specifies the duration, in seconds, for which the monitor subscriber is enabled. The default duration is 300 seconds.
- **internal-messages yes**: Enables internal messages, which are disabled by default.

Framed Route with tag and preference support

A framed route is a routing attribute that

- allows you to define specific IP prefixes and next hops for subscriber sessions
- enables the use of additional attributes such as administrative distance (preference) and tag values, and
- supports both IPv4 and IPv6 address families for flexible routing.

Table 26: Feature History

Feature Name	Release Information	Description
Framed Route with tag and preference support	2025.03.0	You can now prioritize and control subscriber traffic by assigning static routes with preference and tag values during authentication. By allowing the assignment of IPv4 or IPv6 framed routes through RADIUS attributes, this feature enables your network to automatically select the optimal path for both IPv4 and IPv6 subscribers.

Framed routes enable operators to define static routes via RADIUS attributes during subscriber authentication. These routes can include preference (admin distance) and tag values, allowing for prioritization and policy-based routing.

Examples:

- IPv4: 192.168.100.0/24 0.0.0.0 10 tag 123 pref 10
- IPv6: 2001:DB8::/64 2::1 10 tag 123 pref 10

Use case: Multi-homed subscriber routing scenarios

In multi-homed subscriber routing scenarios, a subscriber device connects to the broadband network using two or more access links, typically designating one as the primary (preferred) path and the other as a backup. This configuration is used to provide high availability and uninterrupted connectivity for subscribers.

Key characteristics:

- The subscriber device is anchored at two different BNGs, each associated with a separate access link (for example, fiber for primary and wireless for backup).
- Both BNGs advertise the subscriber's route to the core network, but each assigns a different administrative distance (preference value) to indicate priority.
- The core network uses the administrative distance or preference value to select the route for downstream traffic, ensuring the primary path is preferred whenever available.

Typical workflow:

1. The primary link (such as fiber) is used under normal circumstances for subscriber traffic.
2. If the primary link fails, the backup link (such as mobile broadband) automatically takes over, maintaining service continuity.

3. The administrative distance or preference value is dynamically set using RADIUS attributes provided by the AAA server, allowing flexible path prioritization.

Supported framed route formats

The cnBNG platform supports several variations of the Framed-Route and Framed-IPv6-Route attributes, including the use of tag and preference (admin distance) fields.

Format Example	Description
1. `192.168.100.0/24 0.0.0.0`	Basic static route
2. `192.168.100.0/24 0.0.0.0 10`	With metric
3. `192.168.100.0/24 0.0.0.0 10 tag 123`	With metric and tag
4. `192.168.100.0/24 0.0.0.0 10 tag 123 pref 100`	With metric, tag, and preference
5. `192.168.100.0/24 0.0.0.0 10 pref 100`	With metric and preference
6. `192.168.100.0/24 0.0.0.0 pref 100`	With preference only

The same flexibility applies for Framed-IPv6-Route attributes.

Backward compatibility considerations

This table outlines how different combinations of Control Plane (CP) and User Plane (UP) software versions affect framed route processing and subscriber session behavior.

CP version	UP version	Framed Route attributes	Behavior / outcome
New (2025.03.0 or newer)	Old (25.3.x or lower)	With `pref` and `tag`	UP cannot decode the preference (pref) attribute, but the session still establishes successfully. The UP installs the framed route with an administrative distance or metric of 1, and the pref attribute has no effect on the UP.
New (2025.03.0 or newer)	Old (25.3.x or lower)	Without `pref`/`tag`	CP and UP process normally. Session establishment succeeds.
Old (2025.02.0 or lower)	New (25.4.x or newer)	With `pref` and `tag`	CP cannot process; session establishment not attempted. Subscriber not connected.
Old (2025.02.0 or lower)	New (25.4.x or newer)	Without `pref`/`tag`	CP and UP process normally. Session establishment succeeds.

Standards compliance

The cnBNG framed route feature aligns with relevant industry standards:

- RFC 2865 (RADIUS – Framed-Route attribute for IPv4)
- RFC 3162 (RADIUS – Framed-IPv6-Route attribute for IPv6)
- Broadband Forum TR-459 (BNG requirements and interoperability)

Restrictions on Framed Route with tag and preference support

When deploying framed route preference and tag support on cnBNG, consider the following restrictions:

- **Maximum framed routes:**

  Each subscriber is limited to a maximum of 4 IPv4 framed routes and 4 IPv6 framed routes per address family.

- **Framed-Route attribute formats:**

  Only the supported attribute formats documented for cnBNG are accepted. Unsupported or invalid formats may cause route installation to fail.

- **Route gateway address:**

  If the gateway is specified as `0.0.0.0` (IPv4) or `::` (IPv6), the subscriber's IP address is used as the next-hop gateway by default.

- **Prefix length deduction:**

  If the prefix length is omitted in the framed route, the system infers the length based on the IP class (Class A: /8, Class B: /16, Class C: /24).

- **High availability:**

  Only the active UP installs routes; the standby UP holds the information but does not install until a switchover occurs.

- **Software version compatibility:**

  Full functionality requires both Control Plane (CP) and User Plane (UP) to be running compatible software versions that support framed route preference/tag attributes.

- **Accounting support:**

  Framed-Route attributes are not supported in RADIUS accounting messages on cnBNG.

- **Platform differences:**

  Feature support and limits may vary between different Cisco BNG platforms. Always verify compatibility when planning mixed deployments.

Configure framed route preference support

Enable and customize framed route administrative distance (preference) handling on cnBNG.

Use this task to specify how cnBNG interprets the metric or preference values in framed route attributes.

Before you begin

Ensure the system is running a software version that supports this feature.

Follow these steps to configure framed route preference support:

Procedure

Step 1 By default, cnBNG uses the **pref** value as the administrative distance for framed routes. However, if you apply this configuration on the User Plane, the metric value received in the framed-route will be used as the administrative distance instead.

Example:

```
configure
 cnbng-nal location 0/RSP0/CPU0
  framed-route-metric-as-distance
!
```

Note Apply this configuration when there are no subscriber sessions present on the router.

Step 2 You do not need to execute any commands on the Control Plane to configure this feature.

Step 3 Use the **show subscriber session detail command** command on the Control Plane to view the framed route.

Example:

```
bng# show subscriber session detail | more
Thu Jun  12 04:00:23.852 UTC+00:00
subscriber-details
{
  "subResponses": [
      <snip>
      "subcfgInfo": {
        "committedAttrs": {
          "attrs": {
            "accounting-list": "automation-aaaprofile",
            "acct-interval": "2000",
            "addr-pool": "automation-poolv4",
            "ipv4-mtu": "1400",
           "ipv6-route": "vrf prefix-vrf 2001:db8:1::/64 vrf gw-vrf 2001:db8:100::1 100 tag 101 pref
 200,2001:db8:2::/64 2001:db8:100::2 101 tag 12 pref 200,2001:db8:3::/64 0:0:0:0:0:0:0:0 pref 300
tag 444 199,2001:db8:4::/64 :: 103 tag 12 pref 400",
            "ppp-ipcp-reneg-ignore": "true",
            "ppp-ipv6cp-reneg-ignore": "true",
            "ppp-lcp-reneg-ignore": "true",
            "route": "vrf prefix-vrf 192.168.1.0/24 vrf gw-vrf 192.168.1.1 100 tag 101 pref
200,192.168.3.0/24 192.168.3.100 101 pref 200 tag 102,172.10.1.0 172.10.1.1 pref 200 tag 102
405,10.10.1.0 10.10.1.1 tag 102 555 pref 399",
            "session-acct-enabled": "true",
            "vrf": "automation-vrf"
          }
        },
        "activatedServices": [
          <snip>
        ]
      },
      <snip>
      "v4FramedRoute": [
        "vrf prefix-vrf 192.168.1.0/24 vrf gw-vrf 192.168.1.1 100 tag 101 pref 200",
        "192.168.3.0/24 192.168.3.100 101 pref 200 tag 102",
        "172.10.1.0 172.10.1.1 pref 200 tag 102 405",
```

```
      "10.10.1.0 10.10.1.1 tag 102 555 pref 399"
    ],
    "v6FramedRoute": [
      "vrf prefix-vrf 2001:db8:1::/64 vrf gw-vrf 2001:db8:100::1 100 tag 101 pref 200",
      "2001:db8:2::/64 2001:db8:100::2 101 tag 12 pref 200",
      "2001:db8:3::/64 0:0:0:0:0:0:0:0 pref 300 tag 444 199",
      "2001:db8:4::/64 :: 103 tag 12 pref 400"
    ],
    <snip>
```

Step 4 Use the **show cnbng-nal subscriber all detail** command on the User Plane to view the tag and admin distance in a framed-route session.

Example:

```
Router# show cnbng-nal subscriber all detail
IPv4 Framed Route:
  Prefix:                  192.168.1.0/24
  Next Hop:                192.168.1.1
  Tag:                     101
  Metric:                  100
  Distance:                200
  Next Hop VRF:            abc
  Prefix:                  192.168.2.0/24
  Next Hop:                192.168.2.1
  Tag:                     101
  Metric:                  100
  Distance:                200
  Next Hop VRF:            abc
IPv6 IANA Address:         2001:DB8::7002
IPv6 IAPD Prefix:          ::/0
IPv6 Slaac Prefix:         ::/0
CPE link local Address:    ::
IPv6 Framed Route:
  Prefix:                  2001:DB8:4004:800::/64
  Next Hop:                ::
  Tag:                     101
  Metric:                  120
  Distance:                71
  Next Hop VRF:            abc
  Prefix:                  2001:DB8:4700::/64
  Next Hop:                2001:DB8:35::35
  Tag:                     23
  Metric:                  99
  Distance:                0
  Next Hop VRF:            abc
```

Configuring AAA Functions

This section describes how to configure the following Authentication, Authorization, and Accounting (AAA) functions on the Control Plane (CP).

The configuration of the AAA functions involves the following procedures:

- Configuring AAA Attributes
- Configuring the CoA-NAS Interface
- Configuring Method Order for AAA

- Configuring RADIUS Accounting Options
- Configuring RADIUS Accounting Server Group
- Configuring RADIUS Attributes
- Configuring RADIUS-Dead Time
- Configuring RADIUS-Detect Dead Server
- Configuring RADIUS Pod
- Configuring RADIUS Maximum Retry
- Configuring RADIUS NAS-IP
- Configuring RADIUS Server
- Configuring RADIUS Server Group
- Configuring RADIUS Server Selection Logic
- Configuring RADIUS Timeout
- Configuring Radius PID-Quarantine

Configuring AAA Attributes

Use the following commands to configure a function for the AAA attribute format.

```
config
  profile attribute-format attribute_format_name
    format-order { addr | circuit-id-tag | client-mac-address |
      addr | circuit-id-tag | client-mac-address |
      client-mac-address-custom1 | client-mac-address-custom2 |
      client-mac-address-ietf | client-mac-address-raw |
      dhcp-client-id | dhcp-client-id-spl | dhcp-user-class |
      dhcp-vendor-class | dhcpv4-client-id-spl |
      dhcpv4-vendor-class | dhcpv6-client-id-ent-ident |
      dhcpv6-interface-id | dhcpv6-vendor-class-string |
      inner-vlan-id | outer-vlan-id | physical-adapter |
      physical-chassis | physical-port | physical-slot |
      physical-subslot | port-type | pppoe-session-id |
      remote-id-tag | service-name | user-plane | username }
  format-string format_string
  commit
```

NOTES:

- **profile attribute-format** *attribute_format_name*: Specifies the AAA attributes and enters the Attribute Format Configuration mode.
- **authorization**: Enters the Authorization sub-mode.
- **format-order** *attribute_format* | **identifier { addr | circuit-id-tag | client-mac-address | client-mac-address-custom1 | client-mac-address-custom2 | client-mac-address-ietf | client-mac-address-raw | dhcp-client-id | dhcp-client-id-spl | dhcp-user-class | dhcp-vendor-class |**

dhcpv4-client-id-spl | dhcpv4-vendor-class | dhcpv6-client-id-ent-ident | dhcpv6-interface-id | dhcpv6-vendor-class-string | inner-vlan-id | outer-vlan-id | physical-adapter | physical-chassis | physical-port | physical-slot | physical-subslot | port-type | pppoe-session-id | remote-id-tag | service-name | username } | value *value* **}**: Specifies the AAA attribute format order as follows:

- **addr**: Specifies the IPv4 address of the subscriber.
- **circuit-id-tag**: Specifies the circuit identifier tag.
- **client-mac-address**: Specifies the client MAC address in AABB.CCDD.EEFF format.
- **client-mac-address-custom1**: Specifies the first custom client MAC address in AABB.CCDD.EEFF format.
- **client-mac-address-custom2**: Specifies the second custom client MAC address in AABB.CCDD.EEFF format.
- **client-mac-address-ietf**: Specifies the client MAC address in Internet Engineering Task Force (IETF) format. That is, AA-BB-CC-DD-EE-FF format.
- **client-mac-address-raw**: Specifies the client MAC address in raw (AABBCCDDEEFF) format.
- **dhcp-client-id**: Specifies the DHCP client identifier.
- **dhcp-client-id-spl**: Specifies the DHCP client identifier special string.
- **dhcp-user-class**: Specifies the DHCP user class.
- **dhcp-vendor-class**: Specifies the DHCP vendor class.
- **dhcpv4-client-id-spl**: Specifies the DHCPv4 client identifier special string.
- **dhcpv4-vendor-class**: Specifies the DHCPv4 vendor class.
- **dhcpv6-client-id-ent-ident**: Specifies the DHCPv6 client and enterprise identifiers.
- **dhcpv6-interface-id**: Specifies the DHCPv6 interface identifier.
- **dhcpv6-vendor-class-string**: Specifies the DHCPv6 vendor class string.
- **inner-vlan-id**: Specifies the inner VLAN identifier.
- **outer-vlan-id**: Specifies the outer VLAN identifier.
- **physical-adapter**: Specifies the physical adapter.
- **physical-chassis**: Specifies the physical chassis.
- **physical-port**: Specifies the physical port.
- **physical-slot**: Specifies the physical slot.
- **physical-subslot**: Specifies the physical subslot.
- **port-type**: Specifies the interface or port type.
- **pppoe-session-id**: Specifies the PPPoE physical identifier.
- **remote-id-tag**: Specifies the remote identifier tag.
- **service-name**: Specifies the service name.

- **user-plane**: Specifies the User Plane (UP).
- **username**: Specifies the username.

- **format-string** *format_string*: Specifies the AAA format pattern. The *format_string* specifies the format string. Each identifier is represented by '%s' tuple. Any other character set is treated as a delimiter. For each '%s' in the format-string, the format-order identifier is used.

Note

- Validation on the number of '%s' in format-string and number of entries in format-order are not performed.
- For backward compatibility, the format-order still takes the delimiter configuration. In this scenario, the format-order takes precedence and the format-string is silently ignored.
- Use the delimiters either in the format-order (as in Release 2021.01) or in format-string (as in Release 2021.03).

Configuring the CoA-NAS Interface

Use the following configuration to define Change of Authorization (CoA) NAS interface in the RADIUS endpoint.

```
config
   endpoint radius
      interface coa-nas
         vip-ip ipv4_address vip-port port_number
         vip-ipv6 ipv6_address vip-ipv6-port port_number
         end
```

NOTES:

- **endpoint radius**: Enters the RADIUS endpoint configuration mode.
- **interface coa-nas**: This keyword defines a new interface "coa-nas", and allows to enter the CoA NAS interface configuration mode.
- **vip-ip** *ipv4_address* **vip-port** *port_number*: Configures the IPv4 address of the host. *ipv4_address* must be in standard IPv4 dotted decimal notation.

  You can configure a list of VIP-IPs to listen to the inbound CoA or DM requests.

  vip-port *port_number*: Specify the port number of the UDP proxy. By default, the port number is 3799. This default value is used only when the VIP-IP is specified.

Important This configuration allows only port to be specified per IP.

The BNG (udp-pxy) listens to the inbound CoA or DM request messages on these ports and ACK or NAK messages sent with the respective source IP and port.

- **vip-ipv6** *ipv6_address* **vip-ipv6-port** *port_number*: Configures the IPv6 address of the host.

  vip-ipv6-port*port_number*: Specify the port number of the UDP proxy.

Configuring Method Order for AAA

Use the following commands to assign the method order for the server group to use for subscriber authentication, authorization, and accounting.

Authentication

```
config
  profile aaa aaa_name
    authentication
      method-order custom_server_group
      commit
```

NOTES:

- **profile aaa** *aaa_name*: Specifies the AAA profile name and enters the AAA Configuration mode.
- **authentication**: Enters the Authentication sub-mode.
- **method-order** *custom_server_group*: Specifies the method-order to be applied by default for subscriber authentication.

  custom_server_group specifies the name of the server group where the method-order is applied.

Authorization

```
config
  profile aaa aaa_name
    authorization
      password password
      enc-password password
      type subscriber method-order custom_server_group
     username { format attribute_format | identifier { addr | circuit-id-tag
 | client-mac-address | client-mac-address-custom1 |
client-mac-address-custom2 | client-mac-address-ietf |
client-mac-address-raw | dhcp-client-id | dhcp-client-id-spl |
dhcp-user-class | dhcp-vendor-class | dhcpv4-client-id-spl |
dhcpv4-vendor-class | dhcpv6-client-id-ent-ident | dhcpv6-interface-id |
 dhcpv6-vendor-class-string | inner-vlan-id | outer-vlan-id |
physical-adapter | physical-chassis | physical-port | physical-slot |
physical-subslot | port-type | pppoe-session-id | remote-id-tag |
service-name | username } | value value }
      commit
```

NOTES:

- **profile aaa** *aaa_name*: Specifies the AAA profile name and enters the AAA Configuration mode.
- **authorization**: Enters the Authorization sub-mode.
- **password** *password* : Specifies the password for subscriber authentication.

- **enc-password** *password* : Specifies the password for subscriber authentication.

  The enc-password value is stored in an encrypted format. The show command displays the password as an encrypted string. After upgrading to the 2026.04.0 release, the plain-text password is restored from the previous configuration. If needed, the user can convert the password to the encrypted format.

  The password and enc-password options are mutually exclusive. You can configure only one at a time.

- **type subscriber method-order** *custom_server_group*: Specifies the method-order to be applied by default for subscriber authorization.

  custom_server_group specifies the name of the server group where the method-order is applied.

- **username { format** *attribute_format* **| identifier { addr | circuit-id-tag | client-mac-address | client-mac-address-custom1 | client-mac-address-custom2 | client-mac-address-ietf | client-mac-address-raw | dhcp-client-id | dhcp-client-id-spl | dhcp-user-class | dhcp-vendor-class | dhcpv4-client-id-spl | dhcpv4-vendor-class | dhcpv6-client-id-ent-ident | dhcpv6-interface-id | dhcpv6-vendor-class-string | inner-vlan-id | outer-vlan-id | physical-adapter | physical-chassis | physical-port | physical-slot | physical-subslot | port-type | pppoe-session-id | remote-id-tag | service-name | username } | value** *value* **}**: Specifies the username format, identifier, or value.

  - **format** *attribute_format*: Specifies the username attribute format.
  - **identifier { addr | circuit-id-tag | client-mac-address | client-mac-address-custom1 | client-mac-address-custom2 | client-mac-address-ietf | client-mac-address-raw | dhcp-client-id | dhcp-client-id-spl | dhcp-user-class | dhcp-vendor-class | dhcpv4-client-id-spl | dhcpv4-vendor-class | dhcpv6-client-id-ent-ident | dhcpv6-interface-id | dhcpv6-vendor-class-string | inner-vlan-id | outer-vlan-id | physical-adapter | physical-chassis | physical-port | physical-slot | physical-subslot | port-type | pppoe-session-id | remote-id-tag | service-name | username }**: Specifies the username identifiers as follows:
    - **addr**: Specifies the IPv4 address of the subscriber.
    - **circuit-id-tag**: Specifies the circuit identifier tag.
    - **client-mac-address**: Specifies the client MAC address in AABB.CCDD.EEFF format.
    - **client-mac-address-custom1**: Specifies the first custom client MAC address in AABB.CCDD.EEFF format.
    - **client-mac-address-custom2**: Specifies the second custom client MAC address in AABB.CCDD.EEFF format.
    - **client-mac-address-ietf**: Specifies the client MAC address in Internet Engineering Task Force (IETF) format. That is, AA-BB-CC-DD-EE-FF format.
    - **client-mac-address-raw**: Specifies the client MAC address in raw (AABBCCDDEEFF) format.
    - **dhcp-client-id**: Specifies the DHCP client identifier.
    - **dhcp-client-id-spl**: Specifies the DHCP client identifier special string.
    - **dhcp-user-class**: Specifies the DHCP user class.
    - **dhcp-vendor-class**: Specifies the DHCP vendor class.
    - **dhcpv4-client-id-spl**: Specifies the DHCPv4 client identifier special string.
    - **dhcpv4-vendor-class**: Specifies the DHCPv4 vendor class.

- **dhcpv6-client-id-ent-ident**: Specifies the DHCPv6 client and enterprise identifiers.
- **dhcpv6-interface-id**: Specifies the DHCPv6 interface identifier.
- **dhcpv6-vendor-class-string**: Specifies the DHCPv6 vendor class string.
- **inner-vlan-id**: Specifies the inner VLAN identifier.
- **outer-vlan-id**: Specifies the outer VLAN identifier.
- **physical-adapter**: Specifies the physical adapter.
- **physical-chassis**: Specifies the physical chassis.
- **physical-port**: Specifies the physical port.
- **physical-slot**: Specifies the physical slot.
- **physical-subslot**: Specifies the physical subslot.
- **port-type**: Specifies the interface or port type.
- **pppoe-session-id**: Specifies the PPPoE physical identifier.
- **remote-id-tag**: Specifies the remote identifier tag.
- **service-name**: Specifies the service name.
- **username**: Specifies the username.

Accounting

```
config
  profile aaa aaa_name
    accounting
      method-order custom_server_group
      commit
```

NOTES:

- **profile aaa** *aaa_name*: Specifies the AAA profile name and enters the AAA Configuration mode.
- **accounting**: Enters the Accounting sub-mode.
- **method-order** *custom_server_group*: Specifies the method-order to be applied by default for subscriber accounting.

  custom_server_group specifies the name of the server group where the method-order is applied.

Configuring RADIUS Accounting Options

This section describes how to configure the RADIUS accounting options.

```
config
  profile radius accounting
    algorithm { first-server | round-robin }
    attribute
```

```
      instance instance_id
      { nas-identifier value  | nas-ip ipv4_address  | nas-ipv6    ipv6_address
|
      nas-port { nas_port } | { format-e format_e
      { nas-port-type nas_port_type } }
      deadtime value
      detect-dead-server response-timeout value
      max-retry value
      timeout value
      commit
```

NOTES:

- **profile radius accounting**: Enters the RADIUS accounting configuration mode.
- **algorithm { first-server | round-robin }**: Defines the algorithm for selecting the RADIUS server.
  - **first-server**: Sets the selection logic as highest priority first. This is the default behavior.
  - **round-robin**: Sets the selection logic as round-robin order of servers.
- **attribute**: Configures the RADIUS identification parameters.
- **instance** *instance_id*: Specifies the instance ID.
  - **nas-identifier** *value*: Specifies the attribute name by which the system will be identified in Accounting-Request messages. *value* must be an alphanumeric string.
  - **nas-ip** *ipv4_address*: Specifies the NAS IPv4 address. *ipv4_address* must be an IPv4 address in dotted decimal notation.
  - **nas-ipv6** *ipv6_address*: Specifies the NAS IPv6 address.
  - **nas-port {** *nas_port* **} | { format-e** *format_e* **{ nas-port-type** *nas_port_type* **} }**: Specifies the nas-port attributes.
    - *nas_port* configures the NAS port value. The NAS port value ranges from 1 to 4294967295.

Note If none of the NAS port configurations are present, the existing default nas-port logic is applied. That is, setting a fixed-number per radius-pod.

- **format-e** *format_e_value* : Specifies the custom attribute formation support for nas-port. The nas-port is a 32 bit integer format. The configuration takes a 32 length of characters, each presenting a particular attribute mapping. The *format_e_value* pattern is: 01FSAPRiLUVQ]*):
- 0 – Set bit to 0

  1 – Set bit to 1

  F - PHY_SHELF

  S – PHY_SLOT

  A – PHY_ADAPTER

  P - PHY_PORT

R - PHY_CHASSIS

i - PHY_SUBSLOT

L - PHY_CHANNEL

V - OUTER_VLAN_ID

Q - INNER_VLAN_ID

U - PPPOE_SESSION_ID

nas-port-type *nas_port_type*: Specifies the NAS port type. The supported values range from 0 to 44.

Note

- The nas-port-type configuration is not in scope of the Control Plane. It is derived from the interface-type.
- The supported NAS port types are 36 , 37, 43, and 44.
- The NAS port type value takes precedence over the common NAS port format-e.

- **deadtime** *value*: Sets the time to elapse between RADIUS server marked unreachable and when we can re-attempt to connect.

  value must be an integer from 0 through 65535. Default: 10 minutes.

- **detect-dead-server response-timeout** *value*: Sets the timeout value that marks a server as "dead" when a packet is not received for the specified number of seconds.

  value must be an integer from 1 through 65535. Default: 10 seconds.

- **max-retry** *value*: Sets the maximum number of times that the system will attempt retry with the RADIUS server.

  value must be an integer from 0 through 65535. Default: 2

- **timeout** *value*: Sets the time to wait for response from the RADIUS server before retransmitting.

  value must be an integer from 1 through 65535. Default: 2 seconds.

- **commit**: Commits the configuration.
- All the keyword options under the RADIUS accounting configuration mode are also available within the RADIUS configuration mode.

Configuring RADIUS Accounting Server Group

This section describes how to configure the RADIUS server group.

```
configure
  profile radius
     server-group group_name
     commit
```

NOTES:

- **profile radius**: Enters the RADIUS configuration mode.
- **server group** *group_name*: Specifies the name of server group for use in RADIUS accounting. *group_name* must be an alphanumeric string.
- **commit**: Commits the configuration.

Configuring RADIUS Attributes

This section describes how to configure the RADIUS attributes for authentication and accounting.

```
config
  profile radius
        attribute
          instance instance_id
          { encode-user-plane-ip | nas-identifier value | nas-ip ipv4_address
 | nas-ipv6 ipv6_address }
           commit
```

NOTES:

- **profile radius**: Enters the RADIUS configuration mode.
- **attribute**: Configures the RADIUS identification parameters.
- **instance** *instance_id*: Specifies the instance ID.
  - **nas-identifier** *value*: Specifies the attribute name by which the system will be identified in Accounting-Request messages. *value* must be an alphanumeric string.
  - **nas-ip** *ipv4_address*: Specifies the NAS IPv4 address. *ipv4_address* must be an IPv4 address in dotted decimal notation.
  - **nas-ipv6** *ipv6_address*: Specifies the NAS IPv6 address.
- **encode-user-plane-ip**: Enables encoding User Plane IP address in RADIUS packets.
- **commit**: Commits the configuration.

Sample Configuration

The following is a sample configuration.

```
config
   profile radius
      attribute
        instance 1
          nas-identifier Ciscobng
          nas-ip 10.1.32.83
          nas-ipv6 2001::dB8:0
        exit
     exit
```

Configuring RADIUS Attribute Format

Use the following commands to configure the RADIUS identification parameters.

```
configure
profile radius attribute
  instance instance_id
     called-station-id { format-name format_name
               | nas-port-type nas_port_type }
     calling-station-id { format-name format_name
               | nas-port-type nas_port_type }
     nas-identifier nas_identifier
     nas-identifier-format nas_identifier_format
     nas-ip { ipv4_address |
     nas-ipv6 ipv6_address |
     nas-port { nas_port } | { format-e format_e
               { nas-port-type nas_port_type } }
     nas-port-id nas_port_id { format-name format_name|
               | nas-port-type nas_port_type }
     commit
```

NOTES:

- **profile radius attribute**: Enters the Profile RADIUS Attribute Configuration mode.
- **instance** *instance_id*: Specifies the instance ID.
- **called-station-id { format-name** *format_name* **| nas-port-type** *nas_port_type* **}** : Specifies the AAA called-station-id attribute.

  format-name *format_name*: Specifies the called-station-id format name.

  nas-port-type *nas_port_type*: Specifies the NAS port type. The supported values range from 0 to 44.

  - nas-port-type configuration is not in scope of the Control Plane. It is derived dependng on the interface-type.
  - The supported NAS port types are 36 , 37, 43, and 44.

- **calling-station-id { format-name** *format_name* **| nas-port-type** *nas_port_type* **}** : Specifies the AAA calling-station-id attribute.
- **nas-identifier { format-name** *format_name* **| nas-port-type** *nas_port_type* **}**: Specifies the attribute name with which the system is identified in the Access-Request messages. The identifier string ranges from 1 to 32 characters.
- **nas-identifier-format { format-name** *format_name* **| nas-port-type** *nas_port_type* **}**: Specifies the AAA nas-identifier-format attribute.
- **nas-ip** *ipv4_address*: Specifies the AAA NAS IPv4 address.
- **nas-ipv6** *ipv6_address*: Specifies the NAS IPv6 address.
- **nas-port {** *nas_port* **} | { format-e** *format_e* **{ nas-port-type** *nas_port_type* **} }**: Specifies the nas-port attributes.
  - *nas_port* configures the NAS port value. The NAS port value ranges from 1 to 4294967295.

Note If none of the NAS port configurations are present, the existing default nas-port logic is applied. That is, setting a fixed-number per radius-pod.

- **format-e** *format_e_value* : Specifies the custom attribute formation support for nas-port. The nas-port is a 32 bit integer format. The configuration takes a 32 length of characters, each presenting a particular attribute mapping. The *format_e_value* pattern is: 01FSAPRiLUVQ]*):
- 0 – Set bit to 0

  1 – Set bit to 1

  F - PHY_SHELF

  S – PHY_SLOT

  A – PHY_ADAPTER

  P - PHY_PORT

  R - PHY_CHASSIS

  i - PHY_SUBSLOT

  L - PHY_CHANNEL

  V - OUTER_VLAN_ID

  Q - INNER_VLAN_ID

  U - PPPOE_SESSION_ID

  nas-port-type *nas_port_type*: Specifies the NAS port type. The supported values range from 0 to 44.

Note
- The nas-port-type configuration is not in scope of the Control Plane. It is derived from the interface-type.
- The supported NAS port types are 36 , 37, 43, and 44.
- The NAS port type value takes precedence over the common NAS port format-e.

- **nas-port-id** *nas_port_id*: Specifies the AAA NAS port-id attribute.

Configuring RADIUS Dead Time

This section describes how to configure the RADIUS dead time.

```
config
  profile radius
    deadtime value
    commit
```

NOTES:

- **profile radius**: Enters the RADIUS configuration mode.
- **deadtime** *value*: Sets the time to elapse between RADIUS server marked unreachable and when an reattempt to connect can be made.

  value must be an integer from 0 through 65535. Default: 10 minutes.
- **commit**: Commits the configuration.

Sample Configuration

The following is a sample configuration.

```
config
   profile radius
     deadtime 15
     exit
```

Configuring RADIUS Detect Dead Server

This section describes how to configure the RADIUS detect dead server.

```
config
   profile radius
      detect-dead-server response-timeout value
      commit
```

NOTES:

- **profile radius**: Enters the RADIUS configuration mode.
- **detect-dead-server response-timeout** *value*: Sets the timeout value that marks a server as "dead" when a packet is not received for the specified number of seconds.

  value must be an integer from 1 through 65535. Default: 10 seconds.
- **commit**: Commits the configuration.

Sample Configuration

The following is a sample configuration.

```
config
   profile radius
      detect-dead-server response-timeout 100
      exit
```

Configuring RADIUS NAS-IP

This section describes how to configure the RADIUS NAS-IP.

Global RADIUS NAS-IP Configuration

Use the following configuration to configure the NAS-IP address.

```
config
   endpoint radius-dns
      interface radius-client
         vip-ip ipv4_address
         commit
```

NOTES:

- **endpoint radius-dns**: Enters the endpoint radius-ep configuration mode.
- **interface radius-client**: Enters the radius-client interface-type configuration mode.
- **vip-ip** *ipv4_address*: Sets the NAS-IP value, which is also used as the source-IP in UDP requests towards the RADIUS server.
- **commit**: Commits the configuration.

Configuration Example:

```
config
   endpoint radius-dns
      interface radius-client
      vip-ip 209.165.200.228
      exit
   exit
exit
```

Configuring RADIUS Pod

This section describes how to configure the RADIUS pod.

```
config
   endpoint radius
      replicas number_of_replicas
      commit
```

NOTES:

- **endpoint radius**: Enters the RADIUS endpoint configuration mode.
- **replicas** *number_of_replicas*: Sets the number of replicas required.
- **commit**: Commits the configuration.

Sample Configuration

The following is a sample configuration.

```
config
   endpoint radius
      replicas 3
      exit
```

Configuring RADIUS Retries

This section describes how to configure the maximum RADIUS retries.

```
config
  profile radius
    max-retry value
    commit
```

NOTES:

- **profile radius**: Enters the RADIUS configuration mode.
- **max-retry** *value*: Sets the maximum number of times that the system will attempt retry with the RADIUS server.

  value must be an integer from 0 through 65535. Default: 2
- **commit**: Commits the configuration.

Sample Configuration

The following is a sample configuration.

```
config
   profile radius
      max-retry 2
      exit
```

Configuring RADIUS Server

This section describes how to configure the RADIUS server settings.

```
config
  profile radius
    server ipv4/ipv6_address port_number
      secret secret_key
      priority priority_value
      type { acct | auth }
      commit
```

NOTES:

- **profile radius**: Enters the RADIUS configuration mode.
- **server** *ipv4/ipv6_address port_number*: Specifies the IPv4/IPv6 address and port of the RADIUS server.
- **secret** *secret_key*: Specifies the secret key.
- **priority** *priority_value*: Specifies the server priority.
- **type { acct | auth }**: Specifies the type of the RADIUS server. It can be one of the following:
  - acct: RADIUS server used for the accounting requests
  - auth: RADIUS server used for the authentication requests
- **commit**: Commits the configuration.

Configuring RADIUS Server Group

Use the following commands to configure the RADIUS server group.

```
config
  profile server-group server_group_name
    radius-group radius_server_group_name
    commit
```

NOTES:

- **profile server-group** *server_group_name*: Specifies the profile server group name to enter the Profile Server Group Configuration mode.
- **radius-group** *radius_server_group_name*: Specifies the RADIUS group server name.

Sample Configuration

The following is a sample configuration:

```
server-group automation-server-group
server auth 10.1.36.121 1812
exit
server acct 10.1.36.121 1813
exit
server auth 2001::10:1:36:121 1812
exit
server acct 2001::10:1:36:121 1813
exit
```

Configuring RADIUS Server Selection Logic

This section describes how to configure the RADIUS server selection logic.

```
config
  profile radius
    algorithm { first-server | round-robin | least-outstanding [
batch-size number ] }
    commit
```

NOTES:

- **profile radius**: Enters the RADIUS configuration mode.
- **algorithm { first-server | round-robin | least-outstanding [batch-size] }**: Defines the algorithm for selecting the RADIUS server.
  - **first-server**: Sets the selection logic as highest priority first. This is the default behavior.
  - **round-robin**: Sets the selection logic as round-robin order of servers.
  - **least-outstanding** : Sets the selection logic based on the server with the lowest number of outstanding transactions in its queue.
    - **batch-size** *number*: (Optional) Specifies the size of the batch.

      If you do not configure the **batch-size** value in the CLI command, the system takes the default batch size.

- **commit**: Commits the configuration.

Sample Configuration

The following is a sample configuration.

```
config
   profile radius
      algorithm least-outstanding batch-size 30
      exit
```

Configuring RADIUS Timeout

This section describes how to configure the RADIUS timeout.

```
config
   profile radius
      timeout value
      commit
```

NOTES:

- **profile radius**: Enters the RADIUS configuration mode.
- **timeout** *value_in_seconds*: Sets the time to wait for response from the RADIUS server before retransmitting.

  value must be an integer from 1 through 65535. Default: 2 seconds.
- **commit**: Commits the configuration.

Sample Configuration

The following is a sample configuration.

```
config
   profile radius
      timeout 4
      exit
```

Configuring Radius PID-Quarantine

The RADIUS PID-Quarantine feature temporarily reserves a Port Identifier (PID), comprising a UDP source port and an 8-bit RADIUS Identifier, after a request completes or times out. By preventing the immediate reuse of PIDs, the system avoids incorrect transaction correlation caused by late-arriving responses.

Table 27: Feature History Table

Feature Name	Release Information	Feature Description
Radius PID-Quarantine	2026.04.0	Enables temporary reservation of Port Identifiers (PIDs) for RADIUS requests after completion or timeout. This prevents late-arriving responses from being incorrectly correlated with new requests, enhancing overall RADIUS signaling reliability.

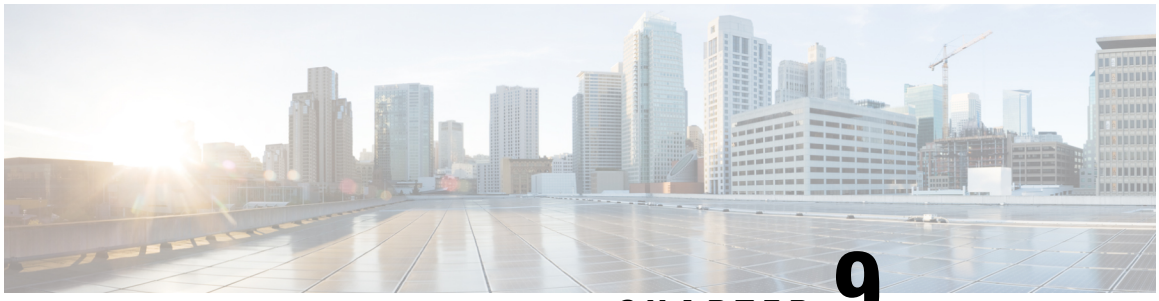
Sample Configuration

To configure the PID quarantine duration:

```
config
 profile radius
  pid-quarantine <5..60>
 commit
```

Parameters:

pid-quarantine: Specifies the PID quarantine value in seconds. The default value is 10 seconds. When a request times out, the PID is moved to a quarantine queue for the configured duration before it can be reused.



CHAPTER 9

Cisco Common Data Layer

- Feature Summary and Revision History, on page 185
- Feature Description, on page 186
- Limitations, on page 186

Feature Summary and Revision History

Summary Data

Table 28: Summary Data

Applicable Product(s) or Functional Area	cnBNG
Applicable Platform(s)	SMI
Feature Default Setting	Disabled - Configuration Required
Related Changes in this Release	Not Applicable
Related Documentation	Not Applicable

Revision History

Table 29: Revision History

Revision Details	Release
Enhancement Introduced: The CDL feature is NSO-integrated.	2021.04.0
First introduced.	2021.01.0

Feature Description

Note This feature is Network Services Orchestrator (NSO) integrated.

The Cisco Common Data Layer (CDL) is a high-performance next generation Key-value (KV) data store layer for all the Cloud Native applications. These applications use the CDL as a state management with High Availability (HA) and Geo HA functions. The CDL provides:

- Different Network Functions (NFs) - such as AMF, cnBNG Control Plane, SMF, and PCF - microservices.
- Multi-master support to achieve low latency read and write.
- Pure in-memory storage.
- Session related timers to notify NF on timer expiry.

Deploying CDL provides the following benefits:

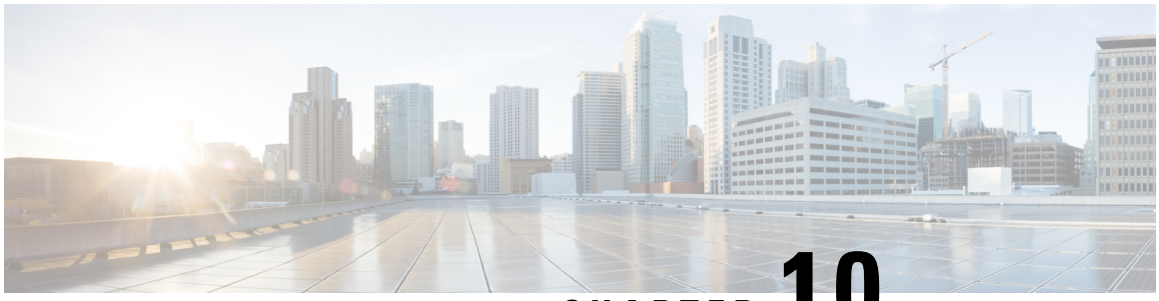
- Service-Based Architecture (SBA) with auto discovery and global accessibility.
- High performance, in-memory caching and in-memory data store.
- Container-based solution from the ground up.
- CDL can deploy and scale with simple API calls.
- Geo Redundant Replication among multiple cnBNG clusters.

For detailed information about CDL, refer to the *UCC SMI Common Data Layer Configuration and Administration Guide* at https://www.cisco.com/c/en/us/support/wireless/ultra-cloud-core-subscriber-microservices-infrastructure/products-installation-and-configuration-guides-list.html.

Limitations

The CDL feature has the following limitation on cnBNG.

Geo-redundancy is not supported.



CHAPTER 10

Control Plane and User Plane Association

- Feature Summary and Revision History, on page 187
- Feature Description, on page 188
- Manually remove CP-UP association, on page 188
- Enabling Control Plane and User Plane Association, on page 189
- Multiple Control Plane associations, on page 189
- Behavior during CP–UP connectivity failure, on page 212
- UP Session Disconnect Notification, on page 213
- UP Rollback Failure Handling Notification, on page 213

Feature Summary and Revision History

Summary Data

Table 30: Summary Data

Applicable Product(s) or Functional Area	cnBNG
Applicable Platform(s)	SMI
Feature Default Setting	Disabled - Configuration Required
Related Changes in this Release	Not Applicable
Related Documentation	Not Applicable

Revision History

Table 31: Revision History

Revision Details	Release
Introduced support for multiple CP associations to an UP.	2026.02.0

Revision Details	Release
Introduced support for IPv6 transport between the Control Plane and a peer User Plane.	2024.02.0
UP Rollback Failure Handling Notification support added.	2022.01.0
UP Session Disconnect Notification support added.	2021.03.0
First introduced.	2021.01.0

Feature Description

The Control Plane (CP) associates with a peer User Plane to synchronize with the number of subscriber sessions and state of each session. The CP and UP must maintain the total number of active sessions and their state on both sides.

To associate a UP to the CP, see the Associating the User Plane, on page 189.

Manually remove CP-UP association

Use this procedure to remove a CP-UP association and prevent stale session states. This process clears subscriber sessions and dynamic routes.

Procedure

Step 1 Remove subscriber sessions on CP to ensure that no session state remains in the database.

Example:

```
clear subscriber sessmgr [ gr-instance gr_instance_id | srg-peer-id srg_peer_id | upf upf_name ]
```

Step 2 Ensure that all dynamic and static routes associated with the UP are removed to prevent routing loops or stale RIB entries.

Example:

```
clear cnbng-nal dynamic-routes afi all
```

Step 3 Remove the Subscriber Redundancy Group (SRG) configuration that binds the UP to the Control Plane.

Example:

```
clear cnbng-nal cp-srg-association srg-group all
```

If Subscriber Redundancy Group of particular instance ID to be cleared, use the keyword **up-instance-id <1-15>**

Step 4 Clear CP-UP Association.

Example:

```
clear cnbng-nal cp-association
```

If association of particular instance ID to be cleared, use the **up-instance-id <1-15>**

Enabling Control Plane and User Plane Association

This section describes how to enable CP to UP association.

Associating the CP and UP involves the following procedure.

Associating the User Plane

Associating the User Plane

Use the following commands to associate the Control Plane (CP) to the peer User Plane.

```
config
  user-plane user_plane_name
  offline
  peer-address { ipv4 ipv4_address | ipv6 ipv6_address }
  port-id port_identifier subscriber-profile subscriber_profile
  subscriber-profile subscriber_profile
  exit
```

NOTES:

- **user-plane** *user_plane_name*: Specifies the User Plane (UP) name and enter UP Configuration mode.
- **offline**: Marks the UP offline for a graceful disconnect.

  The CP-UP association must be down before the **offline** command is enabled. You must first complete the procedure to remove the CP-UP association. For more information, see Manually remove CP-UP association, on page 188.
- **peer-address ipv4** *ipv4_address*: Specifies the peer IPv4 address of the UP.
- **peer-address ipv6** *ipv6_address*: Specifies the peer IPv6 address of the UP.
- **port-id** *port_identifier* **subscriber-profile** *subscriber_profile*: Specifies the port identifier of the UP. **subscriber-profile** *subscriber_profile* associates the subscriber profile at the port identifier level.
- **subscriber-profile** *subscriber_profile*: Associates the subscriber profile at UP level.

Multiple Control Plane associations

A multiple control plane association is a deployment capability that

- allows a single User Plane (UP) to simultaneously connect to multiple Control Plane (CP) instances
- enables Subscriber Redundancy Groups (SRG) to be mapped flexibly across different CPs, and
- improves scalability, redundancy, and migration flexibility in Broadband Network Gateway (BNG) environments.

Table 32: Feature History

Feature Name	Release Information	Description
Multiple Control Plane associations	2026.02.0	You can now improve cnBNG redundancy and flexibility by enabling a single User Plane (UP) to connect to multiple Control Plane (CP) instances at the same time. This allows Subscriber Redundancy Groups (SRG) to be mapped more flexibly across different Control Planes.

Traditionally, a User Plane could connect to only one Control Plane instance, which limited the ability to support complex SRG peering or to optimize resource utilization in large, distributed networks. With this enhancement, a User Plane (such as Cisco ASR 9000) can maintain up to 10 logical associations with different Control Plane instances or Geo-Redundancy (GR) instances. Each logical association acts as a separate User Plane context from the Control Plane perspective.

Benefits of multiple Control Plane associations

These are the key benefits:

- Enhanced network resilience through improved redundancy.
- Reduced operational complexity by simplifying migration processes.
- Support for scalable deployment across different environments.

Restrictions for multiple Control Plane associations

These restrictions apply when configuring multiple Control Plane associations with a single User Plane:

- Only IP over Ethernet (IPoE) hot standby Service Redundancy Group (SRG) sessions are supported across multiple user plane instances. PPPoE, LNS, and LAC session types are not supported in this mode.
- Warm standby is not supported when using logical User Plane instances.
- Support for non-SRG subscribers is available only on the default instance (ID 0).
- Parallel reconciliation involving subscriber, route, or SRG information, in any combination, is not supported.
- Changing a user-plane name on the Control Plane while the association is up is not supported.

Configure and verify multiple CP associations for a UP

To configure the Multiple CP associations feature, you must complete these tasks:

Configure multiple CP associations for a UP

Set up one or more logical associations between a UP and multiple CP instances to enable advanced redundancy and scalability.

Use this task when you want a UP node to support multiple CP clusters or SRG instances simultaneously, such as during migration or when optimizing SRG group mapping.

Before you begin

Ensure you have these prerequisites:

- Confirm the number and addresses of CP instances you wish to connect.

Follow these steps to configure multiple CP associations for a UP.

Procedure

Step 1 On the UP, configure the default (global) association parameters (instance-id 0).

Example:

```
Router# configure
Router(config)# cnbng-nal location 0/RSP0/CPU0
Router(config-cnbng-nal-local)# hostidentifier asr9k-1-cp-1-inst-0
Router(config-cnbng-nal-local)# up-server ipv4 10.1.1.51 vrf default gtp-port 15002 pfcp-port 15003
Router(config-cnbng-nal-local)# cp-server primary ipv4 192.0.2.1
Router(config-cnbng-nal-local)# cp-association retry-count 6
```

By default, the device uses instance ID 0 if no **up-instance-id** is specified.

Note You can also configure **up-server** at the global level. If **up-server** is not configured locally, the individual UP instance inherits the global **up-server** setting.

Step 2 Configure additional logical UP instances to support multiple CP-UP association.

Example:

```
Router# configure
Router(config)# cnbng-nal location 0/RSP0/CPU0
Router(config-cnbng-nal-local)# up-instance-id 1
Router(config-cnbng-nal-local)# hostidentifier asr9k-1-cp-1-inst-1
Router(config-cnbng-nal-local)# up-server ipv4 10.1.1.52 vrf default gtp-port 15002 pfcp-port 15003
Router(config-cnbng-nal-local)# cp-server primary ipv4 192.0.2.2
Router(config-cnbng-nal-local)# cp-association retry-count 6
```

Repeat this step for each additional logical UP instance. Assign a unique **up-instance-id** (1–15) and **hostidentifier** per instance.

Example:

```
Router# configure
Router(config)# cnbng-nal location 0/RSP0/CPU0
Router(config-cnbng-nal-local)# up-instance-id 2
Router(config-cnbng-nal-local)# hostidentifier asr9k-1-cp-1-inst-2
Router(config-cnbng-nal-local)# up-server ipv4 10.1.1.53 vrf default gtp-port 15002 pfcp-port 15003
Router(config-cnbng-nal-local)# cp-server primary ipv4 192.0.2.3
Router(config-cnbng-nal-local)# cp-association retry-count 6
```

Step 3 On the UP, configure SRG groups to be mapped to UP instances.

Example:

Associating a group to default UP instance:

```
Router# configure
Router(config)# cnbng-nal location 0/RSP0/CPU0
Router(config-cnbng-nal-local)# subscriber-redundancy
Router(config-cnbng-nal-sub-red)# group group2001
Router(config-cnbng-nal-srg-grp)# virtual-mac 0000.1111.03e9
Router(config-cnbng-nal-srg-grp)# access-tracking track_group2001
Router(config-cnbng-nal-srg-grp)# access-interface-list
Router(config-cnbng-nal-srg-grp-intf)# interface Bundle-Ether40.2001
```

Example:

Associating a group to UP instance 1:

```
Router# configure
Router(config)# cnbng-nal location 0/RSP0/CPU0
Router(config-cnbng-nal-local)# up-instance-id 1
Router(config-cnbng-nal-local)# subscriber-redundancy
Router(config-cnbng-nal-sub-red)# group group2002
Router(config-cnbng-nal-srg-grp)# virtual-mac 0000.1111.03ea
Router(config-cnbng-nal-srg-grp)# access-tracking track_group2002
Router(config-cnbng-nal-srg-grp)# access-interface-list
Router(config-cnbng-nal-srg-grp-intf)# interface Bundle-Ether40.2002
```

Step 4 On the CP, configure each UP instance as a separate UP.



Important The **user-plane** hostname configured on the CP must exactly match the **hostidentifier** configured for the corresponding UP instance on the UP.

Example:

```
user-plane
  instance 0
    user-plane asr9k-1-cp-1-inst-0
      peer-address ipv4 10.1.1.51
      subscriber-profile SUBS_PROF_1
      exit
      subscriber-redundancy
        group group2001
          preferred-role-active
          peer-identifier peer2001-x
          port-id-map port-name Bundle-Ether40.2001 1
        exit
      exit

  instance 1
    user-plane asr9k-1-cp-1-inst-1
      peer-address ipv4 10.1.1.52
      subscriber-profile SUBS_PROF_1
      exit
      subscriber-redundancy
        group group2002
          peer-identifier peer2002-x
          port-id-map port-name Bundle-Ether40.2002
        exit
      exit
```

Step 5 (Optional) On the UP, configure per-instance flow control thresholds.

Example:

```
Router(config-cnbng-nal-instance-id)# up-cp-notification flow-control 100
Router(config-cnbng-nal-instance-id)# up-cp-stats flow-control 150
```

Adjust per-instance flow and statistics thresholds as needed for scale and performance requirements.

Step 6 (Optional) On the UP, configure global flow control thresholds if required.

Use only if you require system-wide aggregate limits.

Example:

```
Router(config-cnbng-nal-local)# up-cp-notification global-flow-control 500
Router(config-cnbng-nal-local)# up-cp-stats global-flow-control 250
```

Step 7 On the UP, execute the **cnbng-nal cp-association start [up-instance-id 1] location 0/RSP0/CPU0** command to bring up the association.

Verify multiple CP associations to a UP

Ensure each logical UP instance is correctly associated and operating with its SRG CP peer. Diagnose and resolve issues related to SRG mapping, session and route allocation, and association failures.

Procedure

Step 1 Verify CP-UP association status for a specific or all UP instances.

Example:

```
Router# show cnbng-nal cp connection status up-instance-id 1

Tue Jun  2 08:25:19.093 IST
Location: 0/0/CPU0

UP Instance ID: 1

User-Plane configurations:
--------------------------
 IP   : 21.21.21.1
 GTP Port  : 15002
 PFCP Port  : 15003
 VRF   : default

Control-Plane configurations:
-----------------------------
 PRIMARY IP  : 21.21.21.2
 GTP Port  : 2152
 PFCP Port  : 8805

 Association retry count: 5

 Connection Status: Up
 Connection Status time stamp:  Mon Jun  1 19:23:18 2026

 Connection Prev Status: Init
 Connection Prev Status time stamp:  Mon Jun  1 19:20:48 2026

 Association status: Active
 Association status time stamp: Mon Jun  1 19:23:18 2026
```

```
 GTPu socket bind rc: Success
 PFCP socket bind rc: Success

[NAL-CP-INST-EVENTS]

| Event Name                    | Time Stamp              | User Data
| Association Active            | Jun  1 19:23:18.090052  | -- NA --
| Transport Up                  | Jun  1 19:23:18.091136  | -- NA --
```

```
Router# show cnbng-nal cp connection status up-instance-id all

Tue Jun  2 08:25:26.696 IST
Location: 0/0/CPU0
================================================================================
UP Instance ID: 1

User-Plane configurations:
--------------------------
 IP   : 21.21.21.1
 GTP Port  : 15002
 PFCP Port  : 15003
 VRF   : default

Control-Plane configurations:
-----------------------------
 PRIMARY IP  : 21.21.21.2
 GTP Port  : 2152
 PFCP Port  : 8805

 Association retry count: 5

 Connection Status: Up
 Connection Status time stamp:  Mon Jun  1 19:23:18 2026

 Connection Prev Status: Init
 Connection Prev Status time stamp:  Mon Jun  1 19:20:48 2026

 Association status: Active
 Association status time stamp: Mon Jun  1 19:23:18 2026

 GTPu socket bind rc: Success
 PFCP socket bind rc: Success

[NAL-CP-INST-EVENTS]

| Event Name                    | Time Stamp              | User Data
| Association Active            | Jun  1 19:23:18.090052  | -- NA --
| Transport Up                  | Jun  1 19:23:18.091136  | -- NA --

UP Instance ID: 2

User-Plane configurations:
--------------------------
 IP   : 31.31.31.1
 GTP Port  : 15004
 PFCP Port  : 15005
 VRF   : default

Control-Plane configurations:
-----------------------------
 PRIMARY IP  : 31.31.31.2
 GTP Port  : 2152
 PFCP Port  : 8805
```

```
 Association retry count: 5

 Connection Status: Up
 Connection Status time stamp:  Mon Jun  1 19:23:18 2026

 Connection Prev Status: Init
 Connection Prev Status time stamp:  Mon Jun  1 19:20:48 2026

 Association status: Active
 Association status time stamp: Mon Jun  1 19:23:18 2026

 GTPu socket bind rc: Success
 PFCP socket bind rc: Success

[NAL-CP-INST-EVENTS]

| Event Name                    | Time Stamp              | User Data
| Association Active            | Jun  1 19:23:18.092553  | -- NA --
| Transport Up                  | Jun  1 19:23:18.093636  | -- NA --
```

Expected result: Each UP instance reports an **ESTABLISHED** state with its assigned CP group. If state is **FAILED** or **ATTEMPTING**, proceed to troubleshooting.

Step 2 Verify SRG group assignments and operational roles.

Example:

```
Router# show cnbng-nal srg-group

Tue Jun  2 08:25:37.200 IST
====================
Location: 0/0/CPU0
====================

Group-name   SRG role  Access OT  Core OT  Subs Count V4 routes V6 routes UP Inst ID
-------------------------------------------------------------------------------

Router# show cnbng-nal srg-group up-instance-id 1

Tue Jun  2 08:25:41.905 IST
====================
Location: 0/0/CPU0
====================

Group-name  SRG role  Access OT  Core OT  Subs Count V4 routes V6 routes UP Inst ID
-------------------------------------------------------------------------------
group1       Active    Up         Up       1          2         2         1

Total Entries : 1

Summary
-------
Category                  Total      Active     Standby    None
-----------------------------------------------------------------
Groups                    1          1          0          0
Subscribers               1          1          0          0
V4 subnet routes          2          2          0          0
V6 subnet routes          2          2          0          0

Router# show cnbng-nal srg-group detail up-instance-id 1

Tue Jun  2 08:25:55.846 IST
====================
Location: 0/0/CPU0
```

```
====================

SRG group name                    : group1
SRG group admin state             : UP_CP_Configured
SRG group state                   : Up
SRG role                          : Active
SRG ID                            : 0x00000fa1
UP Instance ID                    : 1
SRG VRF name                      : **srg_4001
SRG mode                          : HOT
Last SRG role update time         : Jun 01 19:28:26.046033
Virtual mac                       : 0AAA.0BBB.0C01
V4 Table Id                       : 0xe0000012
V6 Table Id                       : 0xe0800012
V4 Proto Id                       : 0x00000002
V6 Proto Id                       : 0x00000001
Subscriber count                  : 1
Warm Subscriber count             : 0
IPV4 route count                  : 2
IPV6 route count                  : 2
Damping timer interval            : 120 Sec
Subnet route tag                  : 0
Route export on Standby enable    : False
Fast switchover enable            : True
Ready for role change             : Yes [Success]
FSM State                         : COMPLETE
Update Request State              : IDLE
Stats state                       : IDLE
Sub disconnect resp pend          : NA
Sub count for keep alive start    : 0
Group for routed subscribers      : FALSE

Access tracking object
----------------------
    Object name                     : track1
    Tracking state                  : Up
    Last tracking state update time : Jun 01 19:26:49.537142

Core tracking object
--------------------
    Object name                     : core1
    Tracking state                  : Up
    Last tracking state update time : Jun 01 19:26:49.537217

Access Interfaces
-----------------
    Bundle-Ether1.1

IM counters
------------
    Total errors                    : 0

RIB counters
------------
    Total entries                   : 0
    Pending                         : 0
    Total errors                    : 0

Terminate counters
------------------
    IM terminate                    : 0
    RIB terminate                   : 0
    Stats terminate                 : 0
```

```
Flags
-----
Value: [0x0000007C]

 [VRF_CREATE | V4_RIB_CREATE | V6_RIB_CREATE |
  V4_RIB_COMPLETE | V6_RIB_COMPLETE]

Checkpoint Flags
----------------
Value: [0x00000000]
 None

CP Recon data
-------------
    Duration                        : 0 secs
    Replay reqs in progress         : 0
    Replay subs in progress         : 0
    CP Recon Flags                  : 0x0

Subscriber transaction Info
---------------------------
    Subscribers in transaction      : 0
    Subscribers in AF down queue    : 0
    Subscribers in disc queue       : 0

Group role switchover stats Info
--------------------------------
    Last stats interaction time(A->S)   : 1.060 secs
    Last stats interaction time(S->A)   : 0.060 secs
    Max stats interaction time(A->S)    : 1.060 secs (Jun 01 19:25:40.854170)
    Max stats interaction time(S->A)    : 0.060 secs (NA)

Group role switchover RIB Info
------------------------------
    Last rib interaction time(A->S)     : 0.06103 secs
    Last rib interaction time(S->A)     : 0.060 secs
    Max rib interaction time(A->S)      : 0.06103 secs (Jun 01 19:25:40.853550)
    Max rib interaction time(S->A)      : 0.060 secs (NA)

    Active role change count            : 9
    Standby role change count           : 8

[SRG-GROUP-EVENTS]

| Event Name                              | Time Stamp              | User Data
| Group create                            | Jun  1 19:22:58.093098  | -- NA --
| V4 backup vrf create                    | Jun  1 19:22:58.093283  | -- NA --
| V6 backup vrf create                    | Jun  1 19:22:58.093311  | -- NA --
| V4 backup vrf create rsp                | Jun  1 19:22:58.109535  | -- NA --
| V4 backup rib tbl create                | Jun  1 19:22:58.109582  | -- NA --
| V6 backup vrf create rsp                | Jun  1 19:22:58.109585  | -- NA --
| V6 backup rib tbl create                | Jun  1 19:22:58.109593  | -- NA --
| V4 backup rib tbl create rsp            | Jun  1 19:22:58.112594  | -- NA --
| V6 backup rib tbl create rsp            | Jun  1 19:22:58.211073  | -- NA --
| State update held, transport down       | Jun  1 19:24:07.182998  | -- NA --
| Role none                               | Jun  1 19:24:15.954512  | -- NA --
| CP action add                           | Jun  1 19:24:29.121120  | -- NA --
| Subs rib route deletion started         | Jun  1 19:25:40.853550  | -- NA --
| Subs rib route deletion completed       | Jun  1 19:25:40.957261  | -- NA --
| Fast swo timer expiry, Role Active      | Jun  1 19:25:56.235268  | -- NA --
| Notify: State Down                      | Jun  1 19:26:39.621205  |
in_prog:F,comp:T,role:stdby,state:down
| Role standby                            | Jun  1 19:28:04.552720  | -- NA --
| Role standby start                      | Jun  1 19:28:04.552762  | -- NA --
```

```
| Role standby end                          | Jun  1 19:28:04.553567  | -- NA --
| Role active                               | Jun  1 19:28:26.046033  | -- NA --
| Role active start                         | Jun  1 19:28:26.046107  | -- NA --
| Role active end                           | Jun  1 19:28:26.046878  | -- NA --
| Notify: State Up                          | Jun  1 19:28:31.046323  |
in_prog:F,comp:T,role:act,state:up
| State change ack'ed                       | Jun  1 19:28:31.046495  | -- NA --
| Sent srg role response to CP              | Jun  1 19:28:37.642414  |
in_prog:F,comp:T,role:act,state:up

======================================================================

Total Entries : 1
```

```
Router# show cnbng-nal srg-group detail up-instance-id all

Tue Jun  2 08:25:58.913 IST
====================
Location: 0/0/CPU0
====================

SRG group name                    : group1
SRG group admin state             : UP_CP_Configured
SRG group state                   : Up
SRG role                          : Active
SRG ID                            : 0x00000fa1
UP Instance ID                    : 1
SRG VRF name                      : **srg_4001
SRG mode                          : HOT
Last SRG role update time         : Jun 01 19:28:26.046033
Virtual mac                       : 0AAA.0BBB.0C01
V4 Table Id                       : 0xe0000012
V6 Table Id                       : 0xe0800012
V4 Proto Id                       : 0x00000002
V6 Proto Id                       : 0x00000001
Subscriber count                  : 1
Warm Subscriber count             : 0
IPV4 route count                  : 2
IPV6 route count                  : 2
Damping timer interval            : 120 Sec
Subnet route tag                  : 0
Route export on Standby enable    : False
Fast switchover enable            : True
Ready for role change             : Yes [Success]
FSM State                         : COMPLETE
Update Request State              : IDLE
Stats state                       : IDLE
Sub disconnect resp pend          : NA
Sub count for keep alive start    : 0
Group for routed subscribers      : FALSE

Access tracking object
----------------------
    Object name                     : track1
    Tracking state                  : Up
    Last tracking state update time : Jun 01 19:26:49.537142

Core tracking object
--------------------
    Object name                     : core1
    Tracking state                  : Up
    Last tracking state update time : Jun 01 19:26:49.537217

Access Interfaces
-----------------
```

```
    Bundle-Ether1.1

IM counters
------------
    Total errors                    : 0

RIB counters
------------
    Total entries                   : 0
    Pending                         : 0
    Total errors                    : 0

Terminate counters
------------------
    IM terminate                    : 0
    RIB terminate                   : 0
    Stats terminate                 : 0

Flags
-----
Value: [0x0000007C]

 [VRF_CREATE | V4_RIB_CREATE | V6_RIB_CREATE |
  V4_RIB_COMPLETE | V6_RIB_COMPLETE]

Checkpoint Flags
----------------
Value: [0x00000000]
 None

CP Recon data
-------------
    Duration                        : 0 secs
    Replay reqs in progress         : 0
    Replay subs in progress         : 0
    CP Recon Flags                  : 0x0

Subscriber transaction Info
---------------------------
    Subscribers in transaction      : 0
    Subscribers in AF down queue    : 0
    Subscribers in disc queue       : 0

Group role switchover stats Info
--------------------------------
    Last stats interaction time(A->S)   : 1.060 secs
    Last stats interaction time(S->A)   : 0.060 secs
    Max stats interaction time(A->S)    : 1.060 secs (Jun 01 19:25:40.854170)
    Max stats interaction time(S->A)    : 0.060 secs (NA)

Group role switchover RIB Info
------------------------------
    Last rib interaction time(A->S)     : 0.06103 secs
    Last rib interaction time(S->A)     : 0.060 secs
    Max rib interaction time(A->S)      : 0.06103 secs (Jun 01 19:25:40.853550)
    Max rib interaction time(S->A)      : 0.060 secs (NA)

    Active role change count            : 9
    Standby role change count           : 8

[SRG-GROUP-EVENTS]

| Event Name                            | Time Stamp              | User Data
| Group create                          | Jun  1 19:22:58.093098  | -- NA --
```

```
| V4 backup vrf create                     | Jun  1 19:22:58.093283  | -- NA --
| V6 backup vrf create                     | Jun  1 19:22:58.093311  | -- NA --
| V4 backup vrf create rsp                 | Jun  1 19:22:58.109535  | -- NA --
| V4 backup rib tbl create                 | Jun  1 19:22:58.109582  | -- NA --
| V6 backup vrf create rsp                 | Jun  1 19:22:58.109585  | -- NA --
| V6 backup rib tbl create                 | Jun  1 19:22:58.109593  | -- NA --
| V4 backup rib tbl create rsp             | Jun  1 19:22:58.112594  | -- NA --
| V6 backup rib tbl create rsp             | Jun  1 19:22:58.211073  | -- NA --
| State update held, transport down        | Jun  1 19:24:07.182998  | -- NA --
| Role none                                | Jun  1 19:24:15.954512  | -- NA --
| CP action add                            | Jun  1 19:24:29.121120  | -- NA --
| Subs rib route deletion started          | Jun  1 19:25:40.853550  | -- NA --
| Subs rib route deletion completed        | Jun  1 19:25:40.957261  | -- NA --
| Fast swo timer expiry, Role Active       | Jun  1 19:25:56.235268  | -- NA --
| Notify: State Down                       | Jun  1 19:26:39.621205  |
in_prog:F,comp:T,role:stdby,state:down
| Role standby                             | Jun  1 19:28:04.552720  | -- NA --
| Role standby start                       | Jun  1 19:28:04.552762  | -- NA --
| Role standby end                         | Jun  1 19:28:04.553567  | -- NA --
| Role active                              | Jun  1 19:28:26.046033  | -- NA --
| Role active start                        | Jun  1 19:28:26.046107  | -- NA --
| Role active end                          | Jun  1 19:28:26.046878  | -- NA --
| Notify: State Up                         | Jun  1 19:28:31.046323  |
in_prog:F,comp:T,role:act,state:up
| State change ack'ed                      | Jun  1 19:28:31.046495  | -- NA --
| Sent srg role response to CP             | Jun  1 19:28:37.642414  |
in_prog:F,comp:T,role:act,state:up

======================================================================

SRG group name                    : group2
SRG group admin state             : UP_Configured
SRG group state                   : Up
SRG role                          : None
SRG ID                            : 0x00000fa2
UP Instance ID                    : 2
SRG VRF name                      : -NA- (fast-switchover disabled)
SRG mode                          : HOT
Last SRG role update time         : Jun 01 19:24:15.954725
Virtual mac                       : 0AAA.0BBB.0A02
V4 Table Id                       : 0x00000000
V6 Table Id                       : 0x00000000
V4 Proto Id                       : 0x0000ffff
V6 Proto Id                       : 0x0000ffff
Subscriber count                  : 0
Warm Subscriber count             : 0
IPV4 route count                  : 0
IPV6 route count                  : 0
Damping timer interval            : 120 Sec
Subnet route tag                  : 0
Route export on Standby enable    : False
Fast switchover enable            : False
Ready for role change             : Yes [Success]
FSM State                         : UNKNOWN
Update Request State              : IDLE
Stats state                       : IDLE
Sub disconnect resp pend          : NA
Sub count for keep alive start    : 0
Group for routed subscribers      : FALSE

Access tracking object
----------------------
    Object name                     : track1
    Tracking state                  : Up
```

```
    Last tracking state update time : Jun 01 19:26:49.537187

Core tracking object
--------------------
    Object name                     : core1
    Tracking state                  : Up
    Last tracking state update time : Jun 01 19:26:49.537261

Access Interfaces
-----------------
    Bundle-Ether1.2

IM counters
------------
    Total errors                    : 0

RIB counters
------------
    Total entries                   : 0
    Pending                         : 0
    Total errors                    : 0

Terminate counters
------------------
    IM terminate                    : 0
    RIB terminate                   : 0
    Stats terminate                 : 0

Flags
-----
Value: [0x00000000]
 None

Checkpoint Flags
----------------
Value: [0x00000048]

 [STATE_RESP_PENDING | ROLE_CHG_IN_PROG]

CP Recon data
-------------
    Duration                        : 0 secs
    Replay reqs in progress         : 0
    Replay subs in progress         : 0
    CP Recon Flags                  : 0x0

Subscriber transaction Info
---------------------------
    Subscribers in transaction      : 0
    Subscribers in AF down queue    : 0
    Subscribers in disc queue       : 0

Group role switchover stats Info
--------------------------------
    Last stats interaction time(A->S)   : 0.060 secs
    Last stats interaction time(S->A)   : 0.060 secs
    Max stats interaction time(A->S)    : 0.060 secs (NA)
    Max stats interaction time(S->A)    : 0.060 secs (NA)

Group role switchover RIB Info
------------------------------
    Last rib interaction time(A->S)     : 0.060 secs
    Last rib interaction time(S->A)     : 0.060 secs
    Max rib interaction time(A->S)      : 0.060 secs (NA)
```

```
    Max rib interaction time(S->A)      : 0.060 secs (NA)

    Active role change count            : 0
    Standby role change count           : 1

[SRG-GROUP-EVENTS]

| Event Name                              | Time Stamp              | User Data
| Group create                            | Jun  1 19:22:58.095173  | -- NA --
| V4 backup vrf create                    | Jun  1 19:22:58.095296  | -- NA --
| V6 backup vrf create                    | Jun  1 19:22:58.095324  | -- NA --
| Role standby start                      | Jun  1 19:24:07.754079  | -- NA --
| Role standby                            | Jun  1 19:24:07.754079  | -- NA --
| Role standby end                        | Jun  1 19:24:07.755454  | -- NA --
| CP action add                           | Jun  1 19:24:07.755455  | -- NA --
| Sent srg role response to CP            | Jun  1 19:24:07.755510  |
in_prog:T,comp:F,role:stdby,state:up
| State update held, transport down       | Jun  1 19:24:07.756603  | -- NA --
| Role none                               | Jun  1 19:24:15.954725  | -- NA --

=========================================================================

Total Entries : 2
```

Check for correct mapping to UP instances and proper Active/Standby roles.

Step 3 Verify dynamic route distribution across UP instances.

Example:

```
Router# show cnbng-nal dynamic-routes summary up-instance-id all

Tue Jun  2 08:26:15.612 IST

Location: 0/0/CPU0
-----------------------

Counter Name             Value
---------------------------------
V4 OC Entries            2
V6 OC Entries            2
V4 Primary Entries       0
V4 Secondary Entries     2
V4 RIB Entries           2
V6 RIB Entries           2
OC replay entry count    0
V6 OC replay entry count 0
V6 Enabled Entries       0


Router# show cnbng-nal dynamic-routes summary up-instance-id 1

Tue Jun  2 08:26:23.611 IST
Location: 0/0/CPU0
-----------------------
Counter Name             Value
---------------------------------
V4 OC Entries            2
V6 OC Entries            2
V4 Primary Entries       0
V4 Secondary Entries     2
V4 RIB Entries           2
V6 RIB Entries           2
OC replay entry count    0
V6 OC replay entry count 0
```

```
V6 Enabled Entries       0
```

Routes must be correctly distributed per instance. Look for overlaps or missing entries.

Step 4 Verify subscriber session distribution and status.

Example:

```
Router# show cnbng-nal subscriber all up-instance-id all

Tue Jun  2 08:26:39.880 IST

====================
Location: 0/0/CPU0
====================
Codes: CN - Connecting, CD - Connected, AC - Activated,
       ID - Idle, DN - Disconnecting, IN - Initializing
       UN - Unknown


CPID(hex)  Interface              State  Mac Address     Subscriber IP Addr / Prefix (Vrf) Ifhandle
-------------------------------------------------------------------------------------------------
d2         BE1.1.ip3221227008      AC     001b.0000.00d2  100.0.0.210 (default) 0x2b0
                                                          200.0.210.0/24 (Framed IPv4)
                                                          201.0.210.0/24 (Framed IPv4)
                                                          202.0.210.0/24 (Framed IPv4)
                                                          203.0.210.0/24 (Framed IPv4)
                                                          cafe:feed::d2 (IANA)
                                                          cafe:bead:0:d2::/128 (IAPD)
                                                          1:cee:bad:c0de:0:d2::/96 (Framed IPv6)
                                                          1:cee:bad:c0de:100:d2::/96 (Framed IPv6)
                                                          1:cee:bad:c0de:200:d2::/96 (Framed IPv6)
                                                          1:cee:bad:c0de:300:d2::/96 (Framed IPv6)
Session-count: 1

Router# show cnbng-nal subscriber all up-instance-id 1

Tue Jun  2 08:26:42.928 IST
====================
Location: 0/0/CPU0
====================
Codes: CN - Connecting, CD - Connected, AC - Activated,
       ID - Idle, DN - Disconnecting, IN - Initializing
       UN - Unknown


CPID(hex)  Interface              State  Mac Address     Subscriber IP Addr / Prefix (Vrf) Ifhandle
-------------------------------------------------------------------------------------------------
d2         BE1.1.ip3221227008      AC     001b.0000.00d2  100.0.0.210 (default) 0x2b0
                                                          200.0.210.0/24 (Framed IPv4)
                                                          201.0.210.0/24 (Framed IPv4)
                                                          202.0.210.0/24 (Framed IPv4)
                                                          203.0.210.0/24 (Framed IPv4)
                                                          cafe:feed::d2 (IANA)
                                                          cafe:bead:0:d2::/128 (IAPD)
                                                          1:cee:bad:c0de:0:d2::/96 (Framed IPv6)
                                                          1:cee:bad:c0de:100:d2::/96 (Framed IPv6)
                                                          1:cee:bad:c0de:200:d2::/96 (Framed IPv6)
                                                          1:cee:bad:c0de:300:d2::/96 (Framed IPv6)
Session-count: 1

Router# show cnbng-nal subscriber all summary up-instance-id all

Tue Jun  2 08:26:55.300 IST
====================
```

```
Location: 0/0/CPU0
======================

                Type            PPPoE                   IPoE                    LAC             LNS
                ====            =====                   ====                    ===             ===
                                                    L2-Conn         Routed
                                                    =======         ======

Session Counts by State:
        initializing  0  0  0  0  0
          connecting  0  0  0  0  0
           connected  0  0  0  0  0
           activated  0  1  0  0  0
                idle  0  0  0  0  0
       disconnecting  0  0  0  0  0
               Total:  0  1  0  0  0


Session Counts by Address-Family:
                none  0  0  0  0  0
                ipv4  0  0  0  0  0
                ipv6  0  0  0  0  0
                dual  0  1  0  0  0
                 lac  0  0  0  0  0
              Total:  0  1  0  0  0

Router# show cnbng-nal subscriber all summary up-instance-id 1

Tue Jun  2 08:26:58.062 IST
======================
Location: 0/0/CPU0
======================

                Type            PPPoE                   IPoE                    LAC             LNS
                ====            =====                   ====                    ===             ===
                                                    L2-Conn         Routed
                                                    =======         ======

Session Counts by State:
        initializing  0  0  0  0  0
          connecting  0  0  0  0  0
           connected  0  0  0  0  0
           activated  0  1  0  0  0
                idle  0  0  0  0  0
       disconnecting  0  0  0  0  0
               Total:  0  1  0  0  0


Session Counts by Address-Family:
                none  0  0  0  0  0
                ipv4  0  0  0  0  0
                ipv6  0  0  0  0  0
                dual  0  1  0  0  0
                 lac  0  0  0  0  0
              Total:  0  1  0  0  0

Router# show cnbng-nal subscriber all detail up-instance-id 1

Tue Jun  2 08:27:21.590 IST
======================
Location: 0/0/CPU0
======================
Interface:                Bundle-Ether1.1.ip3221227008
```

```
UPID:                       0xc0000600
CPID:                       0x000000d2
UP Instance ID:             1
Type:                       IPoE
  Sub-type:                 L2 Connected
IPv4 Address:               100.0.0.210
IPv4 Framed Route:
  Prefix:                   200.0.210.0/24
  Next Hop:                 3.1.1.1
  Tag:                      0
  Metric:                   0
  Distance:                 0
  Next Hop VRF:
  Prefix:                   201.0.210.0/24
  Next Hop:                 3.1.1.1
  Tag:                      0
  Metric:                   0
  Distance:                 0
  Next Hop VRF:
  Prefix:                   202.0.210.0/24
  Next Hop:                 3.1.1.1
  Tag:                      0
  Metric:                   0
  Distance:                 0
  Next Hop VRF:
  Prefix:                   203.0.210.0/24
  Next Hop:                 3.1.1.1
  Tag:                      0
  Metric:                   0
  Distance:                 0
  Next Hop VRF:
IPv6 IANA Address:          cafe:feed::d2
IPv6 IAPD Prefix:           cafe:bead:0:d2::/128
IPv6 Slaac Prefix:          ::/0
CPE link local Address:     ::
IPv6 Framed Route:
  Prefix:                   1:cee:bad:c0de:0:d2::/96
  Next Hop:                 192::1
  Tag:                      0
  Metric:                   0
  Distance:                 0
  Next Hop VRF:
  Prefix:                   1:cee:bad:c0de:100:d2::/96
  Next Hop:                 192::1
  Tag:                      0
  Metric:                   0
  Distance:                 0
  Next Hop VRF:
  Prefix:                   1:cee:bad:c0de:200:d2::/96
  Next Hop:                 192::1
  Tag:                      0
  Metric:                   0
  Distance:                 0
  Next Hop VRF:
  Prefix:                   1:cee:bad:c0de:300:d2::/96
  Next Hop:                 192::1
  Tag:                      0
  Metric:                   0
  Distance:                 0
  Next Hop VRF:
IPv4 State:                 UP, Mon Jun  1 19:28:40 2026
IPv6 State:                 UP, Mon Jun  1 19:28:40 2026
Mac Address:                001b.0000.00d2
Inner VLAN ID:              Not Set
```

```
Outer VLAN ID:             1
Outer VLAN Cos:            0
Outer VLAN DEI:            0
Created:                   Mon Jun  1 19:28:40 2026
State:                     Activated
Ifhandle:                  0x000002b0
VRF:                       default
Access-interface:          Bundle-Ether1.1
SRG role:                  Active
SRG group name:            group1
IPOE-AFI-Cap:              SUB_AFI_IPV4_IPV6
Loopback-name:             Loopback0
IPV4-GW-addr:              100.0.0.1
 Attribute List: 0xdade20
1:  ipv4-unnumbered len=  9  value= Loopback0
2:  ipv6-enable     len=  4  value= 1(1)
3:  sub-ipv4-gateway len= 12  value= 100.0.0.1/32

Session Accounting:       disabled
Interim Interval:          0 secs
Interim fail count: None
Last interim failed reason: NA
Last stats:
  BytesIn: 0
  BytesOut: 0
  BytesInGiga: 0
  BytesOutGiga: 0

Activated Services: NA
Feature IDs activated : NA

[NAL-SUBSCRIBER-EVENTS]

| Event Name                    | Time Stamp                | User Data
| Transaction start             | Jun  1 19:28:40.024840  | txn id: 0x33, src_port: 0x33
| Interface create req          | Jun  1 19:28:40.024913  | -- NA --
| Interface create resp         | Jun  1 19:28:40.087462  | rc: OK
| SVM create req                | Jun  1 19:28:40.087533  | -- NA --
| Sub GW addr add               | Jun  1 19:28:40.087547  | addr: 100.0.0.1/32
| UP Install req                | Jun  1 19:28:40.087598  | -- NA --
| UP Install resp               | Jun  1 19:28:40.213931  | rc: OK
| UP Assoc req                  | Jun  1 19:28:40.213958  | -- NA --
| UP Assoc resp                 | Jun  1 19:28:40.215440  | rc: OK
| Produce done req              | Jun  1 19:28:40.215456  | req id: 0x3
| Produce done - Cfg gen        | Jun  1 19:28:40.268242  | rc: OK, req id: 0x3
| Produce done resp             | Jun  1 19:28:40.327444  | rc: OK, req id: 0x3
| SVM create resp               | Jun  1 19:28:40.327446  | rc: OK
| IPv6 Caps Up                  | Jun  1 19:28:40.328977  | vrf table id: 0xe0800000
| IPv4 caps wait                | Jun  1 19:28:40.328978  | -- NA --
| IPv4 Caps Up                  | Jun  1 19:28:40.329475  | vrf table id: 0xe0000000
| Init data req                 | Jun  1 19:28:40.329947  | -- NA --
| Init data resp                | Jun  1 19:28:40.383178  | rc: OK
| IPv4 route req                | Jun  1 19:28:40.383188  | addr: 100.0.0.210/32
| IPv4 framed route req         | Jun  1 19:28:40.383250  | addr: 203.0.210.0/24
| IPv6 route req                | Jun  1 19:28:40.383262  | addr: cafe:feed::d2/128
| IPv6 PD route req             | Jun  1 19:28:40.383336  | addr: cafe:bead:0:d2::/128
| IPv6 framed route req         | Jun  1 19:28:40.383361  | addr: 1:cee:bad:c0de:300:d2::/96
| IPv4 route resp               | Jun  1 19:28:40.484099  | rc: OK
| IPv4 framed route resp        | Jun  1 19:28:40.484182  | rc: OK, addr: 203.0.210.0/24
| IPv6 route resp               | Jun  1 19:28:40.484195  | rc: OK
| IPv6 PD route resp            | Jun  1 19:28:40.484215  | rc: OK
| IPv6 framed route resp        | Jun  1 19:28:40.484241  | rc: OK, addr: 1:cee:bad:c0de:300:d2::/96
| IPv4 Up                       | Jun  1 19:28:40.484247  | -- NA --
| IPv6 ND session req           | Jun  1 19:28:40.484300  | -- NA --
```

```
| IPv6 ND session resp           | Jun  1 19:28:40.484617  | rc: OK
| IPv6 Up                        | Jun  1 19:28:40.484618  | -- NA --
| Session up notified            | Jun  1 19:28:40.484620  | -- NA --
| Transaction stop               | Jun  1 19:28:40.484869  | rc: OK, txn id: 0x33, src_port: 0x33
==================__#__======================


Session-count: 1
```

Sessions should align with expected UP instance and SRG group associations.

Step 5 Review PFCP message, reconciliation, and notification history.

Example:

```
Router# show cnbng-nal spa pfcp-api structure dump all up-instance-id 1

Tue Jun  2 08:27:51.777 IST
Entry               : 1 (NAL Response for session create request),
Timestamp           : Jun 01 19:28:40.484898,
UPID                : 0xc0000600
CPID                : 0xd2
Session Txn id      : 261
Return value        : SPA_RC_SUCCESS
Access Interface    : Bundle-Ether1.1
Router name         :
Error message       :

Up-Instance-ID      : 1
================================================================================
Entry               : 2 (SPA Create Request),
Timestamp           : Jun 01 19:28:40.024568,
UPID                : 0x00000000
CPID                : 0x000000d2
Msg Txn id          : 51
Source port id      : 51
Session Txn id      : 261
Oper code           : SPA_OPER_TYPE_CREATE
IPOE AFI Cap        : SPA_SUB_AFI_IPV4_IPV6

Feature mask        : 0x1ee00
SRG report id       : 0
Feature flags
-------------
[SPA_FEATURE_IPv4_ADDR | SPA_FEATURE_IANA_ADDR | SPA_FEATURE_IAPD_ADDR | SPA_FEATURE_V4_ROUTE |
 SPA_FEATURE_V6_ROUTE | SPA_FEATURE_SRG | SPA_FEATURE_OUTER_VLAN]

IPv4 address Info
-----------------
Oper code           : SPA_OPER_TYPE_CREATE
IPv4 address        : 100.0.0.210

IPv6 address Info
-----------------
Oper code           : SPA_OPER_TYPE_CREATE
IPv6 address(iana)  : cafe:feed::d2
IPv6 address(iapd)  : cafe:bead:0:d2::
Link local          : ::
Prefix length       : 128

Session Data
------------
Access interface    : Bundle-Ether1.1
Mac address         : 001b.0000.00d2
VRF name            :
```

```
Session type        : SPA_SESSION_TYPE_IPOE
SRG group name      : group1

    Vlan Info
    ---------
    Outer vlan          : 1

IPv4 route Info
---------------
Oper code           : SPA_OPER_TYPE_CREATE
VRF name            : default
IPv4 address        : 200.0.210.0
Prefix length       : 24
Next hop            : 3.1.1.1
Tag                 : 0
Metric              : 0
Next hop VRF name   :
Distance            : 0

IPv4 route Info
---------------
Oper code           : SPA_OPER_TYPE_CREATE
VRF name            : default
IPv4 address        : 201.0.210.0
Prefix length       : 24
Next hop            : 3.1.1.1
Tag                 : 0
Metric              : 0
Next hop VRF name   :
Distance            : 0

IPv4 route Info
---------------
Oper code           : SPA_OPER_TYPE_CREATE
VRF name            : default
IPv4 address        : 202.0.210.0
Prefix length       : 24
Next hop            : 3.1.1.1
Tag                 : 0
Metric              : 0
Next hop VRF name   :
Distance            : 0

IPv4 route Info
---------------
Oper code           : SPA_OPER_TYPE_CREATE
VRF name            : default
IPv4 address        : 203.0.210.0
Prefix length       : 24
Next hop            : 3.1.1.1
Tag                 : 0
Metric              : 0
Next hop VRF name   :
Distance            : 0

IPv6 route Info
---------------
Oper code           : SPA_OPER_TYPE_CREATE
VRF name            : default
IPv6 address        : 1:cee:bad:c0de:0:d2::
Prefix length       : 96
Next hop            : 192::1
Tag                 : 0
Metric              : 0
```

```
Next hop VRF name   :
Distance            : 0

IPv6 route Info
---------------
Oper code           : SPA_OPER_TYPE_CREATE
VRF name            : default
IPv6 address        : 1:cee:bad:c0de:100:d2::
Prefix length       : 96
Next hop            : 192::1
Tag                 : 0
Metric              : 0
Next hop VRF name   :
Distance            : 0

IPv6 route Info
---------------
Oper code           : SPA_OPER_TYPE_CREATE
VRF name            : default
IPv6 address        : 1:cee:bad:c0de:200:d2::
Prefix length       : 96
Next hop            : 192::1
Tag                 : 0
Metric              : 0
Next hop VRF name   :
Distance            : 0

IPv6 route Info
---------------
Oper code           : SPA_OPER_TYPE_CREATE
VRF name            : default
IPv6 address        : 1:cee:bad:c0de:300:d2::
Prefix length       : 96
Next hop            : 192::1
Tag                 : 0
Metric              : 0
Next hop VRF name   :
Distance            : 0
....
```

```
Router# show cnbng-nal spa notif-history type keep-alive-failure up-instance-id 1

Tue Jun  2 08:28:19.816 IST
---- Total entries: 0 ----
```

```
Router# show cnbng-nal spa notif-history type subscriber-event up-instance-id 1

Tue Jun  2 08:28:29.198 IST
---- Total entries: 0 ----
```

Identify unusual message logs or recon reports that could indicate communication or failover issues.

Step 6 Check UP instance process information and flow control status.

Example:

```
Router# show cnbng-nal process-info detail

Tue Jun  2 08:28:38.242 IST
Location: 0/0/CPU0

 HA Pre_Init Role    : PRIMARY
 HA Role     : PRIMARY
 Restart-flag     : FALSE
 Node-Id     : 0
 Disc-Hist File-logging    : Enable
```

```
Test-server config-enabled: FALSE

Proc-flags : 87FFFF9F

   OT Connection Status: UP
   IM Connection Status: UP
   IPv4 RIB Connection Status: UP
   IPv6 RIB Connection Status: UP
   SUBDB Connection Status: UP
   Stats Connection Status: UP
   L2TP FIB Connection Status: DOWN
   L2TP Netio Connection Status: DOWN
   SPIO PPPoE Connection Status: UP
   SPIO PPP Connection Status: UP

SVM SUBDB PDone Req-ID     : 3
SVM Reconcilation Last UPID    : 0x0
SVM Reconcilation Pending-Req count      : 0
SVM Reconcilation Request Max    : 3072
RSI File-Descriptor     : 158
Bulk-Disconnect Start-time    : Tue Jun  2 08:28:38 2026
Clear-session Stats Timer-Running    : FALSE
Clear-feature Stats Timer-Running    : FALSE
Statistics Flags     : 0x00000005
SPA message Local seq-no    : 0
SPA message Global seq-no    : 0
Stats Hold time-out     : 200 secs
Stats Hold Request count    : 0
Stats Hold Request max     : 1000
Stats Stop Request count    : 0
Stats Interim Request count    : 0
Stats Clear Session Request count   : 0
Stats Clear Feature Request count   : 0
Interim Inflight Quota     : 4000
Interim Inflight LWM     : 4000
Pending Interim Queue Depth    : 0
Pending Interim Enqueue count    : 0
Pending Interim Dequeue count    : 0
Pending Interim Drop count    : 0
Pending Interim Dup-Skip count    : 0
SRG Stats Request Q-size    : 0
SRG Stats Retry Q-size     : 0
Sub-context Replay-In-Progess    : 0
Sub-context Cache Q-size    : 1
Sub-context Stale Q-size    : 0

Session stats:
===============
Create in progress: 0      [Peak: 1      Timestamp: Jun 01 19:25:10.448945]
Delete in progress: 0      [Peak: 1      Timestamp: Jun 01 19:25:14.858014]
Update in progress: 0      [Peak: 0      Timestamp: -NA-]
Total in progress:  0      [Peak: 1      Timestamp: Jun 01 19:25:10.448945]

CP Recon data:
===============
Duration                  : 300 secs
Replay reqs in progress   : 0
Replay subs in progress   : 0
Flags                     : 0x6

Notification flow control data:
================================
Sub disconnect notification count       : 0
LCP timeout notification count          : 0
```

```
Sub disconnect notification drop count  : 0
LCP timeout notification drop count     : 0
Timer expiry count                      : 0
Timer running                           : No
LCP outstanding ka notification count   : 0
Notification global rate limit          : 1200

Stats flow control data:
========================
Interim stats message count      : 0
SRG stats message count          : 0
Stats oustanding message count   : 0
Stats global rate limit          : 1250

SRG Stats flow control data
===========================
Max outstanding requests  : 600
In-flight requests        : 0
Peak requests             : 0
Peak timestamp            : -NA-

Inflight flow configurations
============================
Maximum creates in-flight      : 600
Maximum updates in-flight      : 1000
Maximum deletes in-flight      : 3000
Maximum warm create in-flight  : 100
Delete batch timer             : 400
Delete batch size              : 800

Warm role change inflight data
==============================
Warm create in-progress        : 0

Features enabled on platform:
=============================
IPoE L2 Terminated        : ENABLED
PPPoE-PTA                 : ENABLED
L2TP-LAC                  : ENABLED
L2TP-LNS                  : ENABLED

Standby group cached SPA request statistics
===========================================
Max cache requests        : 200
Inflight count            : 0
Create cache count        : 0
Update cache count        : 0

UP Instance specific statistics:
================================================

UP Instance ID                         : 1
Notification rate limit                : 20
Stats rate limit                       : 20
Interim stats message count            : 0
SRG stats message count                : 0
LCP outstanding ka notification count  : 0
Stats outstanding message count        : 0
Create outstanding message count       : 0
Update outstanding message count       : 0
Delete outstanding message count       : 0
Active session count                   : 1

UP Instance ID                         : 2
```

```
Notification rate limit               : 20
Stats rate limit                      : 20
Interim stats message count           : 0
SRG stats message count               : 0
LCP outstanding ka notification count  : 0
Stats outstanding message count       : 0
Create outstanding message count      : 0
Update outstanding message count      : 0
Delete outstanding message count      : 0
Active session count                  : 0

========================================
```

Investigate abnormal states or threshold limits that could signal resource or configuration issues.

Step 7 If any UP association is not in **ESTABLISHED** state, validate critical configuration parameters:

- Confirm `hostidentifier` values match on both UP and CP for each instance.
- Validate that `cp-server primary` address, `up-server IP`, and SRG mapping are unique and correct for each instance.
- Check for port or IP conflicts, missing exit commands, or duplicate values.

Expected result: The instance re-associates after correcting discrepancies.

Step 8 If association or SRG mapping fails repeatedly, verify CP connectivity and logs.

- Ping `cp-server` and `up-server` addresses to confirm reachability.
- Check system and debug logs for errors (for example, PFCP handshake failures).
- Ensure CP node process health and end-to-end connectivity.

Expected result: Connectivity and process issues are identified and resolved, restoring association.

Adding a new CP-UP association without disrupting existing connections

When you need to establish a new CP-UP association while an existing association is already active, this feature allows each CP GR instance to form its own connection to the User Plane.

If an association from CP GR instance 1 is already running, simply configure and bring up the new association for another CP GR instance. There is no need to change or interrupt the existing connection.

The new association can onboard additional SRG groups to provide greater redundancy or scalability. You can introduce the second association at any time so the active association remains unaffected.

Behavior during CP–UP connectivity failure

SRG mode

When a UP detects that it has lost its CP connection, it changes its active SRG groups to standby. If the CP detects that the UP association is inactive, it activates the corresponding groups on the peer UP. Existing subscriber traffic can continue through the peer UP when available and if it has the synchronized session state.

After connectivity is restored, the UP sends accounting statistics buffered during the outage to the CP. The CP adds these statistics to the accounting state and forwards the updated data to RADIUS. The CP and UP reconcile subscriber session state after recovery

Non-SRG mode

Loss of CP–UP connectivity does not affect forwarding of existing sessions on the UP. These sessions continue to use their installed forwarding state.

After connectivity is restored, accumulated statistics are sent to the CP in the next session report. After recovery, session reconciliation addresses events that may have occurred during the outage, such as lease-timer expiry

CP-GR behavior

CP-GR role behavior applies to both SRG and non-SRG modes. A CP–UP connectivity failure does not change CP-GR roles. CP-GR manages its role transitions and failover independently from CP–UP connectivity and SRG state

UP Session Disconnect Notification

Feature Description

The User Plane (UP) Session Disconnect Notification feature enables the UP to send a session disconnect notification to the Control Plane (CP) in the following scenario:

When the UP deletes a session locally following any local event (for example, mark and sweep).

For a list of scenarios that trigger a session disconnect notification, see the *Broadband Network Gateway Configuration Guide for Cisco ASR 9000 Series Routers* here: https://www.cisco.com/c/en/us/support/routers/asr-9000-series-aggregation-services-routers/products-installation-and-configuration-guides-list.html.

This notification carries the final statistics for the session and for all services, if accounting is enabled.

The CP initiates the session disconnect for this session only in the CP. The CP then deletes this session. If the session accounting is enabled for this session, the CP sends an accounting "stop record" message to RADIUS.

The UP deletes the session after initiating the session disconnect request to the CP. Therefore, the CP doesn’t send a session release Packet Forwarding Control Protocol (PFCP) message to the UP.

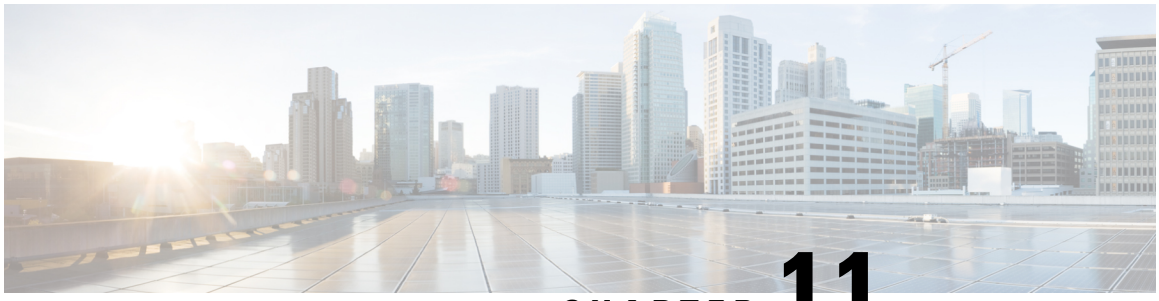
UP Rollback Failure Handling Notification

Feature Description

The User Plane (UP) Rollback Failure Handling Notification feature sends a failure notification to the Control Plane (CP) as follows:

- The CP sends the subscriber configuration in a session modification request to the UP. The UP rolls back the configuration if the application of the configuration fails. The UP deletes the session if the rollback configuration also fails and returns the rollback failure error in the session modification response to the CP.

- The UP sends the final statistics, if applicable, for the session and for each service activated for the failed session, in the session modification response.
- The CP deletes the session in the CP only after receiving a rollback failure in the session modification response from the UP.



CHAPTER 11

DHCP and IPoE Subscriber Management

- Feature Summary and Revision History, on page 215
- Feature Description, on page 216
- Configuring the DHCP and IPoE Subscriber Management Feature, on page 225
- DHCPv6 Raw Option Support , on page 231
- IPv6 Class Configuration and Static IP Allocation Support, on page 234
- Leased IP Hold Time, on page 240
- DHCP IP Lease Reservation, on page 244
- Enhanced support for RADIUS attributes, on page 246

Feature Summary and Revision History

Summary Data

Table 33: Summary Data

Applicable Product(s) or Functional Area	cnBNG
Applicable Platform(s)	SMI
Feature Default Setting	Disabled - Configuration Required
Related Changes in this Release	Not Applicable
Related Documentation	Not Applicable

Revision History

Table 34: Revision History

Revision Details	Release
Enhancement introduced: • Stateful-IPv6-Address-Pool attribute • Delegated-IPv6-Prefix-Pool	2025.02.0
Introduced support for IPv6 Class configuration and Static IP allocation.	2025.01.0
Introduced support for DHCPv6 Raw Option.	2024.04.0
Introduced support for the Leased IP Hold Time feature.	2022.02.0
Enhancement Introduced: The IPoE (DHCP) feature is NSO-integrated.	2021.04.0
First introduced.	2021.01.0

Feature Description

Note This feature is Network Services Orchestrator (NSO) integrated.

A session represents the logical connection between the customer premise equipment (CPE) and the network resource. To enable a subscriber to access the network resources, the network has to establish a session with the subscriber. The Cloud Native Broadband Network Gateway (cnBNG) supports the following subscriber session types:

- IPoE (DHCP)
- PPP (PPPoE)

  For more information, see PPPoE Subscriber Management , on page 399.

In an IPoE subscriber session, subscribers run IPv4 or IPv6 on the CPE device and connect to the BNG through a Layer-2 aggregation or Layer-3 routed network. The IP subscriber sessions that connect through a Layer-2 aggregation network are called L2-connected and sessions that connect through routed access network are called L3-connected or routed subscriber sessions. IPoE subscriber sessions are always terminated on BNG and then routed into the service provider network. IPoE relies on DHCP to assign the IP address.

On the BNG, the DHCPv4 or DHCPv6 trigger creation of these subscribers based on the First-Sign-Of-Life (FSOL) protocol. The IP sessions to the CPE can be either:

- Single stacked, that is, running only IPv4 or IPv6

- Dual stacked, that is, running both IPv4 and IPv6

The DHCP runs as a pod to handle the FSOL for the IPoE subscribers. It handles the DHCP packet encode and decode, IP address assignment, DHCP FSM handling, and DHCP feature and rule application for the IPoE sessions. The DHCP module handles both DHCPv4 and DHCPv6 control packets to bring up corresponding address family interface (AFI).

Note In this release, only the DHCP server mode functionality is supported.

A common DHCP module handles the DHCP finite state machines (FSM) for both 5G subscribers (in SMF service) and wireline subscribers in the cnBNG. The network function (NF) specific DHCP module handles the NF specific functionality.

DHCP and IPoE Functionalities

The DHCP and IPoE Subscriber Management feature supports the following functionalities:

DHCP Server

The cnBNG CP implementation supports the DHCPv4 server mode. The DHCP server FSM handles the DHCP packets from client, IP allocation, and IP lease management.

The FSM handles the following Rx control packets:

- Discover
- Request (DORA request and renew request)
- Decline
- Inform
- Release

The DHCP server FSM sends the following control packets to the client based on the FSM states and events:

- Offer
- Ack (DORA Ack, Renew Ack and Inform Ack)
- Noack

The DHCP server implementation associates a DHCP profile to a group of subscribers. This server implementation supports the following functionalities:

- IP address allocation for the client from the configured pool in the DHCP profile.
- IP address lease allocation based on DHCP profile configuration.
- Passing Host configurations to the client using the following configurable DHCP options in the DHCP profile:
  - IP subnet mask (Option 1)
  - Boot filename (Option 67)

- Domain name (Option 15)
- NetBIOS node type (Option 46)
- NetBIOS name server (Option 44)
- Domain name server (Option 6)
- Default router (Option 3)
- Time server (Option 4)

Processing Option 82

cnBNG supports Option 82, which is the relay agent information option to figure out the sub-options. The various sub-options that the DHCP processes are:

- Circuit ID (Sub option 1)
- Remote ID (Sub option 2)

The circuit ID and remote ID field is passed to the Session Manager during session start trigger and the same is used for north-bound interactions.

DHCPv4 RADIUS Proxy

The cnBNG CP supports DHCP IPv4 RADIUS proxy for RADIUS-based authorization of DHCP leases. This is a RADIUS-based address assignment mechanism in which a DHCP server authorizes remote clients and allocates IP addresses, based on replies from a RADIUS server.

These are the steps involved in the address assignment mechanism:

- The DHCP server sends the DHCP client information to the RADIUS server.
- The RADIUS server returns all required information, primarily IPV4 address, to the DHCP server in the form of RADIUS attributes. The subnet mask is derived from the CP based on the static pool configuration. The IPv4 address sent from the RADIUS must be part of the static pool associated to the UP.
- The DHCP server translates the RADIUS attributes into DHCP options and sends this information back in a DHCP Offer message to the DHCP client.

If the IETF attribute, such as Framed-IP-Address is received from the RADIUS server, and if it is present in the user profile, then this attribute is used instead of allocating the IP address from the configured pool. The basic attributes that can come from the RADIUS server that are relevant for DHCP server options are:

- Framed IPv4 Address
- IPv4 Subnet Mask (derived in the CP from the static pool configuration)
- IPv4 Default gateway (derived in the CP from the static pool configuration)

Apart from these attributes, the dhcp-class name and address pool name attribute also can come from RADIUS. If the RADIUS sets the address pool name, then it uses this for IP allocation instead of the pool that is specified as part of the DHCP profile.

If the RADIUS server sends the dhcp-class attribute to the DHCP server, then that attribute value is used to decide other configuration parameters in the reply that is to be sent to the DHCP client. For example, if the

DHCPv4 server profile has both Class A and Class B in it, and if RADIUS server sends a reply to the DHCP server with the class name as 'B', then Class B is used to send the options back to the DHCP client. Classes can be defined under DHCP profile. The parameters and options that can be configured under DHCP profile can be configured under class also.

Additional RADIUS server attributes are allowed, but not mandatory. If a RADIUS server user profile contains a required attribute that is empty and is not available via configuration as well, the DHCP server does not generate the DHCP options.

DHCPv6 Local Server for IPv6 Subscribers

The DHCPv6 server assigns IPv6 address and prefix and other configuration attributes (such as domain name, the domain name server address and SIP servers and so on) to requesting clients. On receiving a valid request, the server assigns the client IPv6 address or prefix, a lease for the assigned IPv6 address or prefix and other requested configuration parameters. The DHCP server FSM is implemented to handle the address allocation and lease management. The FSM would handle the following control packets from the client:

- Solicit
- Request
- Renew
- Rebind
- Decline
- Information-Request
- Release

The DHCPv6 server FSM sends the following control packets to the client based on the FSM states and events:

- Advertisement
- Reply (SARR Reply, Release Reply, Renew Reply, Rebind Reply and Information request Reply)
- Relay-Reply

The DHCPv6 server implementation associates a DHCPv6 profile to a group of subscribers. The server implementation caters to the following functionalities:

- IANA address and IAPD address allocation for the client from configured pool in DHCPv6 profile.
- IANA and IAPD address lease allocation based on DHCPv6 profile configuration.
- Passing Host configurations to client using below configurable DHCP options in DHCP profile
    - AFTR support (Option 64)
    - Preference option (Option 7)
    - Domain list (Option 24)
    - DNS server IPv6 address (Option 23)

The DHCPv6 server sends the following options to the Policy plane:

- interface-id (DHCP Option 18)

- remote-id (DHCP Option 37)
- vendor-class (DHCP Option 16)
- user-class (DHCP Option 15)
- client-id(DHCP Options 1)

DHCPv6 Server - Prefix Delegation

The DHCPv6 Prefix Delegation feature enables the DHCPv6 server to hand out network address prefixes to the requesting clients. The clients use these network prefixes to assign /128 addresses to the hosts on their network. The RFC-3633 and RFC-3769 is supported for prefix delegation. The DHCPv6 Prefix Delegation feature is enabled by default for cnBNG DHCPv6 server. No other configuration is required to enable the prefix delegation. The DHCPv6 option OPTION_IA_PD (25) and OPTION_IAPREFIX (26) support to meet the prefix delegation requirement.

Note

- Only one delegated prefix per subscriber and client is supported.
- Only one OPTION_IAPREFIX is supported under one OPTION_IA_PD (25).

The cnBNG allocates addresses from the prefix pool configured under the DHCP profile.

DHCPv6 Server - Address Assignment

The DHCPv6 Address Assignment feature enables the DHCPv6 server to hand out /128 addresses to the clients. The cnBNG DHCPv6 server implementation supports the DHCPv6 OPTION_IA_NA(3) and OPTION IAADDR(5) to enable address assignment to the client.

Note

- Only one delegated prefix per subscriber and client is supported.
- Only one OPTION_IAPREFIX is supported under one OPTION_IA_PD (25).

The cnBNG allocates addresses from the prefix pool configured under the DHCP profile.

Prefix and Address Pool Support for IPv6

The cnBNG supports the configuring of the DHCPv6 address and prefix pool and associating it to the DHCPv4 and DHCPv6 server profiles. The address and prefix ranges is under the pool. cnBNG also supports downloading of the address and prefix pool name via the user profile on a per subscriber basis. The pool name downloaded via user profile is given priority over the pool name association via the DHCPv6 profile.

DHCPv6 Server with RADIUS-based Address Assignment

The cnBNG supports RADIUS-based address assignment, that is, the IANA address is downloaded as part of the user profile and is allocated to the client. Address from the user profile is given priority over the local configuration.

DHCPv6 Server with RADIUS-based Prefix Delegation

The cnBNG supports RADIUS-based prefix assignment, that is, the IAPD address is downloaded as part of the user profile and is allocated to the client. The delegated prefix from the user profile is given priority over the local configuration.

DHCPv6-provided IPv6 address of DNS server for IPv6 Subscribers

The cnBNG CP DHCPv6 server implementation supports the provision of DNS server information to clients via the DNS option (23). It supports a configuration of up to 8 DNS server ipv6 addressees via the DHCPv6 profile. The DHCPv6 server information is downloaded via the user profile on a per subscriber basis. The per subscriber DNS information in the user profile is given priority over the profile configuration.

DHCPv4 DHCPv6 Lease Timeout

The cnBNG CP provides the configuration to set the lease value under the DHCPv4 and DHCPv6 profile. This configuration determines the lease for the IP addresses allocated to the clients.

For DHCPv4 clients, the lease is set in the address time (T) option (option 51). By default, the renewal time is set as (½) * T [option 58] and rebinding time is set as (7/8) * T [option 59]. For DHCPv6 client, the lease is populated in the IA address and IA prefix option for the respective address types. By default, preferred time is set as 0.5 * T and valid time T2 is set as 0.8 * T. By default, renewal time (T1) is set as 0.5 * T and rebinding time T2 is set as 0.8 * T in OPTION_IA_PD.

The cnBNG CP tracks the lease time allocated to the clients. Ideally the client should renew (Renew request) the lease at T1 to extend the lease. If renew is failing, the client uses the rebind (broadcast request message for DHCPv4 and rebind message for DHCPv6). If the cnBNG CP does not receive the lease renewal request from the client, the lease times out after T and the corresponding address is released to the pool and removed from the client session. This can lead to an update or disconnect to the Session Manager based on the other address states. The lease timeout is applicable to both IPv4 and IPv6 addresses.

IPv6 IPoE Sessions

The IPv6 subscribers run the IPv6 from the CPE device to the BNG router and are created using the DHCPv6 protocol. The IPv6 subscribers natively run IPv6 on the CPE device and are connected to the router via a Layer-2 network or through Layer-2 aggregation.

The IPv6 subscribers are supported when they are directly connect to the cnBNG UP or via a Layer-2 aggregator. The cnBNG CP DHCPv6 server treats only DHCPv6 SOLICIT message from the subscriber / client as FSOL (First Sign Of Life) packet in case of IPoE and initiates the subscriber session creation.



Note Routed subscribers are not supported.

Dual Stack IPv6/IPv4 over IPoE

The cnBNG CP supports dual-stack IPoE subscribers, that is, both IPv4 and IPv6 address allocation for the same subscriber. In this release, cnBNG supports up to one IPv4 address, one IANA address, and one IAPD address.

Subscriber Termination over Non-default VRF

The cnBNG CP DHCPv4 and DHCPv6 servers are VRF aware. The DHCPv4 and DHCPv6 servers support the access interface in either default VRF or non-default VRF. The following table shows the VRF combination supported by DHCPv6 server.

Table 35: DHCP Supported VRF Combinations

Client Access Interface	Subscriber Interface	DHCPv6 Supported
Default VRF	Default VRF	Supported
Default VRF	Non-default VRF	Supported
Non-default VRF	Non-default VRF	Supported

DHCPv4 Raw Option Support

The cnBNG DHCP Profile configuration enables the operator to configure specific DHCPv4 options, under the DHCPv4 profile. The option value can range from 1 to 255. The option value can be either an ascii string or a hexadecimal string.

DHCPv4 and DHCPv6 Class Support

The cnBNG DHCP Profile configuration enables the operator to configure classes of DHCP options and to selectively associate them during the session setup. The DHCP Options class are selected based on certain matching DHCP options received from access network against the configured class key parameters. The DHCP Options class can also be selected based on the class name received from Policy plane. The priority is always given to the DHCP class name that the Policy plane provides. However, if the Policy plane does not provide a class name, then class selection depends on the operator-configured key parameters. The operator can configure multiple DHCP option classes for DHCPv4 and DHCPv6 separately.

The DHCP Profile consist of profile elements. Each of the DHCPv4 and DHCPv6 profiles contain the ‘default’ DHCP options list and zero or more classes of DHCP options of corresponding DHCP version.

The DHCPv4 and DHCPv6 Options Class contains a list of DHCP options and the “Match-Info” holds the information about the keys to be matched to select that class. The operator can also specify under Match-Info” the class selection that should match ‘any’ or ‘all’ the key parameters of that class.

If the DHCP Option class does not match an ongoing session or any requested DHCP Options is not found in the selected class, then the requested option is selected from the ‘default’ DHCP Options of that profile.

How it Works

This section provides a brief of how the DHCP and IPoE Subscriber Management feature works.

Call Flows

This section includes the following call flow.

cnBNG IPoE Call Flow

For IPoE session establishment, the BNG User Plane (UP) sends the DHCP packets to the BNG Control Plane (CP) using the GTP-U protocol. The following figure shows the DHCP packet call-flow and session programming between the BNG-UP and BNG-CP for IPoE session establishment.

Figure 12: cnBNG IPoE Call Flow

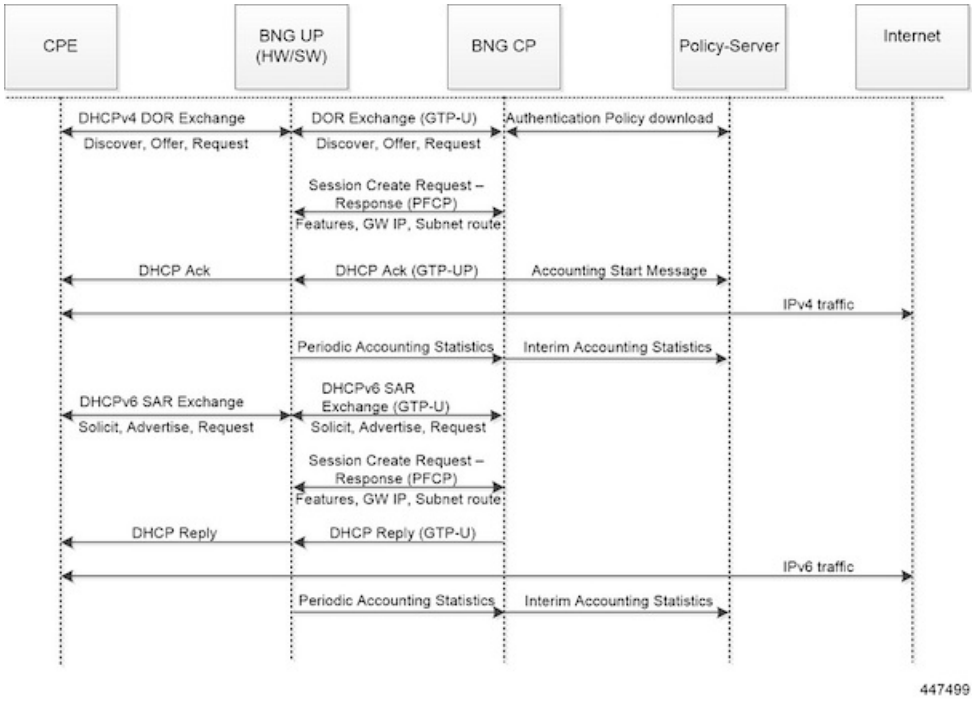


Table 36: cnBNG IPoE Call Flow Description

Step	Description
1	The subscriber running IPv4 or IPv6 stack on the CPE device connects to the BNG-UP via DHCPv4, DHCv6, or DHCPv4 and DHCPv6.
2	The BNG-UP forwards the DHCP(v4/v6) request packets received from the CPE to the BNG-CP over the GTPU protocol. It then returns the DHCP response packets received from the BNG-CP to the CPE device.
3	The BNG-CP performs the subscriber authentication via the Policy Server before establishing a subscriber session on the BNG-UP.
4	After the BNG-CP successfully establishes a session on the BNG-UP, the BNG-UP initiates the Accounting Start and trigger Session Establishment Success (DHCPv4 Ack / DHCPv6 Reply) message towards the CPE via the BNG-UP.
5	The subscriber on the CPE device initiates the data traffic (DHCPv4 / DHCPv6) via the BNG-UP or BNG-CP towards the Internet.
6	The BNG-UP forwards the periodic accounting information to the BNG-CP and the BNG-CP triggers periodic accounting towards the Policy server.

Standard Compliance

The DHCP and IPoE Subscriber Management feature caters to the DHCP server requirements only. The DHCP Server implementation is aligned with the following standards:

- RFC 2131 Dynamic Host Configuration Protocol
- RFC 2132 DHCP Options and BOOTP Vendor Extensions [Subset of options]
- RFC 3046 DHCP Relay Agent Information Option
- RFC 3004 The User Class Option for DHCP
- RFC 3315 Dynamic Host Configuration Protocol for IPv6 (DHCPv6)
- RFC 3633 IPv6 Prefix Options for Dynamic Host Configuration Protocol(DHCP)version 6
- RFC 3646 DNS Configuration options for Dynamic Host Configuration Protocol for IPv6 (DHCPv6)
- RFC 4649 Dynamic Host Configuration Protocol for IPv6 (DHCPv6) Relay Agent Remote-ID Option
- RFC 6334 Dynamic Host Configuration Protocol for IPv6 (DHCPv6) Option for Dual-Stack Lite

Limitations and Restrictions

The DHCP and IPoE Subscriber Management feature has the following limitations in this release:

- DHCPv6 addresses and prefixes do not get released at IPv6CP disconnect.
- For DHCPv4 sessions, subnet mask and default gateway are derived from the IPAM pool configuration and IP pool split logic. The first subnet route, subnet mask, and default gateway IP is derived from the IPAM and pushed to the UP for each chunk of the pool. Subnet mask and default gateway cannot be assigned via the AAA configuration.
- For DHCPv4 sessions, subnet selection is not supported. The IP is selected from the mapped IP pool. Subnet selection cannot be controlled via the AAA gateway IP, giaddr, or subnet selection suboption.
- For DHCPv4 sessions, requested IP option (option 50) that helps in requesting specific IP is not supported. However on client reboot (discover in bound state), the already assigned IP is retained.
- DHCP Inform packet and DHCPv6 Information Request packet handling for unbound sessions are not supported. That is, the client cannot get only the host configurations without requesting for IP assignment via BNG.
- For DHCPv4 session, broadcast flag check, and discovery, offer, request, and acknowledgement (DORA) unicast is not supported.
- If DHCP client initiated packet options like requested options (option 55 for IPv4, ORO option 6 for IPv6), circuit-id, remote-id, user class, vendor class changes in the packet over the session lifecycle, the cnBNG server behaviour is not defined. cnBNG assumes that the client will not change these options over the lifecycle of session. The client should also maintain the same values for attributes like remote-id, vendor class, user class for both IPv4 and IPv6 afi (AFI). In case these value are required to be changed, it is recommended to clear the session and bring it up again.
- Client reboot scenarios do not tear down the session in cnBNG in the following scenarios: If the Discover message is received in the Bound state or Solicit message is received for the already bound IANA, cnBNG does not tear down the existing session. Instead, the already allocated IP is assigned to the subscriber.

In this case, fresh lease is assigned to the client. This is a difference in behaviour from physical BNG where on receiving Discover message in Bound state, IPv4 stack is brought down and new IP is assigned.

- No parity support for RADIUS attribute formatting with ASR 9000. The supported RADIUS attribute list and formatting would be updated based on feedback from customer. For example, some attributes like remote-id format is different for IPv4 and IPv6 clients. Hence, the value going to the Policy Plane differs based on whether the IPv4 or IPv6 afi comes up first.
- Change of Authorization (CoA) for DHCP consumed RADIUS attributes are not supported.
- RFC recommended DHCPv4/v6 packet validations are not supported.
- A common DHCP class attribute is used for class specification for DHCPv4 and DHCPv6 stack via AAA attribute. The attribute is dhcp-class.
- Framed route is not supported.
- Manual pod restart is not supported or entertained. Pod restart can lead to inconsistencies between the CP pods with regard to session count and session state. To recover the inconsistent sessions, the **clear** command must be used explicitly.
- After a subscriber is UP, if the subscriber is deleted from the cnBNG CP—for example, due to an administrative clear or a Pod event—the subscriber is not notified.

  cnBNG CP deletion does not proactively notify the client. During a subsequent renew, cnBNG returns the DHCPNAK message for IPv4 or NoBinding message for IPv6, enabling the client to renegotiate immediately rather than waiting for the lease to expire.

Configuring the DHCP and IPoE Subscriber Management Feature

This section describes how to configure the DHCP and IPoE Subscriber Management feature.

Configuring the DHCP and IPoE Subscriber Management feature involves the following steps:

1. Configure the IPv4 DHCP Profile
2. Configure the IPv4 DHCP Class
3. Configure the IPv6 DHCP Profile
4. Configure the IPv6 DHCP Class

Configuring the IPv4 DHCP Server Profile

Use the following commands to configure the IPv4 DHCP server profile.

```
config
  profile dhcp dhcp_profile_name
  ipv4
  server { boot-filename boot_filename  } |
  { dns-servers dns_server } | { domain-name domain_name } |
  { netbios-name-server netbios_name_server } |
  { netbios-node-type { broadcast-node | hexadecimal | hybrid-node |
mixed-node | peer-to-peer-node } |
```

```
  { next-server ipv4_address } | { ntp-servers ipv4_address } |
  {{ pool-group-tag  pool_group_tag_name} | { pool-name ipam_pool_name }} |
  {option-codes option_codes_range { ascii-string value | force insert { true
 | false } | hex-string value |
  {ip-address ip_address } | { lease { days value | hours value| minutes value
 }
  exit
exit
```

NOTES:

- **profile dhcp** *dhcp_profile_name*: Specifies the DHCP profile name.
- **ipv4**: Enters IPv4 configuration mode.
- **server { boot-filename** *boot_filename* **} | { dns-servers** *dns_server* **} | { domain-name** *domain_name* **} | { netbios-name-server** *netbios_name_server* **} | { netbios-node-type { broadcast-node | hexadecimal | hybrid-node | mixed-node | peer-to-peer-node } | { next-server** *ipv4_address* **} | { ntp-servers** *ipv4_address* **} | { pool-name** *ipam_pool_name* **} | { option-codes** *option_codes_range* **{ ascii-string** *value* **| force insert { true | false } | hex-string** *value* **| { ip-address** *ip_address* **} | { lease { days** *value* **| hours** *value*| **minutes** *value* **}**: Specifies the IPv4 server details.
  - **boot-filename** *boot_filename*: Configures the boot file.
  - **dns-servers** *dns_server*: Specifies the Domain Name System (DNS) IPv4 servers available to a DHCP for an IPv4 client.
  - **domain-name** *domain_name*: Specifies the domain name for the IPv4 client.
  - **netbios-name-server** *netbios_name_server*: Configures the NetBIOS name servers.
  - **netbios-node-type { broadcast-node | hexadecimal | hybrid-node | mixed-node | peer-to-peer-node }**: Configures the NetBIOS node as a broadcast, hexadecimal, hybrid, mixed, or peer-to-peer node. The valid values for each of these nodes are:
    - **broadcast-node**: 0x1 B-node
    - **hexadecimal**: Operator provided custom 1 byte hex value
    - **hybrid-node**: 0x8 H-node
    - **mixed-node**: 0x4 M-node
    - **peer-to-peer-node**: 0x2 P-node
  - **next-server** *ipv4_address*: Specifies the TFTP-server IP address for the client to use.
  - **pool-name** *ipam_pool_name*: Specifies the IP Address Management (IPAM) assigned pool name.
  - **pool-group-tag** *pool_group_tag_name*: Specifies the group-tag mapping to IPAM pool configuration. Configure either **pool-name** or **pool-group-tag**, but not both.

    Specifies the tag group value of the pool. All pools carrying the same tag group value can be associated to a DHCP profile using the keyword **pool-group-tag** and the corresponding tag group value. The value must be a string.

- **option-codes option_codes_range { ascii-string** *value* **| force insert { true | false } | hex-string** *value* **| ip-address** *ip_address* **}**: Specifies the values for the ASCII string of length 128, force insert, hex string of length 128, or IP address (IPv4 IP address).
- **lease { days** *value* **| hours** *value* **| minutes** *value* **}**: Specifies the lease time duration in the number of days, hours, and minutes. The number of lease days supported is from 0 to 365. The number of leave hours supported ranges from 0 to 23 and minutes from 0 to 59.

Configuring the IPv4 DHCP Class

Use the following commands to configure the IPv4 DHCP class.

```
config
  profile dhcp dhcp_profile_name
  ipv4
  class dhcp_class_name
  matches { match { dhcpv4-circuit-id { ascii value | hex value } |
dhcpv4-remote-id { ascii value |
    hex value } | dhcpv4-vendor-class { ascii value | hex value } |
dhcpv4-user-class { ascii value |
    hex value } } | match-type { all match_key_value | any match_key_value } }
  server { boot-filename boot_filename } |
     { dns-servers dns_server } | { domain-name domain_name } |
     { netbios-name-server netbios_name_server } |
     { netbios-node-type { broadcast-node | hexadecimal | hybrid-node |
mixed-node | peer-to-peer-node } |
     { next-server ipv4_address } | { ntp-servers ipv4_address } |
     {{ pool-group-tag  pool_group_tag_name} | { pool-name ipam_pool_name }} |
    {option-codes option_codes_range { ascii-string value | force insert {
true | false } | hex-string value |
    {ip-address ip_address } | { lease { days value | hours value| minutes
value }
  end
```

NOTES:

- **profile dhcp** *dhcp_profile_name*: Specifies the DHCP profile name.
- **ipv4**: Enters IPv4 configuration mode.
- **class** *dhcp_class_name*: Creates a proxy profile class (DHCP), which can be used to enter the proxy profile class sub-configuration mode.
- **matches { match { dhcpv4-circuit-id { ascii** *value* **| hex** *value* **} | dhcpv4-remote-id { ascii** *value* **| hex** *value* **} | dhcpv4-vendor-class { ascii** *value* **| hex** *value* **} | dhcpv4-user-class { ascii** *value* **| hex** *value* **} } | match-type { all** *match_key_value* **| any** *match_key_value* **} }**: Specifies the list of match keys and values. The match values supported are DHCPv4 circuit ID, DHCPv4 remote ID, DHCPv4 vendor class, and DHCPv4 user class. Each of the values must specify either an ASCII or hexadecimal value.

  match-type { all | any }: Specifies if the match value should apply to any of the specified keys or to all the keys.
- **server { boot-filename** *boot_filename* **} | { dns-servers** *dns_server* **} | { domain-name** *domain_name* **} | { netbios-name-server** *netbios_name_server* **} | { netbios-node-type { broadcast-node | hexadecimal**

| hybrid-node | mixed-node | peer-to-peer-node } | { next-server *ipv4_address* **} | { ntp-servers** *ipv4_address* **} | { pool-name** *ipam_pool_name* **} | { option-codes** *option_codes_range* **{ ascii-string** *value* **| force insert { true | false } | hex-string** *value* **| { ip-address** *ip_address* **} | { lease { days** *value* **| hours** *value*| **minutes** *value* **}**: Specifies the IPv4 server details.

- **boot-filename** *boot_filename*: Configures the boot file.
- **dns-servers** *dns_server*: Specifies the Domain Name System (DNS) IPv4 servers available to a DHCP for an IPv4 client.
- **domain-name** *domain_name*: Specifies the domain name for the IPv4 client.
- **netbios-name-server** *netbios_name_server*: Configures the NetBIOS name servers.
- **netbios-node-type { broadcast-node | hexadecimal | hybrid-node | mixed-node | peer-to-peer-node }**: Configures the NetBIOS node as a broadcast, hexadecimal, hybrid, mixed, or peer-to-peer node. The valid values for each of these nodes are:
  - **broadcast-node**: 0x1 B-node
  - **hexadecimal**: Operator provided custom 1 byte hex value
  - **hybrid-node**: 0x8 H-node
  - **mixed-node**: 0x4 M-node
  - **peer-to-peer-node**: 0x2 P-node
- **next-server** *ipv4_address*: Specifies the TFTP-server IP address for the client to use.
- **pool-name** *ipam_pool_name*: Specifies the IP Address Management (IPAM) assigned pool name.
- **pool-group-tag** *pool_group_tag_name*: Specifies the group-tag mapping to IPAM pool configuration. Configure either **pool-name** or **pool-group-tag**, but not both.

  Specifies the tag group value of the pool. All pools carrying the same tag group value can be associated to a DHCP profile using the keyword **pool-group-tag** and the corresponding tag group value. The value must be a string.
- **option-codes option_codes_range { ascii-string** *value* **| force insert { true | false } | hex-string** *value* **| ip-address** *ip_address* **}**: Specifies the values for the ASCII string of length 128, force insert, hex string of length 128, or IP address (IPv4 IP address).
- **lease { days** *value* **| hours** *value* **| minutes** *value* **}**: Specifies the lease time duration in the number of days, hours, and minutes. The number of lease days supported is from 0 to 365. The number of leave hours supported ranges from 0 to 23 and minutes from 0 to 59.

Configuring the IPv6 DHCP Server Profile

Use the following commands to configure the IPv6 DHCP server profile.

```
config
  profile dhcp dhcp_profile_name
    ipv6
      server { aftr-name value | dns-servers dns_server
      domain-name domain_name |
```

```
        {{ iana-pool-name ipam_pool_name}  | {iana-pool-group-tag
iana_pool_group_tag_name}}|
        {{iapd-pool-name ipam_pool_name } | {iapd-pool-group-tag
iapd_pool_group_tag_name}}|
        lease { days value | hours value | minutes value }|
        preference value }
```

NOTES:

- **profile dhcp** *dhcp_profile_name*: Specifies the DHCP profile name.
- **ipv6**: Enters IPv6 configuration mode.
- **server { aftr-name** *value* **| dns-servers** *dns_server* **| domain-name** *domain_name* **| iana-pool-name** *ipam_pool_name* **| iapd-pool-name** *ipam_pool_name*| **lease { days** *value* **| hours** *value* **| minutes** *value* **} | preference** *value* **}**: Specifies the IPv6 server details.
  - **aftr-name** *value*: Specifies the FQDN string.
  - **dns-servers** *dns_server*: Specifies the Domain Name System (DNS) IPv4 servers available to a DHCP for an IPv4 client.
  - **domain-name** *domain_name*: Specifies the domain name for the IPv4 client.
  - **iana-pool-name** *ipam_pool_name*: Specifies the Internet Assigned Numbers Authority (IANA) pool name.
  - **iana-pool-group-tag** *iana_pool_group_tag_name*: Specifies the tag group value of the pool. All pools carrying the same tag group value can be associated to a DHCP profile using the keyword **pool-group-tag** and the corresponding tag group value. The value must be a string. Configure either **iana-pool-name** or **iana-pool-group-tag** , but not both.
  - **iapd-pool-name** *ipam_pool_name*: Specifies the Identity Association for Prefix Delegation (IAPD) pool name.
  - **iapd-pool-group-tag** *iapd_pool_group_tag_name*: Specifies the tag group value of the pool. All pools carrying the same tag group value can be associated to a DHCP profile using the keyword **pool-group-tag** and the corresponding tag group value. The value must be a string. Configure either **iapd-pool-name** or **iapd-pool-group-tag** , but not both.
  - **lease { days** *value* **| hours** *value* **| minutes** *value* **}**: Specifies the lease time duration in the number of days, hours, and minutes. The number of lease days supported is from 0 to 365. The number of leave hours supported ranges from 0 to 23 and minutes from 0 to 59.
  - **preference** *value*: Specifies the DHCP server preference. The preference value ranges from 1 to 255.

Configuring the IPv6 DHCP Class

Use the following commands to configure the IPv6 DHCP class.

```
config
  profile dhcp dhcp_profile_name
    ipv6
      class dhcp_class_name
```

```
        server { aftr-name value | dns-servers dns_server | domain-name
domain_name |
        {{iana-pool-name ipam_pool_name  } | {iana-pool-group-tag
pool_group_tag_name}} |
        {{iapd-pool-name ipam_pool_name } | {iapd-pool-group-tag
pool_group_tag_name}} |
        lease { days value |
        hours value | minutes value }
        preference value
        end
```

NOTES:

- **profile dhcp** *dhcp_profile_name*: Specifies the DHCP profile name.
- **ipv6**: Enters IPv6 configuration mode.
- **class** *dhcp_class_name*: Creates a proxy profile class (DHCP), which can be used to enter the proxy profile class sub-configuration mode.
- **server { aftr-name** *value* **| dns-servers** *dns_server* **| domain-name** *domain_name* **| iana-pool-name** *ipam_pool_name* **| iapd-pool-name** *ipam_pool_name* **| lease { days** *value* **| hours** *value* **| minutes** *value* **} | preference** *value* **}**: Specifies the IPv6 class server details.
  - **aftr-name** *value*: Specifies the FQDN string.
  - **dns-servers** *dns_server*: Specifies the Domain Name System (DNS) IPv6 servers available to a DHCP for an IPv6client.
  - **domain-name** *domain_name*: Specifies the domain name for the IPv6 client.
  - **iana-pool-name** *ipam_pool_name*: Specifies the Internet Assigned Numbers Authority (IANA) pool name.
  - **iana-pool-group-tag** *iana_pool_group_tag_name*: Specifies the tag group value of the pool. All pools carrying the same tag group value can be associated to a DHCP profile using the keyword **pool-group-tag** and the corresponding tag group value. The value must be a string. Configure either **iana-pool-name** or **iana-pool-group-tag**, but not both.
  - **iapd-pool-name** *ipam_pool_name*: Specifies the Identity Association for Prefix Delegation (IAPD) pool name.
  - **iapd-pool-group-tag** *iapd_pool_group_tag_name*: Specifies the tag group value of the pool. All pools carrying the same tag group value can be associated to a DHCP profile using the keyword **pool-group-tag** and the corresponding tag group value. The value must be a string. Configure either **iapd-pool-name** or **iapd-pool-group-tag** , but not both.
  - **lease { days** *value* **| hours** *value* **| minutes** *value* **}**: Specifies the lease time duration in the number of days, hours, and minutes. The number of lease days supported is from 0 to 365. The number of leave hours supported ranges from 0 to 23 and minutes from 0 to 59.
  - **preference** *value*: Specifies the DHCP server preference. The preference value ranges from 1 to 255.

DHCPv6 Raw Option Support

Table 37: Feature History

Feature Name	Release Information	Description
DHCPv6 Raw Option Support	2024.04.0	cnBNG now supports DHCPv6 raw options, enabling you to set any DHCPv6 option type, including Mapping of Address and Port using Encapsulation (MAP-E) for scenarios such as migration to cloud native BNG.

This feature introduces the capability to configure DHCPv6 options in their raw form. This means you can specify any DHCPv6 option type beyond the predefined set, providing greater flexibility in host configurations.

DHCPv6 Raw Option Support is critical for scenarios such as migration from physical BNG to cloud native BNG, where specific DHCPv6 options like MAP-E (DHCPv6 option 94) need to be set for MAP-E translation.

The raw option support is designed to integrate smoothly with existing DHCP server configurations. You can continue to use the familiar settings for common attributes (for example, DNS, domain name, lease time) while leveraging the raw option feature for more specialized requirements, and different network environments.

Configure DHCPv6 Raw Option Support

You can configure option codes in both DHCP profile and DHCP class sub mode. As with existing configurations, priority is given to the class configuration.

Procedure

Step 1 Configure the option codes in IPv6 DHCP server profile.

Example:

```
config
  profile dhcp dhcp_profile_name
   ipv6
     mode server
     server
      option-codes
      option-code 39
       ascii-string abcd.com
     exit
     option-code 31
      ip-address [ 2001:db8:c:641::6401 ]
    exit
```

The following is a sample MAP-E configuration.

```
profile dhcp DHCP_PROFILE
ipv6
```

```
  mode server
  server
   option-codes
    option-code 39
     hex-string
00590016000c1a5e4102002c2a020586f900005d0004040000000059001500 1018c63365002820010db896005d0004040000000005a00102a020586000000000000000000000406

    exit
   exit
  exit
exit
exit
```

Step 2 Configure the option codes in IPv6 DHCP class.

Example:

```
config
   profile dhcp dhcp_profile_name
     ipv6
        class dhcp_class_name
        server
           option-codes
            option-code 23
             ip-address [ 2001::7 ]
          option-code 39
             ascii-string class.com
    exit

....
```

NOTES:

- **option-codes**: Enters the Option Codes Configuration mode (config-option-codes). Configures the OptionCode table.
- **option-code** *code*: Specifies the DHCP option code [up to 255].
- **ascii-string** *value*: Specifies the values for the ASCII string of length 256.
- **ip-address** *ip_address* : Specifies the IPv6 address.
- **force-insert** {*true* | *false* }: If set to true, force insert the option regardless of the DHCPv6 Option Request Option (ORO) value. If set to false, honor the ORO option.

The DHCP class name (for example, `automation-class`) is applied to a subscriber if the user profile has the DHCP class attribute `Cisco-AVPair += "dhcp-class=automation-class"` set. The following are a few scenarios based on the configuration example:

- **Option Code 23**
  - **Subscriber A:** For whom `dhcp-class=automation-class` is set in the user profile, option 23 is included in response packets with the IP address `[2001::7]`.
  - **Subscriber B:** For whom `dhcp-class` is not set in the user profile, option 23 will not be included in response packets.

- **Option Code 39**
  - **Subscriber A:** For whom `dhcp-class=automation-class` is set in the user profile, option 39 is included in response packets with the value `"class.com"`.
  - **Subscriber B:** For whom `dhcp-class` is not set in the user profile, option 39 will be included in response packets with the value `"abcd.com"`.
- **Option Code 31**
  - **Subscriber A:** For whom `dhcp-class=automation-class` is set in the user profile, option 31 would be included in response packets with the IP address `2001:db8:c:641::6401`.
  - **Subscriber B:** For whom `dhcp-class` is not set in the user profile, option 31 will still be included in response packets with the IP address `"2001:db8:c:641::6401"`

Raw Option Code Exclusion List

The following list of options cannot be set using the raw option CLIs. These options are either not intended to be set by the server or require additional functionalities beyond simple option setting:

```
OPTION_CLIENTID - 1
OPTION_SERVERID - 2
OPTION_IA_NA - 3
OPTION_IA_TA - 4
OPTION_IAADDR - 5
OPTION_ORO - 6
OPTION_PREFERENCE - 7
OPTION_ELAPSED_TIME - 8
OPTION_RELAY_MSG - 9
Option - 10
OPTION_STATUS_CODE - 13
OPTION_RAPID_COMMIT - 14
OPTION_USER_CLASS - 15
OPTION_INTERFACE_ID - 18
OPTION_DNS_SERVERS -23
OPTION_DOMAIN_LIST - 24
OPTION_IA_PD - 25
OPTION_IAPREFIX -26
OPTION_INFORMATION_REFRESH_TIME - 32
Option - 35
OPTION_REMOTE_ID - 37
OPTION_SUBSCRIBER_ID - 38
OPTION_CLIENT_FQDN - 39
OPTION_ERO - 43
OPTION_LQ_QUERY - 44
OPTION_LQ_QUERY - 45
OPTION_CLT_TIME - 46
OPTION_LQ_RELAY_DATA - 47
OPTION_LQ_CLIENT_LINK - 48
OPTION_RELAY_ID - 53
OPTION_AFTR_NAME - 64
OPTION_RSOO - 66
OPTION_PD_EXCLUDE - 67
OPTION_VSS - 68
OPTION_CLIENT_LINKLAYER_ADDR - 79
OPTION_LINK_ADDRESS - 80
OPTION_RADIUS - 81
OPTION_DHCPV4_MSG - 87
OPTION_DHCP4_O_DHCP6_SERVER - 88
OPTION_LQ_BASE_TIME - 100
OPTION_LQ_START_TIME - 101
```

```
OPTION_LQ_END_TIME - 102
OPTION_ANI_ATT - 105
OPTION_ANI_NETWORK_NAME - 106
OPTION_ANI_AP_NAME - 107
OPTION_ANI_AP_BSSID - 108
OPTION_ANI_OPERATOR_ID - 109
OPTION_ANI_OPERATOR_REALM - 110
OPTION_MUD_URL_V6 - 112
OPTION_F_BINDING_STATUS - 114
OPTION_F_CONNECT_FLAGS - 115
OPTION_F_DNS_REMOVAL_INFO - 116
OPTION_F_DNS_HOST_NAME - 117
OPTION_F_DNS_ZONE_NAME - 118
OPTION_F_DNS_FLAGS - 119
OPTION_F_EXPIRATION_TIME - 120
OPTION_F_MAX_UNACKED_BNDUPD - 121
OPTION_F_MCLT - 122
OPTION_F_PARTNER_LIFETIME - 123
OPTION_F_PARTNER_LIFETIME_SENT - 124
OPTION_F_PARTNER_DOWN_TIME - 125
OPTION_F_PARTNER_RAW_CLT_TIME - 126
OPTION_F_PROTOCOL_VERSION - 127
OPTION_F_KEEPALIVE_TIME - 128
OPTION_F_RECONFIGURE_DATA - 129
OPTION_F_RELATIONSHIP_NAME - 130
OPTION_F_SERVER_FLAGS - 131
OPTION_F_SERVER_STATE - 132
OPTION_F_START_TIME_OF_STATE - 133
OPTION_F_STATE_EXPIRATION_TIME - 134
OPTION_RELAY_PORT - 135
OPTION_IA_LL - 138
OPTION_LLADDR - 139
OPTION_SLAP_QUAD – 140
```



Note

- The CLI supports setting option codes up to 255, excluding those specified in the **Raw Option Code Exclusion List**.
- The **force insert** flag setting for a given option code must be consistent across the configuration. For example, if a given option code is configured under a class with the **force insert** flag set to **TRUE**, it must also be set to **TRUE** if the same option code is configured under a profile, and conversely.

IPv6 Class Configuration and Static IP Allocation Support

Table 38: Feature History

Feature Name	Release Information	Description
IPv6 Class Configuration and Static IP Allocation Support	2025.01.0	This feature ensures reliable IPv6 client IP allocation by allowing static IP assignments based on class parameters within a DHCPv6 profile.

This feature allows static IP address allocation for clients based on their classification within the DHCPv6 profile. This ensures that specific devices receive predetermined IP addresses based on their identity or other

classification criteria. The feature involves configuring a DHCPv6 profile that contains DHCP options for IPv6. Within this profile, classes can be defined that include specific DHCP options and criteria for selecting which class a client belongs to.

Class selection based on Match-Info:

You can configure the class selection to be based on a match of **any** or **all** key parameters of that class. If no DHCP option class matches for the ongoing session or any requested DHCP Options is not found in the selected class, then the requested options are selected from the default DHCP options of that profile.

Key parameters for DHCPv6 Class selection:

The key parameters to select the DHCP Options Class for DHCPv6 include:

- Interface-id (DHCP Option 18)
- Remote-id (DHCP Option 37)
- Vendor-class (DHCP Option 16)
- User-class (DHCP Option 15)

The DHCPv6 Options Class consists of the following elements:

- IanaPoolName
- IapdPoolName
- DnsServers
- DomainName
- Preference
- AftrName
- Lease
- Static-ip-key

Behavior of Class Selection

- When multiple classes are configured, if the incoming solicit matches more than one class, the order of matching is not specified. The solicit may match any one of the classes.

  For example, if an incoming packet has options that match both class1 and class2, it can match either class, and the matching order may differ from the order in the running configuration.

- If the packet doesn’t match any of the configured classes, it gets an IP from the default pool.

Static IP Key Identifier

It is expected that there is an IP configured in the IPv6 static database for the corresponding MAC address. If no IP address is configured for the matching MAC address, IP allocation fails.

Match-Type Scenarios

- **Match-Type "All":**

With **match-type all**, a solicit must match all configured option values to match the class.

- **Match-Type "Any":**

  With **match-type any**, the class is selected if any one of the specified key parameters matches. This means that the incoming packet only needs to satisfy at least one of the configured match criteria (for example, interface-id, remote-id, vendor-class, or user-class) for the class to be considered a match.

Priority of DHCP-Class Attribute

If the **dhcp-class** attribute from RADIUS and match type coexist, the RADIUS configuration takes precedence.

Configure the IPv6 DHCP Class

Procedure

Define a DHCP profile with IPv6 configuration:

Example:

```
config
  profile dhcp dhcp_profile_name
  ipv6
    class dhcp_class_name

     matches
      match-type { any  match_key_value | all match_key_value }
      match dhcpv6-interface-id { ascii value | hex value }
      match dhcpv6-remote-id { ascii value | hex value }
      match dhcpv6-user-class { ascii value | hex value }
      match dhcpv6-vendor-class { ascii value | hex value }
     exit
    exit
```

The following is a sample configuration for match-type **any**:

```
profile dhcp dhcp-profile1
Ipv6
 mode server
  server
   iana-pool-name poolv6iana
   iapd-pool-name poolv6iapd
   dns-servers    [ 5000::5 ]
   domain-name    cisco.com
   lease days 1
exit
 class class1
   matches
    match-type any
    match dhcpv6-interface-id hex [ 61626365 ]
    match dhcpv6-remote-id hex [ 6656 ]
    match dhcpv6-user-class hex [ 4a696f53746174696332 ]
    match dhcpv6-vendor-class hex [ 726f7574657264657669636532 ]
exit
```

Example Scenarios for Match-type any

- If the incoming packet includes a remote-id with the hex value 6656, an interface-id with the hex value 567644, a user-class with the hex value 6a87d556, and a vendor-class with the hex value 678776346, the class matches. This is because at least one of the specified conditions is met.
- If the incoming packet has a remote-id with the hex value 34566, an interface-id with the hex value 245644, a user-class with the hex value 86787d556, and a vendor-class with the hex value 776d346, the class does not match. This is because none of the specified conditions are met.

The following is a sample configuration for match-type **all**:

```
profile dhcp dhcp-profile1
Ipv6
 mode server
  server
   iana-pool-name poolv6iana
   iapd-pool-name poolv6iapd
   dns-servers    [ 5000::5 ]
   domain-name    cisco.com
   lease days 1
exit
 class class1
   matches
    match-type all
    match dhcpv6-interface-id hex [ 61626365 ]
    match dhcpv6-remote-id hex [ 6656 ]
    match dhcpv6-user-class hex [ 4a696f53746174696332 ]
    match dhcpv6-vendor-class hex [ 726f7574657264657669636532 ]
exit
```

Example Scenarios for Match-type all

- If the incoming packet contains an interface-id with the hex value 61626365, a remote-id with the hex value 2356, a user-class with the hex value 6a87d556, and a vendor-class with the hex value 678776346, the class does not match. This means not all specified conditions are met.
- If the incoming packet has an interface-id with the hex value 61626365, a remote-id with the hex value 6656, a user-class with the hex value 4a696f53746174696332, and a vendor-class with the hex value 726f7574657264657669636532, the class matches. This indicates that all the specified conditions are satisfied.

Note Each match option can have multiple values. For example,

```
match dhcpv6-interface-id hex [ 61626367 ]

match dhcpv6-user-class hex [ 4a696f53746174696111 4a696f53746174696112 4a696f53746174696113
 ]
```

NOTES:

- **profile dhcp** *dhcp_profile_name*: Specifies the DHCP profile name.
- **ipv6**: Enters IPv6 configuration mode.
- **class** *dhcp_class_name*: Creates a proxy profile class (DHCP), which can be used to enter the proxy profile class sub-configuration mode.

- **matches { match { dhcpv6-interface-id { ascii** *value* **| hex** *value* **} | dhcpv6-remote-id { ascii** *value* **| hex** *value* **} | dhcpv6-vendor-class { ascii** *value* **| hex** *value* **} | dhcpv6-user-class { ascii** *value* **| hex** *value* **} }** : Specifies the list of match keys and values. Each of the values must specify either an ASCII or hexadecimal value.

  match-type { all | any }: Specifies if the match value should apply to any of the specified keys or to all the keys.

Configure Static IP Allocation

Procedure

Step 1 Configure the static IP key.

Example:

```
config
  profile dhcp dhcp_profile_name
 ipv6
  class dhcp_class_name
   server
    iana-pool-name iana_pool_name
    iapd-pool-name iapd_pool_name
    static-ip-key
     identifier client-mac-address mac_address
    exit
   exit
 exit
```

NOTES:

- **static-ip-key identifier client-mac-address** *mac_address*: Enables a client with a specific MAC address to receive a pre-configured static IP address from the IPv6 static database.

  Note For the allocation to succeed, there must be an IP address already configured in the IPv6 static database corresponding to the client's MAC address. If no such IP address is configured, the IP allocation will fail.

Step 2 Configure static IP address ranges.

Example:

```
config
  ipam
    instance instance_id
      address-pool pool_name
       vrf-name vrf_name
       static enable user_plane_name
       ipv4
```

```
        split-size
         no-split
        exit
        address-range start_ipv4_address end_ipv4_address
          default-gateway ipv4_address
          exit
      ipv6
        address-ranges
          split-size
            no-split
          exit
         address-range start_ipv6_address end_ipv6_address
        exit
        prefix-ranges
          split-size
           no-split
          exit
          prefix-range ipv6_address length prefix_length
        exit
    exit
```

The following is a sample configuration:

```
ipam
 instance 1
  address-pool staticpool1
   vrf-name vrf1
   static enable user-plane1
   ipv4
    split-size
     no-split
    exit
    address-range 10.44.1.0 10.44.1.255 default-gateway 10.44.1.1
   exit
   ipv6
    address-ranges
     split-size
      no-split
     exit
     address-range 2001:db8::1 2001:db8::ffff
    exit
    prefix-ranges
     split-size
      no-split
     exit
     prefix-range 2002:ab:: length 48
    exit
   exit
```

NOTES:

- **static enable** *user_plane_name*: Configures static IP details. Sets the specified User Plane (UP) as static. *user_plane_name* is the specified UP for this static pool.
- **split-size no-split**: Specifies that the address-ranges should not be into smaller chunks.
- **address-range** *start_ipv4/ipv6_address end_ipv4/ipv6_address*: Configures the IPv4 or IPv6 address range with the starting and ending IPv4/IPv6 address.

- IPAM does only route-validation for Static IP.

Leased IP Hold Time

Feature Summary and Revision History

Summary Data

Table 39: Summary Data

Applicable Product(s) or Functional Area	cnBNG
Applicable Platform(s)	SMI
Feature Default Setting	Disabled - Configuration Required
Related Documentation	Not Applicable

Revision History

Table 40: Revision History

Revision Details	Release
First introduced.	2022.02.0

Feature Description

The Leased IP Hold Time feature provides the flexibility to configure the cnBNG DHCP server to apply a hold time to the leased IP. After release from the client or lease expiry, the IP is not available for reuse for other clients until the hold time elapses.

After the lease expiry or release from client, if the same client requests the IP again within the hold time, the same IP is provided to the client. After the hold time, the IP is made available in the free IP list and can get allocated to other clients.

A new CLI is introduced in the DHCP profile to allow the hold time configuration. For more information, see Configuring Leased IP Hold Time, on page 244.

How it Works

This section provides a brief of how the Leased IP Hold Time feature works.

The DHCP record retains minimal data (like, sublabel, IP, pool information, and so on) in the CDL, to hold the IP for a given session after release or timer expiry after a subscriber is disconnected.

The FSM state moves back to the initialization state for the IP. However, the IP is not be released back to the pool (basically no interaction with IPAM or Node Manager). The information for the session will be unavailable in the Session Manager and User Plane function (UPF). The DHCP record with the IP on hold is removed after the hold time if the session does not come back up within the specified hold time.

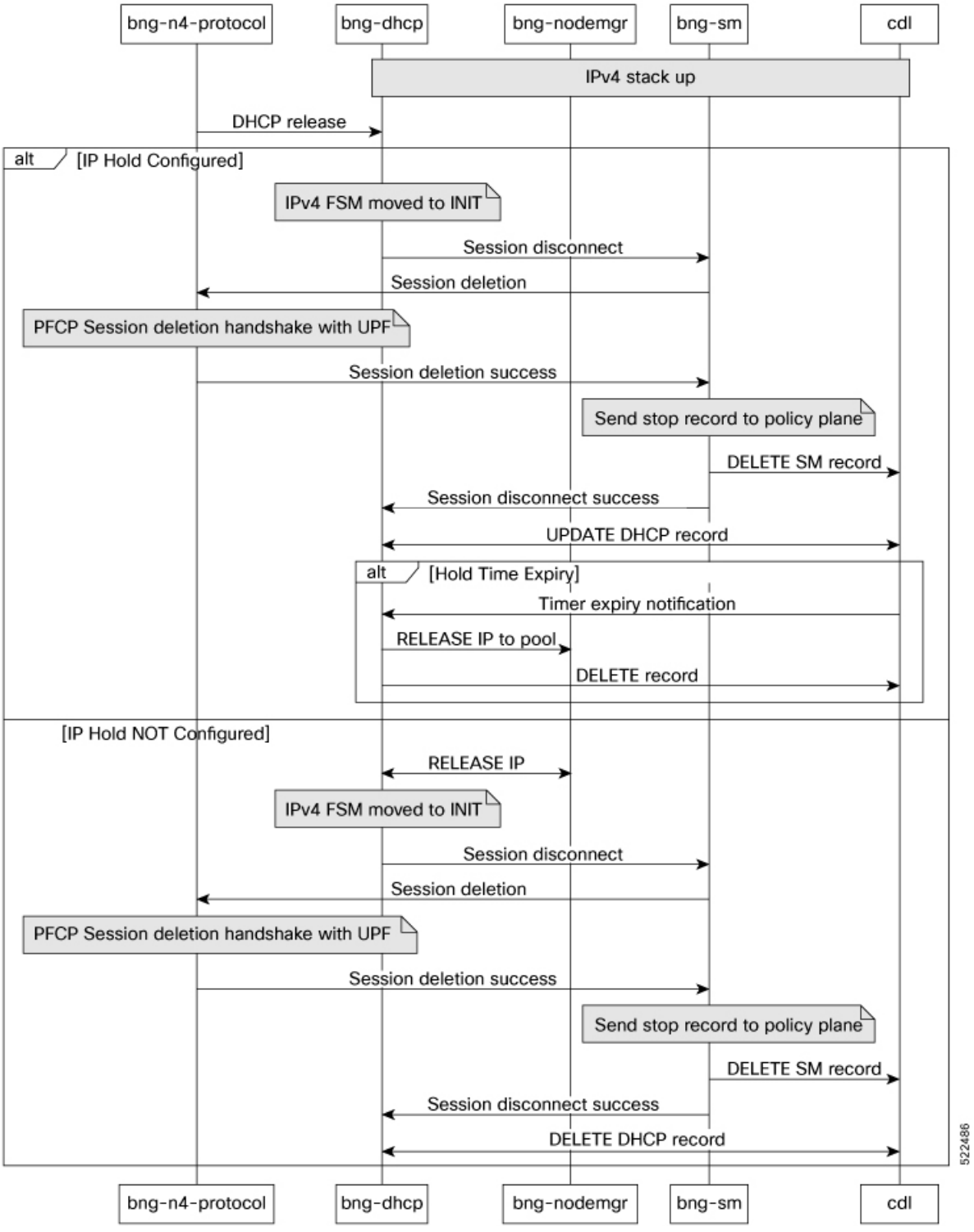
Call Flows

This section includes the following call flows.

IPv4 Session with or without Hold Time Call Flow

The following is an example of an IPv4 release for an IPv4 session with and without a hold time applied.

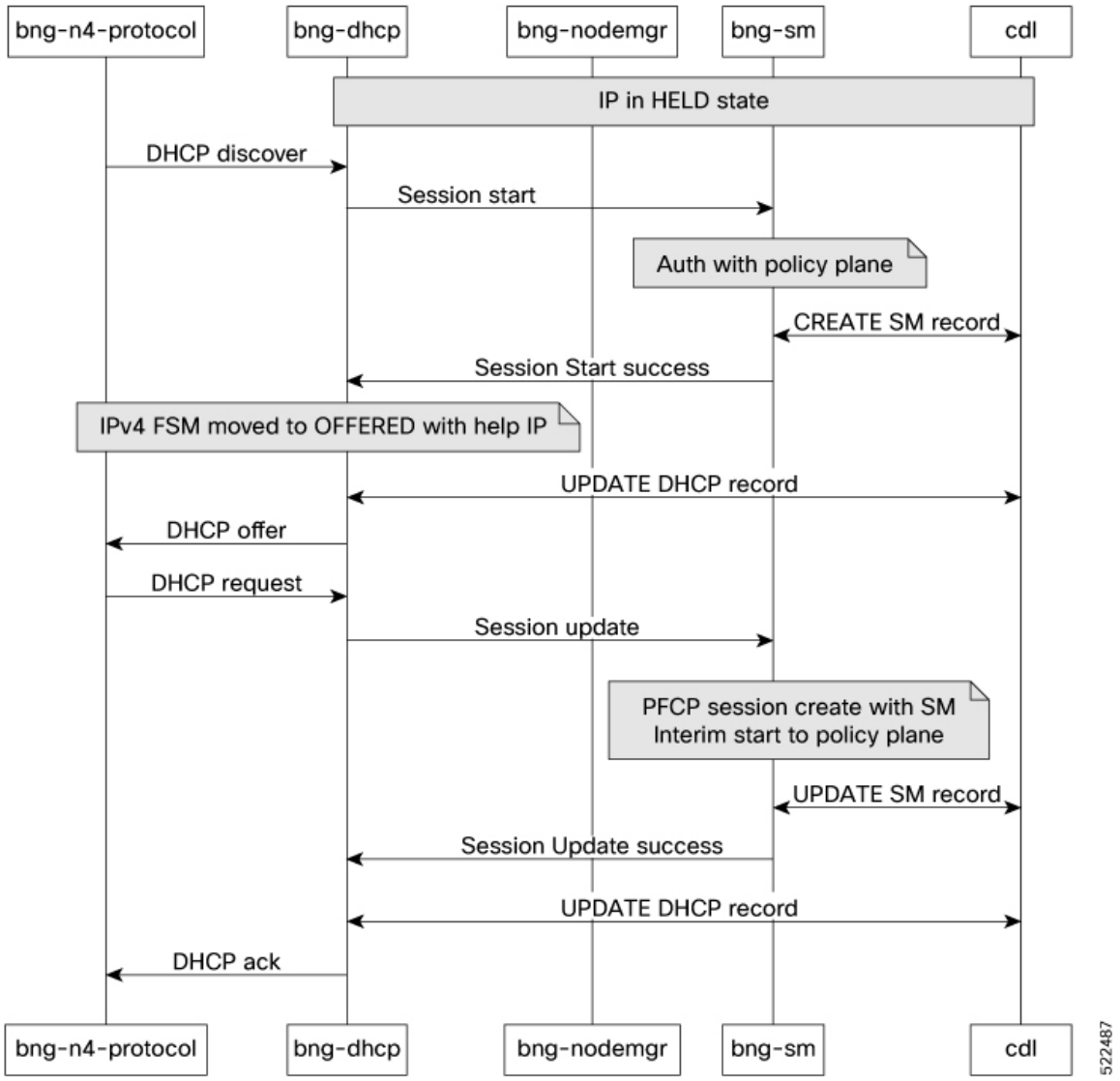
Figure 13: IPv4 Session with or without Hold Time Call Flow



IPv4 Session Bring-Up with IP on Hold

The following is an example of an IPv4 serssion bring-up with IP on hold.

Figure 14: IPv4 Session Bring-Up with IP on Hold Call Flow



Limitations and Restrictions

The Leased IP Hold Time feature has the following limitations:

- When the administrator runs the **clear** and **coa disconnect** CLI comamnds, it cleans up the session even if the IP is in 'hold' state. After the administrato clears the session, the hold timer for a new IP does not take effect.
- If the IP is declined from the client side, the IP is not moved to 'hold' state.
- When Request, Renew, Rebind are not acknowledged due to internal errors or session establishment or modification failures, the session IP is not maintained in the 'hold' state.

- Class level hold time configuration is not supported.
- Only IPoE sessions are qualified for this feature.

Configuring the Leased IP Hold Time Feature

This section describes how to configure the Leased IP Hold Time feature.

Configuring the Leased IP Hold Time feature involves the following procedure.

Configuring Leased IP Hold Time

Use the following commands to configure the leased IP hold time on the DHCP server profile.

```
config
  profile dhcp dhcp_profile_name
  { ipv4 | ipv6 }
  server hold-time minutes timer_interval
  exit
```

NOTES:

- **profile dhcp** *dhcp_profile_name*: Specifies the DHCP profile name.
- **{ ipv4 | ipv6 }**: Enters IPv4 or IPv6 configuration mode.
- **server hold-time minutes** *timer_interval*: Specifies the leased IP hold time in minutes for the IPv4 or IPv6 server. *timer_interval* must be an integer in the range of 1 to 59 minutes.

DHCP IP Lease Reservation

Feature Summary

Table 41: Feature Summary

Applicable Product(s) or Functional Area	cnBNG
Applicable Platform(s)	SMI
Feature Default Setting	Disabled - Configuration Required
Related Documentation	Not Applicable

Revision History

Table 42: Revision History

Revision Details	Release
First introduced	2022.04.0

Feature Description

DHCP IP Lease Reservation feature enables the DHCP to allocate an IP address dynamically when the subscriber logs into the network the first time. Then, the assigned IP address can be reserved permanently for the subscriber, which means, the same IP address is assigned every time the subscriber logs in.

How it Works

This section provides a brief of how the DHCP IP Lease Reservation feature works.

After the DHCP IP Lease Reservation feature is enabled (see Configuring DHCP IP Lease Reservation, on page 245), if a subscriber (CPE) logs into the system for the first time, IPAM allocates an IP address dynamically from the IP pool. Administrators can use the REST API/action command (see Reserving IP Address using CLI (Action Command/REST API), on page 245) to reserve the IP address for the subscriber. So, when the same session is initiated the next time, the DHCP provides the same IP address to the subscriber.

Note If you do not want to reserve the IP address, the administrators can use the same REST API/action command with **delete** option and clear the IP lease reservation.

Limitations and Restrictions

The DHCP IP Lease Reservation feature has the following limitation:

- The DHCP IP Lease Reservation and Leased IP Hold Time features cannot be used together at the same time.

Configuring DHCP IP Lease Reservation

Use the following commands to enable/disable the DHCP IP Lease Reservation feature:

```
config
   [ no ] subscriber featurette dhcp-lease-reservation enable
end
```

NOTES:

- **subscriber featurette dhcp-lease-reservation enable**: Enables the DHCP IP Lease Reservation feature
- **no subscriber featurette dhcp-lease-reservation enable**: Disables the DHCP IP Lease Reservation feature

Reserving IP Address using CLI (Action Command/REST API)

Administrators can use the following action command/REST API to reserve the addresses (IPv4, IANA, and IAPD) that are allocated to the subscriber with a specific username.

```
bng# subscriber lease-reservation subkey username_string [ delete ]
```

NOTES:

- **subkey** *username_string* : Specifies the username for which the IP addresses are reserved.
- **delete**: Clears the lease reservation for the specific username.

Note

- This command/REST API fails if the subscriber is disconnected.
- This command/REST API fails if the DHCP IP Lease Reservation feature is not enabled.

Enhanced support for RADIUS attributes

We have enhanced the RADIUS attributes, especially Stateful-IPv6-Address-Pool, and Delegated-IPv6-Prefix-Pool. These enhancements are crucial for effectively managing and optimizing your modern network infrastructure.

Stateful-IPv6-Address-Pool support

Stateful-IPv6-Address-Pool is a RADIUS attribute that

- enables the RADIUS server to send the name of an assigned pool for a subscriber to the cnBNG Control Plane in Access-Accept, which uses it to assign IPv6 addresses to subscribers via the DHCPv6 protocol
- overrides any local IPv6 address pool configured in the cnBNG CP DHCPv6 profile, and
- is defined in RFC 6911 - *RADIUS Attributes for IPv6 Access Networks.*

Table 43: Feature History

Feature Name	Release Information	Description
Stateful-IPv6-Address-Pool support	2025.02.0	We have enhanced IPv6 address allocation by enabling the RADIUS server to specify the assigned pool name to cnBNG. The cnBNG uses this information to assign IPv6 addresses to subscribers.

IPv6 address allocation

This attribute specifies the name of a pool that must be used by the cnBNG to select an IPv6 address for a user.

Stateful-IPv6-Address-Pool fields

Attribute	Type	Length	String
Stateful-IPv6-Address-Pool	172	Minimum length in bytes: 3.	Contains the name of an assigned IPv6 stateful address pool configured on the cnBNG. This field is not terminated by NULL (hexadecimal 00).

Configure Stateful-IPv6-Address-Pool attribute

You can configure the **Stateful-IPv6-Address-Pool** attribute using the user-profile on the RADIUS server. You can then directly receive it from the RADIUS server. No configuration is needed on the cnBNG CP side.

Procedure

Step 1 Define the **Stateful-IPv6-Address-Pool** attribute on the RADIUS server.

Example:

```
user@example.com    Password="abc"
    Service-Type=Framed-User,
    Stateful-IPv6-Address-Pool="dhcpv6-iana-pool"
```

Step 2 Use the **show subscriber session detail** command to verify the configuration.

Example:

```
bng# show subscriber session detail
Mon Apr  7  10:05:13.666 UTC+00:00
subscriber-details
{
  "subResponses": [
    {
      "subLabel": "16777219",
      "mac": "cc11.0000.0001",
      "acct-sess-id": "01000003",
      "upf": "asr9k-1",
      "port-id": "Bundle-Ether1.100",
      "up-subs-id": "3",
      "sesstype": "ppp",
      "state": "established",
      "subCreateTime": "Mon, 07 Apr 2025 10:04:53 UTC",
      "subsAttr": {
        "attrs": {
          "Authentic": "RADIUS(1)",
          "Framed-Protocol": "PPP(1)",
          "addr": "33.0.0.4",
          "addrv6": "2:3::202",
          "circuit-id-tag": "circuit1",
          "client-mac-address": "cc11.0000.0001",
          "connect-progress": "IPV6CP Open",
          "delegated-prefix": "3002:db0:0:1::/64",
          "dhcpv6-client-id": "0x00030001cc1100000001",
          "inner-vlan-id": "200",
          "outer-vlan-id": "100",
        }
      },
      "subcfgInfo": {
        "committedAttrs": {
          "attrs": {
            "Stateful-IPv6-Address-Pool": "dhcpv6-iana-pool",
            "accounting-list": "aaa-prof1",
            "acct-interval": "2000",
            "addr-pool": "ipv4-pool",
            "delegated-ipv6-pool": "dhcpv6-iapd-pool",
```

```
"session-acct-enabled": "true",
"vrf": "default"
```

Delegated-IPv6-Prefix-Pool support

Delegated-IPv6-Prefix-Pool is an attribute that

- allows a RADIUS server to send the name of a prefix pool for a subscriber to the cnBNG Control Plane in Access-Accept, which uses it to assign IPv6 prefixes to subscribers via the DHCPv6 protocol
- overrides any local IPv6 prefix pool configured in the cnBNG CP DHCPv6 profile, and
- is defined in RFC 6911 - *RADIUS Attributes for IPv6 Access Networks.*

Table 44: Feature History

Feature Name	Release Information	Description
Delegated-IPv6-Prefix-Pool support	2025.02.0	We have enhanced IPv6 prefix delegation by allowing the RADIUS server to specify a prefix pool name to cnBNG. The cnBNG uses this information to assign IPv6 prefixes to subscribers.

IPv6 prefix delegation

This attribute specifies the name of a pool that should be used by the cnBNG to select an IPv6 delegated prefix for a user.

Useful in dynamic IPv6 environments

This attribute is useful in environments where dynamic assignment of IPv6 prefixes is necessary, and it helps streamline the process of prefix delegation by providing predefined pool names that guide the selection process.

Delegated-IPv6-Prefix-Pool attribute fields

Attribute	Type	Length	String
Delegated-IPv6-Prefix-Pool	171	Minimum length in bytes: 3.	Contains the name of an assigned IPv6 prefix pool configured on the cnBNG. This field is not terminated by NULL (hexadecimal 00).

Configure Delegated-IPv6-Prefix-Pool attribute

You can configure the **Delegated-IPv6-Prefix-Pool** attribute using the user-profile on the RADIUS server. You can then directly receive it from the RADIUS server. No configuration is needed on the cnBNG CP side.

Procedure

Step 1 Define the **Delegated-IPv6-Prefix-Pool** attribute on the RADIUS server.

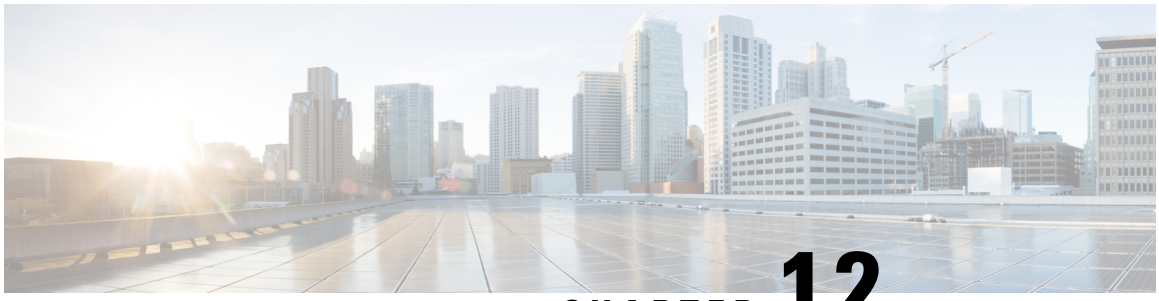
Example:

```
user@example.com    Password="abc"
    Service-Type=Framed-User,
    Delegated-IPv6-Prefix-Pool="dhcpv6-iapd-pool"
```

Step 2 Use the **show subscriber session detail** command to verify the configuration.

Example:

```
bng# show subscriber session detail
Mon Apr  7  10:05:13.666 UTC+00:00
subscriber-details
{
  "subResponses": [
    {
      "subLabel": "16777219",
      "mac": "cc11.0000.0001",
      "acct-sess-id": "01000003",
      "upf": "asr9k-1",
      "port-id": "Bundle-Ether1.100",
      "up-subs-id": "3",
      "sesstype": "ppp",
      "state": "established",
      "subCreateTime": "Mon, 07 Apr 2025 10:04:53 UTC",
      "subsAttr": {
        "attrs": {
          "Authentic": "RADIUS(1)",
          "Framed-Protocol": "PPP(1)",
          "addr": "33.0.0.4",
          "addrv6": "2:3::202",
          "circuit-id-tag": "circuit1",
          "client-mac-address": "cc11.0000.0001",
          "connect-progress": "IPV6CP Open",
          "delegated-prefix": "3002:db0:0:1::/64",
          "dhcpv6-client-id": "0x00030001cc1100000001",
          "inner-vlan-id": "200",
          "outer-vlan-id": "100",
        }
      },
      "subcfgInfo": {
        "committedAttrs": {
          "attrs": {
            "Stateful-IPv6-Address-Pool": "dhcpv6-iana-pool",
            "accounting-list": "aaa-prof1",
            "acct-interval": "2000",
            "addr-pool": "ipv4-pool",
            "delegated-ipv6-pool": "dhcpv6-iapd-pool",
            "session-acct-enabled": "true",
            "vrf": "default"
```



CHAPTER 12

End-to-End Flow Control

- Feature Summary and Revision History, on page 251
- Feature Description, on page 252
- Simplified Control Plane flow control configurations, on page 252
- FSOL Token Mechanism, on page 256
- Limitations and Restrictions, on page 257
- How it Works (Manual Configuration), on page 258
- Configuring End-to-End Flow Control (Manual Configuration), on page 259

Feature Summary and Revision History

Summary Data

Table 45: Summary Data

Applicable Product(s) or Functional Area	cnBNG
Applicable Platform(s)	SMI
Feature Default Setting	Disabled - Configuration Required
Related Documentation	Not Applicable

Revision History

Table 46: Revision History

Revision Details	Release
Introduced support for simplified CP flow control configuration.	2025.04.0

Revision Details	Release
The Flow Control feature supports the following functionalities: • Different rate limit and queue size for different UPFs • Packet priority and differential treatment to packets based on packet type or DSCP value of the packet • FSOL token mechanism	2022.03.0
First introduced.	2021.04.0

Feature Description

Note This feature is Network Services Orchestrator (NSO) integrated.

The Cloud Native Broadband Network Gateway (cnBNG) manages residential subscribers from different access planes in a centralized way. It accepts and identifies subscriber Control Plane (CP) traffic coming from multiple User Planes (UPs) associated with the CP. When the number of UPs scale, the amount of CP traffic coming from each UP multiplies.

The traffic flow between CP and UP must be regulated to ensure that the CP attends all the service requests without service interruption.

The following scenarios create burstiness or higher flow rates in the traffic flows:

- Power outage in a residential area
- Access network outage for a specific period
- UP catastrophic events such as process crash, route processor reboots, and chassis reload

These scenarios generate sudden spike in traffic going to the CP. To handle these traffic spikes, it is necessary to flow control and rate limit the CP ingress to ensure that service applications are not overwhelmed with these bursts. The End-to-End Flow Control feature optimizes flow control and rate limit of the traffic toward the CP ingress.

Simplified Control Plane flow control configurations

A simplified Control Plane flow control configuration is a network automation feature that

- derives dispatcher and overload control settings automatically from user-supplied Calls per Second (CPS) values
- adapts dynamically as User Plane Functions (UPFs) are added or removed from the network, and

- allows for manual overrides when specific customization is required.

Prior to cnBNG Release 2025.04.0, configuring flow control and overload control for User Plane Functions (UPFs) required complex manual steps. Each UPF and interface demanded individual dispatcher and overload control settings, making operations difficult to scale and maintain, especially in environments where UPFs change frequently.

Starting with cnBNG Release 2025.04.0, a simplified configuration model introduces automation by basing dispatcher and overload controls on user-supplied CPS values. This model streamlines setup and adjustment, improves operational consistency, and dramatically reduces manual effort.

Key attributes

- **User Plane CPS (Calls Per Second)**: Each UPF is configured according to its maximum supported CPS value to accurately represent its capacity.
- **CP Cluster CPS:** The cluster-wide configuration sets total Control Plane processing capability. You can optimize rate limiting for specific session types such as **all-types**, **ipoe-only**, or **pppoe-lns-only** to match the network's traffic profile.
- **Automatic adjustment:** Whenever CPS values or UPF counts change, the system recalculates all relevant dispatcher and overload settings automatically, eliminating manual reconfiguration.

Note

- You can change CPS values at any time. The system will update internal settings accordingly. Note that such changes may affect processing of packets already in queues.
- The optional **session-type** parameter enables further optimization based on the predominant protocol used (IPOE or PPPOE). By default, mixed traffic is assumed.

Benefits of simplified CP Flow Control configuration

- Reduced configuration complexity.
- Improved consistency and reliability across UPFs and clusters.
- Faster adaptation to changes in network design or capacity.
- More accurate and efficient resource usage.

Simplified CP Flow Control configuration example

Suppose a network operator adds new UPFs to handle increased subscriber load. With simplified Control Plane flow control configuration, the operator specifies the CPS for each new UPF and updates the cluster CPS if needed. The system immediately recalculates and applies the necessary dispatcher and overload settings, with no manual recalibration required.

Configure user-plane and subscriber CPS values

Configure the CP flow control system for automated dispatcher and overload control, improving efficiency and consistency.

Use this task when adding or updating user-plane functions (UPFs) or adjusting cluster-wide performance limits. This approach replaces complex manual configuration with a simplified model.

Before you begin

- Identify the CPS capabilities of each UPF you want to configure.
- Determine the total processing capacity (CPS) available in the CP cluster.
- Decide if your sessions are primarily of a single type (ipoe-only, pppoe-lns-only) or mixed (all-types).

Follow these steps to configure automated CP flow control:

Procedure

Step 1 For each user-plane, specify the maximum supported CPS value in the configuration.

Example:

```
configure
  user-plane
    instance 1
      user-plane ASR9k-0001
        cps 200
      exit
```

Step 2 At the subscriber (cluster) level, set the total available CPS and, optionally, the session type (**all-types, ipoe-only, or pppoe-lns-only**) to refine flow control parameters. **all-types** is the default.

Example:

```
configure
  subscriber
   cps 2000 [ session-type all-types | ipoe-only | pppoe-lns-only ]
exit
```

Important The configuration becomes active only when you assign CPS values under user-plane or subscriber settings.

NOTES:

- **cps**: Specifies the calls per second value. There is no default CPS value. You must assign a specific value to enable overload-control for the cluster.
- **session-type ipoe-only:** All sessions use the DHCP protocol.
- **session-type pppoe-lns-only:** All sessions use the PPPoE protocol.
- **session-type all-types:** (Default) Sessions can be a mix of both DHCP and PPPoE.

Migrate CP flow control models

You can change the operational mode of CP Flow Control between manual and automated methods for clusters, supporting both geographical redundancy (GR) and non-GR setups.

Use this task to ensure that your CP Flow Control model matches deployment requirements and maintains operational stability during migration. Procedures differ for GR and non-GR configurations.

Upgrade from manual to automated method on a GR setup

Migrate a system with Geographical redundancy (GR) from manual configuration to an automated method.

Before you begin

- Verify that both remote and local clusters are accessible.
- Ensure you have backups of all existing configurations.

Follow these steps to upgrade from manual to automated method on a GR setup:

Procedure

Step 1 Make the remote cluster active for both instances.
Step 2 Shut down the local cluster.
Step 3 Remove the old configuration and apply the new configuration.
Step 4 Bring up the local cluster and restore GR roles as needed.

Upgrade from manual to automated method on a non-GR setup

Migrate a standard (non-GR) cluster from manual configuration to an automated method.

Before you begin

Ensure you have backups of all existing configurations.

Follow these steps to upgrade from manual to automated method on a non-GR setup:

Procedure

Step 1 Shut down the cluster.
Step 2 Remove the old configuration and apply the new configuration.
Step 3 Bring up the cluster.

Downgrade from automated to manual method on GR and non-GR setups

Revert from automated to manual configuration on either Geographical redundancy (GR) or non-GR clusters.

Before you begin

Ensure the system is in a stable state and you have reliable backups for both old and new configurations.

Follow this step to downgrade from automated to manual method on GR and non-GR setups:

Procedure

In a single commit transaction, remove the new configuration and apply the old configuration.

FSOL Token Mechanism

The FSOL token mechanism is a protection method to control the maximum inflight transactions on the control plane at each FSOL pod. This mechanism addresses higher packet rates and works along with UDP proxy flow control to protect the control plane. The UDP proxy flow control configuration acts as the first-level check in the control plane to control incoming GTPU and PFCP message rates. Token mechanism does not support rate control.

To receover from token leakage or stale token usage, FSOL supports the use of the action command **subscriber reset-token { dhcp | pppoe }** to reset the in-use token to zero.



Note During switchover, if both GRs are non-primary, reset the token count to ensure counters start from zero when a GR becomes active again.

FSOL token mechanism supports the following functionalities:

- Token mechanism is applicable for IPoE, PPPoE, and L2TP session types.

  This feature does not support per UPF token.

- DHCP and PPPoE have separate token configurations. DHCPv6 over PPPoE consumes the DHCP token.
- Only transactions that deal with IPCs with multiple pods or that result in interaction with external entities consume the token. The external entities can be Policy plane interaction, N4 PFCP Establishment, N4 PFCP Modification. Handling of DHCPv4 Renew, DHCPv4Rebind, DHCPv4 Information and DHCPv6 Renew, DHCPv6 Rebind, DHCPv6 Information-Request packets will not consume the token.
- Acquires the token and releases across transactions.
- A Discover storm without any request packet for the offered IPs can consume all the tokens. Since the tokens that are acquired during Discover processing are released only after processing the request from CPE. If CPE does not send the request, the token is released only after the temporary lease expiry (around 1 minute).
- IPoE lease expiry handling consumes the token. If the token is not available, the lease expiry handling is postponed with an exponential back off up to 8 minutes.
- If the token is not available when the hold timer expires, then the hold timer is extended for a minute.
- DHCP admin clear consumes the token. If the token is not available, the session will not get cleared.

- Internal failure handling consumes the token. If the tokens are not available, then failure handling proceeds uninterrupted as failure cannot be cached.
- FSOL token will not be consumed in the following scenarios:
  - Session manager admin clear
  - Disconnect message from session manager (due to pod or PPPoE session disconnect)
  - Notify conflict handling, failure callback handling for CDL update, and CP reconciliation
- Caches packets if tokens are unavailable during release packet handling.

Configuring Token for FSOL Pod

To configure the FSOL token mechanism, use the following sample configuration:

```
config
  subscriber token { dhcp | pppoe } token_count
  exit
```

NOTES:

- **subscriber token { dhcp | pppoe }** *token_count*: Set the maximum token available for the FSOL pod. *token_count* is cumulative across instances. For example, if there are 4 DHCP pods and DHCP token is set as 2000, then 500 tokens will be assigned for each pod.
  - **dhcp**: Set the DHCP pod token count.
  - **pppoe**: Set the PPPoE pod token count.

Limitations and Restrictions

The End-to-End Flow Control feature has the following limitations and restrictions:

- Session bring-down rate (DHCP release, PPPOE PADT, L2TP, CDN rate control) cannot be enforced using the CP flow control configuration. Also, UP does not have flow control of these packets. Therefore, solution level flow control for session disconnect triggers for all session types is not supported.
- Packet level flow control for DHCPv4 and DHCPv6 Renew, and DHCPv6 Relay forwarded messages is not supported.
- L2TP LAC and LNS FSOL rate control are not supported on the ASR 9000 UP in this release. The CP does not have rate control based on FSOL. Because PPPoE bring-up controls LAC, PPPoE FSOL rate control on ASR 9000 can be used to control LAC session bring-up.
- Flow control must be configured at the UP level for the following packets at the UPF. This ensures that the packet rate from UP to CP is controlled because CP cannot provide per packet rate control, per UPF.
  - FSOL
  - Session delete notifications
  - LCP keepalive failure notifications

- Session statistics report

How it Works (Manual Configuration)

Note This section is not applicable if you use the automated configuration method.

This section describes how End-to-End Flow Control works in cnBNG.

There are two types of traffic that enter or exit the CP:

- Control traffic that is responsible for subscriber session creation
- Control traffic that is already provisioned for a subscriber session

The following application infrastructure (App-Infra) features facilitates the cnBNG CP ingress packet flow control:

- Dispatcher, on page 258
- Overload Control, on page 259

Dispatcher

In the dispatcher, if you configure the right dequeue rate, the packets do not pile up in the PFCP queue. The dequeue rate must be higher than the incoming rate from the UP.

All PFCP packets land into a single queue because there is no packet segregation. Any rate control that is applied on this queue is per UPF PFCP packet rate control. It is not possible to control a particular type of packet per UPF. For example, DHCP release, PPPoE PADT, or keepalive failure notification packet cannot be controlled per UPF at the dispatcher queue.

The dispatcher queue size configuration handles the burst of packets. This functionality supports the following:

- Dedicated queue for each PFCP or N4 interface, and GTPu interface for each UPF connected to the control plane
- Configuration of queue size and flow control rate limits

Different Rate Limit and Queue Size Support

For N4 or GTPu dispatcher, the flow control feature supports different rate limits and queue sizes for different UPFs. Depending on the UPF's capability, the rates and queue sizes vary for all UPFs.

The rate limit and queue size are configurable for different UPFs. You can configure the flow control group under the endpoint udp-proxy configuration. You can associate the flow control group in the user-plane configuration.

Packet Priority and Differential Treatment to Packets

cnBNG supports packet priority and differential treatment to packets based on the packet type or DSCP value in the packet. When the system is congested or queues are full, certain incoming packets such as PFCP heartbeats, session reports, and subscriber redundancy group notification messages are treated with high priority. These packets will be delivered to the destination without delay and will not be dropped if the queues are full.

The PFCP Heartbeats, PFCP Association Update or Release, and PFCP Node Report packets that cannot be rate controlled and dropped will be bypassed from the dispatcher queues and overload control queues.

Overload Control

Overload control is applied to a packet after it is released from a dispatcher. This creates a queue based on the packet type at the aggregate level across all UPF data. Because overload control enables packet type based queues, rate control is applied for that type of packet at the aggregate level of all UPFs.

Special treatment of the packet is indirectly achieved by having different queues for a packet at overload control feature and aggregate of all UPF level.

The dispatcher supports the following catogories of virtual message groups:

- PFCP keepalive messages between CP and UPF
- PFCP LCP keepalive failure notification messages
- PFCP Response messages
- Session Report messages
- Other message types which are not listed in different categories

The Overload Control feature provides aggregate queues for a message type coming from UPF functions. Group IDs are supported for each message group and the message type is configurable for each group. When configured, a virtual queue is created for each message type and treated based on the configured attributes for that group. For each queue, the size and rate limit can be configured.

For each message, the configured rate of packets are dequeued and sent to the CPF. For priority packets such as PFCP keepalives, dedicated queues are allocated so that they are not impacted with other queue sizes.

Based on the cluster capacity, specific values for each queue and message type must be configured. The values are adjusted based on the capacity.

Configuring End-to-End Flow Control (Manual Configuration)

Note This section is not applicable if you use the automated configuration method.

This section describes how to configure the End-to-End Flow Control feature on Control Plane (CP) manually.

The configuration involves the following procedures:

- Configuring Dispatcher for GTPu Interface, on page 260

- Configuring Dispatcher for N4 Interface, on page 261
- Configuring Overload Control for Message Types, on page 263
- Associating Flow Control Group in User Plane, on page 264
- Configuring Token for FSOL Pod, on page 257

Configuring Dispatcher for GTPu Interface

To configure dispatcher for GTPu interface, use the following commands:

```
config
  instance instance-id instance_id
    endpoint udp-proxy
      interface gtpu dispatcher { cache { true | false } |
      capacity queue_capacity | count queue_count |
      flowctrl-group group_name { capacity inbound_queue_size |
outbound-capacity outbound_queue_size | outbound-rate-limit outbound_rate_limit
| rate-limit inbound_rate_limit } |
      outbound { true | false } | rate-limit rate_limit |
      threshold threshold_value }
      exit
```

NOTES:

- **instance** *instance_id*: Configure multiple instances for the specified instance and enters the instance sub-mode.
- **endpoint udp-proxy**: Configure parameters for the UDP-proxy endpoint and enters the endpoint sub-mode.
- **interface gtpu dispatcher { cache { true | false } | capacity** *queue_capacity* **| count** *count* **| flowctrl-group** *group_name* **{ capacity** *inbound_queue_size* **| outbound-capacity** *outbound_queue_size* **| outbound-rate-limit** *outbound_rate_limit* **| rate-limit** *inbound_rate_limit* **} | outbound { true | false } | rate-limit** *value* **| threshold** *threshold_value* **}**: Specify the dispatcher parameters for the GTPu interface.
  - **cache { true | false }**: Enable (false) or disable (true) cache retransmission support. The default value **false** indicates that the cache retransmission support is enabled.
  - **capacity** *queue_capacity*: Specify the number of packets that this queue holds.

Note Ensure that there is sufficient memory when configuring higher capacity queues.

  - **count** *queue_count*: Specify the number of N4 queues to be created. Each queue is associated or dedicated to an UPF. For example, if the count is 2, two N4 queues are created and two UPs can be connected.
  - **flowctrl-group** *group_name* **{ capacity** *inbound_queue_size* **| outbound-capacity** *outbound_queue_size* **| outbound-rate-limit** *outbound_rate_limit* **| rate-limit** *inbound_rate_limit* **}**: Specify the queue size and rate limit for the specified flow control group.

Note The flow control group name must be the same for N4 and GTPu interfaces for a given group.

- **capacity** *inbound_queue_size*: Specify the capacity of inbound queue.
- **outbound-capacity** *outbound_queue_size*: Specify the capacity of outbound queue.
- **outbound-rate-limit** *outbound_rate_limit*: Specify the rate limit for outbound queue.
- **rate-limit** *inbound_rate_limit*: Specify the rate limit for inbound queue.

- **outbound { true | false }**: Enable (true) or disable (false) queue support for outbound messages. Default value: **false**.

Note Outbound flow control for BNG is not supported.

- **rate-limit** *rate_limit*: Specify the rate limit for each queue, that is, when packets are dequeued. The rate limit is defined in seconds.
- **threshold** *threshold_value*: Specify the queue size before packets are dropped.

Example

The following is a configuration example.

```
interface gtpu
   sla response 150000
   dispatcher
    count 1
    capacity 1000000
    outbound true
    rate-limit 500
    cache true
    threshold 950000
    flowctrl-group group1
     capacity 2000
     rate-limit 200
    exit
   exit
  exit
exit
exit
```

Configuring Dispatcher for N4 Interface

To configure dispatcher for N4 interface, use the following commands:

```
config
   instance instance_id
      endpoint udp-proxy
         interface n4 dispatcher { cache { true | false } |
```

```
        capacity queue_capacity | count queue_count |
        flowctrl-group group_name { capacity inbound_queue_size |
outbound-capacity outbound_queue_size | outbound-rate-limit outbound_rate_limit
| rate-limit inbound_rate_limit } |
        outbound { true | false } | rate-limit rate_limit |
        threshold threshold_value }
        exit
```

NOTES:

- **instance** *instance_id*: Configure multiple instances for the specified instance and enters the instance sub-mode.
- **endpoint udp-proxy**: Configure parameters for the UDP-proxy endpoint and enters the endpoint sub-mode.
- **interface n4 dispatcher { cache { true | false } | capacity** *queue_capacity* **| count** *count* **flowctrl-group** *group_name* **{ capacity** *inbound_queue_size* **| outbound-capacity** *outbound_queue_size* **| outbound-rate-limit** *outbound_rate_limit* **| rate-limit** *inbound_rate_limit* **} | outbound { true | false } | rate-limit** *value* **| threshold** *threshold_value* **}**: Specify dispatcher parameters for the N4 interface.
  - **cache { true | false }**: Enable (false) or disable (true) cache retransmission support. The default value **false** indicates that the cache retransmission support is enabled.
  - **capacity** *queue_capacity*: Specify the number of packets that this queue holds.

Note Ensure that there is sufficient memory when configuring higher capacity queues.

  - **count** *queue_count*: Specify the number of N4 queues to be created. Each queue is associated or dedicated to an UPF. For example, if the count is 2, two N4 queues are created and two UPs can be connected.
  - **flowctrl-group** *group_name* **{ capacity** *inbound_queue_size* **| outbound-capacity** *outbound_queue_size* **| outbound-rate-limit** *outbound_rate_limit* **| rate-limit** *inbound_rate_limit* **}**: Specify the queue size and rate limit for the specified flow control group.

Note The flow control group name must be the same for N4 and GTPu interfaces for a given group.

    - **capacity** *inbound_queue_size*: Specify the capacity of inbound queue.
    - **outbound-capacity** *outbound_queue_size*: Specify the capacity of outbound queue.
    - **outbound-rate-limit** *outbound_rate_limit*: Specify the rate limit for outbound queue.
    - **rate-limit** *inbound_rate_limit*: Specify the rate limit for inbound queue.

  - **outbound { true | false }**: Enable (true) or disable (false) queue support for outbound messages. Default value: **false**.

Note Outbound flow control for BNG is not supported.

- **rate-limit** *per_second*: Specify the rate limit for each queue, that is, when packets are dequeued. The rate limit is defined in seconds.
- **threshold** *threshold*: Specify the queue size before packets are dropped.

Example

The following is an example configuration.

```
endpoint udp-proxy
  replicas 1
  nodes 2
  vip-ip 201.201.201.51
  vip-ipv6 2001::10:1:39.191
  interface n4
   sla response 150000
   dispatcher
    count 1
    capacity 500000
    outbound true
    rate-limit 300
    cache false
    threshold 950000
    flowctrl-group group1
     capacity 1000
     rate-limit 100
    exit
   exit
```

Configuring Overload Control for Message Types

To configure overload control for all message types, use the following commands:

```
config
     overload-control msg-type { all | lcpkeepalive | pfcpkeepalive |
pfcpresponse | sessionreport }
       msg-priority msg_priority | rate-limit rate_value| queue-size queue_size
 | reject-threshold reject_threshold | pending-request pending_request |
discard-behavior { drop | true }
       commit
```

NOTES:

- **overload-control msg-type { all | lcpkeepalive | pfcpkeepalive | pfcpresponse | sessionreport }**: Configure overload control for the specified message type.
- **msg-priority** *msg_priority*: Specify the message priority. This keyword is not applicable in the BNG context.
- **rate-limit** *rate_value*: Specify the rate limit for each queue, that is, when packets are dequeued. The rate limit is defined in seconds.

- **queue-size** *queue_size*: Specify the size of the queue to be created.
- **reject-threshold** *threshold_limit*: Specify the percentage of the pending-request value.
- **pending-request** *pending_request*: Specify the number of packets present in the queue at any time.
- **discard-behavior { drop | true }**: Specify whether to drop or process the packets. Default value: **drop**.

Example

The following is a configuration example.

```
overload-control msg-type all
    rate-limit 13000 queue-size 200000 reject-threshold 95 pending-request 200000
   exit
   overload-control msg-type lcpkeepalive
    rate-limit 1100 queue-size 25000 reject-threshold 95 pending-request 25000
   exit
   overload-control msg-type sessionreport
    rate-limit 1000 queue-size 25000 reject-threshold 95 pending-request 25000
   exit
   overload-control msg-type pfcpkeepalive
    rate-limit 100 queue-size 1000 reject-threshold 95 pending-request 1000
   exit
   overload-control msg-type pfcpresponse
    rate-limit 4000 queue-size 25000 reject-threshold 95 pending-request 25000
   exit
  exit
```

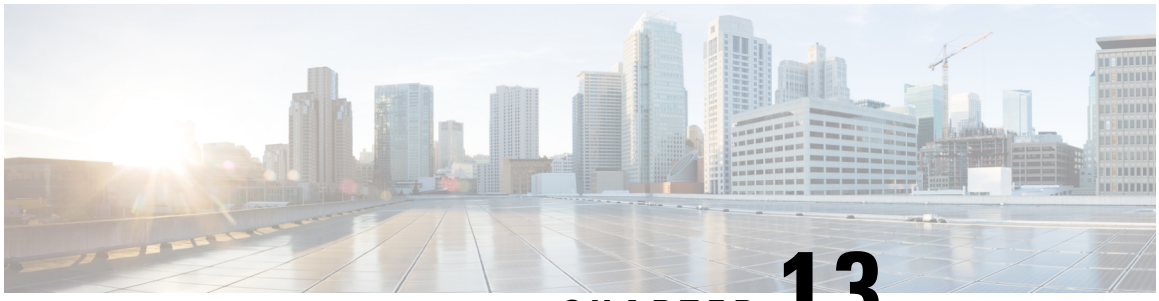
Associating Flow Control Group in User Plane

To associate the flow control group in the user plane, use the following sample configuration:

```
config
   user-plane user_plane_name
      flowctrl-group group_name
      exit
```

NOTES:

- **user-plane** *user_plane_name*: Specify the User Plane (UP) name and enter UP Configuration mode.
- **flowctrl-group** *group_name*: Specify the name of the flow control group to be associated in the user plane.



CHAPTER 13

High Availability and CP Reconciliation

- Feature Summary and Revision History, on page 265
- Feature Description, on page 266
- How it Works, on page 267
- Configuring High Availability and CP Reconciliation, on page 270
- Automatic CP-UP reconciliation for CP events, on page 272

Feature Summary and Revision History

Summary Data

Table 47: Summary Data

Applicable Product(s) or Functional Area	cnBNG
Applicable Platform(s)	SMI
Feature Default Setting	Disabled - Configuration Required
Related Documentation	Not Applicable

Revision History

Table 48: Revision History

Revision Details	Release
Introduced support for automatic CP-UP reconciliation for CP events.	2026.03.0
CP HA and Audit support for LAC and LNS sessions.	2022.03.0
First introduced.	2021.04.0

Feature Description

The high availability (HA) and Reconciliation feature for the control plane supports all cnBNG-specific service pods. This feature supports the IPoE, PTA, LAC, and LNS session types.

CP Audit

When subscriber sessions desynchronize across all pods in the control plane during HA events and session inconsistency, cnBNG runs CP reconciliation (that is, CP audit) to synchronize the sessions.

Note CP reconciliation is also referred to as CP audit in this document.

- Reconciliation between SM and DHCP for IPoE sessions
- Reconciliation between SM, DHCP, and PPP for PTA and LNS sessions
- Reconciliation between SM and PPP for LAC sessions
- Reconciliation between PPP and L2TP for LAC and LNS sessions
- Reconciliation between Node Manager (NM) and FSOLs for all session types

To recover L2TP service after HA events and to avoid service impact, critical information such as L2TP sequence numbers (Ns/Nr), Session Count, and SessionID bitmap must be recovered. The tunnel state for L2TP HA is recovered through recovery from another L2TP service.

Pod Restart

The cnBNG-specific service pods support the pod restart functionality.

This feature supports one BNG-specific service pod restart with a minimum gap of 10 minutes between pod restarts. It also supports one VM restart with a minimum gap of 30 minutes between VM restarts.

The HA support for L2TP is provided using a local peer service to synchronize the necessary L2TP tunnel information and recover it after restart. It also supports hitless operation during pod restart, and the restart of pods without impacting the existing sessions and tunnels. This feature helps to recover the L2TP control connection information such as Ns/Nr sequence numbers, Tunnel Context, and Session Bitmap, and also resume the control channel.

Note VM reboot is not supported for LAC and LNS sessions.

The restart of service pods has the following impact:

- CPU or memory spike can occur if there is a churn of sessions during pod restart. For example, if SM has two replicas such as instance 1 and instance 2, and if instance 1 restarts, there will be spike in the CPU or memory in instance 2.
- Service pod IPCs can timeout if the destination service pod restarts before responding to the ongoing IPCs.

- CDL updates of ongoing sessions can fail and result in desynchronization of sessions between pods.
- If subscriber sessions desynchronize between CP and UP, cnBNG runs CP to UP reconciliation.
- If IP address leaks occur in IPAM, run the IPAM reconciliation CLI command **reconcile ipam**.
- ID leaks (CP SubcriberID and PPPoE Session ID) can occur in the NM.
- Reset of Grafana metrics for the restarted pods.

How it Works

This section describes how the high availability and Reconciliation feature for the control plane works.

CP Reconciliation Process

This section describes the CP reconciliation scenarios and processes.

On issuing the manual CP Audit CLI command, FSOL services (DHCP and PPPoE) start reconciling their respective sessions with the SM service to check if the session exists and if the Audit-ID matches. When this check is passed, it proceeds to the next step, else, FSOL disconnects the session.

In the next step, FSOLs tries to audit the session with the Node Manager (NM) to check if the IP address and ID resources are matching. This is to ensure the consistency of the IP and ID resource across session database and NM.

After reconciliation from FSOLs, SM triggers the final reconcile to remove any extra sessions. At the end of this step, all services are expected to have a consistent session database.

CP reconciliation supports the following functionality:

- Supports a maximum of five CP reconciliations in parallel for different UPs.
- Configure the mandatory **cdl datastore session slot notification max-concurrent-bulk-notifications** CLI command to run CDL bulk notifications in parallel for multiple bulk notification requests. Without this configuration, the CP reconciliation process can be slow.

  For information, see the Configuring CDL Bulk Notifications, on page 271.

  New bulk notification requests are put in queue and these requests are dequeued one at a time when the ongoing request is complete.

  Each CP reconciliation request invokes three bulk notification requests to the CDL. Hence, five CP reconciliation requests invoke a maximum of 15 bulk notifications. With this configuration, the **clear subscriber** CLI command is executed in parallel.

  Each **clear subscriber** CLI command invokes one CDL bulk notification request to the CDL. Executing more than 5 **clear subscriber** and **show subscriber** CLI commands slows down the CP reconciliation process. Therefore, it is recommended to avoid these commands while CP reconciliation is in progress.

- CP reconciliation deletes the session in the following scenarios:
  - Extra sessions in DHCP or PPP compared to SM
  - Extra sessions in SM compared to DHCP or PPP
  - Mismatch in session data between DHCP, PPP, and SM

- Mismatch between IP and ID resources between FSOLs and NM

- When a session is deleted in the CP or UP because of a mismatch, the same deleted session could be present in the CPE. This causes traffic loss for the deleted subscriber until the subscriber is recreated after lease expiry for an IPoE session.

Note

- If any pod (SM, DHCP, or PPPoE) restarts while CP reconciliation is in progress, there may still be a session mismatch across pods even after completing the CP reconciliation.
- CP reconciliation without churn and HA events in CP or UP—if it is executed within the supported TPS limits and sessions across pods in the CP synchronize after completing CP reconciliation.
- CP reconciliation with churn (session bring-up or bring-down, CoA) and no HA events in CP and UP:
  - If CP audit is executed within TPS limits and sessions across pods in CP synchronize after completing CP audit.
  - CP audit reconciles sessions that are created before starting the audit. CP audit does not reconcile sessions that are created after starting audit.
  - CP audit does not reconcile sessions that are updated 60 seconds before audit start time. For example, session update time is T1 and audit start time is T2, if T2 minus T1 is less than or equal to 60 seconds, then that session is not audited.

L2TP Audit Process

Depending on the time of HA event and session or tunnel churn, session inconsistency can occur across CP pods mainly between PPPoE, SM, and L2TP. The L2TP tunnel and session ID syncing is required between NM and L2TP, PPPoE and L2TP respectively.

The L2TP Audit has the following two stages:

1. The PPPoE service is modified to check if the session has L2TP sessions or not. It will sync the session with L2TP.
2. The L2TP service checks the audited bitmap. It sweeps the remaining bitmap to clear any stale or non-matching sessions and synchronizes the PPPoE services. DHCP, SM, and PPPoE services will synchronise for LNS sessions.

Automatic Session Mismatch Detection

An existing Audit ID is incremented and sent to the SM for every new transaction initiated from DHCP or PPP to SM. If the transaction is successful, this Audit ID is stored in DHCP or PPP, and in the SM CDL records.

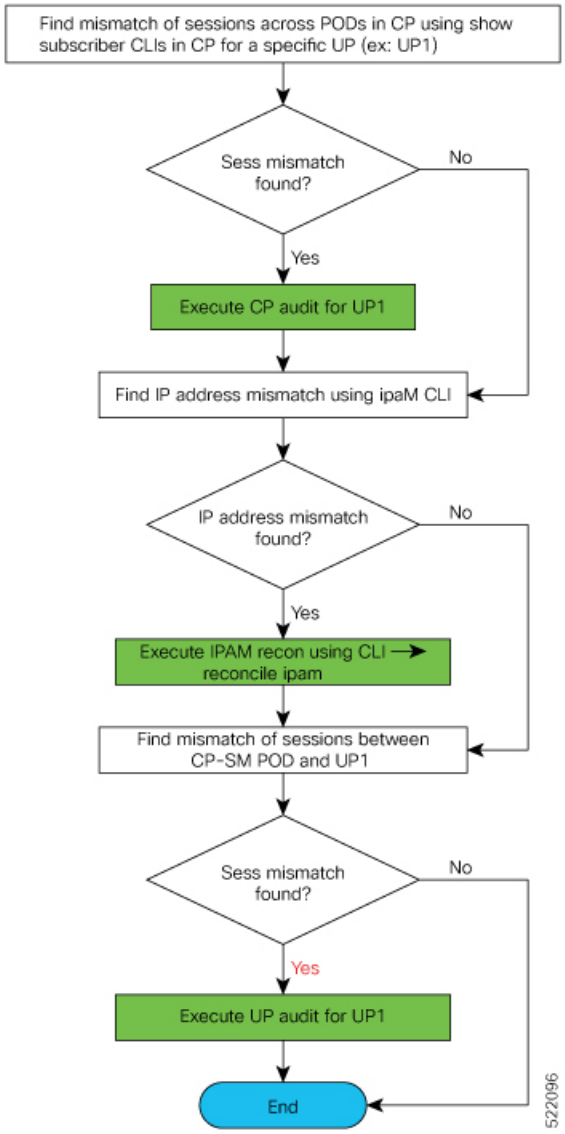
The SM validates the Audit ID received in every request from DHCP or PPP. When a received Audit ID is not incremental to the Audit ID present in the SM, the SM discards the transaction and responds to the DHCP or PPP with an Audit ID mismatch error. The SM then initiates a new transaction to disconnect the session in CP and UP.

Synchronizing Sessions Across CP Pods and UP

CP reconciliation or UP reconciliation (that is, reconciliation between CP-SM pod and UP) is executed for a specific UP.

The following figure depicts the procedure to synchronize sessions across CP pods and UP, for a specific UP.

Figure 15: Synchronizing Sessions Across CP Pods and UP



Limitations and Restrictions

The High Availability and CP Reconciliation feature has the following limitations and restrictions in this release:

- Supports one BNG-specific service pod restart with a minimum gap of 10 minutes between pod restarts.

- Supports one VM restart with a minimum gap of 30 minutes between VM restarts.
- Does not support double fault for infrastructure pods (cache, CDL, and Node Manager). The system goes to "bad" state with double faults.

Configuring High Availability and CP Reconciliation

This section describes how to configure the High Availability and CP Reconciliation feature.

Configuring the High Availability and CP Reconciliation feature involves the following steps:

- Reconciling Sessions Across CP Pods and UP, on page 270
- Configuring CDL Bulk Notifications, on page 271

Reconciling Sessions Across CP Pods and UP

Use the following commands to reconcile subscriber sessions across PPP, DHCP, and SM pods in the CP with the specified UP.

```
subscriber session-synchronize-cp upf upf_name { abort |
     timeout timeout_value | tps tps_value }
```

NOTES:

- **upf** *upf_name* : Configures CP reconciliaton for this UPF.

Note The maximum number of CP reconciliations supported is 5.

- **{ abort | timeout** *timeout_value*| **tps** *tps_value* **}**: Specifies the following parameters for subscriber session reconciliation:

  abort : Aborts the ongoing CP reconciliation for only the specified UPF.

  timeout *timeout_in_seconds* : Specifies the number of seconds the reconciliation can run. If it runs longer than the specified timeout, the reconciliation process is aborted. The valid values range from 2 to 100 minutes. The default value is 60 minutes.

  tps *tps_value* : Specifies the number of notifications sent from the CDL to Node Manager per second. The valid values range from 40 to 1000. The default is 40.

Note Set this value based on the scale and churn of sessions during the CP reconciliation.

Verifying High Availability and CP Reconciliation

Use the following **show** command to verify ongoing or completed CP audit details for the specified UPF.

Note Only one CP audit detail is stored per UPF.

show subscriber synchronize-cp upf *upf_name*

Example

The following is a configuration example.

```
[cnbng-smi-40g-tb03/bng] bng# show subscriber synchronize-cp upf lps_asr9k-1
subscriber-details
{
"Audit ID": 1634722199,
"Session Audit Statistics": {
"DHCP": {
"Audit State": "Completed",
"Session Count": 430,
"Notifications Received": 430
},
"LNS": {
"Audit State": "N/A",
"Session Count": 0,
"Notifications Received": 0
},
"PTA & LAC": {
"Audit State": "N/A",
"Session Count": 0,
"Notifications Received": 0
},
"Session Manager": {
"Audit State": "Completed",
"Session Count": 404,
"Notifications Received": 404
}
},
"Audit State": "Completed",
"Notifications/Sec": 40,
"Timeout": 6000,
"Audit Started": "2021-10-20 09:29:59 +0000 UTC",
"Fsol Audit Started": "2021-10-20 09:29:59 +0000 UTC",
"SM Audit Started": "2021-10-20 09:30:10 +0000 UTC",
"Audit Ended": "2021-10-20 09:30:22 +0000 UTC",
"Total Time Taken": "23 Seconds"
}
```

Configuring CDL Bulk Notifications

Use the following commands to run CDL bulk notifications in parallel for multiple bulk notification requests.

Note This is a mandatory configuration for CP reconciliation.

```
config
  cdl datastore session slot notification max-concurrent-bulk-notifications
 value
  exit
```

NOTES:

- **max-concurrent-bulk-notifications** *value*: Specifies the maximum number of bulk task notifications that CDL can process concurrently. The valid values range from 1 to 20.

  Configure this value to 20 for CP reconciliation.

Sample Configuration

The following is a sample configuration of the CDL bulk notification where a maximum of 20 bulk notifications are executed in parallel for multiple bulk notification requests.

```
config
   cdl datastore session slot notification max-concurrent-bulk-notifications 20
   exit
```

Automatic CP-UP reconciliation for CP events

Automatic CP-UP reconciliation for CP events is a cnBNG feature that

- automatically reconciles Control Plane (CP) and User Plane (UP) subscriber state after certain UPF status changes
- runs as an in-cluster process so no external orchestration tool is required, and
- corrects out-of-sync state through existing internal management interfaces without manual operator steps.

Table 49: Feature History Table

Feature Name	Release Information	Feature Description
Automatic CP-UP reconciliation for CP events	Release 2026.04.0	Added support for UP RPFO, CP-UP connected LC reload, and UPF NAL process restart or upgrade scenarios.
Automatic CP-UP reconciliation for CP events	Release 2026.03.0	You can now reduce downtime and manual intervention after CP or UP events, as this feature automatically detects and reconciles out-of-sync subscriber state between the CP and UP following eligible events, with built-in protection against repeated unstable transitions when enabled.

Benefits

- **Eliminates manual resync effort:** Automatically reconciles CP and UP subscriber state after CP-UP Heartbeat flaps or UPF reload events, replacing the manual resync operators must perform today.
- **Runs without external tooling:** Operates as an in-cluster process, so no external orchestration product is required to trigger or manage resync.
- **Deterministic, repeatable corrections:** Executes a fixed, ordered correction sequence (SRG sync → route sync → CP-UP session sync) every time, rather than relying on ad hoc manual steps.

- **Resilience across restarts**: Persists job and flap state to support automatic resumption of in-progress jobs following a same-environment container restart. For major events like full-pod, node, or GR failovers, job state is preserved to allow for manual operator-led recovery, ensuring system consistency.
- **Prevents disruptive over-triggering** : Flap protection acts as a safety governor, automatically throttling resync attempts for unstable UPFs (For example 3 flaps within 30 minutes). This persistent, stateful mechanism prevents redundant signaling and resource exhaustion, while ensuring full operational visibility through dedicated status commands and proactive alert notifications.
- **Safe-by-default operation**: Feature is disabled until explicitly enabled by an authorized operator, and corrections are applied only through existing internal management interfaces rather than writing directly to UPF configuration.
- **Reduces stale DHCP sessions during switchover**: Handles DHCP in-flight session deletion cases (Client Release, Clear Subscriber, DHCP Lease Expiry, DHCP Hold Timer Subscriber) occurring in parallel with a CP GR switchover.

Operational restrictions for automatic CP-UP reconciliation

These restrictions apply to the automatic CP-UP reconciliation for CP events feature.

- Supported only on SMI CNDP-based cnBNG CP deployments.
- Not supported on OpenShift unless specifically qualified.
- Parallel execution of automatic reconciliation jobs is not permitted - the worker pool size is fixed at one per cluster.
- Only CP-UP Heartbeat flaps, UPF reload, UPF reload-timestamp changes (UP RPFO, CP-UP connected LC reload, and UPF NAL process restart or upgrade), and inactive-to-active UPF events trigger automatic resynchronization jobs.
- Active-to-inactive UPF transitions abort any running automatic reconciliation jobs without triggering new jobs.
- Automatic reconciliation is not triggered if a UPF that is marked as offline transitions to the ACTIVE state.
- Automatic reconciliation is not supported for reassociations initiated via the **clear-cp-assoc** CLI command.
- Double switchover is not supported. If a CP GR role change occurs while a job is running, the job is aborted or continued based on the response received from the node manager and session manager for the scheduled reconciliation.



Note Automatic reconciliation is a best-effort mechanism for cleaning up stale or orphaned sessions. A 100% CP-UP session count match cannot be guaranteed if session churn is high, due to in-flight sessions and timing discrepancies.

Scenarios supported

Scenario / Condition	Support Status	Notes
CP-UP Heartbeat flap	Supported	Primary trigger condition for the current release.
UPF reload (Reassociation when CP-UP already associated)	Supported	Supported when it results in an eligible UPF recovery or reassociation trigger.
CPGR SWO handling for DHCP session deletion	Supported	Use cases include: Client Release, Clear Subscriber, DHCP Lease Expiry, and DHCP Hold Timer Subscriber.
UP RPFO (Remote Processor Failureover)	Supported	Introduced in 2026.04. Triggered by reload timestamp identity change. The Active to Initialized to Active sequence seen during an RPFO is not counted as a flap by flap protection.
CP-UP connected LC reload or OIR	Supported	Introduced in 2026.04. Triggered by reload timestamp identity change.
UPF NAL process restart or upgrade	Supported	Introduced in 2026.04. Triggered by reload timestamp identity change.
Reassociation via explicit clear-cp-assoc CLI	Not Supported	Explicit CLI-driven reassociation is not an automatic trigger source in Release 2026.04. Manual synchronization is recommended.
Cascade case: reconciliation requested by another CP while a reconciliation is already scheduled for the same UPF	Supported	In 2026.03, the first reconciliation for the UPF is accepted and a second request from another CP for the same UPF is rejected while the first is ongoing. From 2026.04, the UP returns recon_retry so that the request can be retried.

Flap protection

Flap protection is an automated safety mechanism designed to maintain system stability within the cnBNG CP-UP Autoresync feature. It functions as a governor that suppresses reconciliation triggers for unstable User Plane Functions (UPFs).

The mechanism continuously monitors for Active-to-Inactive state transitions. When a UPF exhibits frequent flapping behavior, the system dynamically throttles automatic reconciliation to achieve the following:

- Resource conservation: Minimizes unnecessary CP and UP compute utilization during periods of instability.
- Signaling efficiency: Prevents disruptive management REST traffic and redundant synchronization attempts.
- Operational integrity: Avoids race conditions and state inconsistencies that can occur when attempting to perform reconciliation on a node that is not stable.

Protection states

Protection is tracked for each (UPF, GR instance) tuple, utilizing these operational states:

- Normal: Automatic reconciliation is allowed.
- Backoff: The flap threshold was reached within the sliding window. Automatic triggers for the UPF are temporarily throttled. When the backoff period elapses and the flap count has decayed, the UPF returns to Normal.
- AdminActionRequired: The UPF flapped again during Backoff. Automatic reconciliation remains suppressed until you manually clear the flap state with the subscriber `auto-resync clear-flap-state` command, or until the state auto-clears after the configured idle period.

By enforcing these thresholds, flap protection ensures that the Autoresync feature operates only under conditions where the network is sufficiently stable to achieve a successful and meaningful synchronization.

By default, 3 flaps within a 30-minute window cause the UPF to enter Backoff, the minimum backoff period is 300 seconds, and AdminActionRequired auto-clears after 1 hour without flapping. You can change these values using the auto-resync flap-protection configuration.

Note that an active-to-inactive transition also aborts any job currently running for that UPF and counts as one flap iteration.

Restart and recovery

The Autoresync feature provides stateful recovery mechanisms to ensure operational continuity following system events. Job metadata, history, and flap-protection states are maintained in a replicated cache.

- Persistence and Lifecycle: Job data and flap-protection states survive container and pod restarts. Note that this history is cleared following a fresh deployment or a system shut and start sequence.
- Container-level recovery: If only the Autoresync container restarts within an existing pod, interrupted jobs are automatically requeued and resumed.
- System-level recovery: Following a whole-pod restart, node restart, or CP GR failover, interrupted jobs are transitioned to a "held" state to prevent automated interference with system recovery, requiring operator intervention to resume.
- State-dependent behavior: If the automatic reconciliation state is disabled at the time of system startup, interrupted jobs are held and will not resume until the feature is explicitly re-enabled.
- Crash-loop protection: A safety guard limits the number of automatic resume attempts. This prevents a job that consistently causes container instability from entering an indefinite restart loop.

Job execution

Reconciliation jobs follow a deterministic, sequential execution pipeline. Each job consists of multiple phases, with execution governed by the following policies:

- Each phase operates within a strictly bounded timeout period to ensure the system remains responsive. You can configure individual timeout values for both the SRG reconciliation and session synchronization phases to align with your network scale.
- The system enforces an *Aborted* policy by default. If a phase fails, the reconciliation pipeline immediately halts, subsequent phases are skipped, and the job is marked as *Aborted*.

Configure automatic CP-UP reconciliation for CP events

Enable automatic CP-UP reconciliation for CP events.

By default, automatic CP-UP reconciliation is disabled after deployment. You must explicitly enable it before eligible events trigger reconciliation jobs.

- Use this procedure to activate automatic reconciliation and manage job and flap state.

Procedure

Step 1 Use the **subscriber auto-resync state enable** command in config mode to enable automatic reconciliation.

Example:

```
configure
   subscriber auto-resync state enable
```

Step 2 (Optional) Use the **subscriber auto-resync state disable** command in config mode to disable automatic reconciliation.

Example:

```
configure
   subscriber auto-resync state disable
```

Step 3 Use the **show subscriber auto-resync summary** command to view the current automatic reconciliation summary, including job history.

Example:

```
bng# show subscriber auto-resync summary
{
  "job-id": "b81d04a2-2c19-4e7e-9f3a-77e1c0d9aa10",
  "upf": "upf2",
  "result": "in-progress",
  "phase-name": "cp-up-recon",
  "phase-label": "CP-UP Session Reconciliation",
  "phase-order": 3,
  "start-time": "2026-06-30T08:20:10Z"
}
```

This command displays the 10 most recently updated jobs across the fleet, including status and whether automatic reconciliation is enabled.

The result field is one of in-progress, pass, fail, partial, aborted, cancelled, interrupted, or resuming. The trigger-reason field shows why the job was started, for example cp-up reassociation. For a job that aborted or failed, the failure-reason field shows the reason.

Step 4 Use the **show subscriber auto-resync detail [job-id** *uuid* **] [upf** *upf-name* **]** command to view detailed phase-level information for reconciliation jobs.

Example:

```
bng# show subscriber auto-resync detail
{
  "job-id": "3f2a9c1e-7b44-4f0a-9d2e-1c5b6a8e0f31",
  "upf": "upf1",
  "result": "pass",
  "start-time": "2026-06-30T08:14:22Z",
  "end-time": "2026-06-30T08:14:51Z",
```

```
  "phases": [
    { "phase-name": "srg-recon",   "phase-label": "SRG Group Reconciliation",    "phase-order": 1,
"result": "pass", "phase-details": "SRG reconciliation triggered" },
    { "phase-name": "route-recon", "phase-label": "Route Reconciliation",         "phase-order": 2,
 "result": "pass", "phase-details": "route sync triggered and completed" },
    { "phase-name": "cp-up-recon", "phase-label": "CP-UP Session Reconciliation", "phase-order": 3,
 "result": "pass", "phase-details": "CP-UP session sync triggered" }
  ]
```

This command displays the 10 most recently updated jobs across the fleet, including phase details, outcomes, and failure reasons. Review outcomes and any failure reasons for troubleshooting.

Each phase can also show an attempts count and a failure-reason if it was retried or failed. If you specify a job-id, only that job is displayed, with its full phase history.

- Use **job-id** or **upf** parameters to investigate specific jobs or UPF instances, referencing details from the summary output.

Step 5 Clear flap-protection state for a specific UPF if it is throttled or locked.

Example:

```
bng# subscriber auto-resync clear-flap-state upf upf1
```



Warning Flap protection prevents excessive reconciliation attempts during instability. Clearing flap state prematurely may cause repeated jobs or ongoing instability. Only clear flap state after resolving the underlying issue.

Step 6 (Optional) Use the auto-resync flap-protection commands in config mode to change the flap-protection thresholds.

Example:

```
configure
   auto-resync flap-protection slide-window-min 30
   auto-resync flap-protection max-events 3
   auto-resync flap-protection backoff-sec 300
   auto-resync flap-protection disable-auto-clear-hr 1
```

The default values are 30 minutes for the sliding window (slide-window-min), 3 flaps for the threshold (max-events), 300 seconds for the minimum backoff (backoff-sec), and 1 hour for the idle period after which AdminActionRequired auto-clears (disable-auto-clear-hr).

Step 7 (Optional) Use the auto-recon-timeout commands in config mode to set the SRG reconciliation and session synchronization timeouts, in seconds.

Example:

```
configure
   auto-recon-timeout srg <seconds>
   auto-recon-timeout session <seconds>
```

Step 8 Use the show subscriber auto-resync flap-protection upf command to view the flap-protection state of a UPF.

Example:

```
bng# show subscriber auto-resync flap-protection upf upf1
{
  "upf": "upf1",
  "gr-instance": 1,
  "recent-flap-count": 0,
  "flap-window-minutes": 30,
  "protection-status": "Normal",
```

```
  "issue-job-id": "autoresync-upf1-1-1788434082806008812",
  "issue-reason": "aborted: UPF went inactive during autoresync",
  "clear-flap-state-needed": false
}
```

- upf: The UPF being tracked.
- gr-instance: The GR instance associated with the UPF.
- recent-flap-count: The number of active to inactive to active cycles seen within the sliding window. The Active to Initialized to Active sequence of an RPFO is not counted.
- flap-window-minutes: The configured duration for which flap iterations are retained.
- protection-status: Normal, Backoff, or AdminActionRequired.
- issue-job-id: The most recent job associated with an observed flap-related issue, if any.
- issue-reason: The reason retained for that job, for example the UPF became inactive while reconciliation was running.
- clear-flap-state-needed: true when you should clear the flap state with the subscriber auto-resync clear-flap-state upf command; otherwise false.

Monitor reconciliation with alerts

In addition to the show commands, you can configure alerts on the CEE Ops Center to be notified about automatic reconciliation events. The alerts are based on the autoresync metrics. The alert rules are organized in three groups, and you can install each group independently:

- AutoResync (Core): Contains 9 primary fault alerts designed to capture critical operational issues, such as flap protection escalations, component unhealthiness, and job failures.
- AutoResyncExt (Extended): Provides 12 supplemental alerts covering advanced fault scenarios, including trigger queue saturation, configuration reload errors, and systemic phase failure trends.
- AutoResyncNotify (Notifications): Includes 3 informative alerts that provide positive confirmation of job lifecycle stages—such as job initiation, progress, and successful completion—rather than reporting error states.

These rules can be installed independently within the CEE Ops Center, allowing for a tailored alerting strategy based on the specific operational environment.

Table 50: Alerts on the CEE Ops Center

Alert	Group	Severity	For	Description
AutoResyncAdminActionRequired	AutoResync	critical	0m	Repeated UPF flapping requires you to clear the flap state.

Alert	Group	Severity	For	Description
AutoResyncBackoffActive	AutoResync	warning	5m	Automatic reconciliation for a UPF is temporarily throttled because the UPF is flapping.
AutoResyncComponentUnhealthy	AutoResync	major	1m	An internal autoresync component, such as the job store, watcher, configuration reload, flap persistence, or hydration, is unhealthy.
AutoResyncEligibleTriggerSuppressed	AutoResync	major	0m	A valid trigger arrived while automatic reconciliation was disabled.
AutoResyncJobFailed	AutoResync	major	0m	A job ended as failed, aborted, or partial, excluding timeouts and a UPF going inactive.
AutoResyncJobFailedByStepTimeout	AutoResync	major	0m	A configured step timeout terminated a job.
AutoResyncMetricsTargetDown	AutoResync	critical	5m	The autoresync metrics scrape target is unavailable, including a failure of only the autoresync container.
AutoResyncInterruptedJobNeedsAdmin	AutoResync	major	1m	A restart or recovery left a job interrupted, and it is held for outcome verification or retrigger.
AutoResyncRepeatedJobFailures	AutoResync	critical	0m	Three or more non-timeout terminal failures occurred for one UPF and GR instance.

Alert	Group	Severity	For	Description
AutoResyncUpfNotReady	AutoResyncExt	minor	0m	The UPF has no active SRG pair, so every phase fails with upf-not-ready. The UPF must be provisioned.
AutoResyncTriggerEventsDropped	AutoResyncExt	critical	0m	Trigger events were dropped because the trigger queue was full. The affected UPF is not reconciled.
AutoResyncJobDispatchFailed	AutoResyncExt	major	0m	A job was accepted but was never dispatched to a worker.
AutoResyncWatcherUnavailable	AutoResyncExt	critical	0m	The automatic trigger path is unavailable while the pod is ready, so automatic reconciliation does not run.
AutoResyncDisabledForExtendedPeriod	AutoResyncExt	warning	24h	Automatic reconciliation has been disabled for an extended period. The default state is disabled.
AutoResyncJobStuck	AutoResyncExt	major	5m	A job has not progressed from its current state, which indicates a stuck worker.
AutoResyncPersistenceUnavailable	AutoResyncExt	critical	1m	The job store fell back to memory, so jobs do not survive a restart and interrupted-job recovery is not possible.

Alert	Group	Severity	For	Description
AutoResyncConfigReloadFailed	AutoResyncExt	major	1m	A configuration reload failed, and the pod keeps the previous configuration.
AutoResyncReconRetryLoop	AutoResyncExt	warning	0m	A UPF repeatedly requests a reconciliation retry and does not converge.
AutoResyncFlapStateWidespread	AutoResyncExt	critical	5m	Three or more UPFs are flapping at the same time, which indicates a CP-side or network-side fault.
AutoResyncTriggerQueueNearCapacity	AutoResyncExt	warning	2m	The trigger queue is close to its capacity, so events might be dropped.
AutoResyncPhaseFailureConcentration	AutoResyncExt	warning	0m	Failures are concentrated in one reconciliation phase.
AutoResyncJobStartedNotification	AutoResyncNotify	minor	0m	A job was started. The alert shows the trigger reason.
AutoResyncJobInProgress	AutoResyncNotify	minor	0m	A job is running. The alert is raised when the job starts and cleared when it ends.
AutoResyncJobSucceededNotification	AutoResyncNotify	minor	0m	A job completed successfully.

Install alert rules

You can provision alerting groups by applying their respective rule blocks within the CEE Ops Center CLI. These configurations are applied in `config` mode.

To install the `AutoResync` core alert group, use the procedure:

```
config
 alerts rules group AutoResync
  interval-seconds 30
  rule AutoResyncAdminActionRequired
   expression "autoresync_flap_state{state=\"AdminActionRequired\"} == 1"
   duration 0m
   severity critical
```

```
   type "Processing"
   annotation summary
    value "AutoResync requires operator action"
    exit
   annotation description
    value "Repeated flap protection requires explicit operator action."
    exit
  exit
 exit
 commit
end
```

Remove alert rules

To decommission a specific group of rules, use the **no alerts rules group** command in configuration mode.

```
config
 no alerts rules group AutoResyncNotify
commit
end
```

When applying these configurations, ensure that you are using a plain-text editor to prevent the conversion of straight quotes into smart quotes, as curly quotes will result in syntax errors during command execution.

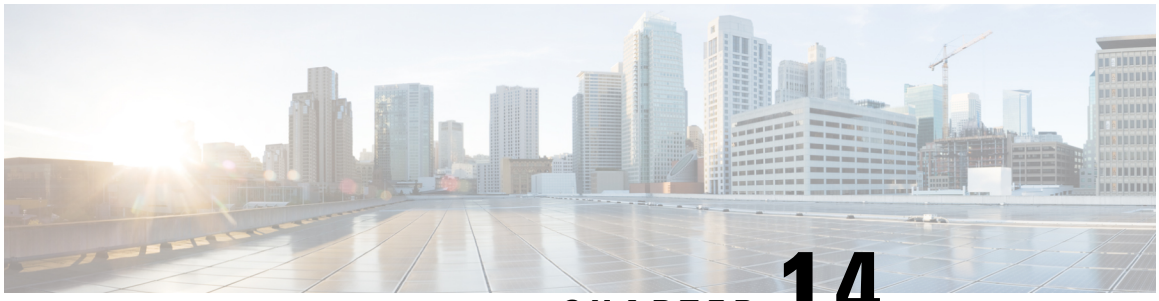
Autoresync metrics

The autoresync container exports these Prometheus metrics, which the alerts use.

Table 51: Prometheus metrics

Metric	Type	Primary labels	Purpose
autoresync_runtime_enabled	Gauge	none	Runtime admissi
autoresync_component_healthy	Gauge	component	Last observed co
autoresync_trigger_events_total	Counter	UPF, GR, trigger reason, event, decision, reason code	Trigger suppressi
autoresync_job_events_total	Counter	UPF, GR, trigger reason, event, result, decision, reason code	Job lifecycle, dis recovery.
autoresync_job_state	Gauge	UPF, GR, trigger reason, state	One-hot current
autoresync_job_state_since_timestamp_seconds	Gauge	UPF, GR, trigger reason	Time of the curr
autoresync_job_duration_seconds_total	Counter	UPF, GR, trigger reason, result	Accumulated ter
autoresync_job_duration_seconds_histogram	Duration-bin counter	job labels, bin	App-infra job-du
autoresync_flap_transitions_total	Counter	UPF, GR, previous/current state, reason	Flap escalation a
autoresync_flap_state	Gauge	UPF, GR, state	One-hot Normal AdminActionRe
autoresync_flap_iterations	Gauge	UPF, GR	Current flap-win
autoresync_step_events_total	Counter	UPF, GR, trigger reason, step, event, result, reason	Step boundaries.

Metric	Type	Primary labels	Purpose
autoresync_step_duration_seconds_total	Counter	step lifecycle labels	Accumulated
autoresync_step_duration_seconds_histogram	Duration-bin counter	step lifecycle labels, bin	App-infra ste
autoresync_correction_events_total	Counter	UPF, GR, trigger reason, step, correction, event, result, reason	Correction bo
autoresync_correction_duration_seconds_total	Counter	correction lifecycle labels	Accumulated
autoresync_correction_duration_seconds_histogram	Duration-bin counter	correction lifecycle labels, bin	App-infra co
autoresync_trigger_queue_depth	Gauge	none	Current trigg
autoresync_trigger_queue_capacity	Gauge	none	Trigger-chan



CHAPTER 14

IP Address Management

- Feature Summary and Revision History, on page 285
- Feature Description, on page 286
- Configuring IPAM Feature, on page 294
- IPAM Enhancements, on page 303
- RADIUS-based address assignment and route-tagging, on page 308
- Reserved IPv4 address controls for PPPoE pools, on page 314
- Periodic reconciliation, on page 316
- Periodic reconciliations for ID Manager, on page 318
- IPAM Route Programming Enhancements, on page 320
- Pre-Allocation of Gateway IP and Address Chunks, on page 321
- IANA and IAPD Allocation from Same IP Range, on page 324
- Static IP Support, on page 327
- Support for circuit-id as ipam-dp-key in static IPAM pools, on page 331
- Static IP Mapping Database, on page 333
- IP, ID, and Chunk event logging , on page 339

Feature Summary and Revision History

Summary Data

Table 52: Summary Data

Applicable Product(s) or Functional Area	cnBNG
Applicable Platform(s)	SMI
Feature Default Setting	Disabled - Configuration Required
Related Changes in this Release	Not Applicable
Related Documentation	Not Applicable

Revision History

Table 53: Revision History

Revision Details	Release
Enhancement to the ipam-address-chunk action command for IPv6. The pool-group-tag keyword is no longer supported.	2026.04.0
Introduced periodic reconciliation support for IPAM.	2025.03.0
Introduced offline address range support to the IANA and IAPD Allocation from Same IP Range feature.	2025.03.0
Introduced support for allocating IANA and IAPD from same IP pool.	2025.01.0
Introduced support for IPAM Asynchronous route programming.	2024.04.0
Introduced support for pre-allocation of Gateway IP and address chunks.	2024.04.0
Introduced enhancements in IPAM chunk allocation with NM pair for SRG.	2024.02.0
Introduced support for dynamic chunk allocation in a CP-GR setup.	2024.02.0
Introduced support for Variable Chunk Size for an IPAM Data Plane.	2024.01.0
Introduced support for the Static IP Mapping Database.	2022.02.0
Enhancement Introduced: The IPAM feature is NSO-integrated.	2021.04.0
The following sections were added: • Configuration Recommendations • IPAM Enhancements • Static IP Pool Enhancements	2021.03.0
First introduced.	2021.01.0

Feature Description

Note This feature is Network Services Orchestrator (NSO) integrated.

IP Address Management (IPAM) is a method of tracking and managing IP addresses of a network. IPAM is one of the core components of the subscriber management system. Traditional IPAM functionalities are insufficient in Cloud-Native network deployments. Hence, IPAM requires additional functionalities to work

with the Cloud-Native subscriber management system. The Cloud-Native IPAM system is used in various network functions, such as Session Management function (SMF), Policy Charging function (PCF), and Broadband Network Gateway (BNG).

The IPAM system includes the following functionalities to serve the Cloud Native and Control and User Plane Separation (CUPS) architecture:

- **Centralized IP Resource Management**—Based on the needs of the Internet Service Provider (ISP), the Control Plane (CP) is deployed either on a single (centralized) cluster or multiple (distributed) clusters. For multiple cluster deployments, the IPAM automatically manages the single IP address space across the multiple CPs that are deployed in the distributed environment.
- **IP Address-Range Reservation per User Plane**—For subscribers connecting to the Internet core, the User Plane (UP) provides the physical connectivity. The UP uses the summary-routes to advertise subscriber routes to the Internet core. For CPs that are managing multiple UPs, the CP reserves a converged IP subnet to the UPs. In such a scenario, the IPAM splits the available address space into smaller address-ranges and assigns it to different UPs.
- **IP Address Assignment from Pre-Reserved Address-Ranges**—When subscribers request for an IP address, the IPAM assigns addresses from the pre-reserved address range of their respective UP.

IPAM Components

This section describes the different components of the IPAM system.

IPAM Sub-Modules

The IPAM functionalities are categorized in the following sub-modules:

IPAM Server

This module manages the complete list of pools and address-space configurations. It splits the configured address-ranges into smaller address-ranges (statically or dynamically) to distribute it to the IPAM Cache modules. The IPAM server can be deployed as a centralized entity to serve a group of CN clusters or as an integrated entity within a single cluster.

IPAM Cache

This module acquires the free address-ranges from the IPAM server and allocates individual IP addresses to the IPAM clients. The IPAM cache is generally deployed in the Distributed mode running within each cluster, to communicate with the co-located or remotely located IPAM server. It is also responsible for address-range reservation per UP and pool threshold monitoring. The IPAM server and cache modules can also run in an integrated mode.

When the **disable-dp-threshold** CLI is configured, threshold calculation is skipped and action commands become the only mechanism for chunk allocation and release.

IPAM Client

This module is tightly coupled with its respective network-function, responsible for handling request and release of individual IP address from the IPAM cache for each IP managed end-device.

Unlike the IPAM server and cache module, the IPAM client caters to use-cases specific to network-functions such as BNG, SMF, PCF, and so on.

IPAM Integration in cnBNG

The Cloud-Native Broadband Network Gateway (cnBNG) function comprises of loosely coupled microservices that provide the functionality of the BNG. The decomposition of these microservices is based on the following three-layered architecture:

1. Layer 1: Protocol and Load Balancer Services (Stateless)
2. Layer 2: Application services (Stateless)
3. Layer 3: Database Services (Stateful)

The IPAM and cnBNG integration occurs in the Application Services layer.

BNG Node Manager Application—The BNG Node Manager application is responsible for the User Plane function (UPF) management, ID and resource management, and IP address management. Therefore, the IPAM Cache is integrated as part of this microservice.

Also, the UPF uses the IPAM Client module for address-range-reservation per UPF.

BNG DHCP and PPPOE Application—The BNG-DHCP and BNG-PPOE pods are responsible for providing IP addresses to the BNG subscriber session. During session bring-up, the IP address is requested and during session bring-down, the IP address is released back. These First Sign of Life (FSOL) applications send the inter-process communications (IPC) to the Resource Manager (RMGR) component in the NodeMgr. The NodeMgr receives the IPC and invokes the IPAM component.

IPAM Server Application—Based on the deployment model, the IPAM Server runs as an independent microservice as part of the same cluster or in a remote cluster.

In standalone deployments, the IPAM Server functionality is an integral part of the IPAM Cache, that is, it runs as part of the Node Manager microservice itself.

Configuration Recommendations

This section provides the following configuration recommendations.

Pool-Size Configuration

While configuring the pool-size, the recommendation is to keep additional buffer of IPs for undisturbed subscriber churn. Consider the following recommendations:

- Released IPs become part of the quarantine queue depending on the quarantine configuration. These IPs cannot be reused until they are free.
- Due to the threshold logic, the CP automatically reserves additional chunks to the User Plane function (UPF) when required. It is based on a first-come-first-server basis.

  In a multiple UPF sharing a pool use case, whichever UPF hits the threshold first is given a chunk. Therefore, free chunks may not be available for other UPFs. That is, 100% pool utilization may not occur for a given pool in certain conditions. In this scenario, configure additional IPs accordingly or split into multiple pools per UPF.

Split-Size Configuration

While configuring the split-size of a dynamic pool, consider the following recommendations:

- Number of IPs in pool
- Total number of subscribers. That is, the number of IPs versus the percentage of the number of sessions.
- Number of UPFs sharing the pool
- Number of sessions per UPF. That is, the number of sessions versus the percentage of the number of UPFs.
- Quarantine time of the pool
- Churn rate. That is CPS of IP allocation, CPS of IP release.
- Max routes a UPF can support (example: ASR9k supports 32 routes per loopback in Release 7.4.x/7.5.x)

Each Node Manager (nodemgr) uses per-cache as the split-size of the chunk.

Each UPF is assigned per-dp as the split-size of a chunk.

Each Node Manager reserves chunk of per-cache size and further splits (if-applicable) and assigns them to a UPF.

A chunk can be in one of the following states:

- <upf-name> (means, this chunk is assigned to an UPF)
- Free:CP (means, it is free in the cache-pod, any Node Manager can use it)
- Free:NM0 (means, it is free in the Node Manager instance-0. Only this Node Manager can use it)
- Free:NM1 (means, it is free in the Node Manager instance-1. Only this Node Manager can use it)
- QT:NM0 (means, it is in quarantine in the Node Manage instance-0. This is moved to free after quarantine timer expiry)
- QT:NM1 (means, it is in quarantine the Node Manage instance-1. This is moved to free after quarantine timer expiry)

Threshold Configuration

Changing threshold configuration on-the-fly takes effect only after the next event, such as address-allocation or address-delete. On an idle-system, there may not be any immediate action.

Variable Chunk Size Support for an IPAM Data Plane

You can now configure multiple pools of IP addresses, each with different chunk sizes for a given IPAM Data Plane (SRG group in SRG deployment, or an UPF in non-SRG deployment). This configuration would require an administrator to divide the available IP ranges into different pools, which might pose challenges in effectively using the IP addresses. To overcome this issue, you can initially configure a reasonable number of IP addresses in the pools as per the deployment plan. By monitoring the IP chunk usage per pool, you can then provide additional IP ranges as required. You can also use a metric to monitor the chunk usage for each pool.

To associate multiple pools to a DHCP profile, you can now attach a pool-tag, which in turn can be associated with one or more pools. In tag-based IP allocation for a given DP, the first chunk is allocated from the highest priority pool (for example, priority 0) for the specified tag, and the next chunk is allocated from the second highest priority pool (priority 1). If one chunk gets allocated from both the pools that are associated to a tag, any subsequent chunk requirement utilizes the least priority pool.

Starting with release 2026.03.0, for IPv4, tagged pool behavior is enhanced with action-command-based chunk management. Support for **pool-group-tag** keyword in IPv4 action commands is removed, and administrators must explicitly select the target pool by using the pool name. This provides direct control over whether a new chunk is allocated from the primary pool or the secondary pool. When you execute the **ipam-address-chunk allocate** action command, the system allocates exactly one new chunk from the specified pool.

For configuration details, see Configuring Variable Chunk Size Support for an IPAM Data Plane section.

Limitations for Variable Chunk Size Support for an IPAM Data Plane

- This feature is supported only for IPoE subscribers.
- Presently, only two chunk sizes (2 pools) are supported for a given Data Plane.
- Configuring same priority for different pools that are associated with the same pool tag is not supported.
- In a DHCP profile, you can either configure pool-name or pool-group-tag. Configuring both together is not supported.
- You cannot change the Pool priority dynamically. Priority must be specified when the pool is configured. The default value is 0.
- You cannot change the Tag configuration under a pool dynamically. Tag information must be configured when the pool is configured.

How it Works

This section describes the call flow pertaining to the integration of the IPAM in the cnBNG.

Call Flows

This section describes the following IPAM call flows in cnBNG:

- IPAM initial sequence call flow
- IPAM call flow
- IPAM static-pool call flow

IPAM Initial Sequence Call Flow

This section describes the cnBNG initial sequence call-flow.

Figure 16: IPAM Initial Sequence Call Flow

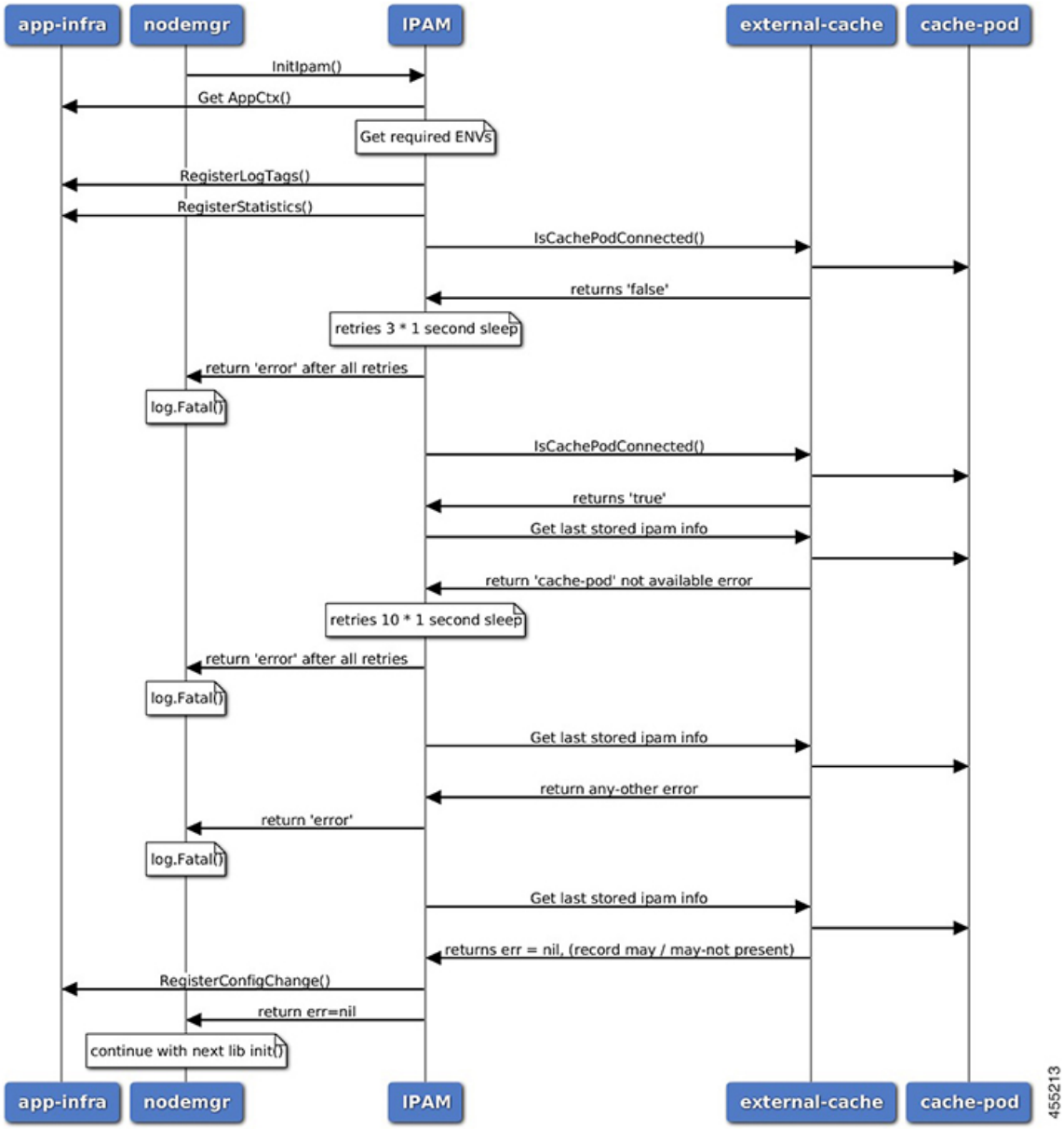


Table 54: IPAM Initial Sequence Call Flow Description

Step	Description
1	IPAM reads the required environments, registers with the application infrastructure for log-tags, metrics, and database connection.
2	IPAM restores the previous state from the cache-pod, if present.
3	IPAM registers for configuration change and applies the new configuration change, if any. -change, apply new config-changes if any

IPAM Call Flow

This section describes the cnBNG IPAM call-flow.

Figure 17: IPAM Call Flow

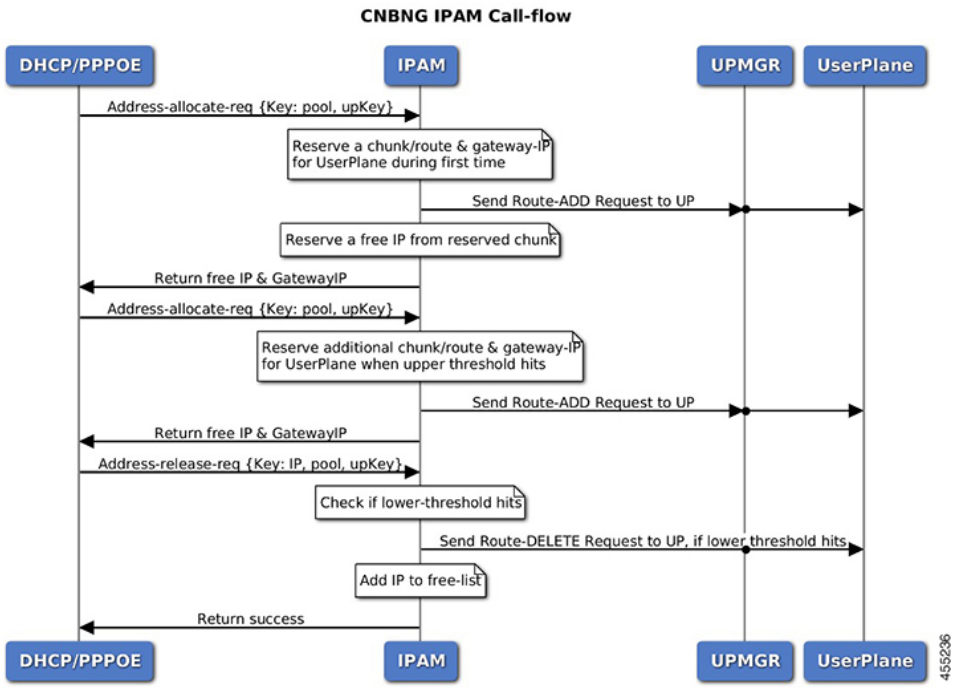


Table 55: IPAM Call Flow Description

Step	Description
1	IPAM receives the ‘addr-alloc’ request from the DHCP or PPPoE pod with pool-name, addr-type and user plane function (UPF) as input.
2	IPAM reserves a new address-range (if not already present for UPF) and sends a ROUTE-ADD message to the UPF. It waits for a success or failure response. If the receives a failure response, it removes the chunk and repeats this step.
3	IPAM reserves a free-IP from the assigned address-range and returns to the DHCP or PPPoE.
4	IPAM monitors the ‘upper-threshold’ for each UPF during each IP address-allocation and also has a background thread that monitors. It then assigns new address-ranges to the UPF and repeats the ROUTE-ADD flow.
5	IPAM receives the ‘addr-free’ request from the DHCP or PPPoE pod with pool-name, addr-type, addr or pfx, and UPF as input.
6	IPAM moves the addr or pfx first to the quarantine-list until the quarantine timer and later moves it to the free-list.

Step	Description
7	IPAM monitors the ‘lower-threshold’ (currently 0%) of the address-range of each UPF, removes the address-range from the UPF, and sends the ROUTE-DELETE message.

IPAM Static-Pool Call Flow

This section describes the IPAM static-pool call flow.

Figure 18: IPAM Static Pool Call Flow

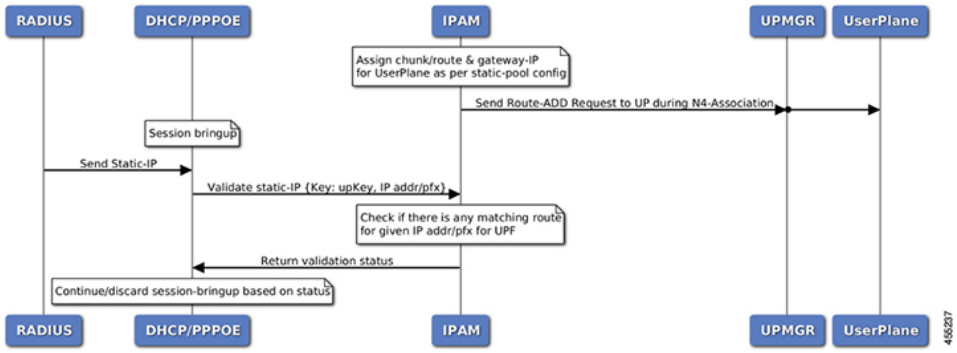


Table 56: IPAM Call Flow Description

Step	Description
1	IPAM receives the ‘addr-alloc’ request from the DHCP or PPPoE pod with pool-name, addr-type and user plane function (UPF) as input.
2	IPAM reserves a new address-range (if not already present for UPF) and sends a ROUTE-ADD message to the UPF. It waits for a success or failure response. If the receives a failure response, it removes the chunk and repeats this step.
3	IPAM reserves a free-IP from the assigned address-range and returns to the DHCP or PPPoE.
4	IPAM monitors the ‘upper-threshold’ for each UPF during each IP address-allocation and also has a background thread that monitors. It then assigns new address-ranges to the UPF and repeats the ROUTE-ADD flow.
5	IPAM receives the ‘addr-free’ request from the DHCP or PPPoE pod with pool-name, addr-type, addr or pfx, and UPF as input.
6	IPAM moves the addr or pfx first to the quarantine-list until the quarantine timer and later moves it to the free-list.
7	IPAM monitors the ‘lower-threshold’ (currently 0%) of the address-range of each UPF, removes the address-range from the UPF, and sends the ROUTE-DELETE message.

Limitations

The IPAM feature has the following limitations:

- Duplicate IP address is not supported within a pool.
- Duplicate IP address is not supported across pools, that belong to same VRF.
- Removal of ‘pool’ is not supported while addresses are already assigned.
- Removal or modification of IP-address-ranges is not supported while addresses are already assigned.
- Change of ‘source’ field is not supported while address or prefixes are already assigned.
- Change of ‘vrf-name’ of pool is not supported while address or prefixes are already assigned.
- Start-address should be less than the End-address.
- Configuring addr-range split-size in wrong manner, that is, size of address-range < size-of-per-cache < size-of-dp, is not supported.
- Configuring IPv6 Address (IANA) and Prefix (IAPD) values interchangeably is not supported.
- Configuring invalid ‘prefix-length’ for Prefix (IAPD) range is not supported.

Configuring IPAM Feature

This section describes how to configure the IPAM feature.

Configuring the IPAM feature involves the following steps:

1. Configuring IPAM source
2. Configuring the global threshold
3. Configure IPAM address pool
4. Configuring IPv4 address ranges
5. Configuring IPv6 address ranges
6. Configuring IPv6 prefix ranges
7. Configuring the IPv4 threshold
8. Configuring the IPv6 threshold
9. Configuring IPv4 address range split
10. Configuring IPv6 address and prefix address-range split

Configuring IPAM Source

Use the following configuration to configure the IPAM source.

```
config
  ipam
```

```
    source local
     threshold { ipv4-add percentage | ipv6-address percentage | ipv6-prefix
percentage }
      commit
```



Note Starting from Release 2026.03.0, when the **disable-dp-threshold** command is configured, the system stops monitoring both the lower and upper thresholds. This disables the automated threshold-monitoring logic and provides you more flexibility in resource management.

NOTES:

- **ipam**: Enters the IPAM Configuration mode.
- **source local**: Enters the local datastore as the pool source.
- **threshold { ipv4-add** *percentage* **| ipv4-address** *percentage* **| ipv6-prefix** *percentage* **}**: Specifies the threshold in percentage for the following:
  - **ipv4-add** *percentage*: Specifies the IPv4 threshold. The valid values range from 1 to 100. The default value is 80.
  - **ipv6-add** *percentage*: Specifies the IPv4 threshold. The valid values range from 1 to 100. The default value is 80.
  - **ipv6-prefix** *percentage*: Specifies the IPv6 threshold prefix. The valid values range from 1 to 100. The default value is 80.

Configuring Global Threshold

Use the following configuration to configure the global threshold.

```
config
  ipam
    threshold
      ipv4-addr percentage
      ipv6-addr percentage
      ipv6-prefix percentage
      commit
```

NOTES:

- **ipam**: Enters the IPAM Configuration mode.
- **threshold**: Enters the threshold sub-mode.
- **ipv4-add** *percentage*: Specifies the IPv4 threshold. The valid values range from 1 to 100. The default value is 80.
- **ipv6-add** *percentage*: Specifies the IPv4 threshold. The valid values range from 1 to 100. The default value is 80.
- **ipv6-prefix** *percentage*: Specifies the IPv6 threshold prefix. The valid values range from 1 to 100. The default value is 80.

Disable DP threshold monitoring

Use the following configuration to disable DP threshold monitoring.

Procedure

Use the **disable-dp-threshold** command to disable DP threshold monitoring.

Example:

```
bng# config
bng(config)# ipam instance id
bng(config-instance-1)# disable-dp-threshold
```

When the **disable-dp-threshold** CLI is configured, the system completely bypasses the automated threshold-based monitoring logic.

In this mode, chunks are allocated and released only through action commands. Administrators must manage the resources manually.

Configuring IPAM Address Pool

Use the following configuration to configure the IPAM address pool.

```
config
   ipam
     address-pool pool_name [ address-quarantine-timer ] [offline ] [ static
 user_plane_name ] [ vrf-name string ]
      commit
```

NOTES:

- **ipam**: Enters the IPAM configuration mode.
- **address-pool** *pool_name* **[address-quarantine-timer] [offline] [static** *user_plane_name* **] [vrf-name** *string* **]**: Configures the address pool configuration. *pool_name* must be the name of the address pool.

  This command configures the following parameters:

  - **address-quarantine-timer**: Specifies the address quarantine time in seconds. The valid values range from 4 to 3600. The default value is 4.
  - **offline**: Sets the address pool to offline mode.
  - **static** *user_plane_name*: Specifies the 'user-plane' name associated to this static-pool.
  - **vrf-name** *string*: Configures the Virtual routing and forwarding (VRF) name of the pool.

Configuring IPAM Quarantine Timer

This section describes how to configure the IPAM quarantine timer.

```
config
   ipam
      address-pool pool_name
         address-quarantine-timer quarantine_timer_value
         vrf-name vrf_name_value
         ip4
            address-range start_ipv4_address end_ipv4_address
            address-range start_ipv4_address end_ipv4_address
         !
       !
     !
```

NOTES:

- **ipam**—Enter the IPAM configuration.
- **address-pool** *pool_name*—Specifies the name of the pool to enter the pool configuration. *pool_name* must be the name of the address pool.
- **address-quarantine-timer** *quarantine_timer_value*—Specifies the value of the quarantine timer in seconds. *quarantine_timer_value* must be in the range of 4-3600 seconds. The default value is 4.
- **vrf-name** *vrf_name_value*—Specifies the name of the VPN routing and forwarding (VRF) for the pool.
- **ipv4**—Enters the IPv4 mode.
- **address-range** *start_ipv4_address end_ipv4_address*—Specifies the IP addresses for start and end IPv4 address-range.

Configuring IP Address Reservation

Use the following configuration to reserve the minimum number of addresses for each User Plane function (UPF), Node Manager, and pool.

pool.

```
config
   ipam
      instance instance_id
         min-dp-addr-size { ipv4-addr | ipv6-addr | ipv6-prefix }
         commit
```

NOTES:

- **ipam**: Enters the IPAM configuration mode.
- **instance** *instance_id*: Specifies the IPAM instance and enters the instance sub-mode. *instance_id* must be an integer. The valid value ranges from 1 to 8.
- **min-dp-addr-size { ipv4-addr | ipv6-addr | ipv6-prefix }**: Specifies the minimum number of addresses to reserve for each UPF, Node Manager, and pool.
  - **ipv4-addr**: Specifies the minimum number of IPv4 addresses to reserve.
  - **ipv6-addr**: Specifies the minimum number of IPv6 addresses to reserve.
  - **ipv6-prefix**: Specifies the minimum number of IPv6 prefixes to reserve.

Configuring IPv4 Address Ranges

Use the following configuration to configure the IPv4 address ranges.

```
config
   ipam
      address-pool pool_name
         ipv4
            address-range start_ipv4_address end_ipv4_address [ default-gateway
ipv4_address ] [ offline ]
            commit
```

NOTES:

- **ipam**: Enters the IPAM configuration mode.
- **address-pool** *pool_name*: Configures the address pool configuration. *pool_name* must be the name of the address pool.
- **ipv4**: Enters the IPv4 mode of the pool.
- **address-range** *start_ipv4_address end_ipv4_address* **[default-gateway** *ipv4_address* **] [offline]**: Configures the IPv4 address range with the starting and ending IPv4 address.
  - **default-gateway** *ipv4_address*: Specifies the IPv4 address of the default gateway.
  - **offline**: Sets the address pool to offline mode.

Configuring IPv6 Address Ranges

Use the following configuration to configure the IPv6 address ranges:

```
config
   ipam
      address-pool pool_name
         ipv6
            address-range start_ipv6_address end_ipv6_address [ offline ]
            commit
```

NOTES:

- **ipam**: Enters the IPAM configuration mode.
- **address-pool** *pool_name*: Configures the address pool configuration. *pool_name* must be the name of the address pool.
- **ipv6**: Enters the IPv6 mode of the pool.
- **address-range** *start_ipv6_address end_ipv6_address* **[offline]**: Configures the IPv6 address range with the starting and ending IPv6 address.

  [offline]: Sets the address pool to offline mode.

Configuring IPv6 Prefix Ranges

Use the following configuration to configure the IPv6 prefix ranges:

```
config
   ipam
      address-pool pool_name
         ipv6
            prefix-ranges
               prefix-range prefix_value prefix-length prefix_length
               commit
```

NOTES:

- **ipam**: Enters the IPAM configuration mode.
- **address-pool***pool_name*: Configures the address pool configuration. *pool_name* must be the name of the address pool.
- **ipv6**: Enters the IPv6 mode of the pool.
- **prefix-ranges**: Enters the prefix ranges mode.
- **prefix-range** *prefix_value* **prefix-length** *length*: Configures the IPv6 prefix range. *prefix_value* specifies the IPv6 prefix range.

  prefix-length *length* specifies the IPv6 prefix length.

Configuring IPv4 Threshold

Use the following configuration to configure the IPv4 threshold:

```
config
   ipam
      address-pool  pool_name
         ipv4
            threshold
               upper-threshold  percentage
               commit
```

NOTES:

- **ipam**: Enters the IPAM Configuration mode.
- **address-pool** *pool_name*: Configures the address pool configuration. *pool_name* must be the name of the address pool.
- **ipv4**: Enters the IPv4 mode of the pool.
- **threshold**: Enters the threshold sub-mode.
- **upper-threshold** *percentage*: Specifies the IPv4 upper threshold value in percentage. The valid values range from 1 to 100. The default value is 80.

The following is a sample configuration:

```
config
   ipam
      address-pool p1
         ipv4
            threshold
               upper-threshold 80
```

Configuring IPv6 Prefix-Range Threshold

Use the following configuration to configure the IPv6 prefix-range threshold.

```
config
   ipam
      address-pool pool_name
         ipv6
            prefix-ranges
               threshold
                  upper-threshold percentage
                  commit
```

NOTES:

- **ipam**: Enters the IPAM configuration mode.
- **address-pool** *pool_name*: Configures the address pool configuration. *pool_name* must be the name of the address pool.
- **ipv6**: Enters the IPv6 mode of the pool.
- **prefix-ranges**: Enters the IPv6 prefix ranges sub-mode.
- **threshold**: Enters the threshold sub-mode.
- **upper-threshold** *percentage*: Specifies the IPv6 upper-threshold value in percentage.

The following is an example configuration:

```
config
   ipam
      address-pool p3
         ipv6
            prefix-ranges
               threshold
                  upper-threshold 78
```

Configuring IPv4 Address Range Split

Use the following configuration to configure the IPv4 address range split.

```
config
   ipam
      address-pool pool_name
         ipv4
            [ no ] split-size { per-cache value | per-dp value }
            commit
```

NOTES:

- **ipam**: Enters the IPAM configuration mode.
- **-address-pool** *pool_name*: Configures the address pool configuration. *pool_name* must be the name of the address pool.
- **ipv4**: Enters the IPv4 mode of the pool.
- **[no] split-size { per-cache** *value* **| per-dp** *value* **}**: Specifies the size of the IPv4 range to be split for each IPAM cache allocation. The IPAM server consumes this configuration. The **no** form of this command disables the splitting of the address-ranges into smaller chunks.

  per-cache *value*: Specifies the size of the IPv4 range to be split for each Data-Plane (User-Plane) allocation. The valid values range from 2 to 262144. The default value is 1024.

  The IPAM cache consumes this configuration.
- **per-dp** *value*: Specifies the size of the IPv4 range to be split for each Data-Plane (User-Plane) allocation. The valid values range from 2 to 262144 The default value is 256.

  The IPAM cache consumes this configuration.

Configuring IPv6 Address and Prefix Address-Range-Spilt

Use the following configuration to configure the IPv6 address and prefix address range spilt.

```
config
   ipam
      address-pool pool_name
         ipv6
            address-ranges
               [ no ] spilt-size { per-cache value | per-dp value }
               commit
            prefix-ranges
               [ no ] spilt-size { per-cache value | per-dp value }
               commit
```

NOTES:

- **ipam**: Enters the IPAM configuration mode.
- **address-pool** *pool_name*: Configures the address pool. *pool_name* must be the name of the address pool.
- **ipv6**: Enters the IPv6 mode of the pool.
- **[no] spilt-size { per-cache** *value* **| per-dp** *value* **}**: Specifies the size of the IPv6 range to be split for each IPAM cache allocation. The IPAM server consumes this configuration. The **no** form of this command disables the splitting of the address-ranges into smaller chunks.

  per-cache *value*: Specifies the size of the IPv6 range to be spilt for each Data-Plane (User-Plane) allocation. The valid values range from 2 to 262144. The default value is 1024.

  The IPAM cache consumes this configuration.
- **per-dp** *value*: Specifies the size of the IPv6 range to be spilt for each Data-Plane (User-Plane) allocation. The valid values range from 2 to 262144 The default value is 256.

  The IPAM cache consumes this configuration.

Configuring Variable Chunk Size Support for an IPAM Data Plane

Use the following commands to configure the IPAM tags:

```
ipam
 instance instance_id
  source local
  address-pool pool_name
   tags
    group tag_value
   exit
   pool-priority
    priority <0>
   exit
```

NOTES:

- **ipam**: Enters the IPAM configuration.
- **instance** *instance_id*: Specifies the IPAM instance and enters the instance sub-mode. *instance_id* must be an integer. The valid value ranges from 1 to 8.
- **source local**: Enters the local datastore as the pool source.
- **address-pool** *pool_name*: Specifies the name of the pool to enter the pool configuration. *pool_name* must be the name of the address pool
- **tags group** *tag_value*: Specifies the tag group value of the pool. All pools carrying the same **tag group** value can be associated to a DHCP profile using the keyword **pool-group-tag** and the corresponding **tag group** value. The value must be a string.
- **pool-priority priority** *value*: Specifies the order of IP chunk and IP allocation among pools with the same tag value. *value* must be the integer **0** or **1**. **0** is the default value, and has the highest priority..

Use the following commands to configure DHCP for pool association:

```
profile dhcp dhcp_profile_name
 ipv4
  server
   pool-group-tag tag_value
   lease hours hours_value
  exit
 exit
exit
```

NOTES:

- **profile dhcp** *dhcp_profile_name*: Specifies the DHCP profile name.
- **ipv4**: Enters IPv4 configuration mode.
- **server**: Enters server configuration mode.
- **pool-group-tag** *tag_value*: Specifies the group tag value that is used to associate the profile with the group tag defined in the pool.

IPAM Enhancements

This section lists the following IPAM enhancements.

IPAM Quarantine Timer

The IP quarantine logic enhancements are as follows:

- The maximum quarantine configuration is increased to 1 hour (Range: 4 to 3600 seconds).
- If the configured quarantine time is <= 15min, additional buffer of 60 seconds is added to the configured quarantine time.
- If the configured quarantine time is > 15min, additional buffer of 5 minutes is added to the configured quarantine time.
- Default quarantine time processing thread interval is changed from 5 to 60 seconds.
- The IP is moved to the free-list after ~(configured-qTime + buffer + delay-from-qt-thread-processing).
- Upon Node Manager pod restart, quarantine time of all older IPs in the quarantine time-queue is reset and will restart from beginning.
- After Node Manager pod restart, all IPs released as part of reconciliation are moved to the quarantine-queue before moving to the free-bitmap (this includes pre-reserved IPs).

Address-Range Level Quarantine

If an address-range is removed from the UPF after releasing all the IPs in a proper manner (that is, each released IP went through quarantine time) then the address-range is moved directly to free-list.

If an address-range is removed from the UPF due to the UPF-release with some of the addresses allocated, then the complete address-range is put under quarantine for the configured time and then moved to free-list.

The **show ipam pool** command displays quarantine-chunks with a special ‘alloc-context’.

Pool and UPF Threshold Monitoring

The UPF threshold monitoring enhancements are as follows:

- **Upper threshold:** Default = 80%, configurable. This is used to add new chunks to the pool or UPF.
- **SafeCutOff:** Default = (upper-threshold-5%), not-configurable. After hitting upper-threshold, new chunks are allocated to the pool or UPF to bring down the current-utilization to safecutoff level , that is, upper-threshold – 5%.
- **Lower threshold:** Default = 90% of upper-threshold, not-configurable. This is used to remove a chunk from the pool or UPF.

Each Node Manager runs a pool level threshold monitoring. When a chunk is assigned to the UPF, the Node Manager checks the pool-threshold hit and reserves additional chunks from the cache-pod for future use.

For pool threshold calculation, the total number of IPs left in free-chunks are considered; not the actual number of allocated IPs on an assigned chunk. That is, after a chunk is assigned to the UPF, it is considered as fully used for pool-threshold monitoring purpose. A complete free address-range can be released back to the cache-pod based on lower-threshold calculation.

For UPF threshold monitoring, the actual number of total IPs and allocated IPs are considered; more chunks are reserved for the UPF when the upper-threshold hits. The Node Manager adds the route to the UPF whenever a new chunk is assigned to it due to the threshold hit. For performance reasons, the route is not deleted if it was added in at the last minute.

The upper threshold is configurable (default=80%), when this threshold hits, new chunks are added until the current-utilization falls back to the safe-cutoff level. That is, 75% is safe cutoff if the upper-threshold is 80%.

Lower threshold is 90% of the upper-threshold. Thati is, if the upper-threshold is 80%, then the lower-threshold is 72%, a chunk can be removed from the UPF only when the remaining threshold is below 72%. Otherwise, the chunk remains in the UPF assigned list. This logic is applied to avoid frequent route-add and route-delete operations around boundary condition. The UPF threshold monitoring is triggered during events such as address-allocate, address-release, and config-change. On idle-system, the behavior may differ, however, in a running system, the threshold calculation occurs regularly.

Marking a pool or address-range as offline overrides the lower-threshold logic. That is, if an offline chunk is completely free, it is removed from the UPF irrespective of the lower-threshold calculation.



Note When the **disable-dp-threshold** CLI is configured, the automated threshold-based monitoring logic is bypassed entirely. In this mode, action commands become the exclusive mechanism for chunk allocation and release, and chunk management is fully controlled by the administrator.

Multiple Replica Handling

IPAM is part of the Node Manager (nodemgr) pod. A maximum of two nodemgr pods are supported per BNG cluster.

During UPF-registration, one of the nodemgr pod gets all the static-pool-routes for the UPF and all the dynamic-pool-routes from both the nodemgr pod if anything is allocated earlier and programs it.

During IP-allocation, the IPC request goes to one of the nodemgr pods. If no routes were assigned earlier, a new route is assigned and if successful, an IP is returned to FSOL. Even if one nodemgr pod goes down, the other nodemgr can handle IP-allocations, provided enough IPs are available. Chunks that are reserved by one nodemgr cannot be used by the other nodemgr for address-allocations.

During IP-release, the IPC request should go to the IP owner nodemgr as best-effort. If the IPC fails due, then the IP become stale on the IPAM. During nodemgr bring-up, the CDL reconciliation occcurs, which recovers the stale IPs. In addition, a new CLI is added **reconcile-ipam** to manually trigger IPAM-CDL reconciliation on a need basis. This command should be executed only during maintenance because it is a heavy operation.

During the UPF release, the N4 release comes to one of the nodemgrs. It sends an internal-IPC to the other nodemgr and both clean-up all the routes assigned to the respective UPF. If one of the nodemgr is down during that time, the other nodemgr takes over the content and releases the chunks on behalf of its peer.

IPAM Chunk-Optimization with Node Manager Pair for SRG

Table 57: Feature History

Feature Name	Release Information	Description
IPAM Chunk-Optimization with Node Manager Pair for SRG	2024.02.0	You can now optimize the allocation of chunks and hence avoid underutilization or wastage of chunks. With this feature, a Subscriber Redundancy Group is persistently associated with a designated Node Manager (NM), and chunks are allocated only to the associated NM. Earlier, subscribers across different groups were load-balanced between both NMs, and chunks were allocated for both NMs. This sometimes resulted in underutilization or wastage of chunks.

In large-scale clusters, chunks are distributed across groups and Node Managers (NMs) in a balanced manner. However, this can lead to inefficiencies, including underutilization and wastage of chunks.

With this feature, each group is persistently associated with a single NM. Chunks are then allocated only to the NM that is sticky to the associated group. This allocation strategy prevents the dispersion of chunks across NMs for a given SRG group, thereby addressing the issue of chunk underutilization and wastage.

When group configurations change, such as addition or deletion of groups, SRG updates the distribution of new groups across the NMs, while maintaining existing associations. If a group is deleted, its stickiness association is removed from the corresponding NM. However, existing mappings for other groups remain intact. When new groups are added, they are balanced among the available NMs, but existing group-NM associations remain unchanged.

Node Manager Responsibility and Failover Handling

The owner NM (pod) is the primary entity responsible for managing all Data Plane (DP) and IP-related activities for its associated groups. These critical functions include:

- Data Plane registration and deregistration
- IP address allocation and release
- Threshold monitoring
- Reconciliation processes
- Monitoring of quarantined chunks

When the active NM instance becomes unavailable, the peer NM takes over and handles requests on behalf of the owner NM instance.



Note The peer NM does not accept the very first subscriber for a new group/pool/AFI combination. Such requests are rejected, as initial subscriber handling is reserved for the owner NM.

To prevent data corruption and avoid rapid toggling of NM instance roles, a buffer period has been implemented before it becomes active on the peer NM. The peer NM resumes active responsibilities only if the owner NM

has been down for more than 3 minutes. Upon becoming active for a remote NM that is down, the peer NM loads the IPAM state to manage any pending requests for the affected NM groups.

Limitations

Following are the known limitations of IPAM Chunk-Optimization with Node Manager Pair for SRG feature:

- The peer NM becomes active for other NM groups if the owner NM has been down for more than three minutes. This delay helps prevent data corruption during consecutive NM reloads.
- Only address allocation and new chunk reservation are supported.
- Reconciliation is not performed on the peer NM on behalf of the active NM if the active NM goes down. This may result in the IPAM system showing a higher number of IP addresses in use compared to the actual subscriber count. Discrepancies are resolved once the owner NM is back online and the IPAM performs reconciliation, syncing the IPAM counter with the session count.
- The peer NM cannot support the allocation of the very first IP address for a new subscriber under a specific combination of group, pool, and AFI. First-time IP allocations must be performed by the owner NM.
- Configuration changes to IPAM are not supported if another NM is active on the peer NM. Accidental configuration changes made during this period will be processed only when both NM instances are operational.
- When route synchronization is performed while one NM is down or an NM is active on the peer NM, chunk programming inconsistencies may occur on the User Plane Function (UPF). Although this is unlikely, any detected discrepancies can be resolved by executing the **route-sync** command after both NMs are fully active.

Verification

Use the **show ipam dp** *peerid* **ipv4-address** to view if the groups and chunks are associated with any one NM.

```
bng# show ipam dp peer-x1 ipv4-addr
Wed Oct  11 19:13:22.877 UTC+00:00

=================================================================================================
Flag  Indication: S(Static) O(Offline) R(For Remote Instance) RF(Route Sync Failed)
G:N/P Indication: G(Cluster InstId) N(Native NM InstId) P(Peer NM InstId)
=================================================================================================
StartAddress          EndAddress      Route            G:N/P    Utilization   Flag
AllocContext
=================================================================================================
33.0.0.0              33.0.1.255      33.0.0.0/23      1:0/-1   100.00%
automation-poolv4(automation-vrf)(asr9k-ACTIVE,asr9k-STANDBY)
33.0.6.0              33.0.7.255      33.0.6.0/23      1:0/-1   95.90%
automation-poolv4(automation-vrf)(asr9k-ACTIVE,asr9k-STANDBY)
33.0.10.0             33.0.11.255     33.0.10.0/23     1:0/-1   0.59%
automation-poolv4(automation-vrf)(asr9k-ACTIVE,asr9k-STANDBY)
44.44.1.0             44.44.1.255     44.44.1.0/24     1:N/A                 S
automation-poolv4(automation-vrf)(asr9k-ACTIVE,asr9k-STANDBY)
```

In the above example, the peer-x1 is associated with NM-0, and all chunks are associated only with NM-0.

Dynamic Chunk Allocation for CP-GR

Table 58: Feature History

Feature Name	Release Information	Description
Dynamic Chunk Allocation for CP-GR	2024.02.0	This feature provides you greater control over how chunks are managed across active and remote clusters. This feature introduces an optimized mechanism that eliminates the need for pre-reserving chunks on the remote cluster, thus conserving resources and improving overall system efficiency.

In the current CP-GR setup, a large number of chunks are reserved in advance for a remote cluster to ensure a smooth switchover in case of failover events. These chunks are used only when there is a switchover or when all chunks in the active cluster are exhausted, causing inefficient chunk reservation.

The Dynamic Chunk Allocation for CP-GR feature cancels the need to reserve additional chunks on the remote cluster. Instead, during a switchover event, the same chunks initially assigned to Network Modules (NMs) in the primary cluster are seamlessly reused in the remote cluster, ensuring a fluid transition without the need for pre-allocated chunk reserves. IPAM automatically performs reconciliation of all chunks during a switchover.

You can use the CLI command **reserve-chunk-for-remote-site { enable | disable }** to either select the traditional method of reserving chunks for the remote server or to adopt the new approach where local chunks are dynamically utilized by the remote site.

Configuration Example

The following is a sample configuration:

```
config
 ipam
  instance 1
   source local
   reserve-chunk-for-remote-site { enable | disable }
   address-pool SAMP_POOL_2048
    vrf-name FTTX-SUB-EAST
    ipv4
     split-size
      per-cache 32768
      per-dp    2048
     exit
     address-range 209.165.200.225 209.165.255.254
     address-range 209.165.201.1 209.165.201.30
    exit
   exit
  exit
 exit
```

NOTES:

- **reserve-chunk-for-remote-site enable**: Reserves chunks on the remote cluster.
- **reserve-chunk-for-remote-site disable**: Chunks are not reserved on the remote cluster. This is the default behavior.

Note The value of the **reserve-chunk-on-remote-site** CLI must not be altered dynamically. To enable or disable the **reserve-chunk-on-remote-site** command post initial IPAM configuration, shut down and restart the system. In case of a GR setup, both the clusters must be shut down and then restarted.

RADIUS-based address assignment and route-tagging

RADIUS-based address assignment and route-tagging is a subscriber management feature that

- allows direct (pool-less) assignment of IPv4 and IPv6 addresses from RADIUS attributes to subscriber sessions
- enables granular and configurable route-tagging for RADIUS-derived subscriber routes (distinct from dynamic pool summary tags), and
- supports fallback to local address pools when RADIUS does not supply addresses.

This feature supports both IPv4 and IPv6 addressing models, including PPPoE, LNS, DHCPv6, and SLAAC deployments.

Table 59: Feature History Table

Feature Name	Release Information	Feature Description
RADIUS-based address assignment and route-tagging	2026.03.0	You can now use IP addresses more efficiently and simplify route control for PPPoE and RADIUS-based subscribers. This feature lets the system use RADIUS-assigned IPv4 and IPv6 addresses or prefixes directly, without requiring global static IPAM pools in many cases. It also enhances routing efficiency through granular policy control.

Key attributes and applicability in subscriber address management

- Supports direct address assignment using RADIUS-supplied values, eliminating the need for static IPAM pool definitions per subscriber.
- Enables scalable and policy-driven routing using route-tags, reducing reliance on IP prefix sets for control.
- Applies to PPPoE and LNS subscriber sessions, including DHCPv6 and SLAAC models, in both legacy and new deployments.
- Supported on Cisco ASR 9000 platforms with IOS XR release 26.3.1 and later.

Comparison of legacy and optimized IP address management approaches

Attribute	Legacy pool-based implementation	Optimized RADIUS-based implementation
Address assignment	Requires static or dynamic IPAM pools	Direct from RADIUS (pool-less)
Route-tagging flexibility	Limited, based on pool prefix	Configurable per subscriber and address type
Operational complexity	Requires prefix policies and pool maintenance	Reduced configuration, more automation via RADIUS

How RADIUS-based address assignment and route-tagging works

Summary

The key components involved in this process are:

- Control Plane (CP): Handles subscriber session setup, manages IP assignment logic, and processes RADIUS responses and configuration policies.
- RADIUS Server: Supplies per-subscriber IP addresses and prefixes using RADIUS attributes (for example, Framed-IP-Address).
- User Plane nodes: Installs and forwards subscriber routes as directed by the Control Plane, using route-tags for policy enforcement.
- IPAM Pools: Provide fallback or dynamic assignment of addresses when RADIUS information is absent.

Workflow

The process involves these stages:

1. Session initiation and RADIUS attribute delivery
   - The CP receives a request to start a subscriber session (for example, PPPoE, LNS, DHCPv6, or SLAAC) and queries the RADIUS server for user address attributes.
   - The RADIUS server returns attributes like Framed-IP-Address, Framed-IPv6-Address, or Delegated-IPv6-Prefix if available.
2. Policy evaluation for IPAM bypass and route-tag
   - The CP checks the subscriber profile or feature template for IPAM bypass and route-tag assignment instructions.

     If IPAM bypass is enabled and there is a RADIUS-assigned address or prefix, the CP prepares for direct assignment, skipping address pool lookup. The CP determines the appropriate route-tag (or tags).
3. Direct assignment and tagging

- With IPAM bypass active, the CP assigns the IP address or prefix provided by RADIUS and applies the configured route-tag for each address or prefix type.

  If multiple address types are assigned (for example, a framed address and a delegated prefix), each can receive its own route-tag.

4. Route installation and subscriber connectivity
   - User Plane nodes install the tagged subscriber routes, enabling forwarding of addresses and enforcing the configured policies.

   Pool-less assignments do not create summary routes or reserve gateway and broadcast addresses, which increases address efficiency.
5. Fallback to pools
   - If RADIUS does not provide an address or prefix, configuration allows fallback to standard IPAM pools.

Limitations for RADIUS-based address assignment and route-tagging

This feature is supported exclusively for cnBNG deployments along with ASR 9000. To ensure optimal performance and system stability, adhere to these operational guidelines and limitations.

- **Route tag capacity:** Limit the configuration to a maximum of four route tags. Exceeding this limit increases the packet length in CP-UP interface messages, which may lead to processing issues.
- **Packet size management:** Monitor CP-UP interface message sizes, especially during session replay, where full session information is exchanged. Ensure packet sizes do not exceed 1500 bytes to prevent issues associated with jumbo packets.

Configure RADIUS-based address assignment and route-tagging

Enable efficient IP address management for PPPoE and LNS subscriber sessions by reducing unnecessary reservations, allowing pool-less assignment from RADIUS, and providing granular route-tagging for both IPv4 and IPv6 subscriber routes.

Before you begin

- Ensure the RADIUS server supplies required address attributes: Framed-IP-Address, Framed-IPv6-Address, Delegated-IPv6-Prefix, Framed-IPv6-Prefix.
- Review planned route-tag values to avoid conflicts with existing routing policies.
- Confirm backup IPAM pools are configured for cases where RADIUS does not provide an address or prefix.

Follow these steps to configure RADIUS-based address assignment and route tagging:

Procedure

Step 1 Configure route tags for the relevant address assignment type (DHCPv6, PPPoE/LNS, or SLAAC).

a) DHCPv6 profile-level configuration

Example:

```
profile dhcp  profile-name
  ipv6 mode server
    server
      framed-ipv6-address route-tag tag-value
      delegated-ipv6-prefix route-tag tag-value
    exit
exit
```

b) DHCPv6 class-level configuration

Example:

```
profile dhcp profile-name
 ipv6
  mode server
  class class-name
   server
    framed-ipv6-address route-tag tag-value
    delegated-ipv6-prefix route-tag tag-value
   exit
  exit
 exit
exit
```

Note Class-level configuration overrides profile-level configuration.

c) PPPoE/LNS feature-template configuration

Example:

```
profile feature-template template-name
  ppp
     ipcp framed-address route-tag tag-value
    exit
exit
```

This feature-template can be applied to both PPPoE and LNS subscriber sessions.

d) SLAAC profile configuration

Example:

```
profile slaac profile-name
 framed-prefix route-tag tag-value
exit
```

This step sets route-tags for subscriber routes derived from RADIUS-provided address attributes.

NOTES:

- **framed-ipv6-address route-tag** *tag-value*: Applies route-tag to DHCPv6 IANA addresses derived from RADIUS.
- **delegated-ipv6-prefix route-tag** *tag-value*: Applies route-tag to DHCPv6 IAPD prefixes derived from RADIUS.
- **ipcp framed-address route-tag** *tag-value*: Applies route-tag to IPv4 host routes derived from Framed-IP-Address.
- **framed-prefix route-tag** *tag-value*: Applies route-tag to SLAAC prefixes derived from RADIUS.

Step 2 Configure IPAM bypass to allow direct address or prefix assignment from RADIUS attributes.

a) DHCPv6 profile-level configuration

Example:

```
profile dhcp profile-name
  ipv6 mode server
    server
      iana-pool-name pool-name
      iapd-pool-name pool-name
      framed-ipv6-address bypass-ipam
      delegated-ipv6-prefix bypass-ipam
    exit
exit
```

b) DHCPv6 class-level configuration

Example:

```
profile dhcp profile-name
 ipv6
  mode server
  server
   iana-pool-name pool-name
   iapd-pool-name pool-name
  exit
  class class-name
   server
    iana-pool-name pool-name
    iapd-pool-name pool-name
    framed-ipv6-address bypass-ipam
    framed-ipv6-address route-tag tag-value
    delegated-ipv6-prefix bypass-ipam
    delegated-ipv6-prefix route-tag tag-value
   exit
  exit
 exit
exit
```

c) PPPoE/LNS feature-template configuration

Example:

```
profile feature-template template-name
  ppp
    ipcp
      peer-address-pool pool-name
      framed-address bypass-ipam
    exit
exit
```

d) SLAAC profile configuration

Example:

```
profile slaac profile-name
 managed-config-flag enable
 other-config-flag enable
 prefix-pool pool-name
 framed-prefix bypass-ipam
 framed-prefix route-tag tag-value
exit
```

Bypassing IPAM enables the system to assign subscriber addresses directly from RADIUS without requiring pre-defined static pools.

- A fallback pool is used if RADIUS does not supply an address or prefix.

Table 60: IPAM bypass logic

If...	Then...
RADIUS supplies address or prefix.	System assigns address or prefix and tags route, bypassing IPAM.
RADIUS does not supply address or prefix.	Fallback to configured IPAM pool for assignment.
IPAM bypass not enabled in configuration.	Subscriber address assigned from pool as usual.

NOTES:

- **framed-ipv6-address bypass-ipam**: Specifies that the IPv6 address (IANA) received from RADIUS is used directly and IPAM is not invoked.
- **framed-ipv6-address route-tag** *tag-value*: Applies route-tag to DHCPv6 IANA addresses derived from RADIUS.
- **delegated-ipv6-prefix bypass-ipam**: Specifies that the delegated IPv6 prefix (IAPD) received from RADIUS is used directly and IPAM is not invoked.
- **delegated-ipv6-prefix route-tag** *tag-value*: Applies route-tag to DHCPv6 IAPD prefixes derived from RADIUS.
- **ipcp peer-address-pool** *pool-name*: Specifies the local IPv4 pool used for fallback dynamic allocation.
- **ipcp framed-address bypass-ipam**: Specifies that the IPv4 address received from RADIUS (Framed-IP-Address) is used directly and IPAM is not invoked.

- **framed-prefix bypass-ipam**: Specifies that the IPv6 prefix received from RADIUS is used directly and IPAM is not invoked.
- **framed-prefix route-tag** *tag-value*: Applies route-tag to SLAAC prefixes derived from RADIUS.
- **managed-config-flag enable**: Enables the managed configuration flag.
- **other-config-flag enable**: Enables the other configuration flag.

Step 3 Verify the operational state to confirm address assignment, route-tagging, IPAM bypass, and reserved address settings are functioning as intended.

- Use the **show route | include tag** *tag-value* command to verify route-tags.
- Use the **show subscriber session detail** command to verify IPAM bypass for a subscriber session.

Reserved IPv4 address controls for PPPoE pools

Reserved IPv4 address control for PPPoE pools is a configuration option that

- allows you to disable gateway, summary-route, and broadcast address reservations on individual PPPoE pools
- maximizes assignable IPv4 space by reclaiming addresses that would otherwise be reserved by default, and
- enables flexible, per-pool IP address management to optimize network resources without impacting service integrity.

Table 61: Feature History Table

Feature Name	Release Information	Feature Description
Reserved IPv4 address controls for PPPoE pools	2026.03.0	You can increase IPv4 address utilization by selectively disabling reservation of gateway, summary-route, and broadcast addresses in PPPoE pools. This feature allows more unused reserved addresses to be assigned to subscribers, helping you maximize available address space without impacting your service.

Traditionally, IPAM reserves the first (gateway), last (broadcast), and summary route addresses within IPv4 pools. For PPPoE pools, these reservations are often unnecessary. This feature allows you to disable these reservations on a per-pool basis, maximizing how many addresses you can assign to subscribers without affecting service integrity.

Limitations for reserved IPv4 address controls for PPPoE pools

This feature is supported exclusively for cnBNG deployments along with ASR 9000. To ensure optimal performance and system stability, adhere to these operational guidelines and limitations.

- **Configuration sequence**: Do not enable gateway, broadcast, or summary-route reserve disable while address ranges are active. You must remove existing address ranges before applying these settings.
- **Version downgrade risks**: Avoid downgrading from Release 2026.03.0 to earlier versions that do not support this capability. Downgrading can result in service impact and may cause inaccuracies in IPAM statistics due to the handling of special IP addresses assigned to subscribers.
- **Compatibility constraints**: When using cnBNG 2026.03.0 with IOS XR 26.2.1, ensure that gateway reservation disable is not enabled under the IPAM pool. Enabling this feature in this specific combination will result in the summary route failing to carry gateway information and will cause the auto loopback selection in NAL (ASR 9000) to fail during session programming.

Configure reserved IPv4 address controls for PPPoE pools

Enable efficient IPv4 address management for PPPoE subscriber sessions by allowing configurable exclusion of gateway, summary-route, and broadcast reservations in IPAM pools.

Before you begin

Review your PPPoE deployment to determine which default reservations can be disabled safely.

Follow these steps to configure reserved IPv4 address controls for PPPoE pools.

Procedure

Step 1 Configure reserved address controls for IPv4 IPAM pools to disable reservation of gateway, summary-route, and broadcast addresses.

Example:

```
ipam address-pool pool-name
  ipv4
    reserve gateway disable
     reserve summary-route disable
     reserve broadcast disable
     address-range start-ip end-ip
exit
```

This step increases address utilization in pools by making previously reserved addresses available.

- Disable reservations only when your deployment does not require these addresses for operation.

Table 62: Reserved address control logic

If...	Then...
Pool is used for PPPoE and does not require gateway reservation.	Disable gateway reservation for maximum address availability.
Summary-route and broadcast reservations are not needed.	Disable summary-route and broadcast reservations.

If...	Then...
Pool settings changed while ranges are online.	Take pool offline and reapply address ranges to avoid conflicts.

NOTES:

- **reserve gateway disable:** Disables reservation of the gateway address from each chunk/subnet.
- **reserve summary-route disable**: Disables reservation of the summary-route address from each chunk/subnet.
- **reserve broadcast disable**: Disables reservation of the broadcast address from each chunk/subnet.
- **address-range** *start-ip end-ip*: Defines the IPv4 address range for the pool.

Step 2 Use the **show ipam pool** *pool-name* command to verify pool address reservations.

Example:

```
bng# show ipam pool poolv4

Tue Aug  4  15:29:28.450 UTC+00:00
-------------------------------------------------------
 Ipv4Addr   [Total/Used/Reserved/Utilization] = 6619136 / 0 / 0 / 0.00%
 Ipv6Addr   [Total/Used/Reserved/Utilization] = 0 / 0 / 0 / 0.00%
 Ipv6Prefix [Total/Used/Reserved/Utilization] = 0 / 0 / 0 / 0.00%
 Instance ID                          = 1
-------------------------------------------------------
```

Periodic reconciliation

Periodic reconciliation is a system process that

- periodically and automatically reviews IP address records for inconsistencies
- identifies and reclaims stale or unused IP addresses, and
- increases system reliability by using robust bulk record retrieval methods.

This feature ensures that IPAM maintains accurate records of IP address usage and reduces the risk of address exhaustion caused by unreleased or orphaned IP allocations.

Table 63: Feature History

Feature Name	Release Information	Description
Periodic reconciliation	2025.03.0	You can now keep your IP address records accurate and avoid address exhaustion. This feature automatically identifies and reclaims stale IP addresses through a periodic, automated process, making management easier and more reliable.

Traditionally, IPAM systems might not have performed regular reconciliation to release stale IP addresses. With this feature, a new automated routine streamlines the cleanup of unused or stale IPs, improving overall IP utilization.

Key aspects:

- **Periodic routine:** The process runs automatically based on a user-configurable frequency (defined in days).
- **Automatic cleanup:** Stale IP addresses are identified and released with no manual intervention required, enhancing IP address utilization and operational efficiency.
- **Configuration:** You can enable reconciliation flows and configure their frequency within the IPAM instance settings. By default, this process is disabled and requires explicit setup.

Reconciliation scenarios

There are three main scenarios for running reconciliation in the current system:

1. **Automatic reconciliation by Node Manager (NM)**

   NM runs reconciliation automatically when a pod restarts or during a CP-GR role switchover.

2. **Manual trigger of reconciliation**

   You can manually trigger reconciliation if you detect inconsistencies between IP allocation in IP address management (IPAM) and the CDL session count.

3. **Periodic reconciliation**

   To avoid manual intervention, you can schedule periodic reconciliation. This feature automatically detects and fix discrepancies between IPAM and CDL session data.



Note It is recommended to schedule periodic reconciliation during periods of low system activity (for example, at midnight) to reduce any potential performance impact.

Configure periodic reconciliation

Set up periodic reconciliation to automatically reclaim stale IP addresses and maintain accurate IPAM records.

Use this configuration when you want IPAM to automatically identify and clean up unused or stale IP allocations without requiring manual intervention.

Procedure

Enter IPAM instance configuration mode, and configure the reconciliation schedule as needed.

Example:

```
config
 ipam
  instance  gr_instance_id
```

```
reconcile-schedule
 time-of-day-hour hour_of_day
 time-of-day-minute mins
 frequency days
end
```

NOTES:

- **reconcile-schedule**: Configures the reconciliation schedule.
- **time-of-day-hour** *hour*: Specifies the hour of day when reconciliation should run. The time values are in the cluster time zone and use the 24-hour format (0–23).
- **time-of-day-minute** *mins*: Specifies the minute of the hour when reconciliation should start. The value ranges from 0 to 59.
- **frequency** *days*: Specifies how frequently (in days) reconciliation should run. The value ranges from 1 to 30.

Example:

This is a sample configuration to run reconciliation daily at 03:15 AM for IPAM instance 1:

```
config
 ipam
  instance 1
   reconcile-schedule
    time-of-day-hour 3
    time-of-day-minute 15
    frequency 1
   end
  exit
 exit
```

IPAM automatically performs reconciliation at the specified time and frequency to regularly clean up stale IP addresses.

Periodic reconciliations for ID Manager

Periodic reconciliation for ID Manager is a feature that automatically identifies and clears stale IDs from the ID Manager and improves reliability and efficiency by preventing resource exhaustion.

Table 64: Feature history

Feature Name	Release Information	Description
Periodic reconciliations for ID Manager	2026.01.0	You can now enhance resource utilization and management efficiency by automatically removing stale IDs from the ID manager. This feature allows you configure the frequency and timing of reconciliations to suit your needs.

Periodic reconciliations address situations where, over time, unused or "stale" IDs may accumulate in the ID Manager due to network events or failures. Without regular cleanup, these stale IDs can reduce the efficiency

of ID manager and potentially impact network operations. By running periodic reconciliation, the system ensures that only active and necessary IDs remain allocated.

Best practice for scheduling periodic reconciliation

We recommend scheduling periodic reconciliation during periods of low network activity (for example, at midnight or during times of minimal session churn). This helps minimize the performance impact on live operations and ensures efficient resource cleanup with minimal disruption.

Configure periodic reconciliation for ID Manager

Enable and schedule automated cleanup of stale IDs in the ID Manager.

Before you begin

Ensure you have the necessary privileges to modify ID Manager settings.

Follow these steps to configure periodic reconciliation for ID Manager.

Procedure

Step 1 Specify the time of day and frequency for reconciliation.

Example:

```
config
idmgr
   reconcile-schedule
    time-of-day-hour hour_of_day
    time-of-day-minute mins
    frequency days
   end
```

NOTES:

- **reconcile-schedule**: Configures the reconciliation schedule.
- **time-of-day-hour** *hour*: Specifies the hour of day when reconciliation should run. The time values are in the cluster time zone and use the 24-hour format (0–23).
- **time-of-day-minute** *mins*: Specifies the minute of the hour when reconciliation should start. The value ranges from 0 to 59.
- **frequency** *days*: Specifies how frequently (in days) reconciliation should run. The value ranges from 1 to 30.

Example:

This is a sample configuration to run reconciliation every day at 2:30 AM:

```
config
 idmgr
   reconcile-schedule
    time-of-day-hour 2
    time-of-day-minute 30
    frequency 1
```

```
  end
 exit
exit
```

Step 2 Use the **show idmgr-reconcile status** command to view the current status and schedule of periodic reconciliation in ID Manager.

IPAM Route Programming Enhancements

Table 65: Feature History

Feature Name	Release Information	Description
IPAM Route Programming Enhancements	2024.04.0	IPAM now programs routes asynchronously, improving system stability and performance. IPAM sends route update requests and handles responses in separate routines, allowing continuous address allocation without delays.

Prior to Release 2024.04.0, IPAM programs routes toward UP in a synchronous manner. This means IPAM waits for a response from UP before processing any other requests for that AFI. If there is a delay in the route update response from UP, goroutines on the Node Manager (NM) start to pile up. Significant delays can cause the NM to crash. Also, new address allocations on IPAM remain blocked until the response is received.

Asynchronous Route Programming Implementation

Starting Release 2024.04.0, the IPAM is enhanced to program routes asynchronously. IPAM sends the route update request and waits for the response in a separate routine. Based on the response, IPAM moves the chunks to the proper state.

Scenarios for Route Programming

IPAM programs routes toward UP in two scenarios:

- When the first subscriber comes to IPAM for a given AFI on an NM.
- When there is a threshold hit for the UP or SRG-Group for that AFI on NM.

Scenario 1: First Subscriber

When the first subscriber comes to IPAM for a given AFI, the behavior remains unchanged. IPAM programs the route for that UP in a synchronous manner. This usually happens during the initial system setup, where delays are not expected. Since IPAM handles UP or SRG-Group registration during this flow, it remains synchronous for simplicity.

Scenario 2: Threshold Hit

When there is a threshold hit for a given UP or SRG-Group for an AFI, IPAM fetches new chunks from NM or Cachepod, and programs the route toward the UP or SRG-Group asynchronously. This ensures no impact on existing or new IP allocations.

No explicit configuration is required for this functionality. This feature becomes applicable as new routes are programmed for the UP. The system checks the PFCP retransmission timeout configuration, and if this configuration is set, the route retry is done based on the configured timeout. Otherwise, by default, the retry occurs after 15 seconds plus a 5-second buffer.

The following is a sample PFCP retransmission timeout configuration:

```
instance instance-id 1
 endpoint n4-protocol
  retransmission timeout 15 max-retry 1
 exit
exit
```

Pre-Allocation of Gateway IP and Address Chunks

Table 66: Feature History

Feature Name	Release Information	Description
Pre-Allocation of Gateway IP and Address Chunks	2026.04.0	The `ipam-address-chunk` action command has been modified to deprecate the **pool-group-tag** keyword and now supports only the **pool-name** keyword.
Pre-Allocation of Gateway IP and Address Chunks	2024.04.0	This feature ensures a smoother and more efficient onboarding process for new subscribers by reserving the first IP address in the allocated chunk for gateway functionalities.

Gateway IP Allocation

In a routed network configuration, the subscriber's gateway IP resides at the first Layer 3 hop on the access side router. The gateway IP is the first IP address within the allocated chunk or address range for the access side router. This first IP address is reserved and will not be assigned to any subscribers. Reserving this IP ensures that it is consistently available for gateway functionalities.

Address Chunk Pre-Allocation

Before onboarding the first subscriber, it is necessary to pre-allocate the address chunk and the first IP address within this chunk. This pre-allocation allows the first IP to be programmed on the access side router in advance, ensuring that the network infrastructure is prepared before any subscribers connect. Pre-allocating the chunk is mandatory for session bring up on routed SRG groups.

Administrative Command for Pre-Allocation

To support this pre-allocation process, we have introduced a new administrator-triggered command, **ipam-address-chunk allocate**. This command enables the pre-allocation of the address chunk and the gateway IP. Administrators must use this command whenever a new access interface or port is onboarded. If there are multiple BNG access interfaces on the same access router, pre-allocation must be performed multiple times.

Each time the **ipam-address-chunk allocate** action command is executed, the system attempts to allocate one new chunk associated with the ipam-dp-key.

Starting with Release 2026.04.0, the **pool-group-tag** keyword is deprecated from the **ipam-address-chunk allocate** action command.

Address Chunk Release

You can also release an IP address chunk when the data plane (DP) is released, particularly after bringing down subscribers. During the removal of an SRG group, it is mandatory to follow the Method of Procedure (MOP), which includes bringing down all subscribers in the group and executing an action command (**ipam-address-chunk release**) to release the DP and its associated chunks.

Starting with Release 2026.04.0, the **pool-group-tag** keyword is deprecated from the **ipam-address-chunk release** action command.

The command now releases resources at the chunk level instead of always releasing the entire ipam-dp-key.

Key behavior:

- Each time the **ipam-address-chunk release** action command is run, the system attempts to release one chunk associated with the ipam-dp-key.
- If multiple chunks are allocated from the selected pool, the system releases any one chunk that is completely free.
- Chunks that still have allocated IP addresses are not released.
- The entire ipam-dp-key is released only if the chunk being removed is the last chunk associated with the ipam-dp-key.

Configure Pre-Allocation of Gateway IP and Address Chunks

Procedure

Step 1 Use the **ipam-address-chunk** action command to configure pre-allocation of Gateway IP and address chunks.

Example:

```
ipam-address-chunk { allocate | release { [   pool-name name ]}
     { address-type [ ipv6-addr | ipv6-prefix | ipv4 | ipv6 ] }
       [ ipam-dp-key dp_key ]
   [ gr-instance  gr_instance ]
     [ srg-peer-id  srg_peer_id ]
```

NOTES:

- **allocate**: Enables pre-allocation of IP address chunk and the gateway IP.
- **release**: Releases an IP address chunk.
- **pool-name**: Provide pool-name . This should match the pool information configured in the DHCP profile.

  Starting with Release 2026.04.0, the **pool-group-tag** keyword is deprecated from the **ipam-address-chunk** command.
- **address-type**: Specify one of the four possible address types:

- ipv6-addr
- ipv6-prefix
- ipv4
- ipv6



Note When the **ipam-address-chunk** action command is executed with the **ipv6** address type, IPAM checks for a pool configured with **split-prefix-iana-first** enabled and allocates both IANA and IAPD from the same prefix. If no such pool is found, an error is returned.

This is a mandatory parameter.

- **ipam-dp-key**: Specifies the data plane key for IP management. This is a mandatory parameter.
  - **Routed SRG case**: Indicates the value of **ipam-dp-key** specified in the DHCP pool. Currently, circuit-id is supported. Essentially, use the value of circuit-id set on the access-side OLT.
  - **Non-Routed SRG case**: The **ipam-dp-key** can either be the same as the srg-peer-id or it can be different.
  - **Non-Routed Non SRG case**: This scenario is not supported currently.
  - **Routed Non SRG case**: This scenario is not supported currently.
- **gr-instance**: Specifies the GR instance information. If not provided, the local gr-instance is used as the default value. This is a mandatory parameter.
- **srg-peer-id**:The SRG group peer-id as specified in the configuration. This is a mandatory parameter.

The output of this action command provides information about the chunk and the first IP address that is reserved. For example,

```
bng# ipam-address-chunk allocate instance-id 1 pool-name dhcp-ipv6-iapd ipv6-prefix ipam-dp-key
INGJRJKTMDHRTW6001ENBESR001 srg-peer-id Peer1
Sat Aug  24 06:27:29.200 UTC+00:00
result
Gateway Address: 2001:DB8::1/50
```

Step 2 Use the **show ipam { dp | dp-tag }** *value***{ ipv6-addr | ipv6-prefix | ipv4-addr }** to view the reserved IP address and the summary route of the allocated chunks. This information is useful for identifying the first IP address that needs to be configured on the access side router.

Example:

```
 show ipam dp INGJRJKTMDHRTW600TB2DEVICE11101 ipv6-addr
```

```
==========================================================================================================


Flag  Indication: S(Static) O(Offline) R(For Remote Instance) RF(Route Sync Failed) F(Fixed Chunk
for DP)

Other Indication: A+(Waiting for route update response) QT*(Quarantined due to route delete failure)
```

```
                    QT+(Waiting for route update response post timeout)

G:N/P Indication: G(Cluster InstId) N(Native NM InstId) P(Peer NM InstId)

================================================================================================================

StartAddress                    EndAddress                Route                       GatewayAddress
       G:N/P    Utilization   Flag      AllocContext

================================================================================================================

2001:DB8::8000                      2001:DB8::bfff               2001:DB8::8000/114
2001:DB8:8001/114           1:1/-1   0.01%         F         dhcp-ipv6-iana-11(FTTX_SUB)

================================================================================================================
```

IANA and IAPD Allocation from Same IP Range

Table 67: Feature History

Feature Name	Release Information	Description
Offline address range support	2025.03.0	You can now mark address ranges as offline. With this feature, when you set a chunk offline during address range deletion, IPAM gets notified and stops allocating new IPs or chunks from that space. This ensures address ranges are managed safely and efficiently.
Static IP pool support	2025.03.0	We have enhanced this feature to support static IP pools, allowing both IANA and IAPD to share a single address scope. This helps you reduce the number of routes in your network and simplifies management.
IANA and IAPD Allocation from Same IP Range	2025.01.0	You can now reduce the route count by using a single summary route for both Internet Assigned Numbers Authority (IANA) and Internet Address Prefix Delegation (IAPD) from a single IP pool. This feature optimizes the use of IPv6 prefix ranges and offers flexibility in address allocation. By introducing a virtual address range within the IPAM data structure, you can reserve the first prefix for IANA and use subsequent prefixes for IAPD with the **split-prefix-iana-first** command.

The IANA and IAPD Allocation from Same IP Range feature allows the allocation of both Internet Assigned Numbers Authority (IANA) and Internet Address Prefix Delegation (IAPD) from a single IP pool, specifically an IPv6 prefix range. The primary objective is to optimize the use of IP address spaces and reduce the route count by using a single address scope for both IANA and IAPD in a subscriber redundancy group (SRG). This feature introduces a virtual address range within the IPAM data structure to facilitate this allocation.

The feature enables the use of a single IPv6 subnet or summary route for both IANA and IAPD, which assists in reducing the route count. When the IPAM configuration includes the **split-prefix-iana-first** command, the first prefix in the allocated range is reserved for IANA, while the subsequent prefixes are used for IAPD.

For IANA, the number of addresses assigned to a dpKey is fixed at 65,536. Even if the usage threshold is reached, no additional IANA addresses can be allocated.

Address range handling for static pools

For static pools, there is no split-size configuration, so the entire address range is treated as a single chunk. As a result, each IAPD address range will have one corresponding IANA virtual chunk derived from it.

Restrictions for IANA and IAPD Allocation from Same IP Range

These restrictions apply to the IANA and IAPD Allocation from Same IP Range feature:

- Once **split-prefix-iana-first** is configured for a pool, it cannot be removed unless the entire pool configuration is deleted.
- When an IPv6 address range or prefix range is configured, you cannot dynamically enable **split-prefix-iana-first**. To change the pool behavior to **split-prefix-iana-first**, you must mark all IPv6 address ranges or prefix ranges offline, delete them, and then make the configuration change.
- IPAM dpKey registration or release using the action command for the address type **ipv6** is only possible for pools configured with **split-prefix-iana-first** using either a pool name or group name. Conversely, you cannot use the address types **ipv6-addr** or **ipv6-prefix** for pools that have **split-prefix-iana-first** configured.

Configure IANA and IAPD Allocation from Same IP Range

Procedure

Step 1 Use the **split-prefix-iana-first** command under IPv6 settings to enable the feature.

Example:

```
config
  ipam
   instance instance_id
    address-pool pool_name
     vrf-name vrf_name
     ipv4
      split-size
       per-cache value
       per-dp value
      exit
       address-range start_ipv4_address end_ipv4_address
     exit
     ipv6
      split-prefix-iana-first
```

```
    prefix-ranges
     split-size
      per-cache value
      per-dp value
     exit
     prefix-range prefix_value prefix-length length
    exit
   exit
  exit
 exit
```

NOTES:

- **ipam**: Enters the IPAM configuration mode.
- **instance** *instance_id*: Specifies the IPAM instance and enters the instance sub-mode. instance_id must be an integer. The valid value ranges from 1 to 8.
- **address-pool***pool_name*: Configures the address pool configuration. *pool_name* must be the name of the address pool.
- **vrf-name** *vrf_name*: Specifies the name of the VPN routing and forwarding (VRF) for the pool.
- **ipv4**: Enters the IPv4 mode of the pool.
- **split-size { per-cache** *value* **| per-dp** *value* **}**: Specifies the size of the IPv4 range to be split for each IPAM cache allocation. The IPAM server consumes this configuration. The **no** form of this command disables the splitting of the address-ranges into smaller chunks.

  per-cache *value*: Specifies the size of the IPv4 range to be split for each Data-Plane (User-Plane) allocation. The valid values range from 2 to 262144. The default value is 1024.

  The IPAM cache consumes this configuration.
- **per-dp** *value*: Specifies the size of the IPv4 range to be split for each Data-Plane (User-Plane) allocation. The valid values range from 2 to 262144 The default value is 256.
- **address-range** *start_ipv4_address end_ipv4_address*: Configures the IPv4 address range with the starting and ending IPv4 address.
- **ipv6**: Enters the IPv6 mode of the pool.
- **split-prefix-iana-first**: Enables allocation of both IANA and IAPD from the same IP pool.
- **prefix-ranges**: Enters the prefix ranges mode.
- **prefix-range** *prefix_value* **prefix-length** *length*: Configures the IPv6 prefix range. *prefix_value* specifies the IPv6 prefix range.

  prefix-length *length* specifies the IPv6 prefix length.

Step 2 Use the IPAM show commands to check the allocation status and utilization of IPv6 addresses and prefixes.

Example:

1

```
bng# show ipam dp INMUNVMBGNSMNB0044FSAOLT001

Thu Dec  5  07:27:53.820 UTC+00:00
```

```
-------------------------------------------------------
 Ipv4Addr   [Total/Used/Reserved/Utilization] = 0 / 0 / 0 / 0.00%
 Ipv6Addr   [Total/Used/Reserved/Utilization] = 65536 / 0 / 2 / 0.00%
 Ipv6Prefix [Total/Used/Reserved/Utilization] = 256 / 0 / 1 / 0.39%
 Instance ID                          = 1
 SRG PeerID                           = Peer1
 L3Routed                             = true
-------------------------------------------------------
```

In this example, **65536** indicates the total address count for virtual IANA.

Example:

2

```
bng# show ipam dp INMUNVMBGNSMNB0044FSAOLT001 ipv6-addr

Thu Dec  5  07:27:57.929 UTC+00:00
=============================================================================================================
Flag  Indication: S(Static) O(Offline) R(For Remote Instance) RF(Route Sync Failed) F(Fixed Chunk
for DP) V(Virtual)
Other Indication: A+(Waiting for route update response) QT*(Quarantined due to route delete failure)

                  QT+(Waiting for route update response post timeout)
G:N/P Indication: G(Cluster InstId) N(Native NM InstId) P(Peer NM InstId)
=============================================================================================================
StartAddress         EndAddress      Route                GatewayAddress     G:N/P    Utilization
  Flag      AllocContext
=============================================================================================================
2001:DB8::8000      2001:DB8::bfff  2001:DB8::8000/114    2001:DB8:8001/114   1:0/-1     0.01%
   V    group-naoverpd5(automation-vrf)
=============================================================================================================
```

Flag **V** - indicates a virtual address range for IANA.

Static IP Support

Feature Summary and Revision History

Summary Data

Table 68: Summary Data

Applicable Product(s) or Functional Area	cnBNG
Applicable Platform(s)	SMI
Feature Default Setting	Disabled - Configuration Required
Related Changes in this Release	Not Applicable
Related Documentation	*Cloud Native BNG Control Plane Command Reference Guide*

Revision History

Table 69: Revision History

Revision Details	Release
Multiple UPFs support with a static IP-pool.	2024.01.0
First introduced.	2021.03.0

Feature Description

IPAM is the core component of the subscriber management system. Traditional IPAM functionalities prove insufficient in the Cloud Native network deployments. Therefore, IPAM requires additional functionalities to work with the Cloud Native subscriber management system.

The Static Pool Enhancements provide more functionalities on the cnBNG using IPAM. These functionalities are as follows:

- Supports addition of a new static-pool or static-addr-range dynamically when system is running.
  - Send routes when User Plane function (UPF) is already associated.
- Supports removal of a new static-pool or static-addr-range dynamically when system is running
  - Mark the static-pool or static-addr-range offline.
  - Clear subscribers, manually.
  - Delete the configuration. IPAM sends the route-delete command to the UPF if it is already associated.
- The cnBNG Control Plane now supports sharing a single static IP-pool with multiple UPFs.
  - To enable sharing of the same static IP-pool across multiple UPFs, you must configure the static IP-pool as a Global Static pool. Global Static pool here refers to a static pool that is used by the IPAM module for sessions with IP addresses set from an external entity such as RADIUS, and not bound to any UPF.

    For configuration details, see Configuring a Static IP-pool to support multiple UPFs.

Note Changing the UPF name directly in the static IP pool is not supported. To change the UPF name, gracefully delete the static IP pool and readd the new UPF.

Note the following guidelines while configuring a static IP pool:

- An address pool is marked as ‘static’ during configuration. A given address pool can be either configured as 'dynamic' or 'static' mode only. It cannot be modified while the system is running.
- Each static pool is associated to an User Plane function (UPF). The IPAM configures all the address-ranges as routes on the UPF. The recommendation is to use the ‘no-split’ configuration to avoid having too many splits and routes.

- The IPV4 address ranges must be configured with the ‘default-gateway’ value, because unlike dynamic address-range, IPAM cannot assign ‘default-gateway’ on its own. Also, the static address-ranges must not be split further. Therefore, use the ‘no-split’ configuration.

Limitations

Support of multiple UPFs with a Static IP-pool feature has the following limitations:

- This feature is supported only for PPPoE subscribers.
- You must not dynamically change Global Static-Pool to non-Global Static-Pool (UPF specific static pool), or the reverse.
- You may encounter the following scenario when you want to delete an IPv4 address range in a static pool:

  Consider the following sample configuration.

```
config
  ipam
    instance 1
      address-pool old-static-pool-ft
        vrf-name vrf_lps
        static enable
        ipv4
          split-size
            no-split
          exit
          address-range 209.165.200.225 209.165.200.254 offline default-gateway
209.165.200.225
          address-range 209.165.201.1 209.165.201.30 default-gateway 209.165.201.1
         address-range 209.165.202.129 209.165.202.158 default-gateway 209.165.202.129
```

  If you want to delete the address range that is marked offline, and if you run the command `bng(config-ipv4)# no address-range 209.165.200.225 209.165.200.254 default-gateway 209.165.200.225`, only the default gateway address is deleted, and the address range is not deleted.

  To delete the address range, you must run the command as follows:

```
bng(config-ipv4)# no address-range 209.165.200.225 209.165.200.254
```

Note To delete the address range, ensure that **default-gateway** *ip_address* is not included in the command.

Configuring Static IP Support

This section describes how to configure the Static IP support feature.

Configuring the Static IP Support involves the following step.

Configuring Static IP Address Ranges, on page 330

Configuring Static IP Address Ranges

Use the following configuration to configure the Static IPv4 address ranges.

```
config
   ipam
      instance instance_id
         source { local | external }
         address-pool pool_name
         vrf-name vrf_name
         static enable [ user-plane value ] | [ srg-peer-id value ] | [
circuit-id value associated-srg-peer-id value ]
         ipv4
            split-size no-split
            exit
         ipv4
         address-range start_ipv4_address end_ipv4_address
            default-gateway ipv4_address
            exit
         exit
      exit
```

Similarly, you can configure static IPv6 address ranges.

NOTES:

- **ipam**: Enters the IPAM configuration mode.
- **address-pool***pool_name*: Configures the address pool configuration. *pool_name* must be the name of the address pool.
- **vrf-name** *vrf_name*: Specifies the name of the VPN routing and forwarding (VRF) for the pool.
- **static enable user-plane***user_plane_name*: Configures static IP details. Sets the specified User Plane (UP) as static. *user_plane_name* is the specified UP for this static pool.
- **static enable srg-peer-id** *value* : Specify the SRG-PeerID directly as the key.
- **circuit-id** *value* **associated-srg-peer-id** *value*: Specify the Circuit-ID as the key and explicitly associate it with the correct SRG-PeerID.
- **ipv4**: Enters the IPv4 mode of the pool.
- **split-size no-split**: Specifies that the address-ranges should not be into smaller chunks.
- **address-range** *start_ipv4_address end_ipv4_address*: Configures the IPv4 address range with the starting and ending IPv4 address.
- **default-gateway** *ipv4_address*: Specifies the IPv4 address of the default gateway.
- IPAM does only route-validation for Static IP. Validate if UPF has a route for a given static IP address or prefix. Individual address-management is not performed.

Configuring a Static IP-pool to Support Multiple UPFs

Use the following configuration to configure a static IP-pool to support multiple UPFs.

```
config
  ipam
    instance instance_id
      address-pool pool_name
        vrf-name string
        static enable
        exit
```

NOTES:

- **ipam**: Enters the IPAM configuration mode.
- **address-pool** *pool_name*: Configures the address pool configuration. *pool_name* must be the name of the address pool.
- **vrf-name** *vrf_name_value*: Specifies the name of the VPN routing and forwarding (VRF) for the pool.
- **static enable** : Enables sharing of the same static IP pool across multiple UPFs (global static pool) and SRGs.

Support for circuit-id as ipam-dp-key in static IPAM pools

Static IPAM pools now support using Circuit-ID as the ipam-dp-key. This enhancement provides you the capability to specify the associated SRG-PeerID directly in the static pool configuration.

Table 70: Feature History

Feature Name	Release Information	Description
Support for circuit-id as ipam-dp-key in static IPAM pools	2025.04.0	You can now use Circuit-ID as the ipam-dp-key in static IPAM pools, which increases flexibility for IP address management. With this feature, administrators can directly specify the associated SRG-PeerID in the static pool configuration.

Key details include:

- **Explicit SRG-PeerID specification:**

  A new CLI option allows administrators to specify the associated SRG-PeerID directly in the static pool configuration when using Circuit-ID as the ipam-dp-key.

- **Flexible User-Plane identification:**

  The system treats the **user-plane**, **srg-peer-id**, **circuit-id**, and **associated-srg-peer-id** CLI parameters as opaque strings. This means the IPAM configuration process does not enforce or validate the format or type of these values.

- **Administrator responsibility:**

Because IPAM cannot inherently distinguish between a user-plane name, SRG-PeerID, or Circuit-ID, administrators must ensure that when a Circuit-ID is used, the appropriate **associated-srg-peer-id** is also configured.

- **Applicability:**

  These configuration options and behaviors apply to both global static pools and non-global (local) static pools.

Configure associated-srg-peer-id when using circuit-id as ipam-dp-key in a static IPAM pool

When you configure a static IPAM pool and use a Circuit-ID as the ipam-dp-key, you must explicitly specify the associated SRG-PeerID.

Before you begin

- Ensure you have the correct Circuit-ID and associated SRG-PeerID values.
- Ensure that you have the required CLI privileges to edit the IPAM configuration.

Procedure

In IPAM configuration mode, use the **static enable** command to enable the static pool and assign the Circuit-ID with its associated SRG-PeerID.

Example:

```
config
  ipam
   instance instance_id
    address-pool pool_name
     vrf-name vrf_name
     static enable [ user-plane value ] | [ srg-peer-id value ] | [ circuit-id value
 associated-srg-peer-id value ]
     ipv4
      split-size
       no-split
      exit
      address-range start_ipv4_address end_ipv4_address
     exit
     ipv6
      address-ranges
       split-size
        no-split
      exit
      address-range start_ipv6_address end_ipv6_address
     exit
     prefix-ranges
       split-size
```

```
      no-split
     exit
     prefix-range prefix_value prefix-length length
    exit
   exit
  exit
 exit
```

NOTES:

- **static enable user-plane** *value* : Specify a user-plane name as the key.
- **static enable srg-peer-id** *value* : Specify the SRG-PeerID directly as the key.
- **circuit-id** *value* **associated-srg-peer-id** *value*: Specify the Circuit-ID as the key and explicitly associate it with the correct SRG-PeerID.

Static IP Mapping Database

Feature Summary and Revision History

Summary Data

Table 71: Summary Data

Applicable Product(s) or Functional Area	cnBNG
Applicable Platform(s)	SMI
Feature Default Setting	Disabled - Configuration Required
Related Documentation	Not Applicable

Revision History

Table 72: Revision History

Revision Details	Release
First introduced.	2022.02.0

Feature Description

The centralized IPAM component of the cnBNG CP enables mapping of static IPs for priority or VIP subscribers. The Static IP Mapping Database feature supports this requirement. It enables mapping of the MAC address of the subscriber to the static IP address. The static IP mapping database is maintained in the

cnBNG CP, which ensures that the particular VIP subscriber receives the mapped static IP address every time the subscriber connects.

Maintaining the static IP mapping database is the key function of this feature.

- Each static IP mapping entry is pushed via the CLI, NETCONF, or RESTCONF interfaces.
- Given the number of entries, the static IP mapping entry cannot be stored as a configuration in CONFD. Instead, mapping values are directly stored in IPAM. Therefore, the **show running config** CLI command does not display these entries.
- If a subscriber receives a framed IP address from RADIUS, this IP address is given priority over the static IP mapped in IPAM and is assigned to the subscriber. However, if RADIUS does not allocate any address and the DHCP profile has a static IP mapping configuration enabled (see, Configuring Static Database Key in DHCP Profile, on page 338), then it attempts to fetch the IP for this subscriber from the static IP mapping database.

Note

- A copy of static IP address mapping entries is stored in volume-mounted ETCD database for handling pod restart and image upgrades.
- The static IP mapping database remains persistent across the system shutdown and running mode.

How it Works

This section provides a brief of how the Static IP Mapping Database feature works.

The DHCP profile is configured with the *subscriber-key-template* that is used for IP mapping search. The DHCP forms the *subscriber-key* based on the DHCP profile configuration and incoming packet, and then forwards the request to the IPAM.

The IPAM searches the static database based on the *subscriber-key* input, finds the mapped IP address, and responds to the DHCP.

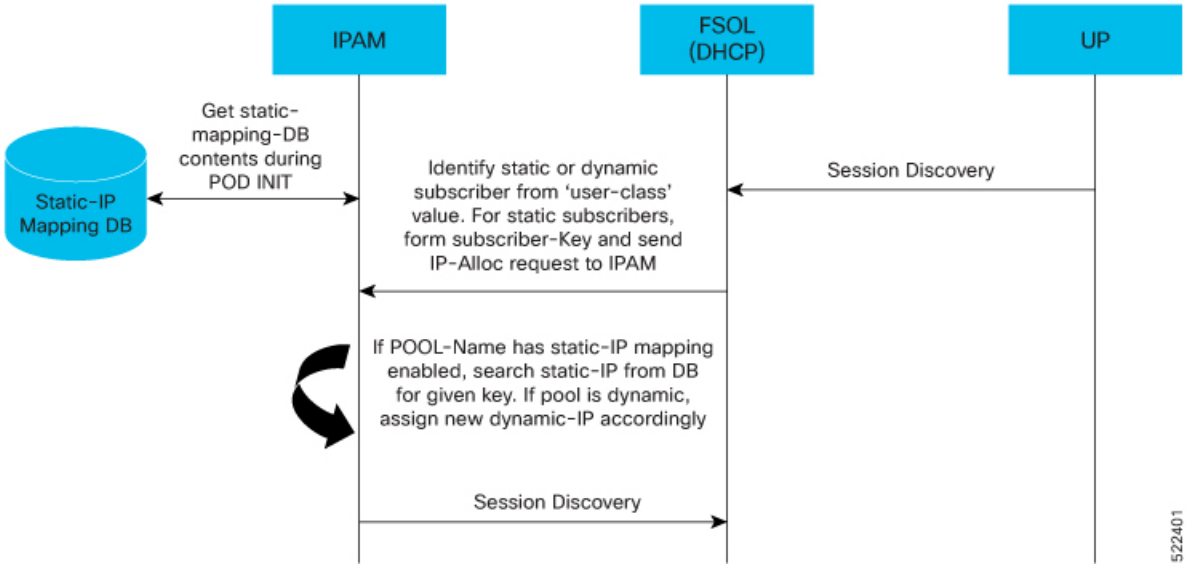
If the IP static database does not have an entry for the *subscriber-key* in the database, the IPAM returns a failure to the DHCP. This causes the DHCP to fail the session.

Call Flows

This section includes the following call flow.

Note The Static IP Mapping Database feature is supported only for IPoE (DHCP) sessions.

Figure 19: Static IP Mapping Call Flow



Configuring the Static IP Mapping Database

This section describes how to configure the Static IP Mapping Database feature.

Configuring the Static IP Mapping Database feature involves the following procedures.

- Configuring Static IP Mapping Database in IPAM, on page 335
- Configuring Static Database Key in DHCP Profile, on page 338

Configuring Static IP Mapping Database in IPAM

Use the following CLI commands to add or delete entries in the static IP mapping database.

```
ipam-staticdb mapping { pool pool_name
{ ipv4 { address ipv4_address subkey subkey }
| ipv6-addr { address ipv6_address subkey subkey }
| ipv6-prefix { address ipv6_prefix_address subkey subkey }
| delete }
```

NOTES:

- **pool** *pool_name*: Specifies the pool name.
- **ipv4 address** *ipv4_address* **subkey** *subkey*: Specifies the IPv4 address and subkey.
- **ipv6-addr address** *ipv6_address* **subkey** *subkey*: Specifies the IPv6 address and subkey.
- **ipv6-prefix address** *ipv6_prefix_address* **subkey** *subkey*: Specifies the IPv6 prefix address and subkey.
- **delete**: Deletes the static IP mapping database.

The static IP mapping database in CP can be in following format.

```
staticMapping:
  - poolName: <pool-name1>
    ipv4List:
     - <subscriber-key1>
       <static-ip-address1>
     - <subscriber-key2>
       <static-ip-address2>
 - poolName: <pool-name2>
    ipv4List:
     - <subscriber-key1>
       <static-ip-address1>
     - <subscriber-key2>
       <static-ip-address2>
```

Note

- The subscriber-key is defined based on the deployment need. For example, the subscriber-key can be the MAC address of the subscriber or a combination of multiple identifiers with some delimiters.
- For static IP key mapping, the only supported identifier is the MAC address.
- The IP address can be IPv4, IPv6-NA, or IPv6-PD.

Example

The following are the CLI and RESTCONF configuration examples.

Sample Output: CLI

The following is a sample CLI of the static IP mapping database.

```
bng# ipam-staticdb mapping pool { pool-name static-pool ipv4 { subkey aa11.0000.0001 address
 20.20.0.5 } ipv4 { subkey aa11.0000.0002 address 20.20.0.6 } ipv6-addr { subkey
aa11.0000.0001 address 2001:cb0::0:1 } ipv6-prefix { subkey aa11.0000.0001 address
2001:ef0::/48 } }
mapping-details
Static IP Mapping Success
```

Sample Output: RESTCONF

The following is a sample RESTCONF of the static IP mapping database.

```
curl -i -k -X "POST"
"https://restconf.bng-ops-center.10.84.102.254.nip.io/restconf/data/cnbng-static-map:ipam-static-db/mapping"
 -H 'Content-Type: application/yang-data+json' -H 'Accept: application/yang-data+json' -u
'admin:*****' \
-d $'{"pool":{"pool-name":"test1",
"ipv4": [
{"subkey": "0000.1111.2221","address": "1.1.1.1"}
{"subkey": "0000.1111.2222","address": "1.1.1.2"}
]}}'
```

Note The following caveats are applicable when configuring the static IP mapping database:

- Data is not validated for admininstrator-configured static database. However, input data is accepted as-is and stored. It is the responsibility of the adminstrator to configure and map valid IP addresses to the subscribers.

  For example, validation of address formats and duplicate IP addresses across subscribers is not supported.

- The following soft limits are applicable on the number of entries per request via the CLI or RESTCONF.
  - Number of static IP mapping entries per CLI request is 100.
  - Number of static IP mapping entries per RESTCONF request is 1000.

  Any number that exceeds more than the specified limits can be time consuming or lead to indeterministic behaviour.

Verifying Static IP Mapping Database in IPAM

Use the following **show** commands to verify the administrator-specified IP to MAC mapping.

- Number of mappings per AFI (IPv4, IANA, IAPD) level.

```
bng# show ipam-staticdb
staticdb-info
{
"staticdbInfo": [
{
"poolName": "static-pool",
"ipv4Total": 7,
"ipv6AddrTotal": 7,
"ipv6PfxTotal": 4
}
]
}
```

- Number of mappings per pool, per AFI level.

```
bng# show ipam-staticdb pool-name static-pool
staticdb-info
{
"staticdbInfo": [
{
"poolName": "static-pool",
"ipv4Total": 7,
"ipv6AddrTotal": 7,
"ipv6PfxTotal": 4
}
]
}
```

- MAC to IP mapping information.

```
bng# show ipam-staticdb pool-name static-pool ipv4-addr start-index 3
staticdb-info
{
"staticdbInfo": [
{
"poolName": "static-pool",
```

```
"ipv4Total": 7,
"ipv6AddrTotal": 7,
"ipv6PfxTotal": 4,
"ipv4Range": [
{
"subkey": "aa11.0000.0003",
"subip": "20.20.0.3"
},
{
"subkey": "aa11.0000.0004",
"subip": "20.20.0.4"
},
{
"subkey": "aa11.0000.0005",
"subip": "20.20.0.5"
},
{
"subkey": "aa11.0000.0006",
"subip": "20.20.0.6"
},
{
"subkey": "aa11.0000.0007",
"subip": "20.20.0.7"
}
]
}
]
}
```


Note By default only 1000 entries are displayed per index-based **show** command.

For example:

- When start-index is 100, entries from 100 to 1099 are displayed.
- When start-index is 1000, entries from 1000 to 1999 are displayed and so on.

Configuring Static Database Key in DHCP Profile

Use the following CLI commands to configure the static database key in the DHCP profile.

```
config
  profile dhcp dhcp_profile_name
    ipv4
      { class class_name | mode server | server }
       server static-ip-key identifier client mac_address client_mac_address

        exit
      exit
    exit
  exit
```

NOTES:

- **profile dhcp** *dhcp_profile_name*: Specifies the DHCP profile name.
- **ipv4**: Enters IPv4 configuration mode.

- **{ class** *class_name* **| mode | server }**: Enters Class, Mode Server, or Server configuration mode.
- **server static-ip-key identifier client mac_address** *client_mac_address*: Specifies the static-ip-key identifier and client MAC address. *client_mac_address* must be in Client MAC Address in AABB.CCDD.EEFF format.

The static database key in the DHCP profile can be in the following format:

```
profile dhcp dhcp-server1
 ipv4
  mode server
  server
   pool-name   pool-ISP
   dns-servers [ 8.8.8.8 ]
   lease days 10
   lease hours 0
   lease minutes 3
   static-ip-key identifier client-mac-address
  exit
  class c1
   server
    pool-name static-pool
    static-ip-key identifier client-mac-address
   exit
  exit
 exit
 ipv6
  mode server
  server
   iana-pool-name static-pool
   lease days 10
   lease hours 0
   lease minutes 4
   static-ip-key identifier client-mac-address
  exit
  class c2
   server
    iapd-pool-name static-pool
    static-ip-key identifier client-mac-address
   exit
  exit
 exit
exit
```

IP, ID, and Chunk event logging

IP, ID, and Chunk event logging is a resource management feature that

- records allocation, release, and movement events for IP addresses, IDs, and address chunks
- uses batch-enabled file-based logging for persistent and reliable tracking, and
- enables transparency, auditing, and debuggability for network resource operations.

Table 73: Feature history

Feature Name	Release Information	Description
IP, ID, and Chunk event logging	2025.04.0	You can now track the allocation, release, and movement of IP addresses, IDs, and chunks for improved debuggability and auditing. The system records detailed lifecycle events, including timestamps and relevant keys, and writes them to dedicated log files at regular intervals or after a set number of events.

This feature provides robust tracking for each change in the state of IP addresses, IDs, and chunks. Every allocation, release, and state transition (such as assignment or quarantine) is recorded with detailed data and timestamps.

Event data is stored using compressed log files located at `/var/log/ipameventlogging` within the NM pod.

Key capabilities

- **Comprehensive event tracking:** Captures all key lifecycle actions, including IP and ID allocations or releases and chunk movements across states (free, assigned, quarantine).
- **Batch and persistent file-based logging:** Aggregates events and periodically writes compressed logs for archival and future analysis.
- **Configurable controls:** Logging can be enabled or disabled per resource and per group (GR), via CLI options.
- **Failure handling:** Implements up to five retry attempts for write failures, with error status monitored in Grafana dashboards.
- **Performance optimization:** Balances retention and performance by batching 50,000 events or logging each minute.

Event types and log file formats

Event type	Logged data fields	Log file location and format
IP events	• IP • DpKey • SubId • Action • Timestamp	`/var/log/ipameventlogging/ipam_ip_eventlog_gr<grID>_<timestamp>.log.gz`

Event type	Logged data fields	Log file location and format
ID events	• ID • NsKey • Key • Action • SubId • Timestamp	/var/log/ipameventlogging/idmgr_id_eventlog_gr<cluster_GR><timestamp>.l
Chunk events	• StartIP • EndIP • DpKey • Action • Timestamp	/var/log/ipameventlogging/ipam_chunk_eventlog_gr<grID><timestamp>.log.

IP, ID, and Chunk event logging example

- When an IP address is assigned to a device, the event log file records the IP, operation (allocation), timestamp, and relevant identifiers.
- If a chunk of IP addresses is moved from “assigned” to “quarantine,” the transition is recorded in the chunk event log with start or end IPs, dp-key, and state change.

Configure event logging

Enable logging for IP, ID, and Chunk lifecycle events in IPAM to support operational monitoring and auditing.

You can selectively enable or disable event logging for IP address, ID, and Chunk activity through CLI commands.

Procedure

Step 1 Use the **ip-event-logging enable** command to enable event logging for IP address events, and the **ip-chunk-event-logging enable** command to enable logging for Chunk events.

Example:

```
ipam
  instance GR-instance-number
    ip-event-logging enable
    ip-chunk-event-logging enable
```

```
  exit
exit
```

NOTES:

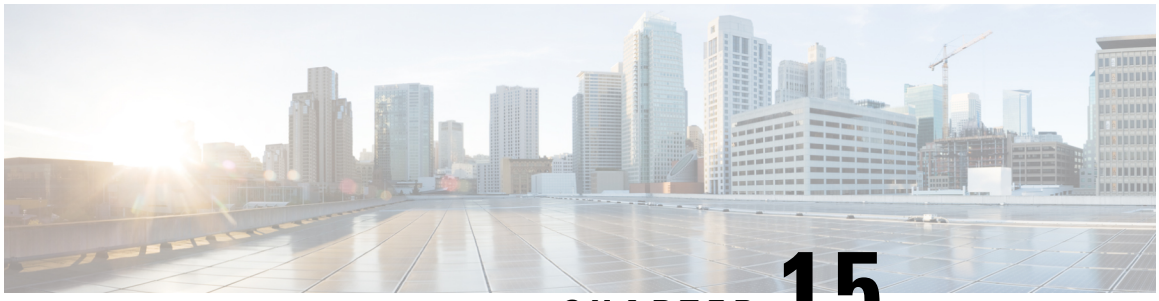
- **ip-event-logging enable**: Enables logging for IP address allocation and release events.
- **ip-chunk-event-logging enable**: Enables logging for Chunk allocation and movement events.

  You can enable either or both options independently, depending on which events you need to track.

Step 2 Use the **id-event-logging enable** command to enable event logging for ID allocation and release.

Example:

```
idmgr
  id-event-logging enable
exit
```



CHAPTER 15

L2TP Subscriber Management

- Feature Summary and Revision History, on page 343
- Feature Description, on page 344
- L2TP tunnel failover support, on page 346
- Seamless L2TP session continuity during server reboot, on page 349
- IETF Tagged Attributes on LAC, on page 350
- How it Works, on page 355
- Configuring the L2TP Subscriber Management Feature, on page 365
- Monitoring and troubleshooting, on page 368

Feature Summary and Revision History

Summary Data

Table 74: Summary Data

Applicable Product(s) or Functional Area	cnBNG
Applicable Platform(s)	SMI
Feature Default Setting	Disabled - Configuration Required
Related Changes in this Release	Not Applicable
Related Documentation	Not Applicable

Revision History

Table 75: Revision History

Revision Details	Release
Introduced support for L2TP tunnel reboot.	2026.01.0

Revision Details	Release
Introduced the **show l2tp-tunnel-instance upf** command for viewing L2TP tunnel instance details.	2026.01.0
Introduced the **show disconnect-history l2tp-tunnel upf** command for viewing L2TP tunnel disconnect history.	2026.01.0
Introduced support for L2TP tunnel failover.	2025.03.0
Enhancement Introduced: • Tunnel-Preference attribute • Tunnel-Client-Auth-ID attribute	2025.02.0
First introduced.	2021.04.0

Feature Description

Note This feature is Network Services Orchestrator (NSO) integrated.

Majority of the digital subscriber line (DSL) broadband deployments use PPPoE sessions to provide Subscriber services. These sessions terminate the PPP link and provide all the features, service, and billing on the same node. These sessions are called PPP Terminated (PTA) sessions .

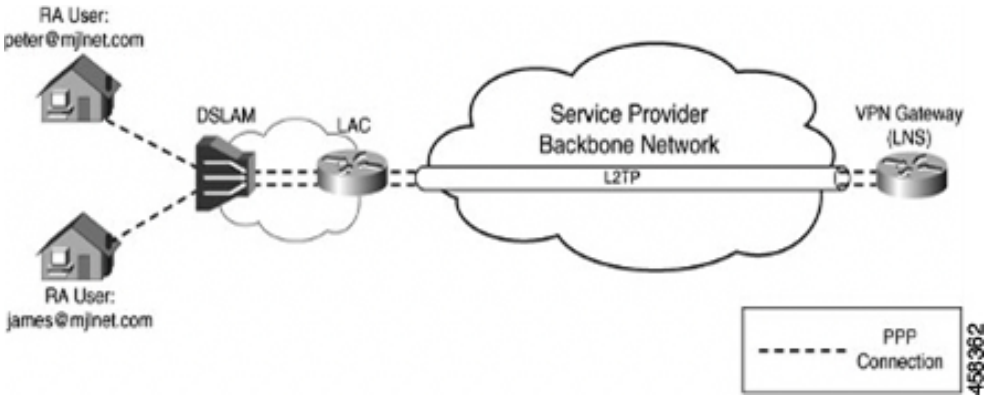
There are some wireline subscriber deployments are in wholesale-retail model where some ISPs work with others to provide the access and core services separately. In such cases, the subscribers are tunneled between wholesale and retail ISPs using the Layer 2 Tunneling Protocol (L2TP) protocol.

L2TP Overview

In cnBNG, the L2TP performs the hand-off task of the subscriber traffic to the Internet service provider (ISP). To do this, L2TP uses two network components:

- L2TP Access Concentrator (LAC)—The L2TP enables subscribers to dial into the L2TP access concentrator (LAC), which extends the PPP session to the LNS. cnBNG provides LAC.
- L2TP Network Server—The L2TP extends PPP sessions over an arbitary netowrk to a remote network server that is, the L2TP network server (LNS). The ISP provides LNS.

The overall network deployment architecture is also known as Virtual Private Dial up Network (VPDN). The overall topology of LAC and LNS is depicted as follows:



The CP for LAC and LNS depend on the L2TP session termination. Developing these two control planes in a single cnBNG microservice has the following benefits:

Simplified Single L2TP Control Plane

- Reduces the configuration complexity of the current XR L2TP

  vpdn-groups, vpdn-templates, l2tp-class and so on are simplified.
- Supports LC subscriber (not supported on the physical BNG)
- Avoids Ns/Nr checkpointing issues of pBNG to support RPFO

Collocated LAC and LNS

- Supports LAC and LNS in the same cnBNG CP, with different User Plane (UPs)
- Enables sharing of the same AAA and Policy Plane
- Simplifies management and troubleshooting

Flexible Deployment Options

The integration of LAC and LNS into a centralized cnBNG CP provides highly flexible deployments options to suit different customer use-cases and needs. For example, the cnBNG CP can host the CP functionality either for a LAC or LNS UP. Also, the same CP cluster can act as a CP for both LAC and LNS UPs from different data centers (or even from the same user-plane, if the user-plane supports it) except for the same session at the same time.

L2TP Features

The cnBNG supports the following Layer 2 Tunneling Protocol (L2TP) features:

- Tunnel authentication
- AVP encryption
- Tunnel Hello interval
- IP ToS value for tunneled traffic
- IPv4 don't fragment bit

- DSL line information attributes
- IPv4 tunnel source address
- IPv4 tunnel destination address
- IPv4 destination load balancing
- Tunnel mode
- MTU for LCP negotiation
- TCP maximum support
- Start-Control-Connection-Request (SCCRQ) timeout
- SCCRQ retries
- Control packet retransmission
- Control packet retransmission retries
- Receive window size for control channel
- Rx and Tx connect speed
- Tunnel VRF
- Tunnel session limit
- Weighed and Equal Loadbalancing
- Tunnel password for authentication
- Domain name and tunnel assignment
- LCP and Authentication renegotiation
- LAC hostnames for tunnelling requests
- Tunnel preference
- Tunnel-Client-Auth-ID support

L2TP tunnel failover support

An L2TP tunnel failover is a network feature that

- maintains subscriber connectivity by switching to a backup LNS server if the preferred LNS server is unavailable
- ensures minimal service interruption during failures, and
- enables seamless broadband service for end users.

Table 76: Feature History

Feature Name	Release Information	Description
L2TP tunnel failover support	2025.03.0	You can now maintain continuous broadband service for subscribers, even if the preferred LNS server becomes unavailable. The L2TP tunnel failover feature automatically switches subscriber sessions to a backup LNS server whenever a failure occurs, minimizing service disruption and ensuring seamless connectivity for end users.

Reliable connectivity with L2TP tunnel failover

The L2TP tunnel failover feature allows the cnBNG CP to provide reliable broadband connectivity. When a preferred L2TP Network Server (LNS) is not available, the system automatically selects the next available server, ensuring continuous service for subscribers.

How L2TP tunnel failover works

This process ensures that remote users always reach their private network, even if a failure occurs at the preferred LNS server.

Summary

The key components involved in the L2TP tunnel failover process are:

- PPPoE client: Initiates a dial-in session to the ISP access server.
- Network Access Server (NAS): Receives PPPoE connection from the client and forwards the session if needed.
- L2TP Access Concentrator (LAC): Establishes and manages L2TP tunnels to available LNS servers.
- L2TP Network Server (LNS): Authenticates and connects the subscriber to the private network.

Workflow

The process involves these stages:

1. The PPPoE client connects to the NAS, which determines if the session should be forwarded to an LNS via L2TP.
2. The LAC establishes L2TP tunnels to one or more LNS servers, maintaining a list of available servers.
3. When the preferred LNS server is unavailable (due to failure or lack of response), the system marks it as DOWN and starts a timer.
4. The LAC selects the next available LNS server from its list, using either Tunnel-Preference or Load-Balancing algorithms, to re-establish the subscriber session.
5. If the original LNS server becomes available again, it is marked as ACTIVE and may be used for new sessions.

Result

Subscribers experience uninterrupted broadband services as the system automatically redirects their sessions to available LNS servers during failures.

Configure tunnel busy timeout for LNS servers

Configure the amount of time that a failed LNS server remains marked as DOWN before being eligible for reconnection attempts.

Before you begin

- Ensure you have access privileges to configure cnBNG Control Plane.
- Know the profile name you wish to configure.

Follow these steps to configure the tunnel busy timeout:

Procedure

Step 1 Set the tunnel busy timeout value in seconds.

Example:

```
profile l2tp profile-name
  tunnel timeout busy value
exit
```

The system waits for the configured period before attempting to reuse an LNS server that was previously marked as DOWN.

This is a sample configuration.

```
profile l2tp prof1
  tunnel timeout busy 77
exit
```

Note We recommend setting the tunnel busy timeout to a value that balances quick failover with the possibility of LNS server recovery. A shorter timeout enables faster recovery if the LNS server returns quickly, while a longer timeout avoids repeated failed attempts during outages.

NOTES:

- **profile l2tp** *profile-name*: Specifies the PPPoE profile name and enters the Profile L2TP mode.
- **tunnel timeout busy** *value*: Specifies the timeout period before marking an LNS server as busy. The valid values range from 60 to 65535 seconds. Default value: 60 seconds.

Step 2 Use the **show l2tp-tunnel-destination upf** *upf_name* command to view the status (active or down) of LNS servers.

Example:

```
bng# show l2tp-tunnel-destination upf lps_asr9k-200
Wed Aug  13 10:57:06.504 UTC+00:00
result
[
  {
    "server": "192.69.1.199",
    "vrf": "",
    "status": "Active"
  }
]
```

Seamless L2TP session continuity during server reboot

Seamless L2TP session continuity during server reboot is a high-availability feature that

- allows ISPs to provide uninterrupted broadband services to subscribers during an L2TP server reboot
- automatically enables the secondary L2TP instance to manage existing sessions for affected User-Plane Functions (UPFs) until the primary instance is available, and
- restores session management to the primary instance once it returns to service.

Table 77: Feature history table

Feature name	Release information	Feature description
Seamless L2TP session continuity during server reboot	2026.01.0	You can now maintain uninterrupted broadband services for subscribers during an L2TP server reboot. If the primary server restarts, a secondary L2TP instance automatically manages active sessions for affected UPFs until the primary returns, ensuring session continuity and consistent service.

Supported platforms and deployment scenarios

This feature is supported in these cnBNG Control Plane deployments:

- All-in-One (AIO) server deployments (with or without geographical redundancy (GR) and subscriber redundancy group (SRG))
- Three-server and four-server deployments (with or without GR and SRG)
- Deployments using OpenShift

L2TP Pod instance limitations

- Only a single L2TP pod instance restart is supported within a specified time window.
- Double fault scenarios, where one L2TP instance is down due to a server reboot and the other L2TP instance restarts during that period, are not supported.
- Split-brain scenarios, where inter-pod connectivity is lost between L2TP pods but both pods remain reachable from the rest of the system, are not supported.

IETF Tagged Attributes on LAC

The IETF Tagged Attributes support on L2TP Access Concentrator (LAC) provides a means of grouping tunnel attributes referring to the same tunnel in an Access-Accept packet sent from the RADIUS server to the LAC. The Access-Accept packet can contain multiple instances of same RADIUS attributes, but with different tags. The tagged attributes support ensures that all attributes pertaining to a given tunnel contain the same value in their respective tag fields, and that each set includes an appropriately-valued instance of the Tunnel-Preference attribute. This conforms to the tunnel attributes that are to be used in a multi-vendor network environment, thereby eliminating interoperability issues among Network Access Servers (NASs) manufactured by different vendors.

For details of RADIUS Attributes for Tunnel Protocol Support, refer RFC 2868.

These examples describe the format of IETF Tagged Attributes:

```
Tunnel-Type = :0:L2TP, Tunnel-Medium-Type = :0:IP, Tunnel-Server-Endpoint = :0:"1.1.1.1",
Tunnel-Assignment-Id = :0:"1", Tunnel-Preference = :0:1, Tunnel-Password = :0:"hello"
```

A tag value of 0 is used in the above example in the format of `:0:`, to group those attributes in the same packet that refer to the same tunnel. Similar examples are:

```
Tunnel-Type = :1:L2TP, Tunnel-Medium-Type = :1:IP, Tunnel-Server-Endpoint = :1:"2.2.2.2",
Tunnel-Assignment-Id = :1:"1", Tunnel-Preference = :1:1, Tunnel-Password = :1:"hello"

Tunnel-Type = :2:L2TP, Tunnel-Medium-Type = :2:IP, Tunnel-Server-Endpoint = :2:"3.3.3.3",
Tunnel-Assignment-Id = :2:"1", Tunnel-Preference = :2:2, Tunnel-Password = :2:"hello"

Tunnel-Type = :3:L2TP, Tunnel-Medium-Type = :3:IP, Tunnel-Server-Endpoint = :3:"4.4.4.4",
Tunnel-Assignment-Id = :3:"1", Tunnel-Preference = :3:2, Tunnel-Password = :3:"hello"

Tunnel-Type = :4:L2TP, Tunnel-Medium-Type = :4:IP, Tunnel-Server-Endpoint = :4:"5.5.5.5",
Tunnel-Assignment-Id = :4:"1", Tunnel-Preference = :4:3, Tunnel-Password = :4:"hello"

Tunnel-Type = :5:L2TP, Tunnel-Medium-Type = :5:IP, Tunnel-Server-Endpoint = :5:"6.6.6.6",
Tunnel-Assignment-Id = :5:"1", Tunnel-Preference = :5:3, Tunnel-Password = :5:"hello"
```

Table 78: Supported IETF Tagged Attributes

IETF Tagged Attribute Name	Value	Type
Tunnel-Type	integer	64
Tunnel-Medium-Type	integer	65
Tunnel-Client-Endpoint	string	66
Tunnel-Server-Endpoint	string	67
Tunnel-Password	string	69
Tunnel-Assignment-ID	string	82
Tunnel-Preference	integer	83
Tunnel-Client-Auth-ID	string	90
Tunnel-Server-Auth-ID	string	91

Tunnel-Preference support

Tunnel-Preference is a RADIUS attribute that

- enables RADIUS servers to communicate tunnel preference selection information
- allows operators to prioritize or prefer certain tunnels over others when multiple tunnels are available for data transmission.

Table 79: Feature History

Feature Name	Release Information	Description
Tunnel-Preference support	2025.02.0	You can now prioritize specific tunnels for data transmission when multiple tunnels are available. The Tunnel-Preference attribute in RADIUS messages allows you to specify the preferred tunnel, offering greater flexibility in tunnel selection.

Tunnel-Preference attribute overview

Tunnel preference is communicated via the Tunnel-Preference attribute in RADIUS messages. This attribute indicates the preference that is given to a specific tunnel. The tunnel with the lowest numerical value in the **Value** field receives the highest preference, with 0x000000 being the most preferred and 0xFFFFFF being the least preferred.

Handling multiple tunneling attributes

If multiple sets of tunneling attributes are returned by the RADIUS server, the Tunnel-Preference attribute should be included in each set to indicate the relative preference assigned to each tunnel.

Attribute tagging and grouping

The Tunnel-Preference attribute includes a Tag field that groups attributes referring to the same tunnel.

Tunnel selection logic in L2TP with Tunnel-Preference and load balancing

In the L2TP tunnel pod, if a tunnel-preference value is received from the RADIUS server, it overrides any existing load-balancing configuration.

If multiple tunnels have the same tunnel-preference value, the system checks whether a load-balancing method, such as **weighted** or **equal**, is configured:

- If a load-balancing method is configured, the tunnel is selected based on the load-balancing method
- If no load-balancing method is configured, the system randomly selects a tunnel from those with the same preference value.

Tunnel-Preference attribute fields

Attribute name	Type	Length	Tag	Value
Tunnel-Preference	83	6	One octet in length. Valid values range from 0x01 to 0x1F. If the Tag field is not used, it must be set to zero (0x00).	Three octets in length. Higher preference is given to lower values, with 0x000000 being most preferred and 0xFFFFFF least preferred.

Benefits of Tunnel-Preference attribute

- Allows service providers to prioritize specific tunnels.
- Overrides load balancing configurations to ensure traffic flows through the preferred tunnel.
- Provides flexibility in tunnel selection based on RADIUS server directives.

Configure Tunnel-Preference attribute

You can configure the **Tunnel-Preference** attribute using the user-profile on the RADIUS server. You can then directly receive it from the RADIUS server. No configuration is needed on the cnBNG CP side.

Procedure

Define the **Tunnel-Preference** attribute on the RADIUS server.

Example:

```
RADIUS:
user@example.com    Password="abc"
        Service-Type = Outbound-User,
        Tunnel-Type = :1:L2TP,
        Tunnel-Medium-Type = :1:IP,
        Tunnel-Client-Endpoint = ":1:10.0.0.201",
        Tunnel-Server-Endpoint = ":1:10.1.10.129",
        Tunnel-Client-Auth-Id = ":1:LAC1",
        Tunnel-Server-Auth-Id = ":1:LNS1",
        Tunnel-Assignment-Id = ":1:one",
        Tunnel-Password = ":1:cisco1",
        Tunnel-Preference = :1:100
        Tunnel-Type = :2:L2TP,
        Tunnel-Medium-Type = :2:IP,
        Tunnel-Client-Endpoint = ":2:10.0.0.201",
        Tunnel-Server-Endpoint = ":2:10.1.10.130",
        Tunnel-Client-Auth-Id = ":2:LAC2",
        Tunnel-Server-Auth-Id = ":2:LNS2",
        Tunnel-Assignment-Id = ":2:two",
```

```
Tunnel-Password = ":2:cisco2",
Tunnel-Preference = :2:200
```

Tunnel-Client-Auth-Id support

Tunnel-Client-Auth-ID is a RADIUS attribute that

- specifies the name used by the tunnel initiator during the authentication phase of tunnel establishment
- is defined in RFC 2868 - *RADIUS Attributes for Tunnel Protocol Support.*

Table 80: Feature History

Feature Name	Release Information	Description
Tunnel-Client-Auth-ID support	2025.02.0	This attribute supports tunneling protocols by specifying the authentication name used by the tunnel inititator during the authentication phase of tunnel establishment.

Tunnel-Client-Auth-ID attribute fields

Attribute	Type	Length	Tag	String
Tunnel-Client-Auth-ID	90 - The identifier for Tunnel-Client-Auth-ID.	Must be greater than or equal to 3 bytes.	One octet in length; groups attributes for the same tunnel: Values between 0x01 and 0x1F indicate the specific tunnel, while values above 0x1F are interpreted as the first byte of the String field.	Mandatory field containing the authentication name of the tunnel initiator, represented in UTF-8 charset.

Example of Tunnel-Client-Auth-ID attribute in a RADIUS message

In a typical **Access-Request** or **Access-Accept** message, the Tunnel-Client-Auth-Id is included in the attributes field, for example,

- `Tunnel-Client-Auth-Id: " :1:CSCO_LAC"`

  Here, " CSCO_LAC" is the unique identifier for a client inside the tunnel.

Configure Tunnel-Client-Auth-ID attribute

You can configure the **Tunnel-Client-Auth-Id** attribute using the user-profile on the RADIUS server. You can then directly receive it from the RADIUS server. No configuration is needed on the cnBNG CP side.

Procedure

Step 1 Define the **Tunnel-Client-Auth-Id** attribute on the RADIUS server.

Example:

```
RADIUS:
user@example.com    Password="abc"
        Service-Type = Outbound-User,
        Tunnel-Type = :1:L2TP,
        Tunnel-Medium-Type = :1:IP,
        Tunnel-Client-Endpoint = ":1:10.0.0.201",
        Tunnel-Server-Endpoint = ":1:10.1.10.129",
        Tunnel-Client-Auth-Id = ":1:LAC1",
        Tunnel-Server-Auth-Id = ":1:LNS1",
        Tunnel-Assignment-Id = ":1:one",
        Tunnel-Password = ":1:cisco1",
        Tunnel-Preference = :1:100
        Tunnel-Type = :2:L2TP,
        Tunnel-Medium-Type = :2:IP,
        Tunnel-Client-Endpoint = ":2:10.0.0.201",
        Tunnel-Server-Endpoint = ":2:10.1.10.130",
        Tunnel-Client-Auth-Id = ":2:LAC2",
        Tunnel-Server-Auth-Id = ":2:LNS2",
        Tunnel-Assignment-Id = ":2:two",
        Tunnel-Password = ":2:cisco2",
        Tunnel-Preference = :2:200
```

Step 2 Use the **show l2tp-tunnel detail** command to view the configured attributes.

Example:

```
bng# show l2tp-tunnel detail
tunnel-details
{
  "tunResponses": [
    {
      "state": "established",
      "profileName": "l2tp-prof1",
      "tunnelType": "lac",
      "sessionCount": 10,
      "IDs Allocated": 10,
      "routerID": "asr9k-1",
      "srcIP": "10.0.0.201",
      "dstIP": "10.1.10.129",
      "tunnelAssignmentID": "one",
      "localTunnelID": 49078,
      "remoteTunnelID": 51810,
      "tunnelClientAuthID": "LAC1",
      "tunnelServerAuthID": "bng-lns"
    }
  ]
}
```

How it Works

This section provides a brief of how the L2TP Subscriber Management feature works.

L2TP Handling

Both LAC and LNS sessions use L2TP protocol for negotiation and creation of L2TP sessions. However for LAC sessions, there is additional PPPoE handling. This section focuses on the L2TP protocol handling.

LAC Sessions

For LAC sessions, the PPP sessions are terminated on a different network node from where the PPPoE sessions are terminated. The PPPoE sessions are terminated on the LAC, but the PPP session is terminated on an LNS upstream, over an L2TP tunnel. Initial PPP negotiations are done on the LAC to determine the appropriate LNS to tunnel the session. When the tunnel has been established, all PPP handling is handed off to the LNS.

- The PPPoE protocol is negotiated in the same way as a PTA session.
- PPPoE service handles all PPPoE packets and the nitial LCP and authorization packets.
- After authentication , if the user-profile contains service=outbound, PPPoE service decides to tunnel the sessions.
- It reaches out the L2TP pod to initiate a L2TP tunnel . The L2TP tunnel pod creates the tunnel and returns the L2TP session ID.
- The PPPoE service continues to handle the L2TP session FSM and bring-up the LAC session and program the UP via the Subscriber Manager.

LNS Sessions

LNS sessions are similar to PTA sessions in the overall functionality. Instead of PPPoE protocol, here the First-Sign-Of-Life (FSOL) packets are the L2TP Incoming-Call-Request (ICRQ) messages. When the L2TP session protocol is up, then the existing PPP protocol finite state machines (FSM) is triggered to bring up and program the session on the UP.

- L2TP Tunnel pod receives tunnel-create request from the remote LAC.
- After Tunnel is up, PPPoE Pod receives ICRQ to create a session.
- PPPoE pod communicates with the L2TP to get L2TP session-id for the given tunnel ID.
- L2TP generates the session ID and checks the session count.
- PPPoE pod checks if there is forced renegotiation configured for the session. Else, it proceeds with the session programming to the UP.

AAA Attributes for L2TP

The following is the list of AAA attributes for L2TP LAC and LNS sessions.

IETF Attribute: AAA_TUNNEL_PASSWORD (69)

Tunnel-Password=<16byte-encrypted-value>

The value of this attribute is defined as an "encrypted-string". RADIUS decrypts the value and sends a plain-text password to the Subscriber Manager (SM).

For more L2TP IETF Attributes, see IETF Tagged Attributes on LAC, on page 350.

CISCO-VSA: AAA_AT_L2TP_TUNNEL_PASSWORD

Cisco-AVPair += "l2tp-tunnel-password=<plain-text>"

The value of this attribute is defined in "plain-text". RADIUS passes the value to SM in the respective Access-accept request.

If required, the RADIUS server can this as an "encrypted-cisco-visa(36)", which is similar to the Layer1 vendor-specific attributes (VSAs).

In that case, RADIUS-Ep decrypts the complete VSA and sends the plain-text value.

For more L2TP VSA attributes, see RADIUS Vendor-Specific Attributes, on page 603.

Handling L2TP Sessions during CP-GR Switchover

This feature enables you to manage Layer 2 Tunneling Protocol (L2TP) sessions and tunnels effectively during a CP-GR switchover event. This ensures seamless transition and minimal disruption in connectivity for users.

Table 81: Feature History

Feature Name	Release Information	Description
Handling L2TP Sessions during CP-GR Switchover	2025.02.0	This feature minimizes downtime and ensures smooth transitions during network changes by disconnecting existing L2TP sessions and tunnels during a CP-GR switchover. It then establishes new sessions on new tunnels.

Here's how L2TP sessions are handled during CP-GR switchover:

1. **Disconnection of existing sessions and tunnels:**

   When a CP-GR switchover gets triggered for a specific instance-id, all L2TP sessions and the corresponding L2TP tunnels on UPFs associated with that instance-id are disconnected.

2. **Establishment of new sessions and tunnels:**

   Following the CP-GR switchover, new L2TP sessions are established on new tunnels.

Call Flows

This section includes the following high-level call flows.

LAC Session Bringup Call Flow

The LAC Session Bringup call flow is as follows.

Figure 20: LAC Session Bringup Call Flow

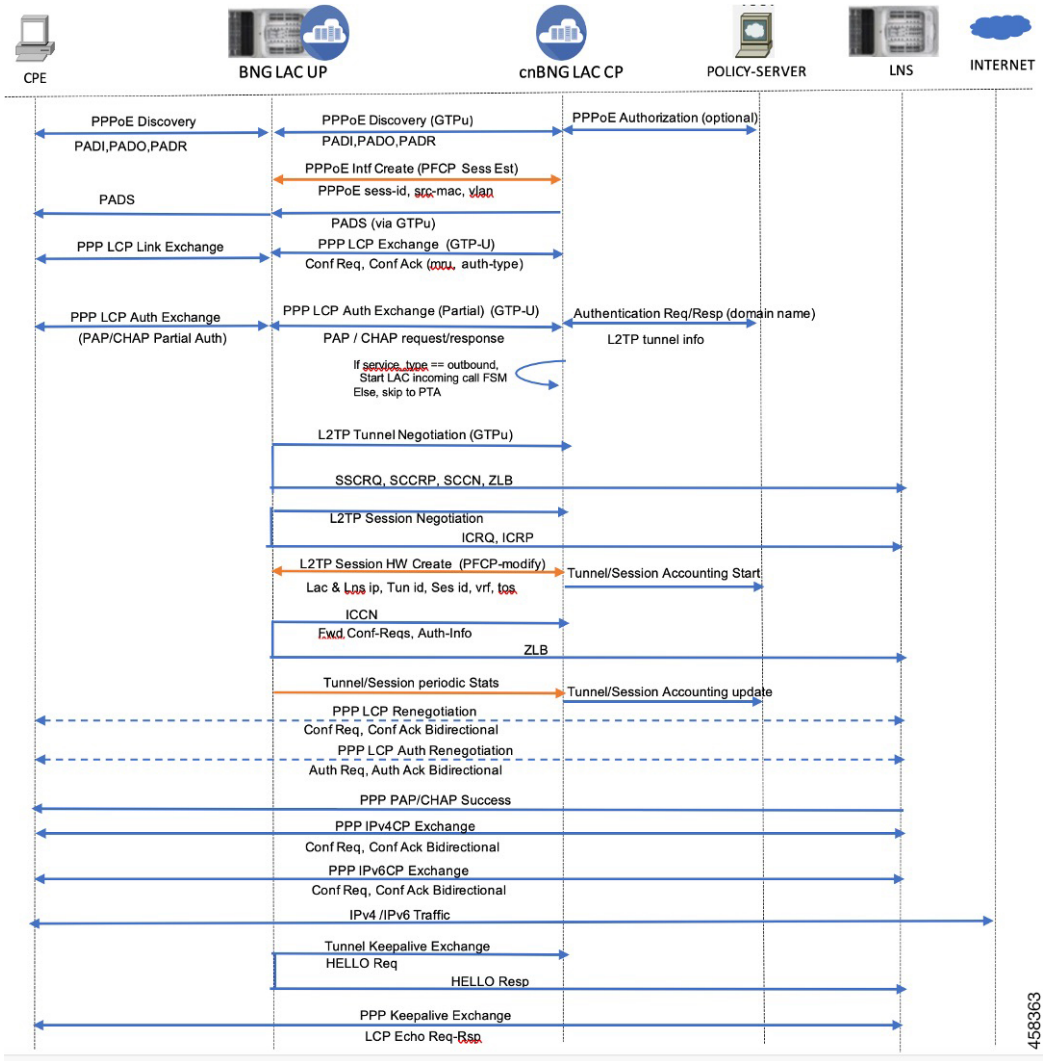


Table 82: LAC Session Bringup Call Flow Description

Steps	Description
1	On learning the first control packet, the BNG-CP sends a Session Creation request to create a new packet forwarding state for the data packet. This updates the BNG-UP state. **Note** At this step, it is possible to create a session from the redirected control packet. By doing so, resources are consumed on the BNG-UP to allow individual subscriber control packet management such as blocking, rate limiting, and specific packet filtering. It is also possible to postpone the session creation. By doing so, additional resources BNG-UP are not consumed, but individual subscriber control packet management is not possible.

Steps	Description
2	The BNG-UP sends the following response to the BNG-CP: • Informs that the states are installed. • Informs that it (BNG-UP) is ready to forward the subscriber PPP control packets.
3	The BNG-CP sends the PADO message back to the CPE through the BNG-UP using the control packet redirect interface.
4	The PADR message is sent from the CPE through the BNG-UP using the control packet redirect interface.
5	The BNG-CP sends the PADS message back to the CPE through the BNG-UP using the control packet redirect interface.
6	The LCP configuration request is sent from the CPE through the BNG-UP using the control packet redirect interface.
7	The BNG-CP sends the LCP configuration acknowledgement back to the CPE through the BNG-UP using the control packet redirect interface. The LCP configuration acknowledgement indicates either a PAP or CHAP authentication challenge. Options: • Option 1: If the client chooses PAP, the CPE sends a PAP request to the BNG-CP through the BNG-UP using the control packet redirect interface. The PAP password is sent as an Access request to the AAA server. • Option 2: If CHAP is required, the BNG-CP initiates a challenge to the CPE though the BNG-UP using the control packet redirect interface. The CPE responds back to the challenge to the BNG-CP. The challenge is sent to the AAA server.
8	The AAA successfully authenticates the CPE and replies to the CPE with a PAP/CHAP success and that this is a L2TP session. **Note** If the RADIUS profile received in AAA Accept-Ack has the field “service-type” with the value as “outbound-user”, this means that the session must be tunneled to the LNS IP address (either specified in the same profile or available in the Control Plane configuration).
9	The BNG-CP sends a Session Establishment message to the BNG-UP. The BNG-CP programs the NBG-UP control packet redirect rules to do the following: • Decapsulate and send the L2TP control message towards the LNS. • Redirect L2TP control message back to the BNG-CP. This session establishment is only on a per-tunnel basis.
10	The BNG-UP sends the following response to the BNG-CP: • Informs that the states are installed. • Informs that it (BNG-UP) is ready to forward the L2TP control packets.

Steps	Description
11	The BNG-CP sends Start-Control-Connection-Request (SCCRQ), Start-Control-Connection-Reply (SCCRP), Start-Control-Connection-Connected (SCCCN), and Zero-Length Body (ZLB) to the LNS via the BNG-UP through the control packet redirect interface.
12	The BNG-CP sends Incoming-Call-Request (ICRQ), Incoming-Call-Reply (ICRP), Incoming-Call-Connected (ICCN), and ZLB to the LNS via the BNG-UP through the control packet redirect interface.
13	The BNG-CP sends a Session Modify request if there is a previous session established to allow for data packet forwarding to the LNS (and control packet if not done already). If a previous session was not established, this is a Session Request message to allow for data packet forwarding to the LNS. This updates the User Plane state. **Note** Subscriber session creation can be performed at any steps prior to this. This step is the last chance for a session creation to avoid subscriber data packets drops. Immediately after this step, the CPE is assigned an address and data packets would be sent immediately.
14	The BNG-UP sends the following response to the BNG-CP: • Informs that the states are installed. • Informs that it (BNG-UP) is ready to forward subscribers PPP control and data packets.
15	If the LNS has cached the LCP configuration and there is no negotiation disagreement, this step can be skipped. If LNS has not cached the LCP configuration or the session requires renegotiation, then the LCP negotiation takes place.
16	If the LNS has cached the authentication information and there is no disagreement on authentication, this step can be skipped. If LCP has not cached the authentication information or authentication has failed, then reauthorization occurs.
17	The IP Control Protocol (IPCP) takes place between the CPE and the LNS through the BNG-UP.
18	The PPP LCP echo hello are exchanged between the CPE and the LNS through the BNG-UP.

LAC Session Bringdown Call Flow

The LAC Session Bringdown call flow is as follows.

Figure 21: LAC Session Bringdown Call Flow

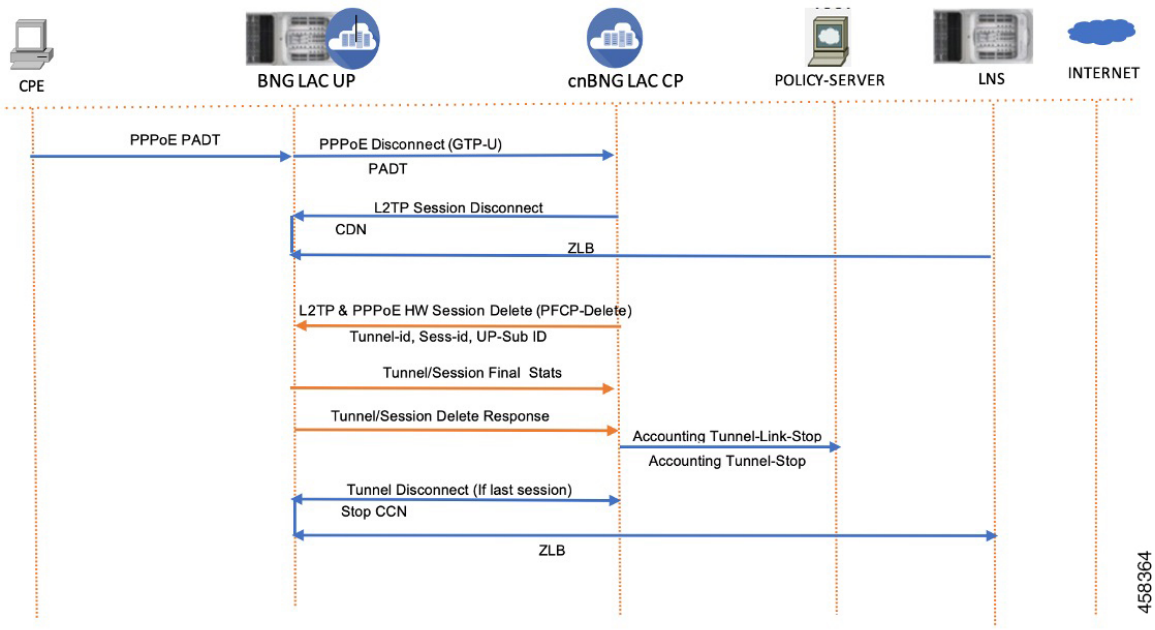


Table 83: LAC Session Bringdown Call Flow Description

Steps	Description
1	The L2TP (LAC) session and tunnel bringdown can occur due to various reasons. For example, CPE can send PADT to gracefully bringdown the subscriber session. This triggers the L2TP session cleanup between LAC and LNS.
2	If it is the last session in the L2TP tunnel, the tunnel is also deleted and the PPPoE session is cleaned up in the LAC. The session or tunnel bringdown occurs in the following scenarios: • PPP keepalive failure between CPE and LNS. • Tunnel keepalive failure. In this case, all sessions in the tunnel are removed first. • Admin clear on either LAC or LNS.

LNS Session Bringup Call Flow

The LNS Session Bringup call flow is as follows.

Figure 22: LNS Session Bringup Call Flow

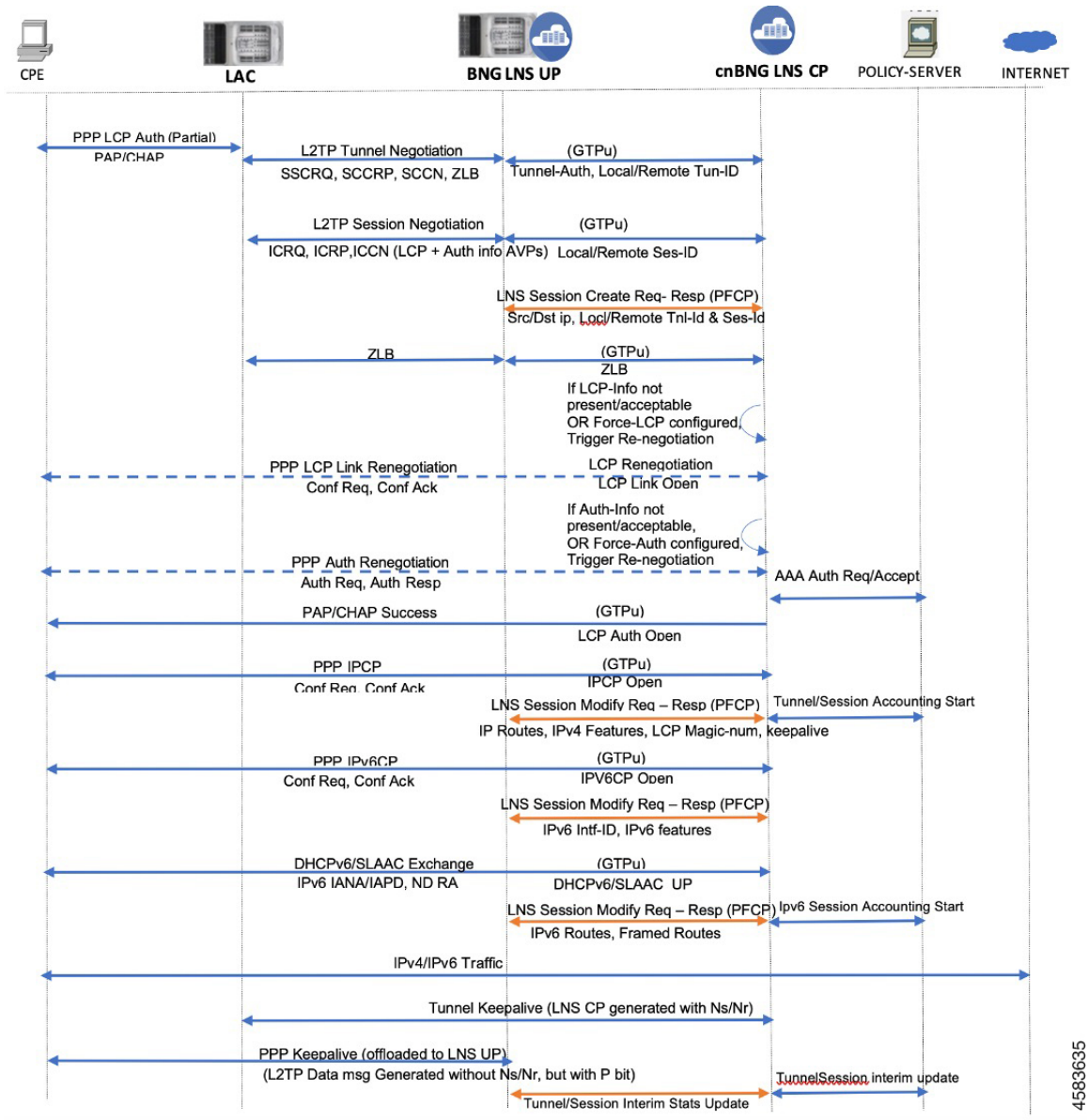


Table 84: LNS Session Bringup Call Flow Description

Steps	Description
1	The Start Control Connection Request (SCCRQ) message is received through the control packet redirect interface following the common packet redirect rule.
2	The BNG-CP sends a Session Establishment request message to the BNG-UP. The BNG-CP programs the DBNG-UP control packet redirect rules to send L2TP control message towards the BNG-CP to only accept particular tunnels.

Steps	Description
3	The BNG-UP sends the following response back to the BNG-CP: • Informs that the states are installed. • Informs that it (BNG-UP) is ready to forward the L2TP control packets.
4	The BNG-CP exchanges Start Control Connection Reply (SCCRP), Start Control Connection Connected (SCCCN), and Zero Length Body (ZLB) with the LAC using the control packet redirect interface.
5	The BNG-CP receives the Incoming Call Request (ICRQ) message (includes AVP defined in RFC 5515).
6	After receiving the ICRQ message, the BNG-CP has the L2TP session ID information. The BNG-CP can send a Session Establishment request to the BNG-UP to ensure only known L2TP sessions are accepted. 1 **Note** At this step, it is possible to create a session from the redirected control packet. By doing so, resources are consumed on the DBNG-UP in order to allow individual subscriber control packet management such as blocking, rate limiting, and specific packet filtering. It is also possible to postpone the session creation. By doing so, additional resources DBNG-UP are not consumed, but individual subscriber control packet management is not possible
7	The BNG-UP sends the following response back to the BNG-CP: • Informs that the states are installed. • Informs that it (BNG-UP) only accepts L2TP control packet from known sessions.
8	The BNG-CP exchanges ICRP, ICCN, and ZLB with the LAC using the control packet redirect interface.
9	If the LNS has cached the LCP configuration and there is no negotiation disagreement, this step can be skipped. If the LCP has not cached the LCP configuration or the session requires renegotiation, then the LCP negotiation takes place.
10	If the LNS has cached the authentication information and there is no disagreement on authentication, this step can be skipped. If LCP has not cached the authentication information or authentication has failed, then reauthorization occurs.

Steps	Description
11	After authentication, the BNG-CP knows the IP address or prefix (or both) for the subscriber either though the local address server or from the AAA returned VSAs. The BNG-CP sends a Session Modify request if there is already an established session to update the User Plane (UP) state. If there are no prior sessions, this requires a Session Establishment request to update the UP. **Note** Subscriber session creation can be performed at any steps prior to this. This step is the last chance for a session creation to avoid subscriber data packets drops. Immediately after this step, the CPE is assigned an address and data packets would be sent immediately.
12	The BNG-UP sends the following response back to the BNG-CP: • Informs that the states are installed. • Informs that it (BNG-UP) is ready to forward subscribers PPP control and data packets.
13	The IPCP takes place between the CPE and the LNS through the BNG-UP.
14	The PPP LCP echo hello are exchanged between the CPE and the LNS through the BNG-UP.

[1] At this step, it is possible to create a session from the redirected control packet. By doing so, resources are consumed on the BNG-UP to allow individual subscriber control packet management such as blocking, rate limiting, and specific packet filtering. It is also possible to postpone the session creation. By doing so, additional resources BNG-UP are not consumed, but individual subscriber control packet management is not possible.

LNS Session Bringdown Call Flow

The LNS Session Bringdown call flow is as follows.

Figure 23: LNS Session Bringdown Call Flow

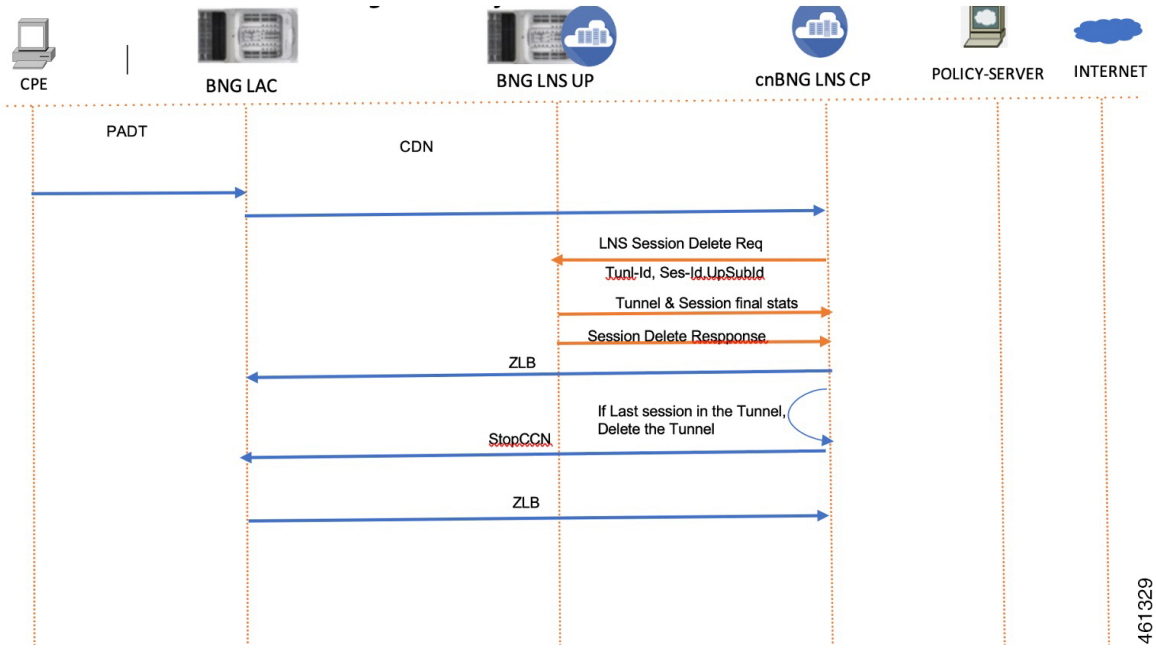


Table 85: LNS Session Bringdown Call Flow Description

Steps	Description
1	LAC sends a Call-Disconnect-Notify (CDN) message to release the session on the LNS.
2	cnBNG CP deletes the session on UP. It releases all the resources and collects the final statistics from the UP and sends the Accounting-Stop message.
	cnBNG CP sends ZLB as ackndoledgement.
	If it is the last sesion on the tunnel, cnBNG CP sends a Stop-Control-Connection-Notification (Stop-CCN) message to bring down the tunnel.

Standard Compliance

The L2TP Subscriber Management feature is aligned with the following standard:

- RFC 2661: Layer Two Tunneling Protocol "L2TP"

Limitations

The LT2P Subscriber Management feature has the following limitations:

- On-the-fly changes to L2TP profile is not supported.
- L2TP attributes should be configured only for session-activate event.

- Tunnel load balancing with Tunnel-Assignment-ID is not supported.
- Weighted Tunnel load balancing can be configured only in the profile.
- The TCP maximum segment size (TCP-MSS) is supported at the global User Plane Function (UPF) chassis level and not at the tunnel or session level. It must be configured on the ASR 9000 UPF.

Configuring the L2TP Subscriber Management Feature

This section describes how to configure the L2TP Subscriber Management feature.

Configuring the L2TP Subscriber Management feature involves the following step:

Creating the L2TP profile

Creating the L2TP Profile

Use the following commands to create the Layer2 Tunnelling Protocol (L2TP) profile and provide the L2TP specific parameters.

```
config
  profile l2tp l2tp_profile_name
    authentication

    encrypt-avp
    hello-interval interval_in_seconds
    hostname local_hostname
    ip-tos { ip_tos_value | reflect }
    ipv4 { df-bit { reflect | set } | source ip_address }
    mode lac
      domain domain_name [ tun-assign-id tunnel_id ]
      dsl-info-forwarding
      ipv4 { destination ip_address | df-bit { reflect | set } |
          source ip_address }
      rx-connect-speed kbps
      tunnel-load-balancing { equal | weighted }
      tx-connect-speed kbps
    mode lns
      force-lcp-renegotiation
      mtu mtu_value
      terminate-from remote_hostname
    password password
    receive-window number_of_packets
    retransmit { retries number_of_retries |
    timeout { max max_timeout | min min_timeout }
    tcp adjust-mss mss_value
    tunnel { session-limit number_of_sessions |
    timeout { no-session timeout_value | busy value }
    vrf vrf_name
    exit
```

NOTES:

- **profile l2tp** *l2tp_profile_name*: Specifies the PPPoE profile name and enters the Profile L2TP mode.
- **authentication**: Enables L2TP tunnel authentication.
- **congestion-control**:Enables L2TP congestion control.
- **encrypt-avp**: Hides attribute-value pair (AVPs) in outgoing control messages.
- **hello-interval** *interval_in_seconds*: Sets the hello interval in seconds. The valid values range from 10 to 1000 seconds.
- **hostname** *local_hostname*: Specifies the local hostname of the tunnel. The valid value is an alphanumeric string ranging from 1 to 256. The name of the Control Plane (CP) is the default local hostname.
- **ip-tos {** *ip_tos_value* **| reflect }**: Sets the IP Type of Service (ToS) value for tunneled traffic. The ToS valid values range from 1 to 255. The control packets use 0xC0 as the default value.
- **ipv4 { destination** *ip_address* **| df-bit { reflect | set } | source** *ip_address* **}**: Specifies the IPv4 settings for the tunnel:
  - **df-bit { reflect | set }**: Specifies the IPv4 Don't Fragment (DF) bit.

    reflect: Reflects the DF bit from the specified inner IP address.

    set: Sets the DF bit.
  - **source** *ip_address*: Specifies the source IP address of the tunnel.
- **mode { lac | lns }**: Configures LAC or LNS.
  - **mode lac { domain** *domain_name* **[tun-assign-id** *tunnel_id* **] | dsl-info-forwarding | ipv4 { destination** *ip_address* **| df-bit { reflect | set } | source** *ip_address* **} | rx-connect-speed** *kbps* **| tunnel-load-balancing { equal | weighted } | tx-connect-speed** *kbps*: Configures a L2TP Access Concentrator (LAC) to request the establishment of an L2TP tunnel to an L2TP Network Server (LNS).
    - **domain** *domain_name* **[tun-assign-id** *tunnel_id* **]**: Specifies the domain name to match. The valid values range from 1 to 255. The control packets use 0xC0 as the default value.

      tun-assign-id *tunnel_id*: Specifies the domain name with a tunnel ID.
    - **dsl-info-forwarding**: Forwards DSL line information attributes.
    - **ipv4 { destination** *ip_address* **| df-bit { reflect | set } | source** *ip_address* **}**: Specifies the IPv4 settings for the tunnel:
      - **destination** *ip_address*: Specifies the destination IP address of the tunnel.
      - **df-bit { reflect | set }**: Specifies the IPv4 Don't Fragment (DF) bit.

        reflect: Reflects the DF bit from the specified inner IP address.

        set: Sets the DF bit.
      - **source** *ip_address*: Specifies the source IP address of the tunnel.
    - **rx-connect-speed** *kbps*: Specifies the receiving (Rx) connection speed in kbps. The valid values range from 9 to 100000000 kbps.

- **tunnel-load-balancing { equal | weighted }** : Specifies equal or weighted load sharing of the tunnel.
- **tx-connect-speed** *kbps*: Specifies the transmitting (Tx) connection speed in kbps. The valid values range from 9 to 100000000 kbps.

- **mode lns { force-lcp-renegotiation | mtu | terminate-from** *remote_hostname*: Configures a LNS to accept requests from LAC to establish L2TP tunnel:
  - **force-lcp-renegotiation**: Forces Link Control Protocol (LCP) and Authorisation renegotiation.

    mtu *mtu_value*: Specifies the MTU for LCP negotiation. The *mtu_value* valid values range from 500 to 2000. The default value is 1492.
  - **terminate-from** *remote_hostname*: Specifies the hostname of the remote peer to accept tunnels.

- **password** *password*: Specifies the password for tunnel authentication.
- **receive-window** *number_of_packets*: Specifies the receive window size for the tunnel. The valid values range from 1 to 5000 packets. The default value is 4.
- **retransmit { retries** *number_of_retries* **| timeout { max** *max_timeout* **| min** *min_timeout* **}**: Specifies the control message retransmission parameters.
  - **retries** *number_of_retries*: Specifies the maximum number of retries for control packets.
  - **timeout { max** *max_timeout* **| min** *min_timeout* **}**: Specifies the control packet retransmission timeout parameters.

    max *max_timeout*: Specifies the control packet retransmission maximum timeout parameters. The valid values range from 1 to 8 seconds. The default value is 8.

    min *min_timeout*: Specifies the control packet retransmission minimum timeout parameters. The valid values range from 1 to 8 seconds. The default value is 1.

- **tcp adjust-mss** *mss_value*: Adjusts the TCP Maximum Segment Size (MSS) value of TCP SYN (synchronize) packets. The valid values range from 500 to 1500 packets.
- **tunnel { session-limit** *number_of_sessions* **| timeout { no-session** *timeout_value* **| busy** *value* **}**: Limits the sessions for a tunnel or deletes the tunnel after timeout
  - **session-limit** *number_of_sessions*: Specifies the maximum number of L2TP sessions per tunnel. The valid values range from 1 to 64000 sessions.
  - **timeout { no-session** *timeout_value* **}**: Specifies the following parameters :

    timeout no-session: No-session timeout for the tunnel. The default value is 0 seconds.

    timeout *timeout_value*: Timeout value in seconds. The valid values range from 1 to 86400 seconds.
  - **timeout busy** *value*: Specifies the timeout period before marking an LNS server as busy. The valid values range from 60 to 65535 seconds.

- **vrf** *vrf_name*: Specifies the Virtual routing and forwarding (VRF) name of the tunnel.

Monitoring and troubleshooting

This section describes key operational commands available for diagnosing tunnel states and analyzing tunnel failures.

View L2TP tunnel instance details

Display detailed information about a specific L2TP tunnel instance for troubleshooting or monitoring.

Use this CLI command to inspect the operational state and key parameters of an L2TP tunnel. This is especially useful for diagnosing tunnel connectivity, performance, or configuration issues.

Procedure

Step 1 Use the **show l2tp-tunnel-instance upf** *upf-name* **tunnel-id** *value* command to display the specific L2TP tunnel instance details.

Example:

```
show l2tp-tunnel-instance upf asr9k-lac tunnel-id 13176

Tue Oct  21 11:38:08.584 UTC+00:00
result
[
  {
    "Tunnel-ID": 13176,
    "Remote-Tunnel-ID": 8858,
    "Tunnel-Type": "LAC",
    "UP": "asr9k-1",
    "FSM-state": "established",
    "Src-Endpoint": "10.1.39.139",
    "Dst-Endpoint": "10.1.34.52",
    "Group-ID": "TWO",
    "Control-NS": 4,
    "Control-NR": 2,
    "Remote-RWS": 512,
    "Disconnect-state": "None",
    "Clear-sessions": "None",
    "Delete-state": "None",
    "Recovery-state": "Initial"
  }
]
```

Step 2 (Optional) Use the **show l2tp-tunnel-instance upf** *upf-name* command to display detailed information about active L2TP tunnel instances associated with the UPF.

Example:

```
show l2tp-tunnel-instance upf asr9k-1
Tue Oct  21 11:36:54.595 UTC+00:00
result
[
  {
    "Tunnel-ID": 13176,
    "Remote-Tunnel-ID": 8858,
    "Tunnel-Type": "LAC",
    "UP": "asr9k-1",
```

```
    "FSM-state": "established",
    "Src-Endpoint": "10.1.39.139",
    "Dst-Endpoint": "10.1.34.52",
    "Group-ID": "TWO",
    "Control-NS": 4,
    "Control-NR": 2,
    "Remote-RWS": 512,
    "Disconnect-state": "None",
    "Clear-sessions": "None",
    "Delete-state": "None",
    "Recovery-state": "Initial"
  },
  {
    "Tunnel-ID": 1391,
    "Remote-Tunnel-ID": 8859,
    "Tunnel-Type": "LAC",
    "UP": "asr9k-1",
    "FSM-state": "established",
    "Src-Endpoint": "10.1.39.139",
    "Dst-Endpoint": "10.1.34.52",
    "Group-ID": "ONE",
    "Control-NS": 6,
    "Control-NR": 3,
    "Remote-RWS": 512,
    "Disconnect-state": "None",
    "Clear-sessions": "None",
    "Delete-state": "None",
    "Recovery-state": "Initial"
  }
]
```


View detailed LNS session information using the sublabel identifier

Enable network administrators to retrieve and analyze comprehensive debug-level information for a specific LNS subscriber session using the session's sublabel.

Procedure

Use the **show subscriber lns sublabel** *sublabel* **debug** command to view detailed debug information for a specific LNS subscriber session.

Example:

```
bng# show subscriber lns sublabel 16777219 debug

Mon Oct  20 06:30:27.240 UTC+00:00
subscriber-details
{
  "subResponses": [
    {
      "state": "complete",
      "key": {
        "routerID": "asr9k-1",
        "portID": "GigabitEthernet0/1/0/1",
        "subLabel": "16777218",
        "upSubID": "1"
      },
```

```
      "flags": [
        "SM_START_DONE",
        "SM_ACTIVATE_DONE",
        "SM_UPDATE_DONE",
        "IPCP_UP",
        "IPV6CP_UP"
      ],
      "lcpInfo": {
        "state": "opened",
        "keepAliveInterval": 60,
        "keepAliveRetries": 5,
        "localMru": 1492,
        "peerMru": 1492,
        "localMagic": "0xe2ab84f",
        "peerMagic": "0xe2ab850",
        "authOption": "PAP",
        "authCompleted": true,
        "username": "cnbng"
      },
      "ipcpInfo": {
        "state": "opened",
        "peerIpv4Pool": "pool-ISP",
        "peerIpv4Address": "11.0.32.2",
        "peerIpv4Netmask": 22,
        "localIpv4Address": "11.0.32.1",
        "isIpamPoolIPaddr": true
      },
      "ipv6cpInfo": {
        "state": "opened",
        "localIntfID": "0x1",
        "peerIntfID": "0xcc11000000010001"
      },
      "lnsInfo": {
        "srcIP": "10.1.39.139",
        "dstIP": "10.1.34.52",
        "state": "established",
        "profileName": "l2tp-prof2",
        "tunnelClientAuthID": "bng-lac",
        "tunnelServerAuthID": "Local-DC",
        "callSerialNumber": 16777408,
        "localTunnelID": 12226,
        "localSessionID": 11428,
        "remoteTunnelID": 1017,
        "remoteSessionID": 24885
      },
      "sessionType": "lns",
      "vrf": "default",
      "AuditId": 4,
      "sess-events": [
        "Time, Event, Status",
        "2025-10-20 06:30:01.966696374 +0000 UTC, Sublabel Alloc, complete",
        "2025-10-20 06:30:01.971563927 +0000 UTC, L2TP-ID Alloc, complete",
        "2025-10-20 06:30:01.986624734 +0000 UTC, UP Interface Create, complete",
        "2025-10-20 06:30:02.010855854 +0000 UTC, SM Session-Start, complete",
        "2025-10-20 06:30:02.038466759 +0000 UTC, L2TP session, complete",
        "2025-10-20 06:30:02.038504381 +0000 UTC, LCP, up",
        "2025-10-20 06:30:02.07193459 +0000 UTC, SM Session-Activate, complete",
        "2025-10-20 06:30:02.102904338 +0000 UTC, IPv6CP, up",
        "2025-10-20 06:30:03.202555529 +0000 UTC, IPCP, up"
      ]
    }
  ]
}
```

This information helps network administrators verify session parameters and troubleshoot LNS connectivity issues.

View L2TP tunnel disconnect history

Retrieve the recent disconnect history for L2TP tunnels on a specified UPF.

Use this task to investigate tunnel stability, common disconnect reasons, or repeated failures.

Procedure

Step 1 Use the **show disconnect-history l2tp-tunnel upf** *upf-name* command to display all recent disconnects for a UPF.

Example:

```
bng# show l2tp-tunnel disconnect-history upf asr9k-lns

Wed Jan  21 04:57:05.678 UTC+00:00
-------------------------------------------------------------------------------
[Disconnect Reason]                    [Last Disconnect Time]       [Tunnel-id]
-------------------------------------------------------------------------------
HELLORetransFailure                    2026/01/21 04:54:22.406      5951
-------------------------------------------------------------------------------
```

Step 2 Use the **show disconnect-history l2tp-tunnel upf** *upf-name* **unique** command to display disconnect records summarized by disconnect reason. For each reason, it shows the disconnect time and tunnel ID of the most recent event, along with a count of how many disconnects occurred for that reason.

Example:

```
bng# show disconnect-history l2tp-tunnel upf asr9k-lns unique

Wed Jan  21 04:56:32.195 UTC+00:00
-----------------------------------------------------------------------------------------------
[Disconnect Reason]                [Last Disconnect Time]       [Tunnel-id]      [Count]
-----------------------------------------------------------------------------------------------
HELLORetransFailure                2026/01/21 04:54:22.406      5951             1
L2TP Admin Delete                  2026/01/21 04:46:17.713      34975            2
No-Session Timeout                 2026/01/21 04:45:32.125      2028             1
-----------------------------------------------------------------------------------------------
```

Step 3 Optionally, use the **show disconnect-history l2tp-tunnel upf** *upf-name* **last** command to display up to the last 1000 disconnect records.

Example:

```
bng# show disconnect-history l2tp-tunnel upf asr9k-lns last

Wed Jan  21 04:56:56.831 UTC+00:00
-----------------------------------------------------------------------------------------------
[Disconnect Reason]                    [Last Disconnect Time]       [Tunnel-id]      [Count]
-----------------------------------------------------------------------------------------------
HELLORetransFailure                    2026/01/21 04:54:22.406      5951             1
L2TP Admin Delete                      2026/01/21 04:46:17.713      34975            1
No-Session Timeout                     2026/01/21 04:45:32.125      2028             1
L2TP Admin Delete                      2026/01/21 04:44:14.714      43738            1
-----------------------------------------------------------------------------------------------
```

Step 4 Optionally, use the **show disconnect-history l2tp-tunnel upf** *upf-name* **details** command to display last 100 disconnect records along with tunnel details.

Example:

```
bng# show disconnect-history l2tp-tunnel upf asr9k-lns details

Wed Jan  21 04:57:21.810 UTC+00:00
{
  "subResponses": [
    {
      "LocalTunnelId": 43738,
      "RemoteTunnelId": 22331,
      "Upf": "asr9k-lns",
      "SrcIp": "10.1.12.103",
      "DstIp": "10.1.15.136",
      "GroupId": "one",
      "TunnelType": "LAC",
      "DiscReason": "L2TP Admin Delete"
    },
    {
      "LocalTunnelId": 2028,
      "RemoteTunnelId": 22332,
      "Upf": "asr9k-lns",
      "SrcIp": "10.1.12.103",
      "DstIp": "10.1.15.136",
      "GroupId": "one",
      "TunnelType": "LAC",
      "DiscReason": "No-Session Timeout"
    },
    ...
    {
```

Step 5 Optionally, use the **show disconnect-history l2tp-tunnel upf** *upf-name* **filter tunnel-id** *tunnel-ID* command to display disconnect history details for tunnels that match the specified tunnel ID and UPF name.

Example:

```
bng# show disconnect-history l2tp-tunnel upf asr9k-lns filter tunnel-id 2028

Wed Jan  21 04:57:39.563 UTC+00:00
filter tunnel-id 2028
{
  "subResponses": [
    {
      "LocalTunnelId": 2028,
      "RemoteTunnelId": 22332,
      "Upf": "asr9k-lns",
      "SrcIp": "10.1.12.103",
      "DstIp": "10.1.15.136",
      "GroupId": "one",
      "TunnelType": "LAC",
      "DiscReason": "No-Session Timeout"
    }
  ]
}
```

Note A maximum of 30,000 L2TP-tunnel disconnect history records can be stored across all UPFs and instances (this is the system-wide limit for the entire Control Plane). Once the limit is reached, the oldest record is overwritten as new records are added.

Clear L2TP-tunnel disconnect history records

Remove L2TP-tunnel disconnect history records from the system for maintenance or troubleshooting.

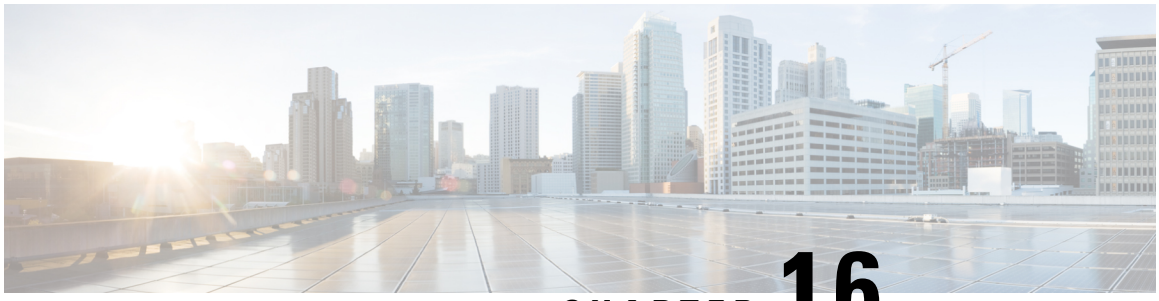
Before you begin

- Confirm which records (all, or specific User Plane) you wish to clear.

Procedure

Step 1 Use the **clear disconnect-history l2tp-tunnel** command to clear all L2TP-tunnel disconnect history records.

Step 2 Use the **clear disconnect-history l2tp-tunnel** *user-plane-name* command to clear records for a specific User Plane.



CHAPTER 16

Log Generation Support

- Feature Summary and Revision History, on page 375
- Feature Description, on page 375

Feature Summary and Revision History

Summary Data

Table 86: Summary Data

Applicable Product(s) or Functional Area	cnBNG
Applicable Platform(s)	SMI
Feature Default Setting	Enabled -Always-on
Related Changes in this Release	Not Applicable
Related Documentation	Not Applicable

Revision History

Table 87: Revision History

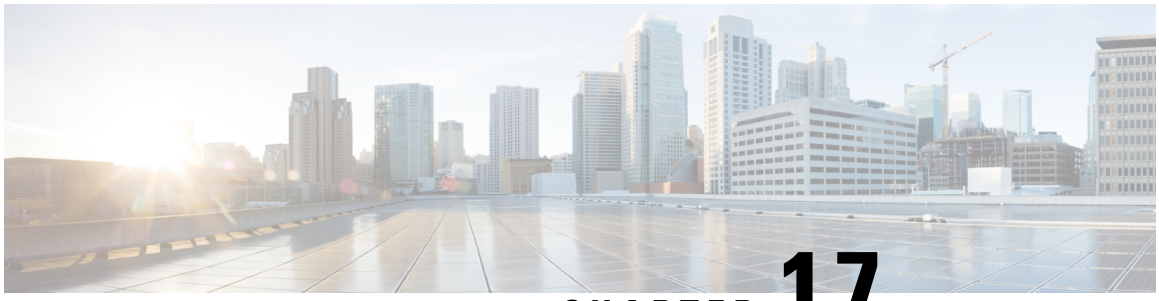
Revision Details	Release
First introduced.	2021.01.0

Feature Description

The Cloud Native Broadband Network Gateway utilizes the common logging framework to generate logs from its microservices.

The supported logging levels are:

- Error
- Warn
- Info
- Debug
- Trace



CHAPTER 17

Monitor Protocol and Subscriber

- Feature Summary and Revision History, on page 377
- Feature Description, on page 378
- Configuring Monitor Subscriber and Protocol, on page 378

Feature Summary and Revision History

Summary Data

Table 88: Summary Data

Applicable Product(s) or Functional Area	cnBNG
Applicable Platform(s)	SMI
Feature Default Setting	Disabled - Configuration Required
Related Documentation	*Cloud Native BNG Control Plane Command Reference Guide*

Revision History

Table 89: Revision History

Revision Details	Release
Enhancement Introduced: The Monitor Subscriber and Protocol feature is NSO-integrated.	2021.04.0
First introduced.	2021.03.0

Feature Description

Note This feature is Network Services Orchestrator (NSO) integrated.

The Monitor Subscriber and Protocol feature supports the debugging functionality.

Monitor Subscriber

The Monitor Subscriber feature captures all the transactional logs for a given subscriber over a specified period of time across all the Kubernetes pods. It also supports the simultaneous monitoring of multiple subscribers on a given cluster. This information allows to track all the events that had occurred for a given subscriber when the subscriber was coming up or going down.

To configure Monitor Subscriber, see Configuring Monitor Subscriber, on page 378

Monitor Protocol

The Monitor Protocol feature replicates the packets from different protocol endpoints of cnBNG and sends it to the OAM pod. There two levels of packet replication that occur:

- First replication dumps only the basic packet information
- Second replication dumps the full packet with details like headers, keys of subscriber, and so on.

This feature captures all ingress and egress packets on the cnBNG protocol pods.

To configure Monitor Protocol, see Configuring Monitor Protocol, on page 389

Configuring Monitor Subscriber and Protocol

This section describes how to configure subscriber and protocol monitoring.

Configuring the Monitor Subscriber and Protocol feature involves the following procedures:

- Configuring Monitor Subscriber
- Configuring Monitor Protocol
- Copying Log Files
- Viewing Log Files

Configuring Monitor Subscriber

Use the following commands to enable the monitoring of a subscriber.

monitor subscriber supi *subcriber_id* **capture-duration** *duration_in_seconds*

NOTES:

- **supi** *subcriber_id* : Enables monitoring of subscribers based on the subscriber identifier (supi). For example: 0000.4096.3e4a.

  The subscriber-id format supported is as follows:

  <mac-adress>@<upf>: This specifies a particular subscriber with the given MAC address from a specific User Plane function (UPF).

  Wildcard subscriber-id is also supported. For example:

  - *@<upf>: This specifies all subscribers from a specific UPF.
  - <mac>@*: This specfies all subscribers having the given MAC and from any UPF.
  - *: This specifies all subscribers from all UPFs.

- **capture-duration** : Specifies the duration in seconds during which the monitor subscriber is enabled. The *duration_in_seconds* can range from 1 to 2147483647 seconds. The default is 300.
- Other sub-options that are present in the CLI command are not supported

Example

```
bng# monitor subscriber supi aabb.0000.0001@automation-userplane
supi: aabb.0000.0001@automation-userplane
captureDuraiton: 300
enableInternalMsg: false
enableTxnLog: false
namespace(deprecated. Use nf-service instead.): none
nf-service: none
gr-instance: 0
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100   337  100   119  100   218  10818  19818 --:--:-- --:--:-- --:--:-- 30636
Command: --header Content-type:application/json --request POST --data
{"command":"mon_sub","params":{"supi":"aabb.0000.0001@automation-userplane","duration":300,"enableTxnLog":false,"enableInternalMsg":false,"action":"start","namespace":"none","nf-service":"none","grInstance":0}}
 http://oam-pod:8879/commands
Result start mon_sub, fileName
->logs/monsublogs/none.aabb.0000.0001@automation-userplane_TS_2021-06-09T12:17:33.838574118.txt
Starting to tail the monsub messages from file:
logs/monsublogs/none.aabb.0000.0001@automation-userplane_TS_2021-06-09T12:17:33.838574118.txt
Defaulting container name to oam-pod.
Use 'kubectl describe pod/oam-pod-0 -n bng' to see all of the containers in this pod.
      Subscriber Id: aabb.0000.0001@automation-userplane
      Timestamp: 2021/06/09 12:19:30.194843
      Message: BNGN4UdpProxyMessage
      Description: Received Packet IPOE, IPC Message from udp-proxy
      Source: bng.udp-proxy.DC.Local.0
      Destination: bng.bng-n4-protocol.DC.Local.0
      PAYLOAD:
        BNGN4UdpProxyMessage:
            BNGN4UdpProxyMessage:
                Type: 6
                L2Data:
                    SrcMac: aabb.0000.0001
                    DstMac: ffff.ffff.ffff
                    Outervlan: 100
                    Innervlan: 200
                    OuterCos: 0
                    InnerCos: 0
                IpAddr:
```

```
                    AfType: 1
                    SrcIpv4:
                    SrcIpv6:
                    DstIPv4: 8.8.8.8
                    DstIPv6:
                    LinkLocal:
                    Port: 8000
                UpData:
                    AccessInterface: GigabitEthernet0/0/0/1
                    CpSubscriberId: 0
                    UpSubscriberId: 0
                    UPSubInterfaceId: 0
                    RouterName: automation-userplane
                    AccessVrf: access-vrf-1
                    NASID: NAS-ID-1
                NasInfo:
                    Port: 4
                    Slot: 2
                    Adapter: 5
                    Subslot: 3
                    Chasis: 1
                    InterfaceType: 1
                L2TPData:
                    PuntPoliceRate: 0
                    L2TPTos: 0
                    TunnelID: 0
                Packet:
                    Payload:
                        BaseLayer:
                        Operation: 1
                        HardwareType: 1
                        HardwareLen: 6
                        HardwareOpts: 0
                        Xid: 1
                        Secs: 0
                        Flags: 32768
                        ClientIP: 0.0.0.0
                        YourClientIP: 0.0.0.0
                        NextServerIP: 0.0.0.0
                        RelayAgentIP: 0.0.0.0
                        ClientHWAddr: aa:bb:00:00:00:01
                        ServerName:
                        File:
                        Options: {
  Option(MessageType:Discover)
  Option(ClientID:[1 170 187 0 0 0 1])
}

-------------------------------------------------

      Subscriber Id: aabb.0000.0001@automation-userplane
      Timestamp: 2021/06/09 12:19:30.205174
      Message: RadiusUdpProxyMsg
      Description: Send Auth/Acct Request Message to UDP-Proxy
      Source: bng.radius-ep.DC.Local.0
      Destination: bng.udp-proxy.DC.Local.0
      PAYLOAD:
        RadiusUdpProxyMsg:
            RadiusUdpProxyMsg:
                SrcIp: 10.105.254.113
                SrcPort: 16384
                DestIp: 10.105.254.114
                DestPort: 1812
                Payload:
```

```
   Code = AccessRequest
   Id = 2
   Authenticator = [148 88 241 197 50 83 83 156 105 245 107 167 117 131 237 165]
   User-Name = "cnbng"
   User-Password = 0x30b19d11f96401290b6410e8a1b324eb
   NAS-IP-Address = 10.105.254.113
   NAS-Port = 16384
   Service-Type = 5
   Called-Station-Id = "1"
   Calling-Station-Id = "1"
   Nas-Identifier = "bng"
   Acct-Session-Id = "Local_DC_16777218"
   Event-Timestamp = 1623241161
   NAS-Port-Type = 41
   NAS-Port-Id = "124536"
   NAS-IPv6-Address = ::/0
   Cisco-Vsa_cisco-nas-port = "124536"
   Cisco-Vsa_cisco-dhcp-client-id = 0x01aabb00000001
   Cisco-Vsa_Cisco AVpair = "client-mac-address=aabb.0000.0001"
   Cisco-Vsa_Cisco AVpair = 0x646863702d636c69656e742d69643d01aabb00000001
               PayloadLen: 231
               SubscriberID: aabb.0000.0001@automation-userplane

-------------------------------------------------

      Subscriber Id: aabb.0000.0001@automation-userplane
      Timestamp: 2021/06/09 12:19:30.206778
      Message: RadiusUdpProxyMsg
      Description: Received Auth/Acct Response Message from UDP-Proxy
      Source: bng.udp-proxy.DC.Local.0
      Destination: bng.radius-ep.DC.Local.0
      PAYLOAD:
        RadiusUdpProxyMsg:
            RadiusUdpProxyMsg:
                SrcIp: 10.105.254.114
                SrcPort: 1812
                DestIp: 10.105.254.113
                DestPort: 16384
                Payload:
   Code = AccessAccept
   Id = 2
   Authenticator = [127 214 195 68 205 142 58 23 126 138 11 70 241 169 153 92]
               PayloadLen: 20

-------------------------------------------------

      Subscriber Id: aabb.0000.0001@automation-userplane
      Timestamp: 2021/06/09 12:19:30.216130
      Message: DHCPPacketTx
      Description: Sending Packet IPOE, IPC Message to udp-proxy
      Source: bng.bng-n4-protocol.DC.Local.0
      Destination: bng.udp-proxy.DC.Local.0
      PAYLOAD:
        DHCPPacketTx:
            DHCPPacketTx:
                Type: 6
                L2Data:
                    DstMac: ff:ff:ff:ff:ff:ff
                    Outervlan: 100
                    Innervlan: 200
                    OuterCos: 0
                    InnerCos: 0
                IpAddr:
                    AfType: 1
```

```
                        SrcIpv4: 33.0.0.1
                        SrcIpv6:
                        DstIPv4: 255.255.255.255
                        DstIPv6:
                        LinkLocal:
                        Port: 68
                    UpData:
                        AccessInterface: GigabitEthernet0/0/0/1
                        CpSubscriberId: 16777218
                        UpSubscriberId: 0
                        UPSubInterfaceId: 0
                        RouterName: automation-userplane
                        AccessVrf: access-vrf-1
                        NASID: NAS-ID-1
                    Packet:
                        Payload:
                            BaseLayer:
                            Operation: 2
                            HardwareType: 1
                            HardwareLen: 6
                            HardwareOpts: 0
                            Xid: 1
                            Secs: 0
                            Flags: 32768
                            ClientIP: 0.0.0.0
                            YourClientIP: 33.0.0.3
                            NextServerIP: 0.0.0.0
                            RelayAgentIP: 0.0.0.0
                            ClientHWAddr: aa:bb:00:00:00:01
                            ServerName:
                            File:
                            Options: {
  Option(MessageType:Offer)
  Option(ClientID:[1 170 187 0 0 0 1])
  Option(SubnetMask:255.255.224.0)
  Option(LeaseTime:90060)
  Option(Timer1:45030)
  Option(Timer2:78802)
  Option(ServerID:33.0.0.1)
}

------------------------------------------------

      Subscriber Id: aabb.0000.0001@automation-userplane
      Timestamp: 2021/06/09 12:19:30.293167
      Message: BNGN4UdpProxyMessage
      Description: Received Packet IPOE, IPC Message from udp-proxy
      Source: bng.udp-proxy.DC.Local.0
      Destination: bng.bng-n4-protocol.DC.Local.0
      PAYLOAD:
        BNGN4UdpProxyMessage:
            BNGN4UdpProxyMessage:
                Type: 6
                L2Data:
                    SrcMac: aabb.0000.0001
                    DstMac: ffff.ffff.ffff
                    Outervlan: 100
                    Innervlan: 200
                    OuterCos: 0
                    InnerCos: 0
                IpAddr:
                    AfType: 1
                    SrcIpv4:
                    SrcIpv6:
```

```
                        DstIPv4: 8.8.8.8
                        DstIPv6:
                        LinkLocal:
                        Port: 8000
                    UpData:
                        AccessInterface: GigabitEthernet0/0/0/1
                        CpSubscriberId: 0
                        UpSubscriberId: 0
                        UPSubInterfaceId: 0
                        RouterName: automation-userplane
                        AccessVrf: access-vrf-1
                        NASID: NAS-ID-1
                    NasInfo:
                        Port: 4
                        Slot: 2
                        Adapter: 5
                        Subslot: 3
                        Chasis: 1
                        InterfaceType: 1
                    L2TPData:
                        PuntPoliceRate: 0
                        L2TPTos: 0
                        TunnelID: 0
                    Packet:
                        Payload:
                            BaseLayer:
                            Operation: 1
                            HardwareType: 1
                            HardwareLen: 6
                            HardwareOpts: 0
                            Xid: 1
                            Secs: 0
                            Flags: 32768
                            ClientIP: 0.0.0.0
                            YourClientIP: 0.0.0.0
                            NextServerIP: 0.0.0.0
                            RelayAgentIP: 0.0.0.0
                            ClientHWAddr: aa:bb:00:00:00:01
                            ServerName:
                            File:
                            Options: {
  Option(MessageType:Request)
  Option(ClientID:[1 170 187 0 0 0 1])
  Option(ServerID:33.0.0.1)
  Option(RequestIP:33.0.0.3)
}

-------------------------------------------------

      Subscriber Id: aabb.0000.0001@automation-userplane
      Timestamp: 2021/06/09 12:19:30.301343
      Message: BNGN4SessionEstablishmentReq
      Description: Sending N4 Session Establishment Request, IPC Message to udp-proxy
      Source: bng.bng-n4-protocol.DC.Local.0
      Destination: bng.udp-proxy.DC.Local.0
      PAYLOAD:
        BNGN4SessionEstablishmentReq:
            BNGN4SessionEstablishmentReq:
                PfcpSessionHeader:
                    Version: 1
                    SeidSet: true
                    MessageType: 50
                    MessageLen: 413
                    SequenceNumber: 5
```

```
    Seid: 0
    Priority: 1
NodeID:
    Valid: true
    Ip: Afi=v4 Ip=10.105.254.113
Fseid:
    Valid: true
    Seid: 16777218
    Ipv4: 0.0.0.0
    Ipv6:
CreatePdrList:
    CreatePdrList[0]:
        Valid: true
        PdrId:
            Valid: true
            RuleId: 1
        Precedence:
            Valid: true
            Val: 1
        Pdi:
            Valid: true
            SrcIface:
                Valid: true
                Value: 1
            TrafficEndptId:
                Valid: true
                Val: 1
        FarId:
            Valid: true
            Val: 1
        OuterHeaderRemoval:
            Valid: false
            Description: 0
        UrrId:
            Valid: true
            Val: 1
    CreatePdrList[1]:
        Valid: true
        PdrId:
            Valid: true
            RuleId: 2
        Precedence:
            Valid: true
            Val: 1
        Pdi:
            Valid: true
            SrcIface:
                Valid: true
                Value: 2
            TrafficEndptId:
                Valid: false
                Val: 0
        FarId:
            Valid: true
            Val: 2
        OuterHeaderRemoval:
            Valid: false
            Description: 0
        UrrId:
            Valid: false
            Val: 0
CreateFarList:
    CreateFarList[0]:
        Valid: true
```

```
        FarId:
            Valid: true
            Val: 1
        ApplyAction:
            Valid: true
            Drop: false
            Forward: true
            Buffer: false
            NotifyCP: false
            Duplicate: false
        ForwParams:
            Valid: true
            DestIface:
                Valid: true
                Value: 2
            OuterHeaderCreation:
                Valid: true
                CprNSH: false
                TfEndpt: true
                L2tp: false
                Ppp: false
                TunnelID: 0
                SessionID: 0
        DuplParams:
            Valid: false
            DestIface:
                Valid: false
                Value: 0
            OuterHeaderCreation:
                Valid: false
                Teid: 0
                Ipv4:
                Ipv6:
                PortNum: 0
            IntrInfo:
                Valid: false
                InterceptId:
                    Valid: false
                Dscp:
                    Valid: false
                    Dscp: 0
    CreateFarList[1]:
        Valid: true
        FarId:
            Valid: true
            Val: 2
        ApplyAction:
            Valid: true
            Drop: false
            Forward: true
            Buffer: false
            NotifyCP: false
            Duplicate: false
        ForwParams:
            Valid: true
            DestIface:
                Valid: true
                Value: 1
            OuterHeaderCreation:
                Valid: true
                CprNSH: false
                TfEndpt: true
                L2tp: false
                Ppp: false
```

```
                TunnelID: 0
                SessionID: 0
            DuplParams:
                Valid: false
                DestIface:
                    Valid: false
                    Value: 0
                OuterHeaderCreation:
                    Valid: false
                    Teid: 0
                    Ipv4:
                    Ipv6:
                    PortNum: 0
                IntrInfo:
                    Valid: false
                    InterceptId:
                        Valid: false
                    Dscp:
                        Valid: false
                        Dscp: 0
    CreateTrafficEndptList:
        CreateTrafficEndptList[0]:
            Valid: true
            Tfid:
                Valid: true
                Val: 1
            AccessPortId:
                Valid: true
                Value: GigabitEthernet0/0/0/1
            UeIPAddr:
                Valid: true
                Flags: 2
                Ipv4Addr: Afi=v4 Ip=33.0.0.3
                Ipv6Addr:
                IPv6PrefixLen: 0
                Ipv6PDAddr:
                Ipv6LLAddr:
            UeMacAddress: aa:bb:00:00:00:01
            PppoeSessId:
                Valid: false
                Value: 0
            AddressFamily:
                Valid: true
                Value: 3
            Cvlan:
                Valid: true
                Pcp: 0
                Dei: 0
                VlanId: 200
            Svaln:
                Valid: true
                Pcp: 0
                Dei: 0
                VlanId: 100
            L2tpTunnel:
                Valid: false
                TunnelEndpoint:
                    Valid: false
                    Choose: false
                    LocalID: 0
                    RemoteID: 0
                SessionID:
                    Valid: false
                    SessionID: 0
```

```
                RemoteSessionID: 0
            TunnelFeatures:
                Valid: false
                SetTOS: false
                ReflectTOS: false
                SetDF: false
                ReflectDF: false
                TcpMssAdjust: false
                TunnelStatsEnabled: false
                SessStatsEnabled: false
                TSI: false
                SSI: false
                TosVal: 0
                TcpMssVal: 0
                TunnelStatsInterval: 0
                SessStatsInterval: 0
SubParams:
    Valid: true
    Stype:
        Valid: true
        Value: 1
    SrgIntfId:
        Valid: false
        Value: 0
    SrgGrpId:
        Valid: false
        Value: 0
    Vrf:
        Valid: true
        Value: automation-vrf
    AccessVrf:
        Valid: false
CreateURR:
    CreateURR[0]:
        Valid: true
        UrrID:
            Valid: true
            Val: 1
        MeasurementMethod:
            Valid: true
            Event: false
            Volume: true
            Duration: false
        Trigger:
            Valid: true
            PeriodicReporting: true
            VolumeThreshold: false
            TimeThreshold: false
            QuotaHoldingTime: false
            StartOfTraffic: false
            StopOfTraffic: false
            DroppedDlTrafficThreshold: false
            ImmediateReport: false
            VolumeQuota: false
            TimeQuota: false
            LinkedUsageReporting: false
            TerminationReport: true
            MonitoringTime: false
            EnvelopeClosure: false
            MacAddressReporting: false
            EventThreshold: false
            EventQuota: false
            TerminationByUP: false
        MeasurementPeriod:
```

```
            Valid: true
            Val: 1940
Keepalive:
    Valid: false
    Tfid:
        Valid: false
        Val: 0
    Timer:
        Valid: false
        TimeInterval: 0
        RetryCount: 0
    MagicNum:
        Valid: false
        LocalMagicNum: 0
        PeerMagicNum: 0
CreateQspList:
    CreateQspList[0]:
        Valid: true
        Service:
            Valid: true
            Length: 0
            Value: automation-feature-template-accounting
        QosIngress:
            Valid: true
            Length: 0
            Name: inpolicy
            Priority: 0
        QosEgress:
            Valid: true
            Length: 0
            Name: outpolicy
            Priority: 0
        Stats:
            Valid: true
            Value: true
        Spi:
            Valid: false
            Value: 0
        PlainQos: false
CreateACL:
    Valid: false
    Ipv4InACL:
        Valid: false
    Ipv4OutACL:
        Valid: false
    Ipv6InACL:
        Valid: false
    Ipv6OutACL:
        Valid: false
CreatePBR:
    Valid: false
    PbrIngress:
        Valid: false
        Length: 0
CreateuRPF:
    Valid: false
    Strictv4: false
    Strictv6: false
    Loosev4: false
    Loosev6: false
CreateICMP:
    Valid: false
    V4: false
    V6: false
```

```
                    RemoveICMP:
                        Valid: false
                        V4: false
                        V6: false
                    CreateMTU:
                        Valid: true
                        V4Mtu: 1400
                        V6Mtu: 0
                        PPPMtu: 0
                    TransactionIdentifier:
                        Valid: true
                        Value: 1
------------------------------------------------
```

Configuring Monitor Protocol

Use the following commands to enable protocol monitoring for a subscriber.

monitor protocol interface *pcap_interface* **capture-duration** *duration_in_seconds*

NOTES:

- **interface** *pcap_interface* : Specifies the packet capture (PCAP) interface. The valid PCAP interfaces are: Packet Forwarding Control Protocol (PFCP), GPRS Tunnelling Protocol User Plane (GTP-U), and Remote Authentication Dial-In User Service (RADIUS).
- **capture-duration** *duration_in_seconds* : Specifies the duration in seconds during which the monitor protocol is enabled. The *duration_in_seconds* can range from 1 to 2147483647 seconds. The default is 300.
- cnBNG uses a custom GTPU packet format. Therefore, packet decode errors are displayed on the screen because the standard decode plugin does not support the cnBNG format. Capture the packet to PCAP and use the cnBNG specific LUA plugin during Wireshark decode.
- Interface names must be entered manually and must match the name mentioned in the description, else the packet capture may fail.
- Only one physical-interface (NIC) packet capture is supported. For PFCP and GTPU this limitation is not applicable as they always run-on a single interface (VIP). However for RADIUS, certain deployments may use different VIPs for Auth/Acct/COA, leading to different physical NICs. Due to the infrastructure limitation, packet-capture can run on only one of the physical-NICs.

Example

monitor protocol interface pfcp

```
InterfaceName = N4:10.86.73.161:8805 |  InterfaceIP = 10.86.73.161 | Filter = (tcp or udp)
 and (port 8805)
<<<<OUTBOUND
from 10.86.73.161:8805 to 10.86.73.162:8805
Protocol: UDP | Sequence Number: 0
Packet Metadata: {Timestamp:2019-10-22 09:22:34.029363 +0000 UTC CaptureLength:72 Length:72
 InterfaceIndex:2 AncillaryData:[]}

Packet Raw Bytes:

0050569c14610050569c8d5c0800450000003a76c5400040111bff0a5649a10a5649a222652265002 6a8262006001a00000004003c0005000a5649a1001300010100600004e159480e

Packet Dump:
```

```
 -- FULL PACKET DATA (72 bytes) ------------------------------------
00000000  00 50 56 9c 14 61 00 50  56 9c 8d 5c 08 00 45 00
00000010  00 3a 76 c5 40 00 40 11  1b ff 0a 56 49 a1 0a 56
00000020  49 a2 22 65 22 65 00 26  a8 26 20 06 00 1a 00 00
00000030  00 04 00 3c 00 05 00 0a  56 49 a1 00 13 00 01 01
00000040  00 60 00 04 e1 59 48 0e
--- Layer 1 ---
Ethernet {Contents=[..14..] Payload=[..58..] SrcMAC=00:50:56:9c:8d:5c DstMAC=00:50:56:9c:14:61
 EthernetType=IPv4 Length=0}
00000000  00 50 56 9c 14 61 00 50  56 9c 8d 5c 08 00
--- Layer 2 ---
IPv4 {Contents=[..20..] Payload=[..38..] Version=4 IHL=5 TOS=0 Length=58 Id=30405 Flags=DF
 FragOffset=0 TTL=64 Protocol=UDP Checksum=7167 SrcIP=10.86.73.161 DstIP=10.86.73.162
Options=[] Padding=[]}
00000000  45 00 00 3a 76 c5 40 00  40 11 1b ff 0a 56 49 a1
00000010  0a 56 49 a2
--- Layer 3 ---
UDP {Contents=[..8..] Payload=[..30..] SrcPort=8805(pfcp) DstPort=8805(pfcp) Length=38
Checksum=43046}
00000000  22 65 22 65 00 26 a8 26                          |"e"e.&.&|
--- Layer 4 ---
Payload 30 byte(s)
00000000  20 06 00 1a 00 00 00 04  00 3c 00 05 00 0a 56 49
00000010  a1 00 13 00 01 01 00 60  00 04 e1 59 48 0e
```

Copying Log Files

Use the following commands to copy the stored log files externally or on the BNG Ops Center.

These files ether can be copied outside or dumped on the bng-opscenter using the following CLI command.

monitor subscriber-dump filename <file path got from monitor subscriber-list CLI>

Example:

```
monitor subscriber dump filename
/opt/workspace/logs/monsublogs/none.aabb.0000.0001@automation-userplane_TS_2021-06-09T12:17:33.838574118.txt.sorted
RELEASE_NAMESPACE: 'bng'
Dumping file
'/opt/workspace/logs/monsublogs/none.aabb.0000.0001@automation-userplane_TS_2021-06-09T12:17:33.838574118.txt.sorted'
**** Received 19 messages ******
      Subscriber Id: aabb.0000.0001@automation-userplane
      Timestamp: 2021/06/09 12:19:30.194843
      Message: BNGN4UdpProxyMessage
      Description: Received Packet IPOE, IPC Message from udp-proxy
      Source: bng.udp-proxy.DC.Local.0
      Destination: bng.bng-n4-protocol.DC.Local.0
      PAYLOAD:
        BNGN4UdpProxyMessage:
            BNGN4UdpProxyMessage:
                Type: 6
                L2Data:
                    SrcMac: aabb.0000.0001
                    DstMac: ffff.ffff.ffff
                    Outervlan: 100
                    Innervlan: 200
                    OuterCos: 0
                    InnerCos: 0
                IpAddr:
                    AfType: 1
                    SrcIpv4:
                    SrcIpv6:
```

```
                    DstIPv4: 8.8.8.8
                    DstIPv6:
                    LinkLocal:
                    Port: 8000
                UpData:
                    AccessInterface: GigabitEthernet0/0/0/1
                    CpSubscriberId: 0
                    UpSubscriberId: 0
                    UPSubInterfaceId: 0
                    RouterName: automation-userplane
                    AccessVrf: access-vrf-1
                    NASID: NAS-ID-1
                NasInfo:
                    Port: 4
                    Slot: 2
                    Adapter: 5
                    Subslot: 3
                    Chasis: 1
                    InterfaceType: 1
                L2TPData:
                    PuntPoliceRate: 0
                    L2TPTos: 0
                    TunnelID: 0
                Packet:
                    Payload:
                        BaseLayer:
                        Operation: 1
                        HardwareType: 1
                        HardwareLen: 6
                        HardwareOpts: 0
                        Xid: 1
                        Secs: 0
                        Flags: 32768
                        ClientIP: 0.0.0.0
                        YourClientIP: 0.0.0.0
                        NextServerIP: 0.0.0.0
                        RelayAgentIP: 0.0.0.0
                        ClientHWAddr: aa:bb:00:00:00:01
                        ServerName:
                        File:
                        Options: {
                                Option(MessageType:Discover)
                                Option(ClientID:[1 170 187 0 0 0 1]).
.


-------------------------------------------------

      Subscriber Id: aabb.0000.0001@automation-userplane
      Timestamp: 2021/06/09 12:19:30.205174
      Message: RadiusUdpProxyMsg
      Description: Send Auth/Acct Request Message to UDP-Proxy
      Source: bng.radius-ep.DC.Local.0
      Destination: bng.udp-proxy.DC.Local.0
      PAYLOAD:
        RadiusUdpProxyMsg:
            RadiusUdpProxyMsg:
                SrcIp: 10.105.254.113
                SrcPort: 16384
                DestIp: 10.105.254.114
                DestPort: 1812
                Payload:
```

```
------------------------------------------------

      Subscriber Id: aa11.0000.0003@asr9k-1
      Timestamp: 2021/06/03 06:26:26.796023
      Message: RadiusUdpProxyMsg
      Description: Send Auth/Acct Request Message to UDP-Proxy
      Source: BNG.radius-ep.DC.Local.0
      Destination: BNG.udp-proxy.DC.Local.0
      PAYLOAD:
        RadiusUdpProxyMsg:
            RadiusUdpProxyMsg:
                SrcIp: 10.1.4.150
                SrcPort: 16384
                DestIp: 10.1.4.151
                DestPort: 1813
                Payload:
                        Code = AccountingRequest
                        Id = 31
                        Authenticator = [88 13 251 114 225 205 9 68 52 194 48 231 234 226
226 184]
                        User-Name = "cnbng"
                        NAS-IP-Address = 10.1.4.150
                        NAS-Port = 16384
                        Service-Type = 5
                        Framed-IP-Address = 1.0.3.13
                        Nas-Identifier = "CISCO-BNG-ACCT"
                        Acct-Status-Type = 1
                        Acct-Delay-Time = 0
                        Acct-Session-Id = "Local_DC_16777230"
                        Event-Timestamp = 1622701602
                        NAS-Port-Type = 41
                        Acct-Interim-Interval = 300
                        NAS-Port-Id = "asr9k-1/2/3/4/100.200"
                        NAS-IPv6-Address = ::/0
                        Cisco-Vsa_cisco-nas-port = "asr9k-1/2/3/4/100.200"
                        Cisco-Vsa_cisco-dhcp-client-id = 0x01aa1100000003
                        Cisco-Vsa_Cisco AVpair = "client-mac-address=aa11.0000.0003"
                        Cisco-Vsa_Cisco AVpair = "dhcp-class=RJIL_DHCPV4_CLASS_2"
                        Cisco-Vsa_Cisco AVpair = "dhcp-class=RJIL_DHCPv6_CLASS_1"
                        Cisco-Vsa_Cisco AVpair = "accounting-list=aaa-prof1"
                        Cisco-Vsa_Cisco AVpair =
0x646863702d636c69656e742d69643d01aa1100000003
                        Cisco-Vsa_Cisco AVpair = "vrf=ISP"
                PayloadLen: 396
                SubscriberID: aa11.0000.0003@asr9k-1

------------------------------------------------

      Subscriber Id: aa11.0000.0003@asr9k-1
      Timestamp: 2021/06/03 06:26:26.800776
      Message: RadiusUdpProxyMsg
      Description: Received Auth/Acct Response Message from UDP-Proxy
      Source: BNG.udp-proxy.DC.Local.0
      Destination: BNG.radius-ep.DC.Local.0
      PAYLOAD:
        RadiusUdpProxyMsg:
            RadiusUdpProxyMsg:
                SrcIp: 10.1.4.151
                SrcPort: 1813
                DestIp: 10.1.4.150
                DestPort: 16384
                Payload:
                        Code = AccountingResponse
```

```
                        Id = 31
                        Authenticator = [168 192 147 70 117 31 151 16 237 80 68 105 42 191
 23 186]
               PayloadLen: 20

------------------------------------------------
 bng#
```



Note

- While receiving CoA or DM packets, the RADIUS pod does not have the subscriber-information, instead the information is available only with the BNG-SM pod. Therefore, the packet related session programming N4-SESS-UPDATE TX and RX is dumped on the screen first followed by the CoA or DM TX and RX dump.
- Packet dumps are not captured for PFCP session report request and response.

Viewing Log Files

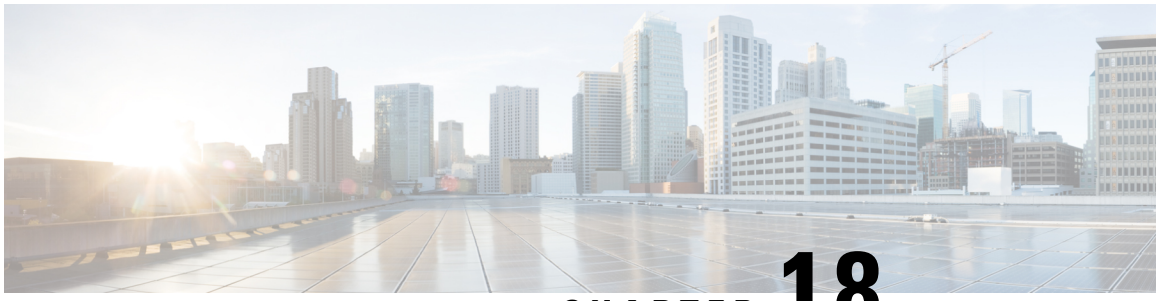
Use the following commands to view the stored log files for a monitor protocol or subscriber.

monitor subscriber list
monitor protocol list

The following is a sample output for the **monitor subscriber list**.

Example:

```
bng# monitor subscriber list
none.aa11.0000.0004*_TS_2021-06-03T06:28:13.564009704.txt.sorted
none.aa11.0000.0003@asr9k-1_TS_2021-06-03T06:26:20.627655233.txt.sorted
none.*_TS_2021-06-03T06:25:04.176857711.txt.sorted
bng#
```



CHAPTER 18

Multiple Replica Support for cnBNG Services

- Feature Summary and Revision History, on page 395
- Feature Description, on page 396
- Configuring Multiple Replica Support for cnBNG Services, on page 397

Feature Summary and Revision History

Summary Data

Table 90: Summary Data

Applicable Product(s) or Functional Area	cnBNG
Applicable Platform(s)	SMI
Feature Default Setting	Disabled - Configuration Required
Related Documentation	*Cloud Native BNG Control Plane Command Reference Guide*

Revision History

Table 91: Revision History

Revision Details	Release
Enhancement Introduced: The Multiple Replica Support for cnBNG Services feature is NSO-integrated.	2021.04.0
First introduced.	2021.03.0

Feature Description

Note This feature is Network Services Orchestrator (NSO) integrated.

The Multiple Replica Support for cnBNG Services is designed to support multiple instances of the cnBNG services and load balance the session transactions to address the following requirements:

- Higher scalability
- Calls per Second (CPS) [CEPS - Call Events per Second (CEPS), TPS - Transactions per Second (TPS)]
- Redundancy

The following services must be configured to support multiple instances for load-balancing the session transactions.

- bng-dhcp
- bng-pppoe
- bng-l2tp-tunnel
- bng-sm
- bng-n4-protocol
- radius-ep
- udp-proxy
- bng-node-manager

To configure Multiple Replica Support, see Configuring Multiple Replica Support for cnBNG Services, on page 397.

How it Works

In a microservices architecture, a service mesh refers to a network of microservices that make up an application and the interactions they have among them. Istio is an open source service mesh that layers transparently on existing distributed applications.

Istio makes it easy to create a network of deployed services with load balancing, service-to-service authentication, monitoring, and so on. Therefore, Istio support is added to the cnBNG services, which intercepts all network communication between the microservices. The CP functionality is used to configure and manage Istio.

The cnBNG pod layout ensures that instances of a service are distributed across vitual machines (VMs) to ensure VM level redundancy.

Configuring Multiple Replica Support for cnBNG Services

This section describes how to configure Multiple Replica Support for cnBNG Services.

Configuring Multiple Replica Support for cnBNG Services involves the following procedure:

Replicating Multiple cnBNG Service Instances

Replicating Multiple cnBNG Service Instances

Use the following commands to replicate multiple cnBNG service instances.

```
config
  instance instance_id
    endpoint { dhcp | geo | l2tp-tunnel | n4-protocol | nodemgr | pppoe
 | radius | sm | udp-proxy }
    nodes node_replicas_for_resiliency
    replicas replicas_per_node
    commit
```

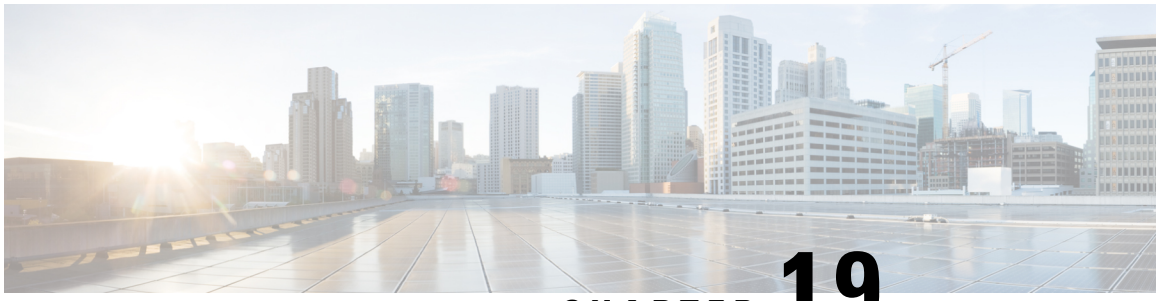
NOTES:

- **instance** *instance_id*: Configures multiple instances for the specified instance and enters instance sub-mode.
- **endpoint { dhcp | geo | l2tp-tunnel | n4-protocol | nodemgr | pppoe | radius | sm | udp-proxy }**: Configures parameters for the selected endpoint. The endpoint options are dhcp, geo, l2tp-tunnel, n4-protocol, nodemgr, pppoe, radius, sm, and udp-proxy.
- **nodes** *node_replicas_for_resiliency*: Specifies the number of node replicas for resiliency. *node_replicas_for_resiliency* must be an integer. The minimum number of nodes supported per replica is one and the maximum is 2. The default value is 1.
- **replicas** *replicas_per_node*: Specifies the number of replicas per node. *replicas_per_node* must be an integer. The minimum number of replicas supported is one and the maximum is 2. The default value is 1.



Note

- The number of replicas depend on the cluster resources and number of nodes assigned to bring up the service pods.
- Currently one replica is supported per node. Therefore, for two nodes the total number of replicas suppored are 2 * 1.



CHAPTER 19

PPPoE Subscriber Management

- Feature Summary and Revision History, on page 399
- Feature Description, on page 400
- Configuring the PPPoE Subscriber Management Feature, on page 408
- View PPPoE or LNS session event history, on page 411
- Stateless Address Autoconfiguration (SLAAC) , on page 414

Feature Summary and Revision History

Summary Data

Table 92: Summary Data

Applicable Product(s) or Functional Area	cnBNG
Applicable Platform(s)	SMI
Feature Default Setting	Disabled - Configuration Required
Related Changes in this Release	Not Applicable
Related Documentation	Not Applicable

Revision History

Table 93: Revision History

Revision Details	Release
Introduced the **show subscriber pppoe [lns] sublabel** *sublabel* **debug** command for viewing event history details for a specific PPPoE or LNS session.	2026.01.0
Introduced support for PPPoE session limit.	2025.02.0

Revision Details	Release
Introduced support for Stateless Address Autoconfiguration (SLAAC).	2024.04.0
Enhancement Introduced: The PPPoE is NSO-integrated.	2021.04.0
First introduced.	2021.01.0

Feature Description

Note This feature is Network Services Orchestrator (NSO) integrated.

Point-to-Point Protocol (PPP) over Ethernet (PPPoE) is a point-to-point link with the subscriber over an Ethernet network where the standard PPP negotiations are used for authentication and IPv4 address assignment. The basic PPPoE is defined in RFC-2516. This RFC defines two distinct stages:

- **Discovery stage:** This sets up a point-to-point session over which PPP can run between two points. For example, between the CPE and Broadband Network Gateway (BNG). This is the PPPoE protocol itself.

  Unlike PPP, the PPPoE discovery protocol defines a client-server relationship with the client initiating the discovery of the server and the subsequent setup of the point-to-point link.

- **Session stage:** This runs over the established point-to-point connection, negotiating the PPP protocols (LCP, Authentication, IPCP) as required for a standard PPP interface.

  The session stage carries the data packets from the PPPoE (this includes PPP protocol negotiation) and the actual data packets to and from the subscriber.

PPPoE Overview

The cnBNG CP supports the standard PPPoE protocol, as defined in RFC-2516. It implements the PPPoE server functionality, that is, providing PPPoE sessions to subscribers who request them. More specifically, it supports the following functionality:

- Handling incoming PPPoE Active Discovery Initiation (PADI) packets and replying with a PPPoE Active Discovery Offer (PADO) packet when the PADI is valid.
- Handling incoming PPPoE Active Discovery Request (PADR) packets and setting up a PPPoE session for the subscriber when the PADR is valid. It also replies with a PPPoE Active Discovery Session (PADS) with an allocated session-id. When the PADR is not valid (or session setup fails), a PADS is sent containing a zero session-id and an error tag.
- Handling incoming PPPoE Active Discovery Termination (PADT) packets and terminating the corresponding PPPoE sessions.
- Sending a PADT packet to the subscriber when terminating a PPPoE session.

PPPoE Features

The cnBNG supports the following PPPoE features.

PPPoE Tag Support

cnBNG supports the following PPPoE tags as defined in RFC-2516.

- Service-Name
- AC-Name tag
- AC-Cookie
- Host-Uniq tag
- Relay-Session-Id tag
- End-Of-List tag
- Vendor-Specific tags
- Error tags
- Max-payload tag

Interface types

PPPoE is generally supported on all types of Ethernet interfaces. The cloud-native CP supports PPPoE if the configuration is present either on the port identifier, NAS level, or at the router level. The UP is responsible for the interfaces where the PPPoE punt inject towards CP can be enabled.

CoS Bits

The cnBNG allows configuration of the Class-of-Service (CoS) bits value used in the Ethernet header of PADx packets. This ensure that the PPPoE control packets get treated at a higher priority. The cnBNG CP passes these values in the inject packet and the UP places these CoS values in the PADx packets it forwards towards the CPE.

Service Selection

The PPPoE Service Selection feature uses service tags to enable a PPPoE server to offer PPPoE clients a selection of different services in the PADO. Then the client chooses one of the services offered and then sends the desired service name in a PADR. This feature enables service providers to offer a variety of services and to charge customers according to the chosen services.

Whenever a PADI is received containing one of the locally configured service-names, the PADO response contains all the configured service-names.

A configuration is also provided to allow the user to disable Service Selection. In this case, the PADO only contains the service-name that was in the original PADI.

PPPoE Session Limits

The PPPoE session limits feature enables you to control the number of PPPoE sessions that can be established on a specific User Plane (UP) and PPPoE profile. By restricting session creation, this feature protects cnBNG system resources when multiple subscribers attempt to access broadband services simultaneously.

This feature provides enhanced configuration flexibility by allowing you to set session limits based on various parameters, including:

- **circuit-id** – Maximum number of sessions allowed per Circuit-ID.
- **circuit-id-and-remote-id** – Maximum number of sessions allowed per combination of Circuit-ID and Remote ID.
- **mac** – Maximum number of sessions allowed per MAC address.
- **max** – Maximum number of sessions allowed under the PPPoE profile.
- **outer-vlan** – Maximum number of sessions allowed per outer VLAN, per access interface.

PPPoE session limit based on circuit-id and remote-id

Table 94: Feature History

Feature Name	Release Information	Description
PPPoE session limit based on circuit-id and remote-id	2025.02.0	You can now limit the number of PPPoE sessions on a designated UP and PPPoE profile, using the combination of Circuit-ID and Remote-ID as filtering criteria. This feature ensures optimal protection of cnBNG system resources, effectively handling simultaneous session requests from multiple subscribers.

This feature restricts the number of PPPoE sessions on a specific UP and PPPoE profile, using the combination of Circuit-ID and Remote-ID as the enforcement criteria. If the predefined session limit is exceeded, cnBNG automatically rejects new subscriber session requests associated with that Circuit-ID and Remote-ID.

Benefits of PPPoE session limit

- **Resource protection**: By controlling the number of active sessions, you can protect cnBNG system resources from potential strain due to multiple connection attempts.
- **Enhanced stability:** The feature contributes to maintaining system stability and performance, particularly during peak usage times when multiple subscribers are attempting to connect.

Configure PPPoE session limits

Follow these steps to configure PPPoE session limits.

Procedure

Step 1 Configure PPPoE session limits to control the maximum number of PPPoE sessions that can be established on a cnBNG router.

Example:

```
config
  profile pppoe pppoe_profile_name
```

```
  session-limit { circuit-id | circuit-id-and-remote-id | mac | max | outer-vlan
 } count
 exit
```

This is a sample configuration.

```
config
   profile pppoe prof1
     session-limit circuit-id-and-remote-id 5
   exit
```

Step 2 Use the **show subscriber pppoe { count | detail }** command to verify the configuration.

Example:

```
bng# show subscriber pppoe count

Tue Oct  21 11:36:40.189 UTC+00:00
subscriber-details
{
  "sessionCount": 3
}
```

PPP Overview

The Point-to-Point Protocol provides a standard method for transporting multiprotocol datagrams over point-to-point links. It defines an encapsulation scheme, a link layer control protocol (LCP) and a set of network control protocols (NCPs) for different network protocols that can be transmitted over the PPP link.

The LCP is used to configure and maintain the data link. PPP peers use the LCP to negotiate various link layer properties or characteristics.

An NCP is used to establish and configure the associated network protocol before data packets for the protocol are transmitted. For example, IP Control Protocol (IPCP) is used to negotiate IPv4 addresses between peers.

Between LCP and NCP negotiation phases there is an optional authentication phase that the LCP exchanges are agreed upon. Several different authentication schemes are selected with Challenge Handshake Authentication Protocol (CHAP) being the most prevalent one. The basic PPP protocol is defined in RFC 1661 and there are extensions to it for various features.

PPP Features

The cnBNG supports the following point-to-point protocols required for bringing up a PPPoE session.

- Link Control Protocol (LCP): This is used for PPP link configuration.
- IP Control Protocol (IPCP): This is used to negotiate IPv4 addresses between peers.
- IPv6 Control Protocol (IPv6CP): This is used to negotiate IPv6 interface ID.
- Password Authentication Protocol (PAP): This is used to verify the identity of the peer by means of a two-way handshake
- Challenge Handshake Authentication Protocol (CHAP): This is used to verify the identity of the peer by means of a three-way handshake.

For more information about the protocols and their negotiation, refer the respective RFCs.

Address Assignment Strategies

The IPv4 address assignment occurs as part of the IPCP negotiation. The address can be part of the RADIUS profile. Often it is the RADIUS profile that specifies the pool to use and the Control Plane (CP) selects an address from that pool. If neither the address nor pool comes from the RADIUS, the PPP profile configuration (on the box) specifies which pool name to use. This profile is attached to the port identifier where the PPP packets are received.

The IPv6 address assignment occurs in two phases:

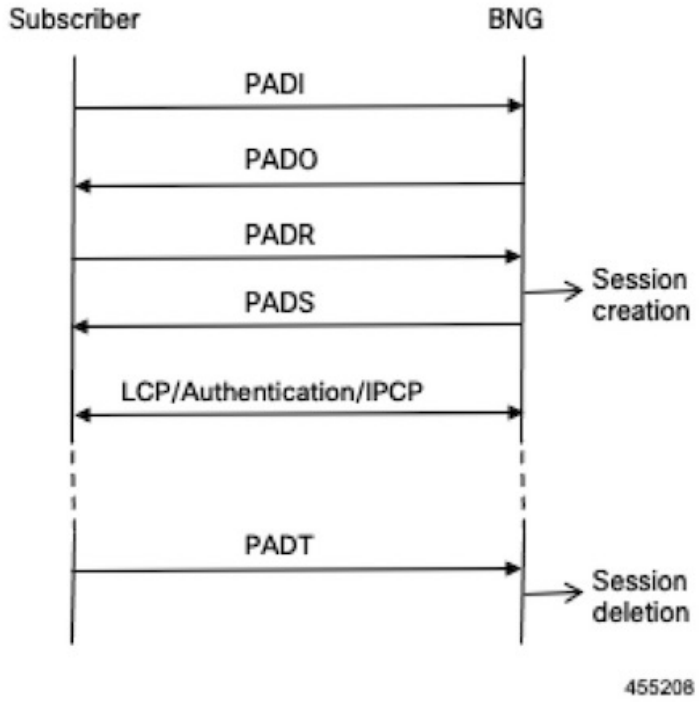
- First, as part of the IPv6CP, the interface-ID is negotiated with the CPE, which is used for link local negotiation.
- Second, after the CPE initiates the DHCPv6 protocol to get IPV6 IANA or IAPD (or both) address allocation, it gets the IPv6 address from either the RADIUS or from a pool.

How it Works

This section provides a brief of how the PPPoE Subscriber Management feature works.

PPPoE Handling

The PPPoE discovery-stage protocol consists of basic packet exchange between the subscriber and server (cnBNG). The following illustration displays the flow of events.



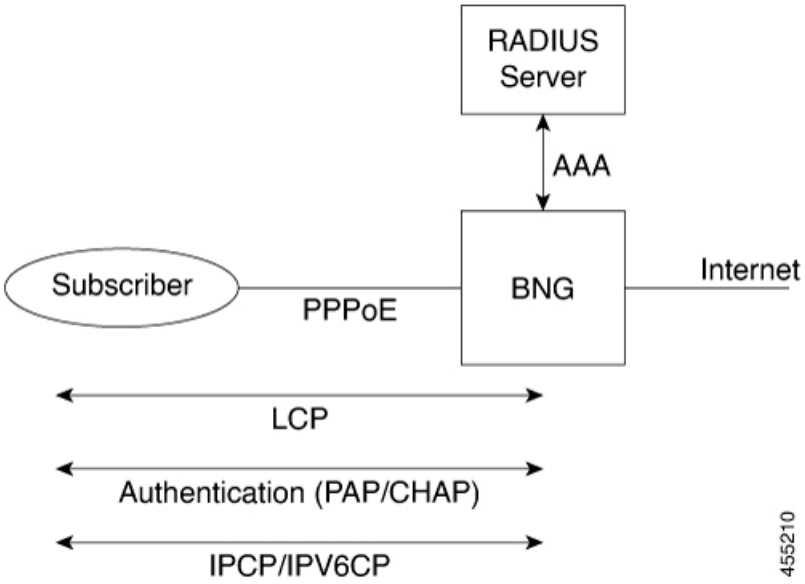
In brief, the protocol can be summarized as follows:

- When the subscriber wishes to establish a PPPoE session, it sends PADI message to the server.
  - The PADI may be multicast, if the subscriber tries to find out if any servers are available.
  - The PADI contains a Service-Name tag, which indicates the service that it wants the server to provide.
- When a server receives a PADI message, it checks if it can provide the service requested to the subscriber. If it can, it replies with a PADO message.
  - The PADO message is unicast to the peer. It contains the Service-Name the client requested.
- When the subscriber receives the PADO messages from the servers, it selects the server to connect to and sends a PADR message.
  - The PADR message is unicast, directed to the specific server with which it wants to establish a session.
  - The PADR message also contains the Service-Name tag.
- When the server receives a PADR message, it checks if it can provide the service to the subscriber.
  - If it can, it chooses a 16-bit Session-Id to identify the session of the subscriber and sets up the necessary state for the subscriber. It then replies with a PADS confirmation, which contains the Session-Id to indicate to the subscriber that the session is established.
  - If it cannot provide a session, it replies with a PADS containing an Error-tag, which indicates the reason it cannot. This PADS contains a zero Session-id.
- After the PADS is sent, the subscriber and server negotiate PPP in the standard way.
- When either the subscriber or the server wants to terminate the session, it sends PADT message to the peer with the Session-Id. This clears up all the states associated with the session.

  This completes the PPPoE discovery stage. the peers can now start the PPP negotiation.

PPP Handling

The network topology of the PPP is the point-to-point link between the BNG and the subscriber (this link is established during the PPPoE Discovery phase):



The PPPoE subscriber is viewed like any other PPP peer – LCP, Authentication and IPv4CP or IPv6CP (or both) are negotiated to establish the PPP link.

The standard scenario where the BNG terminates both the PPPoE and PPP subscriber session is referred to as PPP Termination and Aggregation (PTA). This distinguishes it from the more complex L2TP Access Concentrator (LAC) and L2TP Network Server (LNS) scenarios where the PPPoE is terminated locally on the BNG but the PPP session is terminated on a separate node from over L2TP to an upstream box known as an LNS.

Call Flows

This section includes the following high-level call flow.

PPPoE Bring-Up Call Flow

In cnBNG, the PPPoE and PPP Control Plane runs the overall PTA session bring-up, which includes the PPPoE and PPP negotiation as shown in the following call-flow.

Figure 24: cnBNG PPPoE Bring-Up Call Flow

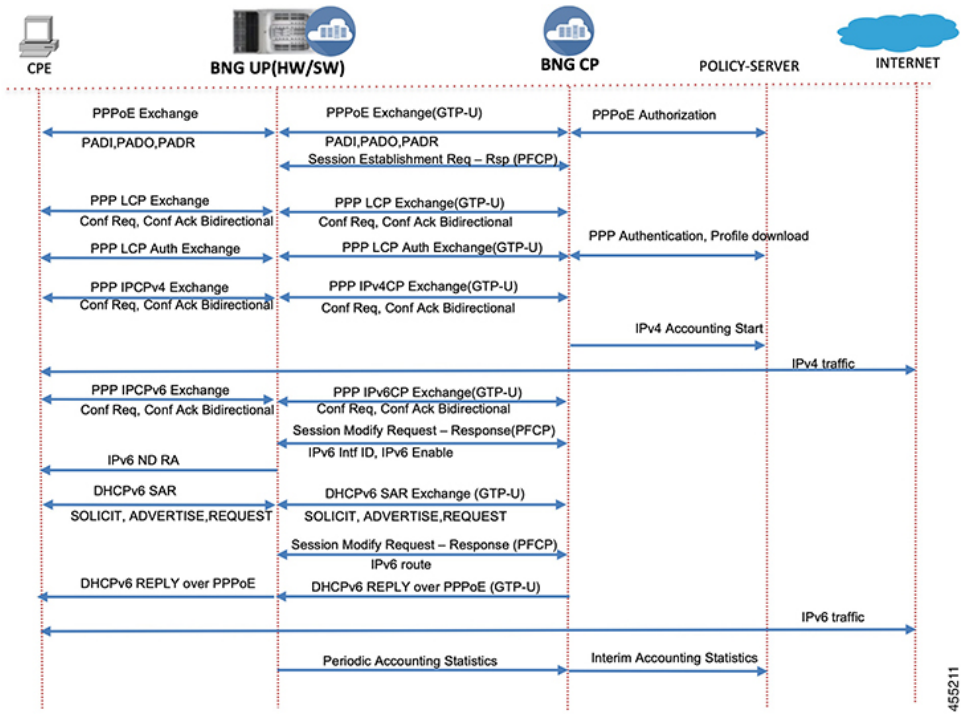


Table 95: cnBNG PPPoE Call Flow Description

Steps	Description
1	The CPE initiates the PPPoE packet exchange. The BNG-CP verifies the tags, session limits, service name, and so on and creates a PPPoE interface.
2	The BNG-CP sends a Success PADS message with an assigned PPPoE session ID.
3	The CPE and BNG-CP negotiate the LCP link parameters and authorization methods to use.
4	The BNG-CP authenticates the CPE with the provided username and password via AAA and downloads the network level parameters.
5	The CPE starts the IPv4CP and gets the IPv4 address. The BNG-CP programs the IPv4 route and features on the BNG-UP. Accounting start is initiated for IPv4.
6	Now bidirectional IPv4 traffic is enabled for the subscriber with the applied features.
7	Optionally, the CPE starts IPv6CP in case of dual stack. The local and peer interface ID are negotiated and the BNG-UP is programmed to allow link local negotiation to occur.
8	The BNG-UP completes the link local addressing with the IPv6 ND router advertisement.

Steps	Description
9	The CPE starts the DHCPv6 packet exchange on the negotiated PPPoE session to get the global IPv6 address assignment.
10	The BNG-CP programs the IPv6 routes and features into the BNG-UP and responds to the CPE with the DHCPv6 Reply packet to acknowledge that the IPv6 is up. At this stage, the session is converted into a dual stack in the CP.
11	The subscriber can now send and receive IPv6 traffic from the Internet.
12	The BNG-UP collects and pushes the interim statistics to the BNG-CP. The BNG-CP pushes these statistics to the Policy Plane for billing.

Standard Compliance

The PPPoE Subscriber Management feature is aligned with the following standards:

- RFC 1661 Point-to-Point Protocol
- RFC 2516. A Method for Transmitting PPP Over Ethernet (PPPoE)

Limitations

The PPPoE Subscriber Management feature has the following limitations:

- Session throttling is not supported
- The PPPoE profile and PPP feature template configuration changes are applied only to the new sessions. These changes are not applied to the existing sessions.
- Update of PPP features via CoA is not supported.
- In deployment, it is not expected that one address family interface (AFI) of session goes down and another AFI bring-up is inflight. This is not supported. For example, the following sequence is not supported.
  1. Dual-stack is UP.
  2. IPv4 release is received (Session becomes IPv6 only).
  3. IPv4 Discover is received (Session is IPv6 only but IPv4 is inflight).
  4. IPv6 release is received (Session becomes IPv4 is inflight).
  5. IPv4 Request is receive (Session is IPv4 only).

     In Step 3, IPv4 session is inflight and in step 4, IPv6 is going down, and in step 5, IPv4 session bring-up is complete. Such sequence of events is not expected in deployment and is not supported.

Configuring the PPPoE Subscriber Management Feature

This section describes how to configure the PPPoE Subscriber Management feature.

Configuring the PPPoE Subscriber Management feature involves the following steps:

1. Creating the PPPoE profile
2. Creating the PPP Feature template

Creating PPPoE Profile

Use the following commands to create a PPPoE profile and provide the PPPoE protocol specific parameters.

```
config
  profile pppoe pppoe_profile_name
  mtu mtu
  service-selection-disable [ true | false ]
  max-payload minimum { payload_value } maximum { payload_value }
  service-name service_name
  ac-name ac_name
  ac-cookie ac-cookie_name
  session max limit { count } threshold { count }
  session mac limit { count } threshold { count }
  session circuit-id limit { count } threshold { count }
  session outer-vlan limit { count } threshold { count }
  timeout-completion period
  control-packets priority cos_value
  exit
```

NOTES:

- **profile pppoe** *pppoe_profile_name*: Specifies the PPPoE profile name.
- **mtu** *mtu*: Specifies the default PPP maximum transmission unit (MTU) value to use if the Max-Payload tag is not provided. The valid values range from 500 to 2000. The default value is 1492.
- **service-selection-disable [true | false]**: Enables or disables the advertising of extra service names in the PADO packets. True enables the service and false disables the service. The default value is false.
- **max-payload minimum {** *payload_value* **} maximum {** *payload_value* **}**: Specifies the supported PPPoE service name. Multiple service names can be configured simultaneously. The valid value is an alphanumeric string ranging from 1 to 256. All service names are accepted.
- **service-name** *service_name*: Specifies the supported PPPoE service name. Multiple service names can be configured simultaneously. The valid value is an alphanumeric string ranging from 1 to 256. All service names are accepted.
- **ac-name** *ac_name*: Specifies the access concentrator (AC) to use in the PADO packets. The valid value is an alphanumeric string ranging from 1 to 256. The default ac-name is the router hostname.
- **ac-cookie** *ac-cookie_name*: Specifies the AC-Cookie to use in the PADO packets. The valid value is an alphanumeric string ranging from 1 to 256.
- **session max limit {** *count* **} threshold {** *count* **}**: Specifies the total maximum number of sessions and threshold allowed per User Plane per profile. The valid values range from 1 to 65535. The default value is 65535.
- **session mac limit {** *count* **} threshold {** *count* **}**: Specifies the maximum number of sessions and threshold allowed per UP per peer profile. The valid values range from 1 to 65535. The default value is 65535.

  When the threshold is passed, a syslog is printed as a warning.

- **session circuit-id limit {** *count* **} threshold {** *count* **}**: Specifies the maximum number of sessions and threshold allowed per circuit-id. The valid values range from 1 to 65535. The default value is 65535.

  When the threshold is passed, a syslog is printed as a warning.

- **session outer-vlan limit {** *count* **} threshold {** *count* **}**: Specifies the maximum number of sessions and threshold allowed per UP per peer profile. The valid values range from 1 to 65535. The default value is 65535.

  When the threshold is passed, a syslog is printed as a warning.

- **timeout-completion** *period*: Specifies the maximum time to wait for the session to be completed (an NCP to come up for PTA sessions or the L2TP tunnel to be setup for LAC sessions) before terminating the session. The valid values range from 30 to 600 seconds. The default value is 120 seconds.

- **control-packets priority** *cos_value*: Specifies the CoS to use in the PADx packets. The valid values range from 0 to 7. The default CoS bits are used.

Creating the PPP Feature Template

Use the following commands to create a PPP feature template.



Note The PPP feature template allows per subscriber PPP parameters.

```
config
   profile feature-template feature_template_name
   ppp
      authentication { chap  | pap }
      chap hostname chap_hostname
      chap password chap_password
      ipcp dns ipv4_address
      ipcp peer-address-pool ipam_pool_name
      ipcp renegotiation ignore
      ipcp wins ipv4_address
      ipcpv6 renegotiation ignore
      ipcp wins ipv4_address
      max-bad-auth count
      max-configure count
      max-failure count
      pap accept-null-password
      timeout absolute seconds
      timeout authentication seconds
      timeout retry seconds
      keepalive interval secondsretryseconds[ disable ]
      exit
```

NOTES:

- **profile feature-template** *feature_template_name*: Specifies the profile feature template name.
- **ppp**: Enters the PPP Configuration mode to configure the PPP feature.

- **authentication { chap | pap }**: Specifies the authentication type as CHAP or PAP.
- **chap hostname** *chap_hostname*: Specifies the hostname to use for CHAP authentication. The valid values range from 1 to 64. The default value is the router hostname.
- **chap password** *chap_password*: Specifies the password to use for CHAP authentication.
- **ipcp dns** *ipv4_address*: Specifies the DNS address to use for the peer.
- **ipcp peer-address-pool** *ipam_pool_name*: Specifies the address pool to use to obtain an IPv4 address for the peer.
- **ipcp renegotiation ignore**: Specifies to ignore the attempts of the peer to renegotiate IPCP. The entire PPPoE session is terminated on renegotiation.
- **ipcp wins** *ipv4_address*: Specifies the Windows Internet Name Service (WINS) address to use for the peer.
- **max-bad-auth** *count*: Specifies the maximum authentication failures to allow. The valid values range from 0 to 10. The default value is 0.
- **max-configure** *count*: Specifies the maximum number of Conf-Reqs to send without a response. The valid values range from 4 to 20. The default value is 10.
- **max-failure** *count*: Specifies the maximum number of Conf-Naks to send. The valid values range from 2 to 10. The default value is 5.
- **pap accept-null-password**:Accepts the null password feature for PAP.
- **max-failure** *count*: Specifies the maximum number of Conf-Naks to send. The valid values range from 2 to 10. The default value is 5.
- **timeout absolute** *seconds*: Specifies the absolute timeout for a PPP session. The valid values range from 0 to 70000000 minutes.
- **timeout authentication** *seconds*: Specifies the total time to allow for authentication to complete. The valid values range from 3 to 30 seconds. The default value is 10.
- **timeout retry** *seconds*: Specifies the maximum time to wait for a response to a Conf-Req. The valid values range from 1 to 10 seconds. The default value is 3.
- **keepalive interval** *seconds* **retry** *seconds* **[disable]**: Specifies the keepalive interval and the retry attempts for the subscribers. The valid values range from 10 to 120 seconds for the keepalive interval. The default is 60 seconds. The valid values range from 1 to 255 for the retry attempt. The default value is 5 counts.

View PPPoE or LNS session event history

Examine the recent event history for a specific PPPoE or LNS session to assist with troubleshooting and monitoring.

Procedure

Step 1 Use the **show subscriber pppoe [lns] sublabel** *sublabel* **debug** command to view detailed debug information for a specific PPPoE subscriber session.

Example:

```
bng# show subscriber pppoe sublabel 16777219 debug

Fri Oct  17 06:12:26.852 UTC+00:00
subscriber-details
{
  "subResponses": [
    {
      "state": "complete",
      "key": {
        "routerID": "asr9k-1",
        "portID": "GigabitEthernet0/0/0/1",
        "outerVlan": 100,
        "innerVlan": 200,
        "macAddr": "cc11.0000.0001",
        "pppoesessionID": 3,
        "subLabel": "16777219",
        "upSubID": "3",
        "HostUniq": "0x686f7374756e71",
        "circuitID": "circuit1",
        "remoteID": "remote1"
      },
      "flags": [
        "SM_START_DONE",
        "SM_ACTIVATE_DONE",
        "SM_UPDATE_DONE",
        "PPPOE_UP_DONE",
        "IPCP_UP",
        "IPV6CP_UP"
      ],
      "pppoeInfo": {
        "profileName": "pppoe-prof1",
        "mtu": 1500
      },
      "lcpInfo": {
        "state": "opened",
        "keepAliveInterval": 60,
        "keepAliveRetries": 5,
        "localMru": 1500,
        "peerMru": 1500,
        "localMagic": "0xf936c66e",
        "peerMagic": "0x112233",
        "authOption": "PAP",
        "authCompleted": true,
        "username": "cnbng"
      },
      "ipcpInfo": {
        "state": "opened",
        "peerIpv4Pool": "pool-ISP",
        "peerIpv4Address": "11.0.160.3",
        "peerIpv4Netmask": 22,
        "localIpv4Address": "11.0.160.1",
        "isIpamPoolIPaddr": true
      },
      "ipv6cpInfo": {
```

```
        "state": "opened",
        "localIntfID": "0x1",
        "peerIntfID": "0xcc11000000010000"
      },
      "sessionType": "pta",
      "vrf": "default",
      "AuditId": 4,
      "sess-events": [
        "Time, Event, Status",
        "2025-10-17 06:12:14.933636323 +0000 UTC, Session-ID Alloc, complete",
        "2025-10-17 06:12:14.94356763 +0000 UTC, UP Interface Create, complete",
        "2025-10-17 06:12:14.963876513 +0000 UTC, SM Session-Start, complete",
        "2025-10-17 06:12:15.978631055 +0000 UTC, LCP, up",
        "2025-10-17 06:12:16.016685284 +0000 UTC, SM Session-Activate, complete",
        "2025-10-17 06:12:16.06998785 +0000 UTC, IPCP, up",
        "2025-10-17 06:12:16.094605971 +0000 UTC, IPv6CP, up"
      ]
    }
  ]
}
```


```
bng# show subscriber lns sublabel 16777219 debug
```


```
Mon Oct  20 06:30:27.240 UTC+00:00
subscriber-details
{
  "subResponses": [
    {
      "state": "complete",
      "key": {
        "routerID": "asr9k-1",
        "portID": "GigabitEthernet0/1/0/1",
        "subLabel": "16777218",
        "upSubID": "1"
      },
      "flags": [
        "SM_START_DONE",
        "SM_ACTIVATE_DONE",
        "SM_UPDATE_DONE",
        "IPCP_UP",
        "IPV6CP_UP"
      ],
      "lcpInfo": {
        "state": "opened",
        "keepAliveInterval": 60,
        "keepAliveRetries": 5,
        "localMru": 1492,
        "peerMru": 1492,
        "localMagic": "0xe2ab84f",
        "peerMagic": "0xe2ab850",
        "authOption": "PAP",
        "authCompleted": true,
        "username": "cnbng"
      },
      "ipcpInfo": {
        "state": "opened",
        "peerIpv4Pool": "pool-ISP",
        "peerIpv4Address": "11.0.32.2",
        "peerIpv4Netmask": 22,
        "localIpv4Address": "11.0.32.1",
        "isIpamPoolIPaddr": true
      },
      "ipv6cpInfo": {
        "state": "opened",
        "localIntfID": "0x1",
```

```
        "peerIntfID": "0xcc11000000010001"
      },
      "lnsInfo": {
        "srcIP": "10.1.39.139",
        "dstIP": "10.1.34.52",
        "state": "established",
        "profileName": "l2tp-prof2",
        "tunnelClientAuthID": "bng-lac",
        "tunnelServerAuthID": "Local-DC",
        "callSerialNumber": 16777408,
        "localTunnelID": 12226,
        "localSessionID": 11428,
        "remoteTunnelID": 1017,
        "remoteSessionID": 24885
      },
      "sessionType": "lns",
      "vrf": "default",
      "AuditId": 4,
      "sess-events": [
        "Time, Event, Status",
        "2025-10-20 06:30:01.966696374 +0000 UTC, Sublabel Alloc, complete",
        "2025-10-20 06:30:01.971563927 +0000 UTC, L2TP-ID Alloc, complete",
        "2025-10-20 06:30:01.986624734 +0000 UTC, UP Interface Create, complete",
        "2025-10-20 06:30:02.010855854 +0000 UTC, SM Session-Start, complete",
        "2025-10-20 06:30:02.038466759 +0000 UTC, L2TP session, complete",
        "2025-10-20 06:30:02.038504381 +0000 UTC, LCP, up",
        "2025-10-20 06:30:02.07193459 +0000 UTC, SM Session-Activate, complete",
        "2025-10-20 06:30:02.102904338 +0000 UTC, IPv6CP, up",
        "2025-10-20 06:30:03.202555529 +0000 UTC, IPCP, up"
      ]
    }
  ]
}
```


Step 2 (Optional) Use the **subscriber event-trace-disable** command to disable event tracing.

Step 3 Use the **subscriber event-trace-max-count** *number* to set or adjust the maximum number of events retained in the history log for sessions.

Stateless Address Autoconfiguration (SLAAC)

Table 96: Feature History

Feature Name	Release Information	Description
Stateless Address Autoconfiguration (SLAAC)	2024.04.0	This feature allows each IPv6 host to generate its own address using local and router-advertised information, simplifying the integration of new IPv6 hosts without extensive configuration.

Stateless Address Autoconfiguration (SLAAC) allows each IPv6 host in a network to generate their own address using a combination of local and router-advertised information. This process is lightweight and does not require a server to track assigned addresses or their states.

Defined in RFC 4862, SLAAC simplifies the integration of new IPv6 hosts into a network without extensive configuration. It also facilitates the migration of existing IPv4 hosts to IPv6 networks.

This DHCPv6 protocol is a stateful counterpart to IPv6 SLAAC, and can be used separately, or concurrently with SLAAC, to obtain configuration parameters.

IPv6 Address Assignment

The Customer Premises Equipment (CPE) or Residential Gateway (RG) is assigned a /64 IPv6 prefix from the Broadband Network Gateway (BNG). Using this prefix, the CPE/RG generates its 128-bit global IPv6 address for the WAN side link.

Configuration Flags Usage

SLAAC relies on the ICMPv6 protocol to advertise the IPv6 prefix to hosts. When an IPv6-enabled host is detected in the network, the IPv6 router sends an ICMPv6 Router Advertisement (ICMPv6.Type = 134) packet. This packet contains the /64 prefix and other necessary information such as MTU and MAC address of the interface. It also includes specific flags relevant to the overall IPv6 enablement of the host:

- **Managed Address Configuration Flag (M flag)**

  When set to 1 (on), the router instructs the host to use a stateful DHCPv6 server for its global unicast address and all other addressing information. If the M flag is set, the O flag can be ignored because the DHCPv6 server returns all available information.

- **Other Configuration Flag (O flag)**

  When set to 1 (on), the DNS information is obtained from a stateless DHCPv6 server. The router instructs the nodes to auto-configure an address using SLAAC and to request DNS information from the DHCPv6 server.

- If neither the M flag nor the O flag is set, this indicates that no DHCPv6 server is available on the segment.

The M and O flags provide the host with a comprehensive view of how it can enable its IPv6 configuration. This includes prefix and DNS information, using SLAAC protocol. This flexibility ensures that hosts can be seamlessly integrated into the IPv6 network with minimal configuration effort.

Configure SLAAC for PPPOE

Procedure

Step 1 Use the following sample configuration to configure SLAAC for PPPOE.

Example:

```
ipam
   instance 1
      source local
      address-pool pool-ipv6
          vrf-name abc
          ipv6
             prefix-ranges
                 split-size
                    per-cache 512
                    per-dp    512
                 exit
             prefix-range 2001:DB8:: length 48
          exit
```

```
      exit
   exit
exit
profile slaac slaac-prof1
    managed-config-flag enable
    other-config-flag enable
    prefix-pool pool-ipv6
exit
profile subscriber subs-prof1
    pppoe-profile pppoe-prof1
    slaac-profile slaac-prof1
exit
user-plane
    instance 1
       user-plane asr9k-1
          peer-address ipv4 10.1.1.1
          subscriber-profile subs-prof1
       exit
    exit
exit
```

NOTES:

- **profile slaac** *slaac_profile_name*: Specifies the SLAAC profile name.
- **prefix-pool** *slaac_prefix_name*: Specifies the /64 IPv6 PD prefix pool to allocate SLAAC prefix to subscribers.
- **managed-config-flag enable**: Enables M-bit for IPv6 RA packets.
- **other-config-flag enable**: Enables O-bit for IPv6 RA packets.
- **profile subscriber** *subscriber_profile_name*: Configures subscriber profiles.
- **pppoe-profile** *pppoe_profile_name*: Specifies the PPPOE-FSOL profile name.
- **slaac-profile** *slaac_profile_name*: Specifies the SLAAC-FSOL profile name.

Step 2 Use the **show subscriber pppoe detail** command to view the PPPoE sessions with SLAAC prefix.

Example:

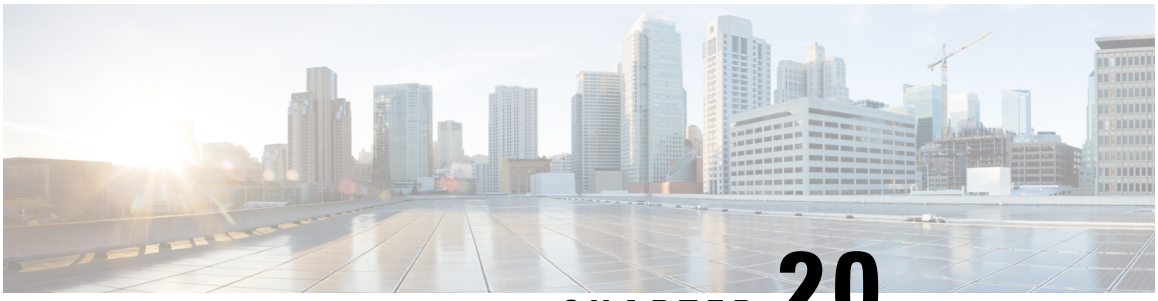
```
bng# show subscriber pppoe detail
Tue Oct  1  13:30:17.888 UTC+00:00
subscriber-details
{
  "subResponses": [
    {
      "state": "complete",
      "key": {
        "routerID": "asr9k-1",
        "portID": "Bundle-Ether1.1",
        "outerVlan": 100,
        "macAddr": "0011.9400.ab01",
        "pppoesessionID": 13200,
        "sublabel": "17321517",
        "upSubID": "2148432560",
       },
      "slaacInfo": {
        "prefix": "3001:ab:0:bc::",
        "prefixlength": 64,
        "poolname": "pool-ipv6",
        "fsmstate": "connected",
        "profilename": "slaac-prof",
```

```
        "otherconfig": true
      }
    }
},
```

Step 3 Use the **show subscriber session detail** command to view the Session Manager (SM) sessions with SLAAC prefix.

Example:

```
bng# show subscriber session detail
Tue Oct  1  13:37:05.476 UTC+00:00
subscriber-details
{
  "subResponses": [
    {
      "subLabel": "17321517",
      "mac": "0011.9400.ab01",
      "acct-sess-id": "01085a5d",
      "sesstype": "ppp",
      "state": "established",
      "subCreateTime": "Tue, 01 Oct 2024 04:56:05 UTC",
      "pppAuditId": 4,
      "transId": "3",
      "subsAttr": {
        "attrs": {
          "port-type": "Virtual PPPoE over VLAN(36)",
          "pppoe-session-id": "13200",
          "prefix": "3001:ab:0:bc::/64",
          "protocol-type": "ppp(2)",
        }
      },
   }
}
```



CHAPTER 20

Rolling Software Update

- Feature Summary and Revision History, on page 419
- Feature Description, on page 420
- How it Works, on page 421
- Installing the Rolling Software Update, on page 422

Feature Summary and Revision History

Summary Data

Table 97: Summary Data

Applicable Product(s) or Functional Area	cnBNG
Applicable Platform(s)	SMI
Feature Default Setting	Disabled - Configuration Required
Related Documentation	• Multiple Replica Support for cnBNG Services • High Availability and CP Reconciliation

Revision History

Table 98: Revision History

Revision Details	Release
Rolling software upgrade support for LAC and LNS sessions.	2022.03.0
First introduced.	2021.04.0

Feature Description

The cnBNG Rolling Software Update feature enables incremental update of pod instances with minimal downtime. In Kubernetes (K8s), this implementation is possible only with rolling updates.

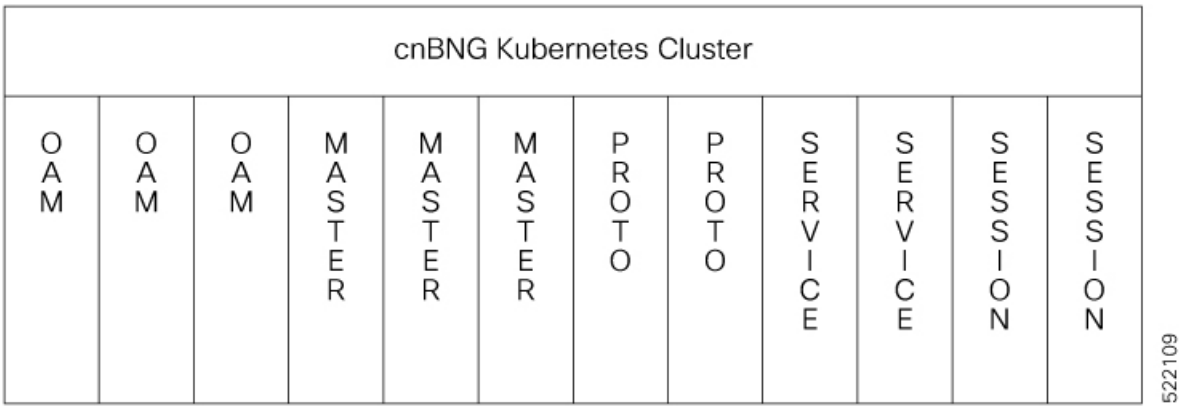
Subscriber Microservices Infrastructure (SMI) platform supports rolling software upgrade for cnBNG pods. The "Pod Restart and Reconciliation" and "Multiple Replica Support for cnBNG Services" features depend on this feature. For more information, see Multiple Replica Support for cnBNG Services and High Availability and CP Reconciliation.

The cnBNG has a three-tier architecture consisting of Protocol, Service, and Session tiers. Each tier includes a set of microservices (pods) for a specific functionality. Within these tiers, there exists a Kubernetes Cluster comprising K8s master and worker nodes (including Operation and Management nodes).

For high availability (HA) and fault tolerance, cnBNG requires a minimum of two K8s worker nodes for each tier. Each worker node can have multiple replicas. K8s orchestrates the pods using the StatefulSets controller. The pods require a minimum of two replicas for fault tolerance.

The following figure depicts the cnBNG K8s Cluster with 12 nodes – three Master nodes, three Operations and Management (OAM) worker nodes, two Protocol worker nodes, two Service worker nodes, and two Session (data store) worker nodes.

Figure 25: cnBNG Kubernetes Cluster



Note

- OAM worker nodes—Host the Ops Center pods for configuration management and metrics pods for statistics and Key Performance Indicators (KPIs).
- Protocol worker nodes—Host the cnBNG protocol-related pods for UDP-based interfaces such as N4, RADIUS, and GTP.
- Service worker nodes—Host the cnBNG application-related pods that perform session and FSOL management.
- Session worker nodes—Host the database-related pods that store subscriber session data.

Rolling Upgrade Support for LAC and LNS

cnBNG supports the rolling upgrade procedure for LAC and LNS sessions in addition to PTA and IPoE sessions. Rolling upgrade for LAC and LNS clusters is supported without impacting the existing tunnels by modifying the L2TP service with an initial readiness delay of 60 seconds. The K8s infrastructure upgrades an instance of L2TP pod only after the active instance is successfully upgraded and resynchronizes all existing tunnel data in the peer L2TP instance.

If more than one L2TP instance exists, rolling upgrade updates only one instance at a time. The stateful set pairing must be done accordingly to avoid peer L2TP instance going down at the same time.

How it Works

This section describes how the cnBNG Rolling Software Update works.

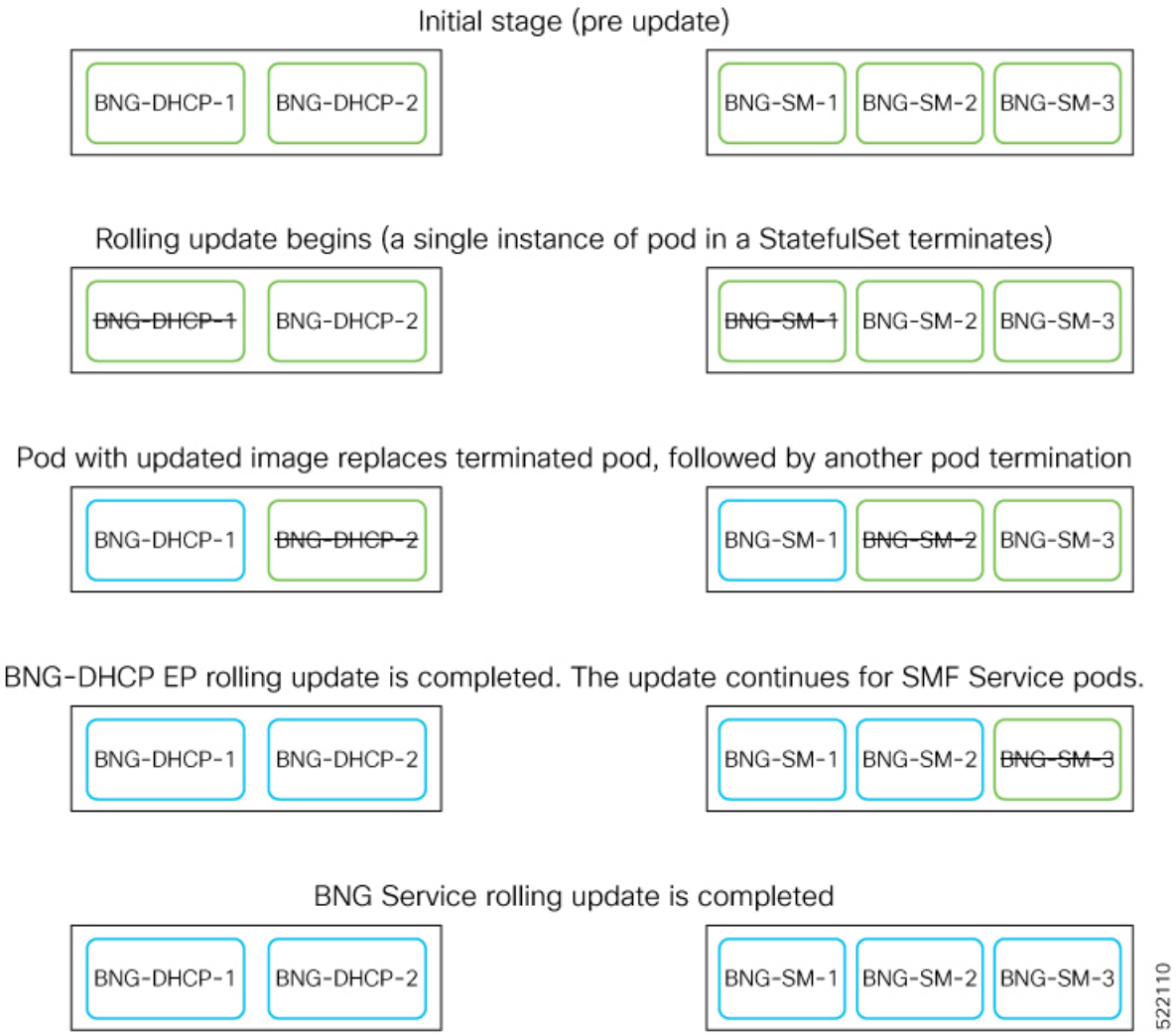
Rolling Software Update Using SMI Cluster Manager

Rolling software upgrade is a process of upgrading or migrating the build from an older to a newer version or upgrading the patch for the prescribed deployment set of application pods.

The cnBNG software update or in-service update procedure utilizes the K8s rolling strategy to update the pod images. In K8s rolling update strategy, the pods of a StatefulSet are updated sequentially to ensure that the ongoing process continues. Initially, a rolling update on a StatefulSet causes a single pod instance to terminate. A pod with an updated image replaces the terminated pod. This process continues until all the replicas of the StatefulSet are updated. The terminating pods exit gracefully after completing all the ongoing processes. Other in-service pods continue to receive and process the traffic with minimal impact. Use the Ops Center CLI. to control the software update process.

The following figure illustrates an cnBNG rolling update for bng-dhcp and bng-sm endpoint pods (two replicas) on Protocol worker nodes along with cnBNG Service pods (three replicas) on Service worker nodes.

Figure 26: cnBNG Rolling Update



Installing the Rolling Software Update

This section describes how to install the cnBNG Rolling Software Update feature.

The Rolling Software Update feature involves the following procedures.

Prerequisites

The prerequisites for installing the rolling software update for cnBNG are as follows:

- Ensure that all the nodes, including all the pods in the node, are up and running.
- Perform the cnBNG health check.
- Prepare for the upgrade.

- Backup the Ops Center configuration.
- Backup the CEE and BNG Ops Center configuration.
- Stage a new cnBNG image.



Attention Trigger rolling upgrade only when the CPU usage of the nodes is less than 50%.



Important Rolling upgrades are not recommended for major release upgrades. Additionally, rolling upgrades are not supported when the upgrade involves an SMI base ISO image.

Performing the cnBNG Health Check

Perform the cnBNG health check to ensure that all the services are running and nodes are in ready state. To perform an health check, log in to the master node and use the following configuration:

```
kubectl get pods -n smi
kubectl get nodes
kubectl get pod --all-namespaces -o wide
kubectl get pods -n bng-bng -o wide
kubectl get pods -n cee-global -o wide
kubectl get pods -n smi-vips -o wide
helm list -A
kubectl get pods -A | wc -l
```



Important Ensure that all the services are running and nodes are in ready state before proceeding further.



Note

- Static calls would not be impacted due to rolling upgrade.
- Inflight transactions and events will see failure during the rolling upgrade.
- For about 1-2 minutes downtime (that is, 100% transaction failures) is expected during upgrade with the suggested replica counts for 500K scale & 1000 CPS. Note: We can achieve zero downtime by increasing the number of replicas for each of the pods [dhcp, sm, pppoe], but that comes with the cost of additional resources.

Backing Up Ops Center Configuration

This section describes the procedure involved in creating a backup of the Ops Center configurations. To backup the Ops Center configurations:

1. Log in to SMI Cluster Manager node as an **ubuntu** user.

2. Run the following command to backup the SMI Ops Center configuration to **/home/ubuntu/smiops.backup** file.

   ```
   ssh -p <port_number> admin@$(kubectl get svc -n smi | grep
   '.*netconf.*<port_number>' | awk '{ print $4 }') "show run | nomore"
   > smiops.backup_$(date +'%m%d%Y_T%H%M')
   ```

3. Run the following command to backup the CEE Ops Center configuration to **/home/ubuntu/ceeops.backup** file.

   ```
   ssh admin@<cee-vip> "show run | nomore" > ceeops.backup_$(date
   +'%m%d%Y_T%H%M')
   ```

4. Run the following command to backup the BNG Ops Center configuration to **/home/ubuntu/bngops.backup** file.

   ```
   ssh admin@<bng-vip> "show run | nomore" > bngops.backup_$(date
   +'%m%d%Y_T%H%M')
   ```

Backing Up CEE and BNG Ops Center Configuration

This section describes the procedure involved in creating a backup of CEE and BNG Ops Center configuration from the master node. To perform a backup of CEE and BNG Ops Center configuration:

1. Log in to the master node as an **ubuntu** user.
2. Create a directory to backup the configuration files.

   ```
   mkdir backups_$(date +'%m%d%Y_T%H%M') && cd "$_"
   ```

3. Backup the BNG Ops Center configuration and verify the line count of the backup files.

   ```
   ssh -p <port_number> admin@$(kubectl get svc -n $(kubectl get namespaces
    | grep -oP 'bng-(\d+|\w+)') | grep <port_number> | awk '{ print $3
   }') "show run | nomore" > bngops.backup_$(date +'%m%d%Y_T%H%M') && wc
    -l bngops.backup_$(date +'%m%d%Y_T%H%M')
   ```

   Example:

   ```
   ubuntu@pobng-mas01:~/backups_09182019_T2141$ ssh -p 2024 admin@$(kubectl get svc -n
   $(kubectl get namespaces | grep -oP 'bng-(\d+|\w+)') | grep <port_number> | awk '{ print
    $3 }') "show run | nomore" > bngops.backup_$(date +'%m%d%Y_T%H%M') && wc -l
   bngops.backup_$(date +'%m%d%Y_T%H%M')
   admin@<ipv4address>'s password: bng-OPS-PASSWORD
   334 bngops.backup
   ```

4. Backup the CEE Ops Center configuration and verify the line count of the backup files.

   ```
   ssh -p <port_number> admin@$(kubectl get svc -n $(kubectl get namespaces
    | grep -oP 'cee-(\d+|\w+)') | grep <port_number> | awk '{ print $3
   }') "show run | nomore" > ceeops.backup_$(date +'%m%d%Y_T%H%M') && wc
    -l ceeops.backup_$(date +'%m%d%Y_T%H%M')
   ```

   Example:

   ```
   ubuntu@pobng-mas01:~/backups_09182019_T2141$ ssh -p <port_number> admin@$(kubectl get
   svc -n $(kubectl get namespaces | grep -oP 'cee-(\d+|\w+)') | grep <port_number> | awk
   '{ print $3 }') "show run | nomore" > ceeops.backup_$(date +'%m%d%Y_T%H%M') && wc -l
   ceeops.backup_$(date +'%m%d%Y_T%H%M')
   admin@<ipv4address>'s password: CEE-OPS-PASSWORD
   233 ceeops.backup
   ```

5. Move the SMI Ops Center backup file (from the SMI Cluster Manager) to the backup. directory.

   scp $(grep cm01 /etc/hosts | awk '{ print $1 }'):/home/ubuntu/smiops.backup_$(date +'%m%d%Y_T%H%M') .

   Example:

```
ubuntu@pobng-mas01:~/backups_09182019_T2141$ scp $(grep cm01 /etc/hosts | awk '{ print
$1 }'):/home/ubuntu/smiops.backup_$(date +'%m%d%Y_T%H%M') .
ubuntu@<ipv4address>'s password: SMI-CM-PASSWORD
smiops.backup                                                 100% 9346    22.3MB/s
   00:00
```

6. Verify the line count of the backup files.

   Example:

```
ubuntu@pobng-mas01:~/backups_09182019_T2141$ wc -l *
  233 ceeops.backup
  334 bngops.backup
  361 smiops.backup
  928 total
```

Staging a New cnBNG Image

The SMI Deployer downloads the new image and verifies it. Provide sha256 details in the "software cnf" section:

```
software cnf bng.2021.04.0.i96
url
https://engci-maven-master.cisco.com/artifactory/mobile-cnat-charts-release/releng-builds/2021.04.0/bng/2021.04.0.i96/bng.2021.04.0.i96-offline/bng.2021.04.0.i96.SSA.tgz
user testuser-deployer.gen
password $8$L1KSfQG9oMTkulzRxFjPTRsOH1O7S9qUVsLgDcFqJO4=
accept-self-signed-certificate true
sha256 d3a440be0e6080f2a83dc3d4e20121f2ceddadd0368a1d1bf41e567a397d35e0
exit
```

Performing Rolling Software Update

The cnBNG uses the SMI Cluster Manager to perform a rolling software update. To update cnBNG using the SMI Cluster Manager:

Important Ensure that cnBNG is up and running with the current version of the software.

1. Log in to SMI Cluster Manager Ops Center.
2. Update the product repository URL with the latest version of the product chart.

Note If the repository URL contains multiple versions, the Ops Center automatically selects the latest version.

```
config
  cluster cluster_name
   ops-centers app_name bng_instance_name
       repository-local local_repository
```

```
  exit
exit
```

Example:

```
SMI Cluster Manager# config
SMI Cluster Manager(config)# clusters test2
SMI Cluster Manager(config-clusters-test2)# ops-centers bng bng
SMI Cluster Manager(config-ops-centers-bng/bng)# repository-local <reference to the
locally downloaded image>
SMI Cluster Manager(config-ops-centers-bng/bng)# exit
SMI Cluster Manager(config-clusters-test2)# exit
```

3. Run the **cluster sync** command to update to the latest version of the product chart.

   clusters *cluster_name* **actions sync run**

   Example:

   ```
   SMI Cluster Manager# clusters test2 actions sync run
   ```



Important

- The cluster synchronization updates the BNG Ops Center, which in turn updates the application pods (through **helm sync** command) in sequence, automatically.
- When the rolling upgrade is in progress on a specific pod, the cnBNG avoids routing new calls to that pod.
- The cnBNG waits for 30 seconds before restarting the pod where rolling upgrade is initiated. Also, the cnBNG establishes all the in-progress calls completely within 30 seconds during the upgrade period (maximum call-setup time is 10 seconds).



Note

- **cluster** *cluster_name* –Specifies the name of the K8s cluster.
- **ops-centers** *app_name instance_name*– Specifies the product Ops Center and instance. *app_name* is the application name. *instance_name* is the name of the instance.
- **repository** *url*–Specifies the local registry URL for downloading the charts.
- **actions**–Specifies the actions performed on the cluster.
- **sync run**–Triggers the cluster synchronization.

Monitoring the Rolling Software Update

Use the following sample configuration to monitor the status of the Rolling Software Update using the SMI Cluster Manager Ops Center:

```
config
  clusters cluster_name actions sync run debug true
  clusters cluster_name actions sync logs
  monitor sync-logs cluster_name
```

```
clusters cluster_name actions sync status
exit
```

NOTES:

- **clusters** *cluster_name*–Specifies the information about the nodes to be deployed. *cluster_name* is the name of the cluster.
- **actions**–Specifies the actions performed on the cluster.
- **sync run**–Triggers the cluster synchronization.
- **sync logs**–Shows the current cluster synchronization logs.
- **sync status**–Shows the current status of the cluster synchronization. **debug true**–Enters the debug mode.
- **monitor sync logs**–Monitors the cluster synchronization process.

Example:

```
SMI Cluster Manager# clusters test1 actions sync run
SMI Cluster Manager# clusters test1 actions sync run debug true
SMI Cluster Manager# clusters test1 actions sync logs
SMI Cluster Manager# monitor sync-logs test1
SMI Cluster Manager# clusters test1 actions sync status
```


Important To view the pod details after the upgrade through CEE Ops Center, see Viewing the Pod Details, on page 427.

Viewing the Pod Details

Use the following sample configuration to view the details of the current pods through CEE Ops Center (in CEE Ops Center CLI):

```
cluster pods instance_name pod_name detail
```

NOTES:

- **cluster pods**–Specifies the current pods in the cluster.
- *instance_name*–Specifies the name of the instance.
- *pod_name*–Specifies the name of the pod.
- **detail**–Displays the details of the specified pod.

The following example displays the details of the pod named udp-proxy-0 in the bng-bng instance.

Example:

```
svi-cn-bng-tb4/global] cee# cluster pods bng-bng udp-proxy-0 detail
details apiVersion: "v1"
kind: "Pod"
metadata:
  annotations:
    prometheus.io/port: "8083"
    prometheus.io/scrape: "true"
    sidecar.istio.io/inject: "false"
  creationTimestamp: "2021-10-25T00:19:28Z"
```

```
generateName: "udp-proxy-"
labels:
  component: "udp-proxy"
  controller-revision-hash: "udp-proxy-5444cc5d74"
  instanceId: "1"
  release: "bng-bng-udp-proxy"
  statefulset.kubernetes.io/pod-name: "udp-proxy-0"
managedFields:
- apiVersion: "v1"
  fieldsType: "FieldsV1"
  fieldsV1:
    f:metadata:
      f:annotations:
        .: {}
        f:prometheus.io/port: {}
        f:prometheus.io/scrape: {}
        f:sidecar.istio.io/inject: {}
      f:generateName: {}
      f:labels:
        .: {}
        f:component: {}
        f:controller-revision-hash: {}
        f:instanceId: {}
        f:release: {}
        f:statefulset.kubernetes.io/pod-name: {}
      f:ownerReferences:
        .: {}
        k:{"uid":"914265b3-8b5b-4301-9433-c748e791c332"}:
          .: {}
          f:apiVersion: {}
          f:blockOwnerDeletion: {}
          f:controller: {}
          f:kind: {}
          f:name: {}
          f:uid: {}
    f:spec:
      f:affinity:
        .: {}
        f:nodeAffinity:
          .: {}
          f:requiredDuringSchedulingIgnoredDuringExecution:
            .: {}
            f:nodeSelectorTerms: {}
        f:podAntiAffinity:
          .: {}
          f:requiredDuringSchedulingIgnoredDuringExecution: {}
      f:containers:
        k:{"name":"udp-proxy"}:
          .: {}
          f:command: {}
          f:env:
            .: {}
            k:{"name":"APPLICATION_NAME"}:
              .: {}
              f:name: {}
              f:value: {}
            k:{"name":"CLUSTER_NAME"}:
              .: {}
              f:name: {}
              f:value: {}
            k:{"name":"COVERAGE_BUILD"}:
              .: {}
              f:name: {}
              f:value: {}
```

```
k:{"name":"CPS_PATCH"}:
  .: {}
  f:name: {}
k:{"name":"DATACENTER_NAME"}:
  .: {}
  f:name: {}
  f:value: {}
k:{"name":"ENABLE_ADD_DYNAMIC_BGP_ROUTE"}:
  .: {}
  f:name: {}
  f:value: {}
k:{"name":"ENABLE_RETRY_CONFIG"}:
  .: {}
  f:name: {}
  f:value: {}
k:{"name":"ENABLE_SGW_CACHE"}:
  .: {}
  f:name: {}
  f:value: {}
k:{"name":"ENABLE_TP_FEATURE"}:
  .: {}
  f:name: {}
  f:value: {}
k:{"name":"GOGC"}:
  .: {}
  f:name: {}
  f:value: {}
k:{"name":"GOMAXPROCS"}:
  .: {}
  f:name: {}
  f:value: {}
k:{"name":"GOTRACEBACK"}:
  .: {}
  f:name: {}
  f:value: {}
k:{"name":"GR_INSTANCE_ID"}:
  .: {}
  f:name: {}
  f:value: {}
k:{"name":"INFRA_ADMIN_PORT"}:
  .: {}
  f:name: {}
  f:value: {}
k:{"name":"INFRA_DIAG_PORT"}:
  .: {}
  f:name: {}
  f:value: {}
k:{"name":"INFRA_PROMETHEUS_PORT"}:
  .: {}
  f:name: {}
  f:value: {}
k:{"name":"INSTANCE_NODE_ID"}:
  .: {}
  f:name: {}
  f:value: {}
k:{"name":"IPC_EP_PORT"}:
  .: {}
  f:name: {}
  f:value: {}
k:{"name":"MY_POD_IP"}:
  .: {}
  f:name: {}
  f:valueFrom:
    .: {}
```

```
        f:fieldRef:
          .: {}
          f:apiVersion: {}
          f:fieldPath: {}
    k:{"name":"MY_POD_NAME"}:
      .: {}
      f:name: {}
      f:valueFrom:
        .: {}
        f:fieldRef:
          .: {}
          f:apiVersion: {}
          f:fieldPath: {}
    k:{"name":"PPROF_EP_PORT"}:
      .: {}
      f:name: {}
      f:value: {}
    k:{"name":"PROTOCOL_POD"}:
      .: {}
      f:name: {}
      f:value: {}
    k:{"name":"PROXY_KEEPALIVED_PORT"}:
      .: {}
      f:name: {}
      f:value: {}
    k:{"name":"SERVICE_NAME"}:
      .: {}
      f:name: {}
      f:value: {}
    k:{"name":"SMF_PROFILE_CONFIGURED"}:
      .: {}
      f:name: {}
      f:value: {}
  f:image: {}
  f:imagePullPolicy: {}
  f:name: {}
  f:readinessProbe:
    .: {}
    f:failureThreshold: {}
    f:initialDelaySeconds: {}
    f:periodSeconds: {}
    f:successThreshold: {}
    f:tcpSocket:
      .: {}
      f:host: {}
      f:port: {}
    f:timeoutSeconds: {}
  f:resources:
    .: {}
    f:limits:
      .: {}
      f:cpu: {}
      f:memory: {}
    f:requests:
      .: {}
      f:cpu: {}
      f:memory: {}
  f:terminationMessagePath: {}
  f:terminationMessagePolicy: {}
  f:volumeMounts:
    .: {}
    k:{"mountPath":"/config/udp-proxy/coverage"}:
      .: {}
      f:mountPath: {}
```

```
        f:name: {}
        f:readOnly: {}
      k:{"mountPath":"/config/udp-proxy/flowcontrol"}:
        .: {}
        f:mountPath: {}
        f:name: {}
        f:readOnly: {}
      k:{"mountPath":"/config/udp-proxy/logging"}:
        .: {}
        f:mountPath: {}
        f:name: {}
        f:readOnly: {}
      k:{"mountPath":"/config/udp-proxy/system"}:
        .: {}
        f:mountPath: {}
        f:name: {}
        f:readOnly: {}
      k:{"mountPath":"/config/udp-proxy/vip-ip"}:
        .: {}
        f:mountPath: {}
        f:name: {}
        f:readOnly: {}
f:dnsPolicy: {}
f:enableServiceLinks: {}
f:hostNetwork: {}
f:hostname: {}
f:imagePullSecrets:
  .: {}
  k:{"name":"regcredbng"}:
    .: {}
    f:name: {}
f:restartPolicy: {}
f:schedulerName: {}
f:securityContext: {}
f:subdomain: {}
f:terminationGracePeriodSeconds: {}
f:volumes:
  .: {}
  k:{"name":"coverage-volume"}:
    .: {}
    f:configMap:
      .: {}
      f:defaultMode: {}
      f:items: {}
      f:name: {}
    f:name: {}
  k:{"name":"flowcontrol-volume"}:
    .: {}
    f:configMap:
      .: {}
      f:defaultMode: {}
      f:items: {}
      f:name: {}
      f:optional: {}
    f:name: {}
  k:{"name":"logging-volume"}:
    .: {}
    f:configMap:
      .: {}
      f:defaultMode: {}
      f:items: {}
      f:name: {}
    f:name: {}
  k:{"name":"system-volume"}:
```

```
          .: {}
          f:configMap:
            .: {}
            f:defaultMode: {}
            f:items: {}
            f:name: {}
          f:name: {}
        k:{"name":"vip-ip-volume"}:
          .: {}
          f:configMap:
            .: {}
            f:defaultMode: {}
            f:items: {}
            f:name: {}
          f:name: {}
  manager: "kube-controller-manager"
  operation: "Update"
  time: "2021-10-25T00:19:28Z"
- apiVersion: "v1"
  fieldsType: "FieldsV1"
  fieldsV1:
    f:status:
      f:conditions:
        k:{"type":"ContainersReady"}:
          .: {}
          f:lastProbeTime: {}
          f:lastTransitionTime: {}
          f:status: {}
          f:type: {}
        k:{"type":"Initialized"}:
          .: {}
          f:lastProbeTime: {}
          f:lastTransitionTime: {}
          f:status: {}
          f:type: {}
        k:{"type":"Ready"}:
          .: {}
          f:lastProbeTime: {}
          f:lastTransitionTime: {}
          f:status: {}
          f:type: {}
      f:containerStatuses: {}
      f:hostIP: {}
      f:phase: {}
      f:podIP: {}
      f:podIPs:
        .: {}
        k:{"ip":"208.208.208.21"}:
          .: {}
          f:ip: {}
      f:startTime: {}
  manager: "kubelet"
  operation: "Update"
  time: "2021-10-25T00:19:38Z"
name: "udp-proxy-0"
namespace: "bng-bng"
ownerReferences:
- apiVersion: "apps/v1"
  kind: "StatefulSet"
  blockOwnerDeletion: true
  controller: true
  name: "udp-proxy"
  uid: "914265b3-8b5b-4301-9433-c748e791c332"
resourceVersion: "1557892"
```

```
  uid: "d519c85b-baae-4131-925b-df46e72757ac"
spec:
  affinity:
    nodeAffinity:
      requiredDuringSchedulingIgnoredDuringExecution:
        nodeSelectorTerms:
        - matchExpressions:
          - key: "smi.cisco.com/vm-type"
            operator: "In"
            values:
            - "protocol"
    podAntiAffinity:
      requiredDuringSchedulingIgnoredDuringExecution:
      - labelSelector:
          matchExpressions:
          - key: "component"
            operator: "In"
            values:
            - "udp-proxy"
        topologyKey: "kubernetes.io/hostname"
  containers:
  - command:
    - "/usr/local/bin/run-app"
    env:
    - name: "GOGC"
      value: "200"
    - name: "GOTRACEBACK"
      value: "crash"
    - name: "GOMAXPROCS"
      value: "12"
    - name: "CPS_PATCH"
    - name: "SERVICE_NAME"
      value: "udp-proxy"
    - name: "INFRA_PROMETHEUS_PORT"
      value: "8083"
    - name: "INFRA_ADMIN_PORT"
      value: "8879"
    - name: "INFRA_DIAG_PORT"
      value: "8979"
    - name: "PPROF_EP_PORT"
      value: "8850"
    - name: "IPC_EP_PORT"
      value: "9004"
    - name: "PROXY_KEEPALIVED_PORT"
      value: "28000"
    - name: "ENABLE_RETRY_CONFIG"
      value: "true"
    - name: "COVERAGE_BUILD"
      value: "false"
    - name: "DATACENTER_NAME"
      value: "DC"
    - name: "CLUSTER_NAME"
      value: "Local"
    - name: "APPLICATION_NAME"
      value: "BNG"
    - name: "INSTANCE_NODE_ID"
      value: "0"
    - name: "GR_INSTANCE_ID"
      value: "1"
    - name: "SMF_PROFILE_CONFIGURED"
      value: "false"
    - name: "ENABLE_TP_FEATURE"
      value: "true"
    - name: "ENABLE_ADD_DYNAMIC_BGP_ROUTE"
```

```
      value: "true"
    - name: "ENABLE_SGW_CACHE"
      value: "true"
    - name: "PROTOCOL_POD"
      value: "bng-n4-protocol"
    - name: "MY_POD_IP"
      valueFrom:
        fieldRef:
          apiVersion: "v1"
          fieldPath: "status.podIP"
    - name: "MY_POD_NAME"
      valueFrom:
        fieldRef:
          apiVersion: "v1"
          fieldPath: "metadata.name"
    image:
"docker.10.81.103.113.nip.io/bng.2021.04.0.i105/mobile-cnat-cn/udp-proxy/rel-2021.04/udp_proxy:0.1.1-16b9200-fe3d3ad-27f9489"

    imagePullPolicy: "IfNotPresent"
    name: "udp-proxy"
    readinessProbe:
      failureThreshold: 3
      initialDelaySeconds: 6
      periodSeconds: 5
      successThreshold: 1
      tcpSocket:
        host: "127.0.0.1"
        port: 28000
      timeoutSeconds: 1
    resources:
      limits:
        cpu: "3"
        memory: "32Gi"
      requests:
        cpu: "3"
        memory: "8Gi"
    terminationMessagePath: "/dev/termination-log"
    terminationMessagePolicy: "File"
    volumeMounts:
    - mountPath: "/config/udp-proxy/logging"
      name: "logging-volume"
      readOnly: true
    - mountPath: "/config/udp-proxy/vip-ip"
      name: "vip-ip-volume"
      readOnly: true
    - mountPath: "/config/udp-proxy/system"
      name: "system-volume"
      readOnly: true
    - mountPath: "/config/udp-proxy/flowcontrol"
      name: "flowcontrol-volume"
      readOnly: true
    - mountPath: "/config/udp-proxy/coverage"
      name: "coverage-volume"
      readOnly: true
    - mountPath: "/var/run/secrets/kubernetes.io/serviceaccount"
      name: "kube-api-access-hn2p5"
      readOnly: true
  dnsPolicy: "ClusterFirstWithHostNet"
  enableServiceLinks: true
  hostNetwork: true
  hostname: "udp-proxy-0"
  imagePullSecrets:
  - name: "regcredbng"
  nodeName: "svi-cn-bng-tb4-proto2"
```

```
preemptionPolicy: "PreemptLowerPriority"
priority: 100000000
priorityClassName: "default-application"
restartPolicy: "Always"
schedulerName: "default-scheduler"
securityContext: {}
serviceAccount: "default"
serviceAccountName: "default"
subdomain: "udp-proxy-i1"
terminationGracePeriodSeconds: 30
tolerations:
- effect: "NoExecute"
  key: "node.kubernetes.io/not-ready"
  operator: "Exists"
  tolerationSeconds: 30
- effect: "NoExecute"
  key: "node.kubernetes.io/unreachable"
  operator: "Exists"
  tolerationSeconds: 30
volumes:
- configMap:
    defaultMode: 420
    items:
    - key: "logging"
      path: "logging.yaml"
    name: "infra-logging-conf"
  name: "logging-volume"
- configMap:
    defaultMode: 420
    items:
    - key: "endpointIp"
      path: "endpointIp.yaml"
    name: "udp-proxy-vip-ip-conf"
  name: "vip-ip-volume"
- configMap:
    defaultMode: 420
    items:
    - key: "system"
      path: "system.yaml"
    name: "infra-system-conf"
  name: "system-volume"
- configMap:
    defaultMode: 420
    items:
    - key: "flowcontrol"
      path: "flowcontrol.yaml"
    name: "udp-proxy-flowcontrol-conf"
    optional: true
  name: "flowcontrol-volume"
- configMap:
    defaultMode: 420
    items:
    - key: "coverage"
      path: "coverage.yaml"
    name: "udp-proxy-coverage-conf"
  name: "coverage-volume"
- name: "kube-api-access-hn2p5"
  projected:
    defaultMode: 420
    sources:
    - serviceAccountToken:
        expirationSeconds: 3607
        path: "token"
    - configMap:
```

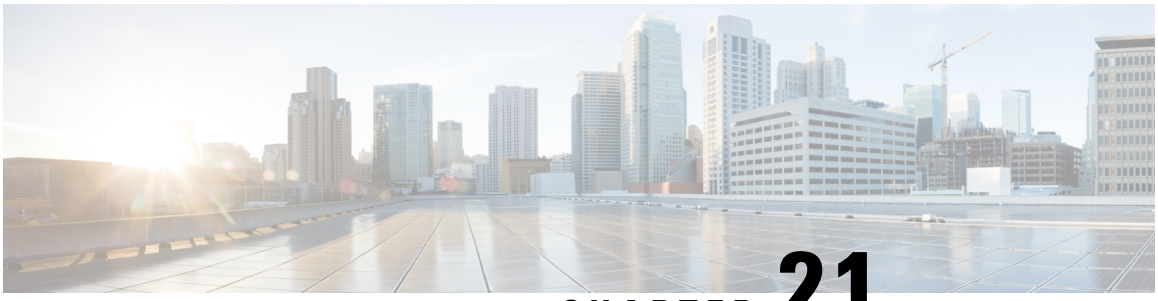
```
          items:
          - key: "ca.crt"
            path: "ca.crt"
          name: "kube-root-ca.crt"
      - downwardAPI:
          items:
          - fieldRef:
              apiVersion: "v1"
              fieldPath: "metadata.namespace"
            path: "namespace"
status:
  conditions:
  - lastTransitionTime: "2021-10-25T00:19:28Z"
    status: "True"
    type: "Initialized"
  - lastTransitionTime: "2021-10-25T00:19:38Z"
    status: "True"
    type: "Ready"
  - lastTransitionTime: "2021-10-25T00:19:38Z"
    status: "True"
    type: "ContainersReady"
  - lastTransitionTime: "2021-10-25T00:19:28Z"
    status: "True"
    type: "PodScheduled"
  containerStatuses:
  - containerID: "docker://9365e5d78de9e7edf427ee92f3aa7e74c4fdf5070874c89045079a1586199358"

    image:
"docker.10.81.103.113.nip.io/bng.2021.04.0.i105/mobile-cnat-cn/udp-proxy/rel-2021.04/udp_proxy:0.1.1-16b9200-fe3d3ad-27f9489"

    imageID:
"docker-pullable://docker.10.81.103.113.nip.io/bng.2021.04.0.i105/mobile-cnat-cn/udp-proxy/rel-2021.04/udp_proxy@sha256:aeb2651b0702990c25a8a1a9818c8bdb6a647e238f1d5e67af90c"

    lastState: {}
    name: "udp-proxy"
    ready: true
    restartCount: 0
    started: true
    state:
      running:
        startedAt: "2021-10-25T00:19:29Z"
  hostIP: "208.208.208.21"
  phase: "Running"
  podIP: "208.208.208.21"
  podIPs:
  - ip: "208.208.208.21"
  qosClass: "Burstable"
  startTime: "2021-10-25T00:19:28Z"

[svi-cn-bng-tb4/global] cee#
```



CHAPTER 21

Subscriber Manager

- Feature Summary and Revision History, on page 437
- Feature Description, on page 438
- Configuring Subscriber Manager Features, on page 440
- Automatic Session Reconciliation, on page 445
- Framed Route Support, on page 445
- Session disconnect history, on page 446
- Subscriber Accounting Functions, on page 461
- Subscriber QoS Policy, on page 463
- RADIUS-Based Policing - QoS Shape-Rate parameterization, on page 464
- Shared Policy Instance, on page 468

Feature Summary and Revision History

Summary Data

Table 99: Summary Data

Applicable Product(s) or Functional Area	cnBNG
Applicable Platform(s)	SMI
Feature Default Setting	Disabled - Configuration Required
Related Changes in this Release	Not Applicable
Related Documentation	*Cloud Native BNG Control Plane Command Reference Guide*

Revision History

Table 100: Revision History

Revision Details	Release
Introduced support for Radius-Based Policy (RaBaPol)	2025.02.0
Introduced support for Shared Policy Instance (SPI)	2025.02.0
Introduced support for session disconnect history.	2025.01.0
Enhancement Introduced: The Subscriber Manager feature is NSO-integrated.	2021.04.0
The following features are supported: • Automatic Session Reconciliation • Framed Route Support • Subscriber QoS Policy	2021.03.0
First introduced.	2021.01.0

Feature Description

Note This feature is Network Services Orchestrator (NSO) integrated.

In the Subscriber Manager (SM) context, a subscriber is a binding between the cnBNG Control Plane (CP) and a single subscriber end device. The SM is designed to provide a generic mechanism to connect edge subscribers to services enabling features. Subscribers are identified, authenticated, authorized, and accounted for in the SM.

Note The Subscriber Manager is also referred to as the Session Manager.

The following is a high-level list of the SM functionalities:

- Provides a generic mechanism for different Broadband Access Protocols such as DHCP and PPPoE.
- Provides an interface with off-box Radius servers using policy-plane to meet protocol and network provisioning requirements.
- Supports different subscriber lifecycle events such as CoA, idle timeout processing, and periodic reauthorization.

- Provides support for configuring subscriber lifecycle events that help customer define the subscriber behavior for the different subscriber lifecycle events.
- Derives per subscriber configuration from multiple sources.
- Maintains the subscriber state and subscriber configuration in a centralized session database.
- Interacts with the User Plane (UP) for subscriber session creation and subscriber feature configurations.

Subscriber features that are configured on cnBNG enable service providers to deploy certain specific functionalities like restricting the use of certain network resources, allowing Law Enforcement Agencies (LEAs) to conduct electronic surveillance, and so on.

Subscriber Features

The cnBNG supports the following subscriber features on the UP. For details, see the latest version of the Broadband Network Gateway Configuration Guide for Cisco ASR 9000 Series Routers listed here: https://www.cisco.com/c/en/us/support/routers/asr-9000-series-aggregation-services-routers/products-installation-and-configuration-guides-list.html.

- IPv4 or IPv6
  - Maximum Transmission Unit (MTU)
  - Unicast Reverse Path Forwarding (URPF)
  - Internet Control Message Protocol (ICMP)
- Access Control List (ACL)
  - Input ACL (IPv4 or IPv6)
  - Output ACL (IPv4 or IPv6)
- QoS (Quality of Service)
  - Input (policing)
  - Output (policing, shaping)
  - Policy merging (up to 6 policy maps and 10 class maps, including the default)
- Policy-based Routing (PBR)
  - Input policy (HTTP redirect)
- Accounting
  - Session Accounting
    - Periodic accounting
  - Service Accounting
    - Periodic accounting

To configure subscriber features, see Configuring Subscriber Manager Features, on page 440.

How it Works

This section provides a brief about how the Subscriber Manager works.

The SM functionality is hosted in a SM pod having one container in it. The SM pod communicates with the BNG Ops Center, policy-plane, and PFCP-EP pods using the APP infrastructure inter-pod communication (IPC).

The Subscriber Microservices Infrastructure (SMI) instantiates the SM pod. There can be more than one SM pod in the cluster. Each SM pod instance is independent. The per subscriber data is stored in a centralized database such that any SM pod can access this data.

Configuring Subscriber Manager Features

This section describes how to configure Subscriber Manager features on the CP.

The configuration of the Subscriber Manager features involves the following procedures:

- Configuring the HTTPR Policy Name, on page 440
- Configuring IPv4 Options, on page 441
- Configuring IPv6 Options, on page 441
- Configuring QoS Parameters
- Configuring the VRF Name, on page 442
- Configuring a Subscriber Profile, on page 443



Note

- To configure PPP feature options, see Creating the PPP Feature Template, on page 410
- To configure service accounting, see Configuring Service Accounting, on page 462
- To configure session accounting, see Configuring Session Accounting, on page 463

Configuring the HTTPR Policy Name

Use the following commands to configure the Policy Based Routing (PBR) HTTP Redirect (HTTPR) policy name.

```
config
  profile feature-template feature_template_name
  httpr-policy httpr_policy_name
  exit
```

NOTES:

- **profile feature-template** *feature_template_name*: Specifies the profile feature template name.
- **httpr-policy** *httpr_policy_name*: Specifies the PBR HTTPR policy name. The *httpr_policy_name* value can range from 1 to 128 characters.

Configuring IPv4 Options

Use the following commands to configure IPv4 options.

```
config
  profile feature-template feature_template_name
  ipv4
    disable-unreachables
    egress-acl string
    ingress-acl string
    mtu mtu_bytes
    verify-unicast-source reachable-via-rx
    exit
```

NOTES:

- **profile feature-template** *feature_template_name*: Specifies the profile feature template name.
- **ipv4**: Enters the IPv4 Configuration mode to configure the IPv4 features.
- **disable-unreachables**: Disables sending the Internet Control Message Protocol (ICMP) Unreachable messages.
- **egress-acl** *string*: Specifies the IPv4-based egress Access Control List (ACL) list. The supported length of the *string* ranges from 1 to 128 characters.
- **ingress-acl** *string*: Specifies the IPv4-based ingress ACL list. The supported length of the *string* ranges from 1 to 128 characters.
- **mtu** *mtu_bytes*: Specifies the maximum transmission unit (MTU). The supported *mtu_bytes* value can range from 68 to 65535 bytes.
- **verify-unicast-source reachable-via-rx**: Enables per packet validation for unicast. The source is reachable via the interface on which packet is received.

Configuring IPv6 Options

Use the following commands to configure IPv6 options.

```
config
  profile feature-template feature_template_name
  ipv6
    disable-unreachables
    egress-acl string
    ingress-acl string
    mtu mtu_bytes
    verify-unicast-source reachable-via-rx
    exit
```

NOTES:

- **profile feature-template** *feature_template_name*: Specifies the profile feature template name.
- **ipv6**: Enters the IPv6 Configuration mode to configure the IPv6 features.

- **disable-unreachables**: Disables sending the Internet Control Message Protocol (ICMP) Unreachable messages.
- **egress-acl** *string*: Specifies the IPv6-based egress Access Control List (ACL) list. The supported length of the *string* ranges from 1 to 128 characters.
- **ingress-acl** *string*: Specifies the IPv6-based ingress ACL list. The supported length of the *string* ranges from 1 to 128 characters.
- **mtu** *mtu_bytes*: Specifies the maximum transmission unit (MTU). The supported *mtu_bytes* value can range from 68 to 65535 bytes.
- **verify-unicast-source reachable-via-rx**: Enables per packet validation for unicast. The source is reachable via the interface on which packet is received.

Configuring QoS Parameters

Use the following commands to configure the Quality of Service (QoS) parameters.

```
config
  profile feature-template feature_template_name
  qos
    in-policy qos_input_policy_name
    merge-level integer
    out-policy qos_output_policy_name
    exit
```

NOTES:

- **profile feature-template** *feature_template_name*: Specifies the profile feature template name.
- **qos**: Enters the QoS Configuration mode to configure the parameters.
- **in-policy** *qos_input_policy_name*: Specifies the QoS input policy name. The supported length of the *qos_input_policy_name* ranges from 1 to 128 characters.
- **merge-level** *integer*: Enables or disables the merge level. A merge value of 0 disables the merge-level. Any value greater than 0, enables the merge level.
- **out-policy** *qos_output_policy_name*: Specifies the QoS output policy name. The supported length of the *qos_output_policy_name* ranges from 1 to 128 characters.

Configuring the VRF Name

Use the following commands to configure the virtual routing and forwarding (VRF) name.

```
config
  profile feature-template feature_template_name
  vrf-name vrf_name
  exit
```

NOTES:

- **profile feature-template** *feature_template_name*: Specifies the profile feature template name.

- **vrf-name** *vrf_name*: Specifies the VRF name. The supported length of the *vrf_name* ranges from 1 to 128 characters.

Configuring a Subscriber Profile

Use the following commands to create a subscriber profile.

```
configure
  profile subscriber subscriber_profile
    aaa { authenticate aaa_profile_for_authentication |
        authorize aaa_profile_for_authorization }
    activate-feature-template feature_template_name
    apply-all-class
    class class_name
      aaa aaa_profile_for_authentication | authorize aaa_profile_for_authorization
      activate-feature-template  feature_template_name
      matches
        match { protocol { dhcp | ppp } } | username { ascii
              ascii_string  | regex reg-exp string}
              | source-mac { ascii ascii_string
              | regex reg-exp string } |
              circuit-id { ascii  ascii_string
              | regex reg-exp string } |
              remote-id { ascii ascii_string
              | regex reg-exp string }
              match-type { all match { protocol | username |
              source-mac | circuit-id | remote-id } | any match {
              protocol | username | source-mac | circuit-id
              | remote-id } }
        exit
      dhcp-profile  dhcp_profile_name
      event session-activate { aaa { authenticate | authorize } |
                             activate-feature-templates
                             feature_templates_list
                             | apply-all-class | class class_name
                             | deactivate-feature-templates
                             feature_templates_list
      pppoe-profile pppoe_profile_name
      session-type { ipv4 | ipv4v6 | ipv6 }
      exit
```

NOTES:

- **profile subscriber** *subscriber_profile_name*: Specifies the profile subscriber name and enters the Profile Subscriber Configuraton mode.
- **aaa { authenticate** *aaa_profile_for_authentication* **| authorize** *aaa_profile_for_authorization* **}**: Specifies the AAA profile to associate for authentication and authorization.
- **activate-feature-templates** *feature_template_name*: Specifies the list of feature-templates in sequence for activation.
- **apply-all-class**: Applies all classes that are enabled.

- **class** *class_name* : Specifies the subscriber class name.
- **matches**: Enters the matches Configuration sub-mode to specify the match values.
  - **match { protocol { dhcp | ppp } | username { ascii** *ascii_string* **| regex** *reg-exp string* **} | source-mac { ascii** *ascii_string* **| regex** *reg-exp string* **} | { circuit-id { ascii** *ascii_string* **|regex** *reg-exp string* **} | remote-id { ascii** *ascii_string* **| regex** *reg-exp string* **} }**: Specifies the list of match values.
    - **match { protocol { dhcp | ppp }**: Specifies the match protocol as DHCP or PPP.
    - **username { ascii** *ascii_string* **| regex** *reg-exp string* **}**: Specifies the username in ascii format or regular express (reg-exp) string.
    - **source-mac { ascii** *ascii_string* **| regex** *reg-exp string* **}**: Specifies the source MAC address in ascii format or regular express (reg-exp) string.
    - **remote-id { ascii** *ascii_string* **| regex** *reg-exp string* **}**: Specifies the remote identifier in ascii format or regular express (reg-exp) string.
    - **circuit-id { ascii** *ascii_string* **| regex** *reg-exp string* **}**: Specifies the circuit identifier in ascii format or regular express (reg-exp) string.
    - **match-type { all match { protocol | username | source-mac | circuit-id | remote-id } | any match { protocol | username | source-mac | circuit-id | remote-id } }**: Specifies the match key and value for matching any or all of the options: protocol, username, source-mac, circuit-id, and remote-id.



Note By default **aaa, activate-feature-templates, apply-all-class**, and **class** are executed as part of the session bring-up. The PPPoE and DHCP access protocols use these events to create a subscriber in the SM. The operator may configure the AAA actions and activate-feature-templates, suitable for a subscriber.

- **dhcp-profile** *dhcp_profile_name*: Associates the DHCP first sign of life (FSOL) profile.
- **event session-activate**: Specifies the subscriber event to activate.

  Some Access-Protocols require a two-stage session bring up. For example with PPPoE subscribers, the PPPoE Access protocol will call the Session-Start event for FSOL followed by Session-Activate during PPP negotiation and authentication. The IPoE subscribers created by DHCP will not use this event. The operator may configure authenticate, authorize AAA actions and feature templates as suitable for a subscriber.
- **pppoe-profile** *pppoe_profile_name*: Associates the PPPoE FSOL profile.
- **session-type { ipv4 | ipv4v6 | ipv6 }**: Specifies the allowed session-types as IPv4, IPv4v6, and IPv6.

Automatic Session Reconciliation

Feature Description

The Automatic Session Reconciliation feature enables reconciliation of sessions that are out of synchronization between the Control Plane (CP) and User Plane (UP).

Desynchronization of a session occurs when the transaction is successful in the UP but times out before receiving a response from the UP.

The existing transaction-id increments by 1 in every request initiated from the CP to the UP. The CDL record stores the transaction-id per session when the UP conveys a successful response to the CP. The UP also stores this transaction-id when the transaction is successful in the UP.

How it Works

This section briefly describes how the Automatic Session Reconciliation feature works.

The UP validates the transaction-id received in every request from the CP. When a received transaction-id is not incremental to the transaction-id present in UP, the UP discards the transaction and responds to the CP with a transaction-id mismatch error.

On receiving the transaction-id mismatch error, the CP discards the current transaction and initiates a new transaction to replay the complete session data to the UP. After this session replay, the session reconciles and synchronizes automatically in the CP and UP.

Framed Route Support

Feature Description

The Framed Route Support on subscriber sessions enables framed (dynamic) routes to be added for individual subscribers. Framed route per subscriber support is provided through RADIUS or Change of Authorization (CoA).

A framed route is pushed from the Control Plane (CP) to the User Plane (UP) only when the IP address is allocated for the respective address family indicator (AFI). The UP withdraws the framed route when the respective AFI goes down (for example, when an IP address is deallocated).

The configuration format of the framed route is as follows:

- **IPv4**

```
Framed-Route = "[vrf <prefix VRF>] {<prefix>/<prefix_length>} [vrf <next hop vrf>]
                [<next hop prefix>] [<metric>] [tag <tag-value>] "
Framed-Route = "[vrf <vrf-name>] {<prefix> <netmask>} [vrf <next hop vrf>]
                [<next hop prefix>]
                [<metric>] [tag <tag-value>] "


cisco-avpair = "[vrf <prefix VRF>] {<prefix>/<prefix_length>} [vrf <next hop vrf>]
```

```
                [<next hop prefix>] [<metric>] [tag <tag-value>]
cisco-avpair = "[vrf <vrf-name>] {<prefix> <netmask>} [vrf <next hop vrf>]
                [<next hop prefix>]
                [<metric>] [tag <tag-value>] "
```

- **IPv6**

```
Framed-IPv6-Route = "[vrf <prefix VRF>] {<prefix>/<prefix_length>}
                     [vrf <next hop vrf>]
                     [<next-hop prefix>] [<metric>] [tag <tag-value>] "
cisco-avpair = "[vrf <prefix VRF>] {<prefix>/<prefix_length>} [vrf <next hop vrf>]
                [<next hop prefix>][<metric>] [tag <tag-value>] "
```

The description of the format of the framed route is as follows:

- **[vrf <prefix VRF>]**: This is an optional parameter. Specfies the virtual routing and forwarding (VRF) prefix.
- **{<prefix>/<prefix_length>} or {<prefix> <netmask>}**: This is a mandatory parameter. Specifies the prefix and prefix mask or prefix length for the destination.
- **[vrf <next hop vrf>]**: This is an optional parameter. Specifies the next hop VRF name.
- **[<next hop prefix>]**: This is an optional parameter. Specifies that when the next hop is specified as "0.0.0.0" for IPv4 or "::" for IPv6, the IP address of the session must be used as the next hop prefix.
- **[<metric>]**: This is an optional parameter. Specifies the route metric.
- **[tag <tag-value>]**: This is an optional parameter. Specifies a tag value that can be used as a match for controlling redistribution using route policies.

For information about the framed-route attributes, see Table 1 and Table 2 in the RADIUS Attributes, on page 601 chapter.

Implementing the framed (dynamic) route support depends on the UP. Therfore, check the *UP Cloud Native BNG User Plane Configuration Guide for Cisco ASR 9000 Series Router* for the following before enabling the framed route.

- IPv4 and IPv6 framed route support for PPP Termination and Aggregation (PTA) and IPoE
- VRF and next hop VRF support for PTA and IPoE
- CoA support for framed route for PTA and IPoE
- Maximum routes supported per subscriber per AFI for PTA and IPoE

Session disconnect history

A session disconnect history is a troubleshooting feature that

- maintains a cache of disconnected subscriber sessions
- enables operators to clear disconnect history records as needed, and
- provides metrics for monitoring session disconnects across User Planes.

You can customize the number of disconnect history records stored per user plane and clear records globally or by specific identifiers.

Table 101: Feature history

Feature Name	Release Information	Description
Session disconnect history enhancements	2025.04.0	You can now clear session disconnect history and increase the storage limit to 168,000 entries for up to two User Planes. New CLI commands allow clearing records by User Plane and setting custom limits. Disconnect history metrics are also available in Grafana for improved monitoring and visibility.
Session disconnect history	2025.01.0	This feature enhances troubleshooting by providing detailed records of past session disconnections in cnBNGs. This feature is crucial for understanding why sessions have been disconnected in the past, allowing for effective problem resolution and network management.

Benefits of session disconnect history enhancements

The enhancements to session disconnect history provide several benefits:

- You can clear disconnect history records globally or for specific User Planes.
- The default limit of 1,000 records per user plane can be increased up to 168,000, but this higher limit is supported for only two User Planes due to performance considerations. Not all User Planes can be set to this maximum value.
- Metrics are exposed for monitoring disconnects, providing visibility through tools such as Grafana and the Ops-center.
- Backward compatibility is maintained, so previous configurations and data remain accessible.

Restrictions for session disconnect history

These restrictions apply to the session disconnect history feature:

- This feature cannot be enabled or disabled via the CLI.
- Display is limited to per UPF or per SRG-peer-id only.

Configure the disconnect history record limit per User Plane

Set a custom limit for the number of disconnect history records stored for each User Plane.

By default, each user plane stores up to 1,000 disconnect history records. You can increase this limit up to 168,000, but this higher limit is supported only for two User Planes. If you attempt to set the higher limit for a third user plane, Ops Center will reject the configuration.

Before you begin

- Ensure you have access privileges to configure User Plane settings.

Procedure

Enter the user-plane configuration mode, and configure the desired record limit.

Example:

```
user-plane
 instance 1
  user-plane upf1
   disconnect-history-records 161000
   peer-address ipv4 10.1.33.94
   subscriber-profile automation-subsprofile
  exit
 exit
exit
```

The user plane will retain up to the specified number of disconnect history records. If the configuration is removed or reset, the default limit of 1,000 records per user plane applies.

Clear session disconnect history records

Remove disconnect history records from the system for maintenance or troubleshooting.

Before you begin

- Confirm which records (all, or specific User Plane) you wish to clear.

Procedure

Step 1 Use the **clear subscriber session disconnect history** command to clear all disconnect history records.

Step 2 Use the **clear subscriber session disconnect-history upf** *user-plane-name* command to clear records for a specific User Plane.

Monitor disconnect history metrics

Track and visualize disconnect history statistics to gain operational insight.

Monitoring platforms such as Grafana provide metrics that help visualize disconnect events across User Planes.

Before you begin

Ensure your monitoring system is configured to collect and display the relevant metrics.

Procedure

Step 1 Access your monitoring system (for example, Grafana).

Step 2 Locate the **Session_Disconnect_History_Total** metric, and review labels such as disconnect_reason, upf, and srg_peer_id.

NOTES:

- **disconnect_reason**: reason for the subscriber call disconnect
- **upf**: User Plane name
- **srg_peer_id**: SRG peer ID

The **srg_peer_id** label is only included if you enable it through a specific configuration in Ops Center. By default, this configuration is not set.

a) Use this sample configuration to configure srg_peer_id.

Example:

```
bng#
config
 infra metrics verbose application
  level trace
  metrics Session_Disconnect_History_Total
   level production
   granular-labels [ srg_peer_id ]
  exit
 exit
```

Step 3 Use these metrics to identify trends and troubleshoot disconnect issues.

Verify session disconnect history

Use the **show disconnect-history sessmgr upf** command to view the disconnected session details.

Note Starting Release 2026.01.0, the **show subscriber session disconnect-history** command is deprecated.

UPF based CLIs

- bng# **show disconnect-history sessmgr upf** *upf1*

```
Wed Jan  21 07:27:20.430 UTC+00:00
Upf = upf1
-------------------------------------------------------------------------------------------------------
[Disconnect Reason]            [Last Disconnect Time]       [Mac-Address]    [Sublabel]
   [Srg-Peer-Id]
-------------------------------------------------------------------------------------------------------
Dhcp admin delete               2026/01/21 07:27:06.874      aa11.0000.0001   33554433
     Peer1
-------------------------------------------------------------------------------------------------------
```

This command provides a one-line summary for the specified UPF name, showing details of only the most recent disconnected call record.

- bng# **show disconnect-history sessmgr upf** *upf1* **unique**

```
Wed Jan  21 07:27:22.560 UTC+00:00
```

```
-------------------------------------------------------------------------------------------------------------
[Disconnect Reason]             [Last Disconnect Time]       [Mac-Address]    [Sublabel]
   [Srg-Peer-Id]     [Count]
-------------------------------------------------------------------------------------------------------------
Client lease released            2026/01/21 07:26:44.333      aa11.0000.0001   16777217
     Peer1              1
Dhcp admin delete                2026/01/21 07:27:06.874      aa11.0000.0001   33554433
     Peer1              1
-------------------------------------------------------------------------------------------------------------
```

This command displays disconnect records summarized by disconnect reason. The count column shows the total number of records for each reason. The Last Disconnect Time, MAC Address, Sublabel, and SRG Peer ID columns display the values from the most recent disconnect record for each reason.

- `bng# **show disconnect-history sessmgr upf** *upf1* **last**`

```
Wed Jan  21 07:27:25.574 UTC+00:00
-------------------------------------------------------------------------------------------------------------
[Disconnect Reason]            [Last Disconnect Time]       [Mac-Address]    [Sublabel]
   [Srg-Peer-Id]    [Count]
-------------------------------------------------------------------------------------------------------------
Dhcp admin delete                2026/01/21 07:27:06.874      aa11.0000.0001   33554433
     Peer1             1
Client lease released            2026/01/21 07:26:44.333      aa11.0000.0001   16777217
     Peer1             1
-------------------------------------------------------------------------------------------------------------
```

This command displays up to the last 1000 disconnected records for the specified UPF, ordered from most recent to oldest. Even if more than 1000 records exist, only the latest 1000 records are displayed.

- `bng# **show disconnect-history sessmgr upf** *upf1* **details**`

```
Wed Jan  21 07:27:29.778 UTC+00:00
{
  "subResponses": [
    {
      "subLabel": "16777217",
      "srgPeerId": "Peer1",
      "srgGroupId": "Group1",
      "srgIntfId": "1",
      "mac": "aa11.0000.0001",
      "acct-sess-id": "01000001",
      "sesstype": "ipoe",
      "state": "deleting",
      "subCreateTime": "Wed, 21 Jan 2026 07:26:34 UTC",
      "dhcpAuditId": 2,
      "transId": "1",
      "subsAttr": {
        "attrs": {
          "Authentic": "RADIUS(1)",
          "addr": "33.0.0.2",
          "client-mac-address": "aa11.0000.0001",
          "dhcp-client-id": "0x01aa1100000001",
          "dhcpv4-client-id": "0x01aa1100000001",
          "inner-vlan-id": "200",
          "outer-vlan-id": "100",
          "physical-adapter": "5",
          "physical-chassis": "1",
          "physical-port": "4",
          "physical-slot": "2",
          "physical-subslot": "3",
          "port-type": "IP over QinQ(41)",
          "protocol-type": "dhcp(3)",
          "string-session-id": "01000001",
```

```
    "username": "cnbng",
    "vrf": "automation-vrf"
  }
},
"subcfgInfo": {
  "committedAttrs": {
    "attrs": {
      "accounting-list": "automation-aaaprofile",
      "acct-interval": "2000",
      "addr-pool": "automation-poolv4",
      "ipv4-mtu": "1400",
      "ppp-ipcp-reneg-ignore": "true",
      "ppp-ipv6cp-reneg-ignore": "true",
      "ppp-lcp-reneg-ignore": "true",
      "session-acct-enabled": "true",
      "vrf": "automation-vrf"
    }
  },
  "activatedServices": [
    {
      "serviceName": "automation-featuretemplate",
      "serviceAttrs": {
        "attrs": {}
      }
    },
    {
      "serviceName": "automation-pppptemplate",
      "serviceAttrs": {
        "attrs": {}
      }
    },
    {
      "serviceName": "automation-feature-template-accounting",
      "serviceAttrs": {
        "attrs": {
          "accounting-list": "automation-aaaprofile",
          "acct-interval": "2000",
          "service-acct-enabled": "true",
          "sub-qos-policy-in": "inpolicy",
          "sub-qos-policy-out": "outpolicy"
        }
      }
    }
  ]
},
"smupstate": "smUpSessionDeleted",
"v4AfiState": "Up",
"v6AfiState": "Down",
"interimInterval": "2000",
"interimSentToUp": "1940",
"sessionAccounting": "enable",
"serviceAccounting": "enable",
"upAttr": {
  "attrs": {
    "addr": "33.0.0.2",
    "subscriber-af-type": "ipv4-v6(3)"
  }
},
"upfsInfo": {
  "upf1": {
    "portName": "GigabitEthernet0/0/0/1",
    "upId": 1,
    "transId": 1,
    "smupState": "smUpSessionCreated"
```

```
        },
        "upf1-stby": {
          "portName": "GigabitEthernet0/0/0/3",
          "upId": 1,
          "transId": 1,
          "smupState": "smUpSessionCreated",
          "lastUpdateTime": "Wed, 21 Jan 2026 07:26:35 UTC"
        }
      },
      "sess-events": [
        "Time, Event, Status",
        "2026-01-21 07:26:34.673873168 +0000 UTC, SessionCreate, success",
        "2026-01-21 07:26:35.07214338 +0000 UTC, N4-Create:upf1, PASS",
        "2026-01-21 07:26:35.086491866 +0000 UTC, SessionUpdate, success",
        "2026-01-21 07:26:35.111517481 +0000 UTC, N4CreateToStdby:upf1-stby, PASS",
        "2026-01-21 07:26:44.307487839 +0000 UTC, BNGN4SessionReleaseReq:upf1, PASS",
        "2026-01-21 07:26:44.322360594 +0000 UTC, N4DeleteToStdby:upf1-stby, PASS"
      ]
    },
  ]
}
```


This command displays up to the last 1000 disconnected records for the specified UPF.

- `bng#` **`show disconnect-history sessmgr upf`** *`upf1`* **`filter mac`** *`aa11.0000.0064`*


```
Wed Jan  21 07:28:03.473 UTC+00:00
filter mac aa11.0000.0001
{
  "subResponses": [
    {
      "subLabel": "16777217",
      "srgPeerId": "Peer1",
      "srgGroupId": "Group1",
      "srgIntfId": "1",
      "mac": "aa11.0000.0001",
      "acct-sess-id": "01000001",
      "sesstype": "ipoe",
      "state": "deleting",
      "subCreateTime": "Wed, 21 Jan 2026 07:26:34 UTC",
      "dhcpAuditId": 2,
      "transId": "1",
      "subsAttr": {
        "attrs": {
          "Authentic": "RADIUS(1)",
          "addr": "33.0.0.2",
          "client-mac-address": "aa11.0000.0001",
          "dhcp-client-id": "0x01aa1100000001",
          "dhcpv4-client-id": "0x01aa1100000001",
          "inner-vlan-id": "200",
          "outer-vlan-id": "100",
          "physical-adapter": "5",
          "physical-chassis": "1",
          "physical-port": "4",
          "physical-slot": "2",
          "physical-subslot": "3",
          "port-type": "IP over QinQ(41)",
          "protocol-type": "dhcp(3)",
          "string-session-id": "01000001",
          "username": "cnbng",
          "vrf": "automation-vrf"
        }
      },
      "subcfgInfo": {
```

```
  "committedAttrs": {
    "attrs": {
      "accounting-list": "automation-aaaprofile",
      "acct-interval": "2000",
      "addr-pool": "automation-poolv4",
      "ipv4-mtu": "1400",
      "ppp-ipcp-reneg-ignore": "true",
      "ppp-ipv6cp-reneg-ignore": "true",
      "ppp-lcp-reneg-ignore": "true",
      "session-acct-enabled": "true",
      "vrf": "automation-vrf"
    }
  },
  "activatedServices": [
    {
      "serviceName": "automation-featuretemplate",
      "serviceAttrs": {
        "attrs": {}
      }
    },
    {
      "serviceName": "automation-ppptemplate",
      "serviceAttrs": {
        "attrs": {}
      }
    },
    {
      "serviceName": "automation-feature-template-accounting",
      "serviceAttrs": {
        "attrs": {
          "accounting-list": "automation-aaaprofile",
          "acct-interval": "2000",
          "service-acct-enabled": "true",
          "sub-qos-policy-in": "inpolicy",
          "sub-qos-policy-out": "outpolicy"
        }
      }
    }
  ]
},
"smupstate": "smUpSessionDeleted",
"v4AfiState": "Up",
"v6AfiState": "Down",
"interimInterval": "2000",
"interimSentToUp": "1940",
"sessionAccounting": "enable",
"serviceAccounting": "enable",
"upAttr": {
  "attrs": {
    "addr": "33.0.0.2",
    "subscriber-af-type": "ipv4-v6(3)"
  }
},
"upfsInfo": {
  "upf1": {
    "portName": "GigabitEthernet0/0/0/1",
    "upId": 1,
    "transId": 1,
    "smupState": "smUpSessionCreated"
  },
  "upf1-stby": {
    "portName": "GigabitEthernet0/0/0/3",
    "upId": 1,
    "transId": 1,
```

```
          "smupState": "smUpSessionCreated",
          "lastUpdateTime": "Wed, 21 Jan 2026 07:26:35 UTC"
        }
      },
      "sess-events": [
        "Time, Event, Status",
        "2026-01-21 07:26:34.673873168 +0000 UTC, SessionCreate, success",
        "2026-01-21 07:26:35.07214338 +0000 UTC, N4-Create:upf1, PASS",
        "2026-01-21 07:26:35.086491866 +0000 UTC, SessionUpdate, success",
        "2026-01-21 07:26:35.111517481 +0000 UTC, N4CreateToStdby:upf1-stby, PASS",
        "2026-01-21 07:26:44.307487839 +0000 UTC, BNGN4SessionReleaseReq:upf1, PASS",
        "2026-01-21 07:26:44.322360594 +0000 UTC, N4DeleteToStdby:upf1-stby, PASS"
      ]
    },

  ]
}
```


This command displays detailed disconnect record information for the specified MAC address.

- `bng#` **`show disconnect-history sessmgr upf`** *`upf1`* **`filter sublabel`** *`33554433`*


```
Wed Jan  21 07:28:14.701 UTC+00:00
filter sublabel 33554433
{
  "subResponses": [
    {
      "subLabel": "33554433",
      "srgPeerId": "Peer1",
      "srgGroupId": "Group1",
      "srgIntfId": "1",
      "mac": "aa11.0000.0001",
      "acct-sess-id": "02000001",
      "sesstype": "ipoe",
      "state": "deleting",
      "subCreateTime": "Wed, 21 Jan 2026 07:26:50 UTC",
      "dhcpAuditId": 2,
      "transId": "1",
      "subsAttr": {
        "attrs": {
          "Authentic": "RADIUS(1)",
          "addr": "33.0.0.3",
          "client-mac-address": "aa11.0000.0001",
          "dhcp-client-id": "0x01aa1100000001",
          "dhcpv4-client-id": "0x01aa1100000001",
          "inner-vlan-id": "200",
          "outer-vlan-id": "100",
          "physical-adapter": "5",
          "physical-chassis": "1",
          "physical-port": "4",
          "physical-slot": "2",
          "physical-subslot": "3",
          "port-type": "IP over QinQ(41)",
          "protocol-type": "dhcp(3)",
          "string-session-id": "02000001",
          "username": "cnbng",
          "vrf": "automation-vrf"
        }
      },
      "subcfgInfo": {
        "committedAttrs": {
          "attrs": {
            "accounting-list": "automation-aaaprofile",
            "acct-interval": "2000",
```

```
        "addr-pool": "automation-poolv4",
        "ipv4-mtu": "1400",
        "ppp-ipcp-reneg-ignore": "true",
        "ppp-ipv6cp-reneg-ignore": "true",
        "ppp-lcp-reneg-ignore": "true",
        "session-acct-enabled": "true",
        "vrf": "automation-vrf"
      }
    },
    "activatedServices": [
      {
        "serviceName": "automation-featuretemplate",
        "serviceAttrs": {
          "attrs": {}
        }
      },
      {
        "serviceName": "automation-ppptemplate",
        "serviceAttrs": {
          "attrs": {}
        }
      },
      {
        "serviceName": "automation-feature-template-accounting",
        "serviceAttrs": {
          "attrs": {
            "accounting-list": "automation-aaaprofile",
            "acct-interval": "2000",
            "service-acct-enabled": "true",
            "sub-qos-policy-in": "inpolicy",
            "sub-qos-policy-out": "outpolicy"
          }
        }
      }
    ]
  },
  "smupstate": "smUpSessionDeleted",
  "v4AfiState": "Up",
  "v6AfiState": "Down",
  "interimInterval": "2000",
  "interimSentToUp": "1940",
  "sessionAccounting": "enable",
  "serviceAccounting": "enable",
  "upAttr": {
    "attrs": {
      "addr": "33.0.0.3",
      "subscriber-af-type": "ipv4-v6(3)"
    }
  },
  "upfsInfo": {
    "upf1": {
      "portName": "GigabitEthernet0/0/0/1",
      "upId": 2,
      "transId": 1,
      "smupState": "smUpSessionCreated"
    },
    "upf1-stby": {
      "portName": "GigabitEthernet0/0/0/3",
      "upId": 2,
      "transId": 1,
      "smupState": "smUpSessionCreated",
      "lastUpdateTime": "Wed, 21 Jan 2026 07:26:50 UTC"
    }
  },
```

```
      "sess-events": [
        "Time, Event, Status",
        "2026-01-21 07:26:50.860329928 +0000 UTC, SessionCreate, success",
        "2026-01-21 07:26:50.937866225 +0000 UTC, N4-Create:upf1, PASS",
        "2026-01-21 07:26:50.940983812 +0000 UTC, SessionUpdate, success",
        "2026-01-21 07:26:50.973314239 +0000 UTC, N4CreateToStdby:upf1-stby, PASS",
        "2026-01-21 07:27:06.857960507 +0000 UTC, BNGN4SessionReleaseReq:upf1, PASS",
        "2026-01-21 07:27:06.871334959 +0000 UTC, N4DeleteToStdby:upf1-stby, PASS"
      ]
    }
  ]
}
```

This command displays disconnect record information for the given sublabel.

SRG peer-id based CLIs

- `bng#` **`show disconnect-history sessmgr srg-peer-id`** *`Peer1`*

```
Wed Jan  21 07:30:09.200 UTC+00:00
Srg-Peer-Id = Peer1
-----------------------------------------------------------------------------------------------
[Disconnect Reason]        [Last Disconnect Time]       [Mac-Address]    [Sublabel]
[Upf]
-----------------------------------------------------------------------------------------------
Dhcp admin delete           2026/01/21 07:27:06.874      aa11.0000.0001   33554433
 upf1
-----------------------------------------------------------------------------------------------
```

This command provides a one-line summary for the specified SRG Peer ID, showing details of only the most recent disconnected call record.

- `bng#` **`show disconnect-history sessmgr srg-peer-id`** *`Peer1`* **`last`**

```
Wed Jan  21 07:30:14.387 UTC+00:00
-----------------------------------------------------------------------------------------------
[Disconnect Reason]          [Last Disconnect Time]       [Mac-Address]    [Sublabel]
  [Upf]    [Count]
-----------------------------------------------------------------------------------------------
Dhcp admin delete             2026/01/21 07:27:06.874      aa11.0000.0001   33554433
   upf1      1
Client lease released         2026/01/21 07:26:44.333      aa11.0000.0001   16777217
   upf1      1
-----------------------------------------------------------------------------------------------
```

This command displays up to the last 1000 disconnected records for the specified SRG Peer ID, ordered from most recent to oldest. Even if more than 1000 records exist, only the latest 1000 records are displayed.

- `bng#` **`show subscriber session disconnect-history srg-peer-id`** *`Peer1`* **`unique`**

```
Wed Jan  21 07:30:10.832 UTC+00:00
-----------------------------------------------------------------------------------------------
[Disconnect Reason]        [Last Disconnect Time]       [Mac-Address]    [Sublabel]
[Upf]   [Count]
-----------------------------------------------------------------------------------------------
Client lease released       2026/01/21 07:26:44.333      aa11.0000.0001   16777217
 upf1    1
Dhcp admin delete           2026/01/21 07:27:06.874      aa11.0000.0001   33554433
 upf1    1
-----------------------------------------------------------------------------------------------
```

This command displays disconnect records summarized by disconnect reason. The count column shows the total number of records for each reason. The Last Disconnect Time, MAC Address, Sublabel, and SRG Peer ID columns show the values from the most recent disconnect record for each reason.

- bng# **show subscriber session disconnect-history srg-peer-id** *Peer1* **details**

```
Wed Jan 21 07:30:18.893 UTC+00:00
{
 "subResponses": [
 {
 "subLabel": "16777217",
 "srgPeerId": "Peer1",
 "srgGroupId": "Group1",
 "srgIntfId": "1",
 "mac": "aa11.0000.0001",
 "acct-sess-id": "01000001",
 "sesstype": "ipoe",
 "state": "deleting",
 "subCreateTime": "Wed, 21 Jan 2026 07:26:34 UTC",
 "dhcpAuditId": 2,
 "transId": "1",
 "subsAttr": {
 "attrs": {
 "Authentic": "RADIUS(1)",
 "addr": "33.0.0.2",
 "client-mac-address": "aa11.0000.0001",
 "dhcp-client-id": "0x01aa1100000001",
 "dhcpv4-client-id": "0x01aa1100000001",
 "inner-vlan-id": "200",
 "outer-vlan-id": "100",
 "physical-adapter": "5",
 "physical-chassis": "1",
 "physical-port": "4",
 "physical-slot": "2",
 "physical-subslot": "3",
 "port-type": "IP over QinQ(41)",
 "protocol-type": "dhcp(3)",
 "string-session-id": "01000001",
 "username": "cnbng",
 "vrf": "automation-vrf"
 }
 },
 "subcfgInfo": {
 "committedAttrs": {
 "attrs": {
 "accounting-list": "automation-aaaprofile",
 "acct-interval": "2000",
 "addr-pool": "automation-poolv4",
 "ipv4-mtu": "1400",
 "ppp-ipcp-reneg-ignore": "true",
 "ppp-ipv6cp-reneg-ignore": "true",
 "ppp-lcp-reneg-ignore": "true",
 "session-acct-enabled": "true",
 "vrf": "automation-vrf"
 }
 },
 "activatedServices": [
 {
 "serviceName": "automation-featuretemplate",
 "serviceAttrs": {
 "attrs": {}
 }
 },
 {
```

```
"serviceName": "automation-ppptemplate",
"serviceAttrs": {
"attrs": {}
}
},
{
"serviceName": "automation-feature-template-accounting",
"serviceAttrs": {
"attrs": {
"accounting-list": "automation-aaaprofile",
"acct-interval": "2000",
"service-acct-enabled": "true",
"sub-qos-policy-in": "inpolicy",
"sub-qos-policy-out": "outpolicy"
}
}
}
]
},
"smupstate": "smUpSessionDeleted",
"v4AfiState": "Up",
"v6AfiState": "Down",
"interimInterval": "2000",
"interimSentToUp": "1940",
"sessionAccounting": "enable",
"serviceAccounting": "enable",
"upAttr": {
"attrs": {
"addr": "33.0.0.2",
"subscriber-af-type": "ipv4-v6(3)"
}
},
"upfsInfo": {
"upf1": {
"portName": "GigabitEthernet0/0/0/1",
"upId": 1,
"transId": 1,
"smupState": "smUpSessionCreated"
},
"upf1-stby": {
"portName": "GigabitEthernet0/0/0/3",
"upId": 1,
"transId": 1,
"smupState": "smUpSessionCreated",
"lastUpdateTime": "Wed, 21 Jan 2026 07:26:35 UTC"
}
},
"sess-events": [
"Time, Event, Status",
       "2026-01-21 07:26:34.673873168 +0000 UTC, SessionCreate,
     success",
       "2026-01-21 07:26:35.07214338 +0000 UTC, N4-Create:upf1,
       PASS",
       "2026-01-21 07:26:35.086491866 +0000 UTC, SessionUpdate,
     success",
       "2026-01-21 07:26:35.111517481 +0000 UTC,
       N4CreateToStdby:upf1-stby, PASS",
       "2026-01-21 07:26:44.307487839 +0000 UTC,
       BNGN4SessionReleaseReq:upf1, PASS",
       "2026-01-21 07:26:44.322360594 +0000 UTC,
       N4DeleteToStdby:upf1-stby, PASS"
]
},
{
```

```
"subLabel": "33554433",
"srgPeerId": "Peer1",
"srgGroupId": "Group1",
"srgIntfId": "1",
"mac": "aa11.0000.0001",
"acct-sess-id": "02000001",
"sesstype": "ipoe",
"state": "deleting",
"subCreateTime": "Wed, 21 Jan 2026 07:26:50 UTC",
"dhcpAuditId": 2,
"transId": "1",
"subsAttr": {
"attrs": {
"Authentic": "RADIUS(1)",
"addr": "33.0.0.3",
"client-mac-address": "aa11.0000.0001",
"dhcp-client-id": "0x01aa1100000001",
"dhcpv4-client-id": "0x01aa1100000001",
"inner-vlan-id": "200",
"outer-vlan-id": "100",
"physical-adapter": "5",
"physical-chassis": "1",
"physical-port": "4",
"physical-slot": "2",
"physical-subslot": "3",
"port-type": "IP over QinQ(41)",
"protocol-type": "dhcp(3)",
"string-session-id": "02000001",
"username": "cnbng",
"vrf": "automation-vrf"
}
},
"subcfgInfo": {
"committedAttrs": {
"attrs": {
"accounting-list": "automation-aaaprofile",
"acct-interval": "2000",
"addr-pool": "automation-poolv4",
"ipv4-mtu": "1400",
"ppp-ipcp-reneg-ignore": "true",
"ppp-ipv6cp-reneg-ignore": "true",
"ppp-lcp-reneg-ignore": "true",
"session-acct-enabled": "true",
"vrf": "automation-vrf"
}
},
"activatedServices": [
{
"serviceName": "automation-featuretemplate",
"serviceAttrs": {
"attrs": {}
}
},
{
"serviceName": "automation-pppptemplate",
"serviceAttrs": {
"attrs": {}
}
},
{
"serviceName": "automation-feature-template-accounting",
"serviceAttrs": {
"attrs": {
"accounting-list": "automation-aaaprofile",
```

```
"acct-interval": "2000",
"service-acct-enabled": "true",
"sub-qos-policy-in": "inpolicy",
"sub-qos-policy-out": "outpolicy"
}
}
}
]
},
"smupstate": "smUpSessionDeleted",
"v4AfiState": "Up",
"v6AfiState": "Down",
"interimInterval": "2000",
"interimSentToUp": "1940",
"sessionAccounting": "enable",
"serviceAccounting": "enable",
"upAttr": {
"attrs": {
"addr": "33.0.0.3",
"subscriber-af-type": "ipv4-v6(3)"
}
},
"upfsInfo": {
"upf1": {
"portName": "GigabitEthernet0/0/0/1",
"upId": 2,
"transId": 1,
"smupState": "smUpSessionCreated"
},
"upf1-stby": {
"portName": "GigabitEthernet0/0/0/3",
"upId": 2,
"transId": 1,
"smupState": "smUpSessionCreated",
"lastUpdateTime": "Wed, 21 Jan 2026 07:26:50 UTC"
}
},
"sess-events": [
"Time, Event, Status",
       "2026-01-21 07:26:50.860329928 +0000 UTC, SessionCreate,
     success",
       "2026-01-21 07:26:50.937866225 +0000 UTC, N4-Create:upf1,
       PASS",
       "2026-01-21 07:26:50.940983812 +0000 UTC, SessionUpdate,
     success",
       "2026-01-21 07:26:50.973314239 +0000 UTC,
       N4CreateToStdby:upf1-stby, PASS",
       "2026-01-21 07:27:06.857960507 +0000 UTC,
       BNGN4SessionReleaseReq:upf1, PASS",
       "2026-01-21 07:27:06.871334959 +0000 UTC,
       N4DeleteToStdby:upf1-stby, PASS"
]
}
]
}
```

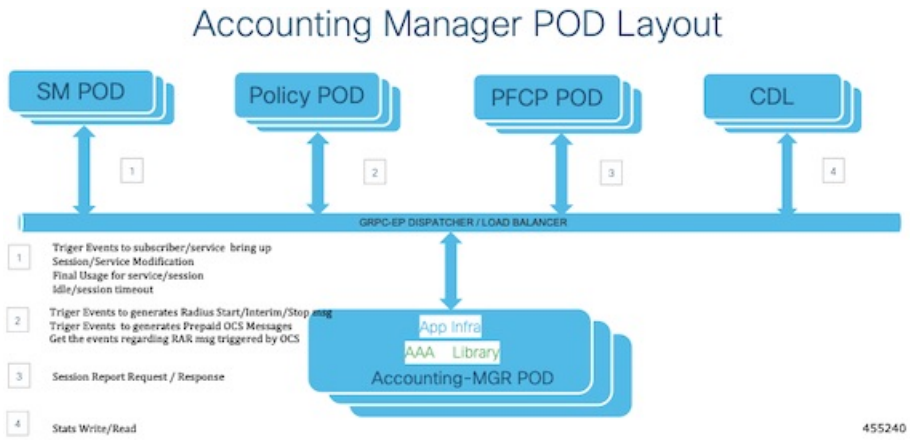

This command displays up to the last 1000 disconnected records for the specified SRG Peer ID.

Subscriber Accounting Functions

Feature Description

The Accounting Manager handles the Subscriber Accounting functions in the cnBNG CP. The Accounting function includes features that track traffic either in volume or duration. It provides accounting information for subscribers on a session or per service. The Accounting function determines the length and duration of a given service that a subscriber has used. Certain regulations require service providers to account for services they provide to the subscriber.

The following figure illustrates the Accounting Manager external interfaces.



The Accounting Manager in cnBNG supports the following forms of accounting:

Service Accounting

ISPs can offer different tiered services to their subscribers with the ability to move between different tiers. Different tiers could correspond to different bandwidths offered to the subscriber. A subscriber can enable a new service that corresponds to temporarily moving from one tier of service to another. ISPs need to keep track of when a new service is enabled and how long it is active for each subscriber. Often there might be a need to count the number of packets and bytes associated with a service. Both of these forms of accountng are referred to as service accounting. When service accounting is enabled, BNG sends a Service-Start request when service is activated and a Service-Stop request when the service is deactivated. A timestamp is sent with both the actions. Service-Stop can also contain statistics associated with the service.

To configure Service Accounting, see Configuring Service Accounting, on page 462.

Session Accounting

When Session Accounting is activated, an Accounting-Start request is sent to AAA when the session is started. When the session is terminated, an Accounting-Stop request is sent. The Accounting-Stop request contains the final session accounting statistics (packets, bytes in, bytes out). An “interim” session accounting can be optionally activated that sends Interim-Updates periodically while the session is active. These updates provide the current session statistics accumulated since the start of the session.

Session Accounting is configured directly on the template.

To configure Session Accounting, see Configuring Session Accounting, on page 463.

Limitations and Restrictions

The Subscriber Accounting Function has the following limitation in this release:

- An interim Interval of zero is not supported.
- AAA profile change at service level is not supported.
- Service-level attributes changes are not supported after service bring-up.
- Session accounting is mandatory to enable Service accounting due to User Plane (UP) (asr9k) limitation.
- Session and Service Accounting enable or disable is not supported after session or service is up because of UP limitations. Session Accounting must be enabled only during session bring-up.

Configuring Subscriber Accounting Functions

This section describes how to configure the Subscriber Accounting Functions.

The configuration of the Subscriber Accounting Functions involve the following procedures:

- Configuring Service Accounting
- Configuring Session Accounting

Configuring Service Accounting

Use the following commands to configure service accounting.

```
config
  profile feature-template feature-template
  service accounting
    aaa-profile aaa_profile_name
    enable
    periodic-interval interval_in_seconds
    exit
```

NOTES:

- **profile feature-template** *feature-template*: Specifies the profile feature template name and enters Feature-Template Configuration mode.
- **service accounting**: Enters the Service Configuration mode to configure service accounting for a AAA profile.
- **aaa-profile** *aaa_profile_name*: Specifies the AAA profile to use for service accounting.
- **enable**: Enables service accounting for the specified AAA profile.
- **periodic-interval** *interval_in_seconds*: Specifies the interim interval in seconds. The valid values range from 60 to 4320000 seconds.

Configuring Session Accounting

Use the following commands to configure session accounting.

```
config
  profile feature-template feature-template
  session accounting
    aaa-profile aaa_profile_name
    dual-stack-delay delay_in_seconds
    enable
    periodic-interval interval_in_seconds
    exit
```

NOTES:

- **profile feature-template** *feature-template*: Specifies the profile feature template name and enters Feature-Template Configuration mode.
- **session accounting**: Enters the Session Configuration mode to configure session accounting for a AAA profile.
- **aaa-profile** *aaa_profile_name*: Specifies the AAA profile to use for session accounting.
- **dual-stack-delay** *delay_in_seconds*: Specifies the dual stack set delay time in seconds. The valid values range from 1 to 30 seconds.
- **enable**: Enables session accounting for the specified AAA profile.
- **periodic-interval** *interval_in_seconds*: Specifies the interim interval in seconds. The valid values range from 60 to 4320000 seconds.

Subscriber QoS Policy

Feature Description

The Subscriber Quality of Service (QoS) Policy feature uses the following Cisco AVPs to apply the subscriber QOS policy through RADIUS.

```
cisco-avpair = "subscriber:sub-qos-policy-in=<ingress qos policy name>"
cisco-avpair = "subscriber:sub-qos-policy-out=<egress qos policy name>",
```

Example:

```
radius profile
cisco-avpair = "subscriber:sub-qos-policy-in=qos_in_100mbps",
cisco-avpair = "subscriber:sub-qos-policy-out=qos_out_100mbps",
```

"qos_in_100mbps" and "qos_out_100mbps" are the QoS policy maps that are configured in the User Plane (UP). The merge-level and accounting features are not supported through RADIUS. If unsupported features are passed from RADIUS, behaviour is undefined.

Applying QoS from profile feature-template and through RADIUS using sub-qos-policy-in or sub-qos-policy-out is not supporetd for the same subscriber. When applied, behaviour is undefined.

For information about the sub-qos-policy-in or sub-qos-policy-out attributes, see Table 2 in the RADIUS Attributes, on page 601 chapter.

RADIUS-Based Policing - QoS Shape-Rate parameterization

RADIUS-Based Policing (RaBaPol) is a network management approach that

- enables the use of customized parameters instead of default parameters to activate cnBNG subscriber services.
- allows for greater flexibility and control over service configurations.

Table 102: Feature History

Feature Name	Release Information	Description
RADIUS-Based Policing - QoS Shape-Rate parameterization	2025.02.0	You can now dynamically manage your cnBNG subscriber services through RADIUS-based activation. With RADIUS-Based Policing (RaBaPol), you can customize service parameters, such as the QoS shape-rate, according to your requirements, giving you greater control over service management.

Parameterization of QoS shape-rate

RaBaPol supports the customization of the QoS shape-rate parameter. This parameter can be sent to the cnBNG Control Plane (CP) by the RADIUS server either during the initial connection setup as Cisco VSAs in an Access Accept message, or through Change of Authorization (CoA) messages.

Configuring QoS shape-rate parameterization

To establish QoS shape-rate parameterization, use the **shape average $var_name = value** command in the policy-map class configuration mode in the cnBNG User Plane (UP). This customization is feature-dependent and requires specific syntax and semantics. For QoS, a dollar sign ($) is added as a prefix to the **shape-rate** variable, and the default value, along with the variables, is configured in the policy-map definition.

Handling service changes and errors

If a service associated with a subscriber needs a change in the variable list, deactivate the current service using CoA Session-Disconnect and activate the updated service using CoA Session-Activate process. If an error occurs during feature activation, the cnBNG UP reverts all features and associated variable lists to their previous states.

Policy merging support

You can merge QoS policies from multiple dynamic templates. Configure these templates through CLI or download them from an AAA server for comprehensive policy integration.

Benefits of RADIUS-Based Policing

The RADIUS-Based Policing feature provides these benefits.

- **Dynamic activation**: Enables dynamic and flexible service activation based on RADIUS messages.

- **QoS customization**: Allows for the customization of QoS parameters to meet specific subscriber needs.
- **Policy merging**: Supports the merging of QoS policies from multiple dynamic templates for a subscriber.
- **Error rollback**: Provides rollback capabilities to previous states in case of errors during service activation.

Use case for QoS shape-rate parameterization

This use case illustrates how to manage and customize network QoS settings when a subscriber starts a session.

1. **Subscriber session initiation**: A user starts a session with specific credentials and settings, such as a username, password, and protocol type. For example,

```
user-cpe@abc.com         Password="abc"
            Framed-Protocol=PPP,
            Service-Type=Framed-User
              .....
               Cisco-avpair = "subscriber:sa=DEFAULT-QOS(shape-rate=120000)
```

2. **AAA server communication**: The Authentication, Authorization, and Accounting (AAA) server sends an Access-Accept message to the cnBNG. This message specifies the service name, action type, and a list of variables with their values, like the QoS shape-rate.
3. **Policy configuration**: The service name from the AAA message maps to a feature-template on the cnBNG's control plane, and the specified QoS shape-rate is used to override the default settings on the cnBNG's user plane. The policy merges these custom values with default values, retaining defaults where no specific values are provided.
4. **Service activation via CoA**: Alternatively, service activation can be achieved using CoA, which involves removing the old policy and configuring a new, merged policy in the hardware.

Limitations of configuring RADIUS-Based Policy

This limitation applies to the RADIUS-Based Policy feature:

- Service modifications with different RaBaPol configurations are not supported.

Configure QoS shape-rate parameterization

Follow these steps to configure QoS shape-rate parameterization.

Procedure

Step 1 Define a feature template with the desired QoS configuration on the cnBNG CP.

Example:

```
config
  profile feature-template feature_template_name
    qos
        in-policy qos_input_policy_name
        out-policy qos_output_policy_name
        merge-level integer
```

```
    exit
  exit
```

NOTES:

- **profile feature-template** *feature_template_name*: Specifies the profile feature template name.
- **qos**: Enters the QoS configuration mode to configure the parameters.
- **in-policy** *qos_input_policy_name*: Specifies the QoS input policy name. The supported length of the *qos_input_policy_name* ranges from 1 to 128 characters.
- **out-policy** *qos_output_policy_name*: Specifies the QoS output policy name. The supported length of the *qos_output_policy_name* ranges from 1 to 128 characters.
- **merge-level** *integer*: Enables or disables the merge level. A merge value of 0 disables the merge-level. Any value greater than 0, enables the merge level.

This is a sample configuration.

```
config
  profile feature-template DEFAULT-QOS
    qos
     in-policy hqos-policy1
     out-policy hqos-policy2
     merge-level 10
   exit
  exit
```

Step 2 Configure the policy map with a shape-rate value, on the cnBNG UP.

Example:

```
config
  policy-map policy_map_name
     class class-default
        shape average $shape-rate =  rate (units)
     exit
     end-policy-map
  exit
```

NOTES:

- **policy-map** *policy_map_name*: Creates or modifies a policy map that can be attached to one or more interfaces to specify a service policy.
- **class class-default**: Configures a traffic policy for the default class of the traffic policy.
- **shape average $shape-rate =** *rate (units)*: Average shaping rate in the specified units. Values can be from 1 to 4294967295. *Units* can be one of the following:
  - **bps**
  - **gbps**
  - **kbps**
  - **mbps**

This is a sample configuration.

```
config
  policy-map hqos-policy2
      class class-default
          shape average $shape-rate = 100000 kbps
      exit
      end-policy-map
  exit
```

In this example, the service named DEFAULT-QOS has QoS features enabled. The associated feature template is configured with outgoing QoS policies. The default value of shape-rate (the rate at which traffic is shaped) is set to 100000 kbps.

Step 3 Add the user profile to the USER file in RADIUS.

Example:

```
user-cpe@abc.com         Password="abc"
               Framed-Protocol=PPP,
               Service-Type=Framed-User
                        .....
               Cisco-avpair = "subscriber:sa=DEFAULT-QOS(shape-rate=120000)”
```

This specified QoS shape-rate value (for example, 120000) overrides the default value configured on the cnBNG UP.

Step 4 Use the **show subscriber session detail** command to verify the configuration, on the cnBNG CP.

Example:

```
show subscriber session detail
```

```
subscriber-details
{
  "subResponses": [
    {
      "subLabel": "16777218",
      "mac": "cc11.0000.0001",
      "acct-sess-id": "01000002",
      "upf": "asr9k-1",
      "port-id": "Bundle-Ether1",
      "up-subs-id": "1",
      "sesstype": "ppp",
      "state": "established",
      "subCreateTime": "Fri, 15 Nov 2024 03:34:47 UTC",
      "pppAuditId": 3,
      "transId": "2",
      "subcfgInfo": {
      "activatedServices": [
          {
            "serviceName": "DEFAULT-QOS",
            "serviceAttrs": {
              "attrs": {
                "accounting-list": "automation-aaaprofile",
                "acct-interval": "900",
                "service-acct-enabled": "true",
                "service-parameters": "shape-rate=120000",
                "sub-qos-policy-in": "hqos-policy1",
                "sub-qos-policy-out": "hqos-policy2"
              }
} } ] } } ] }
```

Step 5 Use the **show policy-map applied interface** command to view sessions configured with RaBaPol, on the cnBNG UP.

Example:

```
bng# show policy-map applied interface Bundle-Ether1.1.pppoe100

Input policy-map applied to Bundle-Ether1.1.pppoe100:

  policy-map hqos-policy1
   class class-default
    police rate 200 kbps
    !
   !

Output policy-map applied to Bundle-Ether1.1.pppoe100:

  policy-map hqos-policy2
   class class-default
    shape average $shape-rate = 100000 kbps
   !
```

Shared Policy Instance

Shared Policy Instance (SPI) is a mechanism that enables

- allocation of a single set of QoS resources among groups of cnBNG sub-interfaces and bundle sub-interfaces
- sharing of these resources across multiple Ethernet flow points, bundle interfaces, or groups of sub-interfaces.

Table 103: Feature History

Feature Name	Release Information	Description
Shared Policy Instance	2025.02.0	You can now allocate and share a single set of QoS resources across multiple cnBNG sub-interfaces and bundle sub-interfaces. By using a single QoS policy instance across multiple sub-interfaces, you can achieve aggregate shaping across your sub-interfaces, promoting streamlined bandwidth management.

Efficient QoS policy sharing across sub-interfaces: SPI allows you to share a single QoS policy instance among multiple sub-interfaces to maintain a unified rate through aggregate shaping. Sub-interfaces sharing the QoS policy must belong to the same physical interface, with the number ranging from 2 to the maximum supported by the port.

Configuration and application of policies: To implement SPI, you must configure a complete hierarchical policy-map that includes both parent and child policies. The SPI name can be defined and linked to a feature template or downloaded from a RADIUS server.

There are two main ways to configure these policies:

- **CLI and Feature Template**: Policy is configured through a Command Line Interface (CLI) and applied through a feature-template.
- **CLI and AAA Server**: Policy is configured through CLI and applied through an AAA server.

Limitations of configuring Shared Policy Instance

Session consistency within S-VLAN interface

Sessions sharing the same SPI must remain within the same S-VLAN interface.

Service accounting

Service accounting is not supported for services configured with an SPI.

SPI name change requirements

- If you modify the policy-map associated with an SPI, you must also change the SPI name.
- Avoid the following scenarios:
  - Applying a new policy with the same policy-map name but a different SPI name to a subscriber who already has an SPI policy applied. The system will reject this configuration.
  - Applying a new policy with a different policy-map name but the same SPI name. The system will reject this configuration as well.

CoA service-update request limitation

When a service policy with a user profile configuration that includes an SPI is enabled, you cannot simultaneously use an SPI in a CoA service-update request.

Configure a policy with SPI using feature template

Perform this task to configure a policy with shared policy instance in the input and output direction using feature template.

Procedure

Step 1 Define a feature template on the Control Plane (CP) that includes the SPI configuration.

Example:

```
config
  profile feature-template feature_template_name
    qos
        in-policy qos_input_policy_name
          in-shared-policy-instance spi_name
        out-policy qos_output_policy_name
          out-shared-policy-instance spi_name
     exit
  exit
```

NOTES:

- **profile feature-template** *feature_template_name*: Specifies the profile feature template name.

- **qos**: Enters the QoS configuration mode to configure the parameters.
- **in-policy** *qos_input_policy_name*: Specifies the QoS input policy associated with SPI. The supported length of the *qos_input_policy_name* ranges from 1 to 128 characters.
- **in-shared-policy-instance** *input_spi_name*: Specifies the input SPI name for the QoS policy. This command applies a shared traffic policy to inbound traffic across multiple interfaces.
- **out-policy** *qos_output_policy_name*: Specifies the QoS output policy associated with SPI. The supported length of the *qos_output_policy_name* ranges from 1 to 128 characters.
- **out-shared-policy-instance** *output_spi_name*: Specifies the output SPI name for the QoS Policy. This command applies a shared traffic policy to outbound traffic across multiple interfaces.

This is a sample configuration.

```
config
   profile feature-template DEFAULT-QOS
      qos
         in-policy hqos-policy1
           in-shared-policy-instance spi1
         out-policy hqos-policy2
           out-shared-policy-instance spi2
     exit
  exit
```

Step 2 Configure traffic policing on the cnBNG UP to monitor the traffic rate and apply actions (such as dropping or remarking packets) when the traffic exceeds the allowed limit.

Example:

```
config
  policy-map policy_map_name
     class class-default
        police rate value
     exit
     end-policy-map
  exit
```

NOTES:

- **policy-map** *policy_map_name*: Creates or modifies a policy map that can be attached to one or more interfaces to specify a service policy.
- **class class-default**: Configures a traffic policy for the default class of the traffic policy.
- **police rate** *value*: Configures traffic policing. The *value* indicates Committed information rate (CIR). Range is from 1 to 4294967295.

Step 3 Configure traffic shaping for a specific interface on the cnBNG UP.

Example:

```
config
  policy-map policy_map_name
     class class-default
        shape average value
     exit
```

```
    end-policy-map
  exit
```

NOTES:

- **shape average** *value*: Specifies the average shaping rate in the specified units. This command limits the average rate of outgoing traffic to a predefined value. Values can be from 1 to 4294967295.

Step 4 Use the **show subscriber session detail** command to verify the configuration.

Example:

```
bng# show subscriber session detail
subscriber-details
{
  "subResponses": [
    {
      "subLabel": "16777220",
      "mac": "0011.9400.0001",
      "acct-sess-id": "01000004",
      "upf": "asr9k-1",
      "port-id": "Bundle-Ether1.1",
      "up-subs-id": "3",
      "sesstype": "ppp",
      "state": "established",
      "subCreateTime": "Fri, 15 Nov 2024 04:18:51 UTC",
      "pppAuditId": 3,
      "transId": "2",
      "subcfgInfo": {
        "committedAttrs": {
        "activatedServices": [
          {
            "serviceName": "DEFAULT-QOS",
            "serviceAttrs": {
              "attrs": {
                "accounting-list": "aaaprofile",
                "acct-interval": "900",
                "service-acct-enabled": "true",
                "sub-qos-policy-in": "hqos-policy1",
                "sub-qos-policy-out": "hqos-policy2",
                "sub-qos-spi-in": "spi1",
                "sub-qos-spi-out": "spi2"
              }
            } } ] } ] }
```

Configure a Policy with SPI using RADIUS

Follow these steps to configure a policy with shared policy instance using RADIUS.

Procedure

Step 1 Configure a policy map that can be shared to one or more interfaces to specify a service policy, on the cnBNG UP.

Example:

```
config
  policy-map policy_map_name1
      class class-default
         police rate value
      exit
      end-policy-map
  exit

policy-map policy_map_name2
      class class-default
         shape average value
      exit
      end-policy-map
  exit
```

This is a sample configuration.

```
config
  policy-map hqos-policy1
      class class-default
          police rate 1024 kbps
      !
      end-policy-map
  !
  policy-map hqos-policy2
      class class-default
          shape average 4096 kbps
      !
      end-policy-map
  !
```

NOTES:

- **police rate** *value*: Specifies the policing rate for the policy-map. The value represents the committed information rate and ranges from 1 to 4294967295.
- **shape average** *value*: Specifies the average shaping rate in specified units. Values can be from 1 to 4294967295.

Step 2 Add the QoS policy with the SPI name to the USER file in RADIUS.

Example:

```
abc@example.com Cleartext-Password:= "xyz"
cisco-avpair += "sub-qos-policy-in=hqos-policy1 shared-policy-instance spi1",
cisco-avpair += "sub-qos-policy-out=hqos-policy2 shared-policy-instance spi2",
```

Step 3 Use the **show subscriber session detail** command to verify the configuration of a subscriber with a user-profile that includes both QoS and SPI settings, on the cnBNG CP.

Example:

```
bng# show subscriber session detail
subscriber-details
      {
      "subResponses": [
      {
      "subLabel": "16777221",
      "mac": "cc11.0000.0001",
      "acct-sess-id": "01000005",
      "upf": "asr9k-1",
      "port-id": "Bundle-Ether1",
```

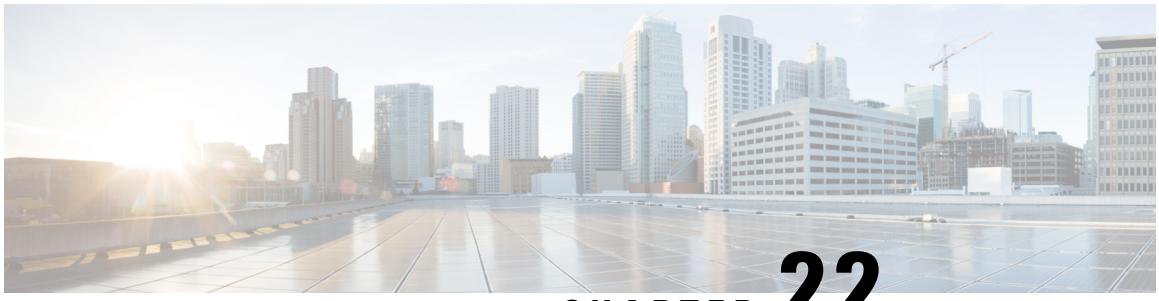
```
"up-subs-id": "4",
"sesstype": "ppp",
"state": "established",
"subCreateTime": "Fri, 15 Nov 2024 04:35:15 UTC",
"pppAuditId": 3,
"transId": "2",
"subcfgInfo": {
  "committedAttrs": {
    "attrs": {
      "accounting-list": "aaaprofile",
      "acct-interval": "900",
      "addr-pool": "pool-ISP",
      "ppp-authentication": "pap,chap",
      "ppp-ipcp-reneg-ignore": "true",
      "ppp-ipv6cp-reneg-ignore": "true",
      "ppp-lcp-delay-seconds": "1",
      "ppp-lcp-reneg-ignore": "true",
      "service-type": "Framed(2)",
      "session-acct-enabled": "true",
      "sub-qos-policy-in": "hqos-policy1 shared-policy-instance spi1",
      "sub-qos-policy-out": "hqos-policy2 shared-policy-instance spi2",
      "vrf": "default"
    }
  } } } ] }
```

Step 4 Use the **show cnbng-nal subscriber all detail** command to display sessions with user-profile having QoS and SPI, on the cnBNG UP.

Example:

```
show cnbng-nal subscriber all detail
Interface:                  Bundle-Ether1.1.pppoe4
UPID:                       0x00000004
CPID:                       0x01000005
Type:                       PPPoE
PPPOE Session Id:           00000006

Attribute List: 0x175d470
1:  ipv4-unnumbered len=  9  value= Loopback0
2:  sub-qos-policy-in len= 59  value= hqos-policy1 shared-policy-instance spi1
3:  sub-qos-policy-out len= 63  value= hqos-policy2 shared-policy-instance spi2
```



CHAPTER 22

Routed packet trigger subscriber sessions

This chapter introduces the routed packet trigger subscriber session feature, designed for Layer 3 access environments where traditional Layer 2-based discovery mechanisms—such as MAC addresses or PPPoE—are unavailable.

By leveraging the first unclassified routed IPv4 packet as a trigger, cnBNG can dynamically establish, manage, and enforce policies for subscribers.

- Feature Summary, on page 475
- Revision History, on page 475
- cnBNG routed packet trigger subscriber sessions, on page 476
- Restrictions for routed packet trigger subscriber sessions, on page 477
- Configure routed packet trigger subscriber sessions, on page 477
- Monitor subscriber sessions, on page 481

Feature Summary

Table 104: Feature Summary

Applicable Product(s) or Functional Area	cnBNG
Applicable Platform(s)	SMI
Feature Default Setting	Disabled – Configuration Required
Related Changes in this Release	First Release

Revision History

Table 105: Revision History

Revision Details	Release
First introduced.	2026.04.0

cnBNG routed packet trigger subscriber sessions

The routed packet trigger subscriber session is an IPoE subscriber session initiation method that

- uses the first unclassified routed IPv4 packet to validate the network access path before dynamically creating subscriber sessions in Layer 3 environments
- identifies subscribers by utilizing their source IP address, parent interface, and User Plane (UP) name instead of traditional Layer 2 identifiers, and
- enables the Control Plane (CP) to apply subscriber policies and program the session on the UP through PFCP signaling.

Table 106: Feature History Table

Feature Name	Release Information	Feature Description
cnBNG routed packet trigger subscriber sessions	2026.04.0	You can now support subscriber session initiation in routed Layer 3 access networks where traditional Layer 2 discovery mechanisms are unavailable. This enables scalable, IP-based session management and dynamic policy enforcement for routed broadband subscribers. The feature uses the first unclassified routed IPv4 packet to trigger the Control Plane, which dynamically programs the User Plane via PFCP.

Routed packet trigger subscriber sessions

Traditional cnBNG subscriber session models depend on Layer 2 identifiers and discovery mechanisms such as MAC addresses, DHCP packets, or PPPoE discovery messages. In routed access environments, these identifiers are not available to the Broadband Network Gateway.

Because subscriber traffic reaches the UP as routed IP traffic, cnBNG cannot dynamically create subscriber sessions by using traditional Layer 2 packet-trigger workflows.

The routed packet trigger subscriber session feature addresses this limitation by using the first unclassified routed IPv4 packet as the trigger for subscriber session creation.

When the UP receives an unclassified routed IP packet, the UP forwards the trigger information to the CP, and the CP dynamically creates the subscriber session through PFCP signaling.

To uniquely identify and maintain subscriber sessions in routed access deployments, the feature uses IP-based session identifiers instead of Layer 2 identifiers such as MAC addresses.

Subscriber session identifiers

Identifier	Description
Subscriber source IP address	Identifies the subscriber session
Parent interface	Identifies the access interface associated with the subscriber
UP name	Identifies the UP handling the subscriber session

Benefits of routed packet trigger subscriber sessions

- Supports routed subscriber access deployments
- Eliminates dependency on Layer 2 discovery mechanisms
- Enables IP-based packet-triggered subscriber session creation
- Supports dynamic policy enforcement for routed IP subscribers
- Provides scalable subscriber session management for routed broadband networks

Restrictions for routed packet trigger subscriber sessions

- IP addresses can’t be reused in same UP.
- Access-Interface and Subscriber should be configured in same VRF.
- If an Access-Interface is used for Routed-Packet-Trigger subscribers, it can’t be used for any other subscriber type.

Configure routed packet trigger subscriber sessions

Configure routed packet trigger subscriber sessions to enable cnBNG to dynamically create IPoE subscriber sessions when the UP receives the first unclassified routed IPv4 packet on an access interface.

Follow these steps to configure routed packet trigger subscriber sessions.

Procedure

Step 1 Create a profile template for the base and accounting feature.

a) Configure feature template with idle-timeout.

Include the idle-timeout command to configure the subscriber inactivity period.

Example:

```
profile feature-template ipoe-template
 idle-timeout 600
 ipv4
```

```
  mtu 1500
 exit
exit
```

b) Configure accounting template with periodic-interval.

Example:

```
profile feature-template acct-template
 session-accounting
  enable
  aaa-profile aaa-prof1
  periodic-interval 1200
 exit
exit
```

c) Set attribute format for routed-ipoe.

Example:

```
profile attribute-format routed-ipoe
format-string %s@ip.broadband.is
format-order [ addr ]
exit
```

d) Define AAA profile for authentication, authorization, and accounting.

Example:

```
profile aaa aaa-prof1
authentication
method-order [ rad-server ]
exit
authorization
type subscriber method-order [ rad-server ]
username format routed-ipoe
password cisco
exit
accounting
method-order [ rad-server ]
exit
exit
```

Step 2 Activate the feature templates within the subscriber profile and authorize the subscriber.

Activate the required feature templates within the subscriber profile, specify the source IP address ranges using the matches command to identify valid trigger traffic, and configure AAA authorization to validate the subscriber.

Example:

```
profile subscriber subs-routed-pkt-trg
 class class-pkt-trg
  activate-feature-templates [ ipoe-template acct-template]
  matches
   match-type any
   match ipv4-source-address ascii [ 10.0.0.0/24 20.0.0.0/24]
  exit
  aaa authorize aaa-prof1
 exit
exit
user-plane
 instance 1
  user-plane asr9k-1
   peer-address ipv4 1.1.1.1
   subscriber-profile subs-routed-pkt-trg
  exit
```

```
 exit
exit
```

Step 3 Enable routed packet trigger subscriber sessions on the cnBNG UP.

Example:

```
interface Bundle-Ether1.100
 cnbng-nal
  ipsubscriber
   ipv4 routed
    initiator unclassified-ip
   !
  !
 !
```

Step 4 Verify the configuration in CP.

a) Verify the session count in the CP cluster.

Example:

```
bng# show subscriber session count
subscriber-details
{
  "sessionCount": 1
}
[cnbng] bng#
```

b) Verify the routed subscriber details using the show subscriber session detail command.

Ensure the `sesstype` is *routed-pkt-trg-ipoe* and `protocol-type` as *ip(1)*.

Example:

```
bng# show subscriber session detail
subscriber-details
{
  "subResponses": [
    {
      "subLabel": "16777218",
      "acct-sess-id": "01000002",
      "upf": "asr9k-1",
      "port-id": "Bundle-Ether1.100",
      "up-subs-id": "5",
      "sesstype": "routed-pkt-trg-ipoe",
      "state": "established",
      "subCreateTime": "Fri, 14 Aug 2026 05:20:04 UTC",
      "transId": "1",
      "initUpfName": "asr9k-1",
      "subsAttr": {
        "attrs": {
          "Authentic": "RADIUS(1)",
          "addr": "10.0.0.1",
          "physical-adapter": "5",
          "physical-chassis": "1",
          "physical-port": "4",
          "physical-slot": "2",
          "physical-subslot": "3",
          "port-type": "IP over Ethernet(39)",
          "protocol-type": "ip(1)",
          "string-session-id": "01000002",
          "username": "cnbng",
          "vrf": "vrf1"
        }
```

```
}
      },
```

Step 5 Verify the configuration on user plane.

a) Ensure that the IPoE routed session count for the 'activated' state matches the total expected session count.

Example:

```
Router-UP#show cnbng-nal subscriber all summary
=====================
Location: 0/RSP0/CPU0
=====================
                Type            PPPoE                   IPoE                    LAC
LNS

                ====            =====                   ====                    ===
===

                                                    L2-Conn         Routed

                                                    =======         ======

Session Counts by State:

        initializing            0               0               0               0
0

          connecting            0               0               0               0
0

           connected            0               0               0               0
0

           activated         0               0               1               0               0

                idle            0               0               0               0
0

       disconnecting            0               0               0               0
0

              Total:            0               0               1               0
0

Session Counts by Address-Family:

                none            0               0               0               0
0

                ipv4            0               0               1               0
0

                ipv6            0               0               0               0
0

                dual            0               0               0               0
0

                 lac            0               0               0               0
0

              Total:            0               0               1               0
0
```

b) Verify that the configured idle timeout matches the value applied on the device.

Example:

```
show cnbng-nal subscriber all detail

Routed IPv4 Prefix:      10.0.0.1
Cover Route Prefix:      4.1.1.0/24
Cover Route State:       Up
Inactivity Timeout:      600 sec(s)
```

What to do next

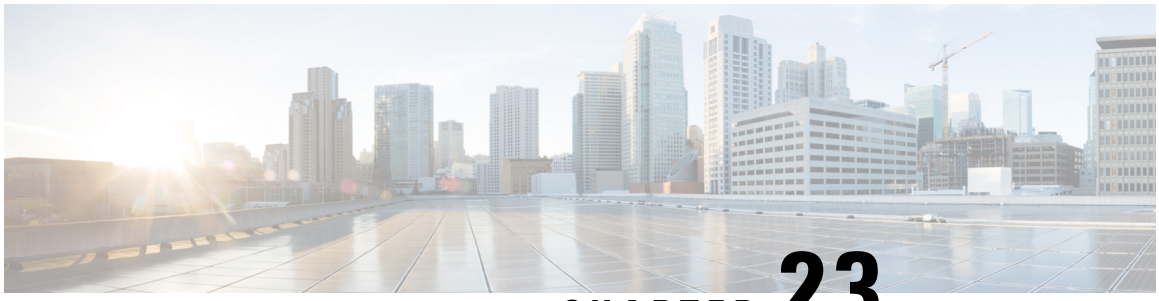
Monitor Protocol and Subscriber, on page 377.

Monitor subscriber sessions

The Monitor Subscriber captures all the transactional logs for a given subscriber over a specified period of time. It also supports the simultaneous monitoring of multiple subscribers on a given cluster. This information allows to track all the events that had occurred for a given subscriber when the subscriber was coming up or going down. For more information, see Monitor Protocol and Subscriber, on page 377.

Use subscriber IP-address and UPF to monitor subscriber for routed packet trigger sessions.

```
bng# monitor subscriber supi 10.0.0.1@asr9k-1
```



CHAPTER 23

CP Geographical Redundancy

- Feature Summary , on page 483
- Revision History, on page 484
- Feature Description, on page 484
- Prerequisites for CP-GR Cluster Bring Up, on page 484
- CP-GR Network Slicing Requirements, on page 486
- Architecture, on page 490
- Active-Active GR Deployment, on page 491
- MED Value, on page 493
- Geo Redundancy Support for AIO Control Plane Cluster, on page 494
- Geographical redundancy for L2TP sessions, on page 496
- GR-Replication Pod, on page 499
- ETCD and Cache Pod Replication , on page 500
- Pod Monitoring, on page 501
- Traffic Monitoring , on page 501
- Instance Roles , on page 503
- Automated standby-state recovery, on page 504
- IPAM, on page 505
- Limitations and Restrictions, on page 505
- Configuring CP Geo-Redundancy, on page 506
- Key Performance Indicators (KPIs), on page 527
- Monitoring and Troubleshooting, on page 530

Feature Summary

Table 107: Summary Data

Applicable Product(s) or Functional Area	cnBNG
Applicable Platform(s)	SMI
Feature Default Setting	Disabled – Configuration Required
Related Changes in this Release	First Release

Related Documentation	Not Applicable

Revision History

Table 108: Revision History

Revision Details	Release
Introduced support for automated standby-state recovery.	2025.02.0
Introduced CP Geo Redundancy support for PPPoE sessions.	2025.01.0
Introduced CP Geo Redundancy support for AIO Control Plane Cluster.	2024.03.0
Introduced Traffic Monitoring functionality for CP-GR sites.	2024.03.0
Introduced support for cnBNG to prepend the AS-path attribute to BGP Virtual IP (VIP) routes.	2024.02.0
Introduced support for BGP IPv6 route advertisement and IPv6 neighbor peering.	2024.02.0
First introduced.	2024.01.0

Feature Description

CP Geographical redundancy provides protection to the cnBNG Control Plane site against service failures that occur due to natural disasters or massive system outages such as power failures. CP Geo redundancy takes place through replication of sessions, and any other data required for seamless failover and failback of services to the remote site.



Note CP Geo redundancy feature is supported for IPoE and PPPoE sessions.

Prerequisites for CP-GR Cluster Bring Up

The following are prerequisites for bringing up the CP-GR cluster:

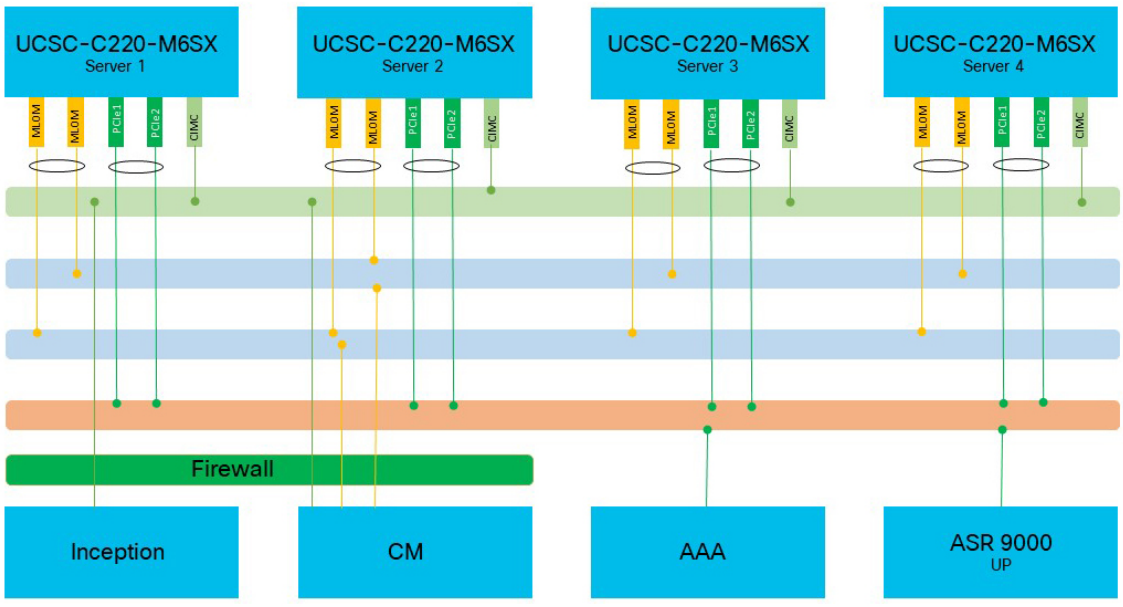
- You can use either Cluster Manager (CM) or Inception server to bring up CP-GR K8s cluster.
- The CIMC subnet of servers must be reachable from Inception or CM.
- The management VLAN can use /28 or /24 subnet masks on Modular LAN On Motherboard (MLOM) bond.
- The customer network can use /29 subnet mask on PCIe bond.

- You can use the number of servers depending on the scale requirements. You need a minimum of three servers per site for the CP-GR cluster to achieve both cluster and local level redundancy.
- You can use a firewall based on your deployment requirement.
- You can use UCS C220M6 or M7SX servers.

Port Connections per CP-GR Site

The following diagram illustrates the port connections per CP-GR site.

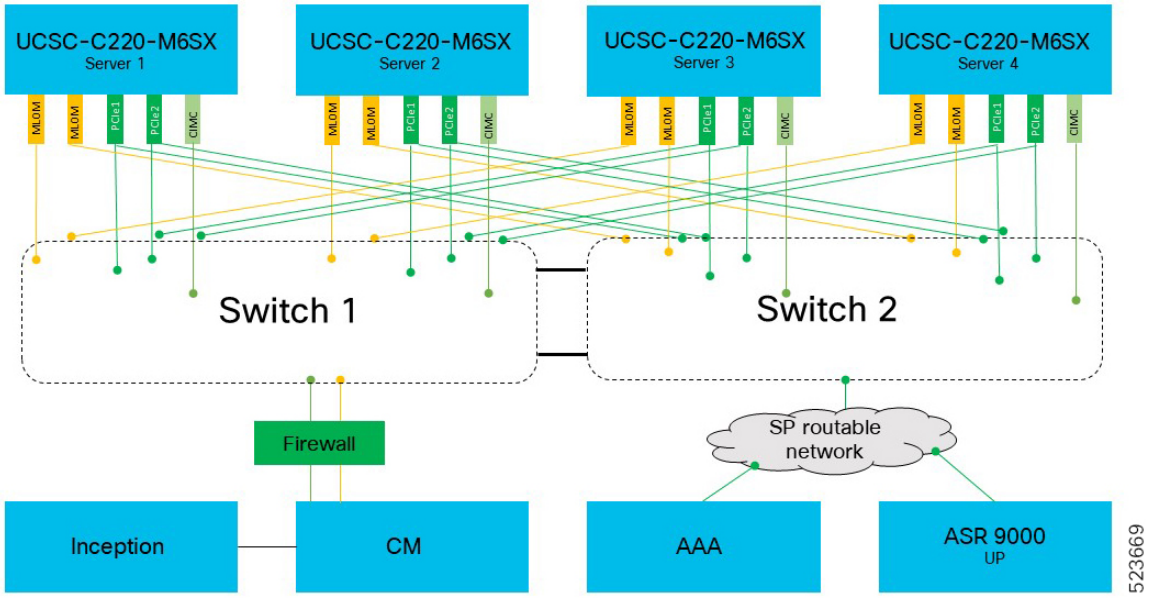
Figure 27: Port Connectivity per CP-GR Site



Physical Connections per CP-GR Site

The following diagram illustrates the physical connections per CP-GR site.

Figure 28: Physical Connections per CP-GR Site



CP-GR Network Slicing Requirements

The following are the CP-GR network slicing requirements:

- You can use VLANs as per your network requirements.
- You can use the same or different VLANs between CP-GR sites.
- VLANs and addresses such as cdl vips, udp vips, and inittcp vips must be reachable from the other site of CP-GR cluster.
- You can use VLANs on different port bundles as mentioned in the following network slicing example.

Sample Network Slicing Details - Site 1

Site 1							
Type	**Description**	**Node**	**Physical Interface**	**Logical Interface**	**VLAN**	**Physical IP**	**Vip IP if any**
BGP	bgp between proto1 and leaf1 server-1	proto1/svr1	enp94s0f0	enp94s0f0.151	151	10.1.1510.1/29	N.A
	bgp between proto1 and leaf2 server-1	proto1/svr1	enp216s0f0	enp216s0f0.152	152	10.1.152.1/29	N.A
	bgp between proto2 and leaf1 server-2	proto2/svr4	enp94s0f0	enp94s0f0.151	151	10.1.151.2/29	N.A
	bgp between proto2 and leaf2 server-2	proto2/svr4	enp216s0f0	enp216s0f0.152	152	10.1.152.2/29	N.A
N4 External VIP	N4 VIP. External VIP	proto1/svr1	bd2	bd2.n4.161	161	10.1.1610.1/29	N4 Site 1 - 209.165.200.1/32 N4 Site 209.165.200.2/32 You can use different addresses
		proto2/sv4	bd2	bd2.n4.161	161	10.10.161.2/29	
N4 Internal VIP	N4 Internal VIP	proto1/svr1	bd2	bd2.intudp.163	163	10.1.163.1/29	10.1.163.100/32
		proto2/svr4	bd2	bd2.intudp.163	163	10.1.163.2/29	
Geo	Geo POD: Active/Standby Internal and External VIP. Grouping required for both	proto1/svr1	bd1	bd1.inttcp.164	164	10.1.164.1/29	Internal: 10.1.164.100/32 Ext: 10.1.164.101/32 You can use different addresses
		proto2/svr4				10.1.164.2/29	
		svr2				10.1.164.3/29	
		svr3				10.1.164.4/29	

Site 1							
Type	**Description**	**Node**	**Physical Interface**	**Logical Interface**	**VLAN**	**Physical IP**	**Vip IP if any**
CDL	CDL services	svr-2	bd1	bd1.cdl.165	165	10.1.165.1/29	CDL: 10.1.165.100, Kafka1: 10.1.165.101, Kafka2: 10.1.165.102. You can use different addresses
		svr-3	bd1	bd1.cdl.165	165	10.1.165.2/29	
K8s mgmt	K8s Management IP address VLAN 125	Primary1	eno5 & eno6	bd0.k8s.125	125	10.100.3.1/28	10.1.125.10/28 gw-10.1.125.101 You can use different addresses
		Primary2	eno5 & eno6	bd0.k8s.125	125	10.100.3.2/28	
		Primary3	eno5 & eno6	bd0.k8s.125	125	10.100.3.3/28	
		worker1	eno5 & eno6	bd0.k8s.125	125	10.100.3.4/28	
mgmt	Management IP address	Primary1	eno5 & eno6	bd0.mgmt.325	325	10.100.2.11/28	10.100.2.10/24 You can use different addresses
		Primary2	eno5 & eno6	bd0.mgmt.325	325	10.100.2.12/28	
		Primary3	eno5 & eno6	bd0.mgmt.325	325	10.100.2.13/28	
		worker1	eno5 & eno6	bd0.mgmt.325	325	10.100.2.14/28	

Sample Network Slicing Details - Site 2

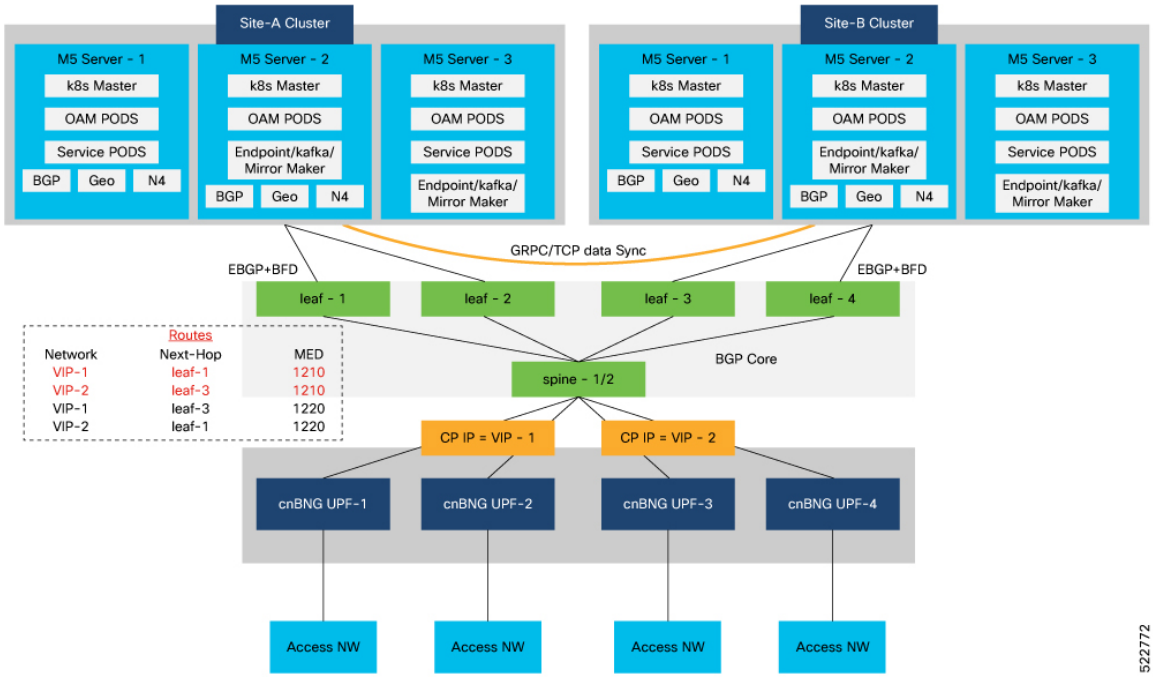
Site-2							
Type	**Description**	**Node**	**Physical Interface**	**Logical Interface**	**VLAN**	**Physical IP**	**Vip IP if any**
BGP	bgp between proto1 and leaf1 server-1	proto1/svr1	enp94s0f0	enp94s0f0.151	151	10.2.151.1/29	N.A
	bgp between proto1 and leaf2 server-1	proto1/svr1	enp216s0f0	enp216s0f0.152	152	10.2.152.1/29	N.A
	bgp between proto2 and leaf1 server-2	proto2/svr4	enp94s0f0	enp94s0f0.151	151	10.2.151.2/29	N.A
	bgp between proto2 and leaf2 server-2	proto2/svr4	enp216s0f0	enp216s0f0.152	152	10.2.15102/29	N.A
N4 External VIP	N4 VIP. External VIP	proto1/svr1	bd2	bd2.n4.161	161	10.2.161.1/29	N4 Site 1- 209.165200.1/32 N4 Site 2- 209.165200.2/32
		proto2/sv4	bd2	bd2.n4.161	161	10.2.161.2/29	
N4 Internal VIP	N4 Internal VIP	proto1/svr1	bd2	bd2.intudp.163	163	10.2.163.1/29	102.163.200/32
		proto2/svr4	bd2	bd2.intudp.163	163	10.2.163.2/29	
Geo	Geo POD: Active/Standby Internal and External VIP. Grouping required for both	proto1/svr1	bd1	bd1.inttcp.164	164	10.2.164.1/29	Internal: 102.164.200/32 Ext: 102.164.201/32 You can use different addresses
		proto2/svr4				10.2.164.2/29	
		svr2				10.2.164.3/29	
		svr3				10.2.164.4/29	
CDL	CDL services	svr-2	bd1	bd1.cdl.165	165	10.2.165.1/29	CDL: 10.2.165.200, Kafka1: 10.2.165.201, Kafka2: 10.2.165.202
		svr-3	bd1	bd1.cdl.165	165	10.2.165.2/29	

Site-2							
Type	**Description**	**Node**	**Physical Interface**	**Logical Interface**	**VLAN**	**Physical IP**	**Vip IP if any**
K8s Management	K8s Management IP address VLAN 125	Primary1	eno5 & eno6	bd0.k8s.125	125	10.200.3.1/28	10.2.126.10/28 gw-10.2.126.101 You can use different addresses
		Primary2	eno5 & eno6	bd0.k8s.125	125	10.200.3.2/28	
		Primary3	eno5 & eno6	bd0.k8s.125	125	10.200.3.3/28	
		worker1	eno5 & eno6	bd0.k8s.125	125	10.200.3.4/28	
Management	Management IP address	Primary1	eno5 & eno6	bd0.mgmt.325	325	10.100.2.14/28	10.100.2.20/24 You can use different addresses
		Primary2	eno5 & eno6	bd0.mgmt.325	325	10.100.2.15/28	
		Primary3	eno5 & eno6	bd0.mgmt.325	325	10.100.2.16/28	
		worker1	eno5 & eno6	bd0.k8s.125	325	10.100.2.16/28	

Architecture

The following figure shows two sites with cnBNG cluster that is connected to the spine-leaf BGP core network.

Figure 29: cnBNG CP Geo Redundancy Architecture



Each cnBNG cluster runs BGP and Geo redundancy pods on Protocol node. The protocol node provides high availability using active-standby topology.

BGP speaker pod runs on protocol node where the BGP routing protocol is hosted. It also runs BFD protocol for detecting BGP link failures.

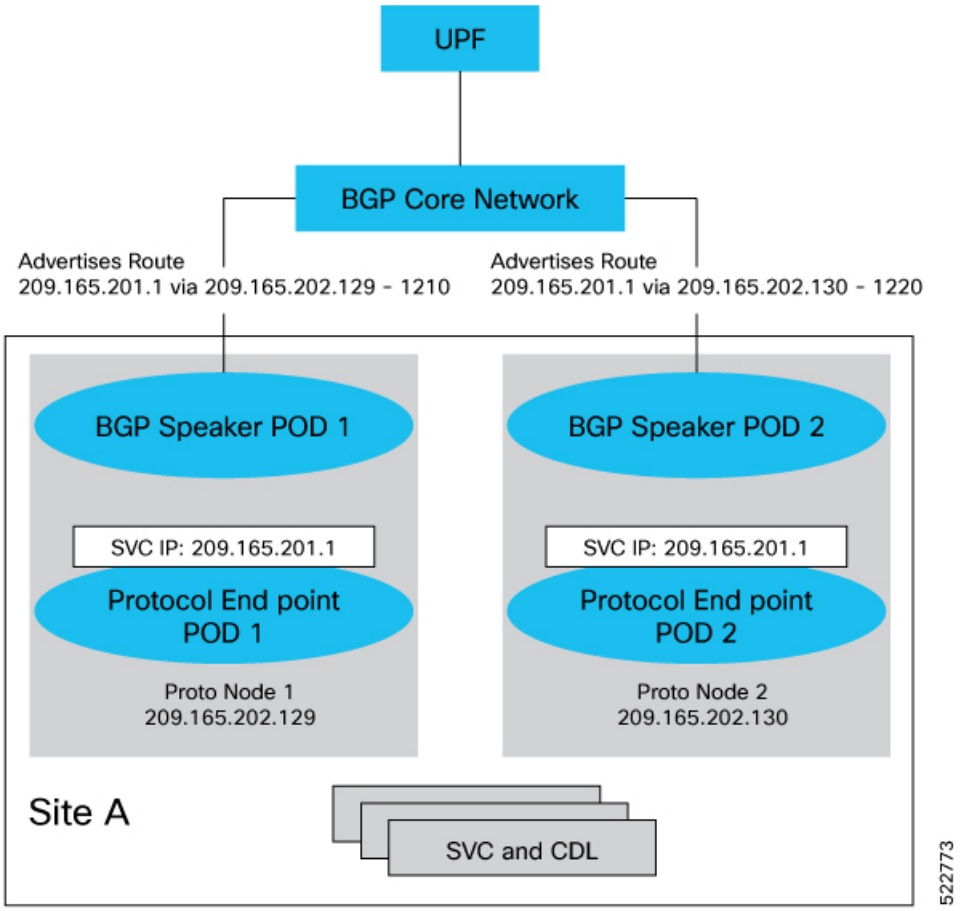
The following sequence of steps set up the BGP speaker pods:

- The BGP speaker pods use TCP as the transport protocol, on port 179. These pods use the AS number that is configured in the Ops Center CLI.
- Connection is established with all the BGP peers provided by the Ops Center CLI.
- All VIP IP addresses of endpoints, which are configured in the Ops Center CLI are published.
- The import policies for routing are configured using CLI configuration.
- Similar to the cache pod, two BGP speaker pods run on each Namespace as Active-Active.

Active-Active GR Deployment

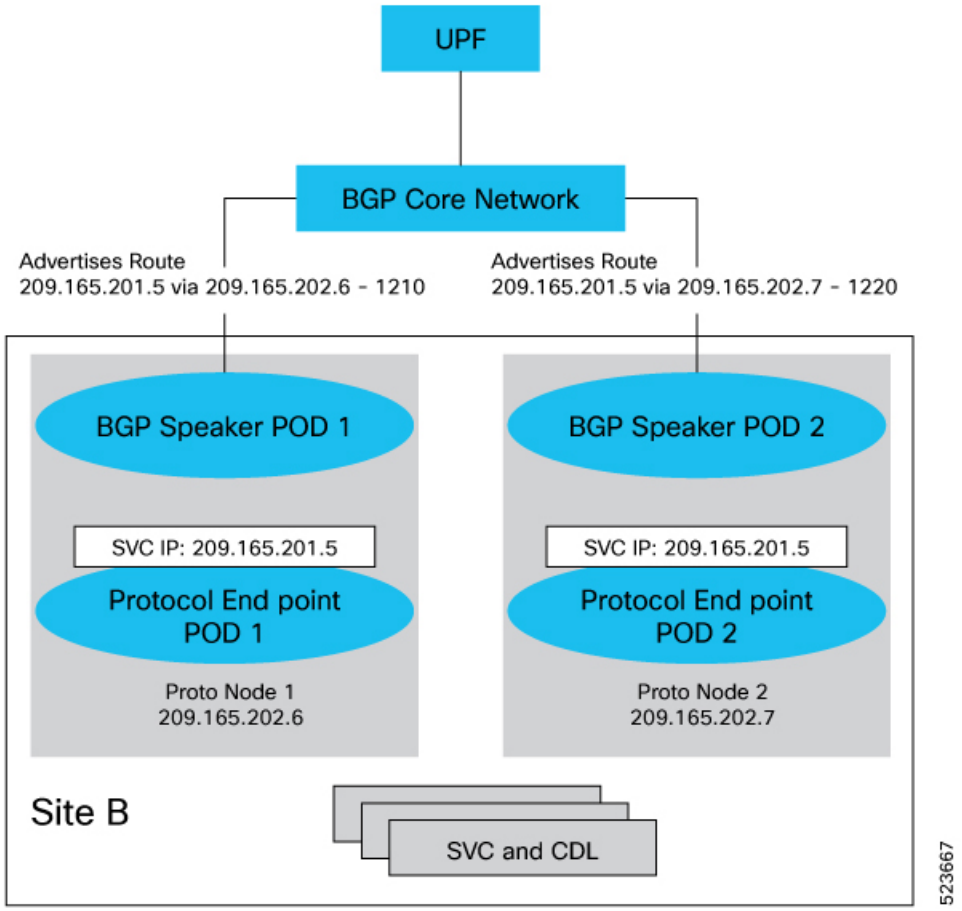
The following figure illustrates the dynamic routing of Active-Active GR deployment, consisting of site A and site B.

Figure 30: Site A



The Service IP address 209.165.201.1 is configured on both the nodes 209.165.202.129 and 209.165.202.130. POD1 is running on host 209.165.202.129 and POD2 on 209.165.202.130. The host IP address exposes the pod services. BGP speaker publishes the route 209.165.201.1 through 209.165.202.129 and 209.165.202.130. It also publishes the MED values 1210 and 1220 to determine the priority of pods.

Figure 31: Site B



MED Value

The Local Preference is used only for IGP neighbours, whereas the Multi Exit Discriminator (MED) Attribute is used only for EGP neighbours. A lower MED value is the preferred choice for BGP.

Table 109: For Primary Role:

Bonding Interface Active	VIP Present	MED Value	Local Preference
Yes	Yes	1210	100
Yes	No	1220	100
No	Yes	1215	100
No	No	1225	100

Table 110: For Standby Role:

Bonding interface active	VIP present	MED value	Local Preference
Yes	Yes	2210	100
Yes	No	2220	100
No	Yes	2215	100
No	No	2225	100

Table 111: For Non Primary/Standby Role:

Bonding interface active	VIP present	MED value	Local Preference
NA	NA	3220	100

BGP Speaker POD periodically checks the VIP status, and Active interface of bonded interface on Protocol node. If a change is detected, then the BGP re-advertises routes based on the VIP/bonded interface state.

Geo Redundancy Support for AIO Control Plane Cluster

Table 112: Feature History

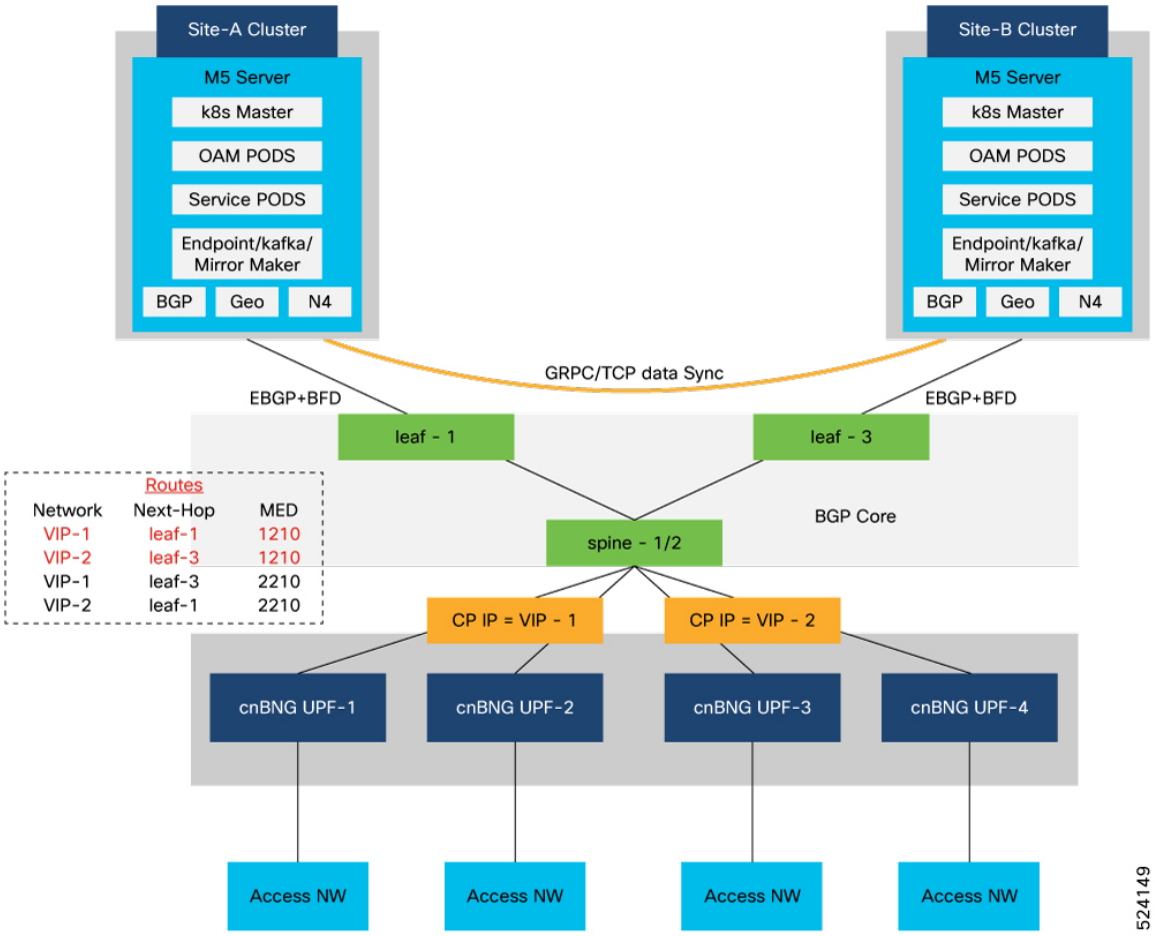
Feature Name	Release Information	Description
Geo Redundancy Support for AIO Control Plane Cluster	2024.03.0	This feature provides higher service availability using Geographical Redundancy for cnBNG all-in-one (AIO) Control Plane clusters, reducing the risk of outages. CP-GR feature, previously available for 3-server and 4-server clusters, has been expanded to include AIO clusters.

The implementation of Geographical redundancy on AIO control plane clusters allows for continuous operation even if one AIO Control Plane cluster experiences a failure, maintaining uninterrupted services.

Architecture

This figure shows two sites with cnBNG AIO Control Plane cluster that is connected to the spine-leaf BGP core network.

Figure 32: AIO CP Geo Redundancy Architecture



Each AIO cluster runs a single instance of BGP-Speaker pod and Geo-Replication pod.

You can use a single server to achieve GR functionality on an AIO CP cluster.

The BGP-Speaker pod is equipped with BGP routing protocol and BFD protocol, providing rapid BGP link failure detection to maintain uninterrupted network connectivity. It establishes secure connections with all BGP peers as per the user-defined configuration, fostering a reliable routing environment.

Furthermore, the BGP-Speaker pod dynamically publishes IPv4 and IPv6 Virtual IP addresses (VIPs) of endpoints, facilitating efficient network traffic management. It also imports and enforces routing policies based on the configuration settings.

CP-GR Switchover Scenarios

The scenarios that can trigger CP-GR switchover on an AIO cluster are:

- Pod monitoring failure
- BGP/BFD monitoring failure
- POD restart of bgpspeaker-pod
- POD restart of georeplication-pod

- Traffic monitoring failure, and
- Manual switchover using **geo switch-role** command.

Active-Active GR Deployment

The BGP speaker advertises Service IP addresses to manage incoming traffic from both the sites (site A and site B).

eBGP is configured between the AIO-site Control Plane and the Leaf switches, and iBGP is set up between Leaf1/2 and Spine switches. The use of Local Preference within iBGP neighbors allows for precise internal traffic prioritization, while the MED Attribute for eBGP neighbors dictates external routing preferences.

Routes advertised with a lower MED value are preferred in the BGP selection process, guiding traffic towards the most efficient path.

Geographical redundancy for L2TP sessions

Geographical redundancy for L2TP sessions is a solution for service providers that

- extends existing control plane geographical redundancy (CP-GR) to L2TP tunnels and sessions
- helps maintain session continuity and service availability during control plane (CP) failover events, and
- enables seamless failover and recovery of L2TP tunnels and sessions without manual intervention.

This feature supports the same redundancy capabilities previously available for IPoE and PPPoE, now extended to L2TP sessions. When a CP failover occurs, the system synchronizes critical state information to ensure that L2TP tunnels and sessions remain operational, minimizing service disruption for end users.

Table 113: Feature History

Feature Name	Release Information	Description
Geographical redundancy for L2TP sessions	2025.04.0	You can now maintain L2TP subscriber sessions without disruption during control plane failover using control plane geographical redundancy (CP-GR). When a failover occurs, the control plane automatically synchronizes critical L2TP tunnel and session state with the user plane, allowing tunnels and sessions to continue seamlessly.

Limitations of configuring geographical redundancy for L2TP sessions

- Do not reload or upgrade the User Plane while L2TP tunnels and sessions are active. Reloading or upgrading will disconnect all active tunnels and sessions.
- L2TP LAC sessions can be established over a SRG, but these sessions are disconnected after an SRG switchover.
- OpenShift deployment is currently not supported.

Configure CP-GR support for L2TP tunnels and sessions

Enable geographical redundancy (GR) for L2TP tunnels and sessions on the User Plane to support failover protection across remote sites.

Procedure

Step 1 Enable geographical redundancy for L2TP tunnels and sessions on the specific User Plane.

Example:

```
Router# configure
Router(config)# cnbng-nal location 0/RSP0/CPU0
Router(config-cnbng-nal-local)# l2tp track-ns-nr
```

Step 2 Use the **show cnbng-nal l2tp-tunnel** [*tunnel-id*] command on the User Plane to display tunnel context details for a specific tunnel or all tunnels.

Example:

```
show cnbng-nal l2tp-tunnel

Tunnel ID              NS              NR
=========             ====            ====
11                    500             501
12                    450             451
```

Step 3 Use the **show l2tp-tunnel detail instance-id** *instance_id* command on the Control Plane to view the configuration and status of a specific L2TP tunnel.

Example:

```
bng# show l2tp-tunnel detail instance-id 1
Wed Jul  2  11:27:52.590 UTC+00:00
tunnel-details
{
  "tunResponses": [
    {
      "state": "established",
      "profileName": "lns-prof1",
      "tunnelType": "lns",
      "sessionCount": 1,
      "IDs Allocated": 1,
      "routerID": "asr9k-lns",
      "srcIP": "41.41.41.1",
      "dstIP": "91.91.91.1",
      "localTunnelID": 39832,
      "remoteTunnelID": 6410,
      "tunnelClientAuthID": "Local-DC",
      "tunnelServerAuthID": "bng-lns"
    }
  ]
}
```

Step 4 Use the **show l2tp-tunnel count instance-id** *instance_id* command on the Control Plane to view the total number of L2TP tunnels.

Example:

```
bng# show l2tp-tunnel count instance-id 1
Thu Jul  3  15:55:02.172 UTC+00:00
```

```
tunnel-details
{
  "tunnelCount": 8996
}
```

Step 5 Use the **show subscriber lns count instance-id** *instance_id* command on the Control Plane to view the number of LNS sessions.

Example:

```
bng# show subscriber lns count instance-id 1
Thu Jul  3  15:55:04.665 UTC+00:00
subscriber-details
{
  "sessionCount": 55964
}
```

Use the **show subscriber pppoe count instance-id** *instance_id* command on the Control Plane to view the number of LAC sessions.

Step 6 Use the **show subscriber lns detail instance-id** *instance_id* command on the Control Plane to view the detailed information about a specific LNS session.

Example:

```
bng# show subscriber lns detail instance-id 1
Mon Oct  20 06:30:27.240 UTC+00:00
subscriber-details
{
  "subResponses": [
    {
      "state": "complete",
      "key": {
        "routerID": "asr9k-1",
        "portID": "Bundle-Ether10",
        "subLabel": "16777218",
        "upSubID": "1"
      },
      "flags": [
        "SM_START_DONE",
        "SM_ACTIVATE_DONE",
        "SM_UPDATE_DONE",
        "IPCP_UP",
        "IPV6CP_UP"
      ],
      "lcpInfo": {
        "state": "opened",
        "keepAliveInterval": 60,
        "keepAliveRetries": 5,
        "localMru": 1492,
        "peerMru": 1492,
        "localMagic": "0xe2ab84f",
        "peerMagic": "0xe2ab850",
        "authOption": "PAP",
        "authCompleted": true,
        "username": "cnbng"
      },
      "ipcpInfo": {
        "state": "opened",
        "peerIpv4Pool": "pool-ISP",
        "peerIpv4Address": "11.0.32.2",
        "peerIpv4Netmask": 22,
        "localIpv4Address": "11.0.32.1",
        "isIpamPoolIPaddr": true
      },
```

```
      "ipv6cpInfo": {
        "state": "opened",
        "localIntfID": "0x1",
        "peerIntfID": "0xcc11000000010001"
      },
      "lnsInfo": {
        "srcIP": "10.1.39.139",
        "dstIP": "10.1.34.52",
        "state": "established",
        "profileName": "l2tp-prof2",
        "tunnelClientAuthID": "bng-lac",
        "tunnelServerAuthID": "Local-DC",
        "callSerialNumber": 16777408,
        "localTunnelID": 12226,
        "localSessionID": 11428,
        "remoteTunnelID": 1017,
        "remoteSessionID": 24885
      },
      "sessionType": "lns",
      "vrf": "default",
      "AuditId": 4,
    }
  ]
}
```


Use the **show subscriber pppoe detail instance-id** *instance_id* command on the Control Plane to view the detailed information about a specific LAC session.

Synchronize L2TP tunnels between CP and UP for CP-GR

Procedure

Step 1 On the Control Plane, use the **subscriber tunnel-synchronize upf** *upf-name* command to manually trigger tunnel synchronization to UP after HA events.

Example:

```
bng# subscriber tunnel-synchronize upf upf1
```

Step 2 On the Control Plane, use the **show subscriber tunnel-synchronize upf** *upf-name* command to view the tunnel synchronization status.

Example:

```
bng# show subscriber tunnel-synchronize upf upf1
```

GR-Replication Pod

GR-replication pod performs the following functions:

- Replicates ETCD and Cache pods data across sites.

- Provides a communication channel between sites.
- Maintains local instances roles of a site in ETCD.
- Monitors local site status (pods status or BFD status or VIP status)

GR-replication pod works in a high availability (HA) setup to maintain the local instances roles of a site in ETCD. Monitoring (local and remote) is disabled in GR-Replication pod in a HA setup. When a site faces an issue, and fails to support the traffic handling at run-time, GR-replication pod internally detects the issue, and allows the standby site to handle the traffic with no or minimum impact.

GR-replication pod is a host networking pod, and it runs on actual worker IP address and not on IP address that is assigned internally by k8.

In a HA setup, one instance of GR-replication pod must be running, and activities related to GR setup such as pod monitoring, and VIP monitoring are not active.

In a GR setup:

- Two instances of GR-replication pod must be running for each cluster. One instance of GR-replication pod is active, and another instance is standby.
- Each GR-replication pod runs on a separate Proto node.
- GR-replication pod requires dedicated VIPs.
  - Internal-VIP for inter-pod (within the same cluster) communication.
  - External-VIP for communication with other clusters.
- The VIPs configured for GR-replication are active on one of the Proto nodes at a time. The GR-pod running on the same Proto node where the VIPs are active is marked as Active GR-replication pod, and the other GR-pod is marked as standby.
- If the active GR-pod is stopped or crashed during runtime, VIP (internal and external) switches to other Proto node, and the standby GR pod becomes Active. The switching of VIP from one Proto node to another Proto node is handled by Keepalived process.
- GR-replication pod uses base port as 15000 (default) + 4 for keepalived monitoring.

ETCD and Cache Pod Replication

Data from ETCD and Cache Pod are replicated to the remote site based on the following two categories:

- Immediate sync
- Deferred sync

Immediate Sync

Data that must be replicated immediately to the remote site belongs to the immediate sync category. Immediate sync data replication is a synchronous call, and replication failure on the remote site returns an error response. Data is replicated to the remote site only for instances whose role is PRIMARY.

Deferred Sync

Data that do not require immediate replication to the remote site belongs to the deferred sync category. This data is maintained in the in-memory cache in GR-replication pod. Data is replicated to the remote site only for instances whose role is PRIMARY.

Deferred sync happens periodically using background thread. Periodicity must be configured before deployment using the YAML file. By default, periodicity is set to 10s.

Deferred sync includes two processes that are executed in a single thread, which runs sequentially.

- **Deferred sync process:** Local site data is pushed to the remote site.
- **Checkpointing process:** Data of the instance whose role is PRIMARY on the remote site is pulled into the current site.

Pod Monitoring

You can configure each pod that need to be monitored. Based on the user configuration, GR-replication pod starts monitoring the pods and detects a pod failure. If the number of replica-sets failed for the pod is greater than the configured threshold, then the GR-replication pod switches over the Role to a mated pair. The current site moves to STANDBY_ERROR state indicating that the site has an issue and cannot serve the traffic.

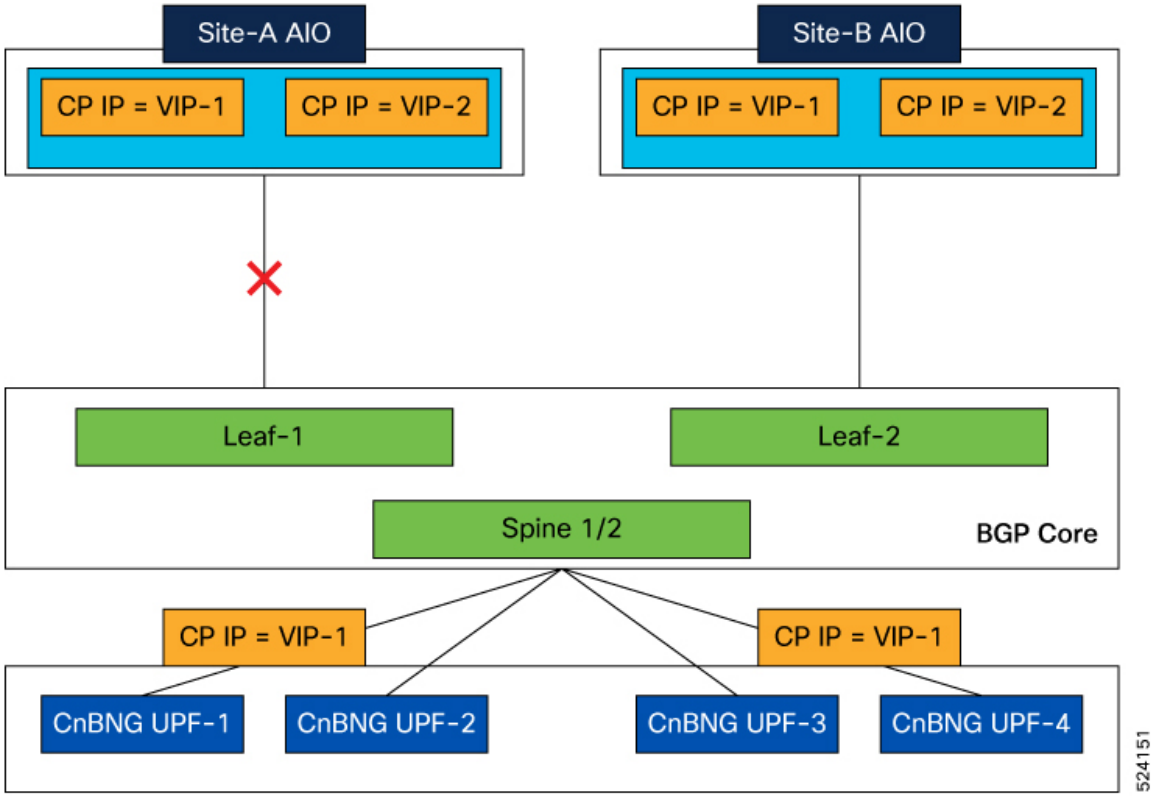
The detection request timeout interval for the first request is set at 2s, and for subsequent request it is set at 1s. In worst-case scenario, the total time to detect a pod failure is approximately 5s to 6s, with the total convergence time between 7s to 9s.

Note Pod monitoring on GR-replication pod starts after 15 minutes of its configuration.

Traffic Monitoring

This figure illustrates the geo-redundant setup between AIO sites.

Figure 33: GR Setup for AIO Sites



Our network architecture introduces a geo-redundant setup between Site-A and Site-B, ensuring continuous service availability. Geo-Redundancy (GR) instance-1 operates actively on Site-A and remains on standby on Site-B, while GR instance-2 functions actively on Site-B with a standby role on Site-A.

In the event of a failure or connectivity loss at Site-A, the Leaf/Spine switches automatically withdraw the BGP routes associated with Site-A. This triggers the switches to reroute incoming traffic towards Site-B. The Traffic Monitoring feature on Site-B detects the surge in traffic and, upon reaching a predefined packet threshold, the GR-pod initiates a transition for GR Instance-1 from Standby to Active status.

Simultaneously, the BGP-Speaker pod on Site-B proactively readvertises the routes for the GR Instance-1 endpoint, assigning a lower MED value to prioritize these routes. This dynamic response ensures minimal service disruption and maintains optimal traffic flow.

Configure Traffic Monitoring

Use this configuration to enable traffic monitoring functionality.

```
config
 geomonitor
  trafficMonitor
   thresholdCount value
   thresholdInterval interval_value
  exit
 exit
```

NOTES:

- **thresholdCount** *value*: Specifies the number of calls received for the standby instance.
- **thresholdInterval** *interval_value*: Specifies the maximum duration window to hit the threshold count value, in milliseconds.

The following is a sample configuration.

```
geomonitor trafficMonitor
thresholdCount 3
thresholdInterval 3000
```

Instance Roles

Each GR setup site contains multiple instances and roles.

- PRIMARY: Site is ready and actively taking traffic for the given instance.
- STANDBY: Site is standby, ready to take traffic but not taking traffic for the given instance.
- STANDBY_ERROR: Site is in problem, not active and not ready to take traffic for the given instance.
- FAILOVER_INIT: Site has started to failover and not in condition to take traffic. Buffer time is 2 sec for application to complete their activity.
- FAILOVER_COMPLETE: Site has completed the failover and attempted to inform the peer site about the failover for given instance. Buffer time is 2 seconds.
- FAILBACK_STARTED: Manual failover is triggered with delay from remote site for the given instance

For fresh installation, site boots up with:

- Role PRIMARY for local instance (each site has local instance-id configured to identify local instance). It is recommended not to configure the pods for monitoring during fresh installation. Once the setup is ready, you can configure the pods for monitoring.
- Role STANDBY for other instances.

For upgrades, site boots up with:

- STANDBY_ERROR role for all the instances as moving the traffic post upgrade needs manual intervention.
- ETCD stores instance roles.



Note Rolling upgrade or in-service upgrade isn't supported.

After SMI cluster upgrades, GR instance roles may come up as PRIMARY/PRIMARY at site-1, and STANDBY/STANDBY at site-2 sometimes. To make local GR-instance role as PRIMARY on the given site, you must trigger CP-GR switchover manually using the **geo switch-role instance-id** *gr_instanceId* command.

Automated standby-state recovery

This feature addresses the issue of instances in a CPGR cluster remaining in a persistent "STANDBY_ERROR" state after a geo-replication (GR) switchover.

Table 114: Feature History

Feature Name	Release Information	Description
Automated standby-state recovery	2025.02.0	This feature eliminates manual intervention by automatically transitioning CPGR cluster instances from the "STANDBY_ERROR" state to "STANDBY" after a geo-replication switchover.

Current failure response:

When failures such as pod failure or BFD failure occur, the active geo-replication pod triggers a switchover, and the cluster with the failures transitions to "STANDBY_ERROR." These instances then require manual intervention to change their state to "STANDBY."

Automated state transition:

This feature automates the transition, automatically changing the state from "STANDBY_ERROR" to "STANDBY" once the failure condition is resolved.

How automated standby-state recovery works

The feature monitors only specific failures that lead to GR switchovers.

- **POD failure:**

  The system switches to STANDBY_ERROR if the failure rate of pods exceeds the configured threshold. For example, if there are 10 pods and the threshold is set at 50%, the cluster GR state changes to STANDBY_ERR when more than 50% (6 or more pods) fail. The remote cluster then becomes Primary. This transition is referred to GR switchover.

  However, the automatic state change from STANDBY_ERROR back to STANDBY occurs only when all pods are fully restored. This means all 10 pods must be up for the system to return to the STANDBY state. A partial recovery (if only 6 to 9 pods are up) will not trigger the autoswitch of GR state to STANDBY.

- **BFD failure:** The geo-replication pod continuously monitors BFD link health. The health monitoring routine introduced by this feature monitors the ETCD BFD status of the leaves. As soon as any leaf is UP, the BFD link is considered UP, and the instance automatically transitions to `STANDBY`.



Note Failures that do not cause GR switchover are not monitored by this feature.

IPAM

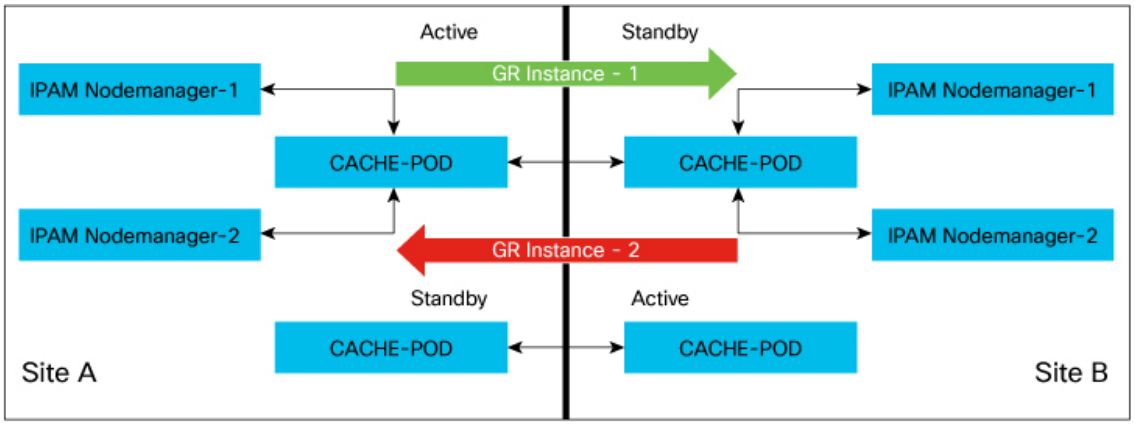
The IP Address Management (IPAM) is a technique for tracking and managing the IP address space of a network.

IPAM stores operational data of active instances in Cache-POD. Cache-POD records are synced to respective standby-cluster of the instances.

During GR switchover of an instance, the new active instance restores IPAM data from the Cache-POD, and continues to allocate IPs for the instance.

The following figure illustrates the IPAM architecture:

Figure 34: IPAM Architecture



- GR Instance-1 is the local instance of Site A and geo-paired with Site B.
- GR Instance-2 is the local instance of Site B and geo-paired with Site A.

During normal operation, Site A handles UPF-association/release, address-allocation/release for subscribers coming up in GR-instance-1. IPAM writes GR Instance-1 specific content to Cache-POD in the Site A cluster. Then, the IPAM's Cache-POD content is synced to the Site B's Cache-POD (geo-paired cluster).

Similarly, Site B handles UPF-association/release, address-allocation/release for subscribers coming up in GR-instance-2. IPAM writes GR Instance-2 specific content to Cache-POD in Site B cluster. Then, the IPAM's Cache-POD content is synced to Site A Cache-POD (geo-paired cluster).

When one of the clusters goes-down, the respective geo-paired cluster restores the content from local Cache-POD. For example, If Site B goes down, Site A gets role-change trigger for GR Instance-2, and IPAM in Site A restores the content of GR Instance-2 from local Cache-POD (which was already synced). Also, IPAM in Site A handles UPF-register/release, and address-allocate/release for subscribers coming up with GR Instance-2 using the restored content.

IPAM uses both "immediate-sync" and "deferred-sync" options to sync Cache-POD content between clusters.

Limitations and Restrictions

The CP Geo Redundancy has the following limitations and restrictions in this release:

- CP Geo Redundancy is not triggered if both the Geo pods are down or deleted. CP GR is triggered only after both the Geo pods are up.
- Restarting Kafka in one site and Mirrormaker pod on the other site is not supported.
- On system reboot, instances are not automatically associated with the right roles. You must set the roles correctly the first time.
- IP address leaks can occur in IPAM. To address this issue, run the **reconcile ipam** CLI command.
- Pod Monitoring is not supported for CDL pods and few App-infra pods.
- Subscriber sessions can desynchronize between the CP and UP. The solution for this issue is to run CP to UP reconciliation for sessions between the CP and UP.
- Node or pod restart can cause mismatch of session records between pods in the cluster. You must use CP audit to rectify this issue.

Configuring CP Geo-Redundancy

CP Geo-Redundancy configuration is classified into the following categories:

- **NF Configuration**—This configuration is similar on all GR instances of the NF.
- **Cluster Instance Specific NF Configuration**—This configuration contains cluster specific data, which differs on each GR instance of the NF.

Configuring NF Instance

Use the following configuration to configure the NF instance. Each NF instance is identified by a unique number.

```
config
   instances instance instance_id
   exit
```

Example:

```
config
   instances instance 1
   exit
   instances instance 2
   exit
```

Local Instance ID Configuration

The local Instance is configured using the **local-instance** command.

```
local-instance instance instance_id
```

Only two instances can be configured on each local and remote site, and corresponding endpoints can be instantiated.

A local instance-id is the identity of the local site irrespective of whether the site is GR aware.

Note Changing the local instance while the system is running is not supported.

Configuring Endpoints

You must configure the endpoints under an instance specified by a unique instance ID.

Use the following configuration to configure endpoints:

```
config
  instance instance-id gr_instanceId
    endpoint radius
      replicas replicas_count
      nodes nodes_count
      interface coa-nas
        vip-ip vip_ipv4_address vip-port vip_port_number vip-interface interface_id
        vip-ipv6 vip_ipv6_address vip-port vip_port_number vip-interface interface_id

      exit
    exit
    endpoint udp-proxy
      nodes nodes_count
      internal-vip vip_ip_address
      vip-ip vip_ipv4_address vip-port vip_port_number vip-interface interface_id
      vip-ipv6 vip_ipv6_address vip-port vip_port_number vip-interface interface_id

      interface n4
        sla response seconds
        vip-ip vip_ipv4_address vip-interface interface_name
        vip-ipv6 vip_ipv6_address vip-interface interface_name
      exit
      interface gtpu
        sla response milliseconds
      exit
```

NOTES:

- **instance instance-id** *gr_instanceId*: Specifies the GR instance ID.
- **endpoint radius**: Configures the parameters for the RADIUS endpoint and enters the endpoint sub-mode.
- **endpoint udp-proxy**: Configures the parameters for the UDP-proxy endpoint and enters the endpoint sub-mode.
- **replicas** *replicas_count*: Specifies the number of replicas per node. Must be an integer.

  In a GR setup for AIO CP cluster, the replica count must be 1 for BGP speaker pod, GR-replication pod, and UDP-proxy pod. For other pods, the replica count should be based on capacity planning.
- **nodes** *nodes_count*: Specifies the number of nodes. Must be an integer.
- **interface coa-nas** : Defines a new interface "coa-nas", and allows to enter the CoA NAS interface configuration mode.

- **interface n4** : Defines the N4 interface, and allows to enter the N4 interface configuration mode.
- **interface gtpu** : Defines the GTPu interface, and allows to enter the GTPu interface configuration mode.
- **vip-ip** *vip_ipv4_address* **vip-port** *vip_port_number* : Specifies the VIP IPv4 address, and VIP port number of the interface.
- **vip-ipv6** *vip_ipv6_address* **vip-port** *vip_port_number* : Specifies the VIP IPv6 address, and VIP port number of the interface.
- **vip-interface** *interface_id*: Specifies the VIP interface name.
- **internal-vip** *vip_ip_address*: Specifies the internal VIP IP address of the additional endpoint.
- **sla response** *response_time*: Specifies the response time in milliseconds.

Examples:

```
endpoint radius
   replicas 1
   nodes 2
   memory limit 16384
   interface coa-nas
     sla response 140000
     vip-ip 209.72.100.1 vip-port 3799 vip-interface bd2.n4.162
   exit
exit

instance instance-id 1
   endpoint pfcp
      vip-ipv6 2001::1 vip-interface bd2.n4.2105
   interface n4
      vip-ipv6 2001::10 vip-interface bd2.n4.2602
```

Configuring Geo Replication

Endpoints must be configured under an instance. Two Geo-Redundancy pods are needed on each GR site. You should also configure VIP for internal and external Geo interface for ETCD/CachePod replication.

instance instance-id *instance_id* **endpoint geo interface { geo-internal | geo-external } { vip-ip** *vip_ipv4_address* **| vip-ipv6** *vip_ipv6_address* **} vip-port** *vip_port_number*

```
config
  instance instance-id instance_id
   endpoint geo
    replicas replicas_count
    nodes node_count
    interface geo-internal
     vip-ip vip_ipv4_address vip-port vip_port_number
     vip-ipv6 vip_ipv6_address vip-port vip_port_number
    exit
    interface geo-external
     vip-ip vip_ipv4_address vip-port vip_port_number
     vip-ipv6 vip_ipv6_address vip-port vip_port_number
    exit
```

```
  exit
exit
```

NOTES:

- **instance instance-id** *instance_id*: Specifies the GR instance ID. One instance ID for local site and the other for remote site.
- **replicas** *replicas_count*: Specifies the number of replicas per node. Must be an integer.

  In a GR setup for AIO CP cluster, the replica count for GR-replication pod must always be 1.
- **vip-ip** *vip_ip_address*: Specifies the VIP IPv4 address for Internal/External Geo interface.
- **vip-ipv6** *vip_ipv6_address*: Specifies the VIP IPv6 address for Internal/External Geo interface.
- **vip-port** *vip_port_number*: Specifies the VIP port number.

The following is a sample configuration:

```
instance instance-id 1
endpoint geo
  replicas 1
  nodes 2
  interface geo-internal
   vip-ip 209.165.201.8 vip-port 7001
  exit
  interface geo-external
   vip-ipv6 2001:DB8:1::1 vip-port 7002
  exit
exit
```

Configuring IPAM

You can configure all the IPAM parameters under an instance specified by a unique instance ID.

Configuring RADIUS

The following is a sample RADIUS configuration:

```
profile radius
 attribute
  nas-identifier CISCO-BNG-SITE-2
  instance 1
   nas-identifier CISCO-BNG-1
   nas-ip 209.165.100.1
  exit
  instance 2
   nas-identifier CISCO-BNG-2
   nas-ip 209.166.100.2
  exit
 exit
 accounting
  deadtime 3
  attribute
   instance 1
    nas-identifier cisco-acct-1
    nas-ip 209.165.100.1
   exit
   instance 2
```

```
    nas-identifier cisco-acct-2
    nas-ip 209.166.100.2
   exit
  exit
 exit
```

NOTES:

- **instance** *instance_id*: Configures multiple instances for the specified instance and enters the instance sub-mode.
- **nas-identifier** *value*: Specifies the attribute name by which the system will be identified in Accounting-Request messages. *value* must be an alphanumeric string.
- **nas-ip** *ipv4_address*: Specifies the NAS IPv4 address. *ipv4_address* must be an IPv4 address in dotted decimal notation.
- **deadtime** *value*: Sets the time to elapse between RADIUS server marked unreachable and when we can re-attempt to connect.

  value must be an integer from 0 through 65535. Default: 10 minutes.

Configuring a Subscriber Profile

Use the following commands to create a subscriber profile.

```
config
  profile subscriber subscriber_profile
      dhcp-profile dhcp_profile_name
      pppoe-profile pppoe_profile_name
      session-type { ipv4 | ipv4v6 | ipv6 }
      activate-feature-template feature_template_name
      aaa authorize aaa_profile_for_authorization }
    exit
```

NOTES:

- **profile subscriber** *subscriber_profile_name*: Specifies the profile subscriber name and enters the Profile Subscriber Configuraton mode.
- **dhcp-profile** *dhcp_profile_name*: Associates the DHCP first sign of life (FSOL) profile.
- **pppoe-profile** *pppoe_profile_name*: Associates the PPPoE FSOL profile.
- **session-type { ipv4 | ipv4v6 | ipv6 }**: Specifies the allowed session-types as IPv4, IPv4v6, and IPv6.
- **activate-feature-templates** *feature_template_name*: Specifies the list of feature-templates in sequence for activation.
- **aaa { authenticate** *aaa_profile_for_authentication* **| authorize** *aaa_profile_for_authorization* **}**: Specifies the AAA profile to associate for authentication and authorization.

Configuring the IPv4 DHCP Server Profile

Use the following commands to configure the IPv4 DHCP server profile.

```
config
  profile dhcp dhcp_profile_name
    ipv4
      mode server
      server
        pool-name ipam_pool_name
        dns-servers dns_server
        lease days value
        lease hours value
        lease minutes value
      exit
  exit
```

NOTES:

- **profile dhcp** *dhcp_profile_name*: Specifies the DHCP profile name.
- **ipv4**: Enters IPv4 configuration mode.
- **server** : Specifies the IPv4 server details.
  - **dns-servers** *dns_server*: Specifies the Domain Name System (DNS) IPv4 servers available to a DHCP for an IPv4 client.
  - **pool-name** *ipam_pool_name*: Specifies the IP Address Management (IPAM) assigned pool name.
  - **lease { days** *value* **| hours** *value* **| minutes** *value* **}**: Specifies the lease time duration in the number of days, hours, and minutes. The number of lease days supported is from 0 to 365. The number of lease hours supported ranges from 0 to 23 and minutes from 0 to 59.

Configuring the IPv6 DHCP Server Profile

Use the following commands to configure the IPv6 DHCP server profile.

```
config
  profile dhcp dhcp_profile_name
     ipv6
      mode server
      server
        iana-pool-name ipam_pool_name
        iapd-pool-name ipam_pool_name
        lease days value
        lease hours value
        lease minutes value
        exit
      exit
```

NOTES:

- **profile dhcp** *dhcp_profile_name*: Specifies the DHCP profile name.
- **ipv6**: Enters IPv6 configuration mode.
- **server** : Specifies the IPv6 server details.

- **iana-pool-name** *ipam_pool_name*: Specifies the Internet Assigned Numbers Authority (IANA) pool name.
- **iapd-pool-name** *ipam_pool_name*: Specifies the Identity Association for Prefix Delegation (IAPD) pool name.
- **lease { days** *value* **| hours** *value* **| minutes** *value* **}**: Specifies the lease time duration in the number of days, hours, and minutes. The number of lease days supported is from 0 to 365. The number of leave hours supported ranges from 0 to 23 and minutes from 0 to 59.

Creating PPPoE Profile

Use the following commands to create a PPPoE profile.

```
config
  profile pppoe pppoe_profile_name
  mtu mtu
```

NOTES:

- **profile pppoe** *pppoe_profile_name*: Specifies the PPPoE profile name.
- **mtu** *mtu*: Specifies the default PPP maximum transmission unit (MTU) value to use if the Max-Payload tag is not provided. The valid values range from 500 to 2000. The default value is 1492.

Creating the PPP Feature Template

Use the following commands to create a PPP feature template.

Note The PPP feature template allows per subscriber PPP parameters.

```
config
  profile feature-template feature_template_name
  ppp
    ipcp peer-address-pool ipam_pool_name
    ipcp renegotiation ignore
   exit
```

NOTES:

- **profile feature-template** *feature_template_name*: Specifies the profile feature template name.
- **ppp**: Enters the PPP Configuration mode to configure the PPP feature.
- **ipcp peer-address-pool** *ipam_pool_name*: Specifies the address pool to use to obtain an IPv4 address for the peer.
- **ipcp renegotiation ignore**: Specifies to ignore the attempts of the peer to renegotiate IPCP. The entire PPPoE session is terminated on renegotiation.

Configuring Dynamic Routing using BGP

This section describes how to configure dynamic routing using BGP.

Configuring AS and BGP Router IP Address

To configure the AS and IP address for the BGP router, use the following commands:

```
config
   router bgp local_as_number
   exit
exit
```

NOTES:

- **router bgp** *local_as_number*: Specifies the identification number for the local Autonomous Systems (AS).

  In a GR deployment, you need to configure two Autonomous Systems.

  - One AS for leaf and spine.
  - Second AS for both racks: Site-1 and Site-2

Configuring BGP Service Listening IP Address

To configure the BGP service listening IP address, use the following commands:

```
config
   router bgp local_as_number
      interface interface_name
   exit
exit
```

NOTES:

- **interface** *interface_name*: Specifies the name of the interface.

Configuring BGP Neighbors

To configure the BGP neighbors, use the following commands:

```
config
   router bgp local_as_number
      interface interface_name
      neighbor neighbor_ip_address remote-as as_number
   exit
exit
```

NOTES:

- **neighbor** *neighbor_ip_address*: Specifies the IPv4/IPv6 address of the neighbor BGP router.
- **remote-as** *as_number*: Specifies the identification number for the AS.

Configuring Bonding Interface

To configure the bonding interface related to the interfaces, use the following commands:

```
config
   router bgp local_as_number
      interface interface_name
      bondingInterface interface_name
   exit
exit
```

NOTES:

- **bondingInterface** *interface_name*: Specifies the related bonding interface for an interface. If the bonding interface is active, then the BGP gives a higher preference to the interface-service by providing a lower MED value.

Configuring Learn Default Route

If you want to configure specific routes on your system and need to support all routes, then set the **learnDefaultRoute** value as **true**.

Note This configuration is optional.

To configure the Learn Default Route, use the following commands:

```
config
   router bgp local_as_number
      learnDefaultRoute true/false
   exit
exit
```

NOTES:

- **learnDefaultRoute** *true/false*: Specifies the option to enable or disable the **learnDefaultRoute** parameter. When set to true, BGP learns default route and adds it in the kernel space. By default, it is false.

Configuring BGP Port

To configure the port number for a BGP service, use the following commands:

```
config
   router bgp local_as_number
      loopbackPort port_number
   exit
exit
```

NOTES:

- **loopbackPort** *port_number*: Specifies the port number for the BGP service. The default value is 179.

Policy Addition

The BGP speaker pods learns many route information from its neighbors. However, only a few of them are used for supporting the outgoing traffic. This is required for egress traffic handling only. Routes are filtered by configuring import policies on the BGP speakers and is used to send learned routes to the protocol pods.

A sample CLI code for policy addition and the corresponding descriptions for the parameters are shown below.

```
$bgp policy <policy_Name> ip-prefix 209.165.200.225/16 mask-range 21..24 as-path-set "^65100"
 interface bd2n.10 gateway 209.165.201.30
```

Table 115: Import Policies Parameters

Element	Description	Example	Optional
as-path-set	AS path value	"^65100"	Yes
ip-prefix	Prefix value in format { IPv4_address/prefix length | IPv6_address/prefix length }	"209.165.200.225/16" or "2001:DB8::1/32"	Yes
mask-range	IPv4 or IPv6 Mask range { 0..32 | 0..128 }	"21..24"	Yes
interface	Interface to set as source IP (default is VM IP)	eth0	Yes
gateWay	IPv4/IPv6 Gateway address in either format { IPv4_address | IPv6_address } based on the type of ip-prefix value given	209.165.201.30	Yes
modifySourceIp	Modify source IP of incoming route Default value is False.	true	Yes
isStaticRoute	Flag to add static IP address into kernel route Default value is False.	true	Yes

AS-Path Prepending for BGP VIP Routes

Table 116: Feature History

Feature Name	Release Information	Description
AS-Path Prepending for BGP VIP Routes	2024.02.0	This feature allows the cnBNG to prepend the AS-path attribute to BGP Virtual IP (VIP) routes when advertising to neighboring routers. By manipulating the AS-path length, cnBNG influences the route preference on the border leaf routers, which programs the BGP routes into the network. With this feature, you can ensure that the correct routing path is selected in a multi-VRF or multi-AS deployment scenario.

AS-Path Prepending for BGP VIP Routes feature enhances the CP-GR functionality in environments where multiple cnBNG clusters are configured across different Virtual Routing and Forwarding instances (VRFs) or Autonomous Systems (ASes).

cnBNG prepends its own AS number to the AS-path of BGP VIP routes before advertising them to BGP neighbors. This action effectively increases the AS-path length, making these routes less preferable compared to other routes with shorter AS-paths under normal BGP path selection criteria. The border leaf routers use this AS-path length information to determine the best route to the VIP, ensuring that traffic is routed through the appropriate cnBNG cluster.

Example Configuration

The following is a sample configuration to enable prepending AS-path attribute for BGP routes.

```
router bgp 65000
   prepend as-path true
```

Configuring BGP Speaker

This configuration controls the number of BGP speaker pods in deployment. BGP speaker advertises service IP information for incoming traffic from both the sites.

Note

- Use non-bonded interface in BGP speaker pods for BGP peering.
- BGP peering per Proto node is supported with only two BGP routers/leafs. Considering two Proto nodes, there can be a maximum of four BGP neighborships.
- In a GR setup for AIO CP cluster, the replica count for BGP speaker pods must always be 1.

```
config
  instance instance-id instance_id
   endpoint bgpspeaker
     replicas replicas_count
     nodes node_count
   exit
```

The following is a sample configuration:

```
config
  instance instance-id 1
    endpoint bgpspeaker
     replicas 1
     nodes    2
    exit
```

Configuring BFD

Bidirectional Forwarding Detection (BFD) protocol is used for Faster Network Failure Detection along with BGP. Whenever connectivity between BGP peering fails with cluster (NF), failover is triggered to minimize traffic failure impact.

```
config
  router bgp as
    bfd interval interval min_rx min_rx multiplier multiplier
    loopbackPort loopbackPort loopbackBFDPort loopbackBFDPort
  interface interface_id (BGP on non-bonded interface <-- loopbackEth)
    bondingInterface bondingInterface (leaf6-nic)
    bondingInterface bondingInterface (leaf6-nic)
    neighbor neighbor_ip_address remote-as remote_as fail-over fail_over_type
  exit
  interface interface_id (BGP on non-bonded interface <-- loopbackEth)
    bondingInterface bondingInterface (leaf7-nic)
    bondingInterface bondingInterface (leaf7-nic)
    neighbor bondingInterface remote-as remote_as fail-over fail_over_type
  exit
  policy-name policy_name
   as-path-set as_path_set
   gateWay gateWay_address
   interface interface_id_source
   ip-prefix ip_prefix_value
   isStaticRoute false | true
   mask-range mask_range
   modifySourceIp false | true
  exit
exit
```

NOTES:

- **bgp** *as*: Specifies the Autonomous System (AS) path set.
- **bfd**: Specifies BFD configuration.
  - **interval** *interval* : Specifies the BFD interval in milliseconds.
  - **min_rx** *min_rx*: Specifies the BFD minimum RX in milliseconds.
  - **multiplier** *multiplier*: Specifies the BFD interval multiplier.
- **interface** *interface_id*: Specifies BGP local interface.
  - **bondingInterface** *bondingInterface*: Specifies the linked bonding interface.

- **neighbor** *neighbor_ip_address*: Specifies the IPv4/IPv6 address of neighbor.
  - **fail-over** *fail_over_type*: Specifies the failover type.
  - **remote-as** *remote_as*: Specifies the Autonomous System (AS) number of BGP neighbor.

- **learnDefaultRoute**: Learns default route and adds it in kernel space
- **loopbackBFDPort** *loopbackBFDPort*: Specifies the BFD local port.
- **loopbackPort** *loopbackPort*: Specifies the BGP local port.
- **policy-name** *policy_name*: Specifies the policy name.
  - **as-path-set** *as_path_set*: Specifies the Autonomous System (AS) path set.
  - **gateWay** *gateWay_address*: Specifies the gateway address.
  - **interface** *interface_id_source*: Specifies the interface to set as source IP.
  - **ip-prefix** *ip_prefix_value*: Specifies the IP prefix value.
  - **isStaticRoute** *false | true*: Specifies whether to add static route in kernel space. Default value is false.
  - **mask-range** *mask_range*: Specifies the mask range.
  - **modifySourceIp** *false | true*: Modifies the source IP of the incoming route. Default value is false.

    true: This option is used for non-UDP related VIPs. Source IP of the given interface is used as Source IP while sending out packets from .

    false: This option is used for all UDP related VIPs. VIP is used as Source IP while sending out packets from .

The following is a sample configuration:

```
router bgp 65142
 learnDefaultRoute false
 bfd interval 250000 min_rx 250000 multiplier 3
 interface enp94s0f0.3921
  bondingInterface enp216s0f0
  bondingInterface enp94s0f0
  neighbor 209.165.201.24 remote-as 65141 fail-over bfd
 exit
 interface enp94s0f1.3922
  bondingInterface enp216s0f1
  bondingInterface enp94s0f1
  neighbor 2001::250 remote-as 65141 fail-over bfd
```

Configuring POD Monitoring

To configure POD monitoring and failover thresholds in the GR setup, use the following configuration. The GR pod monitors the configured POD name.

```
config
 geomonitor
  podmonitor pods pod_name
```

```
   retryCount value
   retryInterval interval_value
   retryFailOverInterval failover_interval
   failedReplicaPercent percent_value
  exit
 exit
```

NOTES:

- **pods** *pod_name*: Specifies the name of the pod to be monitored. For example, Cache-pod, res-ep, and so on.
- **retryCount** *value*: Specifies the retry counter value to retry if the pod fails to ping. After that the pod is marked as down. Must be an integer in the range of 1-10.
- **retryInterval** *interval_value*: Specifies the retry interval in milliseconds if the pod successfully pings. Must be an integer in the range of 200-10000.
- **retryFailOverInterval** *failover_interval*: Specifies the retry interval in milliseconds if the pod fails to ping. Must be an integer in the range of 200-10000.
- **failedReplicaPercent** *percent_value*: Specifies the percent value of failed replica after which GR failover is triggered. Must be an integer in the range of 10-100.

The following is a sample configuration.

```
geomonitor podmonitor pods cache-pod
 retryCount 3
 retryInterval 5
 retryFailOverInterval 1
 failedReplicaPercent 40
exit
```

Configuring CDL Instance Awareness and Replication

In Common Data Layer (CDL), along with existing GR related parameters, GR instance awareness must be enabled using a feature flag on all sites. Also, the mapping of system-id to slice names should also be provided for this feature to work on all sites.

The CDL is also equipped with Geo Replication (GR) failover notifications, which can notify the timer expiry of session data and bulk notifications to the currently active site. The CDL uses Border Gateway Protocol (BGP) through App-Infra for the GR failover notifications.

The CDL subscribes to the key value on both the GR sites. The App-Infra sends notifications to the CDL when there is any change in these key values. A key value indicates the state of the CDL System ID or the GR instance. The GR instance is mapped to the CDL slices using the CDL system ID or the GR instance ID in the key.

The system ID is mandatory on both the sites. The GR instance ID in the NF configuration must match the CDL system ID.

CDL has instance-specific data slices. It also allows users to configure instance-specific slice information at the time of bringing up.

- CDL notifies the data on expiry or upon bulk notification request from the active slices.
- CDL determines the active instance based on the notification from app-infra memory-cache.

- CDL slice is a partition within a CDL instance to store a different kind of data. In this case, NF stores a different instance of data.



Note CDL slice name should match with the slice-name configured in GR.

Configuring CDL Instance Awareness

The following command is used to configure CDL instance awareness.

```
config
cdl
 datastore datastore_session_name
   features
    instance-aware-notification
      enable [ true | false ]
      system-id system_id
       slice-names slice_names
   end
```

NOTES:

- **datastore** *datastore_session_name*: Specifies the datastore name.
- **enable [true | false]**: Enables the GR instance state check for slices.
- **system-id** *system_id*: Maps the system ID to slice name.
- **slice-names** *slice_names*: Specifies the list of slice names associated with the system ID. CDL slice name should match with the slice-name configured in GR.

The following is a sample configuration:

```
cdl datastore session
 features instance-aware-notification enable true
 features instance-aware-notification system-id 1
  slice-names [ sgw1 smf1 ]
 exit
 features instance-aware-notification system-id 2
  slice-names [ sgw2 smf2 ]
 end
```

Configuring CDL Replication

This section describes the CDL replication configuration.

1. Configure Site-1 CDL HA system without any Geo-HA-related configuration parameters.
   a. Set the System ID as 1 in the configuration.
   b. Set the slot map/replica and index map/replica and Kafka replica as per the requirements.

   The following is a sample configuration:

   ```
   cdl system-id 1
   cdl node-type session
   ```

```
cdl datastore session
endpoint replica replica_id
  slot map 4
  slot replica 2
  index map 1
  index replica 2
cdl kafka replica 2
```

1. Configure external IPs on Site-1 for Site-2 to Site-1 communication.
   a. Enable geo-replication on Site-1 and configure the remote Site as 2 for Site-1.

      cdl enable-geo-replication *true*
   b. Configure the external IP for CDL endpoint to be accessed by Site-2.

      cdl datastore session endpoint external-ip *site-1_external_ip*
   c. Configure the external IP and port for all Kafka replicas.

      So, if two replicas (default) are configured for Kafka, user need to provide two different <*ip*>+<*port*> pairs.

      cdl kafka external-ip *site-1_external_ip port1* **cdl kafka external-ip** *site-1_external_ip port2*

2. Add remote site (Site-1) information on Site-2.
   - Remote site cdl-ep configuration on Site-2:

     cdl remote-site 1 db-endpoint host *site-1_cdl_ep_ip*

     cdl remote-site 1 db-endpoint port *site-1_cdl_ep_port*

     (Port Example: 8882)
   - Remote site Kafka configuration on Site-2:

     ```
     cdl remote-site 1 kafka-server site-1_kafka1_ip site-1_kafka1_port
     ```

     ```
     cdl remote-site 1 kafka-server site-1_kafka2_ip site-1_kafka2_port
     ```
   - Direct the session datastore configuration to remote Site-2 configuration:

     ```
     cdl datastore session geo-remote-site 1
     ```
   - (Optional) Configure the SSL certificates to establish a secure connection with remote site on Site-1. All the certificates are in multi-line raw text format. If the certificates are not valid, the server continues with non-secure connection.

     cdl ssl-config certs *site-2_external_ip* **ssl-key** *<ssl_key>*

     cdl ssl-config certs *site-2_external_ip* **ssl-crt** *<ssl_crt>*

3. Commit GR configuration on Site-2:
   - Commit the configuration and let the pods be deployed on Site-2.
   - Verify all pods are in running state.
   - Once both sites are deployed, verify that the mirror maker pods on both sites are running and in ready state.

Examples

HA:

```
cdl node-type db-ims

cdl datastore session
 endpoint replica 2
 index map    1
 index write-factor 1
 slot replica 2
 slot map     4
exit

k8 label cdl-layer key smi.cisco.com/node-type value oam
```

Site-1:

```
cdl system-id          1
cdl node-type          session
cdl enable-geo-replication true

cdl remote-site 2
 db-endpoint host 209.165.201.21 >> Site-2 external CDL IP
 db-endpoint port 8882
 kafka-server 209.165.201.21 10092 >> Site-2 external CDL IP
 exit
exit

cdl label-config session
 endpoint key smi.cisco.com/node-type1
 endpoint value cdl-node
 slot map 1
  key   smi.cisco.com/node-type1
  value cdl-node
 exit
 index map 1
  key   smi.cisco.com/node-type1
  value cdl-node
 exit
exit
cdl logging default-log-level debug

cdl datastore session
 label-config    session
 geo-remote-site [ 2 ]
 slice-names     [ 1 2 ]
 endpoint cpu-request 100
 endpoint replica 2
 endpoint external-ip 209.165.201.25 >> Site-1 external CDL IP
 endpoint external-port 8882
 index cpu-request 100
 index replica 2
 index map   1
 slot cpu-request 100
 slot replica 2
 slot map    1
exit

cdl kafka replica 1
cdl kafka label-config key smi.cisco.com/node-type1
cdl kafka label-config value cdl-node
cdl kafka external-ip 209.165.201.25 10092 >> Site-1 external CDL IP
```

Site-2:

```
cdl system-id          2
cdl node-type          session
cdl enable-geo-replication true

cdl remote-site 1
 db-endpoint host 209.165.201.25 >> Site-1 external CDL IP
 db-endpoint port 8882
 kafka-server 209.165.201.25 10092 >> Site-1 external CDL IP
 exit
exit

cdl label-config session
 endpoint key smi.cisco.com/node-type12
 endpoint value cdl-node
 slot map 1
  key   smi.cisco.com/node-type12
  value cdl-node
 exit
 index map 1
  key   smi.cisco.com/node-type12
  value cdl-node
 exit
exit

cdl datastore session
 label-config    session
 geo-remote-site [ 1 ]
 slice-names     [ 1 2 ]
 endpoint cpu-request 100
 endpoint replica 2
 endpoint external-ip 209.165.201.21 >> Site-2 external CDL IP
 endpoint external-port 8882
 index cpu-request 100
 index replica 2
 index map   1
 slot cpu-request 100
 slot replica 2
 slot map    1
exit

cdl kafka replica 1
cdl kafka label-config key smi.cisco.com/node-type12
cdl kafka label-config value cdl-node
cdl kafka external-ip 209.165.201.21 10092 >> Site-2 external CDL IP
```

CP-GR for AIO - Configuration Example

This is a sample configuration to bring up CP-GR for AIO cluster.

```
instance instance-id 1
 endpoint geo
  replicas 1
  interface geo-internal
   vip-ip 3.3.174.3 vip-port 7001
   vip-ipv6 2002:4888:3:3::174:3 vip-ipv6-port 7001
  exit
  interface geo-external
   vip-ip 3.3.174.4 vip-port 7002
   vip-ipv6 2002:4888:3:3::174:4 vip-ipv6-port 7002
  exit
 exit
 endpoint bgpspeaker
```

```
  replicas 1
 exit
 endpoint sm
  replicas 2
 exit
 endpoint nodemgr
  replicas 2
 exit
 endpoint n4-protocol
  replicas 2
  retransmission max-retry 1
 exit
 endpoint dhcp
  replicas 2
 exit
 endpoint radius
  replicas 2
  interface coa-nas
   sla response 140000
   vip-ip 3.4.100.1 vip-port 3799 vip-interface bd2.n4.172
   vip-ipv6 2002:4888:3:4::100:1 vip-ipv6-port 3799 vip-interface bd2.n4.172
  exit
 exit
 endpoint udp-proxy
  replicas 1
  vip-ip 3.4.100.1 vip-interface bd2.n4.172
  vip-ipv6 2002:4888:3:4::100:1 vip-interface bd2.n4.172
  interface n4
   sla response 140000
  exit
  interface gtpu
   sla response 180000
  exit
 exit
exit
instance instance-id 2
 endpoint geo
  replicas 1
  interface geo-internal
   vip-ip 4.4.184.3 vip-port 7001
   vip-ipv6 2002:4888:4:4::184:3 vip-ipv6-port 7001
  exit
  interface geo-external
   vip-ip 4.4.184.4 vip-port 7002
   vip-ipv6 2002:4888:4:4::184:4 vip-ipv6-port 7002
  exit
 exit
 endpoint bgpspeaker
  replicas 1
 exit
 endpoint sm
  replicas 2
 exit
 endpoint nodemgr
  replicas 2
 exit
 endpoint n4-protocol
  replicas 2
  retransmission max-retry 1
 exit
 endpoint dhcp
  replicas 2
 exit
 endpoint radius
```

```
  replicas 2
  interface coa-nas
   sla response 140000
   vip-ip 4.3.100.1 vip-port 3799 vip-interface bd2.n4.172
   vip-ipv6 2002:4888:4:3::100:1 vip-ipv6-port 3799 vip-interface bd2.n4.172
  exit
 exit
 endpoint udp-proxy
  replicas 1
  vip-ip 4.3.100.1 vip-interface bd2.n4.172
  vip-ipv6 2002:4888:4:3::100:1 vip-interface bd2.n4.172
  interface n4
   sla response 140000
  exit
  interface gtpu
   sla response 180000
  exit
 exit
exit
cdl system-id          1
cdl node-type          session
cdl enable-geo-replication true
cdl zookeeper replica 3
cdl remote-site 2
 db-endpoint host 4.4.185.3
 db-endpoint port 8882
 kafka-server 4.4.185.4 10001
 exit
exit
cdl label-config session
 endpoint key smi.cisco.com/sess-type
 endpoint value cdl-node
 slot map 1
  key   smi.cisco.com/sess-type
  value cdl-node
 exit
 index map 1
  key   smi.cisco.com/sess-type
  value cdl-node
 exit
exit
cdl logging default-log-level error
cdl datastore session
 label-config    session
 geo-remote-site [ 2 ]
 slice-names     [ aio1 aio2 ]
 overload-protection disable true
 endpoint go-max-procs 16
 endpoint replica 2
 endpoint copies-per-node 2
 endpoint settings slot-timeout-ms 750
 endpoint external-ip 3.3.175.3
 endpoint external-port 8882
 index go-max-procs 8
 index replica 2
 index map    1
 index write-factor 1
 features instance-aware-notification enable true
 features instance-aware-notification system-id 1
  slice-names [ aio1 ]
 exit
 features instance-aware-notification system-id 2
  slice-names [ aio2 ]
 exit
```

```
 slot go-max-procs 8
 slot replica 2
 slot map     1
 slot write-factor 1
 slot notification limit           1500
 slot notification max-concurrent-bulk-notifications 20
exit
cdl kafka replica 1
cdl kafka label-config key smi.cisco.com/sess-type
cdl kafka label-config value cdl-node
cdl kafka external-ip 3.3.175.4 10001
exit
etcd replicas 3
etcd backup disable false
```

Cluster Maintenance Mode

cnBNG-CP supports the maintenance mode flag to disable the impact on a cluster if the cluster in GR setup is scheduled for maintenance. This is useful so that the standby cluster executes its responsibility and other activities on the targeted cluster without any issue.

Use the **Geo maintenance mode { true | false }** CLI command to enable or disable the maintenance mode in a cluster.

When the **Geo maintenance mode** value is set to **true**,

- All monitoring activities are paused
- The standby cluster can't trigger failover in any case
- Only CLI-based failover is allowed from the cluster where the maintanence mode is enabled.
- Replication activities continue on the cluster.
- Maintenance mode doesn't change instance roles of the site implicitly. However, role change is possible using `geo switch-role role` CLI command.

Whenever there is a change in the maintenance mode flag value:

- The instance role of the cluster is unchanged
- The standby site is notified of the new flag value, so that the standby site refrains from sending any messages. It also stops remote cluster monitoring.

Note Both the clusters can be in maintenance mode at the same time. You can push the system into maintenance mode even if the standby cluster is already under maintenance mode.

Viewing the Maintenance Mode Status

To check the maintenance mode status, use the **show geo-maintenance-mode** command.

Manual CLI Switchover

The following section provides information on manual CLI based switchover commands.

Geo Switch Role

To switch GR role (for example, role Primary to Standby), use the following command.

geo switch-role { role *role* | **instance-id** *gr_instanceId* }

NOTES:

- **role** *role*: Specifies new role for the given site.

  Role can be primary or standby.

- **instance-id** *gr_instanceId*: Specifies the GR Instance ID

`geo switch-role` command triggers manual failover from one site to another site for specific instance ID. The site which triggers the failover is moved from PRIMARY role to STANDBY_ERROR role. In between, the site which triggers failover, sends a failover (trigger GR) message to another site. The other site which receives the failover message is moved from STANDBY role to PRIMARY role.

Geo Reset Role

To reset the GR instance role (for example, role from STANDBY_ERROR to STANDBY), use the following command:

geo reset-role { role *role* **instance-id** *gr_instanceId* }

NOTES:

- **role** *role*: Specifies new role for the given site.

  Role must be standby.

- **instance-id** *gr_instanceId*: Specifies the GR Instance ID.

`geo reset-role` command triggers change in the role for the given instance on local site. Remote site will not receive any message for the same command. It is only possible to change the role for the given instance ID from STANDBY_ERROR to STANDBY. Another role change is not possible.

Key Performance Indicators (KPIs)

The following section describes KPIs.

Table 117: Monitoring KPIs

KPI Name	Description	Labels	Possible Values
geo_monitoring_total	This KPI displays the total number of successful / failure messages of different kinds such as, heartbeat / remoteNotify / TriggerGR and so on.	ControlAction Type	AdminMonitoring ActionType / AdminRemote MessageAction Type / AdminRole ChangeActionType
		ControlAction NameType	MonitorPod / MonitorBfd / MonitorVip RemoteMsgHeartbeat / RemoteMsgNotify TriggerGRApi / ResetRoleApi
		Admin Node	Any string value. For example, GR Instance ID or instance key / pod name
		Status Code	Error / Success code
		Status Message	Message string
geo_RejectedRoleChanged_total	This KPI displays the total number of control packets coming to Standby GR-instance from the User Plane or RADIUS server.	RejectedCount GRInstanceId	{10, 1} / {20, 2}

Table 118: BGP Routing KPIs

KPI Name	Description	Labels	Possible Values
bgp_peers_total	Total number of peers added	peer_ip	BGP neighbor IP address
		as_path	AS value (in digit format) of BGP peer.

KPI Name	Description	Labels	Possible Values
bgp_failed_peerstotal	Total number of failed peers	peer_ip	BGP neighbor IP address
		as_path	AS value (in digit format) of BGP peer.
		error	Error message
bgp_incoming_routerequest_total	Total number of incoming routes	interface	Interface name of incoming route
		next_hop	Gateway IP address (next hop address).
		service_IP	Service IP to publish
bgp_incoming_failedrouterequest_total	Total number of failed incoming routes	peer_ip	BGP neighbor IP address
		as_path	AS value (in digit format) of BGP peer.
		service_IP	Service IP to publish
bgp_outgoing_routerequest_total	Total number of outgoing routes	local_pref	BGP neighbor IP address
		med	AS value (in digit format) of BGP peer.
		next_hop	Gateway IP address (next hop address).
		service_IP	Service IP to publish
bgp_outgoing_failedrouterequest_total	Total number of failed outgoing routes	local_pref	BGP neighbor IP address
		med	AS value (in digit format) of BGP peer.
		next_hop	Gateway IP address (next hop address).
		service_IP	Service IP to publish
bgp_speaker_bfd_status	BFD status	status	BFD_STATUS

Monitoring and Troubleshooting

This section provides information about the CLI commands available to monitor and troubleshoot the feature.

You can use the following monitor, show, and clear commands:

- monitor protocol interface pfcp instance-id <instance_id>
- show subscriber session count instance-id <instance_id>
- show subscriber dhcp count instance-id <instance_id>
- show subscriber pppoe count instance-id <instance_id>
- show subscriber pppoe detail instance-id <instance_id>
- show subscriber redundancy detail instance-id <instance_id>
- show role instance-id <instance_id>
- show role-additional-info instance-id <instance_id>
- clear subscriber sessmgr [srg-peer-id <srg_peer_id> | upf <upf_name> | instance-id <instance_id>]
- clear subscriber dhcp [srg-peer-id <srg_peer_id> | upf <upf_name> | instance-id <instance_id>]
- clear subscriber pppoe [srg-peer-id <srg_peer_id> | upf <upf_name> | instance-id <instance_id>]



Note

- All monitor and show commands must include an instance ID.
- The monitor and clear commands work only for instances whose role is PRIMARY.

From release 2024.02 onwards, we have enhanced the Show CLI commands with an address family specific filtering capability. With this feature you can filter output based on the address family type, either IPv4 or IPv6, enabling you to view the configurations and status of network elements that are specific to an address type.

"show bgp" Command Outputs in an AIO CP-GR Setup

In an AIO CP-GR setup, all **show bgp** command outputs display data only from **bgpspeaker-pod-0**.

show bgp kernel route

To view BGP kernel configured routes, use the following command:

show bgp-kernel-route [ipv4 | ipv6]

Example

The following is a sample configuration:

```
show bgp-kernel-route ipv4

-----bgpspeaker-pod-1  ----
```

```
 DestinationIP  SourceIP        Gateway

 209.165.202.133    209.165.202.148   209.165.202.142

-----bgpspeaker-pod-2 ----

 DestinationIP      SourceIP          Gateway

 209.165.202.134    209.165.202.148   209.165.202.142
```

show bgp global

To view BGP global configuration, use the following command:

show bgp-global [ipv4 | ipv6]

Example

The following is a sample configuration:

```
show bgp-global ipv4
global-details
-----bgpspeaker-pod-1  ----
AS:        65000
Router-ID: 209.165.202.149
Listening Port: 179, Addresses: 209.165.202.149
-----bgpspeaker-pod-2  ----
AS:        65000
Router-ID: 209.165.202.148
Listening Port: 179, Addresses: 209.165.202.148
```

show bgp neighbors

To view BGP neighbors status, use the following command:

show bgp-neighbors [ipv4 | ipv6]
show bgp-neighbors ip *ipv4_address | ipv6_address*

Example

The following is a list of few configuration examples:

```
show bgp-neighbors ipv4
-----bgpspeaker-pod-2  ----
Peer             AS  Up/Down State       |#Received  Accepted
209.165.202.142 60000 00:25:06 Establ      |        3         3
-----bgpspeaker-pod-1  ----
Peer             AS Up/Down State       |#Received  Accepted
209.165.202.142 60000   never Idle        |        0         0
```

```
show bgp-neighbors ip 209.165.202.142
-----bgpspeaker-pod-1  ----
BGP neighbor is 209.165.202.142, remote AS 60000
  BGP version 4, remote router ID unknown
  BGP state = ACTIVE
  BGP OutQ = 0, Flops = 0
  Hold time is 0, keepalive interval is 0 seconds
  Configured hold time is 90, keepalive interval is 30 seconds
```

```
  Neighbor capabilities:
    multiprotocol:
        ipv4-unicast:   advertised and received
    route-refresh:      advertised and received
    extended-nexthop:   advertised
        Local:  nlri: ipv4-unicast, nexthop: ipv6
    4-octet-as: advertised and received
  Message statistics:
                         Sent       Rcvd
    Opens:                  1          1
    Notifications:          0          0
    Updates:                1          2
    Keepalives:            70         70
    Route Refresh:          0          0
    Discarded:              0          0
    Total:                 72         73
  Route statistics:
    Advertised:             0
    Received:              10
    Accepted:              10

-----bgpspeaker-pod-2  ----
BGP neighbor is 209.165.202.142, remote AS 60000
  BGP version 4, remote router ID 209.165.202.136
  BGP state = ESTABLISHED, up for 00:25:20
  BGP OutQ = 0, Flops = 0
  Hold time is 90, keepalive interval is 30 seconds
  Configured hold time is 90, keepalive interval is 30 seconds

  Neighbor capabilities:
    multiprotocol:
        ipv4-unicast:   advertised and received
    route-refresh:      advertised and received
    extended-nexthop:   advertised
        Local:  nlri: ipv4-unicast, nexthop: ipv6
    4-octet-as: advertised and received
  Message statistics:
                         Sent       Rcvd
    Opens:                  1          1
    Notifications:          0          0
    Updates:                1          1
    Keepalives:            51         51
    Route Refresh:          0          0
    Discarded:              0          0
    Total:                 53         53
  Route statistics:
    Advertised:             0
    Received:               3
    Accepted:               3
```

show bgp route summary

To view BGP route summary, use the following command:

show bgp-route-summary [ipv4 | ipv6]

Example

The following is a sample configuration.

```
show bgp-route-summary ipv4
route-details
```

```
-----bgpspeaker-pod-1  ----
Table afi:AFI_IP safi:SAFI_UNICAST
Destination: 5, Path: 5
-----bgpspeaker-pod-2 ----
Table afi:AFI_IP safi:SAFI_UNICAST
Destination: 2, Path: 2
```

show bgp routes

To view BGP routes information, use the following command:

show bgp-routes [ipv4 | ipv6]

Example

The following is a sample configuration:

```
show bgp-routes ipv4
bgp-route

-----bgpspeaker-pod-1  ----
   Network              Next Hop             AS_PATH      Age        Attrs
*> 209.165.202.133/24   209.165.202.142        60000      00:25:55   [{Origin: i} {Med: 0}]
*> 209.165.200.225/32   209.165.202.148                  00:26:00   [{Origin: e} {LocalPref:
 100} {Med: 600}]
*> 209.165.202.134/24   209.165.202.142        60000      00:25:55   [{Origin: i} {Med: 0}]
*> 209.165.202.140/24   209.165.202.142        60000      00:25:55   [{Origin: i} {Med: 0}]
*> 209.165.202.146/32   209.165.202.148                  00:26:00   [{Origin: e} {LocalPref:
 100} {Med: 600}]

-----bgpspeaker-pod-2  ----
   Network              Next Hop             AS_PATH      Age        Attrs
*> 209.165.200.225/32   209.165.202.149                  00:26:24   [{Origin: e} {LocalPref:
 100} {Med: 600}]
*> 209.165.202.146/32   209.165.202.149                  00:26:24   [{Origin: e} {LocalPref:
 100} {Med: 600}]
```

show bfd neighbor

To view the BFD status of neighbors, use the following command:

show bfd-neighbor [ipv4 | ipv6]

Example

The following is a sample configuration.

```
show bfd-neighbor ipv4
Mon Jan 29 06:34:39.776 UTC+00:00
status-details

-----bgpspeaker-pod-0 ----

OurAddr NeighAddr Vrf State OurInt OurIntState

209.165.202.140 209.165.201.146 UP - -
-----bgpspeaker-pod-1 ----

OurAddr NeighAddr Vrf State OurInt OurIntState
```

```
209.165.202.141 209.165.202.146 UP - -
```

show bgp-learned-routes

To view information about BGP learned routes, use the following command:

show bgp-learned-routes [ipv4 | ipv6]

Example

The following is a sample configuration:

```
show bgp-learned-routes ipv4
bgp-route

-----bgpspeaker-pod-1 ----
   Network              Next Hop            AS_PATH
         Age        Interface           Vrf                  Attrs
*> 209.165.201.22/27   209.165.200.225      63100 65100
        03:06:15   enp216s0f0.2119      Default              [{Origin: i}]
*> 209.165.201.0/27    209.165.200.225      63100
        03:06:15   enp216s0f0.2119      Default              [{Origin: i}]

-----bgpspeaker-pod-0 ----
   Network              Next Hop            AS_PATH
         Age        Interface           Vrf                  Attrs
*> 209.165.201.22/27    209.165.200.225      63100 65100
         03:06:19   enp216s0f0.2119      Default              [{Origin: i}]
*> 209.165.200.25/27    209.165.200.225      63100
         03:06:19   enp216s0f0.2119      Default              [{Origin: i}]
```

show bgp-advertised-routes

To view BGP advertised routes information, use the following command:

show bgp-advertised-routes [ipv4 | ipv6]

Example

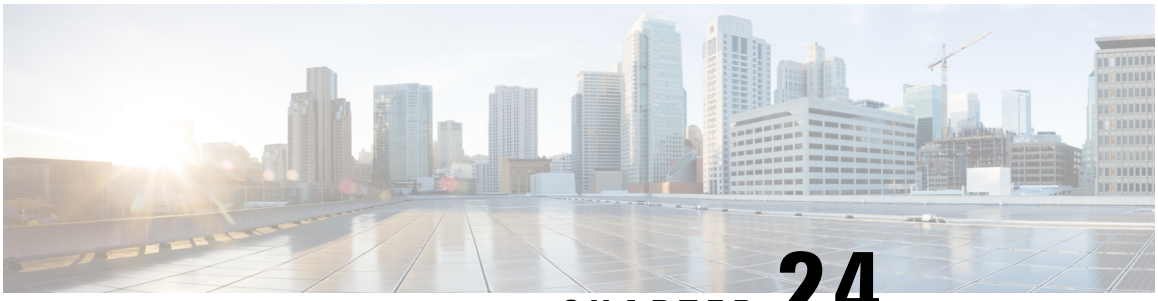
The following is a sample configuration:

```
show bgp-advertised-routes ipv4
bgp-route

-----bgpspeaker-pod-0 ----
   Network              Next Hop            AS_PATH
         Age        Interface           Vrf                  Attrs
*> 209.165.200.25/27   209.165.200.225      63200 63200 63200
        02:39:47   enp216s0f0.2119      Default            [{Origin: e} {LocalPref: 2210}
 {Med: 1220}]
*> 209.165.200.22/27   209.165.200.225      63200 63200 63200
        02:39:47   enp216s0f0.2119      Default            [{Origin: e} {LocalPref: 2210}
 {Med: 1220}]

-----bgpspeaker-pod-1 ----
   Network              Next Hop            AS_PATH
         Age        Interface           Vrf                  Attrs
*> 209.165.200.25/27   209.165.200.224      63200 63200 63200
        02:39:45   enp216s0f0.2119      Default            [{Origin: e} {LocalPref: 2220}
```

```
 {Med: 1210}]
*> 209.165.200.22/27   209.165.200.224       63200 63200 63200
        02:39:45   enp216s0f0.2119      Default             [{Origin: e} {LocalPref: 2220}
 {Med: 1210}]
```



CHAPTER 24

UP Geo Redundancy

- Feature Summary, on page 537
- Revision History, on page 538
- Feature Description, on page 538
- Benefits of UP Geo Redundancy, on page 550
- Supported Features in UP Geo Redundancy, on page 550
- UP Geo Redundancy Configuration Guidelines, on page 551
- Configuring UP Geo Redundancy , on page 551
- Range-based interface list , on page 556
- L3 Routed Subscriber Sessions with Subscriber Redundancy Group, on page 557
- Session Synchronization between UPs, on page 569
- Route Synchronization between CP and UP , on page 572
- Order of Reconciliation, on page 572
- Monitoring Support, on page 572

Feature Summary

Table 119: Feature Summary

Applicable Product(s) or Functional Area	cnBNG
Applicable Platform(s)	SMI
Feature Default Setting	Disabled – Configuration Required
Related Changes in this Release	First Release
Related Documentation	Not Applicable

Revision History

Table 120: Revision History

Revision Details	Release
Introduced IPv4 and dual-stack support for L3 routed subscriber sessions with SRG.	2026.02.0
Introduced support for SRG warm standby mode.	2025.03.0
Introduced support for L3 routed subscriber sessions with SRG.	2025.01.0
Introduced support for SRG PPPoE sessions.	2025.01.0
First introduced.	2022.04.0

Feature Description

Note This feature is Network Services Orchestrator (NSO) integrated.

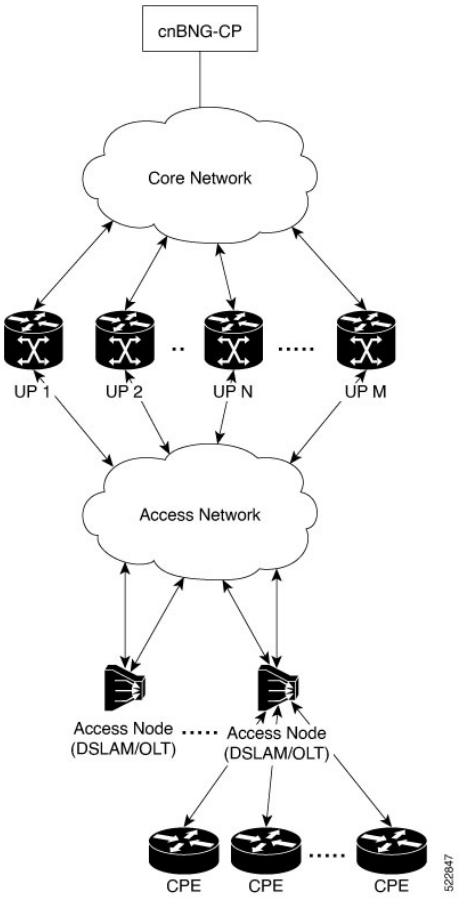
To provide redundancy for the subscriber sessions, cnBNG supports Geographical Redundancy across multiple User Planes (UPs), without having any L1 or L 2 connectivity between them. The UPs may be located in multiple geographical locations, and they have L3 connectivity over a shared core network through IP or MPLS routing.

The UP Geo redundancy feature supports IPoE DHCP-triggered sessions (IPv4, IPv6 and dual-stack), as well as PPPoE subscriber sessions.

UP Geo Redundancy Architecture

The following figure depicts a UP geo redundancy deployment network model:

Figure 35: UP Geo Redundancy Deployment Network Model



The redundancy pairing between UPs work by synchronizing the subscriber state from cnBNG CP to primary (active) and its subordinate (standby).

Geo redundancy works in conjunction with any of the access technologies. The CPEs are agnostic to redundancy; they see only one UP or gateway. The access nodes are dual or multi-homed for redundancy using a variety of technologies based on the service provider network design and choices. Multi-chassis Link Aggregation (MC-LAG), dual-homed (Multiple Spanning Tree - Access Gateway or MST-AG), Ring (MST-AG or G.8032), xSTP and seamless MPLS (pseudowires) are a few such access networks.

For more information on access technologies supported on UP, see the *Broadband Network Gateway Configuration Guide for Cisco ASR 9000 Series Routers* guide.

Subscriber Redundancy Group

Geo redundancy for subscribers is delivered by transferring the relevant session state from primary UP to subordinate UP which can then help in failover (FO) or planned switchover (SO) of sessions from one UP to another. Subscriber Redundancy Group (SRG) which is a set of access-interface (or a single access-interface) is introduced in cnBNG, and all subscribers in an SRG would FO or SO as a group.

The SRG has two modes of operation:

- Hot-standby

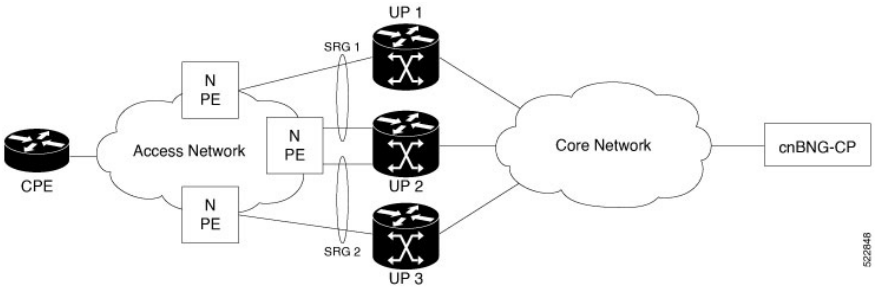
- Warm-standby

Hot-standby mode

Hot-standby is achieved by a 1:1 mirroring of subscriber session state from the primary to the subordinate where the entire provisioning is done before the FO or SO. The sessions provisioned on subordinate is in sync with the set up on the primary. Because the data plane is already set up for sub-second traffic impact, there is minimal action on switchover in the case of hot-standby mode and therefore, it is suitable for subscribers requiring high service level agreement (SLA). With appropriate capacity planning, the sessions can also be distributed across multiple UPs to achieve an M: N model. The primary-subordinate terminology is always in the context of a specific SRG; not for the UP as a whole.

The following figure depicts a typical subscriber redundancy group (SRG):

Figure 36: Subscriber Redundancy Group



SRG Virual MAC

For seamless switchover between two UPs, the L2-connected CPE devices must not detect change in gateway MAC and IPv4 or IPv6 addresses. The access technology like MC-LAG uses the same MAC address on both UPs with active-standby roles, providing seamless switchover. Where MAC sharing is not provided by the access technology or protocol (like MST-AG, G.8032), the cnBNG SRG virtual MAC (vMAC) must be used.

For more information on SRG Virtual MAC, see the *BNG Geo Redundancy* chapter of *Broadband Network Gateway Configuration Guide for Cisco ASR 9000 Series Routers* guide.

SRG warm-standby mode

SRG warm-standby is a subscriber redundancy mode that

- pre-creates and maintains subscriber session context in process memory on standby user-plane nodes,
- enables rapid activation of subscriber services after a switchover event, and
- improves cost effectiveness by allowing a single standby User Plane (UP) node to provide redundancy for multiple user-plane instances.

Table 121: Feature History

Feature Name	Release Information	Description
SRG warm-standby mode	2025.03.0	This feature improves cost efficiency and resource utilization by allowing a single warm standby User Plane (UP) to provide redundancy for multiple UP instances. Instead of needing a separate standby for each active UP, you can use a single standby node to protect several UPs at once.

Session management between active and standby nodes

In Subscriber Redundancy Group (SRG) warm-standby mode, when a subscriber session is established, the control plane ensures that both the active and standby user-plane nodes are aware of the subscriber. However, only the active node programs subscriber sessions into hardware, while the standby node holds the session context in memory.

Switchover process and service restoration

During a switchover, the standby node quickly programs the cached sessions into hardware, reducing service restoration time to a few seconds. This approach is especially beneficial for PPPoE subscriber sessions, aligning with standards such as BBF TR-459 and providing a seamless experience during failover scenarios.

Compatible User Plane version

This feature works with cnBNG User Plane Release 25.4.1 and later versions.



Note SRG warm-standby mode is supported only for Point-to-Point Protocol over Ethernet (PPPoE) subscriber sessions.

Configure SRG warm-standby mode

Follow these steps to enable SRG warm-standby mode:

Procedure

Step 1 Use the **standby-mode warm** command in user-plane configuration mode to enable SRG warm-standby mode:

Example:

```
config
  user-plane instance instance_id
    user-plane user_plane_name
      peer-address ipv4 ipv4_address
      subscriber-profile  profile_name
      subscriber-redundancy
        group group_name
          standby-mode warm
        exit
      exit
```

```
    exit
  exit
 exit
```

NOTES:

- **subscriber-redundancy**: Configures subscriber geo-redundancy. All SRG groups are configured in this mode.
- **group** *group_name*: Specifies the name of the subscriber redundancy group that is unique to a user plane.
- **standby-mode warm**: Enables the SRG warm-standby mode.

This is a sample configuration.

```
user-plane
 instance 1
  user-plane asr9k-1
   peer-address ipv4 1.1.1.1
   subscriber-profile subs-upf
   subscriber-redundancy
    group Group1
     standby-mode warm
    exit
   exit
  exit
 exit
exit
```

Step 2 Use the **show subscriber redundancy srg-peer-id** command to verify the SRG warm-standby configuration.

Example:

```
bng# show subscriber redundancy srg-peer-id Peer1 detail
Tue Mar  18 12:19:14.448 UTC+00:00
subscriber-details
{
  "subResponses": [
    {
      "PeerID": "Peer1",
      "GroupID": "Group1",
      "Domain": "d1",
      "UP List": {
        "asr9k-1": {
          "N4 State": "Connected",
          "Srg State": "Up",
          "Srg Role": "Active",
          "Interface map": {
            "BundleEther1.100": 1,
            "BundleEther1.200": 2
          }
        },
        "asr9k-2": {
          "N4 State": "Connected",
          "Srg State": "Up",
          "Srg Role": "Standby",
          "Interface map": {
            "BundleEther1.100": 1,
            "BundleEther1.200": 2
          },
          "Standby Mode": "Warm"
        }
      },
      "Preferred Active": "asr9k-1"
    }
```

```
    ]
}
```

SRG cold standby mode

SRG cold standby mode is a subscriber redundancy configuration that

- defers subscriber session programming on the standby UPF until an SRG switchover occurs
- minimizes standby UPF resource usage by establishing subscriber state on the standby only after a switchover, and
- results in longer switchover times compared to warm standby, because state must be created after the event rather than beforehand.

Table 122: Feature History Table

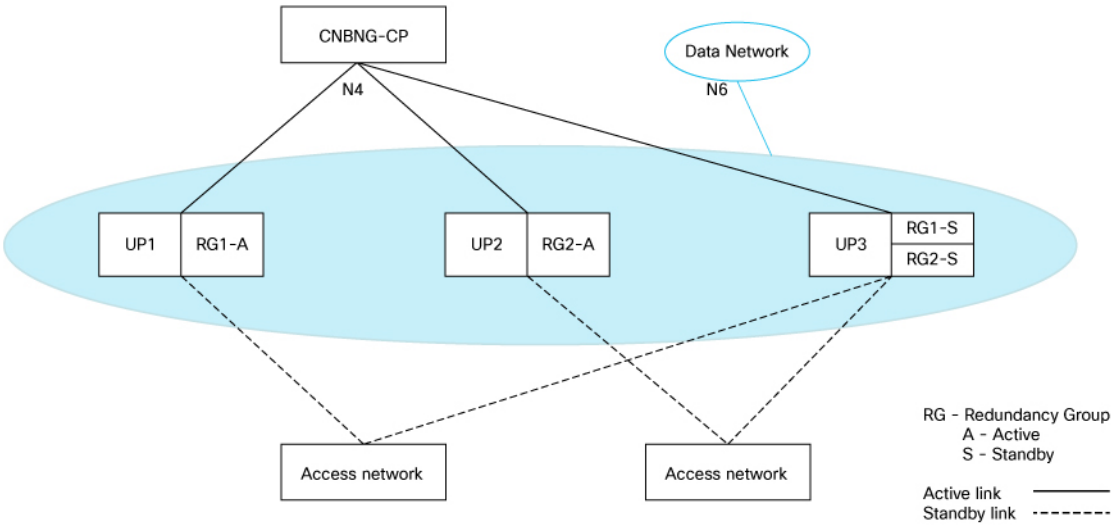
Feature Name	Release Information	Feature Description
SRG cold standby mode	Release 2026.03.0	You can optimize standby resource usage and maintain high availability for subscriber sessions using cold standby mode in Subscriber Redundancy Groups. In this mode, the Control Plane programs subscribers only on the active UPF; programming for the standby UPF is deferred until a switchover occurs. Once the standby group becomes active, the Control Plane automatically programs the subscribers on that UPF.

Recommendations and compatibility

SRG cold standby mode is recommended for scenarios where reduced resource usage on standby nodes is preferable to achieving the fastest possible switchover time. This mode supports L2-Connected IPoE and L3-Routed IPoE sessions, and works with CP-GR and rolling upgrade.

How session programming works in SRG cold standby mode

This figure illustrates how subscriber session programming is handled in SRG cold standby mode.



When a subscriber session is established in SRG cold standby mode, the Control Plane programs the session only on the active UPF. The standby UPF remains unprogrammed while in standby. During a switchover, when the standby SRG group becomes active, the Control Plane starts programming subscribers on the now-active UPF. When the cold UPF transitions back to SRG standby, the Control Plane deletes subscriber sessions from that UPF.

Configure SRG cold standby mode

Configure SRG cold standby mode to provide high availability and session continuity for IPoE subscriber sessions, while minimizing resource consumption on standby UPFs.

Before you begin

- Verify that your cnBNG platform and software version support SRG cold standby mode.
- Identify which SRG groups and session types (such as IPoE) you want to configure for cold standby mode.

Follow these steps to configure SRG cold standby mode for IPoE subscriber sessions:

Procedure

Step 1 Define the SRG group and configure the standby mode to **cold** for the group.

Example:

```
user-plane instance 1
  user-plane asr9k-1
   peer-address ipv4 10.1.1.1
   subscriber-profile subs-upf
   subscriber-redundancy
    group Group1
     standby-mode cold
    exit
   exit
```

```
  exit
 exit
exit
```

This configuration sets the SRG group to cold standby mode, ensuring the standby UPF does not maintain subscriber state until switchover.

Step 2 Use the **show subscriber redundancy srg-peer-id** command to verify that SRG cold standby mode is active.

Example:

```
bng# show subscriber redundancy srg-peer-id Peer1 detail

Tue Mar  18 12:19:14.448 UTC+00:00
subscriber-details
{
  "subResponses": [
    {
      "PeerID": "Peer1",
      "GroupID": "Group1",
      "Domain": "d1",
      "UP List": {
        "asr9k-1": {
          "N4 State": "Connected",
          "Srg State": "Up",
          "Srg Role": "Active",
          "Interface map": {
            "BundleEther1.100": 1,
            "BundleEther1.200": 2
          }
        },
        "asr9k-2": {
          "N4 State": "Connected",
          "Srg State": "Up",
          "Srg Role": "Standby",
          "Interface map": {
            "BundleEther1.100": 1,
            "BundleEther1.200": 2
          },
          "Standby Mode": "Cold"
        }
      },
      "Preferred Active": "asr9k-1"
    }
  ]
}
```

Standby Mode displays **Cold** to indicate that the SRG cold standby mode is active.

Step 3 Use the **show subscriber session count instance-id** command to view the subscriber session count information for a specific instance ID.

Example:

```
bng# show subscriber session count instance-id 1
Mon Mar  9  06:19:25.634 UTC+00:00
subscriber-details
{
  "sessionCount": 100
}
```

Step 4 Use the **show cnbng-nal subscriber all summary** command in the User Plane to view the summary of all subscriber sessions.

Example:

```
UP-asr9k-1# show cnbng-nal subscriber all summary
Mon Mar  9 06:13:42.700 UTC
=====================
Location: 0/RSP0/CPU0
=====================

               Type            PPPoE                   IPoE                    LAC             LNS
               ====            =====                   ====                    ===             ===
                                                 L2-Conn         Routed
                                                 =======         ======

Session Counts by State:
        initializing           0               0               0               0               0
          connecting           0               0               0               0               0
           connected           0               0               0               0               0
           activated           0               100             0               0               0
                idle           0               0               0               0               0
       disconnecting           0               0               0               0               0
               Total:          0               100             0               0               0


Session Counts by Address-Family:
                none           0               0               0               0               0
                ipv4           0               0               0               0               0
                ipv6           0               0               0               0               0
                dual           0               100             0               0               0
                 lac           0               0               0               0               0
              Total:           0               100             0               0               0
```

The sample output from UP-asr9k-1 shows that the sessions are programmed on the active UPF.

Example:

```
UP-asr9k-2# show cnbng-nal subscriber all summary
Mon Mar  9 06:13:42.700 UTC
=====================
Location: 0/RSP0/CPU0
=====================

               Type            PPPoE                   IPoE                    LAC             LNS
               ====            =====                   ====                    ===             ===
                                                 L2-Conn         Routed
                                                 =======         ======

Session Counts by State:
        initializing           0               0               0               0               0
          connecting           0               0               0               0               0
           connected           0               0               0               0               0
           activated           0               0               0               0               0
                idle           0               0               0               0               0
       disconnecting           0               0               0               0               0
               Total:          0               0               0               0               0


Session Counts by Address-Family:
                none           0               0               0               0               0
                ipv4           0               0               0               0               0
                ipv6           0               0               0               0               0
                dual           0               0               0               0               0
                 lac           0               0               0               0               0
              Total:           0               0               0               0               0
```

The sample output from UP-asr9k-2 shows that the sessions are not programmed on the cold standby UPF.

Session Distribution Across SRG

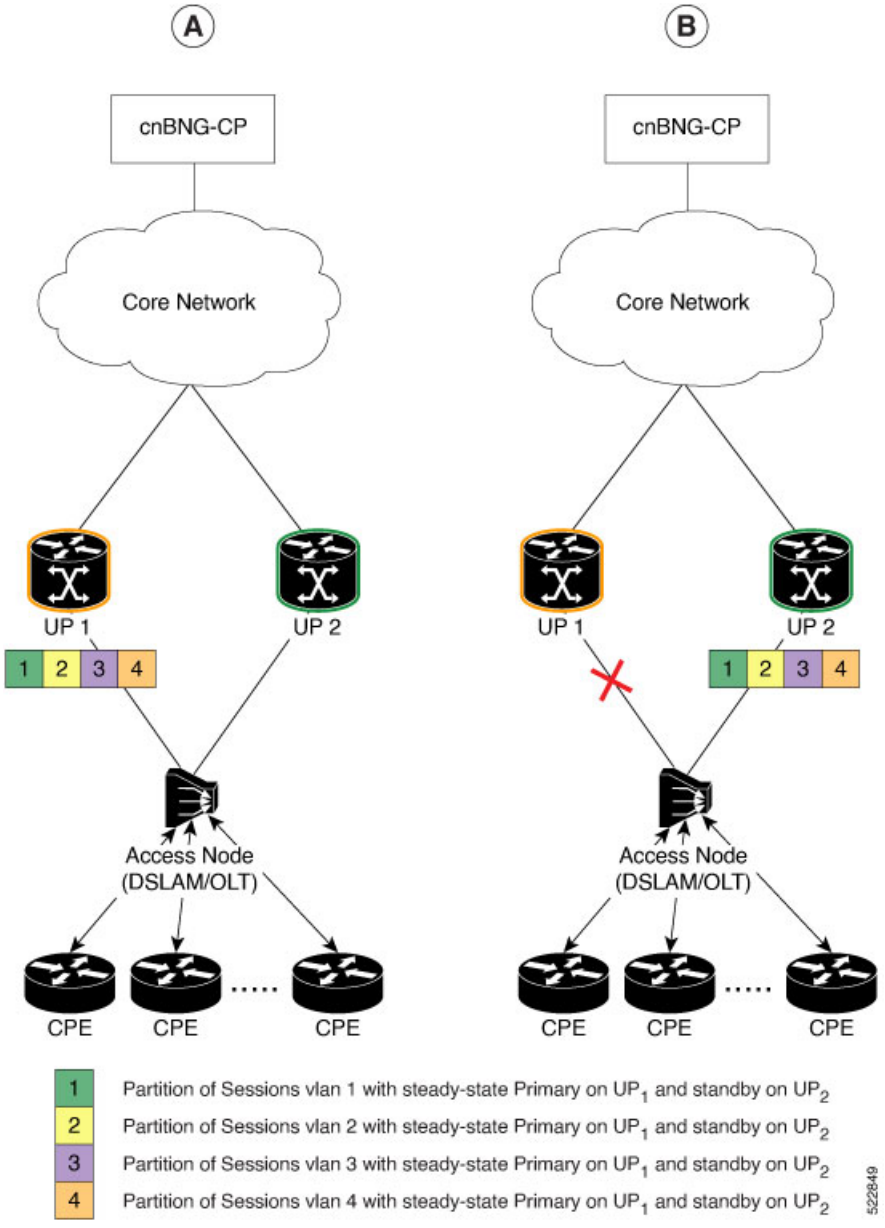
The session distribution across SRGs can be in either of these modes:

- Active-standby mode:

  In this mode, a dedicated standby UP can be a subordinate for multiple SRGs from different active UPs which are primaries for those respective SRGs.

  This figure shows an active-standby mode of session distribution across SRGs:

  Figure 37: Active-standby Mode of Session Distribution



In figure A:

- Sessions are associated with partitions (VLAN 1, 2, 3 and 4) on UP1, with each VLAN mapped to separate SRG configured as primary role.
- UP2 acts as standby for all VLANs.
- Each VLAN has 8K sessions terminated on it.

In figure B:

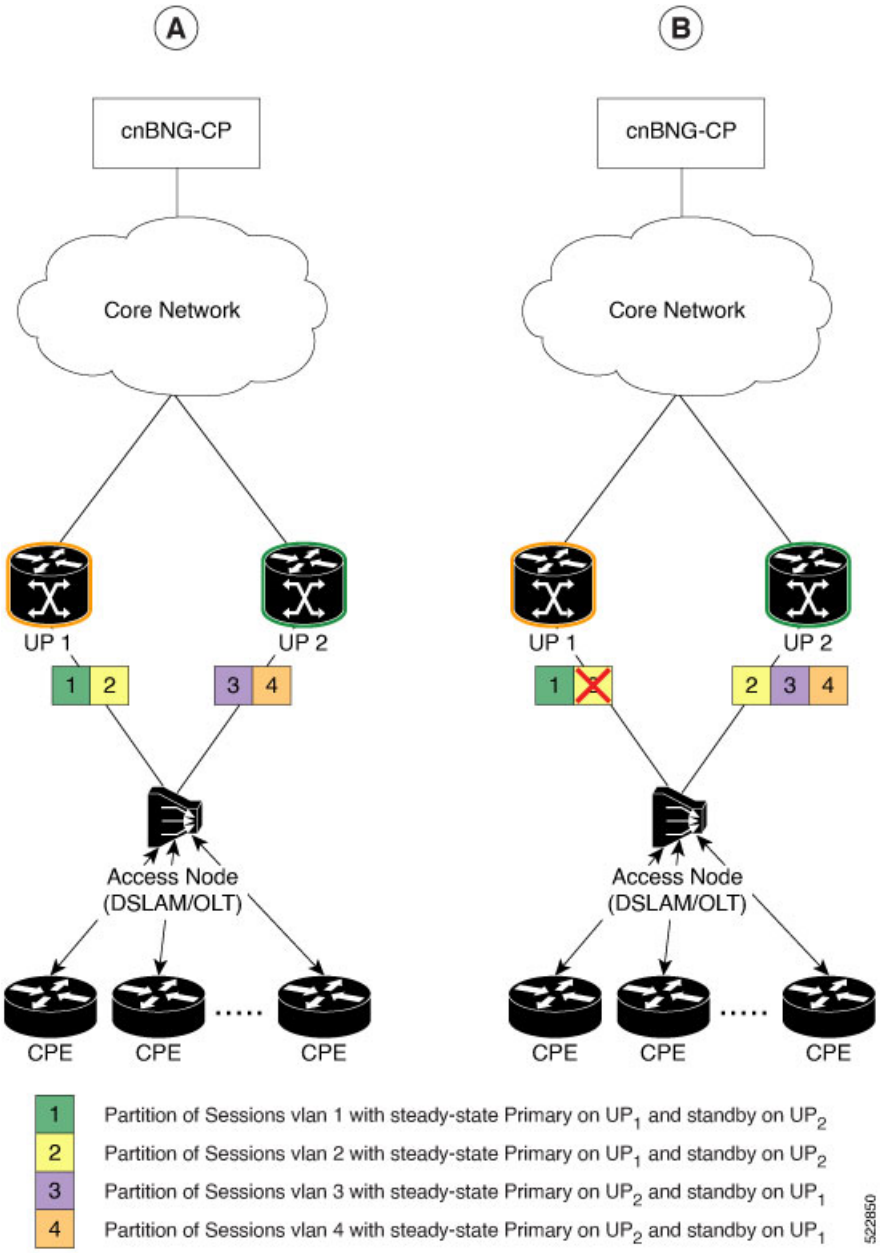
- An interface failure gets detected (using object-tracking of the access-interface).
- SRG for each VLAN on UP2 gets the primary role.
- All 32K sessions are switched to UP2.
- UP2 sees a session termination count of 32K.

- Active-active mode:

  In this mode, an UP can be primary for one SRG and a standby for another SRG at the same time.

  The following figure shows an active-active mode of session distribution across SRGs:

Figure 38: Active-active Mode of Session Distribution



In figure A:

- Sessions are associated with partitions (VLAN 1, 2) on UP1, with each VLAN mapped to separate SRG configured as primary role.
- Sessions are associated with partitions (VLAN 3, 4) on UP2, with each VLAN mapped to separate SRG configured as primary role.
- Each VLAN has 8K sessions terminated on it.
- Each UP has 16K sessions terminated on it.

In figure B:

- The interface associated with VLAN 2 on UP1 goes down.
- Sessions associated with partitions (VLAN 2) on UP1 are switched to UP2.
- UP1 sees a session termination count of 8K and UP2 sees a session termination count of 24K.

Benefits of UP Geo Redundancy

Major benefits of UP Geo Redundancy include:

- Supports various redundancy models such as 1:1 (active-active) and M:N, including M:1.
- Provides flexible redundancy pairing on access-link basis.
- Works with multiple access networks such as MC-LAG, dual-home and OLT rings.
- Supports various types of subscribers such as IPv4, IPv6 and dual-stack IPoE sessions.
- Provides failure protection to access link failures, N4 link failures, LC failures, RP failures and chassis failures.
- Performs automatic switchovers during dynamic failures or planned events such as maintenance, upgrades and transitions.
- Co-exists with other high availability (HA) or redundancy mechanisms.
- Does switchover of the impacted session group only; other session groups remain on the same UP.
- Provides fast convergence and rapid setup of sessions, with minimal subscriber impact during switchover.
- Provides automatic routing convergence towards core and efficient address pool management.
- Provides seamless switchover for subscriber CPE without the need for any signaling.
- Integrates with RADIUS systems.
- Does not impact session scale and call-per-second (CPS) during normal operation.

Supported Features in UP Geo Redundancy

These base geo redundancy features are supported:

- Multiple SRG groups to different peer routers.
- Hot-standby mode for subordinate (that is, subscribers provisioned in hardware on the subordinate as they are synchronized).
- Dynamic role negotiation between peers.
- Manual SRG switchover through command line interface (CLI).
- Dynamic failure detection using object tracking (link up-down, route and IPSLA tracking).
- Revertive timer per SRG group.

- SRG active-active mode without any access protocol.
- G.8032 (dual-home and ring) access technologies.

These DHCP features are supported:

- DHCPv6 IA-NA and IA-PD support for L2 connected sessions.
- DHCPv4 support for L2 connected sessions.
- DHCPv4 or DHCPv6 dual-stack support.
- DHCP server mode.
- Session initiation through DHCPv4 or DHCPv6 protocol.

UP Geo Redundancy Configuration Guidelines

UP Configuration Consistency

- Geo redundancy feature infrastructure synchronizes individual subscriber session state from primary to subordinate. But, it does not synchronize the UP related configurations (namely dynamic-template, DHCP profiles, policy-maps, access-interface configurations, external RADIUS or DHCP server, and so on).
- For successful synchronization and setup of subscriber sessions between the two UPs, it is mandatory that the relevant UP configurations must be identical on the two routers and on the access-interfaces pairs in the SRG.
- While the access-interfaces or their types (or both) may vary between the paired UPs, their outer-VLAN tag (that is, S-VLAN imposed by the access or aggregation devices) must be identical.
- Inconsistencies in base UP or SRG configurations may result in synchronization failure and improper setup of sessions on the subordinate.

Session Sync

Once the session is up on the primary node, the entire session information gets synced to the subordinate node. This includes dynamic synchronization of updates such as CoA or service logon.

Configuring UP Geo Redundancy

To configure the subscriber redundancy group in the control plane, use the following sample configuration:

```
config
  user-plane instance instance_id
    user-plane user_plane_name
      subscriber-redundancy
        group group_name
          disable
          domain-identifier domain_name
          peer-identifier peer_id
```

```
port-id-map port-name port_name port_number
preferred-role-active
revertive-timer revertive_timer_value
exit
```

NOTES:

- **subscriber-redundancy**: Configures subscriber geo-redundancy. All SRG groups are configured in this mode.
- **group** *group_name*: Specifies the name of the subscriber redundancy group that is unique to a user plane.
- **disable**: Disables an SRG group without deleting the entire configuration of the group. By default, an SRG group is enabled.
- **domain-identifier** *domain_name*: Specifies the domain name to identify all groups between two user planes.
- **peer-identifier** *peer_id*: Identifies the peer user-plane for the group. This identifier must be unique across all groups in the control plane. The same peer-identifier must be configured in the peer user-plane.
- **port-id-map port-name** *port_name port_number*: Specifies the mapping of access interfaces between user planes. At least one **port-map-id** must be configured.
- **preferred-role-active**: This is an optional configuration.

  Sets the preferred role active for user plane. Default value: false.
- **revertive-timer** *revertive_timer_value*: This is an optional configuration.

  Specifies the revertive timer in seconds. *revertive_timer_value* must be an integer in the range of 60 to 3600. This command is available only when **preferred-role-active** is configured.

Configuration Example

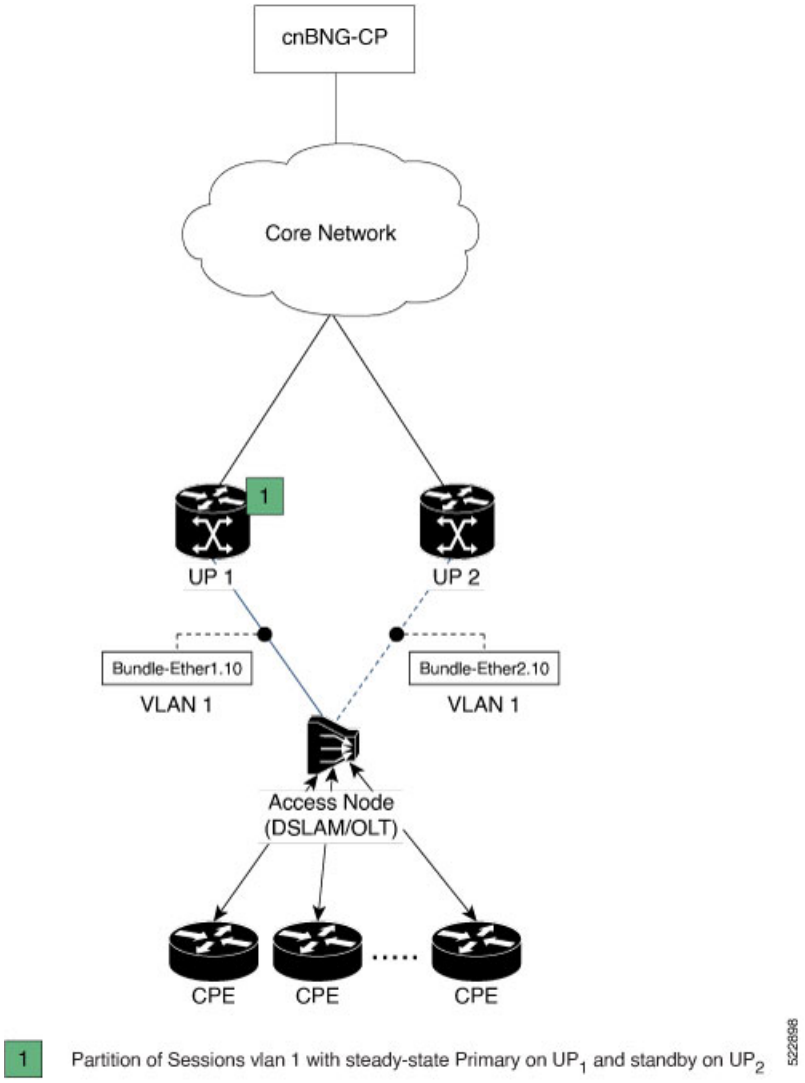
The following is a sample configuration for configuring UP Geo Redundancy, as illustrated in Figure 5.

```
config
  user-plane
   instance 1
    user-plane user-plane1
    peer-address ipv4 {UP1 ipv4-address}
     subscriber-redundancy
      group Group1
       preferred-role-active
       revertive-timer       3600
       domain-identifier     domain1
       peer-identifier       Peer1
       port-id-map port-name Bundle-Ether1.10 1
      exit
     exit
    exit
    user-plane user-plane2
    peer-address ipv4 {UP2 ipv4-address}
     subscriber-redundancy
      group Group1
       domain-identifier domain1
       peer-identifier   Peer1
       port-id-map port-name Bundle-Ether2.10 1
      exit
```

```
    exit
   exit
  exit
 exit
```

The following diagram illustrates the sample configuration.

Figure 39: Sample Configuration



Configuration Verification

To verify the configuration, execute the following commands:

- **show subscriber redundancy [count | debug | detail | gr-instance** *gr_instance_id* **| srg-peer-id** *srg_peer_id* **| upf** *upf_name* **]**
- **show subscriber redundancy-sync [gr-instance** *gr_instance_id* **| srg-peer-id** *srg_peer_id* **| upf** *upf_name* **]**

- **show subscriber dhcp [count | detail | filter** *filter_value* **| gr-instance** *instance_id* **| sublabel** *sublabel_name* **]**
- **show subscriber session [detail | filter { smupstate {** *upf_name*/**smUpSessionCreated } }]**
- **show subscriber synchronize [srg-peer-id** *peer_id* **| upf** *upf_name* **]**
- **show subscriber pppoe [detail | filter { srg-peer-id** *peer_id* **}]**

For more information on these commands, see the Monitoring Support, on page 572 section.

Configuring IPAM

Dynamic Pool Configuration

Use the following configuration to configure dynamic pool:

```
config
  ipam
    instance instance_id
    source local
      address-pool pool_name
        vrf-name string
        ipv4
          split-size
            per-cache value
            per-dp value
          exit
          address-range start_ipv4_address end_ipv4_address
        exit
        ipv6
          address-ranges
            split-size
              per-cache value
              per-dp value
            exit
            address-range start_ipv6_address end_ipv6_address
          exit
         prefix-ranges
            split-size
              per-cache value
              per-dp value }
            exit
            prefix-range prefix_value length prefix_length
          exit
        exit
      exit
```

Static Pool Configuration

Use the following configuration to configure static pool:

```
config
   ipam
    instance instance_id
     address-pool pool_name
        vrf-name string
        static enable user-plane srg_id
        ipv4
           split-size
           no-split
           exit
           address-range start_ipv4_address end_ipv4_address
        exit
        ipv6
           address-ranges
             split-size
               no-split
             exit
             address-range start_ipv6_address end_ipv6_address
           exit
           prefix-ranges
             split-size
               no-split
             exit
             prefix-length prefix_length
             prefix-range prefix_value length prefix_length
           exit
         exit
       exit
```

NOTES:

- **ipam**: Enters the IPAM Configuration mode.
- **instance** *instance_id*: Configures multiple instances for the specified instance and enters the instance sub-mode.
- **source local**: Enters the local datastore as the pool source.
- **address-pool** *pool_name* [**address-quarantine-timer**] [**offline**] [**static** *user_plane_name*] [**vrf-name** *string*]: Configures the address pool configuration. *pool_name* must be the name of the address pool.
- **ipv4**: Enters the IPv4 mode of the pool.
- **split-size** { **per-cache** *value* | **per-dp** *value* }: Specifies the size of the IPv4 range to be split for each IPAM cache allocation. The IPAM server consumes this configuration. The **no-split** command disables the splitting of the address-ranges into smaller chunks.

  per-cache *value*: Specifies the size of the IPv4 range to be split for each Data-Plane (User-Plane) allocation. The valid values range from 2 to 262144. The default value is 1024.

per-dp *value*: Specifies the size of the IPv4 range to be split for each Data-Plane (User-Plane) allocation. The valid values range from 2 to 262144 The default value is 256.

- **address-range** *start_ipv4_address end_ipv4_address*: Configures the IPv4 address range with the starting and ending IPv4 address.
- **ipv6**: Enters the IPv6 mode of the pool.
- **address-ranges**: Enters the IPv6 address ranges sub-mode.
- **prefix-ranges**: Enters the prefix ranges mode.
- **prefix-length** *prefix_length*: Specifies the IPv6 prefix length.
- **prefix-range** *prefix_value* **length** *prefix_length*: Specifies the IPv6 prefix range, and prefix length.
- **static enable user-plane** *srg_id*: Associates an user plane for the static pool.

Range-based interface list

The range-based interface list feature streamlines access interface management by enabling the configuration of contiguous bundle sub-interface ranges in a single command.

Table 123: Feature History Table

Range-based Interface List	2026.04.0	Introduces port-id-list-map for mapping sub-interface ranges to SRG port-id ranges and port-id-list for attaching subscriber profiles to sub-interface ranges.

Configuration guidelines

The **port-id** and **port-id-list** parameters are mutually exclusive. You must choose only one for configuration within a group or user-plane.

Configure Range-based Interfaces

SRG Configuration (port-id-list-map): Use this command to map a contiguous range of bundle sub-interfaces to a corresponding range of SRG port-ids.

```
configure
 user-plane instance <instance_id>
  user-plane <up_name>
   subscriber-redundancy
    group <group_name>
     port-id-list-map port-name <name> <identifier> sub-interface-range <start> <end>
id-range <start> <end>
     commit
```

- **port-id-list-map**: The command used to map a range of sub-interfaces to a range of port IDs for subscriber redundancy.
- **port-name**<*name*>: Specifies the physical or logical interface identifier.

- **sub-if-range<start> <end>**: Defines the range of sub-interface to be mapped.
- **id-range<start> <end>**: Defines the range of port IDs that correspond to the specified sub-interface range.

User-Plane Configuration (port-id-list): Use this command to associate a specific subscriber profile with a contiguous range of bundle sub-interfaces.

```
configure
 user-plane instance <instance_id>
  user-plane <up_name>
   subscriber-profile <profile_name>
    port-id-list <name> <identifier> sub-if-range <start> <end>
     subscriber-profile <profile_name>
     exit
     commit
```

- **port-id-list**: The command used to associate a specific subscriber profile.
- **port-name**: Specifies the physical or logical interface identifier.
- **sub-if-range <start> <end>**: Defines the range of sub-interface to be mapped.

L3 Routed Subscriber Sessions with Subscriber Redundancy Group

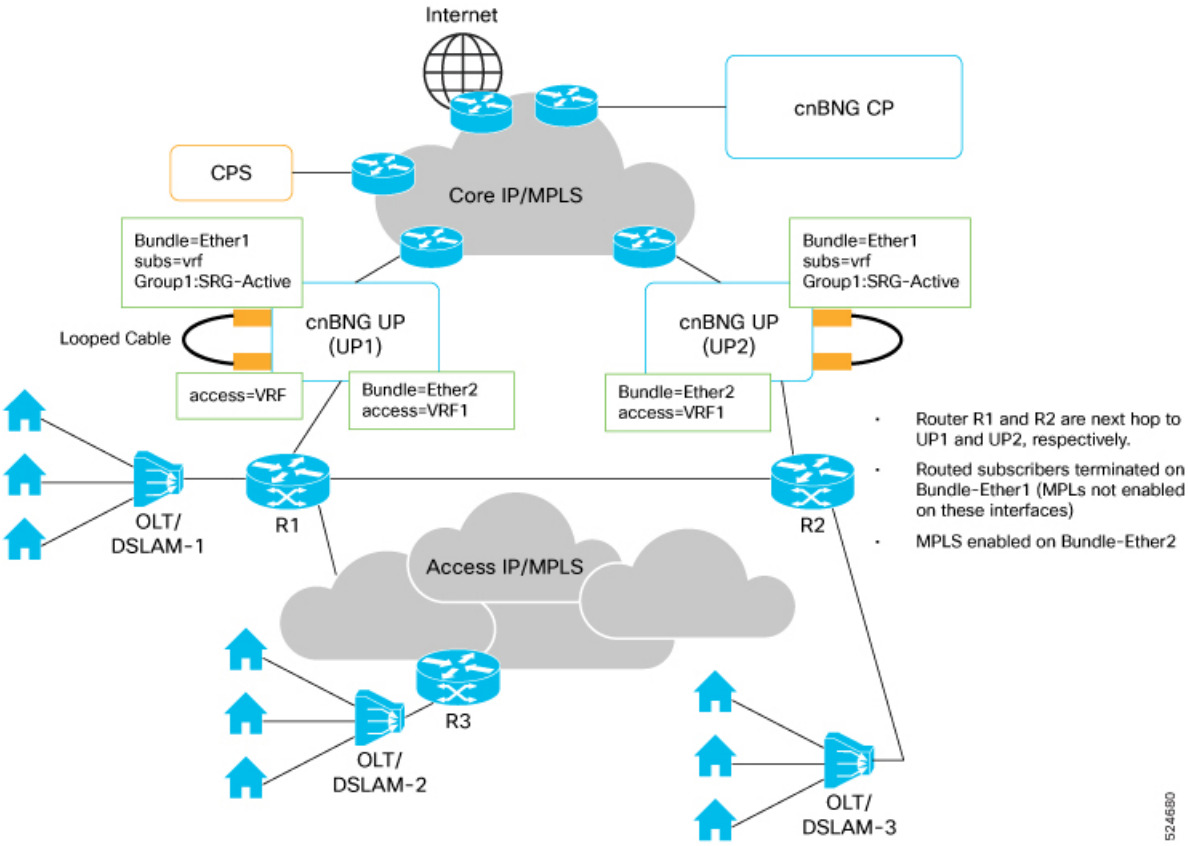
Table 124: Feature History

Feature Name	Release Information	Description
L3 Routed Subscriber Sessions with SRG	2026.04.0	Enabled support for sharing a single IPAM address pool across multiple L3 routed SRG groups within a VRF, using State Control Routes for address isolation.
L3 Routed Subscriber Sessions with SRG	2026.02.0	We have enhanced this feature to support IPv4, and dual-stack routed subscriber sessions, including on bundle-ether VLAN interfaces, which improves scalability and deployment flexibility in Layer 3 access networks. Previously, only IPv6 was supported.
L3 Routed Subscriber Sessions with SRG	2025.01.0	This feature enhances routing and redundancy for subscriber sessions by allowing subscribers to connect through a routed (L3) access network.

L3 Routed Subscriber Sessions allow subscriber connections via a routed access network, using SRG to provide redundancy. The SRG feature involves grouping access interfaces for failover (FO) and switchover (SO), ensuring continuous service in case of an active unit failure. This enhances network robustness and service continuity for subscribers connected through cnBNG.

The following topology illustrates the L3 routed subscriber sessions with SRG.

Figure 40: SRG Routed subscribers with access IP network



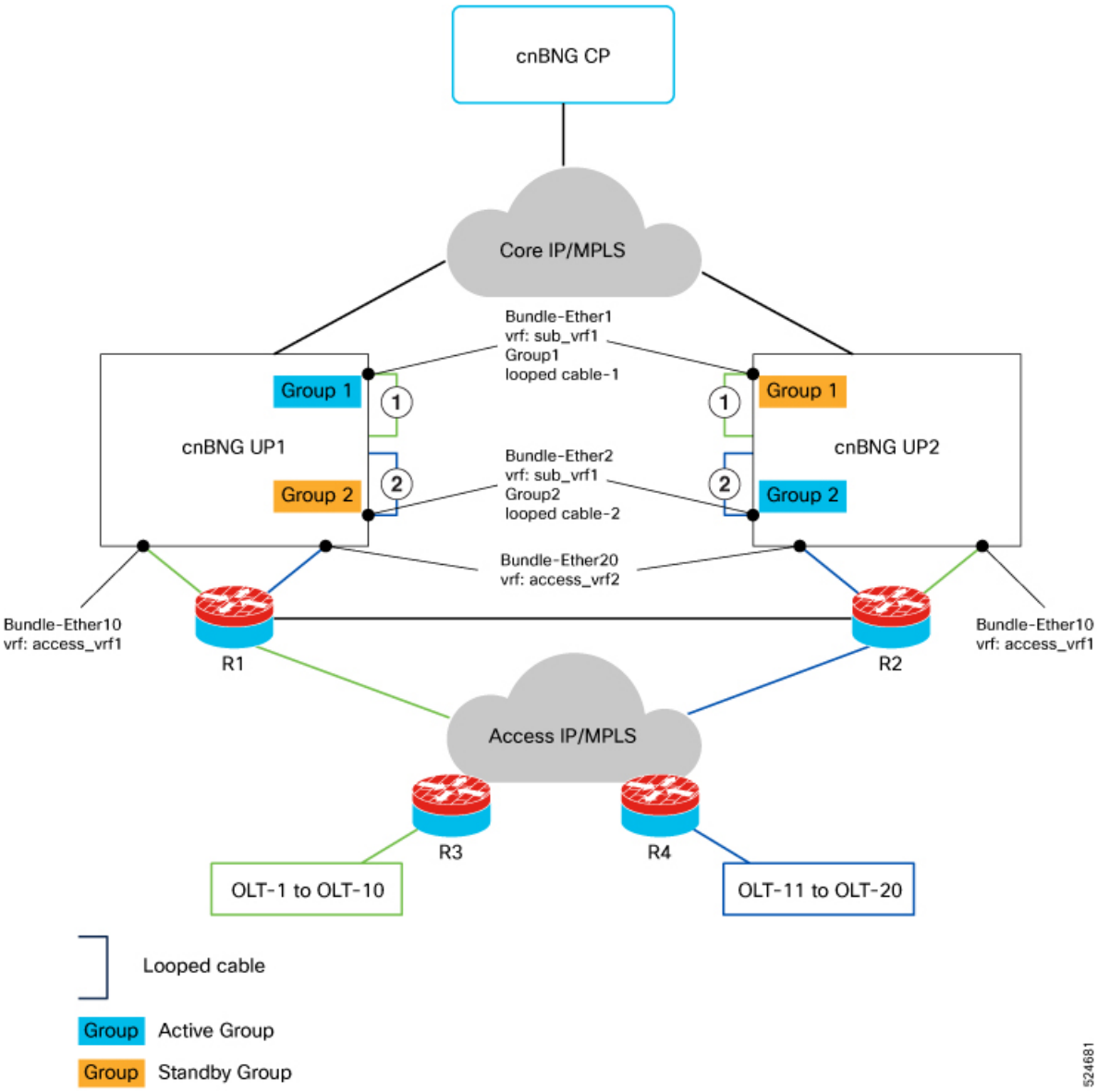
The session type determines the identifiers used for the session. For example, L2 connected sessions always use the MAC address as one of the key identifiers in both the control and data planes. In contrast, routed sessions use the MAC address as the session identifier in the control plane and the RADIUS server to uniquely identify the session. The session IP address is used in the data plane for data forwarding.

Supported Modes

The session distribution across SRGs can be in either of these modes:

- **Active-Active mode**: Distributes SRG groups across multiple user planes to balance load and ensure service continuity.
- **Hot-Standby Mode**: Uses standby user planes for quick failover, reducing downtime in case of active unit failure.

Figure 41: Active-Active mode with access IP or MPLS network



In this sample topology, two SRG groups, Group1 and Group2, are configured on both UP1 and UP2. For Group1, UP1 is the preferred active unit, while for Group2, UP2 is preferred. Ideally, Group1 should be active on UP1, and Group2 should be active on UP2. In an Active-Active mode setup, at least two SRG groups are required so that 50% of the groups are active on UP1 and the other 50% are active on UP2.

Both UPs have Bundle-Ether10 interfaces that are MPLS-enabled and are part of the same access VRF: `access_vrf1`. Since cnBNG is not supported on MPLS-enabled interfaces, a looped cable must be used to convert MPLS traffic into IP traffic for cnBNG-enabled access interfaces. Similarly, Bundle-Ether20 is configured on both UPs and is part of access VRF: `access_vrf2`.

For Group1, one port of looped-cable 1 is connected to Bundle-Ether1, and cnBNG is enabled on this interface. Note that routed subscribers are supported only on the main bundle interfaces. Bundle-Ether1 belongs to

Group1, so all subscribers on Bundle-Ether1 are considered Group1 subscribers. The subscriber VRF for Group1 is `sub_vrf1`.

For Group2, one port of looped-cable 2 is connected to Bundle-Ether2, and cnBNG is enabled on this interface. Bundle-Ether2 is part of Group2, meaning all subscribers on Bundle-Ether2 are Group2 subscribers. The subscriber VRF for Group2 is also `sub_vrf1`.



Note The SRG groups can share the same subscriber VRF or have different ones. There are no restrictions on subscriber VRF configuration, but each SRG group must have a unique access VRF.

Traffic Flow:

- **Upstream for Group1**: Traffic flows from CPE to the core network via OLTx, R3, R1, Bundle-Ether10, looped-cable 1, and Bundle-Ether1.
- **Downstream for Group1**: Traffic returns from the core network to CPE via Bundle-Ether1, looped-cable 1, Bundle-Ether10, R1, R3, OLTx.
- **Upstream for Group2**: Traffic flows from CPE to the core network via OLTx, R4, R2, Bundle-Ether20, looped-cable 2, and Bundle-Ether2.
- **Downstream for Group2**: Traffic returns from the core network to CPE via Bundle-Ether2, looped-cable 2, Bundle-Ether20, R2, R4, OLTx.

State-Control-Route: This is an aggregate route for each group, requiring configuration on both UPs. Each SRG group can have multiple state-control-routes, with a next hop specified for each. Once a group is created on the UP, these routes are sent from the Control Plane to the UP, where they are installed into the Routing Information Base (RIB). A route policy on the UP must advertise these routes to the core network to receive downstream traffic. The next hop in the state-control-route directs outgoing traffic from the subscriber interface to the access VRF.

Publish Gateway for upstream traffic from CPE

To effectively manage upstream traffic from Customer Premises Equipment (CPE), the IP address of the BNG access interface must be advertised to the IP/MPLS access network. This allows the network to route traffic originating from CPEs.

- **For Group1**: When Group1 is active in UP1, the IP address of the access interface on Bundle-Ether1 needs to be advertised to Router R3. This is done via the path: looped cable1 -> Bundle-Ether10 -> R1. The advertisement from UP1 should have higher metrics compared to UP2 to indicate that UP1 is the preferred route for Group1 traffic.
- **For Group2**: When Group2 is active in UP2, the IP address of the access interface on Bundle-Ether2 should be advertised to Router R4. This follows the path: looped cable2 -> Bundle-Ether20 -> R2. The advertisement from UP2 should have higher metrics compared to UP1 to make UP2 the preferred path for Group2 traffic.

DHCPv6 Relay Chaining

The DHCPv6 Relay Chaining feature is designed to manage DHCP packets relayed over multiple hops within a network, specifically intended for routed subscribers using DHCP as the First Sign of Life (FSOL).

Relay message processing in DHCPv6 networks

In the case of DHCPv6, each relay hop adds a Relay Forward message as the packet moves forward and removes a Relay Reply header on the return path. For example, if there are two relays between the end subscriber and the Broadband Network Gateway (BNG), the cnBNG control plane's DHCP server will encounter two Relay Forward headers. Typically, the first relay is a Light Weight Relay Agent (LDRA), and the second is an L3 relay agent. Consequently, when cnBNG functions as a DHCPv6 server, incoming packets will contain multiple relay forward headers, and the response packets must include corresponding Relay Reply headers based on these incoming headers.

Packet forwarding and address allocation in multi-hop relay environments

The Optical Line Terminal (OLT), acting as the LDRA, inserts a hop zero relay-forward header with a DHCPv6 circuit-id (interface-id) and remote-id, then sends it to the access router (R3). R3 is configured as a DHCPv6 L3 relay, using the cnBNG UP access interface IP as the helper address. R3 adds its own relay-header with its remote-id and circuit-id before forwarding the packet to cnBNG-UP. If the UP access interface is set as CNBNG routed, the multihop relay forward message is sent to the control plane as a routed DHCP packet of the GPTU type. In the DHCP pod, the hop zero circuit-id/interface-id is used by the cnBNG control plane for address chunk allocation.

Prefix route management and relay binding in multihop DHCPv6 setups

In a multihop relay setup, the L3 relay agent must install a prefix route (IAPD) for routing IAPD traffic. This route is typically installed when the Relay-Reply packet from the BNG confirms prefix allocation by parsing the IA prefix option. The relay agent also maintains a binding for this route, which is removed when a Relay Forward Release packet is received from the LDRA. By default, relay binding and route installation are enabled for IAPD (prefix) and can be configured for IANA if needed. Manual clearing of the relay binding is required for session termination triggers other than a CPE DHCPv6 Release, such as an admin clear on the BNG or a POD from RADIUS.

Address Chunk or Range Allocation

For Layer 2 connected subscribers, the BNG acts as the gateway, and address chunk allocation is based on the access interface or UPF, depending on whether the deployment is SRG or non-SRG. However, in routed scenarios, the gateway IP for subscribers is located at the first L3 hop, such as the Cell Site Switch (CSS) or R3. Therefore, address chunks should be allocated per CSS to ensure subscriber traffic is routable. The first IP in the chunk should be configured on the subscriber-facing interface of the CSS, serving as the gateway for subscribers. For IPv6, the source address of the DHCPv6 response packet, which may be a link-local address, acts as the gateway.

To allocate subnets per CSS (R3), each CSS must be identifiable by a unique identifier shared among its subscribers. In the cnBNG control plane, the identifiers for chunk allocation (DP/Dataplane keys) are:

- UPF Name for non-SRG L2 deployments
- SRG peer-id for SRG L2 deployments
- DHCP option based method, for example, **circuit-id**.

  This is a sample configuration for the DHCP option based method:

```
profile dhcp DHCP_1
ipv6
 mode server
 server
  iana-pool-name pool-1
  ipam-dp-key circuit-id delimiter # substring 0
  exit
 exit
ipv4
```

```
 mode server
 server
  pool-name pool-1
  ipam-dp-key circuit-id delimiter # substring 0
 exit
exit
```

In this context, the **ipam-dp-key** is specified as the circuit-id. This means the circuit-id, which is the DHCPv6 option 18 interface-id of the hop zero relay forward header, is used to determine the data plane identifier. There is an option to use the entire interface-id or a portion of it as the DP-key. You can extract a substring of the interface-id by using a delimiter to split the string.

When configured, DHCP uses the circuit-id as the dp-key information during IP allocation, release, or validation processes. The DHCP pod also sends an indicator to the node manager or IPAM during these processes to help IPAM identify IP interactions related to routed sessions. In routed sessions, aggregate route management is handled at the group level, so during chunk allocation, IPAM does not configure subnet or summary routes to the UP.

Pre-Allocation of Gateway IP and Address Chunks

For detailed information about this feature, see Pre-Allocation of Gateway IP and Address Chunks.

Disabling L3 Routed Subscriber Sessions with SRG

The L3 Routed Subscriber Sessions with SRG feature functions only when BNG is enabled on the access interface in both cnBNG UP and CP. To disable the feature, remove the BNG enablement configuration from the respective access interface. Similarly, to disable SRG, configure **disable** in the respective SRG group's configuration in the cnBNG CP.

Single pool for L3 routed SRG groups

Starting with 2026.04.0, we support the allocation of a single IPAM address pool per VRF instance for L3 routed subscriber sessions. This feature allows a single IPAM address pool to be shared across multiple L3 SRG groups, significantly reducing configuration complexity in Layer 3 access networks.

Guidelines to configure single pool for L3 routed SRG groups

Use these guidelines to configure a single pool for L3 routed SRG groups.

- Prefix validation: IP prefixes defined in the `state-control-route` within the SRG configuration must be valid subsets of the address ranges defined in the shared IPAM address pool.
- Route matching: The calculated subnet route for each address range must match the configured `state-control-route`.
- VRF consistency: The VRF on the `state-control-route` must match the IPAM VRF. If multiple `state-control-routes` exist in an srg-peer, all must share the same VRF; otherwise, the request is rejected.
- Active UPF: The cnBNG CP considers routes only from the active UPF for the specific SRG group.
- Address range fragmentation: If you configure multiple `state-control-route` under a single SRG group, cnBNG CP may not always use routes in the same order for each allocation request. Chunks may be allocated across different address ranges instead of using a single range first. This distribution can result in fragmentation across the available address space. Consider this distribution pattern when designing high-density deployments that require contiguous address allocation.

- SRG peer allocation: The cnBNG CP does not enforce strict L2 or L3 constraints at the pool level. The IPAM module allows the pool to be utilized according to the specific requirements of the SRG group without restricting a pool to exclusively L2 or L3 usage.
  - L3 SRG peer allocation: When an action command includes an L3 srg-peer-id and `state-control-route`, the cnBNG CP performs a context-aware lookup. Allocation is scoped specifically to the provided `state-control-route`, ensuring the chunk is fetched from the associated address range.
  - L2 SRG peer allocation: In the absence of L3-specific routing requirements, the system performs a general search in the cache pod and retrieves the first available free chunk, optimizing for immediate resource availability.

Restrictions for L3 Routed Subscriber Sessions with SRG

These restrictions apply to the L3 Routed Subscriber Sessions with SRG feature:

- Only dynamic IP allocation is supported.
- The **ipam-address-chunk** action command is supported only for SRG deployment.
- There is no impact on groups onboarded before SRG is disabled.
- Ensure to follow the MOP to remove all address pre-allocations using **ipam-address-chunk release** command before deleting an SRG group.
- The control plane does not restrict using the same ipam-dp-key for different pools or pool-group-tags. So, make sure to maintain uniqueness across pools.
- When associating multiple pools with a profile using a pool-group-tag, ensure that the same tag is used for the **ipam-address-chunk allocate** command. The **ipam-address-chunk release** command will release chunks across all pools associated with this tag. This is applicable for IPv6, IPv6-Addr, IPv6-prefix

  The **ipam-address-chunk allocate** and **ipam-address-chunk release** commands are not supported with pool group tag for IPv4.
- The control plane does not restrict using the same ipam-dp-key for different SRG groups. So, make sure to maintain uniqueness across SRG groups.
- The control plane does not restrict the use of **ipam-address-chunk release** command even when the IP is currently in use. Performing this action can cause inconsistencies within the system.
- Make sure that routed and state-control-route configurations on both user-planes are aligned. Otherwise, the system's behavior may become unpredictable.
- The control plane does not reject conflicting state-control-routes.
- Marking an address-range or pool offline without freeing address-chunks (via **ipam-address-chunk release**) is not supported.

Configure L3 Routed Subscriber Sessions with SRG

To configure L3 Routed Subscriber Sessions with SRG, use the following sample configuration:

Procedure

Step 1 Define SRG groups, assign access interfaces, and configure state-control-routes to manage traffic routing within SRG groups.

Example:

```
config
   user-plane instance instance_id
      user-plane user_plane_name
         peer-address ipv4 ipv4_address
         subscriber-profile subs-ipoe
         subscriber-redundancy
            group group_name
               peer-identifier peer_id
               l3-routed
               port-id-map port-name port_name port_number
               state-control-route route_name afi ipv6 aggregate_route vrf vrf_name
               state-control-route route_name afi ipv4 aggregate_route vrf vrf_name
               exit
```

Note Each SRG group can support multiple state-control-routes, and these must be configured on both UPs.

The following is a sample configuration:

```
user-plane
instance 1
 user-plane asr9k-1
  peer-address ipv4 10.6.1.1
  subscriber-profile subs-routedipoe-1
  subscriber-redundancy
   group Group1
    peer-identifier Peer1
    l3-routed
    port-id-map port-name Bundle-Ether5 1
    state-control-route r1 ipv6 2002:ab::/48 vrf FTTX_SUB
    state-control-route r2 ipv4 10.1.1.1/15 vrf FTTX_SUB
   exit
  exit
 exit
 user-plane asr9k-2
  peer-address ipv4 10.6.1.2
  subscriber-profile subs-routedipoe-1
  subscriber-redundancy
   group Group1
    peer-identifier Peer1
    l3-routed
    port-id-map port-name Bundle-Ether5 1
    state-control-route r1 ipv6 2002:ab::/48 vrf FTTX_SUB
    state-control-route r2 ipv4 10.1.1.2/15 vrf FTTX_SUB
   exit
  exit
 exit
```

NOTES:

- **subscriber-redundancy**: Configures subscriber geo-redundancy. All SRG groups are configured in this mode.
- **group** *group_name*: Specifies the name of the subscriber redundancy group that is unique to a user plane.
- **peer-identifier** *peer_id*: Identifies the peer user-plane for the group. This identifier must be unique across all groups in the control plane. The same peer-identifier must be configured on the peer user-plane.
- **port-id-map port-name** *port_name port_number*: Specifies the mapping of access interfaces between user planes. At least one **port-map-id** must be configured.
- **preferred-role-active**: This is an optional configuration.

  Sets the preferred role active for user plane. Default value: false.
- **state-control-route** *route_name* **afi** *ipv6/ipv4 aggregate_route* **vrf** *vrf_name*: Programs the route to the UP for a specific routed SRG group based on the active or standby state of the UP.

Step 2 Configure IP pools for SRG groups. One IP pool must be configured per SRG group.

Example:

```
config
   ipam
     instance instance_id
     address-pool pool_name
        ipv6 | ipv4
           address-range start_ip_address end_ip_address
           exit
```

The following is a sample configuration:

```
ipam
 instance 1
  address-pool pool-1
   vrf-name vrf-1
   ipv4
    split-size
     per-cache 8192
     per-dp    4096
    exit
    address-range 10.1.0.1 10.1.255.255
   exit
   ipv6
    address-ranges
     split-size
      per-cache 8192
      per-dp    4096
     exit
     address-range 2001:DB8:: 2001:DB8:3::fff
    exit
    prefix-ranges
     split-size
      per-cache 8192
      per-dp    4096
     exit
     prefix-range 2002:ab:: length 48
    exit
   exit
  exit
```

```
 exit
exit
```

Step 3 Configure the state control route nexthop and access control route on both UPs. For configuration details, see Configure SRG section in the *cnBNG User Plane Configuration guide*.

The following is a sample configuration:

```
cnbng-nal location 0/RSP0/CPU0
subscriber-redundancy
 group Group1
  access-tracking track1
  access-control-route ipv6 ::/0 vrf vrf_1 next-hop-address 2001:dB8:4:0:4:4:1:3 active-tag 10
standby-tag 20
  access-control-route ipv4 10.0.0.0/0 vrf vrf_1 next-hop-address 10.5.0.3 active-tag 10 standby-tag
 20
  access-interface-list
   interface Bundle-Ether5
   exit
  state-control-next-hop-ip ipv4 10.5.0.1
  state-control-next-hop-ip ipv6 2001:dB8:4:0:4:4:1:1
  exit
 exit
exit
```

NOTES:

- **access-control-route ipv6** *ipv6_address* **vrf** *vrf_name* **next-hop-address** *next_hop_address* **active-tag** *value* **standby-tag** *value*: Configures the access control route (IPv6) in the hub VRF as access interface IP.
- **access-control-route ipv4** *ipv4_address* **vrf** *vrf_name* **next-hop-address** *next_hop_address* **active-tag** *value* **standby-tag** *value*: Configures the access control route (IPv4) in the hub VRF as access interface IP.
- **state-control-next-hop-ip ipv6** *ipv6_address* : Configures the specified IPv6 address as the next-hop IP for the state-control route, designating it as the hub VRF IP.
- **state-control-next-hop-ip ipv4** *ipv4_address* : Configures the specified IPv4 address as the next-hop IP for the state-control route, designating it as the hub VRF IP.

Step 4 Configure the IPAM data plane key.

Example:

```
profile dhcp dhcp_profile_name
 ipv6
  server
   iana-pool-name ipam_pool_name
   ipam-dp-key circuit-id delimiter value substring value
   end
  end
 end
```

The **ipam-dp-key** configuration in the DHCP profile specifies how to create the DP key.

The following is a sample configuration:

```
profile dhcp server-1
 ipv6
  server
   iana-pool-name dhcp-ipv6-iana
   iapd-pool-name dhcp-ipv6-iapd
```

```
   dns-servers [ 2001::5 ]
   ipam-dp-key circuit-id delimiter # substring 0
   domain-name cisco.com
   lease days 1
   lease hours 1
   lease minutes 1
  exit
 exit
exit
```

NOTES:

- **profile dhcp** *dhcp_profile_name* : Specifies the DHCP profile name.
- **ipv6** : Enters IPv6 configuration mode.
- **server**: Specifies the IPv6 server details.
- **iana-pool-name**: Specifies the Internet Assigned Numbers Authority (IANA) pool name.
- **ipam-dp-key circuit-id** *value* **delimiter***value* **substring** *value*: Specifies the data plane key for IP management.
  - **circuit-id** *value*: The DHCPv6 interface-id found in the hop zero relay header will be used as the key for IPAM in the data plane.
  - **delimiter** *value*: The delimiter value must be a single character and can be one of the following: [!@#$%^&*()_+].
  - **Substring** *value*: This option can only be set to 0 or 1. It allows the string to be split into two substrings based on the first occurrence of the specified delimiter.

Step 5 Use the **ipam-address-chunk** action command to configure the pre-allocation of gateway IP and address chunks. For configuration details, see Configure Pre-Allocation of Gateway IP and Address Chunks, on page 322.

The output of this action command provides information about the chunk and the first IP address that were reserved. For example,

```
bng# ipam-address-chunk allocate instance-id 1 pool-name dhcp-l3-routed-vlan-11 ipv6-prefix
ipam-dp-key INGJRJKTMDHRTW6001ENBESR001 srg-peer-id Peer1

Sat Aug  24 06:27:29.200 UTC+00:00
result
Gateway Address: 2001:DB8::1/112
```

```
bng# ipam-address-chunk allocate instance-id 1 pool-name dhcp-l3-routed-vlan-11 ipv6-addr ipam-dp-key
 INGJRJKTMDHRTW6001ENBESR001 srg-peer-id Peer1

Sat Aug  24 06:27:29.200 UTC+00:00
result
Gateway Address: 2002::1/48
```

```
bng# ipam-address-chunk allocate instance-id 1 pool-name dhcp-l3-routed-vlan-11 ipv4-addr ipam-dp-key
 INGJRJKTMDHRTW6001ENBESR001 srg-peer-id Peer11

Sat Aug  24 06:27:29.200 UTC+00:00
result
Gateway Address: 10.1.64.1/20
```

Step 6 Use the **show ipam { dp | dp-tag }** *value* **{ ipv6-addr | ipv6-prefix | ipv4-addr }** command to view the reserved IP address and the summary route of the allocated chunks.

Example:

```
bng# show ipam dp INGJRJKTMDHRTW6001ENBESR001 ipv6-addr

Tue Apr  28 09:03:28.87 UTC+00:00
==========================================================================================================
Flag  Indication: S(Static) O(Offline) R(For Remote Instance) RF(Route Sync Failed) F(Fixed Chunk
for DP)
Other Indication: A+(Waiting for route update response) QT*(Quarantined due to route delete failure)

                  QT+(Waiting for route update response post timeout)
G:N/P Indication: G(Cluster InstId) N(Native NM InstId) P(Peer NM InstId)
==========================================================================================================
StartAddress          EndAddress       Route              GatewayAddress        G:N/P    Utilization
   Flag      AllocContext
==========================================================================================================
2001:DB8::8000        2001:DB8::bfff   2001:DB8::8000/114    2001:DB8::1/112    1:1/-1   0.01%
     F         dhcp-ipv6-iana-11(FTTX_SUB)
==========================================================================================================
```

```
bng# show ipam dp INGJRJKTMDHRTW6001ENBESR001 ipv4-addr

Tue Apr  28 08:43:28.377 UTC+00:00
==========================================================================================================
Flag  Indication: S(Static) O(Offline) R(For Remote Instance) RF(Route Sync Failed) F(Fixed Chunk
for DP) V(Virtual)
Other Indication: A+(Waiting for route update response) QT*(Quarantined due to route delete failure)

                  QT+(Waiting for route update response post timeout)
G:N/P Indication: G(Cluster InstId) N(Native NM InstId) P(Peer NM InstId)
==========================================================================================================
StartAddress          EndAddress      Route          GatewayAddress   G:N/P    Utilization   Flag
     AllocContext
==========================================================================================================
10.1.64.0            10.1.79.255    10.1.64.0/20       10.1.64.1/20    1:0/-1   2.34%         F
      L3RLSP-vlan-111(vrf-1)
==========================================================================================================
```

The flag value **F** signifies that it is a fixed chunk for the DP, assigned when the **ipam-address-chunk allocate** command is executed.

Step 7 Configure the gateway address on the access side router.

The following is a sample configuration:

```
interface TenGigE0/0/0/13.100
 vrf vrf1
 ipv6 nd other-config-flag
 ipv6 nd managed-config-flag
 ipv6 address 2001:DB8::1/112
 ipv6 address 2002:ab::1/48
 ipv6 enable
exit
 encapsulation dot1q 100
exit
```

In this example, the IPv6 addresses, *2001:DB8::1/112*, and *2002:ab::1/48* are configured as gateway addresses on the access side router.

Step 8 Use the **show subscriber session filter** command to verify the routed session details.

Example:

```
bng# show subscriber session filter { mac aa11.0000.0001 } detail
```

```
Thu Jul  11 16:37:30.579 UTC+00:00
subscriber-details
{
  "subResponses": [
    {
      "subLabel": "16777228",
      "srgPeerId": "Peer1",
      "srgId": "Group1",
      "mac": "aa11.0000.0001",
      "acct-sess-id": "Local_DC_16777228",
      "sesstype": "routedipoe",
      "state": "established",
      "subCreateTime": "Thu, 11 Jul 2024 15:59:49 UTC",
      "dhcpAuditId": 2,
      "transId": "1",
<snip>
```

Step 9 You can also filter sessions based on the session type. Use the **show subscriber session filter { sesstype routedipoe }** command to filter routed subscriber sessions.

Step 10 Use the **show subscriber session filter { ipam-dp-key** *dp_key* **}** command to filter sessions based on ipam-dp-key.

Step 11 Use the **show subscriber dhcp filter { ipam-dp-key** *dp_key* **}** command to filter sessions based on ipam-dp-key in DHCP profile.

Session Synchronization between UPs

This section describes different scenarios where the subscriber needs to be synchronized to a UP manually.

Scenario 1

One UP in a Subscriber Redundancy group is active, and a session is created. Now, another UP in the same SRG is connected for the first time. All the groups in the second UP become standby. To synchronize the sessions with the second (standby) UP, use the following CLI command:

```
bng# subscriber redundancy session-synchronize add domain [ domain_ID ]
target-upf upf_ID
```

You can also use the following CLI command, if there are only two UPs involved (as in Scenario 1):

```
bng# subscriber redundancy session-synchronize add upf-id [ upf_ID ]
target-upf upf_ID
```

Example-1:

```
subscriber redundancy session-synchronize add domain [ Domain12 ] target-upf Upf2
```

The above CLI command synchronizes all the subscribers from active UP, which are part of *Domain12*, to the target UP (*Upf2*).

Or,

```
subscriber redundancy session-synchronize add upf-id [ Upf1 ] target-upf Upf2
```

The above CLI command synchronizes all the subscribers from *Upf1* to *Upf2*.

Example-2:

The following is a sample configuration if two UPs are active, and a third UP is connected later.

```
subscriber redundancy session-synchronize add domain [ Domain12 Domain13 ] target-upf Upf1
```

The above CLI command synchronizes all the subscribers from the active UPs, which are part of *Domain12*, and *Domain13* to the target UP (*Upf1*).

Scenario 2

Initially, a Subscriber Redundancy group is configured on only one UP, and a session is created. Later, the second UP is configured with SRG. Now, to synchronize the session with the second UP in the group, use the following CLI command:

bng# **subscriber redundancy session-synchronize add peer-id [** *peer_ID* **] target-upf** *upf_ID*

Example:

```
subscriber redundancy session-synchronize add peer-id [ Peer1 ] target-upf Upf2
```

The above CLI command synchronizes subscribers that are part of a group with peer-id *Peer1* to target UP (*Upf2*).

Scenario 3

A group is removed from an UP. To remove sessions in the group, use the following CLI command:

bng# **subscriber redundancy session-synchronize delete peer-id [** *peer_ID* **] target-upf** *upf_ID*

Example:

```
subscriber redundancy session-synchronize delete peer-id [ Peer1 ] target-upf Upf2
```

The above CLI command removes subscribers from target UP (*Upf2*) that are part of the SRG group with peer-id *peer1*.

Scenario 4

All groups are removed from an UP. To remove all sessions in an UP, use the following CLI command:

bng# **subscriber redundancy session-synchronize delete domain [** *domain_list* **] target-upf** *upf_ID*

Example:

```
subscriber redundancy session-synchronize delete domain [ domain12 domain13 ] target-upf
Upf3
```

The above CLI command deletes all the subscribers that are part of the domains *domain12*, and *domain13* from the target UP (Upf3).

Or,

```
subscriber redundancy session-synchronize delete upf-id [ Upf3 ] target-upf Upf3
```

The above CLI command deletes all the subscribers that are related to *Upf3* from the target UP (*Upf3*).

Note You can also delete all non-SRG sessions in the UP.

Scenario 5

An UP from a group is replaced with another UP. To synchronize the sessions, use the following CLI commands:

```
bng# subscriber redundancy session-synchronize delete peer-id [ peer_ID ]
target-upf old_upf_id
```

```
bng# subscriber redundancy session-synchronize add peer-id [ peer_ID ]
target-upf new_upf_id
```

Example:

```
subscriber redundancy session-synchronize delete peer-id [ peer1 ] target-upf Upf1
subscriber redundancy session-synchronize add peer-id [ peer1 ] target-upf Upf2
```

The above CLI commands remove the sessions in the group with peer-id *peer1* from *Upf1*, and add the group to *Upf2*.

Scenario 6

An UP is replaced with another UP in all the groups in a domain. To synchronize the sessions, use the following CLI commands:

```
bng# subscriber redundancy session-synchronize delete domain [ domain_ID ]
 target-upf upf_ID
```

```
bng# subscriber redundancy session-synchronize add domain [ domain_ID ]
target-upf upf_ID
```

Example:

```
subscriber redundancy session-synchronize delete domain [ domain1 ] target-upf Upf1
subscriber redundancy session-synchronize add domain [ domain1 ] target-upf Upf2
```

The above CLI commands remove the sessions in the groups that are part of *domain1* from *Upf1*, and add the groups to *Upf2*.

Scenario 7

All domain/group/peers are moved from one UP to another. Initially, to delete all subscribers from the UP, use the following CLI command:

```
bng# subscriber redundancy session-synchronize delete upf [ upf_ID ]
target-upf upf_ID
```

Example:

```
subscriber redundancy session-synchronize delete upf [ Upf1 ] target-upf Upf1
```

The above CLI command removes all the sessions from *Upf1*.

Configure the second UP with the configurations deleted from the first UP. Then, to synchronize the sessions, use the following CLI command:

```
bng# subscriber redundancy session-synchronize add domain [ domain_list ]
target-upf upf_ID
```

Example:

```
subscriber redundancy session-synchronize add domain [ domain1...domainN ] target-upf Upf2
```

The above CLI command synchronizes all the sessions that are in the list of given domains to the new UP (Upf2).

Route Synchronization between CP and UP

Use the following CLI command to synchronize the routes between the Control Plane and the User Plane.

subscriber route-synchronize upf *upf_name*

To check the status of route synchronization, use the following CLI commands:

- **subscriber route-synchronize upf** *upf-name* **status**
- **show subscriber route-synchronize upf** *upf-name*

Order of Reconciliation

It is recommended to perform the reconciliation activity in the following order:

1. Group reconciliation
2. Route reconciliation
3. CP reconciliation (CP-Audit)
4. CP-UP reconciliation

Monitoring Support

This section describes the monitoring support information for the UP Geo Redundancy feature.

Use the following show and clear commands for troubleshooting. The output of these commands provides specific configuration and status information.

clear subscriber sessmgr

Use this command to clear subscribers.

clear subscriber sessmgr [gr-instance *gr_instance_id* **| srg-peer-id** *srg_peer_id* **| upf** *upf_name* **]**

NOTES:

- **clear subscriber sessmgr srg-peer-id** *srg_peer_id*: Clears subscribers in CP and both UPs.
- **clear subscriber sessmgr upf** *upf_name* **srg-group-id** *srg_group_id*: If the group is active, this command clears sessions in CP and both UPs. If the group is standby, this command clears sessions in the standby UP.

clear subscriber pppoe

Use this command to clear PPPoE subscriber sessions.

clear subscriber pppoe { srg-peer-id *srg_peer_id* **| upf** *upf_name* **}**

NOTES:

- **clear subscriber pppoe srg-peer-id** *srg_peer_id*: Clears PPPoE sessions based on the SRG peer ID.
- **clear subscriber pppoe upf** *upf_name*: Clears PPPoE sessions based on the UPF name.

show subscriber redundancy

Use this command to display the key values of SRG groups.

show subscriber redundancy [count | debug | detail | gr-instance *gr_instance_id* **| srg-peer-id** *srg_peer_id* **| upf** *upf_name* **]**

NOTES:

- **show subscriber redundancy count**: Displays the count of SRG groups.
- **show subscriber redundancy detail**: Displays the detailed content of SRG groups.
- **show subscriber redundancy upf** *upf_name*: Displays all the groups related to UPF.
- **show subscriber redundancy peer-id** *peer_id* **debug**: Displays the detailed output with event history.

The following is a sample output of the **show subscriber redundancy detail** command:

```
bng# show subscriber redundancy detail
Fri Apr  29 14:48:36.840 UTC+00:00
subscriber-details
{
  "subResponses": [
    {
      "PeerID": "Peer15993-x",
      "GroupID": "Group-5-3-15993-x",
      "UP List": {
        "asr9k-3": {
          "N4 State": "Connected",
          "Srg State": "Up",
          "RoleChangeInProgress": true,
          "Srg Role": "Active",
          "Interface map": {
            "GigabitEthernet11636": 1,
            "GigabitEthernet11637": 2
          }
        },
        "asr9k-5": {
          "N4 State": "Disconnected",
          "Srg State": "Init",
          "Srg Role": "Standby",
          "Interface map": {
            "GigabitEthernet58174": 1,
            "GigabitEthernet58175": 2
          }
        }
      }
    }
```

```
  ]
}
```

show subscriber redundancy-sync

Use this command to display the subscriber reconciliation details.

show subscriber redundancy-sync [gr-instance *gr_instance_id* **| srg-peer-id** *srg_peer_id* **| upf** *upf_name* **]**

NOTES:

- **gr-instance** *gr_instance_id*: Displays the reconciliation details for the specified GR instance.
- **srg-peer-id** *srg_peer_id*: Displays the reconciliation details for the specified SRG peer ID.
- **upf** *upf_name*: Displays the reconciliation details for the specified UPF.

The following is a sample output of the **show subscriber redundancy-sync upf** *upf_name* command:

```
bng#  show subscriber redundancy-sync upf asr9k-1
Tue Apr  5  17:31:15.659 UTC+00:00
subscriber-details
{
  "Upf": "asr9k-1",
  "State": "Completed",
  "Status": "Passed",
  "Total Number of Groups": 2914,
  "Number of enabled Groups": 2914,
  "Maximum Duration": 180,
  "Started": "2022-04-05 17:31:30 +0000 UTC",
  "Ended": "2022-04-05 17:31:33 +0000 UTC",
  "Time Taken": "3 Seconds"
}
```

show subscriber dhcp

Use this command to display the DHCP CDL record keys per session.

show subscriber dhcp [count | detail | filter *filter_value* **| gr-instance** *instance_id* **| sublabel** *sublabel_name* **]**

NOTES:

- **show subscriber dhcp detail**: Displays the session details from DHCP CDL record.

The following is a sample output of the **show subscriber dhcp** command:

```
bng# show subscriber dhcp
Mon Mar  14 09:12:59.135 UTC+00:00
subscriber-details
{
  "subResponses": [
    {
      "records": [
        {
          "cdl-keys": [
            "aa11.0000.0001:m:100:v1:200:v2:1:p:Peer1:r@dhcp",
            "sublabel:33554433@dhcp",
            "type:dhcp",
            "mac:aa11.0000.0001",
```

```
            "srg-peer-id:Peer1",
            "upf:asr9k-2",
            "upf:asr9k-1",
            "port-id:asr9k-1/GigabitEthernet0/0/0/1",
            "port-id:asr9k-2/GigabitEthernet0/0/0/3",
            "vrf:ISP",
            "ipv4-addr:pool-ISP/11.0.96.2",
            "ipv4-pool:pool-ISP",
            "ipv4-range:pool-ISP/11.0.0.1",
            "ipv4-startrange:pool-ISP/11.0.96.0",
            "ipv4-state:bound",
            "ipv6-addr-startrange:pool-ISP/1:2::2000",
            "ipv6-addr:pool-ISP/1:2::2000",
            "ipv6-addr-pool:pool-ISP",
            "ipv6-addr-range:pool-ISP/1:2::1",
            "ipv6-addr-state:bound",
            "afi:dual"
          ]
        }
      ]
    }
  ]
}
```

show subscriber pppoe

Use this command to display information about PPPoE subscribers.

show subscriber pppoe [detail | filter { srg-peer-id *srg_peer_id* **}]**

NOTES:

- **show subscriber pppoe detail**: Displays detailed information about PPPoE subscriber sessions on a router.
- **show subscriber pppoe filter { srg-peer-id** *srg_peer_id* **}** : Filters PPPoE sessions based on the SRG peer ID.

Examples

The following is a sample output of the **show subscriber pppoe detail** command:

```
bng# show subscriber pppoe detail

Fri Jun  14 12:44:52.471 UTC+00:00
subscriber-details
{
  "subResponses": [
    {
      "state": "complete",
      "key": {
        "routerID": "asr9k-1",
        "portID": "GigabitEthernet0/0/0/1",
        "outerVlan": 100,
        "innerVlan": 200,
        "macAddr": "cc11.0000.0001",
        "pppoesessionID": 32771,
        "sublabel": "33554435",
        "upSubID": "4",
        "SrgPeerID": "Peer1",
        "SrgGroupID": "Group1",
```

```
        "SrgIntfID": "1"
      },
      "flags": [
        "SM_START_DONE",
        "SM_ACTIVATE_DONE",
        "SM_UPDATE_DONE",
        "PPPOE_UP_DONE",
        "IPCP_UP",
        "IPV6CP_UP"
      ],
      "pppoeInfo": {
        "profileName": "abc",
        "mtu": 1500
      },
      "lcpInfo": {
        "state": "opened",
        "keepAliveInterval": 60,
        "keepAliveRetries": 5,
        "localMru": 1500,
        "peerMru": 1500,
        "localMagic": "0xc23c756",
        "peerMagic": "0x112233",
        "authOption": "PAP",
        "authCompleted": true,
        "username": "cnbng"
      },
      "ipcpInfo": {
        "state": "opened",
        "peerIpv4Pool": "pool-ISP",
        "peerIpv4Address": "11.0.32.2",
        "peerIpv4Netmask": 22,
        "localIpv4Address": "11.0.32.1",
        "isIpamPoolIPaddr": true
      },
      "ipv6cpInfo": {
        "state": "opened",
        "localIntfID": "0x1",
        "peerIntfID": "0xcc11000000010000"
      },
      "sessionType": "pta",
      "vrf": "default",
      "AuditId": 4,
      "slaacInfo": {
        "prefix": "3001:ab::",
        "prefixlength": 64,
        "poolname": "slaac-pool",
        "fsmstate": "connected",
        "profilename": "profile1",
        "otherconfig": true
      }
    }
  ]
}
```


show subscriber session

Use this command to display the session manager (SM) CDL record keys per session.

show subscriber session [detail | filter { smupstate { *upf_name*/**smUpSessionCreated } }]**

NOTES:

- **show subscriber session detail**: Displays the session details from SM CDL record.
- **show subscriber session filter { smupstate {** *upf_name*/**smUpSessionCreated } }**: Use this command to check whether the session is created in the respective UPF for the SRG sessions.

  The session count for both UPFs show up in both SM and DHCP CDL records after SRG is created successfully in the respective UPFs.

The following is a sample output of the **show subscriber session** command:

```
bng# show subscriber session
Mon Mar  14 09:12:52.653 UTC+00:00
subscriber-details
{
  "subResponses": [
    {
      "records": [
        {
          "cdl-keys": [
            "33554433@sm",
            "acct-sess-id:Local_DC_33554433@sm",
            "upf:asr9k-1",
            "port-id:asr9k-1/GigabitEthernet0/0/0/1",
            "feat-template:svc1",
            "feat-template:automation-feature-template-accounting",
            "type:sessmgr",
            "mac:aa11.0000.0001",
            "sesstype:ipoe",
            "sesstype:ipoeRouted"
            "srg-peer-id:Peer1",
            "smupstate:smUpSessionCreated",
            "up-subs-id:asr9k-1/1",
            "smupstate:asr9k-1/smUpSessionCreated",
            "srg-group-id:asr9k-1/Group1",
            "upf:asr9k-2",
            "port-id:asr9k-2/GigabitEthernet0/0/0/3",
            "srg-group-id:asr9k-2/Group1",
            "smstate:established",
            "up-subs-id:asr9k-2/1",
            "smupstate:asr9k-2/smUpSessionCreated",
            "afi:dual"
          ]
        }
      ]
    }
  ]
}
```

show subscriber synchronize

The **subscriber session-synchronize [srg-peer-id** *peer_id* **| upf** *upf_name* **]** command is used to synchronize subscriber information on the UP.

To view the status of subscriber information synchronization, use the following CLI command:

show subscriber synchronize [srg-peer-id *peer_id* **| upf** *upf_name* **]**

The following is a sample output of the **show subscriber synchronize** command:

```
bng# show subscriber synchronize srg-peer-id Peer108-x
Tue Apr  5  06:31:51.167 UTC+00:00
subscriber-details
```

```
{
  "asr9k-11": {
    "upf": "asr9k-11",
    "sync status": "sync start in progress",
    "sync state": "Start",
    "sync startTIme": "05 Apr 22 06:31 UTC",
    "sync srgGroupId": "Group-11-8-108-x"
  },
  "asr9k-8": {
    "upf": "asr9k-8",
    "sync status": "sync start in progress",
    "sync state": "Start",
    "sync startTIme": "05 Apr 22 06:31 UTC",
    "sync srgGroupId": "Group-11-8-108-x"
  }
}
```


show ipam dp

Use this command to view the list of UPFs to which the corresponding routes (both static and dynamic) are pushed.

- **show ipam dp** *peerid* { **ipv4-address** | **ipv6-address** | **ipv6-prefix** }

NOTES:

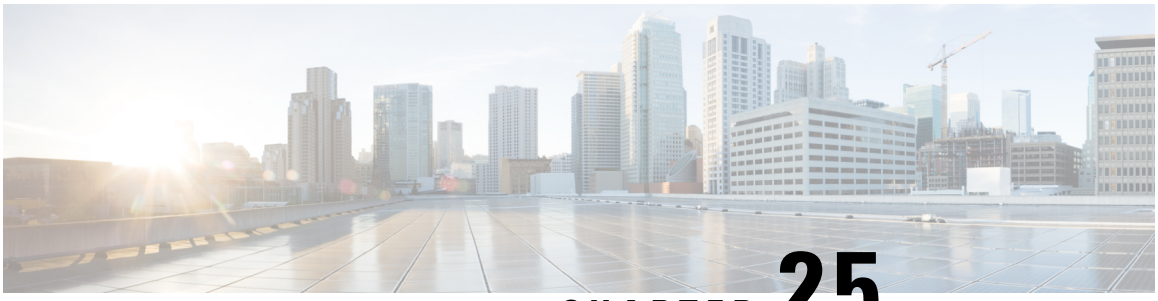
- **show ipam dp** *peerid* **ipv4-address**: Displays the UPFs of IPv4 address type
- **show ipam dp** *peerid* **ipv6-address**: Displays the UPFs of IPv6 address type
- **show ipam dp** *peerid* **ipv6-prefix**: Displays the UPFs of IPv6 prefix type

The following is a sample output of the **show ipam dp** *peerid* **ipv4-address**:

```
bng# show ipam dp peer-asr9k2 ipv4-addr
Wed Mar  30 12:43:09.313 UTC+00:00

=================================================================================================
Flag  Indication: S(Static) O(Offline) R(For Remote Instance) RF(Route Sync Failed)
G:N/P Indication: G(Cluster InstId) N(Native NM InstId) P(Peer NM InstId)
=================================================================================================
StartAddress          EndAddress      Route            G:N/P    Utilization   Flag
AllocContext
=================================================================================================
7.67.133.0            7.67.133.255    7.67.133.0/24    1:N/A                  S
srg-9k-static2(default)(asr9k-11,asr9k-12)
7.67.134.0            7.67.134.255    7.67.134.0/24    1:N/A                  S
srg-9k-static2(default)(asr9k-11,asr9k-12)
7.67.135.0            7.67.135.255    7.67.135.0/24    1:N/A                  S
srg-9k-static2(default)(asr9k-11,asr9k-12)
7.67.136.0            7.67.136.255    7.67.136.0/24    1:N/A                  S
srg-9k-static2(default)(asr9k-11,asr9k-12)
7.67.137.0            7.67.137.255    7.67.137.0/24    1:N/A                  S
srg-9k-static2(default)(asr9k-11,asr9k-12)
7.67.138.0            7.67.138.255    7.67.138.0/24    1:N/A                  S
srg-9k-static2(default)(asr9k-11,asr9k-12)
7.67.139.0            7.67.139.255    7.67.139.0/24    1:N/A                  S
srg-9k-static2(default)(asr9k-11,asr9k-12)
7.67.140.0            7.67.140.255    7.67.140.0/24    1:N/A                  S
srg-9k-static2(default)(asr9k-11,asr9k-12)
7.67.141.0            7.67.141.255    7.67.141.0/24    1:N/A                  S
srg-9k-static2(default)(asr9k-11,asr9k-12)
```

```
7.67.142.0            7.67.142.255    7.67.142.0/24    1:N/A                  S
srg-9k-static2(default)(asr9k-11,asr9k-12)
33.0.0.0              33.0.7.255      33.0.0.0/21      1:0/-1   0.20%
automation-poolv4(default)(asr9k-11,asr9k-12)
=============================================================================================================
```



CHAPTER 25

CnBNG Control Plane Quality of Service

Quality of Service (QoS) is a technique for prioritizing network traffic. It ensures that time-sensitive and mission-critical applications receive sufficient bandwidth, low latency, and minimal data loss. QoS forwards high-priority packets by identifying, classifying, and prioritizing them throughout the network.

To achieve these QoS objectives, the cnBNG utilizes Differentiated Services Code Point (DSCP) marking as the primary mechanism for traffic prioritization.

Understanding DSCP marking

Differentiated Services Code Point (DSCP) marking is a field in the IP header used to classify and manage network traffic. By assigning a specific DSCP value to a packet, the cnBNG Control Plane (CP) tags the packet so routers along the data-forwarding path prioritize it. This ensures that traffic receives preferential treatment, such as lower latency and reduced packet loss, during periods of network congestion.

This chapter details the deployment of QoS for the cnBNG CP.

- DSCP marking for replication traffic (CP-CP): This feature prioritizes inter-instance state-synchronization data. By applying consistent DSCP values, the cnBNG CP prevents synchronization delays between redundant CP clusters. This process maintains geo-redundancy and failover readiness.
- DSCP marking for CP-UP control packets (CP-UP): This feature provides granular, message-level classification for PFCP and GTP-U signaling. It allows the cnBNG CP to distinguish between high-priority node-level operations (for example associations) and standard session-level updates, preventing signaling failures during congestion.
- DSCP marking for RADIUS traffic: This feature enables QoS prioritization for AAA signaling exchanged between the cnBNG CP and external RADIUS servers. By prioritizing authentication and accounting messages, the system ensures reliable subscriber lifecycle management even during high network load.
- CP-UP liveliness optimization: This feature complements QoS by replacing static PFCP heartbeat intervals with user-configurable, dynamic timers. By employing an adaptive backoff algorithm, the cnBNG CP accelerates link failure detection and reduces overall system convergence time. This approach minimizes subscriber session impact during CP-UP link failures.

- Feature Summary and Revision History, on page 582
- DSCP marking for replication traffic, on page 582
- DSCP marking for CP-UP control packets, on page 585
- DSCP marking for RADIUS traffic, on page 588
- CP-UP liveliness optimization, on page 589

Feature Summary and Revision History

Summary Data

Table 125: Summary Data

Applicable Products or Functional Area	cnBNG
Applicable Platform(s)	SMI
Feature Default Setting	Disabled - Configuration Required
Related Changes in this Release	Not Applicable
Related Documentation	Cloud Native BNG Control Plane Command Reference Guide

Revision History

Table 126: Revision History

Revision Details	Release
First introduced.	2026.03.0

DSCP marking for replication traffic

The DSCP marking for replication traffic is a state-synchronization feature that

- applies configurable DSCP values to inter-instance state data
- ensures geo-redundancy and failover readiness by protecting synchronization messages from congestion, and
- prevents synchronization delays between CP clusters to maintain state consistency.

Table 127: Feature History Table

Feature Name	Release Information	Feature Description
DSCP marking for replication traffic	2026.03.0	Enables DSCP marking for internal state-synchronization traffic between redundant control plane clusters to ensure deterministic synchronization. By applying consistent DSCP values, the cnBNG protects critical replication flows from congestion and eliminates data lag, facilitating faster and more reliable failover convergence for geo-redundancy.

Inter-instance state data

Inter-instance state data is the critical operational information that must be synchronized between redundant CP instances to ensure the system functions as a single, cohesive entity. In a distributed cnBNG architecture, this data is the source of truth that allows the system to maintain high availability and seamless failover capabilities.

This data is essential for maintaining consistency across distributed clusters and typically includes:

- Cluster state (etcd): Information regarding the overall health and configuration of the CP cluster, including SRG caches and UP manager data.
- Cache synchronization: Data related to IP Address Management (IPAM) and system uptime, ensuring all CP instances have an identical view of network resources.

Need for DSCP marking for replication traffic

To ensure reliable failover and geo-redundancy, CP instances require real-time state synchronization. Currently, this critical traffic competes for bandwidth within the data center network, making it susceptible to congestion-related delays. Without traffic prioritization, these delays can compromise system stability and increase the risk of state inconsistency during failover events.

The DSCP marking for replication traffic feature mitigates these risks by implementing a deterministic QoS policy for internal traffic flows. By assigning a specific DSCP value to state-synchronization data at the geo-replication endpoint, the system ensures that this traffic receives preferential treatment across the network infrastructure.

Benefits of DSCP marking for replication traffic

- Deterministic sync: Guarantees the timely delivery of state information even during periods of high network load, ensuring that synchronization data is forwarded ahead of non-critical flows.
- Failover readiness: Ensures that redundant CP nodes remain perfectly synchronized, eliminating data lag and facilitating faster, more reliable failover convergence.
- Congestion resilience: Protects critical administrative and control messages from being dropped during congestion, minimizing queuing delays and preventing packet loss.

- Reduced latency and jitter: Ensures that cluster state, cache information, and geo-redundancy data reach their destination with minimal jitter and maximum reliability.

Limitations of DSCP marking for replication traffic

DSCP marking for CDL is not supported.

How DSCP marking for replication traffic works

Summary

The geo-replication endpoint and the data center network coordinate to prioritize internal state-synchronization data. This process ensures the high-priority delivery of critical replication flows, such as `etcd` and `cache-pod` synchronization, to maintain cluster state consistency during periods of network congestion.

Workflow

These are the stages for DSCP marking for replication traffic.

1. Traffic classification: The cnBNG CP applies the configured DSCP value to all outgoing replication traffic at the geo-replication endpoint.
2. Header marking: The geo-replication endpoint embeds the DSCP marking into the IP header of the replication packets.
3. Preferential forwarding: The data center network identifies the marked packets and applies preferential forwarding treatment based on the assigned DSCP value.
4. Reliable delivery: The replication flows reach the destination CP instance with reduced latency and minimal packet loss.

Configure DSCP marking for replication traffic

Configure DSCP marking for replication traffic to ensure the prioritized delivery of inter-instance state synchronization data.

Note You must apply this configuration to all redundant cnBNG CP clusters before they are deployed, as this is a static setting that cannot be modified on the fly.

Procedure

Step 1 Set the DSCP value for the geo-replication endpoint.

Ensure that all instances on each cluster must have the same DSCP value.

Example:

```
instance instance-id 1
 endpoint geo
```

```
  dscp 63
 end
```

Step 2 Verify the configuration using the show running configuration.

Example:

```
cnbng# show running-config instance instance-id 1 endpoint geo
instance instance-id 1
 endpoint geo
  replicas 1
  dscp 63
  interface geo-internal
   vip-ip 10.1.35.1 vip-port 7001
  exit
  interface geo-external
   vip-ip 10.1.35.1 vip-port 7002
  exit
 exit
exit
```

Step 3 Verify that the traffic packets contain the configured DSCP value for replication traffic.

Example:

```
Frame 3: Packet, 100 bytes on wire (800 bits), 100 bytes captured (800 bits)
Ethernet II, Src: Intel_18:99:f8 (40:a6:b7:18:99:f8), Dst: Cisco_b2:2d:c5 (04:76:b0:b2:2d:c5)
Internet Protocol Version 4, Src: 10.1.35.1, Dst: 10.2.35.1
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
    Differentiated Services Field: 0xfc (DSCP: Unknown, ECN: Not-ECT)
        1111 11.. = Differentiated Services Codepoint: Unknown (63)
        .... ..00 = Explicit Congestion Notification: Not ECN-Capable Transport (0)
    Total Length: 86
```

DSCP marking for CP-UP control packets

The CP-UP control traffic prioritization is a granular QoS feature that

- enables message-level DSCP marking for PFCP and GTP-U control signaling
- distinguishes between critical node-level operations and standard session-level updates, and
- ensures reliable signaling between the Control Plane (CP) and User Plane (UP) to prevent session instability.

Table 128: Feature History Table

Feature Name	Release Information	Feature Description
DSCP marking for CP-UP Control Packets	2026.03.0	You can improve CP-UP signaling reliability during congestion by enabling granular, message-level QoS classification for N4 signaling protocols, such as PFCP and GTP-U, to prioritize critical node-level operations over session-level updates.

Need for DSCP marking for CP-UP control packets

Existing cnBNG devices treat all CP-UP signaling traffic equally. Critical node-level operations, such as associations and heartbeat exchanges, are essential for maintaining system health, yet they currently compete for bandwidth with standard session-level updates. During periods of network congestion, this lack of traffic differentiation can lead to packet loss, signaling failures, and subscriber session instability.

The DSCP marking for CP-UP control packets feature introduces granular QoS capabilities to the N4 interface. By enabling message-level DSCP classification for PFCP and GTP-U control protocols, CP can distinguish between high-priority node-level operations and standard session-level updates. This feature allows operators to apply specific DSCP tags to different message types, ensuring that vital signaling packets receive preferential forwarding treatment across the data center network.

Benefits of DSCP marking for CP-UP control packets

- Message-Level granularity: Enables operators to assign higher priority to critical node-level messages compared to standard session-level updates.
- Signaling reliability: Reduces CP packet loss, ensuring that essential association and heartbeat messages are delivered reliably even during periods of network congestion.

How DSCP marking for CP-UP control packets works

Summary

The CP and UP utilize the N4 interface to classify and prioritize signaling traffic. This process ensures that critical node-level operations, such as associations and heartbeats, receive higher priority than standard session-level updates to maintain system stability

Workflow

These are the stages of DSCP marking for CP-UP control packets.

1. The cnBNG CP classifies outgoing signaling packets based on their message type, for example node-level or session-level, at the control plane.
2. The CP applies the corresponding DSCP marking to the classified signaling packets before transmission.
3. The N4 interface transmits the prioritized packets across the network to the UP.

4. The UP processes the incoming packets, and response packets inherit the DSCP marking from the original request to ensure consistent priority treatment.

Configure DSCP marking for CP-UP control packets

Set DSCP values for N4 protocol endpoints to prioritize CP-UP control packet traffic across clusters.

DSCP marking enables network devices to identify and prioritize CP-UP control traffic. You can configure DSCP values globally or for specific message types and must apply the configuration consistently across all instances in a cluster.

Procedure

Step 1 On each cluster instance, set the DSCP value for the N4 protocol endpoint.

You must configure every instance with the same configuration.

Example:

```
instance instance-id 1
 endpoint n4-protocol
  dscp node-association af12
 end
```

Step 2 Verify the DSCP marking for CP-UP control packets configuration.

Example:

```
cnbng# show running-config instance instance-id endpoint n4-protocol
instance instance-id 1
 endpoint n4-protocol
  replicas 2
  dscp node-association af11
  dscp session-message af21
  dscp srg-message    af12
  dscp reconcile-message af13
  dscp gtpu-control-packet af43
  retransmission max-retry 1
 exit
exit
instance instance-id 2
 endpoint n4-protocol
  replicas 2
  dscp node-association af11
  dscp session-message af21
  dscp srg-message    af12
  dscp reconcile-message af13
  dscp gtpu-control-packet af43
  retransmission max-retry 1
 exit
exit
```

DSCP marking for RADIUS traffic

The DSCP marking for RADIUS traffic is a QoS feature that

- enables configurable DSCP marking for RADIUS protocol traffic
- prioritizes AAA signaling and CoA responses between the cnBNG CP and external AAA servers, and
- ensures reliable signaling and subscriber session stability during network congestion.

Table 129: Feature History Table

Feature Name	Release Information	Feature Description
DSCP marking for RADIUS traffic	2026.03.0	Enables configurable DSCP marking for RADIUS protocol traffic to prioritize AAA signaling and ensure subscriber session stability during network congestion.

Benefits of RADIUS DSCP marking

- Signaling reliability: Protects time-sensitive RADIUS signaling from being dropped or delayed during periods of high network congestion.
- Enhanced session stability: Minimizes authentication timeouts and signaling failures, ensuring that subscriber session establishment and lifecycle management remain consistent.
- Deterministic performance: Guarantees that AAA signaling packets receive preferential forwarding treatment across the data center network, reducing jitter and latency for critical control-plane traffic.
- Operational determinism: Provides a consistent QoS policy for RADIUS interactions, allowing network administrators to maintain predictable control-plane behavior in large-scale deployments.

How RADIUS DSCP marking works

Summary

The cnBNG CP classifies and marks outgoing RADIUS signaling traffic to ensure it receives preferential treatment across the data center network. This process ensures that critical authentication and accounting messages are prioritized, maintaining subscriber session stability during periods of network congestion.

Workflow

These are the stages of RADIUS DSCP marking.

1. Policy configuration: Define a traffic priority level within the RADIUS endpoint configuration.
2. Traffic classification: The cnBNG CP identifies all outgoing RADIUS signaling packets associated with the configured instance.

3. Header marking: The cnBNG CP embeds the configured DSCP value into the IP header of each RADIUS packet at the egress point.
4. Preferential forwarding: The data center network identifies the marked packets and applies preferential forwarding treatment based on the assigned DSCP value.
5. Reliable delivery: The RADIUS signaling packets reach the destination aaa server with reduced latency and minimal packet loss.

Configure RADIUS DSCP marking

Use this task to assign a specific DSCP value to RADIUS signaling traffic, ensuring preferential forwarding treatment for authentication, authorization, and accounting messages across the data center network.

Procedure

Configure the DSCP value for RADIUS traffic.

Apply the same configuration on all instances of each cluster.

Example:

```
instance instance-id 1
 endpoint radius
  dscp 20
  exit
 exit
```

CP-UP liveliness optimization

The CP-UP liveliness optimization is a failure-detection cnBNG feature that

- replaces static Packet Forwarding Control Protocol (PFCP) heartbeat intervals with user-configurable, dynamic timers
- employs an adaptive backoff algorithm to accelerate link failure detection, and
- reduces overall system convergence time to minimize subscriber session instability during CP-UP link failures.

Table 130: Feature History Table

Feature Name	Release Information	Feature Description
CP-UP liveliness optimization	2026.03.0	You can now reduce CP-UP failure detection time and improve SRG convergence by optimizing CP-UP liveliness detection with configurable heartbeat timers and an adaptive backoff mechanism.

Key terms

- Packet Forwarding Control Protocol (PFCP): The control plane protocol used between the CP and UP to manage sessions and exchange heartbeat messages for liveliness detection.
- Liveliness detection: The process of verifying the operational status of a network link or connection by periodically exchanging keep-alive or heartbeat messages.
- Convergence time: The duration required for the network to detect a failure and successfully switch traffic to a redundant path; reducing this is a primary goal of this feature.
- Heartbeat interval: The configurable time period between consecutive heartbeat messages sent between the CP and UP.

Benefits of CP-UP liveliness optimization

- Faster failure detection: Reduces the detection time from a fixed 90 seconds to a configurable range, with a minimum of 10 seconds for a heartbeat configuration using a five-second interval and a miss-count of two.
- Configurable convergence: Provides the flexibility to tune heartbeat intervals (5 to 60 seconds) to meet specific SRG convergence targets.
- Adaptive efficiency: Automatically reduces heartbeat intervals during failure scenarios to speed up detection, while maintaining a stable, bounded probing rate during healthy operation.
- Reduced subscriber impact: Minimizes the duration of service disruption during CP-UP link failures by triggering failover mechanisms more rapidly.

How CP-UP liveliness optimization works

Summary

The CP and UP coordinate through an adaptive heartbeat mechanism to monitor the health of their connection. This mechanism enables rapid failure detection and improves SRG convergence by dynamically adjusting heartbeat intervals based on the CP-UP link status

Workflow

These stages describe how CP-UP liveliness optimization works.

1. The UP sends heartbeat messages at a user-configured interval, which ranges from 5 to 60 seconds, during normal operation.
2. The CP receives and responds to these heartbeat messages to confirm link health.
3. The CP detects a missed heartbeat request, indicating a potential link failure.
4. Upon each consecutive missed heartbeat, CP progressively reduces the heartbeat interval to accelerate failure detection.
5. When a successful heartbeat request is received, CP immediately restores the heartbeat interval to the configured value.
6. After the configured number of consecutive heartbeat failures, the CP marks the CP-UP connection as inactive and triggers failover procedures.

Configure the CP-UP heartbeat timer

Use this task to configure the CP-UP heartbeat timer for liveliness detection between the CP and UP.

Procedure

Set the heartbeat interval.

You can configure the heartbeat either globally or for a specific user plane. Configuration for a user plane takes precedence over the global configuration.

The heartbeat configuration must be identical on both CP and UP instances to ensure synchronization.

Example:

This example shows how to configure the heart beat at the global level.

```
user-plane
 pfcp-heartbeat interval 10 miss-count 4
 exit
```

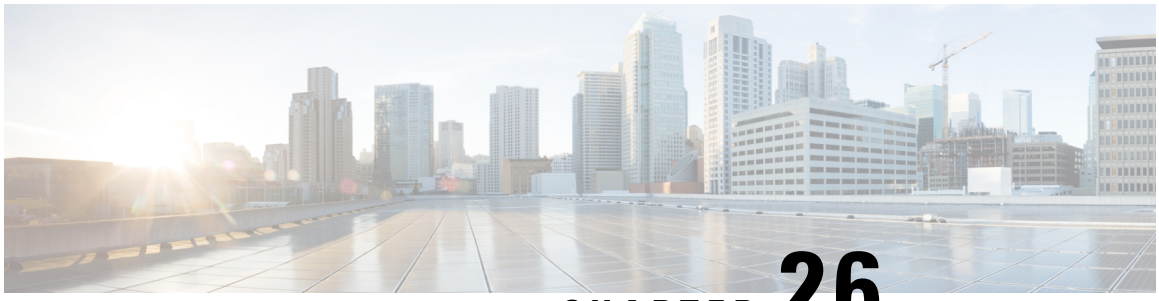
This example shows how to configure the heart beat for the specific user plane configuration

```
user-plane
 instance 1
  user-plane asr9k-0-1
   pfcp-heartbeat interval 10 miss-count 4
  exit
  user-plane asr9k-0-2
   pfcp-heartbeat interval 10 miss-count 4
  exit
 exit
exit
```

Example:

Note If no values are configured, the cnBNG CP applies a default interval of 30 seconds and a miss-count of 3.

When the interval is set to 30 seconds and the miss-count is 3, either by configuration or default, the heartbeat interval is not reduced and remains fixed at 30 seconds. This ensures backward compatibility with existing UPs.



CHAPTER 26

Lawful Intercept for cnBNG

- Lawful Intercept for cnBNG, on page 593
- How cnBNG LI works, on page 594
- Configure LI for cnBNG subscriber session, on page 596
- Redundancy Handling, on page 596
- Call flow for LI provisioning, on page 597
- Packet format, on page 598

Lawful Intercept for cnBNG

Lawful intercept (LI) is a surveillance framework that enables law enforcement agencies (LEAs) to conduct electronic surveillance as authorized by a judicial or administrative order.

The cnBNG LI feature is a lawful intercept capability that

- identifies the target subscriber session
- specifies the mediation device destination to which the target subscriber's traffic is copied, and
- ensures that duplicated subscriber traffic is sent to the mediation device without changing the subscriber's normal forwarding behavior.

Table 131: Feature History Table

Feature Name	Release Information	Feature Description
Lawful Intercept for cnBNG	Release 2021.01.0	Enables service providers to meet lawful intercept compliance requirements while maintaining normal subscriber traffic forwarding.RADIUS-driven support identifies target PPPoE and IPoE subscriber sessions and replicates their traffic to an authorized mediation device in the default or a nondefault VRF.

cnBNG supports LI as a RADIUS-driven subscriber control function for both PPPoE and IPoE subscribers. The LI workflow is initiated and controlled exclusively through AAA/RADIUS signaling, and is not supported through cnBNG-local configuration or northbound management APIs.

RADIUS-driven LI for cnBNG subscriber sessions

Service providers require a secure, automated method to replicate traffic from specific PPPoE or IPoE subscriber sessions to an authorized Mediation Device (MD) for lawful interception.

The challenge without native LI integration:

- The network cannot dynamically identify or target subscriber sessions via AAA.
- Operators must rely on manual, device-specific traffic mirroring, which is operationally complex and lacks standardized intercept identifiers.
- Interception cannot be centrally managed; the RADIUS server cannot dynamically start or stop monitoring for active sessions.
- The user plane lacks the necessary MD destination and LI parameters to facilitate automated traffic delivery.

cnBNG supports RADIUS-based Lawful Intercept, enabling centralized control over traffic replication. By utilizing `Access-Accept` or `CoA-Request` packets, the RADIUS server dynamically pushes LI parameters—including MD destination, port, and intercept identifiers—directly to the network access server (NAS) or L2TP access concentrator (LAC).

How cnBNG LI works

Summary

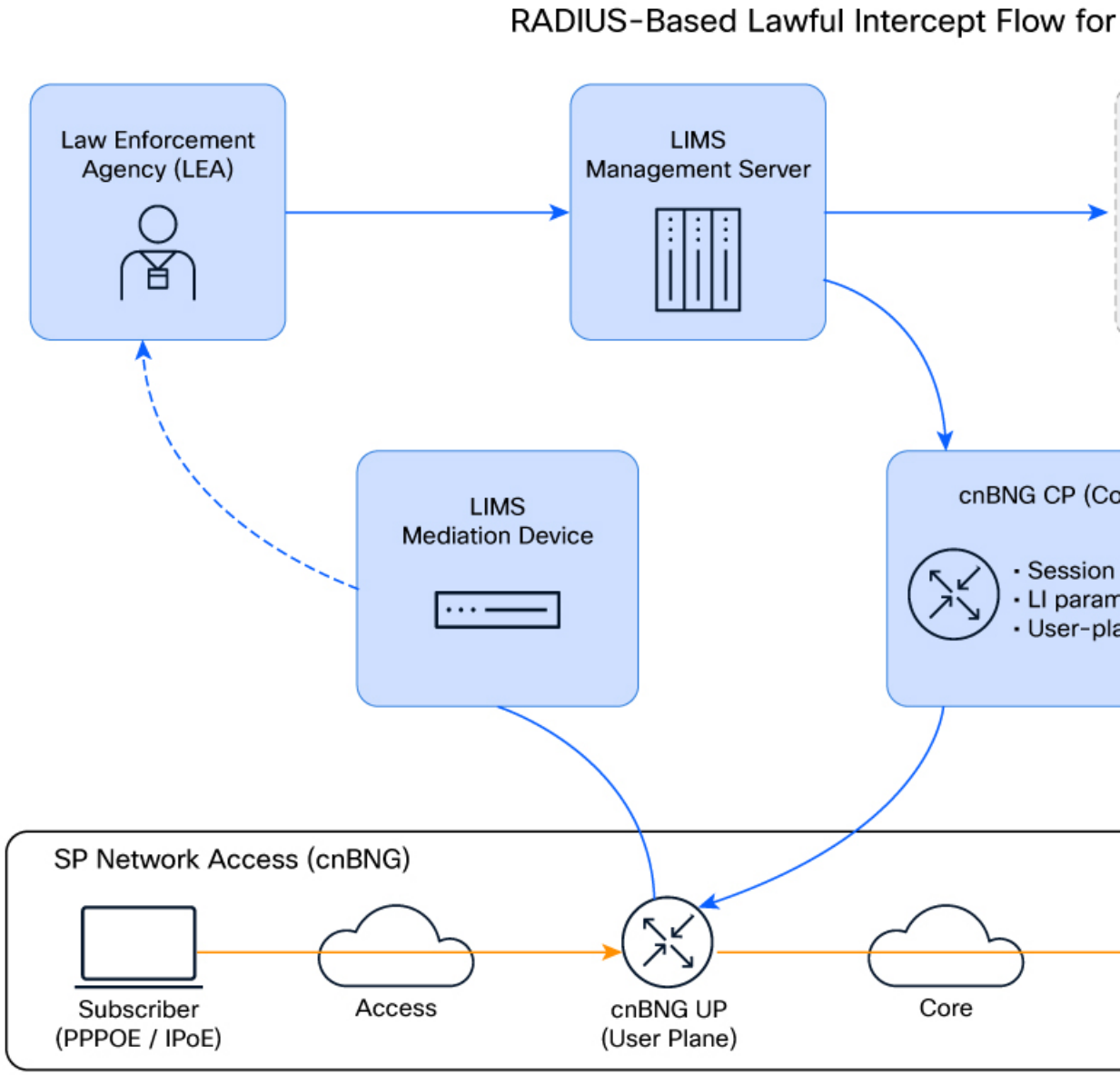
The key components involved in the cnBNG LI process are:

- RADIUS server: Provides LI Cisco VSA attributes via Access-Accept or CoA-Request messages.
- cnBNG: Acts as the LI control input; it parses LI parameters and transparently forwards them to the user-plane.
- cnBNG user-plane: Executes the actual subscriber traffic duplication and mediation-device delivery.
- Mediation device: Receives the duplicated subscriber traffic.

The process involves identifying the target subscriber session, conveying the mediation device destination, and ensuring the replicated copy is sent without altering the subscriber's normal forwarding behavior.

Workflow

Figure 42: Radius-Based LI flow



These stages describe how the cnBNG LI process works.

1. RADIUS attribute reception: The RADIUS server sends LI Cisco VSAs in either an Access-Accept packet (for new sessions) or a CoA-Request packet (for existing sessions).

   The cnBNG parses these attributes as LI service parameters.

   - md-ip-addr / md-ipv6-addr: Specifies the MD address.
   - md-port: Specifies the MD UDP destination port.
   - md-dscp: Specifies the DSCP value for replicated traffic.

- intercept-id: Identifies the intercepted subscriber traffic.
- li-action: Starts (1), stops (0), or validates (2) the intercept.

2. Subscriber context binding: The cnBNG binds the parsed LI parameters.
3. User-plane programmingThe cnBNG forwards the LI service parameters transparently to the cnBNG UP:
   - For new sessions, parameters are sent in the Session Establishment Request.
   - For existing sessions, parameters are sent in the Session Modification Request

   The duplicating parameters grouped information element (IE) conveys LI parameters within these messages, consisting of these attributes:
   - Destination interface
   - outer header creation
   - BNG LI info
   - BNG LI id

   Note: The cnBNG Duplicating Parameters IE is implemented in accordance with the ETSI 29.244 standard.
4. Traffic interception and duplicationThe user-plane receives the programming. Subscriber traffic is then intercepted and duplicated toward the mediation device.

Configure LI for cnBNG subscriber session

LI for cnBNG subscriber sessions is enabled by default and requires no local configuration. To initiate interception, ensure the following attributes are configured on the RADIUS server:

- md-ip-addr or md-ipv6-addr: Specifies the mediation device (MD) address.
- md-port: Specifies the MD UDP destination port.
- md-dscp: Specifies the DSCP value for replicated traffic.
- intercept-id: Identifies the intercepted subscriber traffic.
- li-action: Starts, stops, or validates the intercept.

Redundancy Handling

- SRG or UP Redundancy: During session establishment, the cnBNG propagates the duplicating parameters to both the active and standby user planes. This ensures that in the event of an SRG switchover, the standby UP is already provisioned with the correct mediation device details, allowing for seamless traffic replication.
- CP Redundancy: CP redundancy has no impact on LI operations, as the session state and LI parameters are synchronized or re-established as part of the standard CP failover process.

Call flow for LI provisioning

Figure 43: New subscriber session

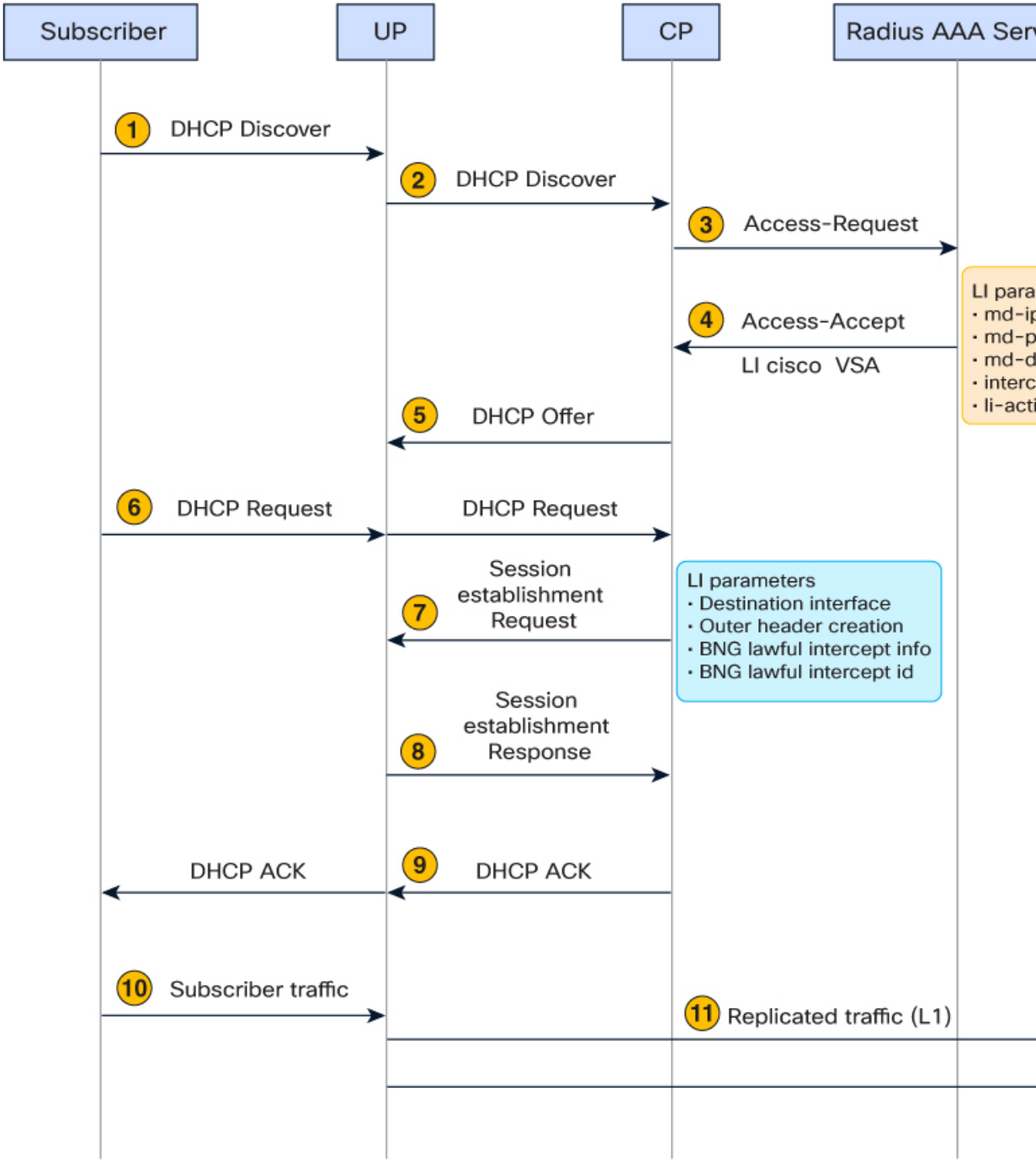
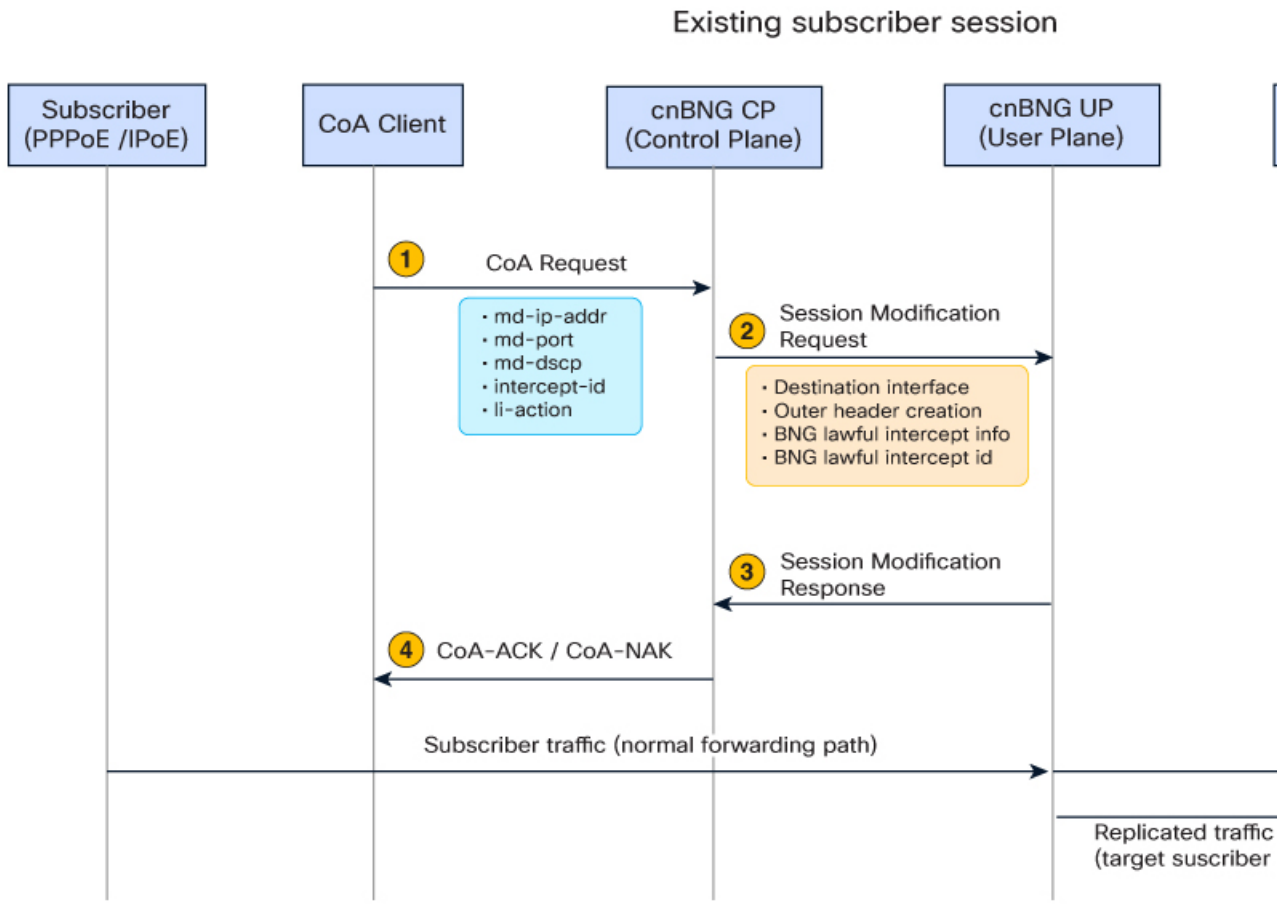


Figure 44: Existing subscriber session



Packet format

Access-Accept message with li Cisco AVP

```
 Subscriber Id: aa11.0000.0001@automation-userplane
      Message: RadiusUdpProxyMsg
      Description: Received Auth/Acct Response Message from UDP-Proxy
      Source: bng.udp-proxy.DC.Local.0
      Destination: bng.radius-ep.DC.Local.0

      PAYLOAD:
        RadiusUdpProxyMsg:
            RadiusUdpProxyMsg:
                SrcIp: 10.1.6.218
                SrcPort: 1812
                DestIp: 10.1.0.58
                DestPort: 16384
                Payload:

Code = AccessAccept
Id = 23
Authenticator = [143 103 48 233 240 179 78 238 166 222 143 157 215 13 172 152]
```

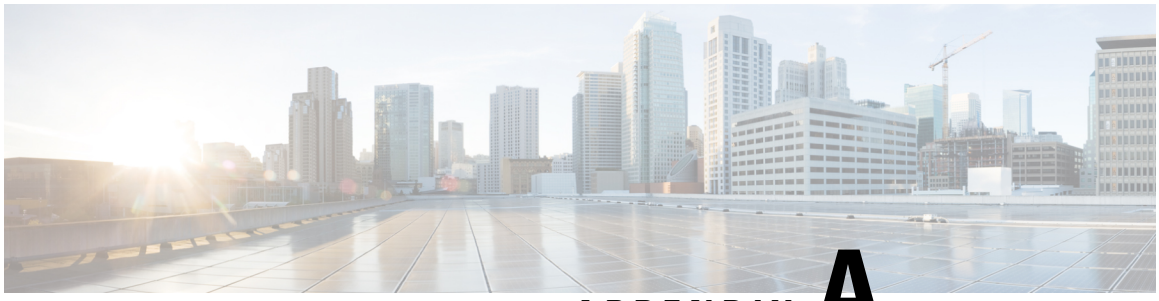
```
Cisco-Vsa_Cisco-AVpair_1 = "intercept-id=abcd3235"
Cisco-Vsa_Cisco-AVpair_1 = "md-ip-addr=20.0.0.xx"
Cisco-Vsa_Cisco-AVpair_1 = "md-port=4546"
Cisco-Vsa_Cisco-AVpair_1 = "md-dscp=7"
Cisco-Vsa_Cisco-AVpair_1 = "li-action=1"
               PayloadLen: 133
               SrcIpv6:
               DestIpv6:
               IsIpv6: false
```

Packet format of CoA-Request

```
Subscriber Id: aa11.0000.0001@automation-userplane-stby
      Timestamp: 2026/10/06 11:22:53.443569
      Message: RadiusUdpProxyCoaMsg
      Description: Received COA/DM Request Message from UDP-Proxy
      Source: bng.udp-proxy.DC.Local.0
      Destination: bng.radius-ep.DC.Local.0
      PAYLOAD:
        RadiusUdpProxyCoaMsg:
            RadiusUdpProxyCoaMsg:
               SrcIp: 10.1.6.218
               SrcPort: 55291
               DestIp: 10.1.0.58
               DestPort: 2000
               Payload:

Code = CoARequest
Id = 8
Authenticator = [197 92 21 13 169 188 0 98 55 89 118 223 25 23 60 84]
Acct-Session-Id = "01000007"
Cisco-Vsa_Cisco-AVpair_1 = "md-ip-addr=192.0.2.10"
Cisco-Vsa_Cisco-AVpair_1 = "li-action=1"
Cisco-Vsa_Cisco-AVpair_1 = "md-port=5000"
Cisco-Vsa_Cisco-AVpair_1 = "intercept-id=a0000312"

               PayloadLen: 127
               SrcIpv6:
               DestIpv6:
               IsIpv6: false
```



APPENDIX A

RADIUS Attributes

Remote Authentication Dial-In User Service (RADIUS) attributes are used to define specific authentication, authorization, and accounting (AAA) elements in a user profile, which is stored on the RADIUS daemon.

This appendix describes the following types of RADIUS attributes supported in Broadband Network Gateway (BNG):

- RADIUS IETF Attributes, on page 601
- RADIUS Vendor-Specific Attributes, on page 603
- RADIUS ADSL Attributes, on page 607
- RADIUS ASCEND Attributes, on page 607
- RADIUS Disconnect-Cause Attributes, on page 607

RADIUS IETF Attributes

IETF Attributes Versus VSAs

RADIUS Internet Engineering Task Force (IETF) attributes are the original set of 255 standard attributes that are used to communicate AAA information between a client and a server. Because IETF attributes are standard, the attribute data is predefined and well known; thus all clients and servers who exchange AAA information via IETF attributes must agree on attribute data such as the exact meaning of the attributes and the general bounds of the values for each attribute.

RADIUS vendor-specific attributes (VSAs) derived from one IETF attribute-vendor-specific (attribute 26). Attribute 26 allows a vendor to create an additional 255 attributes however they wish. That is, a vendor can create an attribute that does not match the data of any IETF attribute and encapsulate it behind attribute 26; thus, the newly created attribute is accepted if the user accepts attribute 26.

Table 132: Supported RADIUS IETF Attributes

Name	Value	Type
Acct-Delay-Time	integer	41
Acct-Input-Giga-Words	integer	52
Acct-Input-Octets	integer	42
Acct-Input-Packets	integer	47

Name	Value	Type
Acct-Interim-Interval	integer	85
Acct-Link-Count	integer	51
Acct-Output-Giga-Words	integer	53
Acct-Output-Octets	integer	43
Acct-Output-Packets	integer	48
Acct-Status-Type	integer	40
Acct-Terminate-Cause	integer	49
CHAP-Challenge	binary	40
CHAP-Password	binary	3
Delegated-IPv6-Prefix	binary	123
Dynamic-Author-Error-Cause	integer	101
Event-Timestamp	integer	55
Filter-Id	string	11
Framed-Interface-Id	binary	96
Framed-IP-Address	ipv4addr	8
Framed-IPv6-Route	string	99
Framed-Pool	string	88
Framed-Protocol	integer	7
Framed-Route	string	22
Nas-Identifier	string	32
NAS-IP-Address	ipv4addr	4
NAS-IPv6-Address	string	95
NAS-Port	integer	5
Reply-Message	binary	18
Service-Type	integer	6
Session-Timeout	integer	27
Stateful-IPv6-Address-Pool	binary	123
X-Ascend-Client-Primary-DNS	ipv4addr	135
X-Ascend-Client-Secondary-DNS	ipv4addr	136

Filter-Id

Filter-ID specifies the access control list (ACL) that is applied to the subscriber interface. The format of the Filter-Id attribute is as follows:

```
Filter-Id = <ACL-Name> <in | out>
```

Where, **in** and **out** indicate the direction of the ACL feature to be applied. **ACL in** is mapped to the input direction (IPv4 Ingress), and **ACL out** is mapped to the output direction (IPv4 Egress) of the CP-UP session programming interface. You can configure only one attribute per direction.

Session-Timeout

Session-Timeout sets the maximum number of seconds of service to be provided to the user before the session terminates. Session-Timeout attribute can be sent as part of CoA request, Access-Accept, or Access-Challenge messages.

You can enable session-timeout using the user-profile on a RADIUS server. For example:

```
user1 Cleartext-Password := "cisco"
  Session-timeout = 90'
```

Once the timer expires, the subscriber is removed from the server.

For session deletion due to session-timeout, the reason of disconnect can be observed as “Session-Timeout” in accounting messages. For example,

```
(5) Sent Access-Accept Id 7 from 10.1.35.10:1812 to 10.1.32.83:16384 length 0
(5)   Session-Timeout = 90

(8)   Acct-Terminate-Cause = Session-Timeout
(8)   Ascend-Disconnect-Cause = Session-Timeout
```

Verification:

You can verify the session-timeout configuration using the **show subscriber session detail** command:

```
show subscriber session detail
    "subcfgInfo": {
        "committedAttrs": {
          "attrs": {
            "accounting-list": "automation-aaaprofile",
            "acct-interval": "2000",
            "addr-pool": "automation-poolv4",
            "ipv4-mtu": "1400",
            "ppp-ipcp-reneg-ignore": "true",
            "ppp-ipv6cp-reneg-ignore": "true",
            "ppp-lcp-reneg-ignore": "true",
            "session-acct-enabled": "true",
            "session-timeout": "90",
            "vrf": "automation-vrf"
          }
        },
```

RADIUS Vendor-Specific Attributes

The Internet Engineering Task Force (IETF) draft standard specifies a method for communicating vendor-specific information between the network access server and the RADIUS server by using the vendor-specific attribute (attribute 26). Attribute 26 encapsulates vendor specific attributes, thereby, allowing vendors to support their own extended attributes otherwise not suitable for general use.

The Cisco RADIUS implementation supports one vendor-specific option using the format recommended in the specification. Cisco's vendor-ID is 9, and the supported option has vendor-type 1, which is named "cisco-avpair." The value is a string of this format:

```
protocol : attribute sep value *
```

"Protocol" is a value of the Cisco "protocol" attribute for a particular type of authorization; protocols that can be used include IP, IPX, VPDN, VOIP, SHELL, RSVP, SIP, AIRNET, OUTBOUND. "Attribute" and "value" are an appropriate attribute-value (AV) pair defined in the Cisco TACACS+ specification, and "sep" is "=" for mandatory attributes and "*" for optional attributes. This allows the full set of features available for TACACS+ authorization to also be used for RADIUS.

The following example shows how to cause a user logging in from a network access server to have immediate access to EXEC commands:

```
cisco-avpair= "shell:priv-lvl=15"
```

The following example shows how to configure avpair aaa attribute to enable IPv6 router advertisements from an IPv4 subscriber interface:

```
Cisco-avpair= "ipv6:start-ra-on-ipv6-enable=1"
```

Attribute 26 contains these three elements:

- Type
- Length
- String (also known as data)
  - Vendor-ID
  - Vendor-Type
  - Vendor-Length
  - Vendor-Data



Note It is up to the vendor to specify the format of their VSA. The Attribute-Specific field (also known as Vendor-Data) is dependent on the vendor's definition of that attribute.

Table 133: Supported Cisco Vendor-Specific RADIUS Attributes

Name	Value	Type	Present in AAA message type
accounting-list	string	1	Access-accept, CoA, Accounting-request
acct-input-gigawords-ipv4	integer	1	Accounting-request
acct-input-octets-ipv4	integer	1	Accounting-request
acct-input-packets-ipv4	integer	1	Accounting-request
acct-input-gigawords-ipv6	integer	1	Accounting-request
acct-input-octets-ipv6	integer	1	Accounting-request
acct-input-packets-ipv6	integer	1	Accounting-request

Name	Value	Type	Present in AAA message type
acct-output-gigawords-ipv4	integer	1	Accounting-request
acct-output-octets-ipv4	integer	1	Accounting-request
acct-output-packets-ipv4	integer	1	Accounting-request
acct-output-gigawords-ipv6	integer	1	Accounting-request
acct-output-octets-ipv6	integer	1	Accounting-request
acct-output-packets-ipv6	integer	1	Accounting-request
addrv6	string	1	Access-accept, Accounting-request
circuit-id-tag	string	1	Access-accept, Accounting-request
cisco-nas-port	string	2	Access-accept, Accounting-request
client-mac-address	string	1	Access-accept, Accounting-request
command	string	1	CoA
connect-progress	string	1	Accounting-request
delegated-ipv6-pool	string	1	Access-accept
dhcp-class	string	1	Access-accept
dhcp-client-id	string	1	Accounting-request
dhcp-vendor-class	string	1	Access-request, Accounting-request
disc-cause-ext	string	1	Accounting-request
disconnect-cause	string	1	Accounting-request
dual-stack-delay	integer	1	Access-accept
inacl	string	1	Access-accept
intercept-id	integer	1	Access-accept
ip-addresses	string	1	Access-request, Accounting-request
ip:route	string	1	Access-accept, CoA
ip:ipv6-route	string	1	Access-accept, CoA
ipv6_inacl	string	1	Access-accept, CoA
ipv6_outacl	string	1	Access-accept, CoA
ipv6-dns-servers-addr	string	1	Access-accept
ipv6-mtu	integer	1	Access-accept

Name	Value	Type	Present in AAA message type
ipv6-strict-rpf	integer	1	Access-accept
ipv6-unreachable	integer	1	Access-accept
l2tp-tunnel-password	string	1	Access-accept
md-dscp	integer	1	Access-accept
md-ip-addr	ipaddr	1	Access-accept
md-port	integer	1	Access-accept
outacl	string	1	Access-accept
parent-session-id	string	1	Accounting-request
pppoe_session_id	integer	1	Accounting-request
primary-dns	ipaddr	1	Access-accept
remote-id-tag	string	1	Access-request, Accounting-request
sa	string	1	Access-accept, CoA
sd	string	1	RADIUS CoA
secondary-dns	ipaddr	1	Access-accept
service-name	string	1	Accounting-request
source-ip	ipv4addr	1	Access-accept
Stateful-IPv6-Address-Pool	string	1	Access-accept
sub-pbr-policy-in	string	1	Access-accept, CoA
subscriber:sub-qos-policy-in	string	1	Access-accept, CoA
subscriber:sub-qos-policy-out	string	1	Access-accept, CoA
tunnel-tos-reflect	string	1	Access-accept
tunnel-tos-setting	string	1	Access-accept
tunnel-type	string	1	Access-accept
username	string	1	Access-request, Accounting-request
user-plane-ipv4-address	string	1	Access-request, Accounting-request
user-plane-ipv6-address	string	1	Access-request, Accounting-request
vpdn-group	string	1	Access-accept
vpn-vrf	string	1	Access-accept

Name	Value	Type	Present in AAA message type
vrf	string	1	Access-accept
vrf-id	string	1	Access-accept

Vendor-Specific Attributes for Account Operations

Table 134: Supported Vendor-Specific Attributes for Account Operations

RADIUS AVP	Value	Type	Action
subscriber:command=account-update	string	1	account update
subscriber:sa=<service-name>	string	1	service activate
subscriber:sd=<service-name>	string	1	service de-activate

RADIUS ADSL Attributes

Table 135: Supported RADIUS ADSL Attributes

Name	Value	Type
Agent-Circuit-Id	string	1
Agent-Remote-Id	string	2

RADIUS ASCEND Attributes

Table 136: Supported RADIUS Ascend Attributes

Name	Value	Type
Ascend-Client-Primary-DNS	ipv4addr	135
Ascend-Client-Secondary-DNS	ipv4addr	136
Ascend-Connection-Progress	integer	196
Ascend-Disconnect-Cause	integer	195

RADIUS Disconnect-Cause Attributes

Disconnect-cause attribute values specify the reason a connection was taken offline. The attribute values are sent in Accounting request packets. These values are sent at the end of a session, even if the session fails to

be authenticated. If the session is not authenticated, the attribute can cause stop records to be generated without first generating start records.

lists the cause codes, values, and descriptions for the Disconnect-Cause (195) attribute.

Note The Disconnect-Cause is incremented by 1000 when it is used in RADIUS AVPairs; for example, disc-cause 4 becomes 1004.

Table 137: Supported Disconnect-Cause Attributes

Cause Code	Value	Description
2	Unknown	Reason unknown.
3	Call-Disconnect	The call has been disconnected.
11	Lost-Carrier	Loss of carrier.
21	Idle-Timeout	Timeout waiting for user input. **Note** Codes 21, 100, 101, 102, and 120 apply to all session types.
28	EXEC-Process-Destroyed	EXEC process destroyed.
33	Insufficient-Resources	Insufficient resources.
40	Timeout-PPP-LCP	PPP LCP negotiation timed out. **Note** Codes 40 through 49 apply to PPP sessions.
41	Failed-PPP-LCP-Negotiation	PPP LCP negotiation failed.
42	Failed-PPP-PAP-Auth-Fail	PPP PAP authentication failed.
45	PPP-Remote-Terminate	PPP received a Terminate Request from remote end.
47	NCP-Closed-PPP	PPP session closed because there were no NCPs open.
52	Invalid-IP-Address	IP address is not valid for Telnet host.
100	Session-Timeout	Session timed out.
150	RADIUS-Disconnect	Disconnected by RADIUS request.
151	Local-Admin-Disconnect	Administrative disconnect.
170	PPP-Authentication-Timeout	PPP authentication timed out.