Ultra Cloud Core 5G Policy Control Function, Release 2022.02 - Configuration and Administration Guide

First Published: 2022-04-29

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

CONTENTS

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 *Ultra Cloud Core 5G Policy Control Function Configuration and Administration Guide*, the document conventions, and the customer support details.

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.

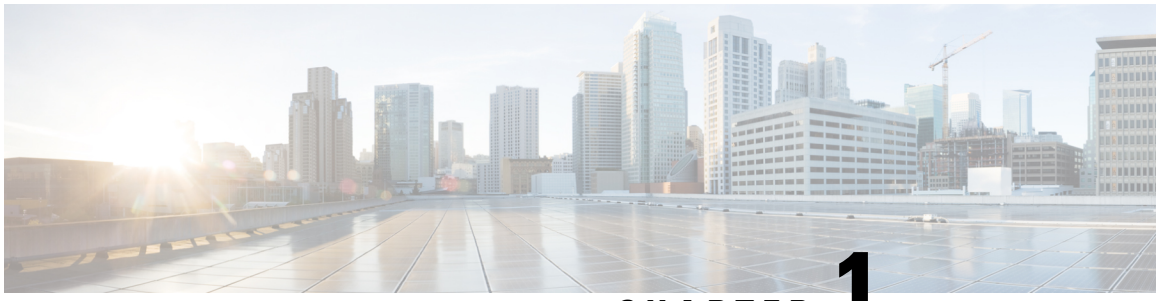
Typeface Conventions	Description
Text represented as a `screen display`	This typeface represents displays that appear on your terminal screen, for example: `Login:`

Typeface Conventions	Description
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 applicable 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**

Contacting Customer Support

Use the information in this section to contact customer support.

Refer to the support area of http://www.cisco.com for up-to-date product documentation or to submit a service request. A valid username and password are required to access this site. Please contact your Cisco sales or service representative for additional information.



CHAPTER 1

5G Architecture

Feature Summary and Revision History

Summary Data

Table 1: Summary Data

Applicable Product(s) or Functional Area	• PCF • SMF • UPF
Applicable Platform(s)	SMI
Feature Default Setting	Not Applicable
Related Documentation	Not Applicable

Revision History

Table 2: Revision History

Revision Details	Release
First introduced.	Pre-2020.02.0

Overview

The Ultra Cloud Core is Cisco's solution supporting 3GPP's standards for 5G new radio (NR) standalone (SA) mode. These standards define various network functions (NFs) based on the separation of control plane (CP) and user plane (UP) (for example CUPS) functionality for increased network performance and capabilities.

Control Plane Network Functions

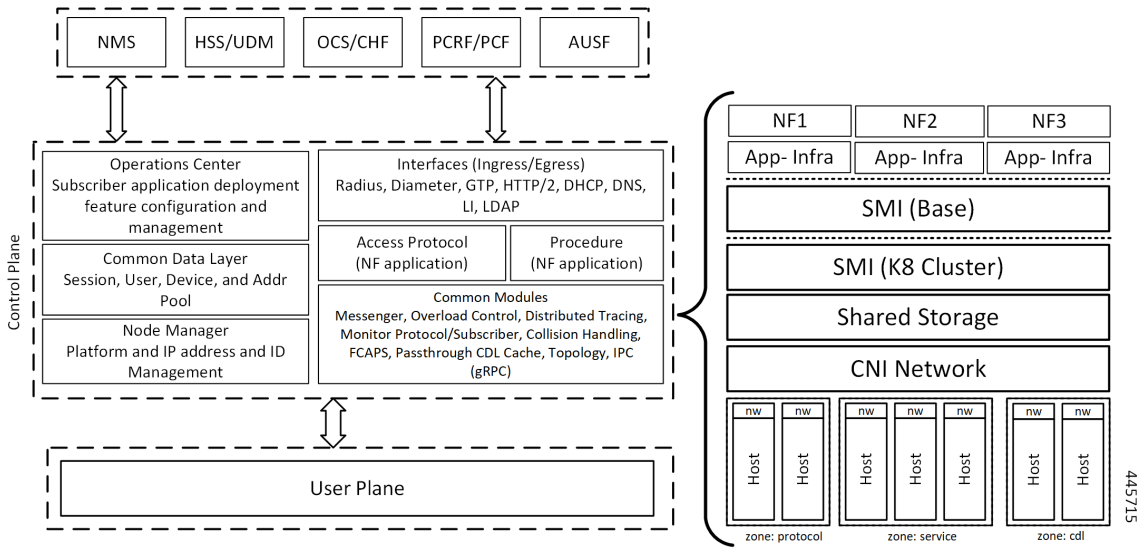
The CP-related NFs that comprise the Ultra Cloud Core are based on a common architecture that is designed around the following tenants:

- Cloud-scale—Fully virtualized for simplicity, speed, and flexibility.
- Automation and orchestration—Optimized operations, service creation, and infrastructure.
- Security—Multiple layers of security across the deployment stack from the infrastructure through the NF applications.
- API exposure—Open and extensive for greater visibility, control, and service enablement.
- Access agnostic—Support for heterogeneous network types (for example 5G, 4G, 3G, Wi-Fi, and so on).

These control plane NFs are each designed as containerized applications (for example microservices) for deployment through the Subscriber Microservices Infrastructure (SMI).

The SMI defines the common application layers for functional aspects of the NF such as life-cycle management (LCM), operations and management (OAM), and packaging.

Figure 1: Ultra Cloud Core CP Architectural Components



User Plane Network Function

The 5G UP NF within the Ultra Cloud Core is the User Plane Function (UPF). Unlike the CP-related NFs, the 5G UPF leverages the same Vector Packet Processing (VPP) technology currently in use by the user plane component within Cisco 4G CUPS architecture. This commonality ensures the delivery of a consistent set of capabilities between 4G and 5G such as:

- Ultrafast packet forwarding.
- Extensive integrated IP Services such as Subscriber Firewall, Tethering, Deep-Packet Inspection (DPI), Internet Content Adaption Protocol (ICAP), Application Detection and Control (ADC), and header enrichment (HE).
- Integrated third-party applications for traffic and TCP optimization.

For more information on UPF, see *Ultra Cloud Core 5G UPF Configuration and Administration Guide*.

Subscriber Microservices Infrastructure Architecture

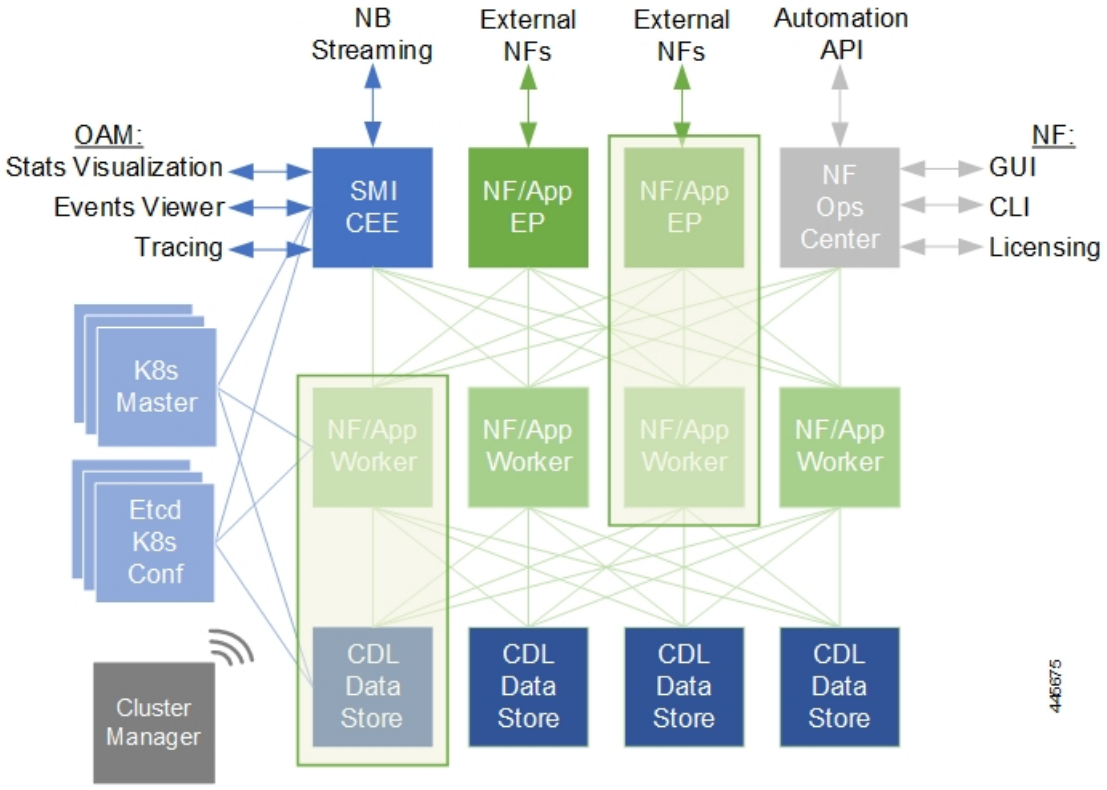
The Ultra Cloud Core Subscriber Microservices Infrastructure (SMI) is a layered stack of cloud technologies that enable the rapid deployment of, and seamless life-cycle operations for microservices-based applications.

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 primary and etcd functions, which provide LCM for the NF applications that are 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. Also, 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 high availability in local or geo-redundant deployments.
- Service Mesh—Provides sophisticated message routing between application containers, enabling managed interconnectivity, extra 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)—Provides various APIs for deployment, configuration, and management automation.

The following figure depicts how these components interconnect to comprise a microservice-based NF/application.

Figure 2: SMI Components



For more information on SMI components, see Ultra Cloud Core Subscriber Microservices Infrastructure documentation— *Deployment Guide* > *Overview* chapter.

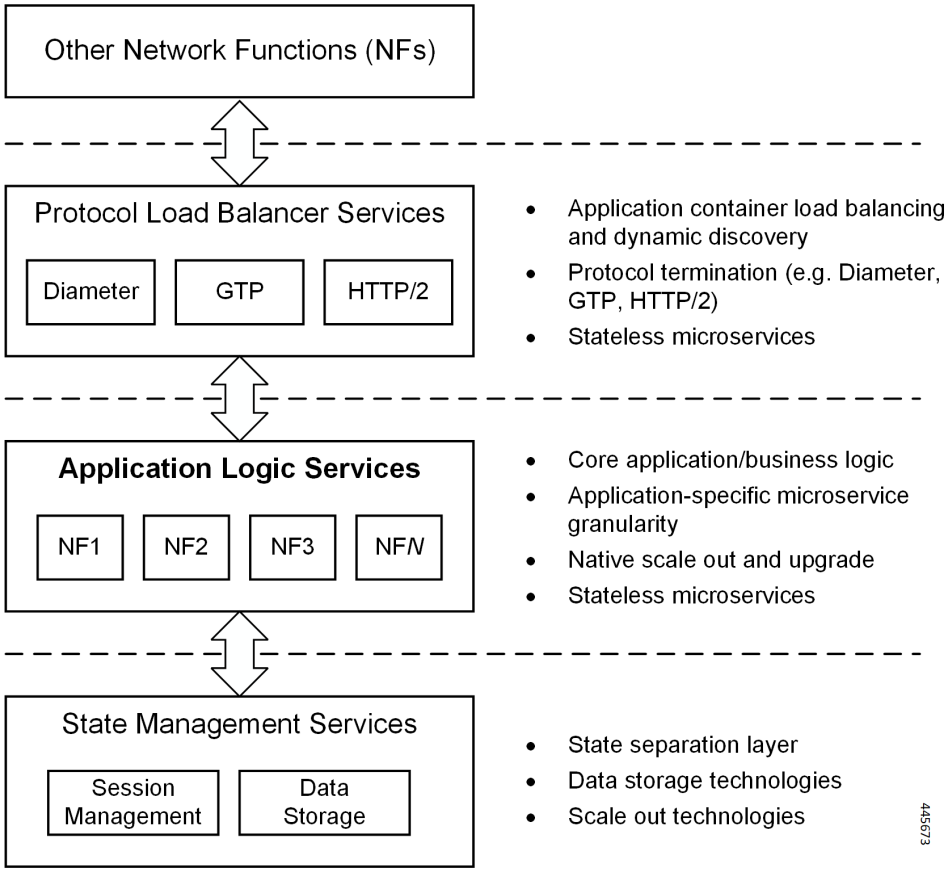
Control Plane Network Function Architecture

Control plane (CP) NFs are designed around a three-tiered architecture that take advantage of the stateful or stateless capabilities that are afforded within cloud native environments.

The architectural tiers are as follows:

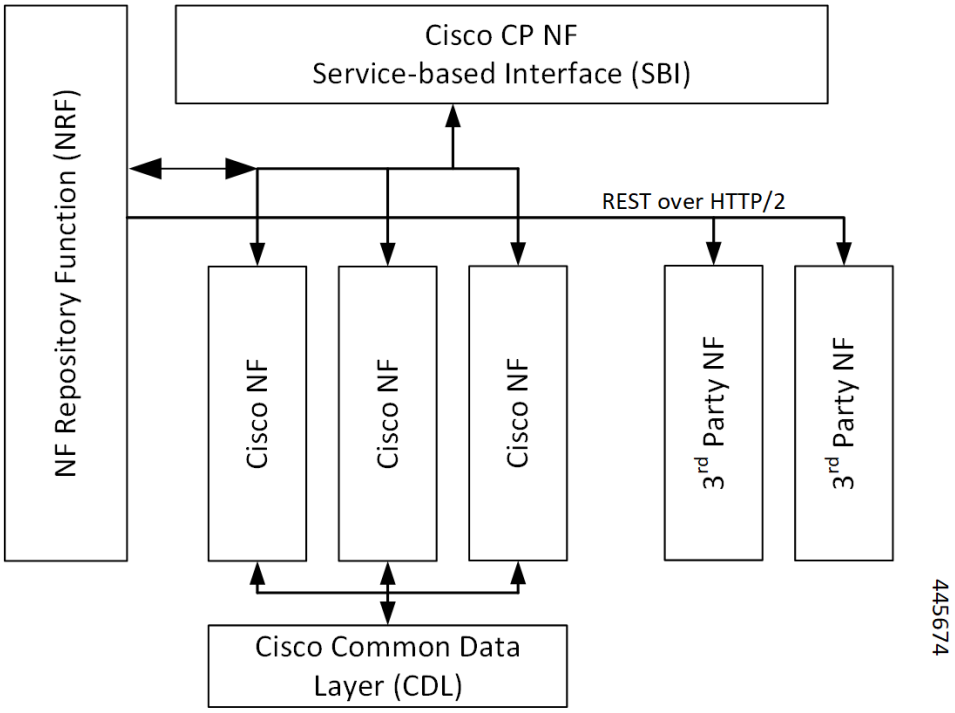
- Protocol Load Balancer Services—These are stateless microservices that are primarily responsible for dynamic discovery of application containers as well as for protocol proxy and termination. These include traditional 3GPP protocols and new protocols that are introduced with 5G.
- Applications Services—Responsible for implementing the core application or business logic, these are the stateless services that render the actual application based on the received information. This layer may contain varying degrees of microservice granularity. Application services are stateless.
- State management services—Enable stateless application services by providing a common data layer (CDL) to store or cache state information (for example session and subscriber data). This layer supports various data storage technologies from in-memory caches to full-fledge databases.

Figure 3: Control Plan Network Function Tiered Architecture

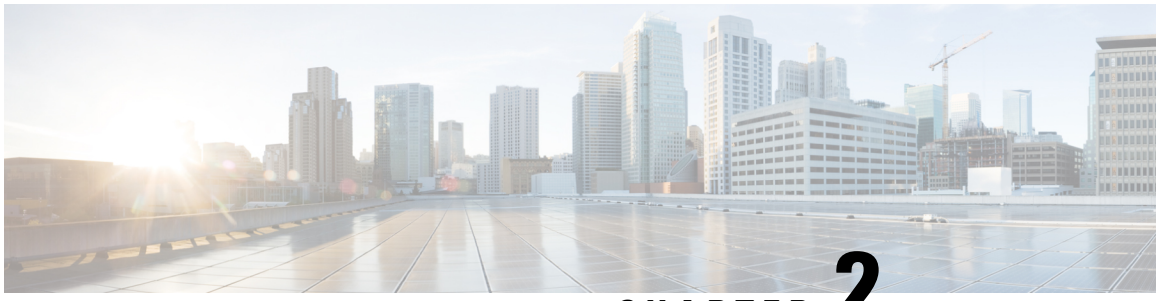


The three-tiered architecture on which Cisco CP NFs are designed fully support the 5G core (5GC) Service-based Architecture (SBA) defined by 3GPP. These NFs communicate with each other and with third-party NFs over the Service-based Interface (SBI) using HTTP/2 over TCP as defined by 3GPP.

Figure 4: Cisco CP NF Service-based Architecture Support



For more information on the Cisco network functions, see their corresponding network function documentation.



CHAPTER 2

PCF Overview

Product Description

The Cisco Policy Control Function (PCF) is one of the control plane network functions (NF) of the 5G core network (5GC). Cisco PCF is an evolution from Cisco Policy and Charging Rules Function (PCRF) on the existing Cisco Policy Suite Cloud Native Docker container-based platform.

In the 5G network, PCF has the following features and functions:

- Support 5G QoS policy and charging control functions and the related 5G signaling interfaces. The 3GPP standards, such as N5, N7, N15, N28, N36, and Rx, define these interfaces for the 5G PCF.
- Provide policy rules for control plane functions, which include network slicing, roaming, and mobility management.
- Collect the subscriber metrics in context with their network, usage, applications, and more. The operators analyze this information to optimize resources and make informed decisions to segment users.
- Provide the real-time management of subscribers, applications, and network resources based on the business rules configured for a service provider.
- Accelerate and simplify deployment and upgrades using the ConfD CLI, increased speed and efficiency, and low latency by adopting the cloud-native implementation.
- Collaborate with other NFs through NRF, which provides a unified communication platform for the NFs to interact with each other.

For information on how to deploy and configure PCF, see Deploying and Configuring PCF through Ops Center, on page 15.

Use Cases

The policy charging solution can be potentially applied to address various business scenarios. Some of the key application scenarios are described in this section.

Base PCF Configuration

PCF base configuration provides a detailed view of the configurations that are required for making PCF operational. This includes setting up the infrastructure to deploy PCF, deploying PCF through SMI, and configuring the Ops Center for exploiting the PCF capabilities over time.

This use case involves the following steps:

1. Prerequisites—Provides the list of resources that are required to deploy PCF in your environment successfully. See Prerequisites, on page 16 for details.
2. Deployment through SMI—All the 5G network functions are deployed through the SMI platform. The platform simplifies the cloud-native NF deployments and monitors the NF performance while providing an integrated experience.

   See Deploying PCF, on page 16 for details.
3. Configuring Ops Center—The PCF Ops Center provides an intuitive console for interacting with PCF in terms of configuring and gaining visibility into resources and features that you have subscribed to.

   The Ops Center lets you review the current and historical configurations corresponding to your environment. See Accessing the PCF Ops Center, on page 16 for details.

Infrastructure

With moving to 5G Core, Cisco has built PCF to have a robust and flexible infrastructure. Considering the rapidly evolving industry trends in the area of capacity and bandwidth, the infrastructure is also continuously altered by converging various components to make it more reliable, scalable, and secure.

Some of the key integrations that PCF infrastructure has undergone include the Cisco Common Data Layer—PCF supports the Geographic Redundancy (GR) for the Cisco Common Data Layer (CDL). See Cisco Common Data Layer, on page 57 for more information.

Interoperability with CHF

Complying with the charging architecture published in 3GPP December 2018 release 15. In the 5G Service-based architecture, PCF interoperates with the CHF. For instance, PCF determines the policy decisions that are based on the status of the policy counters available in the CHF.

This use case involves the following steps:

- N28 Interface—PCF allows retrieval of policy counters and their use in policy decisions. See N28 Interface, on page 205 for details.
- Forwarding the NAP and LDAP requests—The Policy Server relies upon the NAP and LDAP server to collect the subscriber details. With the revised Policy Server, PCF processes the subscriber detail requests and sends it to the appropriate function that is PCF or PCRF. It determines the function considering the technology that the subscriber has subscribed to. See Session Queries over LDAP, on page 301 for details.

Interoperability with NRF

The Network Repository Function (NRF) is one of the key network entities in the 5G Core Network (5GC). It primarily maintains the NF profile of the available NF instances and their supported services. It permits the NF instances to subscribe to, and get notified about the registration in NRF of new NF instances. The NRF supports the service discovery function by receiving the NF Discovery Requests from NFs and providing the information of the available NF instances by satisfying specific criteria such as supporting a given service.

This use case involves the following:

- NRF Interface—The NRF offers a platform for the NFs to communicate with each other and to exchange information for carrying out their operations. However, to build this communication framework, the NFs similar to PCF must register their profiles and services with the NRF. The NFs use the NRF's native management and discovery services to establish this framework. See NRF Interface, on page 191 for details.
- NRF Subscription to Notifications—PCF supports NRF and the associated repository functions such as the interface discovery, registration for renaming NRF, change type, and removal or addition of new API attributes. PCF extends this support as per the 3GPP December 2018 specification compliance. See Network Repository Function Subscription to Notifications, on page 179 for details.
- Heartbeat—The NF heartbeat configuration enables the network functions to notify their operational status to the NRF periodically. PCF invokes a heartbeat at the configured intervals. If the NRF is unavailable, then PCF switches between the registered primary, secondary, and tertiary NRF depending on their availability. See Heartbeat, on page 131 for more information.
- N28 Interface—PCF discovers the NFs based on the Instance ID which the NFs provide such as CHF and UDR. See for N28 Interface, on page 205 and UDR Interface, on page 335 for details.

Configuring LDAP for Subscriber Query

The policy charging solution combines with LDAP to sends and receives trusted information about the modified subscriber or subscriber details through the LDAP interface.

PCF has constructed the following capabilities to optimize the services that LDAP offers:

- **PCF as an LDAP Client**

  LDAP and Sh Interface—PCF acts as an LDAP client and establishes communication with Home Subscriber Server (HSS) and downloads the subscription profile over a Sh Interface. This enables PCF to update the policies automatically in the SMF when the Sh, LDAP, or local configuration sends a subscription change notification. See LDAP and Sh Interface, on page 135 for details.

- **PCF as an LDAP Server**

  Forwarding the NAP and LDAP requests—PCF acts as an LDAP server. The Policy Server relies upon NAP and the LDAP server to collect the subscriber details. With the revised Policy Server, it now processes the subscriber detail requests and sends it to the appropriate function that is PCF or PCRF. It determines the function considering the technology that the subscriber has subscribed to. See Session Queries over LDAP, on page 301 for details.

Parity with 4G

4G introduced cutting-edge solutions that redefined the way humans consumed cellular technology. It turned out to be an inherent part of exponential growth and amplified human advancement with AI, IoT, and other applications that exploit the technology. When 5G was conceived, some of the key capabilities of 4G were rebuilt on the 5G’s tech stack and infrastructure to provide a more scalable and positive experience to the customer base.

PCF has adopted the following feature from the 4G implementation:

Rx Authorization—PCF provides a method for service providers to regulate the services available to individual subscribers. You can configure the bearer-level regulation through the configuration of the Rx Authorization.

The configuration lets you control the services available to each subscriber. See Rx Authorization, on page 265 for details.

VoNR

In the new 5G spectrum, the subscribers are aware of the transitioning infrastructure that offers high-speed, increased capacity, reduced cost, real-time interaction, and other innovative offerings. However, the expectation that is associated with telecommunication still revolves around making regular voice calls, emergency calls, exceeding quality audio, and sending SMS. Service providers are being competitive over providing a positively differentiated experience to the user while making the Audio, Video, and Emergency calls. Like 4G, the providers can access the VoNR through PCF, which is the preferred approach.

This use case involves the following:

- VoNR through the Rx Interface—With PCF in 5G supporting full Diameter stack with the supported standard Diameter Rx interfaces, PCF accepts Rx messages for processing and Rx session binding with N7 sessions. See VoNR through the Rx Interface, on page 351 for details.
- Specification Compliance - N5, N7 and N28—Enhancements to the N7 and N28 interfaces of PCF to comply with the 3GPP December 2018 specification and enhancements to the N5 interface of PCF to comply with the 3GPP December 2020 specification. See Specification Compliance - N7 and N28, on page 315 for details.
- Predefined Rule and Rulebase—Provision to configure PCC rule ID for predefined rule and rulebase is available in PCF. SMF uses these rules when configuring the User Plane Function (UPF) for performing data flow tasks, such as shaping, policing to provide bandwidth, and charging functions. See Predefined Rules and Rulebase, on page 263 for more information.
- Dynamic Rules and Table-driven Charging Rules—PCF supports the provisioning of the table-driven dynamic charging rules. See Dynamic Rules and Table-Driven Charging Rules, on page 105 for more information.
- Dummy N7 Notify Request—If PCF has not subscribed to specific event triggers during the session initiation, it can send a dummy N7 Notify Request, which is an intermediate request to fetch those event triggers. The events must correspond to the configured Media-Type specified in the AAR message from the IMS. See Dummy N7 Notify Request, on page 89 for more information.

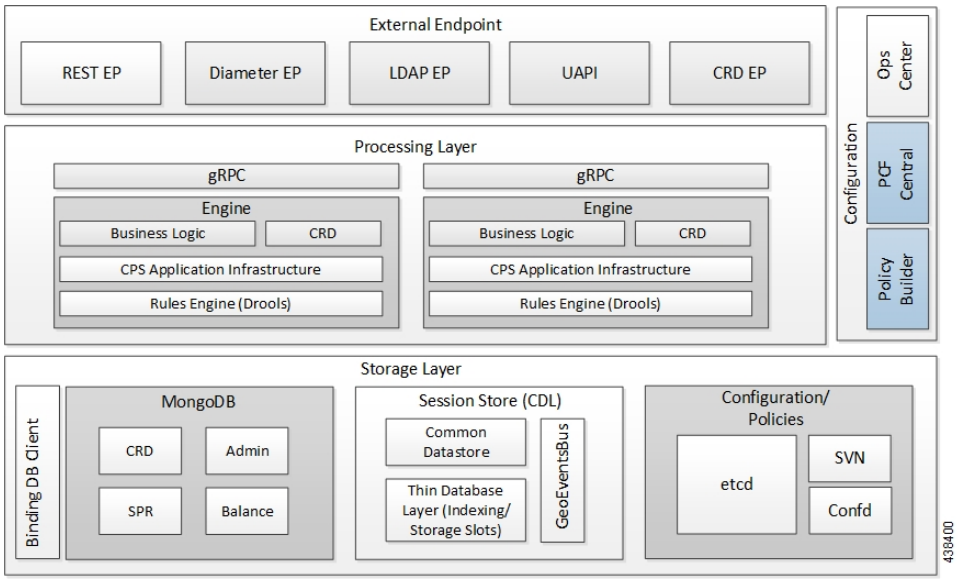
Deployment Architecture and Interfaces

The Cisco PCF is part of the 5G core network functions portfolio with a common mobile core platform architecture. These network functions include Access and Mobility Management Function (AMF), Session Management Function (SMF), Network Function Repository Function (NRF), Policy Control Function (PCF), Network Slice Selection Function (NSSF), and User Plane Function (UPF).

PCF Architecture

The PCF architecture is built on a multi-layer platform, which enables efficient policy control and management in the 5G Core network.

Figure 5: PCF Architecture



At a high level, the components in the architecture perform the following:

1. External Endpoint
   - REST-EP—It is a RESTful interface, which provides a channel for the 5G inbound and outbound messages.
   - LDAP-EP, UAPI, and CRD API—Provides interfaces for PCF communications.
   - Diameter-EP—Responsible for routing the Diameter traffic.
2. Processing Layer
   - grPC—Provides a framework that enables the internal processes to communicate with each other and synchronize their events.
   - PCF-Engine—Hosts the business logic of PCF and responsible for driving the rules engine for making crucial policy decisions.
3. Configurations
   - Policy Builder—Allows configuration of the PCF cluster of virtual machines (VMs) and configuration of services and advanced policy rules.
   - PCF Central—Provides a unified GUI that allows you to configure Policy Builder, manage custom reference table data, and start the Web-based applications and utilities.
   - Ops Center—Allows you to configure and manage the applications and pods configuration.
4. Storage Layer
   - Binding Database Client—Provisions the client to look up the PCRF Mongo Binding Database for information about the secondary key lookup across 4G and 5G.
   - MongoDB—Preserves the subscriber-specific, balance data, and admin configuration data.

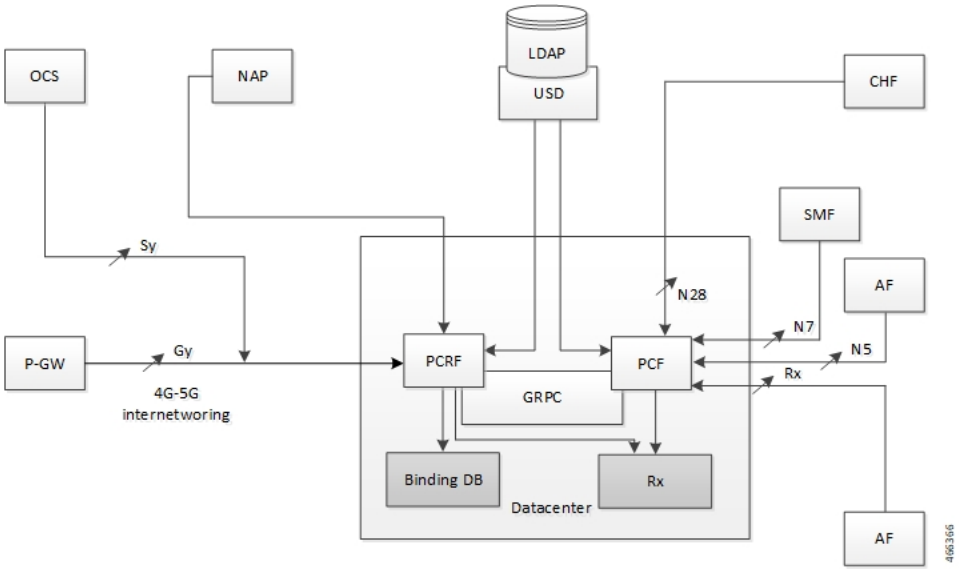
- Session store—Contains the data which CDL accesses for processing a session persistence activity. Stores the PCF sessions.
- Etcd—Contains the Diameter endpoint configurations.

PCF Deployment Architecture

The PCF reduces the deployment complexity by integrating PCRF and PCF in a unified environment.

The following figure illustrates the PCF deployment.

Figure 6: PCF Deployment



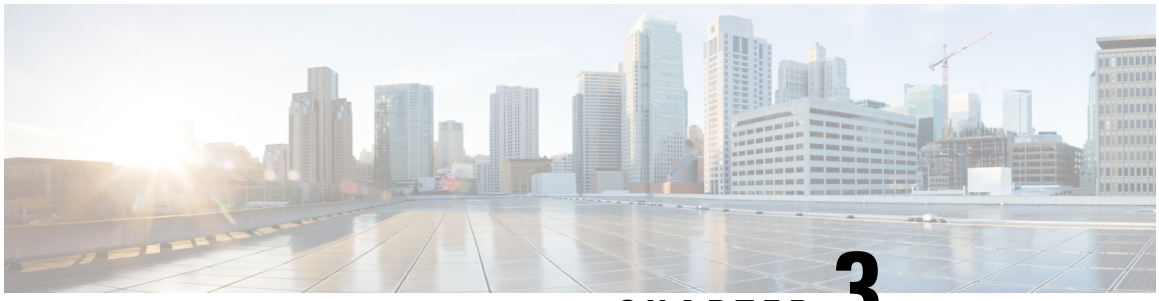
Note

- The PCRF's deployment architecture includes:
  - One Region = Two sites. Each site has one cluster (total two clusters in a region).
  - Noncloud-native deployment along with cloud-native 5G PCF.
  - External binding database is the local database for PCRF.
  - MongoDB is the dedicated session database for PCRF.
- The PCF's deployment architecture includes:
  - One Region = Two sites. Each site has one cluster (total two clusters in a region).
  - Cloud-native deployment that deployed along with 4G PCRF.
  - Cisco CDL is the dedicated session database for PCF.

Supported Interfaces

PCF and other NFs in 5GC use the following:

- Rx– Reference point for interworking with AF, PCRF, and PCF
- N5– Reference point between PCF and AF
- N7– Reference point between PCF and SMF
- N15– Reference point between PCF and AMF
- N28– Reference point between PCF and CHF
- N36– Reference point between PCF and UDR
- LDAP– Reference point between PCF and external subscriber profile



CHAPTER 3

Deploying and Configuring PCF through Ops Center

Feature Summary and Revision History

Summary Data

Table 3: Summary Data

Applicable Product(s) or Functional Area	PCF
Applicable Platform(s)	SMI
Default Setting	Enabled – Always-on
Related Documentation	Not Applicable

Revision History

Table 4: Revision History

Revision Details	Release
First introduced.	2020.01.0

Feature Description

The PCF deployment and configuration process involve deploying PCF through the SMI Deployer and configuring the settings or customization through the PCF Ops Center. The Ops Center is based on the ConfD

CLI. Configuration of PCF also includes the NRF profile data configuration and setting up the externally visible IP address and port numbers.

PCF Ops Center

The PCF Ops Center allows you to configure the PCF features such as configuring the license, PCF Engine, REST endpoint, and CDL. You can also configure the NRF components that enable the interworking of various NFs.

Policy Ops Center reuses the existing Ops Center image from mobile-cnat-infrastructure, and is accessible via the ingresses that are defined by that chart.

Prerequisites

Before deploying PCF on the SMI layer, complete the following prerequisites.

- Ensure that all the virtual network functions (VNFs) are deployed.
- Run the SMI sync operation for the PCF Ops Center and Cloud Native Common Execution Environment (CN-CEE).

Deploying and Accessing PCF

This section describes how to deploy PCF and access the PCF Ops Center.

Deploying PCF involves the following steps:

1. Deploying PCF
2. Accessing the PCF Ops Center

Deploying PCF

The Subscriber Microservices Infrastructure (SMI) platform is responsible for deploying and managing the Cloud Native 5G PCF application and other network functions.

For information on how to deploy PCF Ops Center on a vCenter environment, see *Configuring the vCenter Environment* section in *Ultra Cloud Core SMI Cluster Deployer Operations Guide*.

For deploying PCF Ops Center on an OpenStack environment, see *UAME-based VNF Deployment* section in the *UAME-based 4G and 5G VNF Deployment Automation Guide, Release 6.9*.

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

Accessing the PCF Ops Center

This section describes how to access the PCF Ops Center.

You can access the PCF Ops Center from the console application or the Web-based CLI console. Depending upon your selection, access one of the following from the master node:

1. **CLI:**

   ssh admin@*ops_center_pod_ip* **-p 2024**

2. **Web-based console:**

   a. Log in to the Kubernetes master node.

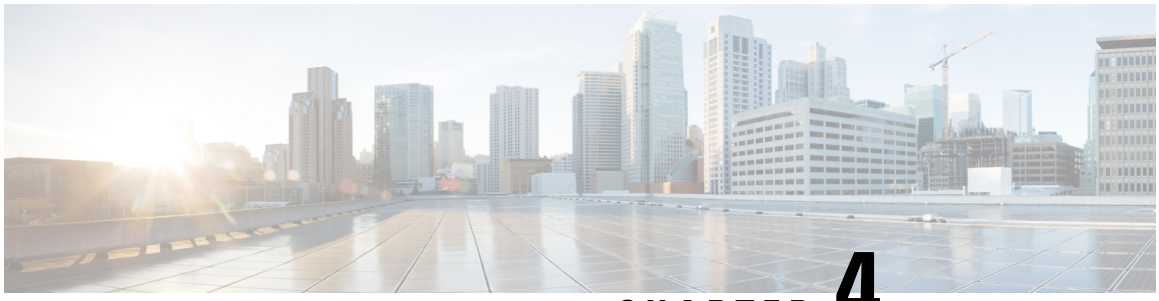
   b. To view the available ingress connections, use the following configuration:

      kubectl get ingress *namespace*

      The available ingress connections are displayed.

   c. Select the appropriate ingress from where you want to run Ops Center and open the following URL from the browser:

      cli.*namespace***-ops-center.***ip_address***.nip.io**



CHAPTER 4

Smart Licensing

Feature Summary and Revision History

Summary Data

Table 5: Summary Data

Applicable Products or Functional Area	PCF
Applicable Platform(s)	SMI
Feature Default Setting	Disabled – Configuration required to enable
Related Documentation	Not Applicable

Revision History

Table 6: Revision History

Revision Details	Release
First introduced.	2020.02.0

Smart Software Licensing

Smart Licensing is a cloud-based approach to licensing that simplifies the purchase, deployment, and management of Cisco software assets. Entitlements are purchased through your Cisco account via Cisco Commerce Workspace (CCW) and immediately deposited into your Virtual Account for usage. This eliminates

the need to install license files on every device. Products that are smart enabled communicate directly to Cisco to report consumption. Cisco Software Central (CSC) is a single location which is available to customers to manage Cisco software licenses. License ownership and consumption are readily available to help make better purchase decision based on consumption or business need. See https://www.cisco.com/c/en/us/buy/smart-accounts/software-licensing.html for more information about Cisco Smart Licensing.

Cisco Software Central

Cisco Software Central (CSC) enables the management of software licenses and Smart Account from a single portal. The interface allows you to activate your product, manage entitlements, and renew and upgrade software. A functioning Smart Account is required to complete the registration process. To access the Cisco Software Central, see https://software.cisco.com.

Smart Accounts/Virtual Accounts

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

A Virtual Account exists as a subaccount within the Smart Account. Virtual Accounts are a customer-defined structure based on organizational layout, business function, geography, or any defined hierarchy. They are created and maintained by the Smart Account administrator.

See https://software.cisco.com to learn about, set up, or manage Smart Accounts.

Request a Cisco Smart Account

A Cisco Smart Account is an account where all products enabled for Smart Licensing are deposited. 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 your organization's Smart Account through the Software Central.

1. In a browser window, enter the following URL:

   ```
   https://software.cisco.com
   ```

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

   The **Smart Account Request** window is displayed.

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 authorization to create and manage product and service entitlements, users, and roles on behalf of your 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 will create the Smart Account.

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).

5. Click **Continue**.

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

PCF Smart Licensing

At present, the Smart Licensing feature supports application entitlement for online and offline licensing for all Cisco 5G applications (PCF and SMF). The application usage is unrestricted during all stages of licensing including Out of Compliance (OOC) and expired stages.

Note A 90-day evaluation period is granted for all licenses in use. Currently, the functionality and operation of the 5G applications is unrestricted even after the end of the evaluation period.

Software Tags and Entitlement Tags

Tags for the following software and entitlements have been created to identify, report, and enforce licenses.

Software Tags

Software tags uniquely identify each licensable software product or product suite on a device. The following software tags exist for the PCF.

Product Type / Description	Software Tag
Ultra Cloud Core - Policy Control Function (PCF), Base Minimum	regid.2020-04.com.cisco.PCF,1.0_a0b80e76-1cc3-4a0f-bbf5-c7a8dafea5f8

Entitlement Tags

The following entitlement tags identify licenses in use:

Product Type / Description	Entitlement Tag
Ultra Cloud Core - Policy Control Function (PCF), Base Minimum	regid.2020-04.com.cisco.PCF_BASE,1.0_60b1da6f-3832-4687-90c9-8879dc815a27

Note The license information is retained during software upgrades and rollback.

Configuring Smart Licensing

You can configure Smart Licensing after a new PCF deployment.

Users with Access to CSC

This section describes the procedure involved in configuring Smart Licensing for users with access to CSC portal from their internal 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 in to your CSC account.
2. Click **Add Product** and enter the following details.
   - **Product name**—Specify the name of the deployed product. For example, PCF.
   - **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 Account's team.
3. Click **Create**.
4. Select your product from the **Product/Entitlement Setup** grid.
5. Click **Entitlement** drop-down and 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, PCF_BASE.
   - **Description**—(Optional) Specify a brief description about the license entitlement.
   - **Entitlement Tag**—Specify the entitlement tag provided by the Cisco Account's 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**, provide 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, provide 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 using the following command.

    `show license all`

    Example:

```
pcf# show license all

Smart Licensing Status
======================
Smart Licensing is ENABLED

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

License Authorization:
  Status: EVAL MODE
  Evaluation Period Remaining: 83 days, 0 hr, 15 min, 8 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: 83 days, 0 hr, 15 min, 8 sec

License Usage
=============
License Authorization Status: EVALUATION MODE
  Evaluation Period Remaining: 83 days, 0 hr, 15 min, 8 sec

UCC 5G PCF BASE (PCF_BASE)
  Description: Ultra Cloud Core - Policy Control Function (PCF), Base Minimum
  Count: 1
  Version: 1.0
  Status: EVAL MODE
```

```
  Export status: RESTRICTED_NOTALLOWED
  Feature Name: <empty>
  Feature Description: <empty>

Product Information
===================
UDI: PID:PCF,SN:6GKJ2OA-NMUWA7Y

Agent Version
=============
Smart Agent for Licensing: 3.0.13
```

Registering Smart Licensing

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

1. Log in to your CSC account.
2. Choose **General > New Token** and enter the following details:
   - **Description**—Specify a brief description about the ID token.
   - **Expires After**—Specify the number of days for the token to expire.
   - **Max. Number Users**—Specify the maximum number users.
3. Click **Create Token**.
4. Select **new ID token** in **Product Instance Registration Token**.
5. Choose **Actions > Copy**.
6. Log in to PCF Ops Center CLI and paste the **ID token** using the following configuration:

   `license smart register` *`idtoken`*

   Example:

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

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

   `show license all`

   Example:

   ```
   pcf# show license all

   Smart Licensing Status
   ======================
   Smart Licensing is ENABLED

   Registration:
     Status: REGISTERED
     Smart Account: Cisco Systems, Inc.
     Virtual Account: PCF-SMF
     Export-Controlled Functionality: Allowed
     Initial Registration: SUCCEEDED on Apr 15 05:45:07 2020 GMT
     Last Renewal Attempt: SUCCEEDED on Apr 15 05:45:07 2020 GMT
   ```

```
  Next Renewal Attempt: Oct 12 05:45:07 2020 GMT
  Registration Expires: Apr 15 05:40:31 2021 GMT

License Authorization:
  Status: AUTHORIZED on Apr 15 05:45:12 2020 GMT
  Last Communication Attempt: SUCCEEDED on Apr 15 05:45:12 2020 GMT
  Next Communication Attempt: May 15 05:45:12 2020 GMT
  Communication Deadline: Jul 14 05:40:40 2020 GMT

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

Utility:
  Status: DISABLED

Transport:
  Type: CALLHOME

Evaluation Period:
  Evaluation Mode: Not In Use
  Evaluation Period Remaining: 83 days, 0 hr, 10 min, 43 sec

License Usage
=============
License Authorization Status: AUTHORIZED as of Apr 15 05:45:12 2020 GMT

UCC 5G PCF BASE (PCF_BASE)
  Description: Ultra Cloud Core - Policy Control Function (PCF), Base Minimum
  Count: 1
  Version: 1.0
  Status: AUTHORIZED
  Export status: RESTRICTED_ALLOWED
  Feature Name: <empty>
  Feature Description: <empty>

Product Information
===================
UDI: PID:PCF,SN:6GKJ2OA-NMUWA7Y

Agent Version
=============
Smart Agent for Licensing: 3.0.13
```

NOTES:

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

Deregistering Smart Licensing

You can deregister the registered product from Smart Licensing if required.

1. Log in to PCF Ops Center CLI and use the following configuration:

   `license smart deregister`

   Example:

   ```
   pcf# license smart deregister
   pcf#
   ```

2. Verify the status of Smart Licensing using the following command.

   show license all

   Example:

```
pcf# show license all

Smart Licensing Status
======================
Smart Licensing is ENABLED

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

License Authorization:
  Status: EVAL MODE
  Evaluation Period Remaining: 83 days, 0 hr, 10 min, 43 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: 83 days, 0 hr, 10 min, 43 sec

License Usage
=============
License Authorization Status: EVALUATION MODE
  Evaluation Period Remaining: 83 days, 0 hr, 10 min, 43 sec

UCC 5G PCF BASE (PCF_BASE)
  Description: Ultra Cloud Core - Policy Control Function (PCF), Base Minimum
  Count: 1
  Version: 1.0
  Status: EVAL MODE
  Export status: RESTRICTED_NOTALLOWED
  Feature Name: <empty>
  Feature Description: <empty>

Product Information
===================
UDI: PID:PCF,SN:6GKJ2OA-NMUWA7Y

Agent Version
=============
Smart Agent for Licensing: 3.0.13

pcf#
```

   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). This 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

You can enable Smart License reservation through PCF Ops Center CLI.

1. Log in to PCF Ops Center CLI and use the following configuration:

```
config terminal
  license smart reservation
  commit
  end
```

NOTES:

- **license smart reservation** —Enables license reservation.

Generating Smart License Reservation Request Code

You can generate the Smart License reservation request code through PCF Ops Center CLI.

1. Log in to PCF Ops Center CLI and using the following configuration to enable the reservation:

```
config terminal
  license smart reservation
  commit
  end
```

2. Use the following configuration to request a reservation code:

```
license smart reservation request
```

Example:

```
pcf# license smart reservation request
reservation-request-code CJ-ZPCF:6GKJ2OA-NMUWA7Y-Ai75GxtBs-3B
pcf#
Message from confd-api-manager at 2020-04-15 05:51:37...
Global license change NotifyReservationInProgress reason code Success - Successful.
pcf#
```

NOTES:

- **license smart reservation** —Enables license reservation request code.
- **license smart reservation request** —Generates the license reservation request code.

Important You need to copy the generated license request code from the PCF Ops Center CLI.

Generating an Authorization Code from CSC

You can generate an authorization code from CSC using the license reservation request code.

1. Log in to your CSC account.
2. Click **License Reservation**.
3. Enter the Request Code: Paste the license reservation request code copied from the PCF Ops Center CLI in the **Reservation Request Code** text-box.
4. Select the Licenses: Click the **Reserve a Specific License** radio button and select **UCC 5G PCF BASE**.

Note In the **Reserve** text-box enter the value *1*.

5. Review your selection.
6. Click **Generate Authorization Code**.
7. Download the response file: The authorization code is generated and displayed on-screen. Click **Download as File** to download the authorization code.
8. Click **Close**.

Reserving Smart Licensing

You can reserve Smart License for the deployed product using the authorization code generated in CSC.

1. Log in to PCF Ops Center CLI and use the following configuration:

   `license smart reservation install`
   `authorization_code`

   Example:

   ```
   pcf# license smart reservation install
   Value for 'key' (<string>):
   <specificPLR><authorizationCode><flag>A</flag><version>C</version>
   <piid>35757dc6-2bdf-4fa1-ba7e-4190f5b6ea22</piid><timestamp>1586929992297</timestamp>
   <entitlements><entitlement><tag>regid.2020-04.com.cisco.PCF_BASE,1.0_60b1da6f-3832-4687-90c9-8879dc815a27</tag>
   <count>1</count><startDate>2020-Apr-08 UTC</startDate><endDate>2020-Oct-05 UTC</endDate>
   <licenseType>TERM</licenseType><displayName>UCC 5G PCF BASE</displayName>
   <tagDescription>Ultra Cloud Core - Policy Control Function (PCF), Base
   Minimum</tagDescription>
   <subscriptionID></subscriptionID></entitlement></entitlements></authorizationCode>
   <signature>MEYCIQC/9v5IpgFoEk2l4cmIgjjk83g5WXjzs09kQnsO8D0jRgIhAMn+D6DRuYmqh1TlfJoZxNte0fPKw6fHEY5CEF3+kPQj</signature>
   <udi>P:PCF,S:6GKJ2OA-NMUWA7Y</udi></specificPLR>
   pcf#
   ```

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

   `show license all`

   Example:

   ```
   pcf# 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 Wed Apr 15 05:53:31 GMT 2020
  Last Renewal Attempt: None

License Authorization:
  Status: AUTHORIZED - RESERVED on Wed Apr 15 05:53:31 GMT 2020

Utility:
  Status: DISABLED

Transport:
  Type: CALLHOME

Evaluation Period:
  Evaluation Mode: Not In Use
  Evaluation Period Remaining: 83 days, 0 hr, 5 min, 15 sec

License Usage
=============
License Authorization Status:
  Status: AUTHORIZED - RESERVED on Wed Apr 15 05:53:31 GMT 2020
  Last Communication Attempt: SUCCEEDED on Apr 15 05:53:31 2020 GMT
  Next Communication Attempt: NONE
  Communication Deadline: NONE

UCC 5G PCF BASE (PCF_BASE)
  Description: Ultra Cloud Core - Policy Control Function (PCF), Base
 Minimum
  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: 2020-Oct-05 GMT

Product Information
===================
UDI: PID:PCF,SN:6GKJ2OA-NMUWA7Y

Agent Version
=============
Smart Agent for Licensing: 3.0.13
```

NOTES:

- **license smart reservation install** *authorization_code* – Installs a Smart License Authorization code.

Returning the Reserved License

You can return the reserved license to CSC if required. Use the following procedures to return the reserved license:

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

      `license smart reservation return`

      Example:

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

   b. Copy the license reservation return code generated in PCF Ops Center CLI.
   c. Log in to your CSC account.
   d. Select your product instance from the list.
   e. Choose **Actions** > **Remove**.
   f. Paste the license reservation return code in **Return Code** text-box.

   NOTES:

   - **license smart reservation return** – Returns a reserved Smart License.

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

      `license smart reservation return`
      `authorization_code`

      Important Paste the license reservation authorization code generated in CSC to generate the return code.

   b. Log in to your CSC account.
   c. Select your product instance from the list.
   d. Choose **Actions** > **Remove**.
   e. Paste the license reservation return code in **Return Code** text-box.

3. Verify the status of smart licensing using the following command.

   `show license all`

   Example:

   ```
   pcf# show license all

   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: 83 days, 0 hr, 5 min, 15 sec
  Last Communication Attempt: SUCCEEDED on Apr 15 05:53:31 2020 GMT
  Next Communication Attempt: NONE
  Communication Deadline: 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: 83 days, 0 hr, 5 min, 15 sec

License Usage
=============
License Authorization Status: EVALUATION MODE
  Evaluation Period Remaining: 83 days, 0 hr, 5 min, 15 sec

UCC 5G PCF BASE (PCF_BASE)
  Description: Ultra Cloud Core - Policy Control Function (PCF), Base Minimum
  Count: 1
  Version: 1.0
  Status: EVAL MODE
  Export status: RESTRICTED_NOTALLOWED
  Feature Name: <empty>
  Feature Description: <empty>

Product Information
===================
UDI: PID:PCF,SN:6GKJ2OA-NMUWA7Y

Agent Version
=============
Smart Agent for Licensing: 3.0.13

pcf#
```

OAM Support

This section describes operations, administration, and maintenance support for this feature.

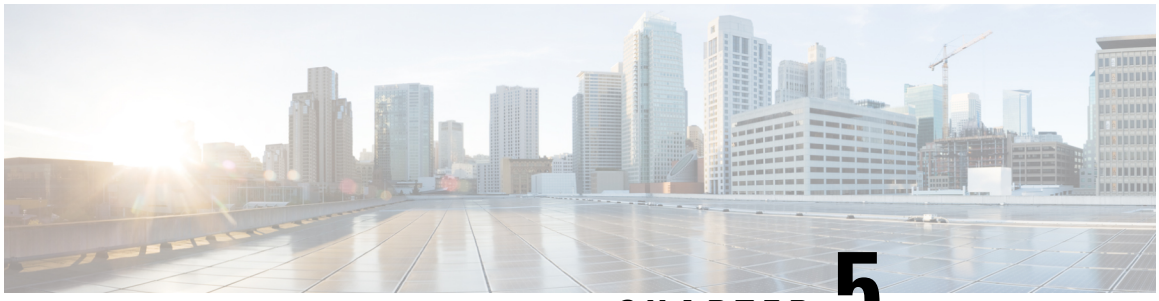
Monitoring and Troubleshooting Smart Licensing

You can use the following show commands to display information about Smart Licensing in the PCF 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 and, 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 5

PCF Rolling Software Update

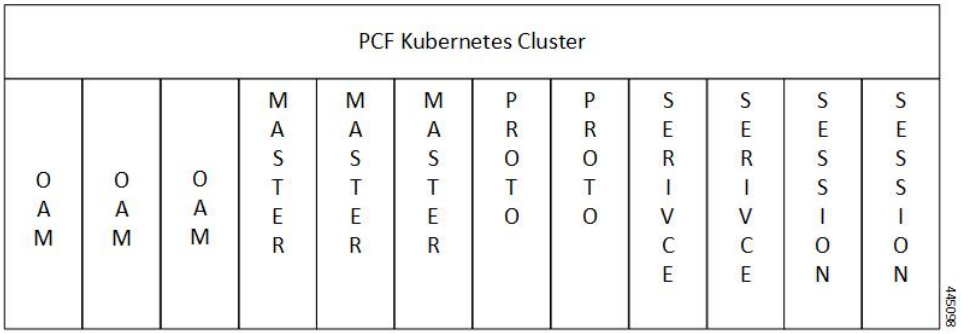
Introduction

The Cisco PCF has a three-tier architecture which consists 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 of Kubernetes (K8s) master, and worker nodes (including Operation and Management nodes).

For high availability and fault tolerance, a minimum of two K8s worker nodes are required for each tier. You can have multiple replicas for each worker node. Kubernetes orchestrates the pods using the StatefulSets controller. The pods require a minimum of two replicas for fault tolerance.

The following figure depicts a PCF K8s Cluster with 12 nodes – 3 Master nodes, 3 Operations, and Management (OAM) worker nodes, 2 Protocol worker nodes, 2 Service worker nodes, 2 Session (data store) worker nodes.

Figure 7: PCF Kubernetes Cluster



Note

- OAM worker nodes - These nodes host the Ops Center pods for configuration management and metrics pods for statistics and Key Performance Indicators (KPIs).
- Protocol worker nodes - These nodes host the PCF protocol-related pods for service-based interfaces (N5, N7, N28, N36, and NRF) and Diameter Rx Endpoint.
- Service worker nodes - These nodes host the PCF application-related pods that perform session management processing.
- Session worker nodes - These nodes host the database-related pods that store subscriber session data.

Updating PCF

The following section describes the procedure involved in updating the PCF software:

- Rolling Software Update Using SMI Cluster Manager

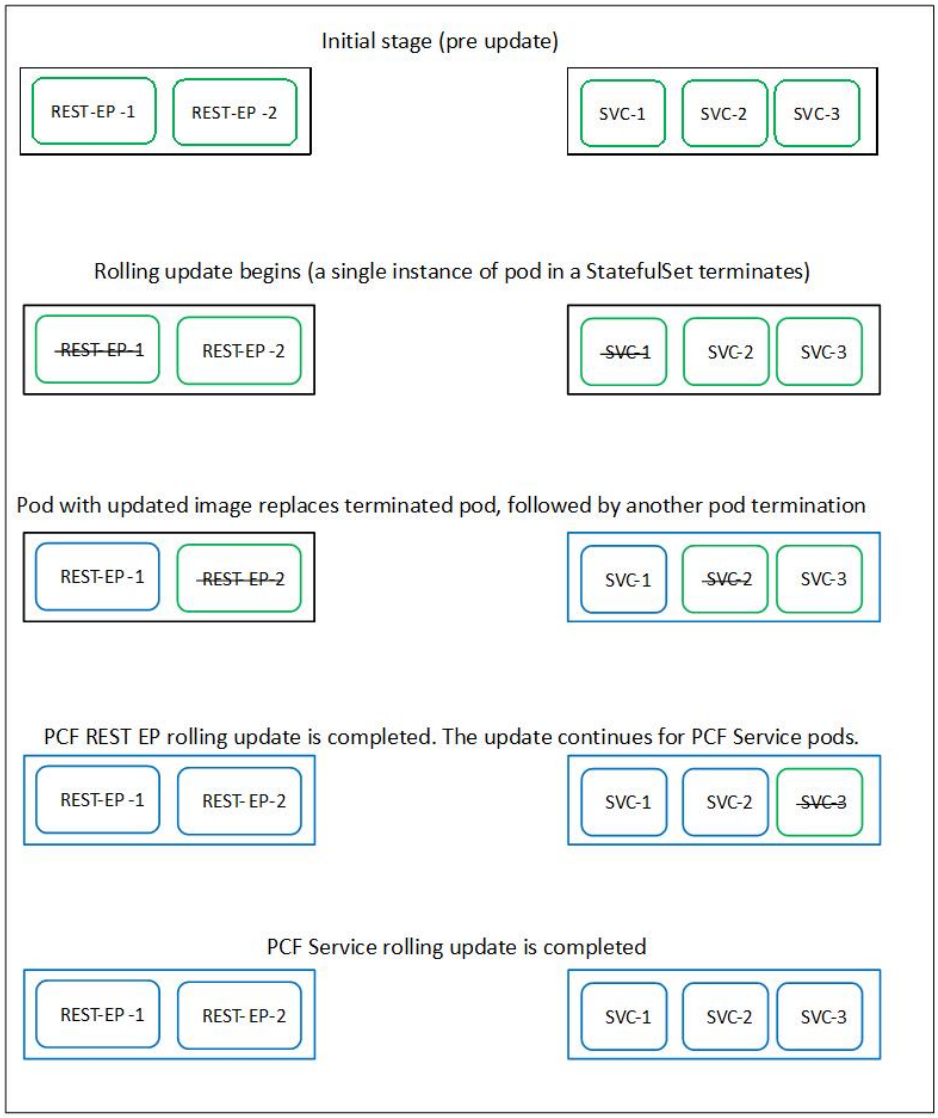
Rolling Software Update Using SMI Cluster Manager

The PCF 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 updates sequentially to ensure that the ongoing process remains unaffected. 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 competing all the ongoing processes. Other in-service pods continue to receive and process the traffic to provide a seamless software update. You can control the software update process through the Ops Center CLI.

Note Each pod needs a minimum of two replicas for high availability. For example, Policy Engine must have 2 Engine replicas. In a worst-case scenario, the processing capacity of the pod may briefly reduce to 50% while the software update is in-progress.

The following figure illustrates a PCF rolling update for PCF REST endpoint pods (two replicas) on Protocol worker nodes along with PCF Service pods (three replicas) on Service worker nodes.

Figure 8: PCF Rolling Update



Prerequisites

The prerequisites for upgrading PCF are:

- All the nodes – including all the pods in the node – are up and running.
- A patch version of the PCF software.

Note Currently, major versions do not support the rolling upgrade. The major version represents the release year, release number, and maintenance number. Cisco follows the versioning format as YYYY.RN.MN such as 2020.03.0.

Important Trigger rolling upgrade only when the CPU usage of the nodes is less than 50%.

PCF Health Check

You need to perform a health check to ensure that all the services are running and nodes are in ready state. To perform a health check:

1. Log in to 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 pcf-wsp -o wide
kubectl get pods -n cee-wsp -o wide
kubectl get pods -n smi-vips -o wide
helm list
kubectl get pods -A | wc -l
```

Important Ensure that all the nodes are in the ready state before you proceed further. Use the kubectl get nodes command to display the node states.

Preparing for Upgrade

This section describes the procedure involved creating a backup configuration, logs, and deployment files. To back up the files:

1. Log in to the SMI Cluster Manager Node as an **ubuntu** user.
2. Create a new directory for deployment.

   Example:

   ```
   test@smipcf-cm01:~$ mkdir -p "temp_$(date +'%m%d%Y_T%H%M')" && cd "$_"
   ```

3. Move all the *pcf* deployment file into the newly created deployment directory.
4. Untar the *pcf* deployment file.

   Example:

   ```
   test@smi1pcf01-cm01:~/temp_08072019_T1651$ tar -xzvf pcf.2020.01.0-1.SPA.tgz
   ./
   ./PCF_REL_KEY-CCO_RELEASE.cer
   ./cisco_x509_verify_release.py
   ./pcf.2020.01.0-1.tar
   ./pcf.2020.01.0-1.tar.signature.SPA
   ./pcf.2020.01.0-1.tar.SPA.README
   ```

5. Verify the downloaded image.

   Example:

   ```
   test@smi1pcf01-cm01:~/temp_08072019_T1651$ cat pcf.2020.01.0-1.tar.SPA.README
   ```

Important Follow the procedure mentioned in the *SPA.README* file to verify the build before proceeding to the next step.

Back Up SVN, Policy, and CRD Data

This section describes the procedure involved in creating a backup of SVN, Policy, and CRD data. To perform a backup of SVN and Policy files:

1. Log in to the master node as an **ubuntu** user.
2. Use the following command to retrieve the Policy Builder URL.

   ```
   kubectl get ing -n $( kubectl get namespaces | grep -oP 'pcf-(\d+|\w+)'
    | cut -d\  -f1) | grep policy-builder | awk '{ print $2 }'
   pb.pcf-02-pcf-engine-app-blv02.ipv4address.nip.io
   ```

   Example:

   ```
   ubuntu@ mas01:~/backups_09182019_T2141$ kubectl get ing -n $( kubectl get namespaces |
   grep -oP 'pcf-(\d+|\w+)' | cut -d\  -f1) | grep policy-builder | awk '{ print $2 }'
   ```

   Sample output:

   ```
   pb.pcf-02-pcf-engine-app-blv02.ipv4address.nip.io
   ```

3. Navigate to the Policy Builder home page.
4. Click **Import/Export**.
5. Click **All Data**.
   - **Export URL**—Specify the export URL.
   - **Export File Prefix**—Specify an appropriate name for the export file.
6. Click **Export**.

Important You can find the exported file in your local **Downloads** directory.

To perform a backup of CRD data:

1. Navigate to the Policy Builder Home page.
2. Click **Custom Reference Data.**
3. Click **Import/Export CRD data.**
4. Click **Export**.

Important You can find the CRD data in your Web browser's **Downloads** directory.

Back Up Ops Center Configuration

This section describes the procedure involved in creating a backup of the Ops Center configurations.

To perform a backup of 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')
   ```

   NOTES:

   - **ssh -p <port_number>**: Specifies the port number of the system on which the SMI Ops Center service is running. Use the **Kubectl get service** command to display the ports on which is the services are running.
   - ***netconf.*<port_number>**': Specifies the port number of the system on which the Netconf service is running.

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')
   ```

   NOTES:

   - **cee-vip**: Specifies the CEE VIP that is configured in the SMI Ops Center. Use the **show running-config** to display the SMI Ops Center configuration.

4. Run the following command to backup the PCF Ops Center configuration to **/home/ubuntu/pcfops.backup** file.

   ```
   ssh admin@<pcf-vip> "show run | nomore" > pcfops.backup_$(date
   +'%m%d%Y_T%H%M')
   ```

   NOTES:

   - **pcf-vip**>: Specifies the PCF VIP that is configured in the SMI Ops Center. Use the **show running-config** to display the SMI Ops Center configuration.

Back Up CEE and PCF Ops Center Configuration

This section describes the procedure involved in creating a backup of CEE and Ops Center configuration from the master node. To perform a backup of CEE and 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. Back up the PCF 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 'pcf-(\d+|\w+)') | grep <port_number> | awk '{ print $3
}') "show run | nomore" > pcfops.backup_$(date +'%m%d%Y_T%H%M') && wc
-l pcfops.backup_$(date +'%m%d%Y_T%H%M')
```

Example:

```
ubuntu@popcf-mas01:~/backups_09182019_T2141$ ssh -p <port_number> admin@$(kubectl get
svc -n $(kubectl get namespaces | grep -oP 'pcf-(\d+|\w+)') | grep <port_number> | awk
'{ print $3 }') "show run | nomore" > pcfops.backup_$(date +'%m%d%Y_T%H%M') && wc -l
pcfops.backup_$(date +'%m%d%Y_T%H%M')
admin@<admin_ip_address> password: PCF-OPS-PASSWORD
334 pcfops.backup
```

4. Back up 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@popcf-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@<admin_ip_address> 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@popcf-mas01:~/backups_09182019_T2141$ scp $(grep cm01 /etc/hosts | awk '{ print
$1 }'):/home/ubuntu/smiops.backup_$(date +'%m%d%Y_T%H%M') .
ubuntu@<admin_ip_address> password: SMI-CM-PASSWORD
smiops.backup                                                 100% 9346    22.3MB/s
   00:00
```

6. Verify the line count of the backup files.

Example:

```
ubuntu@popcf-mas01:~/backups_09182019_T2141$ wc -l *
  233 ceeops.backup
  334 pcfops.backup
  361 smiops.backup
  928 total
```

Upgrading the PCF

This section describes the procedures involved in upgrading PCF.

Staging a New PCF Image

This section describes the procedure involved in staging a new PCF image before initiating the upgrade.

To stage the new PCF image:

1. Download and verify the new PCF image.
2. Log in to the SMI Cluster Manager node as an **ubuntu** user.
3. Copy the images to **Uploads** directory.

   ```
   sudo mv <pcf_new_image.tar> /data/software/uploads
   ```

   Note The SMI uses the new image present in the **Uploads** directory to upgrade.

4. Verify whether the image is picked up by the SMI for processing from the **Uploads** directory.

   ```
   sleep 30; ls /data/software/uploads
   ```

   Example:

   ```
   ubuntu@popcf-cm01:~/temp_08072019_T1651$ sleep 30; ls /data/software/uploads
   ubuntu@popcf-cm01:~/temp_08072019_T1651$
   ```

5. Verify whether the images were successfully picked up and processed.

   Example:

   ```
   auser@unknown:$ sudo du -sh /data/software/packages/*
   1.6G /data/software/packages/cee.2019.07
   5.3G /data/software/packages/pcf.2019.08-04
   16K /data/software/packages/sample
   ```

   Note The SMI must unpack the images into the **packages** directory successfully to complete the staging.

Triggering the Rolling Software Upgrade

The PCF utilizes the SMI Cluster Manager to perform a rolling software update. To update PCF using SMI Cluster Manager, use the following configurations:

Important Before you begin, ensure that PCF is up and running with the current version of the software.

1. Log in to the SMI Cluster Manager console.
2. Run the following command to log in to the SMI Ops Center.

   ```
   ssh -p <port_number> admin@$(kubectl get svc -n smi | grep
   '.*netconf.*<port_number>' | awk '{ print $4 }')
   ```

   Example:

   ```
   ubuntu@popcf-cm01:~$ ssh -p <port_number> admin@$(kubectl get svc -n smi | grep
   '.*netconf.*<port_number>' | awk '{ print $4 }')
   admin@<admin_ip_address> password: SMI-CONSOLE-PASSWORD
   Welcome to the CLI
   admin connected from <admin_ip_address> using ssh on
   ops-center-smi-cluster-manager-85869cf9b6-7j64k
   ```

3. Download the latest TAR ball from the URL.

software-packages download *URL*

Example:

```
SMI Cluster Manager# software-packages download <URL>
```

NOTES:

- **software-packages download** *url*—Specify the software packages to be downloaded through HTTP/HTTPS.

4. Verify whether the TAR balls are loaded.

   software-packages list

   Example:

   ```
   SMI Cluster Manager# software-packages list
   [ PCF-2019-08-21 ]
   [ sample ]
   ```

   NOTES:

   - **software-packages list** —Specify the list of available software packages.

5. Update the product repository URL with the latest version of the product chart.

   Note If the repository URL contains multiple versions, the Ops Center selects the latest version automatically.

   ```
   config
     cluster cluster_name
      ops-centers app_name PCF_instance_name
           repository url
            exit
          exit
   ```

   Example:

   ```
   SMI Cluster Manager# config
   SMI Cluster Manager(config)# clusters test2
   SMI Cluster Manager(config-clusters-test2)# ops-centers PCF data
   SMI Cluster Manager(config-ops-centers-PCF/data)# repository <url>
   SMI Cluster Manager(config-ops-centers-PCF/data)# exit
   SMI Cluster Manager(config-clusters-test2)# exit
   ```

   NOTES:

   - **cluster** —Specify the K8s cluster.
   - *cluster_name* —Specify the name of the cluster.
   - **ops-centers** *app_name instance_name* —Specify the product Ops Center and instance. *app_name* is the application name. *instance_name* is the name of the instance.
   - **repository** *url*—Specify the local registry URL for downloading the charts.

6. Run the **cluster sync** command to update to the latest version of the product chart. For more information on **cluster sync** command, see the Important section.

clusters *cluster_name* **actions sync run**

Example:

```
SMI Cluster Manager# clusters test2 actions sync run
```

Important The cluster synchronization updates the PCF Ops Center, which in turn updates the application pods (through **helm sync** command) one at a time automatically.

NOTES:

- **cluster** —Specify the K8s cluster.
- *cluster_name* —Specify the name of the cluster.
- **actions** —Specify the actions performed on the cluster.
- **sync run** —Triggers the cluster synchronization.

Monitoring the Upgrade

You can monitor the status of the upgrade through SMI Cluster Manager Ops Center. To monitor the upgrade status, use the following configurations:

```
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
   end
```

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
```

NOTES:

- **clusters** *cluster_name*—Specify the information about the nodes to be deployed. *cluster_name* is the name of the cluster.
- **actions**—Configures the actions performed on the cluster.
- **sync run**—Triggers the cluster synchronization.
- **sync logs**—Displays the current cluster synchronization logs.
- **sync status**—Displays the current status of the cluster synchronization.
- **debug true**—Enters the debug mode.
- **monitor sync logs** – Monitors the cluster synchronization process.

Important You can view the pod details after the upgrade through CEE Ops Center. For more information on pod details, see Viewing the Pod Details section.

Validating the Upgrade

This section describes the procedures involved in validating the upgrade process.

Viewing the Pod Details

You can view the details of the current pods through CEE Ops Center. To view the pod details, use the following command (in CEE Ops Center CLI):

cluster pods *instance_name pod_name* **detail**

Note

- **cluster pods**—Specify the current pods in the cluster.
- *instance_name*—Specify the name of the instance.
- *pod_name*—Specify the name of the pod.
- **detail**—Displays the details of the specified pod.

The following example displays the details of the pod named *alertmanager-0* in the *PCF-data* instance.

Example:

```
cee# cluster pods PCF-data alertmanager-0 detail
details apiVersion: "v1"
kind: "Pod"
metadata:
  annotations:
    alermanager.io/scrape: "true"
    cni.projectcalico.org/podIP: "<ipv4address/subnet>"
    config-hash: "5532425ef5fd02add051cb759730047390b1bce51da862d13597dbb38dfbde86"
  creationTimestamp: "2020-02-26T06:09:13Z"
  generateName: "alertmanager-"
  labels:
    component: "alertmanager"
    controller-revision-hash: "alertmanager-67cdb95f8b"
    statefulset.kubernetes.io/pod-name: "alertmanager-0"
  name: "alertmanager-0"
  namespace: "PCF"
  ownerReferences:
  - apiVersion: "apps/v1"
    kind: "StatefulSet"
    blockOwnerDeletion: true
    controller: true
    name: "alertmanager"
    uid: "82a11da4-585e-11ea-bc06-0050569ca70e"
  resourceVersion: "1654031"
  selfLink: "/api/v1/namespaces/PCF/pods/alertmanager-0"
  uid: "82aee5d0-585e-11ea-bc06-0050569ca70e"
spec:
  containers:
  - args:
```

```
    - "/alertmanager/alertmanager"
    - "--config.file=/etc/alertmanager/alertmanager.yml"
    - "--storage.path=/alertmanager/data"
    - "--cluster.advertise-address=$(POD_IP):6783"
    env:
    - name: "POD_IP"
      valueFrom:
        fieldRef:
          apiVersion: "v1"
          fieldPath: "status.podIP"
    image: "<path_to_docker_image>"
    imagePullPolicy: "IfNotPresent"
    name: "alertmanager"
    ports:
    - containerPort: 9093
      name: "web"
      protocol: "TCP"
    resources: {}
    terminationMessagePath: "/dev/termination-log"
    terminationMessagePolicy: "File"
    volumeMounts:
    - mountPath: "/etc/alertmanager/"
      name: "alertmanager-config"
    - mountPath: "/alertmanager/data/"
      name: "alertmanager-store"
    - mountPath: "/var/run/secrets/kubernetes.io/serviceaccount"
      name: "default-token-kbjnx"
      readOnly: true
  dnsPolicy: "ClusterFirst"
  enableServiceLinks: true
  hostname: "alertmanager-0"
  nodeName: "for-smi-cdl-1b-worker94d84de255"
  priority: 0
  restartPolicy: "Always"
  schedulerName: "default-scheduler"
  securityContext:
    fsGroup: 0
    runAsUser: 0
  serviceAccount: "default"
  serviceAccountName: "default"
  subdomain: "alertmanager-service"
  terminationGracePeriodSeconds: 30
  tolerations:
  - effect: "NoExecute"
    key: "node-role.kubernetes.io/oam"
    operator: "Equal"
    value: "true"
  - effect: "NoExecute"
    key: "node.kubernetes.io/not-ready"
    operator: "Exists"
    tolerationSeconds: 300
  - effect: "NoExecute"
    key: "node.kubernetes.io/unreachable"
    operator: "Exists"
    tolerationSeconds: 300
  volumes:
  - configMap:
      defaultMode: 420
      name: "alertmanager"
    name: "alertmanager-config"
  - emptyDir: {}
    name: "alertmanager-store"
  - name: "default-token-kbjnx"
    secret:
```

```
      defaultMode: 420
      secretName: "default-token-kbjnx"
status:
  conditions:
  - lastTransitionTime: "2020-02-26T06:09:02Z"
    status: "True"
    type: "Initialized"
  - lastTransitionTime: "2020-02-26T06:09:06Z"
    status: "True"
    type: "Ready"
  - lastTransitionTime: "2020-02-26T06:09:06Z"
    status: "True"
    type: "ContainersReady"
  - lastTransitionTime: "2020-02-26T06:09:13Z"
    status: "True"
    type: "PodScheduled"
  containerStatuses:
  - containerID: "docker://821ed1a272d37e3b4c4c9c1ec69b671a3c3fe6eb4b42108edf44709b9c698ccd"

    image: "<path_to_docker_image>"
    imageID:
"docker-pullable:<path_to_docker_image>@sha256:c4bf05aa677a050fba9d86586b04383ca089bd784d2cb9e544b0d6b7ea899d9b"

    lastState: {}
    name: "alertmanager"
    ready: true
    restartCount: 0
    state:
      running:
        startedAt: "2020-02-26T06:09:05Z"
  hostIP: "<host_ipv4address>"
  phase: "Running"
  podIP: "<pod_ipv4address>"
  qosClass: "BestEffort"
  startTime: "2020-02-26T06:09:02Z"
cee#
```

Verifying the Helm Status

This section describes the procedure involved in verifying the helm status. You need to determine whether the deployed helm chart is listed in the helm list successfully.

To determine the helm status:

1. Run the following on the master node to view the list of deployed helm charts.

   helm list

2. If the helm chart is not found, run the following in the operational mode to view the charts irrespective of their deployment status.

   show helm charts

Verifying the Pods

This section describes the procedure involved in determining the pod and container status after upgrading PCF. You need to ensure that the pods and containers are up and running.

Use the following commands to view the PCF pod logs.

kubectl describe pod *pod_name* **-n** *namespace*

Note If the **Status** column displays the state as *Running*, and the **Ready** column has the same number of containers on both sides of the forward-slash (/), then the pod is healthy and operational.

Rollback the Upgrade

You can rollback the upgrade if you encounter any issues during the upgrade process. This section describes the procedure involved rolling back the upgrade.

Reloading PCF Ops Center Configuration

This section describes the procedure involved in reloading the PCF Ops Center configuration from the backup file.

To reload the PCF Ops Center configuration:

1. Log in to the SMI console as an **ubuntu** user.
2. Untar the backup file created on SMI and move it into a directory.

   Example:

   ```
   ubuntu@popcf-cm01:~$ cd ~/backups && tar –zxf popcf-cfg-backup_110219-053530.tar.gz
   ubuntu@popcf-cm01 :~/backups$
   ```

3. Move the backup configuration file into the newly created **backups** directory.

   Example:

   ```
   ubuntu@popcf-cm01 :~/backups$ cd popcf-cfg-backup_110219-053530
   ubuntu@popcf-cm01 :~/backups/popcf-cfg-backup_110219-053530$
   ```

4. Convert the exported PCF Ops Center configuration into a clean file, which is ready for import.

   Example:

   ```
   ubuntu@popcf-cm01 :~/backups/popcf-cfg-backup_110219-053530$ cat pcfops*.cfg | perl -pe
    's/vendor.*\[(.*)\]/vendor $1/g' | perl -pe 's/(^\s+ips).*\[(.*)\]/$1$2/g' | perl -pe
   's/(\w)\s+(\w)/$1 $2/g' | perl -pe 's/^\s+//g' | grep -v "system mode run" > pcfops.txt
   ubuntu@popcf-cm01 :~/backups/popcf-cfg-backup_110219-053530$
   ```

Updating PCF Ops Center Configuration

This section describes the procedure involved in updating the PCF Ops Center configuration after restoring it. To update the PCF Ops Center configuration:

1. Log in to the master node as an **ubuntu** user.
2. Run the following command to log in to the PCF Ops Center CLI.

   Example:

   ```
   ubuntu@popcf-mas01:~$ ssh -p <port_number> admin@$(kubectl get svc -n $(kubectl get
   namespaces | grep -oP 'pcf-(\d+|\w+)') | grep <port_number> | awk '{ print $3 }')
   admin@<admin_ip_address> password: PCF-OPS-PASSWORD
   Welcome to the pcf CLI on popcf01
   admin connected from <admin_ip_address> using ssh on
   ops-center-pcf-01-ops-center-68dd9f588-htjdf
   ```

3. Paste the contents of the exported PCF configuration file (the **pcfops.txt** file mentioned in this example) in the PCF Ops Center.

   Example:

   ```
   product pcf# config
   Entering configuration mode terminal
   product pcf(config)# <PASTE CONTENTS OF pcfops.txt AND RETURN TO 'config' mode. Don't
   Paste Default Configuration>
   product pcf(config)#
   ```

   Important Fix any sections in the configuration file that did not import properly.

4. Ensure that the helm URLs are inline with the updated PCF image.

   Example:

   ```
   product pcf(config)# helm repository base-repos
   product pcf(config-repository-base-repos)# url <url>
   product pcf(config-repository-base-repos)# exit
   product pcf(config)# k8s registry <registry_url>
   product pcf(config)# commit
   Commit complete.
   product pcf(config)#
   ```

Restoring the Configuration from Back Up

This section describes the procedure involved in restoring all the Policy Builder and CRD configuration files from the backup.

Restoring Policy Builder Configuration

1. Log in to the master node as an **ubuntu** user.
2. Retrieve the Cisco Policy Suite Central URL.

   Example:

   ```
   ubuntu@popcf-mas01:~/backups_09182019_T2141$ kubectl get ing -n $( kubectl get namespaces
    | grep -oP 'pcf-(\d+|\w+)' | cut -d\  -f1) | grep policy-builder | awk '{ print $2
   }'
   pb.pcf-02-pcf-engine-app-blv02.<ipv4address>.nip.io
   ```

3. Navigate to the Cisco Policy Suite Central URL.
4. Log in with your user credentials.
5. Click **Import/Export**.
6. Click **Import** tab.
7. Click **File to Import**.
8. Select the exported policy backed up in the Back Up SVN, Policy, and CRD Data section.
9. In **Import URL**, specify the following URL:

   `http://svn/repos/configuration`

10. Enter a brief description in **Commit Message** text-box.

11. Click **Import**.
12. Log in to the master node as an **ubuntu** user.
13. Run the following command to retrieve the Cisco Policy Builder URL.

    Example:

```
kubectl get ing -n $(kubectl get namespaces | grep -oP 'pcf-(\d+|\w+)' | cut -d\  -f1)
 | grep policy-builder | awk '{ print "https://"$2"/pb" }'
https://pb.pcf-02-pcf-engine-app-blv02.<ipv4address>.nip.io/pb
ubuntu@popcf-mas01:~/backups_09182019_T2141$
```

14. Navigate to the Cisco Policy Builder URL.
15. Click **Build Policies using version controlled data**.
16. Choose **Repository** from the drop-down list.
17. Click **OK**.
18. Log in with your user credentials.
19. Click **File**.
20. Click **Publish to Runtime Environment**.
21. Enter a brief description in **Commit Message**.
22. Click **OK.**

Restoring CRD Data

1. In CPS Central home page, click **Custom Reference Data**.
2. Check the **Export CRD to Golden Repository** check-box.
3. Specify the SVN host name in **Please enter valid server Hostname or IP** text-box.

Note For PCF the SVN host name value is *svn*.

4. Click **+**.
5. Click **Export**.

Note You receive a success message when the data is exported successfully.

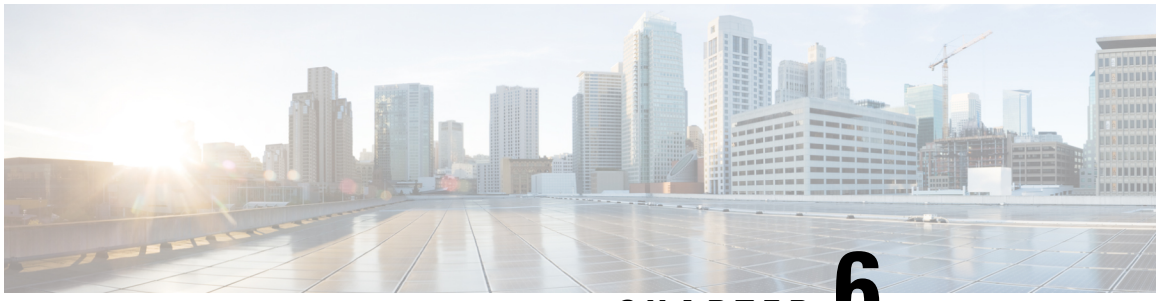
Removing Temporary Files

1. Log in to SMI Cluster Manager as an **ubuntu** user.
2. Delete the temporary directory.

Note Ensure that a copy of the image is stored on OSPD before deleting.

Example:

```
ubuntu@popcf-cm01:~$ ls | grep temp
temp_09192019_T0143
ubuntu@popcf-cm01:~/temp_08072019_T1651$
ubuntu@popcf-cm01:~/temp_08072019_T1651$ rm -f temp_09192019_T0143
ubuntu@popcf-cm01:~/temp_08072019_T1651$
```



CHAPTER 6

3GPP Specification Compliance for PCF Interfaces

Feature Summary and Revision History

Summary Data

Table 7: Summary Data

Applicable Products or Functional Area	PCF
Applicable Platform(s)	SMI
Feature Default Setting	Not Applicable
Related Documentation	Not Applicable

Revision History

Table 8: Revision History

Revision Details	Release
Enhancement introduced. PCF supports N5 Interface.	2022.02.0
First introduced.	2020.02.0

Feature Description

The PCF is compliant with the December 2018 and June 2019 compliance version of 3GPP specification for the PCF interfaces such as N7, N25, N28, and Nnrf. The PCF processes the messages from these interfaces as per the compliance profile configured for the corresponding services.

Currently, IE encoding and decoding are supported. Only the existing features work with the June 2019 specification versions. No additional features in the June 2019 version are supported.

Note The PCF continues to support the older versions of 3GPP specifications and the compliance profile configuration controls the same for the PCF interfaces.

Standards Compliance

The PCF is one of the control plane network functions (NFs) of the 5G core network. The PCF uses different interfaces to communicate with the other NFs or nodes, for example, the N7 interface exists between the SMF and PCF. Each of the PCF interfaces complies with a specific version of 3GPP specification.

Use the following table to determine the compliance mapping of each PCF interface and the 3GPP Standards specification versions.

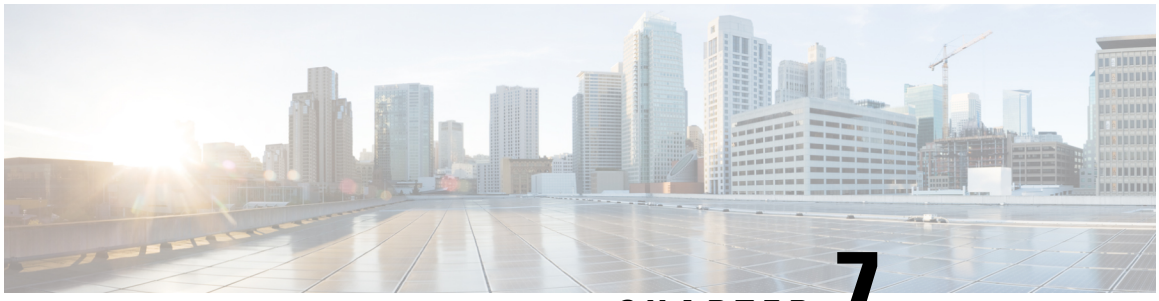
Table 9: Compliance Mapping

Interface	Relationship	3GPP Specification	Version
Rx	Reference point for interworking with AF and PCF.	29.214 Release 15	15.1.0
N5	Reference point between AF and PCF.	29.514 Release 16	16.7.0
N7	Reference point between SMF and PCF.	29.510 Release 15	15.4.0 and 15.2.0
N15	Reference point between AMF and PCF.	29.507 Release 15	15.4.0
N36	Reference point between UDR and PCF	29.519 Release 15	15.4.0
N28	Reference point between PCF and CHF	29.594 Release 15	15.4.0 and 15.2.0
Lightweight Directory Access Protocol (LDAP)	Reference point between PCF and external subscriber profile.	NA	RFC 4511 Lightweight Directory Access Protocol (LDAP)
Nnrf	Reference point between PCF and NRF.	29.510 Release 15	15.4.0 and 15.4.0

Configuring Interfaces and Endpoints

This section describes how to configure the interfaces/endpoints that interact with PCF.

- For configuring the N5, N7, N15, N25, and N28, see Configuring the REST Endpoints, on page 163.
- For configuring the LDAP endpoint, see Configuring the LDAP Endpoint, on page 310.
- Configuring the NRF interface involves the following steps:
  - Configuring the NRF Endpoint for Management Services, on page 196
  - Configuring the NRF Endpoint for Discovery Service, on page 199



CHAPTER 7

Basic Systems Configuration

Feature Summary and Revision History

Summary Data

Table 10: Summary Data

Applicable Product(s) or Functional Area	PCF
Applicable Platform(s)	SMI
Feature Default Setting	Enabled – Configuration required to disable
Related Documentation	Not Applicable

Revision History

Table 11: Revision History

Revision Details	Release
First introduced.	2020.05.01

Overview

The PCF provides the Policy Builder as an interface for policy management. Policies translate a Service Provider’s business rules into actionable, logical processing methods that the PCF enforces on the network.

The PCF provides some standard base policies that creates a starting point for customization to suit a Service Provider's specific business rules.

Adding a System

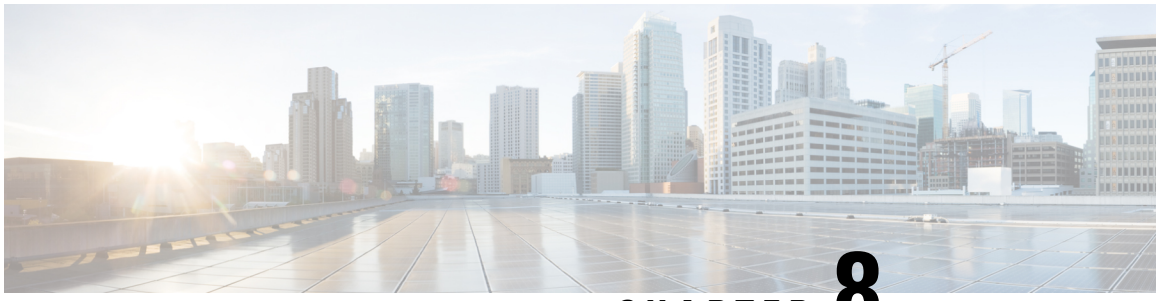
This section describes how to add a system.

After installation, use this procedure to set up your Policy Builder by using an example populated with default data. You can change anything that does not apply to your deployment.

1. Click the **Reference Data** tab, and then click the **Systems** node to display the **Systems** tree.
2. Click **System...** under **Create Child:** to open the **System** pane on the right side.
3. Fill in the **Name** field, and provide a description of this system. Enter the rest of the parameters based on your network requirements.

Table 12: System Parameters

Parameter	Description
Name	The name of the PCF system.
Description	Describes the system using which you can uniquely identify the system.
Session Expiration Hours	An event occurs whenever a session is updated, which in turn increments the session expiry duration. If no session update event occurs in the specified session expiration duration (combination of **Session Expiration Hours** and **Session Expiration Minutes**), then the session will be removed. **Note** The combined value of **Session Expiration Hours** multiplied by 60 plus **Session Expiration Minutes** should not exceed 35,400 minutes. Default value is 8.
Session Expiration Minutes	An event occurs whenever a session is updated, which in turn increments the session expiry duration. If no session update event occurs in the specified session expiration duration (combination of **Session Expiration Hours** and **Session Expiration Minutes**), then the session will be removed. **Note** The combined value of **Session Expiration Hours** multiplied by 60 plus **Session Expiration Minutes** should not exceed 35,400 minutes. Default value is 0.



CHAPTER 8

Cisco Common Data Layer

Feature Summary and Revision History

Summary Data

Table 13: Summary Data

Applicable Product(s) or Functional Area	PCF
Applicable Platform(s)	SMI
Feature Default Setting	Disabled – Configuration required to enable
Related Documentation	Not Applicable

Revision History

Table 14: Revision History

Revision Details	Release
Enhancement introduced. PCF supports N5 Interface.	2022.02.0

Revision Details	Release
Enhancement introduced. PCF can handle issues of multiple CDL entry updates when multiple RxSTR received on PCF within a short gap. Added configuration support for: • Stale sessions cleanup • Remote index synchronization	2021.04.0
First introduced.	2020.01.0

Feature Description

Geographic Redundancy

The PCF extends support to the Geographic Redundancy (GR) version of the Cisco Common Data Layer (CDL). When the highest rated CDL endpoint fails, PCF attempts the same operation on the next highly rated CDL endpoint thus providing a nondisrupted message handling. If the next rated endpoint is unavailable, then PCF reattempts the operation on the subsequent endpoint that has the highest rating and so on.

PCF can handle issues of multiple CDL entry updates when multiple RxSTR received on PCF within a short gap.

Note It is recommended to enable this feature after upgrading both local and remote sites to the latest PCF version.

For more information on the CDL concepts, see the *Ultra Cloud Core Common Data Layer Configuration Guide*.

Limitations

This GR support feature has the following limitations:

- The PCF attempts to reroute the calls only when it encounters gRPC errors such as UNAVAILABLE. It does not acknowledge errors that the datastore returns and actual gRPC timeouts such as DEADLINE_EXCEEDED gRPC status code.
- The PCF Engine does not resolve failures occurring with the datastore such as indexing and slot failures. The CDL layer must resolve these failures and if necessary, send an API call on the remote.

Stale Sessions Cleanup

In the CDL sessions, PCF adds the unique session key SupiDnnKey and the pre-existing unique keys that include FramedIpv6PrefixKey. With the CDL's index overwrite detection command in the PCF Ops Center,

the administrators can configure the ability to delete the old session using the same unique key while the new session is created.

The unique keys that should be used in the overwrite detection configuration are SupiDnnKey and FramedIpv6PrefixKey with the action as delete_record.

Note If two unique keys (one key mapped to the notify action and the other to the delete action) point to the same primary key, then only the notify action is considered for the primary key.

For more information on CDL components, see *Cisco Common Data Layer* documentation.

Limitations

This Stale Sessions Cleanup feature has the following limitations:

- Operations that depend on indexes for the stale sessions require either of the following:
  - The sessions must be present at the same subscriber that is reconnecting with the same DNN.
  - The associated framed IPv6 prefix is assigned to the same or the different subscriber session.

  If the subscriber has reconnected with a different DNN or framed IPv6 prefix is not reassigned to a different session, the sessions are not identified as stale.
- The stale detection and cleanup procedures use the SupiDnnKey values. Indexes of the older session are not created based on the SupiDnnKey values.

  If the stale session is created before an upgrade, and a new session is created for the same SUPI and DNN combination postupgrade, then the older session is not identified as stale.
- If the system has multiple stale sessions with the framed IPv6 prefix key, the corresponding index is associated only with the latest session.

  When a new session is created with then same key then only one session gets associated.

Synchronizing the Index Records

Sometimes after the local site is reinstated, the index data on both the sites may not be consistent. To reconcile the records and eliminate the discrepancy in the sites, configure the sync operation that initiates index data synchronization on the site with its remote peers.

For information on how site isolation works in PCF, see Site Isolation, on page 281.

Note Configuring the sync operation may cause a negative performance impact. It is recommended to perform this operation in a production environment that experiences a high number of inconsistent index records.

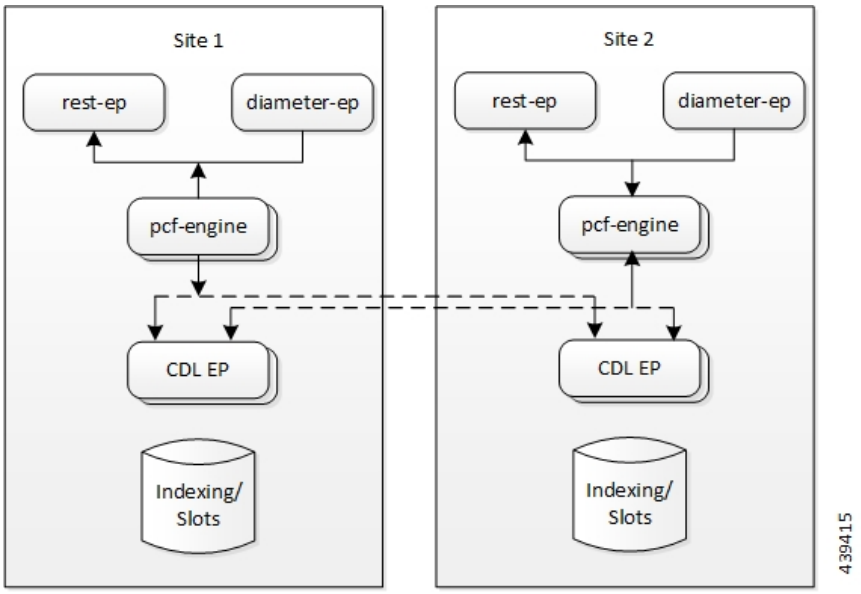
A sync operation cannot be initiated for an index instance where the remote sync is in progress.

Architecture

You can configure CDL through PCF Ops Center. CDL in the GR mode replicates the session data across the configured sites. When PCF connects to the CDL, it always treats the local CDL endpoints as the primary endpoint and the remote endpoints as secondaries (with the appropriate rating). PCF uses the secondary endpoints when the connection to the primary endpoint fails.

The following illustration depicts the failover that happens when the PCF Engine is unable to access the primary CDL datastore endpoint.

Figure 9: CDL Datastore Architecture



How it Works

This section describes how this feature works.

Geographic Redundancy

When you configure the CDL in PCF through the PCF Ops Center, PCF gets enabled to support multiple CDL datastore endpoints. You can configure the endpoints by specifying the IP addresses, port numbers, and assigning ratings to each endpoint. By default, PCF considers the local endpoint as the primary endpoint, which has the highest rating. PCF performs CDL API operations on the primary endpoint. If this endpoint is unavailable, then PCF routes the operations to the next highest rated endpoint. PCF keeps failing over to the accessible secondary endpoint or until all the configured endpoints are exhausted. It does not reattempt a query on the next rated endpoint if the endpoint is reachable but responds with error or timeout.

If PCF is unable to access any of the endpoints in the cluster, then CDL operation fails with the "Datastore Unavailable" error.

When Rx STR or N5 Delete messages are received on two different sites (site A and site B) for the same subscriber session, a conflict occurs while each PCF site tries to update and replicate the session data. In this situation:

- PCF receives notification from CDL with session record from both the sites.
- After receiving the notification from CDL based on the session creation state only one site must processes the notification to resolve the conflict and save the session.

Processing of CDL Conflict Notification

The local and remote sites receive the same CDL conflict notification. The site where session is created will process the notification and the other site ignores the notification.

Note Based on GeoSiteName, PCF identifies whether the session is created at current site or not.

PCF decodes the records (local and remote) into session objects available in the CDL notification.

PCF considers the decoded local session object as a base and checks whether the Rx or N5 SessionIds are available in LastActionList of remote session object. If Rx or N5 SessionIds are available, PCF removes the following from base session object.

- Rx or N5 device session.
- Rx or N5 session tags (secondary keys).
- Rx or N5 session rules.

PCF then saves the modified local session.

Call Flows

This section describes the key call flows for this feature.

CDL Endpoint Failure Call Flow

This section describes the CDL Endpoint Failure call flow.

Figure 10: CDL Endpoint Failure Call Flow

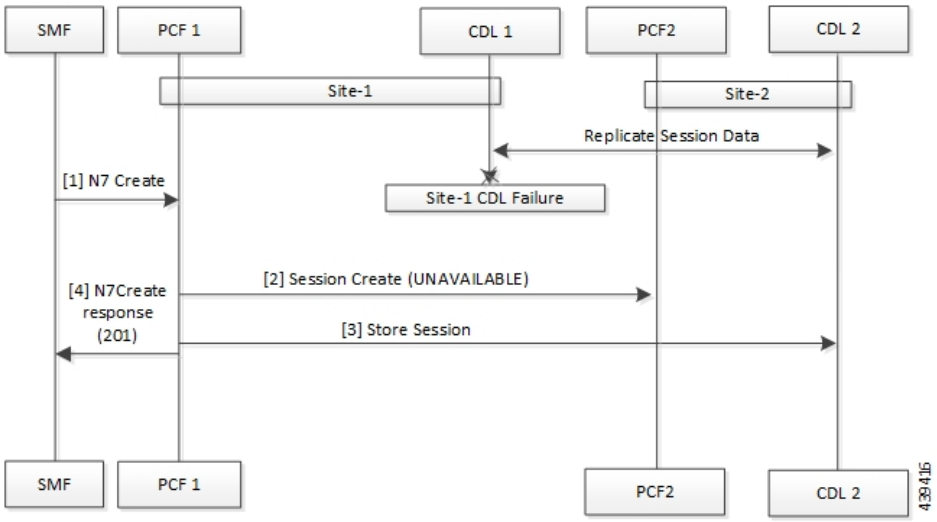


Table 15: CDL Endpoint Failure Call Flow Description

Step	Description
1	In the Site 1 environment, the SMF sends a N7 Create Request to the PCF 1 over the N7 interface.
2	The PCF 1 sends Session Create Request to the PCF 2.
3	The PCF 1 sends a Session Store Request to the CDL2.
4	The PCF 1 sends N7 Create Response to the SMF.

GR Call Flows

This section describes the possible CDL GR mode call flows scenarios that could start a failover to another site.

Indexing Shard Failure Call Flow

This section describes how the failover happens when two index replicas that belong to the same shard are down or unavailable.

The indexing shard failure is an example of two points-of-failure scenario where the two replicas reside on different virtual machines or hosts.

The PCF REST endpoint and PCF Engine redirect the traffic to the secondary CDL endpoint site (Site 2) based on the highest rating when the primary CDL site (Site 1) is unavailable.

Figure 11: Indexing Shard Failure Call Flow

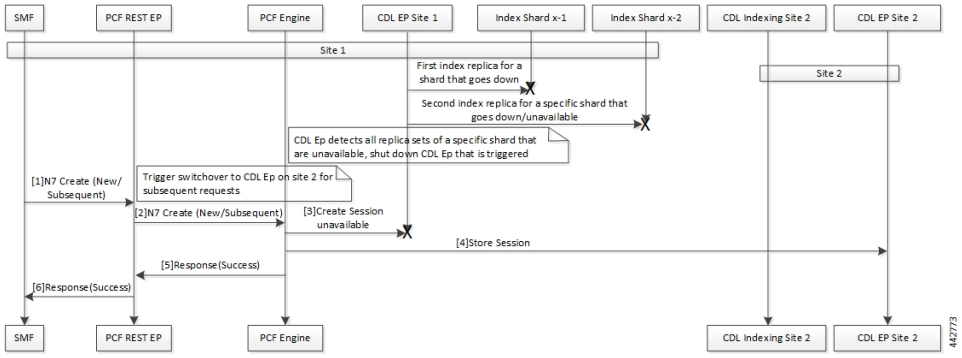


Table 16: Indexing Shard Failure Call Flow Description

Step	Description
1	In the Site 1 environment, index replica 1 and replica 2 for a configured shard has failed or unavailable. Since both the replicas for the shard are unavailable, the CDL endpoint in Site 1 is shut down and all the subsequent requests are directed to the CDL endpoint on Site 2. In the Site 1 environment, the SMF sends a Create Request to PCF REST endpoint over the N7 interface.
2	After receiving the request, the PCF REST endpoint forwards the Create Request to the PCF Engine.
3	The PCF Engine attempts to reach the CDL endpoint to send the Session Create Request. However, the CDL endpoint is unreachable. The PCF Engine sorts the CDL points across Site 1 and Site 2 to recognize the endpoint with the highest rating or priority.
4	The Create Request is evaluated in the stored session and the PCF Engine forwards the request to the CDL endpoint residing in Site 2.
5	After the call request is successful, the PCF Engine notifies the Success Message to the PCF REST endpoint.
6	The PCF REST endpoint forwards the Success Message to the SMF.

Slot Replica Set Failure Call Flow

This section describes how the failover happens when two slot replicas that belong to the same replica set are down or unavailable.

The slot failure is an example of two points-of-failure scenario where the two slot replicas reside on different virtual machines or hosts.

Figure 12: Slot Replica Set Failure Call Flow

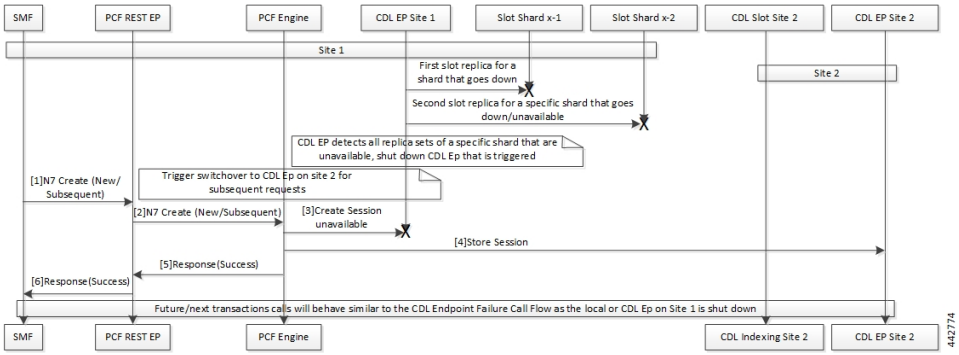


Table 17: Slot Replica Set Failure Call Flow Description

Step	Description
1	In the Site 1 environment, slot replica 1 and replica 2 for a configured shard is down or unavailable. Since both the replicas for the shard are unavailable, the CDL endpoint in Site 1 is shut down and all the subsequent requests are directed to the CDL endpoint on Site 2. In the Site 1 environment, the SMF sends a N7 Create request to PCF REST endpoint over the N7 interface.
2	The PCF REST endpoint receives the request and forwards it to the PCF Engine.
3	The PCF Engine attempts to connect the CDL endpoint to send the Session Create request. If the CDL endpoint is unreachable, the PCF Engine sorts the CDL points across Site 1 and Site 2 to recognize the endpoint with the highest rating or priority.
4	The Create Request is evaluated in the stored session and the PCF Engine forwards the request to the CDL endpoint residing in Site 2.
5	After the call request is successful, the PCF Engine notifies the Success message to the PCF REST endpoint.
6	The PCF REST endpoint forwards the Success message to the SMF.

Local and Remote Sites Receive Rx_STR Without Any Time Gap Call Flow

This section describes the local and remote sites receive Rx_STR without any time gap call flow.

Figure 13: Local and Remote Sites Receive Rx_STR Without Any Time Gap Call Flow

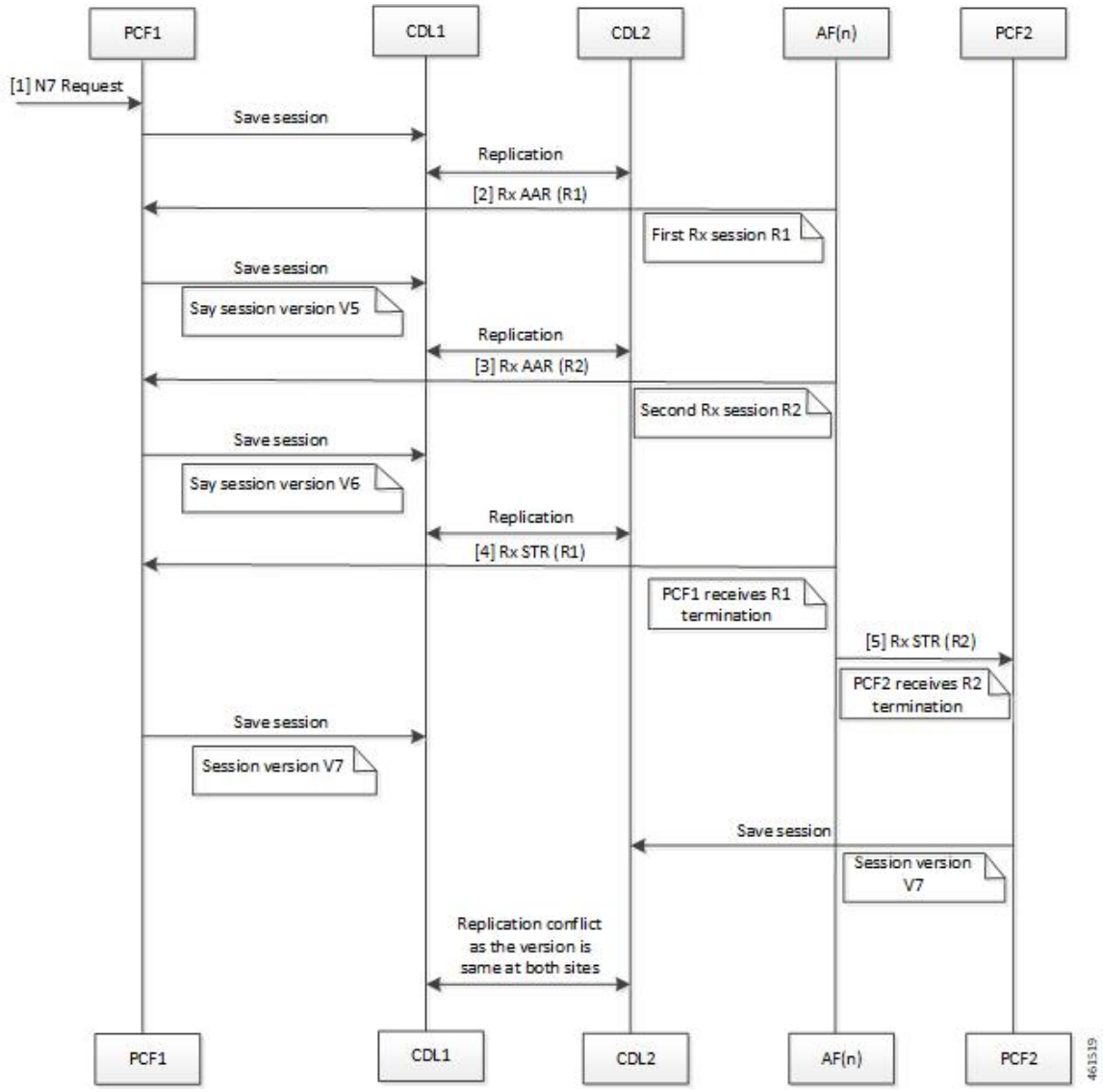


Table 18: Local and Remote Sites Receive Rx_STR Without Any Time Gap Call Flow Description

Step	Description
1	The SMF sends a N7 Create Request to the PCF 1 over the N7 interface.
2	The AF(n) sends a request Rx-AAR (R1) to the PCF 1.
3	The AF(n) sends a request Rx-AAR (R2) to the PCF 1.
4	The AF(n) sends the Rx Session-Termination-Request R1 to the PCF 1.
5	The AF(n) sends the Rx Session-Termination-Request R2 to the PCF 2.

Local and Remote Sites Receive N5 Delete Request Without Any Time Gap Call Flow

This section describes the local and remote sites receive N5 Delete Request without any time gap call flow.

Figure 14: Local and Remote Sites Receive N5 Delete Request Without Any Time Gap Call Flow

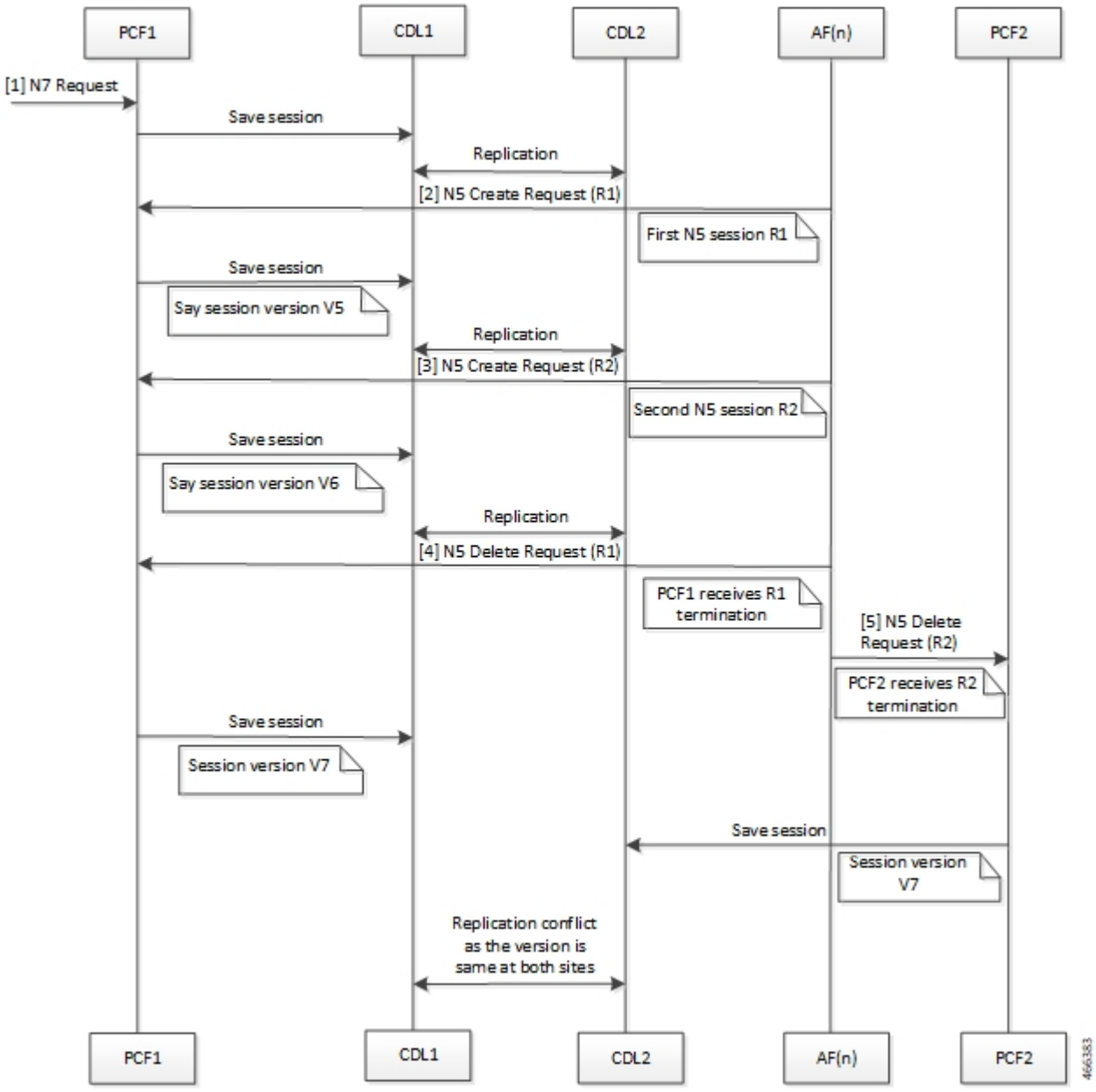


Table 19: Local and Remote Sites Receive Rx_STR Without Any Time Gap Call Flow Description

Step	Description
1	The SMF sends a N7 Create Request to the PCF 1 over the N7 interface.
2	The AF(n) sends a request N5 Create Request (R1) to the PCF 1.

Step	Description
3	The AF(n) sends a request N5 Create Request (R2) to the PCF 1.
4	The AF(n) sends the N5 Delete Request R1 to the PCF 1.
5	The AF(n) sends the N5 Delete Request R2 to the PCF 2.

Configuring Cisco Common Data Layer

This section describes how to configure the CDL endpoints.

Configuring the CDL using PCF Ops Center involves the following steps:

1. Configuring the CDL Session Database and Defining the Base Configuration
2. Configuring Kafka in CDL
3. Configuring Zookeeper in CDL

Configuring the CDL Session Database and Defining the Base Configuration

This section describes how to configure the CDL session database and define the base configuration in PCF.

To configure the CDL session database and define the base configuration in CDL, use the following configuration in the Policy Ops Center console:

```
config
  cdl
    system-id system_id
    node-type node_type
    enable-geo-replication [ true | false ]
    zookeeper replica zookeeper_replica_id
    remote-site remote_system_id
      db-endpoint host host_name
      db-endpoint port port_number
      kafka-server remote_kafka_host1 remote_port1
      kafka-server remote_kafka_host2 remote_port2
      kafka-server remote_kafka_host3 remote_port3
      exit
   cdl logging default-log-level debug_level
      cdl datastore session
      cluster-id cluster_id
      geo-remote-site remote_site_value
      endpoint replica replica_number
      endpoint external-ip ip_address
      endpoint external-port port_number
         index map map_value
         slot replica replica_slot
         slot map map/shards
         slot write-factor write_factor
         slot notification host host_name
```

```
slot notification port port_number
slot notification limit tps
slot notification include-conflict-data [ true | false ]
index replica index_replica
index map map/shards
index write-factor write_factor
end
```

NOTES:

- **system-id** *system_id*—(Optional) Specify the system or Kubernetes cluster identity. The default value is 1.
- **node-type** *node_type*—(Optional) Specify the Kubernetes node label to configure the node affinity. The default value is “session.” *node_type* must be an alphabetic string of 0-64 characters.
- **enable-geo-replication [true | false]** —(Optional) Specify the geo replication status as enable or disable. The default value is false.
- **zookeeper replica** *zookeeper_replica_id*—Specify the Zooker replica server ID.
- **remote-site** *remote_system_id*—Specify the endpoint IP address for the remote site endpoint. Configure this command only when you have set the cdl enable-geo-replication to true.
- **db-endpoint host** *host_name*—Specify the endpoint IP address for the remote site. Configure this command only when you have set the cdl enable-geo-replication to true.
- **db-endpoint port** *port_number*—Specify the endpoint port number for the remote site endpoint. The default port number is 8882. Configure this command only when you have set the cdl enable-geo-replication to true.
- **kafka-server** *remote_kafka_host1 remote_port1*—Specify the Kafka server’s external IP address and port number of the remote site that the remote-system-id identifies. You can configure multiple host address and port numbers per Kafka instance at the remote site. Configure this command only when you have set the cdl enable-geo-replication to true.
- **endpoint replica** *replica_number*—(Optional) Specify the number of replicas to be created. The default value is 1. *replica_number* must be an integer in the range of 1 – 16.
- **endpoint external-ip** *ip_address*—(Optional) Specify the external IP address to expose the database endpoint. Configure this command only when you have set the cdl enable-geo-replication to true.
- **endpoint external-port** *port_number*—(Optional) Specify the external port number to expose the database endpoint. Configure this command only when you have set the cdl enable-geo-replication to true. The default value is 8882.
- **slot replica** *replica_slot*—(Optional) Specify the number of replicas to be created. The default value is 1. *replica_slot* must be an integer in the range of 1 – 16.
- **slot map** *map/shards*—(Optional) Specify the number of partitions in a slot. The default value is 1. *map/shards* must be an integer in the range of 1 – 1024.
- **slot write-factor** *write_factor*—(Optional) Specify the number of copies to be written before successful response. The default value is 1. *write_factor* must be an integer in the range of 0 – 16. Make sure that the value is lower than or equal to the number of replicas.

- **slot notification host** *host_name*—(Optional) Specify the notification server hostname or IP address. The default value is datastore-notification-ep.
- **slot notification port** *port_number*—(Optional) Specify the notification server port number. The default value is 8890.
- **slot notification limit** *tps*—(Optional) Specify the notification limit per second. The default value is 2000.
- **slot notification include-conflict-data [true | false]**—(Optional) Specify whether to receive the original data and the data from the request along with the DB conflict notification. This command is used to send conflict record data from CDL.
- **index replica** *index_replica*—(Optional) Specify the number of replicas to be created. The default value is 2. *index_replica* must be an integer in the range of 1 – 16.
- **index map** *map/shards*—(Optional) Specify the number of partitions in a slot. The default value is 1. *map/shards* must be an integer in the range of 1 – 1024. Avoid modifying this value after deploying the CDL.
- **index write-factor** *write_factor*—(Optional) Specify the number of copies to be written before successful response. The default value is 1. *write_factor* must be an integer in the range of 0 – 16.

Configuring Kafka in CDL

This section describes how to configure Kafka in CDL.

To configure the Kafka in CDL, use the following configuration:

1. Open the Policy Ops Center console and navigate to the datastore CLI.
2. To configure Kafka, use the following configuration:

```
config
   cdl kafka replica number_of_replicas
      enable-JMX-metrics [ true | false ]
      external-ip ip_address  port_number
      enable-persistence [ true | false ]
      storage storage_size
      retention-time retention_period
      retention-size retention_size
      end
```

NOTES:

All the following parameters are optional.

- **cdl kafka replica** *number_of_replicas*—Specify the number of replicas to be created. The default value is 3. *number_of_replicas* must be an integer in the range of 1 – 16.
- **enable-JMX-metrics [true | false]**—Specify the status of the JMX metrics. The default value is true.
- **external-ip** *ip_address port_number*—Specify the external IPs to expose to the Kafka service. Configure this command when you have set the **enable-geo-replication** parameter to true. You are required to define an external IP address and port number for each instance of the Kafka replica. For

example, if the **cdl kafka replica** parameter is set to 3, then specify three external IP addresses and port numbers.

- **enable-persistence [true | false]**—Specify whether to enable or disable persistent storage for Kafka data. The default value is false.
- **storage** *storage_size*—Specify the Kafka data storage size in gigabyte. The default value is 20 GB. *storage_size* must be an integer in the range of 1-64.
- **retention-time** *retention_period*—Specify the duration (in hours) for which the data must be retained. The default value is 3. *retention_period* must be an integer in the range of 1 – 168.
- **retention-size** *retention_size*—Specify the data retention size in megabyte. The default value is 5120 MB.

Configuring Zookeeper in CDL

This section describes how to configure Zookeeper in CDL.

To configure Zookeeper in CDL, use the following configuration:

1. Open the Policy Ops Center console and navigate to the datastore CLI.
2. To configure the parameters, use the following configuration:

```
config
  cdl zookeeper data-storage-size data_storage
    log-storage-size log_storage
    replica number_of_replicas
    enable-JMX-metrics [ true | false ]
    enable-persistence [ true | false ]
    end
```

NOTES:

All the following parameters are optional.

- **cdl zookeeper data-storage-size** *data_storage*—Specify the size of the Zookeeper data storage in gigabyte. The default value is 20 GB. *data_storage* must be an integer in the range of 1-64.
- **log-storage-size** *log_storage*—Specify the size of the Zookeeper data log's storage in gigabyte. The default value is 20 GB. *log_storage* must be an integer in the range of 1-64.
- **replica** *number_replicas*—Specify the number of replicas that must be created. The default value is 3. *number_replicas* must be an integer in the range of 1-16.
- **enable-JMX-metrics [true | false]**—Specify the status of the JMX metrics. The default value is true.
- **enable-persistence [true | false]**—Specify the status of the persistent storage for Zookeeper data. The default value is false.

Sample Configuration

The following is a sample configuration of CDL in the HA environment.

```
cdl system-id   system_i
cdl enable-geo-replication true
cdl zookeeper replica num_zk_replica
cdl datastore session
 endpoint replica ep_replica
index map index_shard_count
 slot replica slot_replica
 slot map slot_shard_count
exit
cdl kafka replica kafka_replica
```

Configuring the CDL Engine

To configure this feature use the following configuration:

```
config
   cdl
      engine
         properties
            enable.conflict.merge [ true | false ]
            GeoSiteName geosite_name
            conflict.tps conflict_number
            conflict.resolve.attempts
            end
```

NOTES:

- **properties**—Specify the system properties.
- **enable.conflict.merge [true | false]**—Specify to enable the feature at application end.
- **GeoSiteName** *geosite_name*—Specify which site notification to be processed.
- **conflict.tps** *conflict_number*—Specify the rate limit of the confliction notification. The default value is considered as '5 tps'.
- **conflict.resolve.attempts**—Specify the number of attempts that application can try to merge the record. The default value is considered as '2 attempts'.

Note The **enable.conflict.merge [true | false]**, **conflict.tps** *conflict_number*, and **conflict.resolve.attempts** are to be configured manually.

Configuring the CDL Endpoints

This section describes how to configure the CDL endpoints.

Configuring the CDL endpoints involves the following steps:

1. Configuring the External Services
2. Associating the Datastore with the CDL Endpoint Service

Configuring the External Services

This section describes how to configure the external services in PCF.

CDL gets deployed in the GR environment as part of the SMI deployment procedure. By default, the CDL endpoints are available in the Datastore CLI node of the PCF Ops Center. However, you are required to configure these endpoints.

For each CDL site and instance, configure external service with the IP address and port number that corresponds to the site and instance.

1. Open the Policy Ops Center console and navigate to the datastore CLI.
2. To configure the parameters, use the following configuration:

```
config
    external-services site_name
    ips ip_address
    ports port_number
    end
```

NOTES:

- **external-services** *site_name*—Specify the CDL site or instance name.
- **ips** *ip_address*—Specify the IP address on which the CDL endpoint is exposed.
- **ports** *port_number*—Specify the port number on which the CDL endpoint is exposed.

Associating the Datastore with the CDL Endpoint Service

This section describes how to configure the external service for each CDL endpoint service that you plan to use.

To configure the external service for each CDL endpoint service, use the following configuration:

1. Open the Policy Ops Center console and navigate to the datastore CLI.
2. To associate the datastore with CDL endpoint service, use the following configuration:

```
config
  datastore external-endpoints service_name
  port port_number
  rating rating_priority
  end
```

NOTES:

- **datastore external-endpoints** *service_name*—Specify the service name that belongs to the external services.
- **port** *port_number*—Specify the port number where the external service resides.

- **rating** *rating_priority*—Specify the rating or priority of the external service. PCF gives preference to the endpoints with the higher ratings.

Starting the Remote Index Synchronization

This section describes how to start the remote index synchronization.

Before configuring the remote index sync, ensure that the geo-remote-site parameter for CDL is configured.

To start the remote index synchronization, use the following configuration:

```
cdl
  actions
    remote-index-sync start [ map-id map_id | slice-name slice_name ]
    end
```

NOTES:

- **cdl**—Enters the CDL configuration mode.
- **remote-index-sync start**—Specify the remote index sync feature.
- **map-id** *map_id* —Specify the index mapID for which the remote index sync procedure should start. By default, remote index sync is initiated for all the index instances.

  Using this parameter you can specify a maximum of 5 mapIDs.
- **slice-name** *slice_name*—Specify the slice name for which the remote index sync procedure should start. By default, remote index sync is initiated for all the sliceNames. There is no limit to the number of sliceNames that you can specify.

Sample Configuration

```
cdl actions remote-index-sync start map-id { 1 } map-id { 2
} slice-name { session-1 } slice-name { session-2 }
```

Viewing the Remote Index Synchronization Status

This section describes how to view the status of the index synchronization procedure that you have executed.

To view the status of the index sync procedure, use the following configuration:

```
cdl
  actions
    remote-index-sync status
    end
```

NOTES:

- **remote-index-sync status**—Displays the status of the index instances for which the syncing with the remote peers is in progress.

Sample Output

```
syncing-instances 'index-mapID-1-instanceID-1, index-mapID-
1-instanceID-2, index-mapID-2-instanceID-1, index-mapID-2-
instanceID-2'
```

Configuring the Stale Session Cleanup Using the Unique Key

The section describes how to configure stale session cleanup using the unique key.

To configure the stale session cleanup, use the following configuration:

```
config
  cdl
    datastore session datastore_name
      features
             index-overwrite-detection [ max-tps | queue-size |
unique-keys-prefix [ SupiDnnKey | FramedIpv6PrefixKey ]
             action [ notify-record | log-record | delete-record ]
             exit
              end
```

NOTES:

- **cdl**—Enter the CDL configuration mode.
- **datastore session** *datastore_name*—Specify the CDL datastore session.
- **index-overwrite-detection [max-tps | queue-size | unique-keys-prefix]**—Configures the index keys overwrite detection capability. The parameter has the following subparameters:
  - **max-tps**—Specify the TPS per cdl-endpoint at which the stale record notification is sent. This parameter is applicable only when the action is "notify-record". The accepted value range is 1..2000. The default value is 200.
  - **queue-size**—Specify the queue size for each cdl-endpoint. The default value is 1000.
  - **unique-keys-prefix [SupiDnnKey | FramedIpv6PrefixKey]**—Specify the list of uniqueKey prefixes for the index overwrite detection and the action that must be performed on successful detection.
- **action [log-record | delete-record]**—Specify the action that must be taken on detecting a stale record.

Note If configuring the stale session cleanup feature for the first time on your system, Cisco recommends performing the configuration after both the GR sites are upgraded to the same software version.

Important The delete-record action on key SupiDnnKey command takes effect only when the required key SupiDnnKey is added in the CDL sessions.

Sample Configuration

The following is a sample configuration:

```
cdl datastore session
features index-overwrite-detection unique-keys-prefix SupiDnnKey
action delete-record
exit
features index-overwrite-detection unique-keys-prefix FramedIpv6PrefixKey
action delete-record
exit
end
```

Stale Sessions Cleanup Troubleshooting Information

To view the status of the clean up status of the stale sessions, review the warning logs in index pods.

You can review the logs to debug the stale sessions issues by setting the index.overwrite.session log to INFO level.

Example:

```
cdl logging logger index.overwrite.session
level info
exit
```

OAM Support

This section describes operations, administration, and maintenance support for this feature.

Statistics

Following are the list of counters that are generated for scenarios where the stale session cleanup process is initiated.

The following metrics track the counter information:

- overwritten_index_records_deleted - Captures the total number of records deleted due to overwritten or duplicate unique keys at index

  Sample query: overwritten_index_records_deleted

  The following labels are defined for this metric:

  - errorCode - The error code in the DB response. Example: 0, 502.
  - sliceName - The name of the logical sliceName. Example: session

- overwritten_index_records_skipped - Captures the total number of records detected as stale, but dropped when the queue is full while processing the records for notify or delete.

  Sample query: overwritten_index_records_skipped

  The following labels are defined for this metric:

  - action - The action that was supposed to be performed for the stale record. Example: delete, notify.

- sliceName - The name of the logical sliceName. Example: session

The following statistics are supported to handle issues of multiple CDL entries updates when multiple Rx STR or N5 Delete received on PCF within a short gap feature:

Note The following values apply to all the statistics:

- Unit - Int64
- Type - Counter
- Nodes - Service

- record_conflict_merge_total - Captures the total count of conflict merge actions.

The following label is defined for this metric:

- action

The "action" label supports the following values:

- ok: Captures success processing of conflict notification.
- submit: Captures the number of messages submitted to the engine when conflict notification is received from CDL.
- retry: Captures the number of retry operations occurred during conflict merge.
- skip_<reason for skip>: Indicates that the PCF is expecting some data validation before the records are merged when CDL notification is received. If that data is missing, PCF logs these skip counters with reason to skip the data.

  Reasons to skip the data are:

  - throttle: Due to throttle check.
  - feature_disabled: Feature is disabled.
  - no_geositename: GeoSiteName is not configured.
  - flag1_mismatch: Mismatch of CDL flag 1 at both sites.
  - unsupported_record: Invalid data records of CDL notification.
  - retry_unsupported_record: Invalid data records available during retry.
  - unsupported_lastaction: Invalid lastAction objects are available in notification records.
  - retry_ unsupported_lastaction: During retry lastAction objects are invalid.
  - no_sessionid - Session id is not available in remoteLastAction object.
  - retry_no_sessionid: Session id is not available in remoteLastActoin object.
  - no_remotesessionid: Remote session id is not available to remove in local record object.

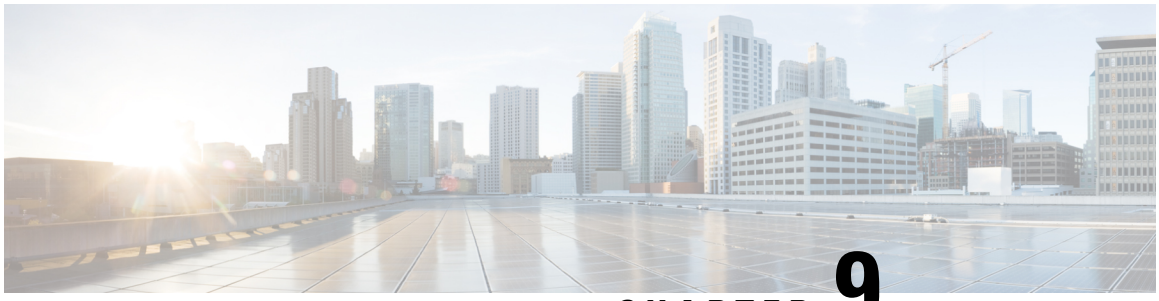
- retry_no_remotesessionid: Remote session id not available to remove in local record object during retry.
- attemptsdone: Total number of attempts completed.

- error_<cause of error>: Indicates while merging the records some error/exception occurred in that case pcf logs this error related counter.

  Following are the types of cause of errors:

  - deletesession: Remote rx session delete operation from the local record failed.
  - retry_deletesession: During retry remote rx session delete operation from the local record failed.
  - removeflags: after deleting remote rx session from local record while removing corresponding flags from local record failed.
  - addactionlist: Failed to consolidated all action list objects from local and remote records.
  - nopk: Primary key is not available in the record.
  - retry_nopk: Primary key is not available in the record during retry.

For information on statistics, see *Ultra Cloud Core Common Data Layer Configuration and Administration Guide*.



CHAPTER 9

Configuring HTTP or HTTPS and SSL for SBA Interface

Feature Summary

Summary Data

Table 20: Summary Data

Applicable Product(s) or Functional Area	PCF
Applicable Platform(s)	SMI
Feature Default Setting	Enabled – Configuration required to disable
Related Documentation	Not Applicable

Revision History

Table 21: Revision History

Revision Details	Release
First introduced.	2020.03.0

Feature Description

In the SBA framework, the PCF exchanges data across the interconnected network functions and data repositories. The communication within this framework is established over a secure layer comprising of Hypertext Transfer Protocol (HTTP) or HTTPS using Transport Layer Security (TLS). TLS offers a secure network layer transportation of data between the components. However, PCF also offers support for HTTPS without TLS.

In this release, TLS provides a transport layer encryption between the nodes for the security compliance purposes. This feature does not support the NF security requirements as per the 3GPP specifications of 5G.

The PCF provides HTTP or HTTPS support for the N7, N15, N28, N36, and NRF interface. The information transmission between the client and server happens through the HTTPS requests.

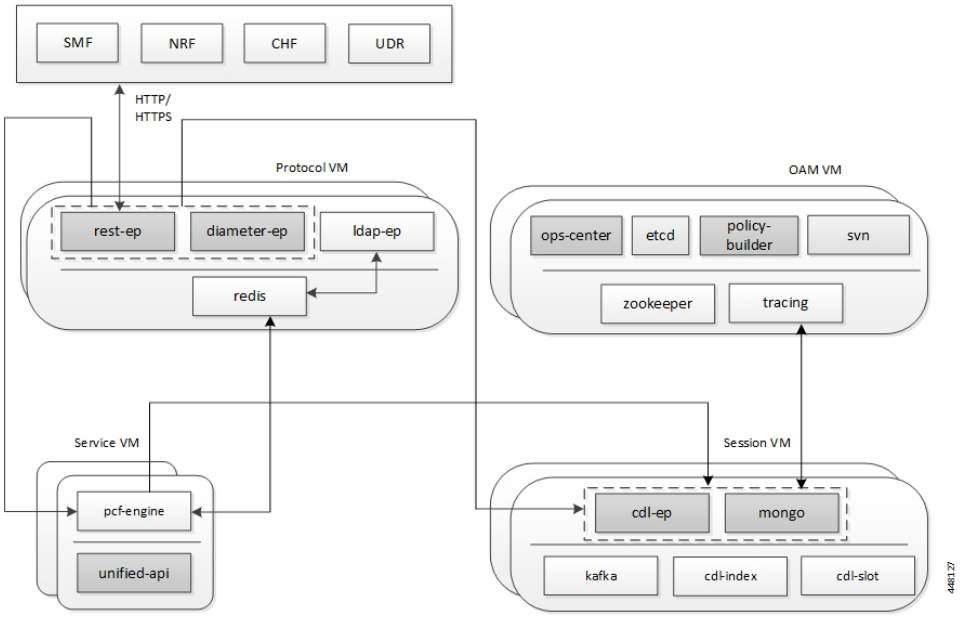
How it Works

This section describes how this feature works.

The implementation of HTTP or HTTPS with TLS in PCF requires you to configure the HTTP and HTTPs secure port for each interface. To enable a TLS handshake, import the signed certificate into the PCF Ops Center from a trusted source. The PCF supports both server and client HTTPS requests. By default, the PCF supports HTTP requests without TLS.

The following graphic illustrates the communication flow between NFs and REST endpoint.

Figure 15: HTTP or HTTPS Communication Flow



The support for HTTP and HTTPS in PCF involves the following steps:

1. Configure the ca-certificates which are required for TLS. The certificate data must be in the PEM format and residing in the Java KeyStore (JKS).

2. Configure the certificate and private key for establishing the TLS channel between the server and client. Obtain the private key from the certificate.
3. By default, the uri-scheme is associated with the HTTP. Enable HTTPS by associating the rest-endpoint uri-scheme with the HTTPS. PCF invokes the configured server certificate when starting up the pcf-rest-ep pod. This step ensures that the SSL context is set for the REST server. When PCF is a client that initiates N28, nNRF, or the N15 requests, the HTTP or HTTPS protocol is specified in the endpoint profile.

   The rest-endpoint server detects all the certificates from the Kubernetes secrets during a startup. An individual Kubernetes secret is created for each certificate. These secrets are mounted on the rest-endpoint pods, at /config/secrets location during its deployment. All the certificates are loaded into the keystore that is located at /opt/workspace/rest-ep/certs/server/keystore. If the HTTPS is configured as the uri-scheme, then the HTTP server initiates the SSL context with the certificate name configured. For messages initiated from the REST endpoint (PCF as client), the HTTP client loads all the certificates from the keystore.

Configuring Support for HTTP or HTTPS and TLS

This section describes how to configure the HTTP or HTTPS and TLS from the PCF Ops Center.

Configuration of HTTP or HTTPS and TLS involves:

- Configuring Server and Client Certificates

Configuring Server and Client Certificates

This section describes how to configure the certificates for the server and client.

To configure the certificates for the server and client, use the following configuration in the PCF Ops Center:

```
config
   pcf-tls
     ca-certificates [name]
       cert-data certificate_pem
     certificates [name]
       cert-data certificate_pem
       private-key certificate_private_key
       end
```

NOTES:

- **ca-certificates [name]** —Specify the certificate name. The list of certificates names is displayed based on the configured certificates.
- **certificates [name]**—Specify the certificate name. The list of certificates names is displayed based on the configured certificates.
- **cert-data** *certificate_pem*—Specify the cert-data value in the PEM format.
- **private-key** *certificate_private_key*—Specify the private key value in the Public-Key Cryptography Standards (PKCS) #8 format.

Obtaining the Private key

This section describes how to obtain the private key from a certificate.

To obtain the private key, perform the following procedure:

1. Convert the certificate from PEM to PKCS12 format using the following:

   openssl pkcs12 -export -out *pkcscertificate.p12* **-inkey** *certificatekey.pem* **-in** *inputcertificate.pem*

2. Extract the private key from the PKCS12 certificate created in the previous step by using the following:

   openssl pkcs12 -in *pkcscertificate.p12* **-nocerts -nodes -out** *privatekey.pem*

3. Convert the private key to a PKCS8 key using the following:

   openssl pkcs8 -in *privatekey.pem* **-topk8 -nocrypt -out** *privatekey.p8*

Verifying the Certificate Status

This section describes how to verify the configuration status of the certificates.

The following configuration is a sample output of the **show rest-endpoint certificate-status** command:

```
CERTIFICATE
NAME         TIME TO EXPIRE
---------------------------------------------
pcfserver    3649 days 10 hours 25 minutes
cacert       3610 days 13 hours 55 minutes
pcfclient    334 days 21 hours  26 minutes
```

HTTP and SSL for SBA Interface OA&M Support

This section describes the operations, administration, and maintenance information for this feature.

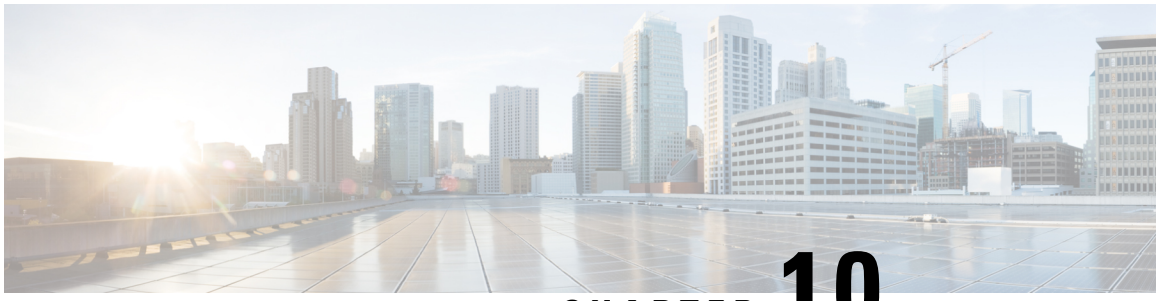
Statistics

This section provides gauge that is generated for computing the HTTP TLS certificate validity information.

- http_tls_cert_validity: This gauge fetches the duration (in milliseconds) after which the certificate expires. The cert_name label fetches the certificate name.

  An example of the Prometheus query:

  ```
  abs(http_tls_cert_validity)>0
  ```



CHAPTER 10

Content Filtering

Feature Summary and Revision History

Summary Data

Table 22: Summary Data

Applicable Products or Functional Area	PCF
Applicable Platform(s)	SMI
Feature Default Setting	Disabled – Configuration required to enable
Related Documentation	Not Applicable

Revision History

Table 23: Revision History

Revision Details	Release
First introduced.	2020.01.0

Feature Description

PCF offers fine-grained control over the content that SMF processes. The filtering policy provides methodical control over the content during the PCF and SMF interaction by identifying and limiting the access to inappropriate content.

Configuration Support for Content Filtering

This section describes how to configure the filtering policy using the following service:

- CiscoContentFilteringPolicy

CiscoContentFilteringPolicy

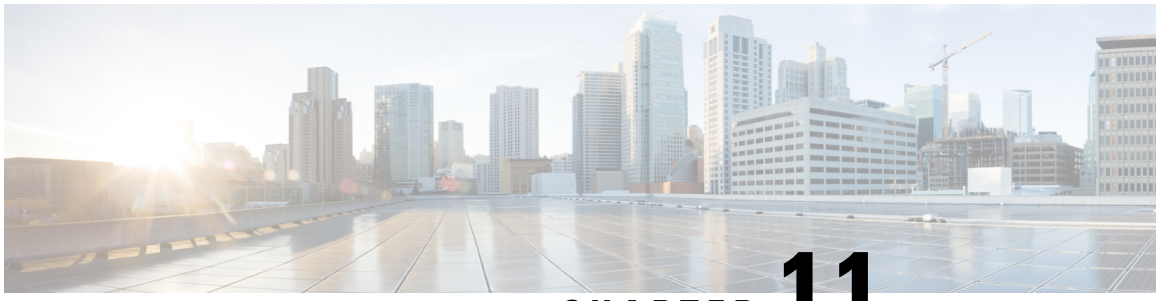
This section describes the parameters for the CiscoContentFilteringPolicy configuration.

Before configuring the CiscoContentFilteringPolicy service, ensure that you have created the use case templates and added the CiscoContentFilteringPolicy service. Use case templates are the building blocks of the PCF architecture. The use case templates allow you to define the Service Configuration objects set by a Service Option.

For information on how to create a use case template and add a service for this configuration, see Configuring the Use Case Template, on page 293 and Adding a Service, on page 292.

Table 24: CiscoContentFilteringPolicy Parameters

Parameters	Description
Priority	Indicates the priority of the service configuration object to be used in case multiple service initiator conditions match.
Cisco Content Filtering Policy	Specifies the policy ID that PCF filters when transmitting content.



CHAPTER 11

Diameter Endpoint

Feature Summary and Revision History

Summary Data

Table 25: Summary Data

Applicable Products or Functional Area	PCF
Applicable Platform(s)	SMI
Feature Default Setting	Enabled – Always-on
Related Documentation	Not Applicable

Revision History

Table 26: Revision History

Revision Details	Release
Enhancement introduced. PCF supports dual stack (IPv4 and IPv6) connectivity on its external interfaces/endpoints.	2022.01.0
Enhancement introduced. PCF supports IPv6 connectivity on its external interfaces/endpoints.	2021.04.0
First introduced.	Pre 2020.01.0

Feature Description

You can enable the Diameter endpoint to dynamically create pods on a designated node or host. This feature might be a requirement when you want to ensure that the nodes are meeting specific security and regulatory parameters, or the node is closer to the datacenter in terms of geographical proximity. The node affinity determines the node where PCF creates the Diameter endpoint pods, which are based on the affinity towards a node or group of nodes. Node affinity is a set of rules that allows you to define the custom labels on nodes and specify the label selectors within the pods. Based on these rules, the scheduler determines the location where the pod can be placed.

Note If you do not specify a node, then the Kubernetes scheduler determines the node where the Diameter endpoint creates a pod.

PCF supports both IPv4 and IPv6 connectivity on its external interfaces/endpoints (inbound and outbound).

Configuring the Node for the Diameter Endpoint Pod

This section describes how to specify the node or host where the Diameter endpoint must spawn the pod.

Note Configuration changes to the diameter endpoint cause the endpoint to restart automatically. Cisco recommends making such changes only within the maintenance window.

To specify the node where you want Diameter endpoint to spawn the pod, use the following configuration:

```
config
   diameter group diameter_group_name
   mode server server_name
   stack stack_name
     application application_name
     bind-ip ipv4 host_address
     bind-ipv6 ipv6 host_address
     bind-port port_number
     fqdn fqdn_address
     realm realm_address
     node-host node_host_address
     end
```

NOTES:

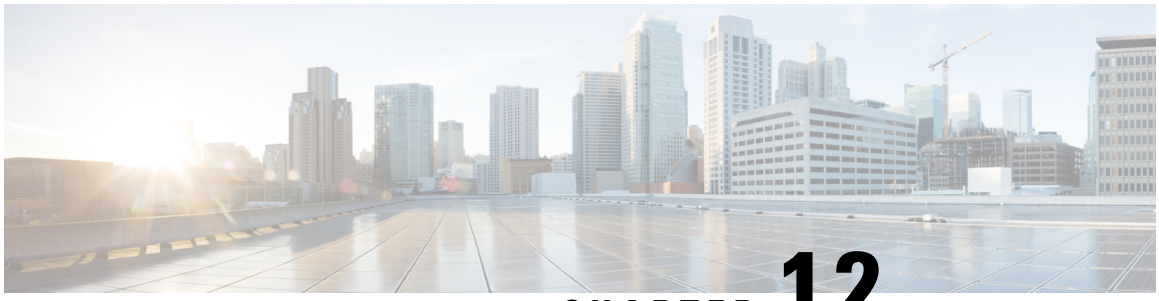
- **diameter group** *diameter_group_name*—Specify the Diameter group name.
- **mode server** *server_name*—Specify the server name that operates as the mode server.
- **stack** *stack_name*—Specify the stack name.
- **application** *application_name*—Specify the application name.

- **bind-ip** *host_address*—Specify the host address IPv4 to bind the stack.
- **bind-ipv6** *host_address*—Specify the host address IPv6 to bind the stack.
- **bind-port** *port_number*—Specify the port number to bind the stack.
- **fqdn** *fqdn_address*—Specify the FQDN address.
- **realm** *realm_address*—Specify the realm address.
- **node-host** *node_host_address*—Specify the host IP address of the node.

Sample Configuration

The following is a sample configuration of the node configuration.

```
mode server
  stack cicdsite
  application rx
  bind-ip 192.0.2.18
  realm cisco.com
  node-host for-node-2a-worker39e1587354h
  exit
```



CHAPTER 12

Dummy N7 Notify Request

Feature Summary and Revision History

Summary Data

Table 27: Summary Data

Applicable Products or Functional Area	PCF
Applicable Platform(s)	SMI
Feature Default Setting	Disabled – Configuration required to enable
Related Documentation	Not Applicable

Revision History

Table 28: Revision History

Revision Details	Release
Enhancement introduced. PCF supports N5 Interface.	2022.02.0
First introduced.	2020.01.0

Feature Description

PCF is equipped to retrieve the event triggers from the SMF by sending the N7 Notify Request. Certain features such as Rx or N5 bearer authorization, QoS derivation, and rule evaluation depends on specific event triggers such as RAT-Type and AccessType for taking the appropriate action. For instance, the RAT-Type event trigger lets you determine the Radio Access Technology (RAT) that is serving the User Equipment.

Typically, PCF does not subscribe to all the event triggers during the session initiation using N7 Create. If the features are dependent on specific event triggers and PCF did not subscribe to them, then PCF does not fetch the values associated with those events. With the dummy N7 Notify Request, you can send an intermediate request to fetch the event triggers corresponding to the configured Media-Type which is specified in the Rx AAR or N5 Create message from the IMS.

How it Works

This section describes how this feature works.

In the Dummy N7 Notify Request feature, the PCF interactions happen in the following sequence:

1. The SMF sends an N7 Create Request to PCF. The PCF responds to this request with the configured event triggers.
2. When IMS initiates multimedia calls containing the AAR message with the Media-Type, PCF initiates an N7 Notify Request to assign the PCC rules that are evaluated based on the media details received in the AAR message. However, in some situations PCF cannot compute the rules as it did not subscribe to the specific event triggers such as RAT-Change and AccessType AVP.
3. The PCF attempts to determine the corresponding event trigger from the SMF by sending an intermediate (dummy) N7 Notify Request to SMF.
4. The SMF responds with the applicable event triggers that are specified in the dummy N7 request.
5. The PCF uses these triggers to compute the PCC rules and transmits it to the SMF in the subsequent N7 Notify Request.

Configuration Support for the Dummy N7 Notify Request

This section describes how to configure the dummy Notify Request through which PCF retrieves the triggers for the AAR messages with Media-Type.

The configuration of the proxy N7 Notify Request involves the following steps:

1. Creating the STG for the N7 Notify Request
2. Configuring the Dummy N7 Notify Parameters
3. Configuring the Event Triggers

Creating the STG for the N7 Notify Request

This section describes how to create the STG N7 Notify request which is referred by the CRD.

1. Log in to Policy Builder.
2. Click the **Reference Data** tab, and from the left pane click **Custom Reference Data Tables** to view the options.
3. On the left pane, click the **Search Table Groups** folder. A default folder is created under the Search Table Groups folder.
4. Expand the default folder and select the table icon to view the Custom Reference Data Table parameters on the right pane. A default STG is created under the **Search Table Groups** folder.
5. Enter the parameters in the **Custom Reference Data Table** pane. Rename the CRD with a unique name.
6. Navigate to the **Column** field and click **Add**. In the **Columns** pane, click the row to enter the **Name**, **Display Name**, and **Type**. Select the **Use In Condition**, **Key**, and **Required** check box.
7. Specify a row for the Media-Type and a row for the event triggers that you want to fetch.

   The event triggers row does not require the **Use In Condition**, **Key**, and **Required** check box to be selected.
8. Save and publish the changes.

Configuring the Dummy N7 Notify Parameters

This section describes how to configure the dummy N7 Notify event trigger parameters through the PCF Central.

1. Log in to PCF Central.
2. Select the **Custom Reference Data**.
3. In the **Custom Reference Data Tables** pane, click the table that you have created in Creating the STG for the N7 Notify Request, on page 91.
4. In the dialgox box, click **Add Row** to include the Media-Type and event trigger. The information is populated based on the configured STG table.
5. Click **Done** to save your changes.

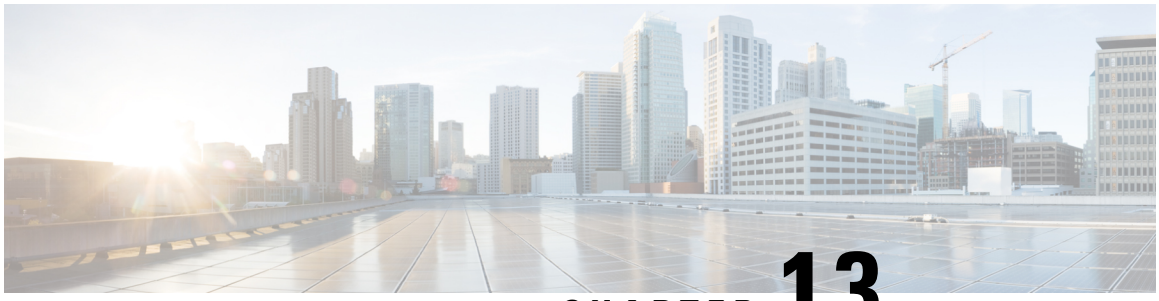
Configuring the Event Triggers

This section describes how to subscribe to the N7 event triggers through the dummy N7 Notify request.

1. Log in to Policy Builder.
2. Click the **Reference Data** tab.
3. In the left pane, choose **Diameter Defaults > Rx Profiles**.
4. In the **Rx Profiles Summary** pane, under **Create Child**, click **Rx Profile**.

5. On the left pane, choose **Rx Profiles > default**. In the **Rx STG lookup binding** pane, rename the default profile name.
6. In the **Stg Reference** field, click **select** to select the STG table that you have configured.
7. In the **List Of Input Column Avp Pairs** section, click **Add**.
   a. Click the row in the Avp Name to specify the media type.
   b. In the **Column** column, hover the cursor on the first row to click the ellipsis (...) and select the media type that you specified in the STG. For more information, see Creating the STG for the N7 Notify Request, on page 91.
8. In the **List Of Output Column Avp Pairs** section, click **Add**.
   a. Click the row in the Avp Name to specify the event trigger.
   b. In the **Column** column, hover the cursor on the first row to click the ellipsis (...) and select the event trigger that you specified in the STG. For more information, see Creating the STG for the N7 Notify Request, on page 91.
9. In the left pane, navigate to the **Diameter Client > Summary**. In the **Summary** pane, choose **Rx Client > default**.
10. In the **Rx Client** pane, navigate to **Request Gx RAA for Event-Triggers section** and select the check box in the corresponding row.
11. In the **Rx CRD AVP name to extract Event-Triggers** section, click **Add**.
12. In the **Add Values** dialog box, add the triggers. Specify the same values that you have entered for the **Diameter Defaults** parameters.

    The events in the list are populated based on the values of the CRD table (dummy notify table). The input columns of the CRD table are associated to the Rx media details such as Media-Type, and the output columns indicate the event trigger numbers that are to be subscribed or enabled on the SMF.



CHAPTER 13

Dynamic ARP Functionality for PC and PV

Feature Summary and Revision History

Summary Data

Table 29: Summary Data

Applicable Product(s) or Functional Area	PCF
Applicable Platform(s)	SMI
Feature Default Setting	Enabled – Always-on
Related Documentation	Not Applicable

Revision History

Table 30: Revision History

Revision Details	Release
Enhancement introduced. PCF supports N5 Interface.	2022.02.0
First introduced.	2021.04.0

Feature Description

PCF supports the dynamic ARP feature to send the same Priority-Level value in the dedicated bearers as that of the default bearer.

The dynamic ARP functionality is extended to Preemption Capability (PC) and Preemption Vulnerability (PV).

The PC parameter defines whether a bearer with a lower priority level can be dropped to free up the required resources.

The PV parameter defines whether a bearer is applicable for such dropping by a preemption capable bearer with a higher priority value.

To support this functionality for Rx interface, add two new columns, Rx_Dynamic_Vulnerability and Rx_Dynamic_Capability to the Rx_QoS_Table and for N5 interface, add two new columns, N5_Dynamic_Vulnerability and N5_Dynamic_Capability to the N5_QoS_Table.

How it Works

This section describes how this feature works.

For a WPS user, the default bearer ARP value includes a Priority-Level value with PC set to enabled and PV set to disabled.

In case, when a non-WPS user calls a WPS user in Rx interface, the dynamic ARP attribute in the Rx_QoS_Table initiates the PCF to set the Priority-Level value in the dedicated bearer rules to match that of the default bearer value. But the PVI/PCI values sent in the dedicated bearer rules use the enforced values from the Rx_QoS_Table (typically PVI enabled, PCI disabled).

In case, when a non-WPS user calls a WPS user in N5 interface, the dynamic ARP attribute in the N5_QoS_Table initiates the PCF to set the Priority-Level value in the dedicated bearer rules to match that of the default bearer value. But the PVI/PCI values sent in the dedicated bearer rules use the enforced values from the N5_QoS_Table (typically PVI enabled, PCI disabled).

For WPS user, if dynamic ARP attribute for PVI and PCI is set to "D", then the PVI and PCI values will be mirrored from the default bearer instead using the configured Rx QoS Table values in Rx interface and configured N5 QoS Table values in N5 interface.

Configuring CRD Table and RxSTGConfiguration AVP

Configuring CRD table and RxSTGConfiguration AVP involves the following steps:

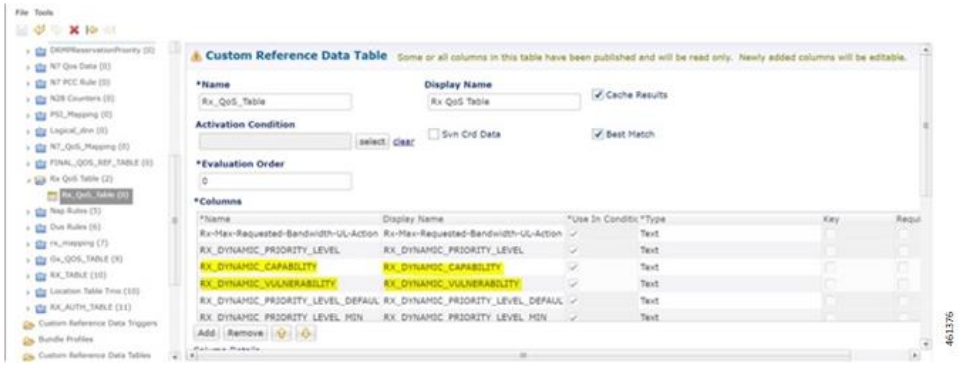
Adding Rx_Dynamic_Capability and Rx_Dynamic_Vulnerability

To add Rx_Dynamic_Capability and Rx_Dynamic_Vulnerability columns to the Rx_QoS CRD table, use the following steps:

1. Log in to Policy Builder.

2. Click the **Reference Data** tab, and from the left pane click **Custom Reference Data Tables** to view the options.
3. On the left pane, expand the **Search Table Groups** folder.
4. Expand the **Rx_QoS_Table** sub folder of **Search Table Groups** and click the.**Rx_QoS_Table**
5. Go to the ***Columns** field and click the **Add**.
6. Add the column **Name** and **Display Name** as **RX_DYNAMIC_CAPABILITY** and **RX_DYNAMIC_VULNERABILITY**.

Figure 16: Adding Rx_Dynamic_Capability and Rx_Dynamic_Vulnerability



Configuring RxSTGConfiguration AVP

This section describes the parameters that can be configured for RxSTGConfiguration.

The RxSTGConfiguration service configuration supports the following output AVPs that allow the dynamic value expression.

Before setting the service parameters, ensure that you create a use case template and add a service for this configuration. For details, see Configuring the Use Case Template, on page 293 and Adding a Service.

The following table describes the RxSTGConfiguration service parameter.

Table 31: RxSTGConfiguration ParameterD

Parameters	Description
Dynamic-QoS-ARP-Pre-Emption-Capability	If the value is configured as "D" then the feature is enabled for PC. If the value is configured with any other value except "D" or is empty then the feature is disabled for PC.
Dynamic-QoS-ARP-Pre-Emption-Vulnerability	If the value is configured as "D" then the feature is enabled for PV. If the value is configured with any other value except "D" or is empty then the feature is disabled for PV.

Configuring CRD Table and N5STGConfiguration AVP

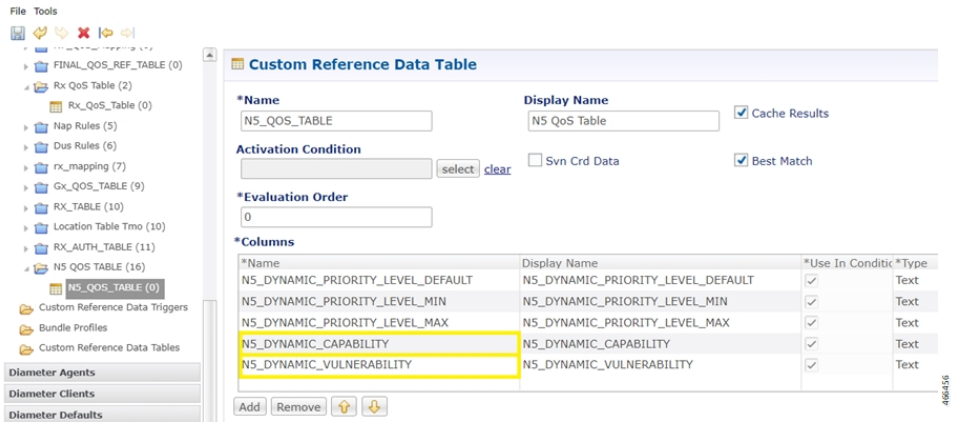
Configuring CRD table and N5STGConfiguration AVP involves the following steps:

Adding N5_Dynamic_Capability and N5_Dynamic_Vulnerability

To add N5_Dynamic_Capability and N5_Dynamic_Vulnerability columns to the N5_QoS CRD table, use the following steps:

1. Log in to Policy Builder.
2. Click the **Reference Data** tab, and from the left pane click **Custom Reference Data Tables** to view the options.
3. On the left pane, expand the **Search Table Groups** folder.
4. Expand the **N5_QoS_Table** sub folder of **Search Table Groups** and click the **N5_QoS_Table**.
5. Go to the ***Columns** field and click the **Add**.
6. Add the column **Name** and **Display Name** as **N5_DYNAMIC_CAPABILITY** and **N5_DYNAMIC_VULNERABILITY**.

Figure 17: Adding N5_Dynamic_Capability and N5_Dynamic_Vulnerability



Configuring N5STGConfiguration AVP

This section describes the parameters that can be configured for N5STGConfiguration.

The N5STGConfiguration service configuration supports the following output AVPs that allow the dynamic value expression.

Before setting the service parameters, ensure that you create a use case template and add a service for this configuration. For details, see Configuring the Use Case Template, on page 293 and Adding a Service.

The following table describes the N5STGConfiguration service parameter.

Table 32: N5STGConfiguration ParameterD

Parameters	Description
Dynamic-QoS-ARP-Pre-Emption-Capability	If the value is configured as "D" then the feature is enabled for PC. If the value is configured with any other value except "D" or is empty then the feature is disabled for PC.
Dynamic-QoS-ARP-Pre-Emption-Vulnerability	If the value is configured as "D" then the feature is enabled for PV. If the value is configured with any other value except "D" or is empty then the feature is disabled for PV.

OAM Support

This section describes operations, administration, and maintenance support for this feature

Bulk Statistics Support

The following statistics are supported for the dynamic ARP functionality for PC and PV feature.

Note The following values apply to all the statistics:

- Unit - Int64
- Type - Counter
- Nodes - Service

- qos_rule_pc_total - Indicates the number of N5/N7/Rx rule installs (per qci/Media Type) provisioned with dynamic QoS PCI.

  The following labels are defined for this metric:

  - Interface
    - N5
    - N7
    - Rx
  - type
    - default_qos_pc
    - dynamic_qos_pc
  - identifier
    - qci

- media-type
- arp_pc

- qos_rule_pv_total - Indicates the number of N5/N7/Rx rule installs (per qci/Media Type) provisioned with dynamic QoS PVI.

  The following labels are defined for this metric:

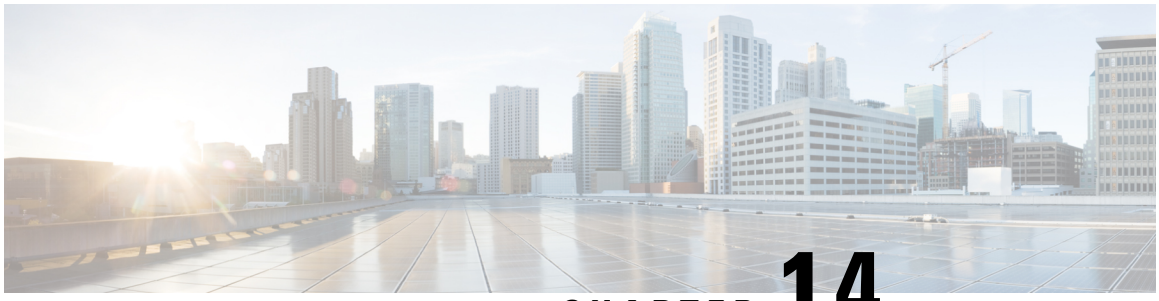
  - Interface
    - N5
    - N7
    - Rx
  - type
    - default_qos_pv
    - dynamic_qos_pv
  - identifier
    - qci
    - media-type
  - arp_pv

Modified Stats

Table 33: Modified Stats

Old Stats	New Stats	Description

qos_rule_total	qos_rule_pl_total	Indicates the number of N5/N7/Rx rule installs (per qci/Media Type) provisioned with dynamic QoS PL.
		The following labels are defined for this metric:
		• Interface
		  • N5
		  • N7
		  • Rx
		• type
		  • default_qos_pl
		  • dynamic_qos_pl
		• identifier
		  • qci
		  • media-type
		• arp_pl



CHAPTER 14

Dynamic ARP Functionality for PL

Feature Summary and Revision History

Summary Data

Table 34: Summary Data

Applicable Product(s) or Functional Area	PCF
Applicable Platform(s)	SMI
Feature Default Setting	Enabled – Always-on
Related Documentation	Not Applicable

Revision History

Table 35: Revision History

Revision Details	Release
First introduced.	2022.02.0

Feature Description

PCF supports Dynamic QoS ARP feature to calculate ARP (Priority Level) based on dynamic expression.

How it Works

This section describes how this feature works.

When PCF evaluates Rx_QoS_Table or N5_QoS_Table to derive QoS for dedicated bearer PCC rules, if a dynamic value expression is configured for ARP Priority-Level, then PCF evaluates the expression and set the result as Priority-Level.

Feature Configuration

To configure this feature, use N5STGConfiguration for Dynamic QoS ARP

Configuring N5STGConfiguration for Dynamic QoS ARP

This section describes the parameters that can be configured for N5STGConfiguration.

The N5STGConfiguration service configuration supports the following output AVPs that allow the dynamic value expression and their ranges to be defined.

Before setting the service parameters, ensure that you create a use case template and add a service for this configuration. For details, see Configuring the Use Case Template, on page 293 and Adding a Service, on page 292.

The following table describes the N5STGConfiguration service parameters.

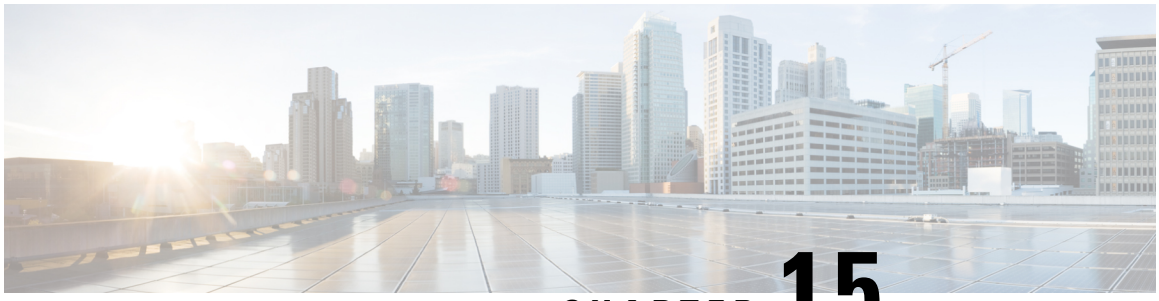
Table 36: N5STGConfiguration Parameters

Parameters	Description
Dynamic-QoS-ARP-Priority-Level	**Note** This is a mandatory parameter if the Dynamic QoS ARP feature is enabled. This AVP is bound to the dynamic expression Priority-Level column. If the value is null/not configured, then Dynamic QoS ARP feature is disabled. If the value is configured, it overrides the integer PL value (if configured). The dynamic PL expression is either expected to match the java regex: `^[dD](\\s*([+-/*])\\s*([0-9]+))?$` or must be an offset value (of syntax: [+-][0-9]+). In case the value is provided in offset form, the “D” is implicit. Thus “+8” corresponds to “D+8” in expression form, “-5” corresponds to “D-5” and similarly, “0” corresponds to “D”.
Dynamic-QoS-ARP-Priority-Level-Default	If the default bearer doesn’t have a Priority-Level, this value is used as dedicated bearer PL. If the value is null/not configured, the default value (15) is used.

Parameters	Description
Dynamic-QoS-ARP-Priority-Level-Min	This output AVP provides upper/lower bound for the calculated PL value using the Dynamic expression provided under Dynamic-QoS-ARP-Priority-Level. If the value is null/not configured, the default value (1) is used.
Dynamic-QoS-ARP-Priority-Level-Max	The upper end of the valid PL range. If the value is null/not configured, the default value (15) is used.
Dynamic-QoS-Update-On-Change	This AVP controls whether the PCC rules must be updated on change in the dynamic PL value (for example, due to change in default bearer PL value). If value is null/not configured, the PCC rules are not updated with new dynamic PL value once installed.

Note

- Using the offset form may have minor performance gains as compared to a full expression.
- Range limits are not applied for the default dynamic values.
- Dynamic expression has an implicit “Enforce” QoS action. The Action column value is ignored.
- If dynamic expression configured for Priority-Level is invalid, PCF ignores the expression and does not include the ARP parameters (since PL is set as null) in the rule install. This is true even if absolute PL value is configured (absolute value is ignored).



CHAPTER 15

Dynamic Rules and Table-Driven Charging Rules

Feature Summary and Revision History

Summary Data

Table 37: Summary Data

Applicable Product(s) or Functional Area	PCF
Applicable Platform(s)	SMI
Feature Default Setting	Enabled – Configuration required to disable
Related Documentation	Not Applicable

Revision History

Table 38: Revision History

Revision Details	Release
First introduced.	Pre 2020.01.0

Feature Description

PCF supports the provisioning of the following dynamic and table-driven charging rules.

- Table-driven dynamic PCC rules in PCF
- N7 session retrievers:
  - SUPI

- GPSI
- DNN
- PLMN ID
- N7 Access Type
- N7 Cell Global Identifier
- N7 DNN
- N7 GPSI
- N7 IMEI TAC
- N7 MCC (SUPI Based)
- N7 MNC (SUPI Based)
- N7 Permanent Equipment Identifier
- N7 RAT Type
- N7 Serving Network
- N7 SliceInformation
- N7 SUPI
- N7 Tracking Area Identifier

Standards Compliance

This feature complies with the following standards specifications:

- *3GPP TS 29.512 V15.1.0 (2018-09) "Session Management Policy Control Service"*
- *3GPP TS 29.571 V15.1.0 (2018-09) "Common Data Types for Service Based Interfaces"*

Restrictions

The values configured for the maxbrUl, maxbrDl, gbrUl, and gbrDL attributes under QosData and TableDrivenQosDecision service configuration objects as well as any other attribute configured in Policy Builder that corresponds to an attribute defined as having the BitRate data type must match the format that is described in *3GPP TS 29.571, Table 5.5.2-1: Simple Data Types*.

Use the following pattern in Policy Builder to validate the format:

'^\d+(\.\d+)? (bps|Kbps|Mbps|Gbps|Tbps)$'

Configuration Support for Dynamic and Table-Driven Charging Rules

This section describes how to configure the dynamic and table-driven charging rules using the following services:

- TableDrivenQosDecision
- TableDrivenDynamicPccRule

TableDrivenQosDecision

The TableDrivenQosDecision service configuration object provides a way for the different refQosData values that are encountered while adding the PCC rules to be expanded to actual QosData objects.

The different refQosData are added to a bucket, the duplicates (if any) are eliminated, and the QosData objects are added for all the PCC rules that are added or updated. The addition happens even if a PCC rule having the same refQosData value is removed. A one-time CRD lookup is executed for each QosData object using a refQosData as a key value.

Note

- Do not use the QosData service configuration object and TableDrivenQosDecision service configuration object in the same policy if there are overlapping QoS references.
- Since the actual QoS attributes are stored in a CRD table, it is assumed they do not change over time. However, if the values change in the CRD, the new values are going to be pushed next time when the policy gets evaluated. Changing the values in the CRD does not automatically trigger a policy update.

Before setting the service parameters, ensure that you create a use case template and add a service for this configuration. For details, see Configuring the Use Case Template, on page 293 and Adding a Service, on page 292.

The following table describes the TableDrivenQosDecision service parameters.

Table 39: TableDrivenQosDecision Parameters

Parameter	Description
Priority	The priority assigned for this service configuration object (among similar service configuration objects) is used for policy evaluation by the Policy Engine. Higher the value, higher is its priority. Default: 0
Search Table	CRD table that is queried for the PCC rule data.
Search Column	Primary key column in the table configured under the Search Table field. **Note** The search value corresponding to this column is passed in the code and not exposed.

Parameter	Description
QoS Id Source	Primary key value for the column configured under Search Column.
5qi Source	Additional primary key column-value pairs in the table configured under the Search Table field.
Maxbr Ul Source	Maxbr Ul column in the table that is configured under the Search Table field corresponding to the maxbrUl attribute. The values that are allowed for this attribute are specified in *3GPP TS 29.571*, *Table 5.5.2-1: Simple Data Types*. **Note** The values that are provided for this attribute must match the specific format. See the Restrictions, on page 106 section for more details.
Maxbr Dl Source	Maxbr Dl column in the table that is configured under the Search Table field corresponding to the maxbrDl attribute. The values that are permitted for this attribute are specified in *3GPP TS 29.571*, *Table 5.5.2-1: Simple Data Types*. **Note** The values that are provided for this attribute must match the specific format. Refer the *Restrictions* section for more details.
Gbr Ul Source	Gbr Ul column in the table that is configured under the Search Table field corresponding to the gbrUl attribute. The values that are allowed for this attribute are specified in *3GPP TS 29.571*, *Table 5.5.2-1: Simple Data Types*. **Note** The values that are provided for this attribute must match the specific format. Refer the *Restrictions* section for more details.
Gbr Dl Source	Gbr Dl column in the table configured under the Search Table field corresponding to the gbrDl attribute. The values that are allowed for this attribute are specified in *3GPP TS 29.571*, *Table 5.5.2-1: Simple Data Types*. **Note** The values that are provided for this attribute must match the specific format. Refer the *Restrictions* section for more details.
Priority Level Source	Priority Level Source column in the table configured under the Search Table field corresponding to priorityLevel attribute.
Preempt Cap Source	Preempt the Cap Source column in the table configured under Search Table field corresponding to the preemptCap attribute. The values that are allowed for this attribute are specified in *3GPP TS 29.571*, section *5.5.3.1 Enumeration: PreemptionCapability*.
Preempt Vuln Source	Preempt the Vuln Source column in the table that is configured under the Search Table field corresponding to the preemptVuln attribute. The values that are allowed for this attribute are specified in *3GPP TS 29.571*, section *5.5.3.2 Enumeration: PreemptionVulnerability*.
Qnc Source	Indicates whether the notifications are requested from the 3GPP NG-RAN when the GFBR can no longer (or again) be guaranteed for a QoS Flow during the lifetime of the QoS Flow.
Authorized Qos Priority Level	Indicates a priority in scheduling the resources among the QoS Flows.

Parameter	Description
Aver Window Source	Indicates the duration over which the guaranteed and maximum bitrate is calculated.
Max Data Burst Vol Source	Indicates the largest amount of data that is required to be transferred within a period of 5G-AN PDB.
Reflective QoS Source	Indicates applying reflective QoS for the SDF.
Sharing Key DI Source	Indicates resource sharing in downlink direction with the service data flows having the same value in their PCC rule.
Sharing Key UI Source	Indicates resource sharing in an uplink direction with the service data flows having the same value in their PCC rule.
Max Packet Loss Rate DI Source	The maximum rate for lost packets that can be tolerated in the downlink direction for the service data flow.
Max Packet Loss Rate UI Source	The maximum rate for lost packets that can be tolerated in the uplink direction for the service data flow.
Def QoS Flow Indication Source	Indicates that the dynamic PCC rule shall always have its binding with the default QoS Flow.

TableDrivenDynamicPccRule

This section describes the parameters for the TableDrivenDynamicPccRule configuration.

TableDrivenDynamicPccRule service configuration object provides a mapping between the PCC rule attributes and the CRD table that backs the service. A one-to-many relation is supported between the service configuration object and the PCC rules.

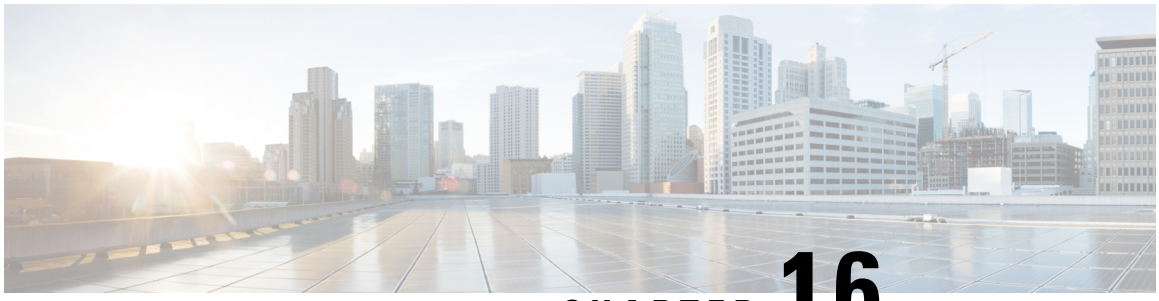
Before setting the service parameters, ensure that you create a use case template and add a service for this configuration. For details, see Configuring the Use Case Template, on page 293 and Adding a Service, on page 292.

The following table describes the TableDrivenDynamicPCCRule service parameters.

Table 40: TableDrivenDynamicPCCRule Parameters

Parameter	Description
Priority	The priority assigned for this service configuration object (among similar service configuration objects) is used for policy evaluation by the Policy Engine. Higher the value, higher is its priority. Default: 0
Search Table	The CRD table that is to be queried for the PCC rule data.
Search Column	Primary key column in the table configured under the Search Table field.
Search Value	Primary key value for the column configured under Search Column.

Parameter	Description
Input List (List)	Additional primary key column-value pairs in the table configured under the Search Table field.
PCC Rule Id Source	PCC Rule Id column in the table that is configured under the Search Table field corresponding to the pccRuleId attribute.
Precedence Source	Precedence column in the table that is configured under the Search Table field corresponding to the precedence attribute.
App Id Source	App Id column in the table that is configured under the Search Table field corresponding to the appId attribute.
QoS Id Source	QoS Id column in the table configured under the Search Table field corresponding to refQosData attribute. Per *3GPP TS 29.512 v15.1.0*, refQosData can be an array of string objects. To accommodate multiple string values in the CRD, the following convention is used: • The different refQosData objects are separated by ",". • Any blank characters before and after the actual data is dropped.
Chg Id Source	The value must be bound to the Chg Id column in the STG. The value in the STG column must be of Type Text.
Flow Information Source	Flow Information column in the table configured under the Search Table field corresponding to flowInfos attribute. Per *3GPP TS 29.512 v15.1.0*, flowInfos can be an array of FlowInformation objects. To accommodate multiple FlowInformation values in the CRD, the following convention is used: • The different FlowInformation objects are separated by ",". • The different attributes within each FlowInformation object are separated by ";". • ethFlowDescription attribute within FlowInformation is not currently supported. • The expected format for each FlowInformation attribute is as follows: `flowDescription;packetFilterUsage;tosTrafficClass;spi;flowLabel;flowDirection`. • If any of the FlowInformation is missing, leave the corresponding placeholder empty while preserving the format (for example tosTrafficClass and spi are missing: flowDescription;packetFilterUsage;;;flowLabel;flowDirection) • Any blank characters before and after the actual data is dropped. • The values that are allowed for flowDirection attribute are the ones that are specified in *3GPP TS 29.512*, section *5.6.3.3 Enumeration: FlowDirection*.



CHAPTER 16

Flexible QoS Actions

Feature Summary and Revision History

Summary Data

Table 41: Summary Data

Applicable Products or Functional Area	PCF
Applicable Platform(s)	SMI
Feature Default Setting	Enabled – Configuration required to disable
Related Documentation	Not Applicable

Revision History

Table 42: Revision History

Revision Details	Release
First introduced.	2020.01.0

Feature Description

PCF supports the Flexible QoS Actions feature on the N7 interface. During policy management on the N7 interface, PCF sends session rules that are based on the configured QoS to the SMF.

The following QoS actions are applicable to an uplink AMBR and downlink AMBR are configurable for calculating the session rules. The default configuration is QoS-Enforcement.

- QoS-Bounding facilitates PCF to calculate the minimum QoS between the Requested QoS (from the SMF) and the calculated QoS based on the internal logic, and authorize that in the response message to the SMF.
- QoS-Mirroring is the ability for the PCF to grant or authorize the requests from the SMF.
- QoS-Enforcement is the ability for the PCF to enforce the calculated QoS (computed based on PCF's internal logic) back to the SMF in the request or response message. This is the default configuration.

Configuring QoS Actions on N7 Interface

This section describes how to configure the QoS Actions on the N7 interface using the following service.

- OverrideSessionRule

OverrideSessionRule

This section describes the parameters for the OverrideSessionRule configurations.

The OverrideSessionRule service configuration is used to override the N7 default bearer QoS APN AMBR UL/DL values.

PCF first evaluates the derived QoS values for default bearer and then assesses the table provided in OverrideSessionRule service configuration using the key values. It further determines the result APN AMBR UL/DL values. If the "Condition to Override" is "LT", then PCF limits the derived QoS values with these override values. If the "Condition to Override" is "GT", then PCF selects the maximum UL/DL among the derived values and overrides these values.

Note The OverrideSessionRule configuration works in conjunction with the SessionRuleAction configuration.

For information on SessionRuleAction, see SessionRuleAction, on page 319.

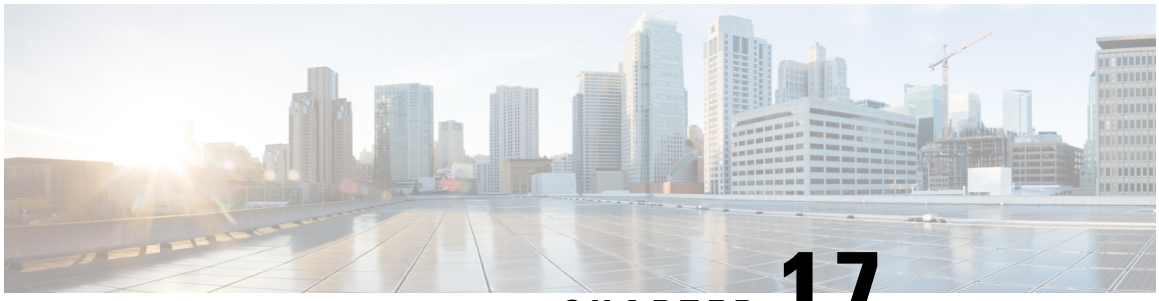
Before setting the service parameters, ensure that you create a use case template and add a service for this configuration. For details, see Configuring the Use Case Template, on page 293 and Adding a Service, on page 292.

The following table describes the OverrideSessionRule service parameters.

Table 43: OverrideSessionRule Parameters

Parameters	Description
Priority	Indicates the priority of the server when sending requests. Higher number is equal to higher priority.
Stg Reference	Refers to the STG that contains the QoS reference and the QoS parameter values such as QCI and APN-MBRUL.

Parameters	Description
List Of Input Column Avp Pairs (List)	The list that specifies the mapping for input (key) columns to determine their values. Based on these values, the STG is queried. ColumnAndAvpPair • Avp Name: Specify the AVP name whose value is used to map to the corresponding key Column for querying the STG. • Column: The key column in the STG that corresponds to the specified AVP.
Apn Agg Max Bit Rate U L	Reference to the STG output column that gives the "APNAggregate-Max-Bitrate-UL" value for limiting QoS. This value and the corresponding value derived after QoS actions are compared to determine the final value for APN-Aggregate-Max-Bitrate-UL.
Apn Agg Max Bit Rate D L	Reference to the STG output column that gives the "APNAggregate-Max-Bitrate-DL" value for limiting QoS. This value and the corresponding value derived after QoS actions are compared to determine the final value for APN-Aggregate-Max-Bitrate-DL.
Condition to Override	Provides the condition to compare the values. Only two values are supported "LT" and "GT". If LT is selected, PCF uses the lowest QoS parameter value from the two QoS references. If GT is selected, PCF considers the highest QoS parameter value from the two QoS references. Possible Values: • LT: Less than (Default) • GT: Greater than



CHAPTER 17

Handling the Network Provided Location Information Requests

Feature Summary and Revision History

Summary Data

Table 44: Summary Data

Applicable Products or Functional Area	PCF
Applicable Platform(s)	SMI
Feature Default Setting	Enabled – Always-on
Related Documentation	Not Applicable

Revision History

Table 45: Revision History

Revision Details	Release
Enhancement introduced. PCF supports N5 Interface.	2022.02.0
First introduced.	2020.05.0

Feature Description

The Network Provided Location Information (NPLI) service is responsible for retrieving the access network information in the IMS network architecture. Depending on the service operator's policy configuration and subscription, the NPLI service fetches the UE time zone information and the user location information from the access network.

The PCF provides the NPLI information over the Rx or N5 interface to the Application Function (AF) based on the response that it receives from SMF over the N7 interface.

How it Works

This section describes how this feature works.

The AF initiates a request toward the PCF to provide the network information.

For AF supporting Rx interface, the request is sent over Rx through the Required-Access-Info AVP. When the Access Network Information is available the SMF provides the required Access Network Information to the PCF within the 3GPP-User-Location-Info AVP or 3GPP-MS-TimeZone AVP or both as requested.

For AF supporting N5 interface, the request is sent over N5 interface by subscribing to AF event ANI_REPORT and specifying the required access network information (user location or user time zone information).

Upon receiving the request, PCF triggers an N7 Update Notify request with ‘Access Network Info’ event trigger (if not already subscribed for) towards SMF. The SMF responds to PCF with the required information, which PCF further forwards to the AF.

When the SMF responds with ServingNetwork attribute instead of UserLocationInfo, then to set the Mobile Country Codes (MCC) and Mobile Network Code (MNC) ensure that the NetLoc features is enabled. For information on how to enable the NetLoc, see Enabling the NetLoc Feature, on page 129 .

For Rx interface, PCF provides the following information during an ACCESS_NETWORK_INFO_REPORT event trigger within the Event-Trigger AVP:

- 3GPP-User-Location-Info AVP (If available)
- User-Location-Info-Time AVP (If available)
- 3GPP-SGSN-MCC-MNC AVP (If the location information is not available) or 3GPP-MS-TimeZone AVP or both.

For N5 interface, PCF provides the following information which includes notification for AF event ANI_REPORT:

- 3GPP User Location Information (If available and required)
- Serving Network Identity (If user location is required and not available)
- UE Timezone (If available and required)

Considerations

This section defines the considerations that apply for successful handling of the NPLI requests:

- For Rx Interface, navigate to **Policy Builder > Diameter Clients > Rx Client**, set the **STA Hold Time Ms** parameter to maximum duration of 3000 milliseconds. The parameter indicates the duration by which the STA is held back.
- For N5 Interface, navigate to **Policy Builder > SBA Profiles > N5 Profiles**, set the **N5 Delete Response Hold Time Ms** parameter to maximum duration of 3000 milliseconds. The parameter indicates the duration by which the N5 Delete response is held back.

A lower timer value minimizes the performance impact that occurs when AF and PCF continue to wait for a response from each other and eventually timeout.

Call Flows

This section describes the key call flows for this feature.

NPLI in Rx RAR Call Flow

This section describes the NPLI in Rx RAR call flow.

Figure 18: NPLI in Rx RAR Call Flow

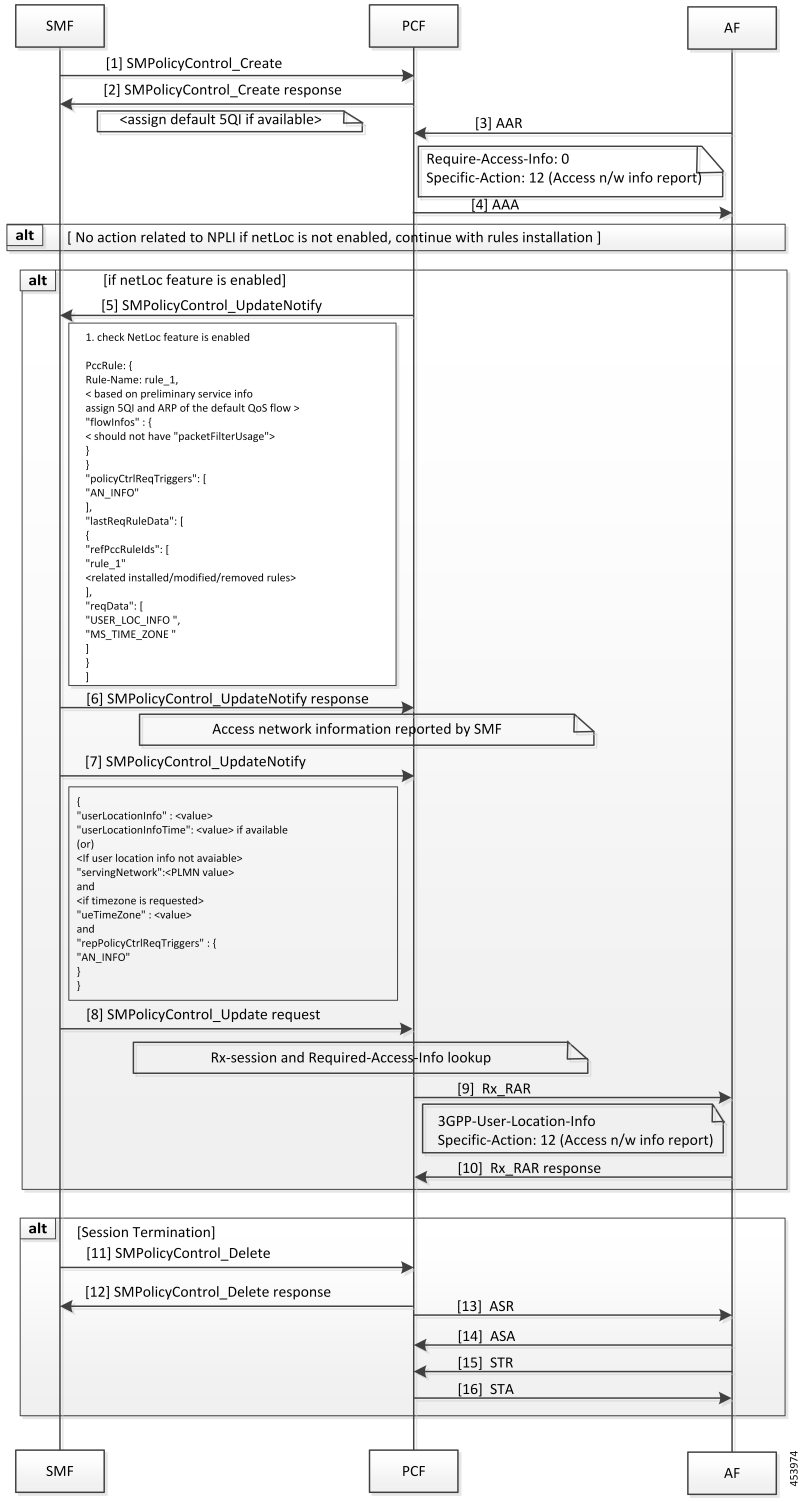


Table 46: NPLI in Rx RAR Call Flow Description

Step	Description
1	The SMF sends a SMPolicyControl_Ceate request to the PCF.
2	The PCF responds to the SMPolicyControl_Create request.
3	The AF sends an Authenticate-Authorize-Request (AAR) message to the PCF. The message contains Required-Access-Info AVP requesting the access network information required for the AF session.
4	The PCF sends the AAA request to the AF.
5	If the NetLoc feature is enabled, then the PCF sends an SMPolicyControl_UpdateNotify request toward the SMF.
6	In response to the SMPolicyControl_UpdateNotify request, the SMF sends the access network information to the PCF.
7	The PCF sends the SMPolicyControl_Update request to the SMF.
8	The SMF sends the SMPolicyControl_Update request to the PCF.
9	After the establishing the Rx-session and the Required-Access-Info lookup, the PCF sends the Rx Re-Authorization Request message to the AF.
10	The AF sends the Rx Re-Authorization Request response containing the 3GPP-User-Location-Info AVP and access network information report to the PCF.
11	If the session terminates, the SMF sends a SMPolicyControl_Delete request to the PCF.
12	The PCF responds to SMF for the SMPolicyControl_Delete request.
13	The PCF sends the Abort-Session-Request message to the AF.
14	The AF responds with the Abort-Session-Answer to the PCF.
15	The AF sends the Session-Termination-Request to the PCF.
16	The PCF responds with the Session-Termination-Answer message to the AF.

NPLI in Rx STA Call Flow

This section describes the NPLI in Rx STA call flow.

Figure 19: NPLI in Rx STA Call Flow

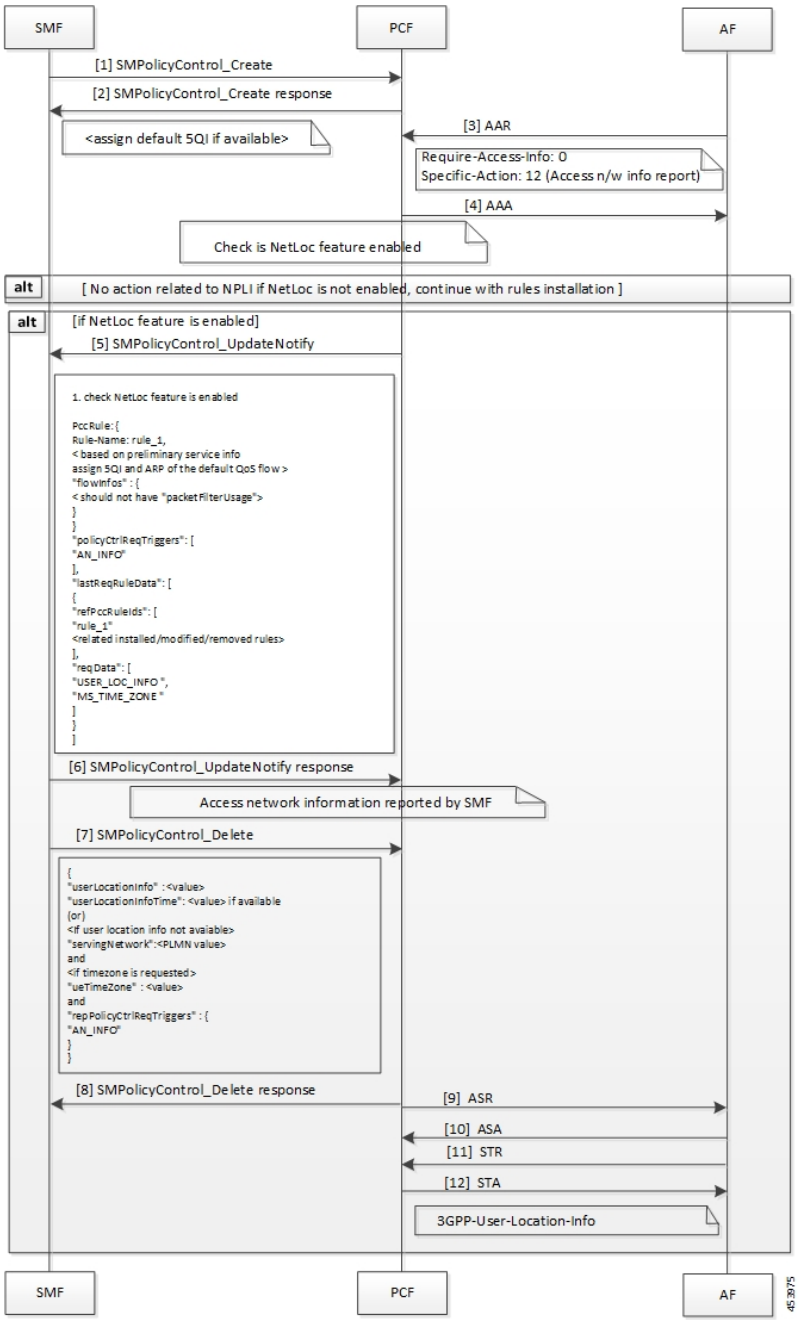


Table 47: NPLI in Rx STA Call Flow Description

Step	Description
1	The SMF sends a SMPolicyControl_Ceate request to the PCF.
2	The PCF responds with the SMPolicyControl_Create response to the SMF.

Step	Description
3	The AF sends an Authenticate-Authorize-Request message to the PCF. The message contains Required-Access-Info AVP requesting the access network information required for the AF session.
4	The PCF responds with an AA-Answer message to the AF.
5	If the NetLoc feature is enabled, the PCF sends the SMPolicyControl_UpdateNotify request to the SMF.
6	The SMF responds with the SMPolicyControl_UpdateNotify message to the PCF. This message contains the access network information.
7	The SMF sends the SMPolicyControl_Delete request to the PCF.
8	The PCF responds to the SMF with the SMPolicyControl_Delete message.
9	The PCF sends the Abort-Session-Request message to the AF.
10	The AF responds with the Abort-Session-Answer to the PCF.
11	The AF sends the Session-Termination-Request to the PCF.
12	The PCF responds with the Session-Termination-Answer message to the AF. This message contains the user location information.

Required Access Information in STR Call Flow

This section describes the Required Access Information in STR call flow.

Figure 20: Required Access Information in STR Call Flow

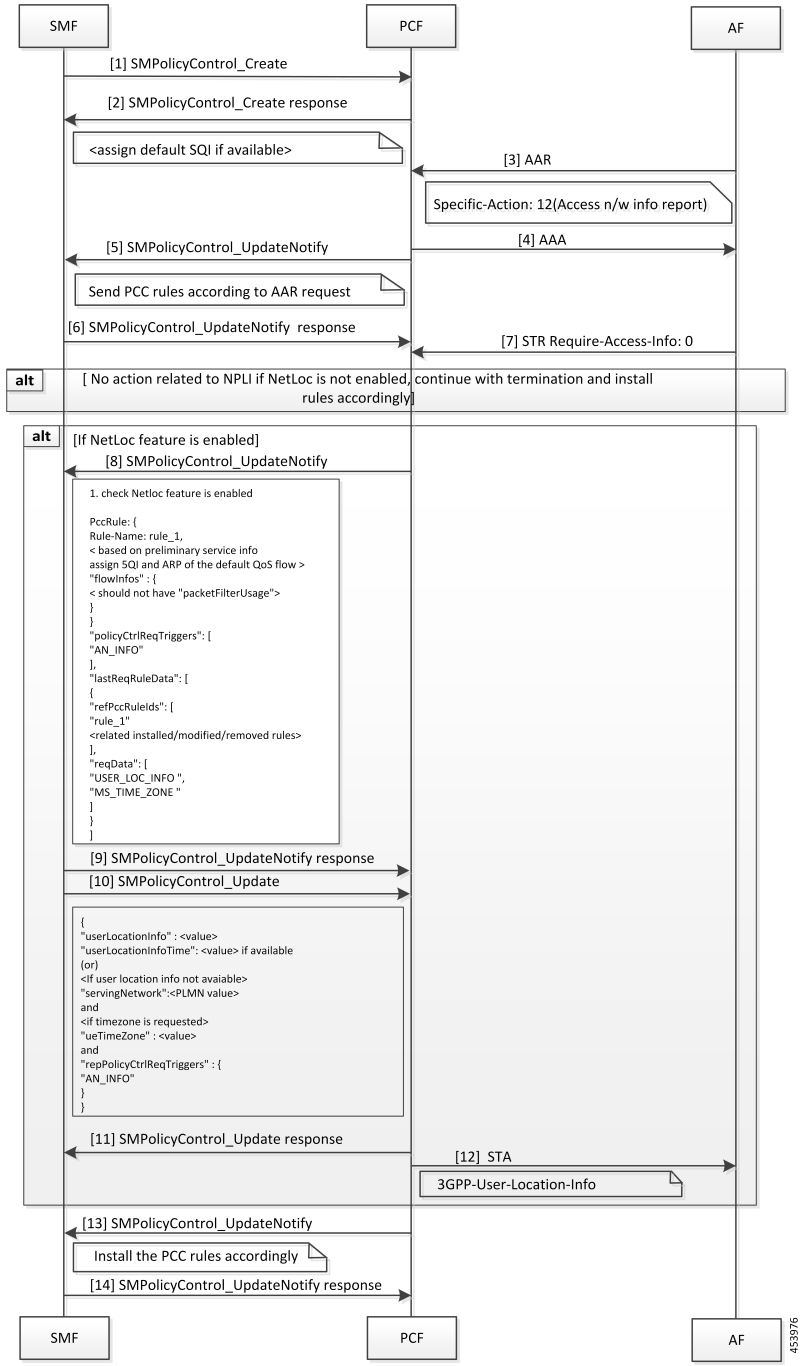


Table 48: Required Access Information in STR Call Flow Description

Step	Description
1	The SMF sends an SMPolicyControl_Create request to the PCF.

Step	Description
2	The PCF responds to the SMF with the SMPolicyControl_Create response.
3	The AF sends the Specific-Action: 12 (Access n/w info. report) message to the PCF.
4	The PCF sends an AA-Answer message to the AF.
5	The PCF sends an SMPolicyControl_UpdateNotify request to the SMF.
6	The SMF sends PCC rules as requested in the Authenticate-Authorize-Request in the SMPolicyControl_UpdateNotify response to the PCF.
7	The AF sends a Session-Termination-Request to PCF to retrieve the Required-Access-Info AVP.
8	If the NetLoc feature is enabled, the PCF sends as SMPOlicyControl_UpdateNotify request to the SMF.
9	The SMF sends an SMPOlicyControl_UpdateNotify response to the PCF.
10	The SMF sends an SMPOlicyControl_Update request to the PCF.
11	The PCF sends a response for the SMPOlicyControl_Update request to the SMF.
12	The PCF sends the Session-Termination-Answer message to the AF with the user location information.
13	The PCF sends the SMPolicyControl_UpdateNotify request to the SMF.
14	On installing the PCC rules, the SMF sends SMPolicyControl_UpdateNotify response to the PCF.

NPLI in N5 Notify Call Flow

This section describes the NPLI in N5 Notify call flow.

Figure 21: NPLI in N5 Notify Call Flow

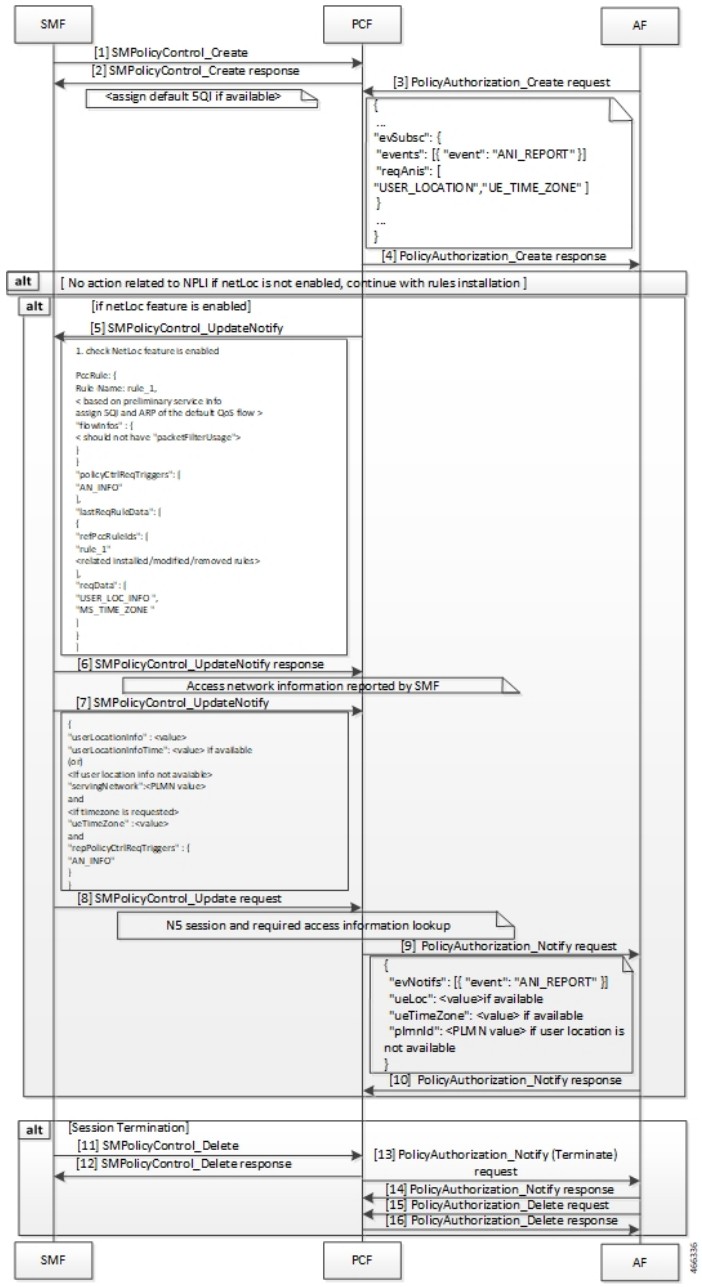


Table 49: NPLI in N5 Notify Call Flow Description

Step	Description
1	The SMF sends a SMPolicyControl_Ceate request to the PCF.
2	The PCF responds to the SMPolicyControl_Create request.

Step	Description
3	The AF sends an PolicyAuthorization_Create request to the PCF. The request contains the access network information required for the AF session.
4	The PCF sends the PolicyAuthorization_Create response to the AF.
5	If the NetLoc feature is enabled, then the PCF sends an SMPolicyControl_UpdateNotify request toward the SMF.
6	In response to the SMPolicyControl_UpdateNotify request, the SMF sends the access network information to the PCF.
7	The PCF sends the SMPolicyControl_Update request to the SMF.
8	The SMF sends the SMPolicyControl_Update request to the PCF.
9	After the establishing the N5 session and the required access information lookup, the PCF sends the PolicyAuthorization_Notify request to the AF containing access network information report with 3GPP user location, UE Timezone and serving network PLMN ID if available
10	The AF sends the PolicyAuthorization_Notify response to the PCF.
11	If the session terminates, the SMF sends a SMPolicyControl_Delete request to the PCF.
12	The PCF responds to SMF for the SMPolicyControl_Delete request.
13	The PCF sends the PolicyAuthorization_Notify (Terminate) request to the AF.
14	The AF responds with the PolicyAuthorization_Notify response to the PCF.
15	The AF sends the PolicyAuthorization_Delete request to the PCF.
16	The PCF responds with the PolicyAuthorization_Delete response to the AF.

NPLI in N5 Delete Response Call Flow

This section describes the NPLI in N5 Delete Response call flow.

Figure 22: NPLI in N5 Delete Response Call Flow

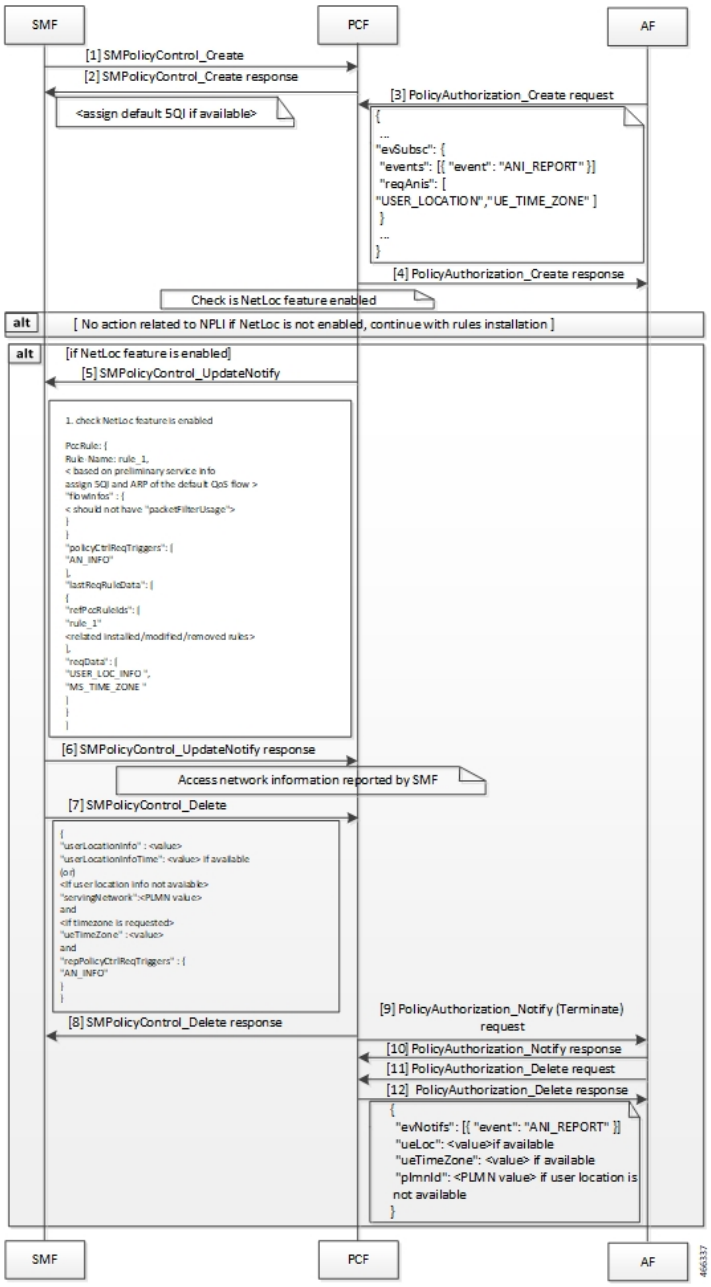


Table 50: NPLI in N5 Delete Response Call Flow Description

Step	Description
1	The SMF sends a SMPolicyControl_Ceate request to the PCF.
2	The PCF responds with the SMPolicyControl_Create response to the SMF.

Step	Description
3	The AF sends a PolicyAuthorization_Create request to the PCF. The request contains the access network information required for the AF session.
4	The PCF responds with an PolicyAuthorization_Create response to the AF.
5	If the NetLoc feature is enabled, the PCF sends the SMPolicyControl_UpdateNotify request to the SMF.
6	The SMF responds with the SMPolicyControl_UpdateNotify message to the PCF. This message contains the access network information.
7	The SMF sends the SMPolicyControl_Delete request to the PCF.
8	The PCF responds to the SMF with the SMPolicyControl_Delete message.
9	The PCF sends the PolicyAuthorization_Notify (Terminate) request to the AF.
10	The AF responds with the PolicyAuthorization_Notify response to the PCF.
11	The AF sends the PolicyAuthorization_Delete request to the PCF.
12	The PCF responds with the PolicyAuthorization_Delete response to the AF. This response contains the user location information.

Required Access Information in N5 Delete Request Call Flow

This section describes the Required Access Information in N5 Delete Request call flow.

Figure 23: Required Access Information in N5 Delete Request Call Flow

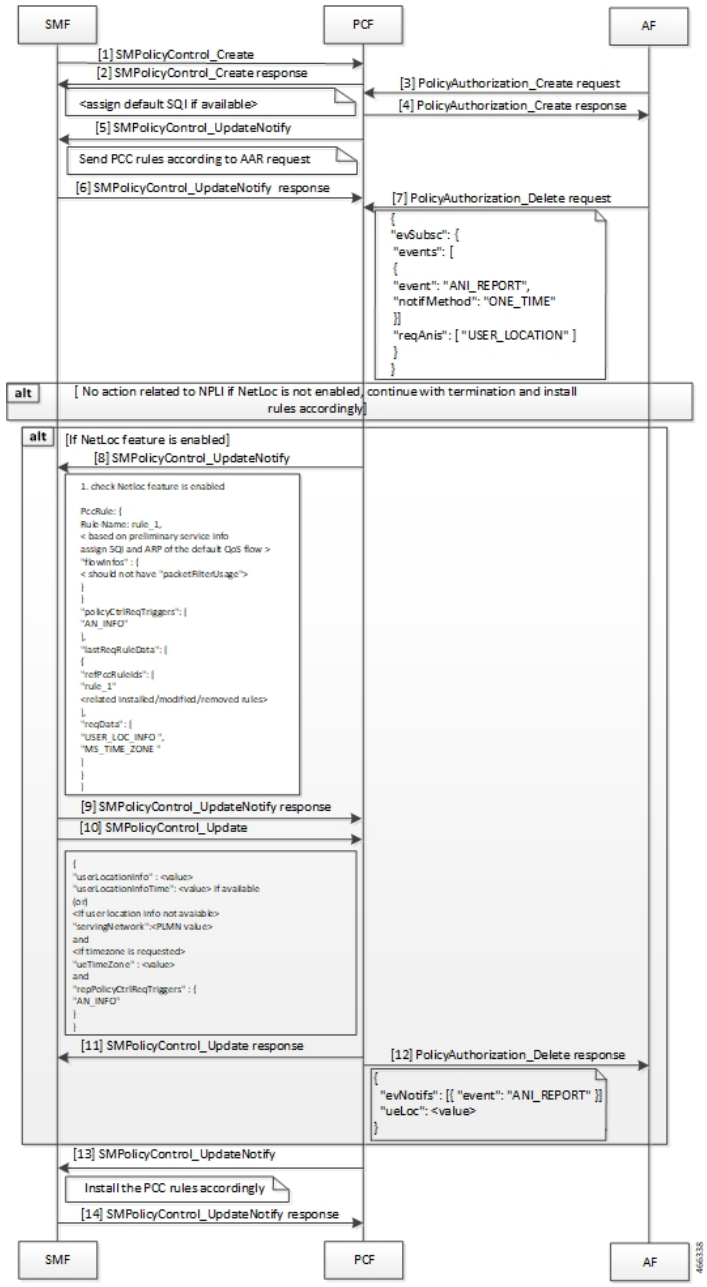


Table 51: Required Access Information in N5 Delete Request Call Flow Description

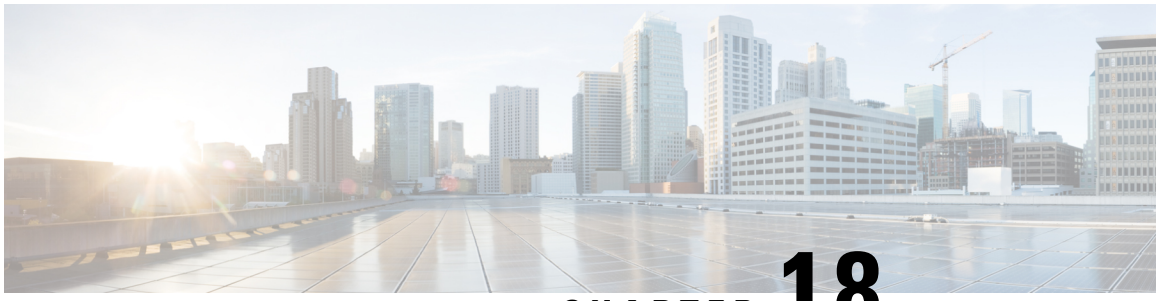
Step	Description
1	The SMF sends an SMPolicyControl_Create request to the PCF.
2	The PCF responds to the SMF with the SMPolicyControl_Create response.
3	The AF sends a PolicyAuthorization_Create request to the PCF.

Step	Description
4	The PCF sends a PolicyAuthorization_Create response to the AF.
5	The PCF sends an SMPolicyControl_UpdateNotify request to the SMF.
6	The SMF sends PCC rules as requested in the Authenticate-Authorize-Request in the SMPolicyControl_UpdateNotify response to the PCF.
7	The AF sends a PolicyAuthorization_Delete request to PCF to retrieve the required access network information.
8	If the NetLoc feature is enabled, the PCF sends as SMPOlicyControl_UpdateNotify request to the SMF.
9	The SMF sends an SMPOlicyControl_UpdateNotify response to the PCF.
10	The SMF sends an SMPOlicyControl_Update request to the PCF.
11	The PCF sends a response for the SMPOlicyControl_Update request to the SMF.
12	The PCF sends the PolicyAuthorization_Delete response to the AF with the user location information.
13	The PCF sends the SMPolicyControl_UpdateNotify request to the SMF.
14	On installing the PCC rules, the SMF sends SMPolicyControl_UpdateNotify response to the PCF.

Enabling the NetLoc Feature

This section describes how to enable the NetLoc feature that supports the Access Network Information Reporting in 5G.

To enable the NetLoc feature, in the initial N7 request set the "suppFeat" value's 6th binary digit to 1.



CHAPTER 18

Heartbeat

Feature Summary and Revision History

Summary Data

Table 52: Summary Data

Applicable Product(s) or Functional Area	PCF
Applicable Platform(s)	SMI
Feature Default Setting	Enabled – Configuration required to disable
Related Documentation	Not Applicable

Revision History

Table 53: Revision History

Revision Details	Release
Enhancement introduced. PCF is configured to send the cluster load information in the heartbeat.	2020.05.0
First introduced.	2020.02.0

Feature Description

PCF registers with NRF and sends a heartbeat message to the same NRF to infer its status as active or inactive. Complying with *3GPP TS 29.510*, PCF performs the following tasks when sending a heartbeat:

- Sends a heartbeat in the form of a PATCH request to, and processes responses with the NRF that it has registered with.
- Performs the failover operation when the registered NRF is unavailable due to connectivity issues or some unknown reasons. In such situations, PCF registers and uses the available secondary or tertiary NRF when the primary NRF is unresponsive. Simultaneously, PCF attempts to register with the primary NRF. When registration to the original (primary) NRF is successful, PCF stops sending heartbeats to the secondary or tertiary NRF.

  In the absence of the primary NRF, PCF performs the failover and failback in the following sequence:

  - Failover: Primary > Secondary or Tertiary > Tertiary
  - Failback: Tertiary > Secondary or Primary > Primary

- When PCF registers with a nonprimary NRF, it attempts to register with the primary NRF in the interval that is configured in the interval-in-secs parameter. For more information, see the nfServices information in the Network Repository Function Subscription to Notifications, on page 179 chapter.
- When sending two consecutive heartbeat messages, PCF honors the time interval that is available in the heartBeatTimer attribute in the registration response or the heartbeat response.
- Subscription management:
  - PCF subscribes to notifications from NRF for profile changes based on the ServiceName attribute. The subscription happens through a PATCH request.
  - After a subscription validity time has elapsed, PCF resubscribes to NRF through a PATCH request.
  - PCF sends a remove or delete request to NRF to cancel the subscription.

How it Works

This section describes how this feature works.

The PCF registers with the NRF to create a passage for interacting with the other NFs to perform operations such as discovery and selection.

The overview of how NF and PCF interact through NRF in the following sequence:

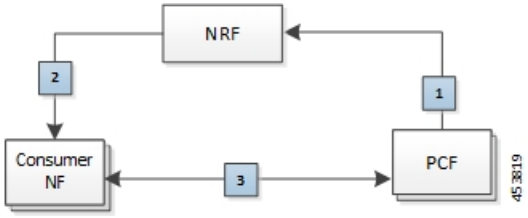
1. Registration: PCF registers its profile that defines the services or capabilities with the NRF. The registration service request contains the load parameters. The cluster load value is a collective value of the cluster memory and cluster CPU usage derived from Prometheus.
2. Discovery: After the registration is successful, the NRF sends the information about the registered PCF instances to the (consumer NF) NFs through an NRF query. The NFs that are registered with the NRF periodically send a heartbeat in the form of an NFUpdate service. The NF discovery response carries attributes such as load, capacity.

3. Selecton: When the NF wants to establish a connection with a PCF instance, it determines the appropriate instance based on the attributes such as load and location.

   If the NRF receives an NF query with the preferred-locality attribute, then, NRF assigns a higher priority value (higher the value, lower the priority) to the profiles or services that do not match the preferred-locality parameter. The NRF sorts the NF profiles and services based on the load, capacity, and priority in the next step. The consumer NF determines the registered NF based on the criteria that NRF has used for sorting. For example, if UPF wants to send a service request to a PCF instance in a preferred locality, then it selects the PCF profiles with the lowest value.

The following figure depicts how these components interconnect.

Figure 24: PCF-NRF-NFs Interaction Flow



Standards Compliance

This feature complies with the following standards specifications:

- *3GPP TS 29.510 (2018-12) "Network Function Repository Services;"*

Configuring the Cluster Load Attribute

This section describes how to enable PCF to send the cluster load information in the heartbeat request.

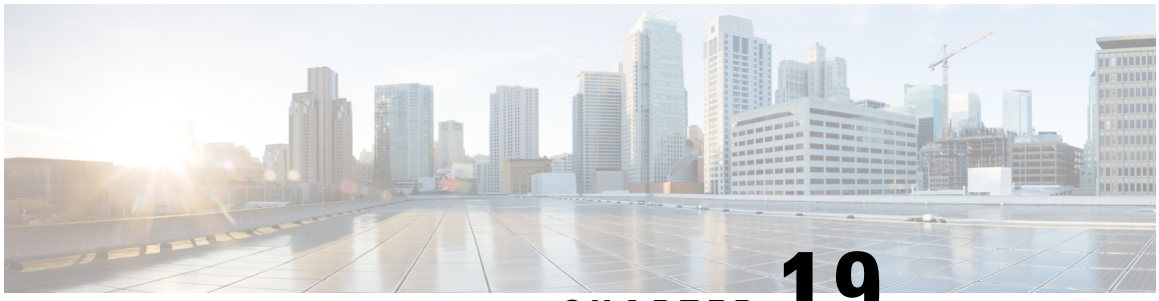
To configure the ability that allows PCF to send the cluster load parameter in the heartbeat request to the NRF, use the following configuration:

```
config
  group
    nf-mgmt [ name ]
     load-report-enabled [ false | true ]
     end
```

NOTES:

- **group**—Enters the group configuration mode.
- **nf-mgmt [name]**—Specify the management group that is associated to a network function.
- **load-report-enabled [false | true]**—Configures the ability to send the cluster load size in the heartbeat service request. The default value is set to true.

  The registration request may fail if the cluster load size (cluster memory usage and cluster CPU usage) is unavailable in the request. In such situations, you can disable the capability by setting this attribute to false.



CHAPTER 19

LDAP and Sh Interface

Feature Summary and Revision History

Summary Data

Table 54: Summary Data

Applicable Product(s) or FunctionalArea	PCF
Applicable Platform(s)	SMI
Feature Default Setting	Disabled – Configuration required to enable
Related Documentation	Not Applicable

Revision History

Revision Details	Release
Enhancement introduced. PCF supports IPv6 connectivity on LDAP endpoint.	2021.04.0
First introduced.	2020.01.0

Feature Description

PCF supports the LDAP and Sh versions of the N36 reference point to and from the simulated UDR to access subscriber profile information and to write dynamic session data as required for session processing.

This feature provides the following capabilities:

- Support for Sh Interface: PCF communicates with HSS and downloads the subscription profile. It sends policies that are based on the subscription profile.
- Support for policy changes based on subscription changes in PCF: Based on subscription changes that are received from Sh or LDAP or local configuration, PCF invokes the Npcf_SMPolicyControl_UpdateNotify service to update the policies in SMF.
- PCF supports both IPv4 and IPv6 connectivity on LDAP endpoint.

Call Flows

This section describes the key call flow for this feature.

Sh Interface Call Flow

This section describes the Sh Interface call flow.

Figure 25: Sh Interface Call Flow

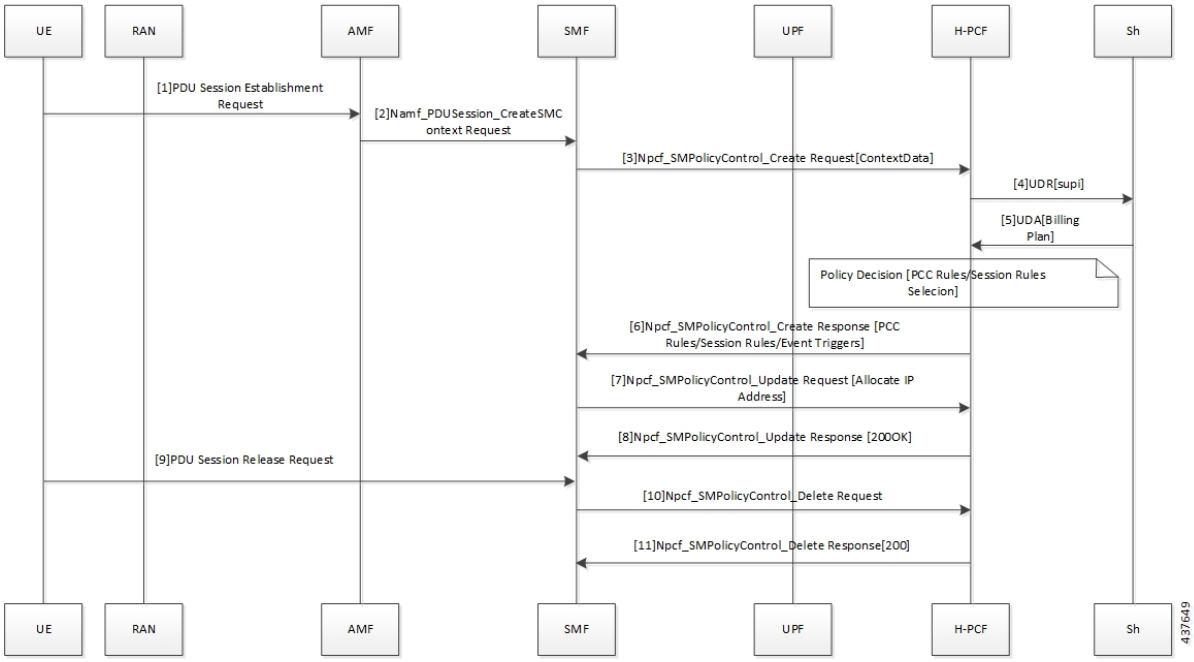


Table 55: Sh Interface Call Flow Description

Step	Description
1	The User Equipment (UE) sends a PDU Session Establishment request to the Access and Mobility Management (AMF) function.
2	The AMF creates the Namf_PDUSession_CreateSMConext_Request service and sends it to SMF.

Step	Description
3	The SMF creates and sends the Npcf_SMPolicyControl_CreateRequest[ContextData] service to H-PCF.
4	The PCF sends a request to the User Data Repository (UDR) through the Sh interface.
5	The Sh interface sends the UDA (Billing Plan) to the PCF.
6	The PCF responds with the Npcf_SMPolicyControl_Create service to the SMF.
7	The SMF sends the Npcf_SMPolicyControl_Update request to the PCF.
8	The PCF responds with the Npcf_SMPolocyControl_Update response to the SMF.
9	The UE sends the PDU Session Release request to SMF.
10	The SMF forwards the Npcf_SMPolicyContro_Delete request to the PCF.
11	The PCF sends the Npcf_SMPolicyControl_Delete response to SMF.

Configuring PCF to use LDAP

This section describes how to configure PCF to leverage the LDAP interface.

The configuration support for LDAP involves the following steps:

1. Setting Up Additional Profile Data
2. Associating PCF with LDAP

Setting Up Additional Profile Data

This section describes how to set up the profile data.

PCF establishes a connection with an LDAP server to access the subscriber profile data that resides on an external database. Upon receiving the PCF query, the LDAP searches its database to retrieve the user profile and other information.

You can set an LDAP interface profile for a new or an existing domain. By configuring the Domain, you direct PCF to retrieve data from an LDAP query.

1. Log in to Policy Builder and select the **Services** tab.
2. Navigate to the **Domains** tab and select DATA_5G.
3. In the **Domains** pane, click the **Additional Profile Data** tab.
4. Select **Generic Ldap Search** in the drop-down menu on the right-hand side of the **Additional Profile** section heading.
5. Under **Profile Mappings**, click **Add** to configure a new row for each attribute that is retrieved from the LDAP server. In the **Profile Mappings** table, the following parameters can be configured for the new row:

a. External Code: The LDAP attribute name to retrieve.

b. Mapping Type: The mapping of the data to an internal PCF data type.

c. Regex Expression and Regex Group: If parsing of the incoming AVP is required then define a regular expression and regular expression group to support retrieval of the parsed values.

d. Missing AVP: Defines the default AVP value when the subscriber attribute that is received from the external profile is missing.

Note

- If a subscriber attribute is missing and its missing AVP value is not configured, PCF does not create or update policy derived AVP for this subscriber with Missing AVP Value.
- This parameter is applicable only for Mapping Type as Subscriber Attribute or Service. For all other mapping types, this column is not applicable.

e. Empty AVP Value: Defines the default AVP value when a subscriber attribute that is received from an external profile has empty or blank value.

Note

- If a subscriber attribute is empty or blank and its empty or blank AVP value is not configured, PCF does not create or update policy derived AVP for this subscriber with Empty AVP Value.
- This parameter is applicable only for Mapping Type as Subscriber Attribute or Service. For all other mapping types, this column is not applicable.

f. Apply Timer: This check box indicates whether Timer Attribute is applicable to other subscriber attributes or not. Select the check box if Timer Attribute that must be applied for that subscriber attribute.

g. Discard If Empty: When checked, deletes the LDAP attribute from the session (thus preventing any further use) if regex (when configured) does not match the received value. By default, the check box is unchecked (false).

6. Enter the appropriate value in the following fields for completing the configuration:

   The following table describes the configuration service parameters.

Table 56: Configuration Parameters

Field	Description
Ldap Server Set	Associate the LDAP server set defined in the LDAP Server Set Definition.
Base Dn	Specify the Base DN that is sent in the LDAP query. If not defined, then the request does not contain a base DN.

Field	Description
Filter	Set to the filter value that is sent in the LDAP query. If not defined, then the request does not contain a filter. **Note** This string supports string replacement using the find and replace of strings with variables from the policy state as defined in the "Replacement Rules" table.
Dereference Policy	This is an optional field that controls whether to disable the LDAP query. This is often used along with Custom Reference Data tables and other session attributes to optionally disable an LDAP query. If the calculated CRD AVP has a value (ignoring case) of "false", then the LDAP query is skipped.
Avp Code to Disable Query	Set this to the dereference policy that the LDAP query requires. Default value is NEVER.
Profile Refresh Interval (mins)	Set this value to automatically refresh a profile by querying the profile after specified delay.
Replacement Rules	In the replacement rules table, add one row per replacement string to substitute into the Base DN or Filter string on a request by request basis.
Subscriber Timer Attribute	Indicates which attribute is a timer attribute among all the LDAP server attributes. The timer follows the ISO 8601 time standards. See ISO 8601 for more information.
Lower Bound For Timer Attribute In Minutes	Indicates how much time before the start time of Subscriber Timer Attribute PCF has to accept when LDAP server sends timer attribute. Default value is 30 mins.

Associating PCF with LDAP

This section describes how to associate PCF with LDAP.

When you configure PCF environment to interact with a defined LDAP, PCF must connect to the LDAP server using a trusted authentication method. This method is known as binding. PCF uses the binding information while making LDAP queries to retrieve the required subscriber information from the LDAP server.

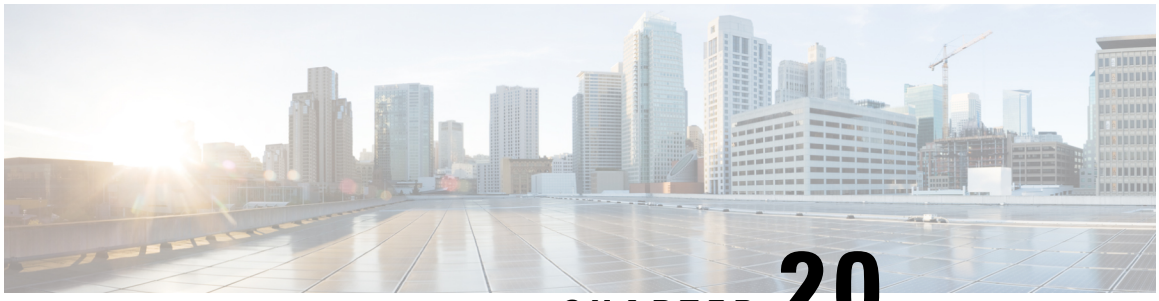
To associate PCF with LDAP, use the following configuration:

```
config
 product pcf
    ldap replicas replica_count
    ldap server-set server_set
       search-user dn cn=username,dc=C ntdb
       search-user password
       health-check interval-ms interval
       initial-connections connection_count
```

```
max-connections maximum_connections
retry-count retry_count
retry-timer-ms retry_time
max-failover-connection-age-ms maximum_failover
binds-per-second binds
number-consecutive-timeout-for-bad-connection consecutive_timeout
connection ip_address
  priority priority
  connection-rule connection_type
  auto-reconnect [ true | false ]
  timeout-ms timeout
  bind-timeout-ms bind_timeout
  end
```

NOTES:

- **product pcf**—Enters the PCF configuration mode.
- **ldap replicas** *replica_count*—Specify the LDAP replica count. Depending on the count, the LDAP pods are created.
- **ldap server-set** *server_set*—Specify the LDAP server set details.
- **search-user dn cn**=*username*, **dc**=*C ntdb*—Specify the domain details.
- **search-user** *password*—Specify the password.
- **health-check interval-ms** *interval*—Specify the interval at which the health check should be initiated.
- **initial-connections** *connection_count*—Specify the number of connections that can be attempted initially.
- **max-connections** *maximum_connections*—Specify the maximum number of connections at any point of time.
- **retry-count** *retry_count*—Specify the number of retries that the PCF Engine must attempt on a timeout.
- **retry-timer-ms** *retry_time*—Specify the interval after which the PCF Engine must reattempt.
- **max-failover-connection-age-ms** *maximum_failover*—Specify the maximum number of connection failures after which failover must happen
- **binds-per-second** *binds*—Specify the interval in seconds for the bind operation.
- **number-consecutive-timeout-for-bad-connection** *consecutive_timeout*—Specify the number of bad connections after which the timeout occurs.
- **connection** *ip_address*—Specify the IPv4/IPv6 address of the LDAP server that attempts the connection.
- **priority** *priority*—Specify the priority of the connection.
- **connection-rule** *connection_type*—Specify the connection type. The default rules are "Fastest" or "Round Robin".
- **auto-reconnect [true | false]**—Specify if the auto-connect capability should be enabled or disabled.
- **timeout-ms** *timeout*—Specify the period between the LDAP client or endpoint when the timeout must happen.
- **bind-timeout-ms** *bind_timeout*—Specify the bind timeout.



CHAPTER 20

Managing Custom Reference Data

Feature Summary and Revision History

Summary Data

Table 57: Summary Data

Applicable Products or Functional Area	PCF
Applicable Platform(s)	SMI
Feature Default Setting	Enabled – Always-on
Related Documentation	Not Applicable

Revision History

Table 58: Revision History

Revision Details	Release
First introduced.	2020.05.0

Feature Description

The Custom Reference Data (CRD) is the reference data specific to a service provider, such as their networks or cell sites' names and characteristics. This data is required to operate the policy engine but not used for evaluating the policies. The CRD is represented in the table format. The service providers have the flexibility to create custom data tables and manage them as per their requirements.

Note Make sure to start all the policy servers after a CRD table schema is modified (for example, column added/removed).

CRD supports the pagination component, which controls the data displayed according to the number of rows configured for each page. You can change the number of rows to be displayed per page. Once you set the value for rows per page, the same value is used across the Central unless you change it. Also, you can navigate to other pages using the arrows.

Configuration Support for Importing CRD

This section describes the procedure to import CRD when the CRD schema is modified.

Importing of CRD involves the following steps:

- Backing Up the Existing SVN Repository
- Backing Up the Existing CRD
- Removing the Existing CRD from MongoDB
- Importing and Publishing the New CRD Schema
- Importing the New CRD Table

Backing Up the Existing SVN Repository

This section describes how to import the SVN repository when the CRD schema is modified.

To take a backup of the existing SVN repository and store it on another environment, use the following configuration:

1. Log in to the PCF Central GUI.
2. On the **Cisco Policy Suite Central** page, navigate to **Policy Builder** and click the **Import/Export** link.

   The **Import/Export** form opens.
3. In the **Export** tab, select the **All data** option to configure the export type.

   The following table describes the export/import options:

Table 59: Export and Import Options

Parameters	Description
All data	Exports service configuration with environment data, which acts as a complete backup of both service configurations and environmental data.

Parameters	Description
Exclude Environment	Exports without environment data, which allows exporting configuration from a lab and into another environment without destroying the new system's environment-specific data.
Only Environment	Exports only environment data, which provides a way to back up the system-specific environmental information.
Export URL	The URL can be accessed from the Policy Builder or viewed directly in Subversion.
Export File Prefix	Provide a name (prefix) for the export file. **Note** The exported filename automatically includes the date and time when the export was performed.

4. If you want to export the file in the compressed format, select the **Use 'zip' file extension** check box.
5. Click **Export**.
6. Navigate to the file and save it to your local machine. The file must include the cluster name and date.

Backing Up the Existing CRD

This section describes how to import an existing CRD when the CRD schema is modified.

To take a backup of the configured CRD and store it to another environment, use the following configuration:

1. Log in to the PCF Central GUI.
2. On the **Cisco Policy Suite Central** page, navigate to **Custom Reference Data** and click the **Custom Reference Data** link.

   The **Import/Export CRD data** form opens.
3. Under **Export Custom Reference Data**, the following options are displayed:

   Table 60: Export Custom Reference Data Options

Options	Description
Use 'zip' file extension	Enables easier viewing of the exported contents for the advanced users.
Export CRD to Golden Repository	When the system is in a BAD state, the CRD cache is built using the golden-crd data.

4. Click **Export**.

Removing the Existing CRD from MongoDB

This section describes how to remove the existing CRD tables that have schema change from MongoDB.

To remove a configured CRD schema change, use the following configuration:

1. Log in to the admin-db pod that has the CRD (cust_ref_data) database.
2. Access the cust_ref_data using the following command:

   ```
   use cust_ref_data
   ```
3. Delete the data from one or more existing CRD tables using the following command:

   ```
   db.table_name.remove({})
   ```
4. Exit the admin-db pod.

Importing and Publishing the New CRD Schema

This section describes how to import and publish the new CRD schema.

To import and publish the CRD schema, use the following configuration:

1. Log in to the PCF Central GUI.
2. On the **Cisco Policy Suite Central** page, navigate to **Policy Builder** and click the **Import/Export** link.

   The **Import/Export** form opens.
3. In the **Import** tab, browse to the file that you want to import.
4. In the **Import URL** field, enter the URL where the file must be imported. We recommend importing a new URL and verify it using the Policy Builder.
5. In the **Commit Message** field, enter the appropriate information.
6. To enforce import in situations where the checksums don't match, select the **Force import even if checksums don't match** check box.
7. Click **Import**.

Importing the New CRD

To import the new CRD, use the following configuration:

1. Access the Policy Builder URL and add a new repository.
   a. In the **Choose Policy Builder data reposiorty...** window, select **<Add New Repository>** from the drop-down.

      The **Repository** dialog box appears.

      The following parameters can be configured under **Repository**:

      Configure the parameters according to the network requirements.

Table 61: Repository Parameters

Parameter	Description
Name	This is a mandatory field. Ensure that you specify a unique value to identify your repository's site. **Note** We recommend the following format for naming the repositories: customername_project_date, where underscores are used to separate customer name, project, and date. Date can be entered in the MMDDYYYY format.
Username and Password	Enter a username that is configured to view the Policy Builder data. The password can be saved for faster access, but it is less secure. A password, used with the Username, permits, or denies access to make changes to the repository.
Save Password	Select this check box to save the password on the local hard drive. This password is encrypted and saved as a cookie on the server.
Url	You can have several branches in the version control software to save different versions of configuration data. Create a branch in the version control software before assigning the URL in this screen. Enter the URL of the branch of the version control software server that is used to check in this version of the data.

Parameter	Description
Local Directory	Do not modify the value in this field. This is the location on the hard drive where the Policy Builder configuration objects are stored in the version control. When you click either Publish or Save to Repository, the data is saved from this directory to the version control application specified in the Url text field. The field supports the following characters: • Uppercase: A to Z • Lowercase: a to z • Digits: 1–9 • Nonalphanumeric: / **Note** The user must use only the supported characters.
Validate on Close	Select this check box to see if the values for Username, Password, or the URL are legitimate and unique. If not, the screen displays an error message and provides a chance to correct the errors.
Remove	Removes the display of the repository in Cisco Policy Builder. **Note** The remove link here does not delete any data at that URL. The local directory is deleted.

b. Click **OK** to save your work to the local directory.

Note When you change screens, the Policy Builder automatically saves your work. We recommend saving your work to the local directory by clicking on the diskette icon on the Policy Builder GUI or CTRL-S on the keyboard.

c. If you are ready to commit these changes to the version control software, choose **File > Save to Client Repository** on the Policy Builder home screen.

2. Log in to the new repository.
3. Verify the new CRD table schema and publish the changes.

4. Review the crd-api pod logs for any exception or error related to the duplicate key or duplicate index. If there are no errors, then the CRD is successfully imported.

Importing the New CRD Table

This section describes how to import the CRD table.

To import new CRD tables, use the following configuration:

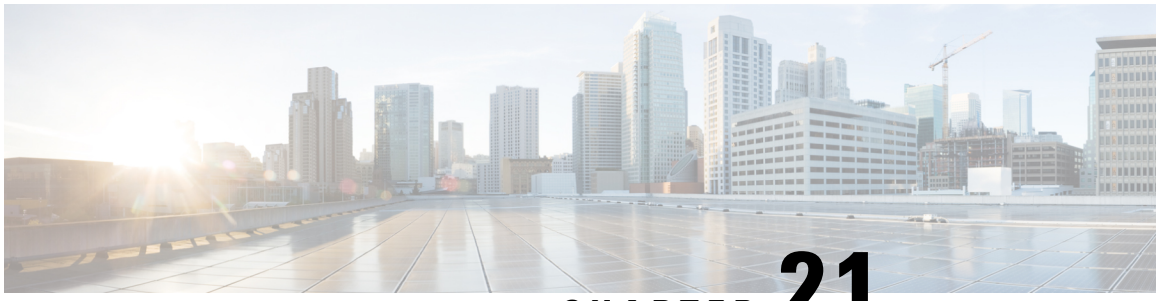
Before importing the CRD table, ensure that the CRD data archive is saved as dot (.) crd or dot (.) zip.

1. Log in to the PCF Central.
2. Click **Custom Reference Data**.
3. Click **Import/Export CRD Data**.
4. Under **Export Custom Reference Data**, the following options are displayed:
   - Select the **Use 'zip' file extension** check box to enable easier viewing of export contents for advanced users.
   - Select the **Export CRD to Golden Repository** check box to export CRD to golden repository which is used to restore cust_ref_data in case of error scenarios. A new input text box is displayed.
5. Add a valid SVN server hostname or IP address to push CRD to repository. You can add multiple hostnames or IP addresses by clicking on the plus sign.
6. Click **Export**.

Verifying the Successful Export of CRD Table to Golden Repository

To verify of the export of the custom CRD table to the golden repository is successful, use the following configuration:

1. Log in to the PCF Central.
2. Click **Custom Reference Data**.
3. Click **Import/Export CRD Data**.
4. In **Import Custom Reference Data**, click **Field to Import field and browse for the CRD archive**.
5. Click the **Import** button to import the CRD data.
6. On successful import, verify that you receive a "Data imported" message on the PCF Central GUI.
7. Review crd-api pod logs for any exception or error related to duplicate key or duplicate index. If there are no errors, then the CRD is successfully imported.



CHAPTER 21

Message Prioritization and Overload Handling

Feature Summary and Revision History

Summary Data

Table 62: Summary Data

Applicable Product(s) or Functional Area	PCF
Applicable Platform(s)	SMI
Feature Default Setting	Disabled – Configuration required to enable
Related Documentation	Not Applicable

Revision History

Table 63: Revision History

Revision Details	Release
First introduced.	2022.01.0

Feature Description

PCF supports the following message prioritizations:

- Message priority handling framework—Provides configuration to handle the incoming message rules with priority.

  - Diameter configuration
  - PCF configuration
- Inbound WPS Rx Message prioritization—During engine processing the WPS messages are prioritized over non-WPS messages.
- Inbound WPS SBI Message prioritization—During engine processing the 3gpp-Sbi-Message-Priority header messages are prioritized.

How it Works

This section describes how this feature works.

- Message Prioritization Handling Framework
  - Diameter Configuration—Use Message Handling Rules parameters (Diameter Client, Protocol, Command Code, Request Type, Priority, Per Instance TPS, and Discard Behavior) to identify and prioritize the diameter messages.
  - PCF Configuration—Use Message Handling Rules parameters (Request Type, Priority, Per Instance TPS, and Discard Behavior) to identify and prioritize the SBI messages.
- Inbound WPS Rx Message Prioritization—Use the Rx Message Prioritization parameters to mark the WPS specific MPS-Identifier and Reservation Priority. Rx message should be prioritized based on the MPS-Identifier and Reservation-Priority AVPs received in the request message.
- Inbound WPS SBI Message Prioritization—Based on the SBI Message Priority value in the incoming message, a user specified priority (Inbound SBI-Message-Priority prioritization table in PCF configuration) is assigned. SBI messages with higher priority are considered for processing earlier than the messages with lower priority.

Note Currently, the priority queue and rate limiting for REST and Diameter messages are independent of each other.

Feature Configuration

To configure this feature, use the following configurations:

- Configuring Inbound Message Overload Handling
- Configuring SBI-Message-Priority Prioritization

Configuring Inbound Message Overload Handling

This section describes how to configure the Inbound Message Overload Handling for the diameter and PCF configurations.

Diameter Configuration

1. Log in into Policy Builder.
2. Select **Reference Data** tab.
3. From the left pane, select **Systems**.
4. Select and expand your system name.
5. Select **Diameter Configuration**.
6. In the right pane, to add the parameters of the inbound message overload handling, check the **Inbound Message Overload Handling** check box.
7. In the Inbound Message Overload Handling area, define the following parameter details.

Table 64: Inbound Message Overload Handling Parameters

Parameter	Description
Default Priority	Default priority to be assigned to an incoming message if no specific priority is defined in the Message Handling Rules table. Default value is 0.
Message Sla Ms	Service Level Agreement (SLA) in milliseconds, defines the number of milliseconds that are associated with an incoming event or message. In case the configured duration times out, the Discard Behavior configured in the Message Handling Rules is applied else the Default Discard Behavior is used. Maximum time (in millisec) that a message has in an inbound message handling queue waiting for a worker thread. Configuring this value avoids processing a message to time out by a remote peer. Default value is 1500 ms.
Inbound Message Queue Size	Allows the maximum number of messages in the Inbound Message Queue. When the number of messages exceeds this value, messages are discarded as defined in the Message Handling Rules and the Default Discard Behavior. Default value is 1000.
Default Instance Rate Limit	This parameter is applied to messages that do not have an applicable overload handling rule configured in the Message Handling Rules table. Default value is 0.

Parameter	Description
Default Discard Behaviour	Default behavior to be applied to an incoming message if no specific discard behavior for that message is defined in the Message Handling Rules table. • MESSAGE_DROP: Discards the request. • DIAMETER_TOO_BUSY: Sends a response message havingResult-Code AVP value set to DIAMETER_TOO_BUSY (3004). Default value is MESSAGE_DROP
Rx Message Prioritization	Defines Rx eMPS message handling priority based on the Rx message MPS-Identifier and Reservation-Priority AVPs. For more information see Table 65: Rx Message Prioritization Parameters, on page 152.
Message Handling Rules	Defines specific inbound message overload handling rules based on different criteria. For more information, see Table 66: Message Handling Rules Parameters, on page 153.

Figure 26: Inbound Message Overload Handling Parameters

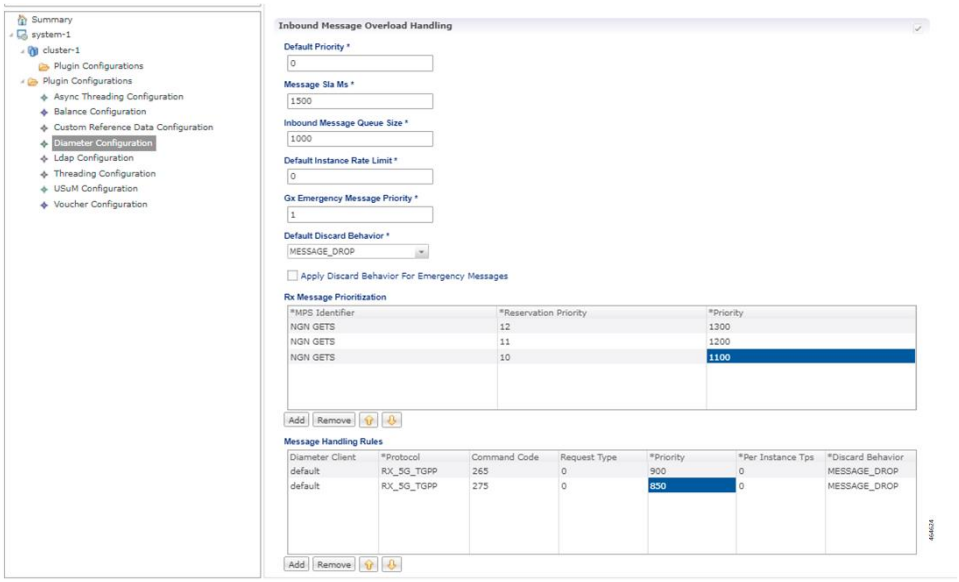


Table 65: Rx Message Prioritization Parameters

Parameter	Description
MPS Identifier	MPS-Identifier indicates that an AF session relates to an MPS session. It contains the national variant forMPS service name. For example, NGN GETS.

Parameter	Description
Reservation Priority	The AF specifies the Reservation-Priority AVP at request level in the AA-Request in order to assign a priority to the AF session as well as specify the Reservation-Priority AVP at the media-component-description AVP level to assign a priority to the IP flow. The Reservation-Priority AVP available at the request level only is used under Rx Message Prioritization table. If Reservation priority is not found at the message level in Rx message, then best value of Reservation Priority is calculated from the MCD and used for lookup. Range: 1 to 15, where 15 is considered as the highest priority and 1 is considered as the least priority.
Priority	A user defined priority based on MPS-Identifier and Reservation-Priority combination. Higher Priority messages are processed before lower priority messages.

Table 66: Message Handling Rules Parameters

Parameter	Description
Diameter Client	This is used to configure different priorities for different clients based on realms.
Protocol	Specific application id value to be used for scoring. This value is used to match Auth-Application-Id AVP value.
Command Code	Specific command code value to be used for scoring. This value is used to match the Command-Code field. These command codes map to different types of Diameter messages.

Parameter	Description
Request Type	Specific request type value to be used for scoring. This value should match the value of the CC-Request-Type AVP for Gx CCR messages. • 0: Request Type not used for scoring • 1: INITIAL_REQUEST (1) • 2: UPDATE_REQUEST (2) • 3: TERMINATION_REQUEST (3) Default value is 0. Request type should match the value of the Rx-Request-Type AVP for Rx messages. • 0: INITIAL_REQUEST (0) • 1: UPDATE_REQUEST (1) Request type should match the value of SL-Request-Type AVP for Sy SLR messages. The possible values are: • INITIAL_REQUEST (0) • INTERMEDIATE_REQUEST (1) It has to be configured to zero if the incoming message does not have a request type AVP. For example, Rx STR does not have a request type AVP or Rx-Request-Type AVP is unavailable in Rx message as it is not a mandatory AVP per 3GPP TS 29.214.
Priority	Priority value assigned to the message. Higher numerical value has the higher priority. Default value is 0. For example, 10, 20, 100, 200, 300, 500 and so on.
Per Instance Tps	Transactions per second limit per process. This value is the TPS that these messages are limited to. The actual system's transaction per second limit can be calculated using the following formula: Per Instance Tps x Number of instances per VM x Number of VMs. Default value is 0. For example, 1000, 2000, 5000 and so on.

Parameter	Description
Discard Behavior	Behavior to be applied to an incoming message. • MESSAGE_DROP: Discards the request. • DIAMETER_TOO_BUSY: Sends a response message having Result-Code AVP value configured to DIAMETER_TOO_BUSY (3004). Default value is MESSAGE_DROP.

PCF Configuration

1. Log in into Policy Builder.
2. Select **Reference Data** tab.
3. From the left pane, select **Systems**.
4. Select and expand your system name.
5. Select **PCF Configuration**.
6. In the right pane, to add the parameters of the inbound message overload handling, check the **Inbound Message Overload Handling** check box.
7. In the Inbound Message Overload Handling area, define the following parameter details.

Table 67: Inbound Message Overload Handling Parameters

Parameter	Description
Default Priority	Default priority to be assigned to an incoming message if no specific priority is defined in the Message Handling Rules table. Default value is 0.
Message Sla Ms	Service Level Agreement (SLA) in milliseconds, defines the number of milliseconds that are associated with an incoming event or message. In case the configured duration times out, the Default Discard Behavior is applied. Maximum time (in millisec) that a message has in an inbound message handling queue waiting for a worker thread. Configuring this value avoids processing a message to time out by a remote peer. Default value is 1500 ms.

Parameter	Description
Inbound Message Queue Size	Allows the maximum number of messages in the Inbound Message Queue. When the number of messages exceeds this value, messages are discarded as defined in the Message Handling Rules and the Default Discard Behavior. Default value is 1000.
Default Instance Rate Limit	This parameter is applied to messages that do not have an applicable overload handling rule configured in the Message Handling Rules table. Default value is 0.
N7 Emergency Message Priority	Default priority assigned to messages related to an emergency session.. Emergency message priority is applied when the DNN matches an emergency DNN configured under PCF Configuration. Default value is 1.
Default Discard Behaviour	Default behavior to be applied to an incoming message if no specific priority is defined in the Message Handling Rules table. • MESSAGE_DROP: Discards the request. • SERVICE_UNAVAILABLE: Service is not available. Default value is MESSAGE_DROP.
Message Handling Rules	Defines specific inbound message overload handling rules based on different criteria. For more information, see Table 68: Message Handling Rules Parameters, on page 157

Figure 27: Inbound Message Overload Handling Parameters

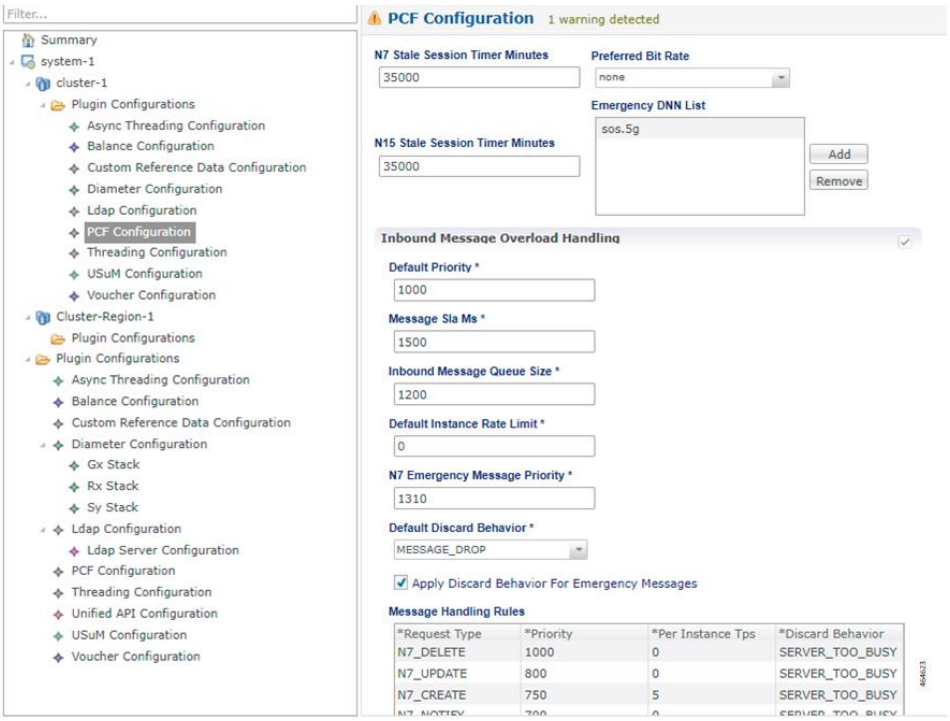


Table 68: Message Handling Rules Parameters

Parameter	Description
Request Type	Specifies request type value to be used for scoring. For example N7_CREATE, N28_NOTIFY, and so on.
Priority	Priority value assigned to the message. Higher numerical value has the higher priority. For example, 700, 800 and so on.
Per Instance Tps	Transactions per second limit per process. This value is the TPS that these messages are limited to. Default value is 0.
Discard Behavior	Behavior to be applied to an incoming message. • SERVER_TOO_BUSY: Sends a responsive message having result code attribute value with HTTP code 503. • MESSAGE_DROP: Discards the request.

Configuring SBI-Message-Priority Prioritization

This section describes how to configure the SBI-Message-Priority Prioritization.

1. Log in into Policy Builder.
2. Select **Reference Data** tab.
3. From the left pane, select **Systems**.
4. Select and expand your system name.
5. Select **PCF Configuration**.
6. In the right pane, to add the parameters of the SBI-Message-Priority prioritization, check the **SBI-Message-Priority Prioritization** check box.
7. In the SBI-Message-Priority Prioritization area, define the following parameter details.

Table 69: SBI-Message-Priority Prioritization Parameters

Parameter	Description
Default Inbound Priority	The default value is used if priority value does not match a value in Inbound SBI-Message-Priority Prioritization table
Inbound SBI-Message-Priority Prioritization	A user defined priority based on SBI Message Priority and Priority combination. For more information, see Table 70: Inbound SBI-Message-Priority Prioritization Parameters, on page 159

Figure 28: SBI-Message-Priority Prioritization

Table 70: Inbound SBI-Message-Priority Prioritization Parameters

Parameter	Description
SBI Message Priority	The range of values allowed for SBI-Message-Priority are from 0–31, where 0 is considered as the highest priority and 31 is considered as the least priority value.
Priority	It provides the queue priority value. A higher numerical priority value equates to a higher priority.

OAM Support

This section describes operations, administration, and maintenance information for this feature.

Bulk Statistics Support

The following statistics are supported for the message prioritization and overload handling feature.

Note The following values apply to all the statistics:

- Unit - Int64
- Type - Counter
- Nodes - Service

The following metrics track the counter information:

- input_queue_result - Captures the status of the message in the inbound queue whether it is dropped or rate limited.

  The following labels are defined for this metric:

  - appid
  - message-type
  - result

- wps_rx_priority - Captures the Rx message queue priority.

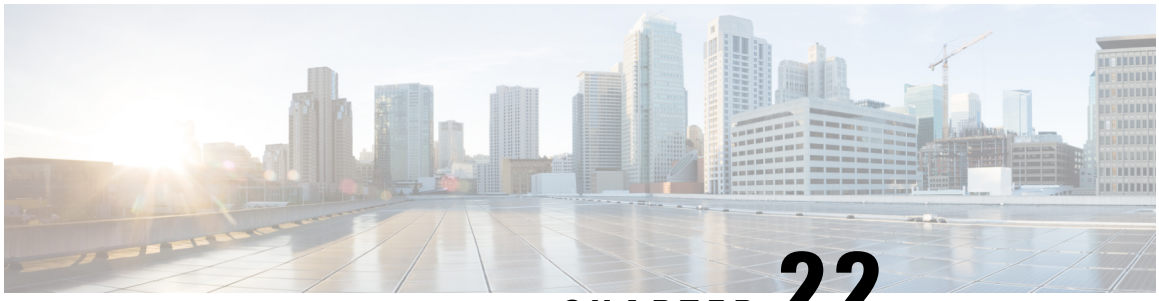
  The following labels are defined for this metric:

  - command_code
  - priority

- sbi_priority_total - Captures the SBI message queue priority.

  The following labels are defined for this metric:

- interface
- message_type
- priority



CHAPTER 22

Multiple Virtual IP Address

Feature Summary and Revision History

Summary Data

Table 71: Summary Data

Applicable Product(s) or Functional Area	PCF
Applicable Platform(s)	SMI
Feature Default Setting	Disabled – Configuration required to enable
Related Documentation	Not Applicable

Revision History

Table 72: Revision History

Revision Details	Release
Enhancement introduced. PCF supports N5 Interface.	2022.02.0
Enhancement introduced. PCF supports dual stack (IPv4 and IPv6) connectivity on N7, N28 and NNRF external interfaces/endpoints.	2022.01.0

Revision Details	Release
Enhancement introduced. PCF supports IPv6 connectivity on N7, N28 and NNRF external interfaces/endpoints.	2021.04.0
Enhancement introduced. Support added for HTTP IDLE Connection Timeout on Server	2021.02.0
First introduced.	2020.01.0

Feature Description

You can now enable the IPv4 communication between PCF and the other network functions such as AF, SMF, NRF, CHF, and UDR through multiple virtual IP addresses (VIP). With a provision to configure discrete VIP addresses or external IP addresses for each rest-ep service and link them to an endpoint, you can prevent sharing of IP addresses between the NFs. Multiple VIPs take the role of a load balancer to offer a high availability environment.

In a scenario where multiple calls are simultaneously made to a distinct network function, the policy service spawns different REST ep services to complete each interaction. PCF spawns a service using the IP address available in the IP pool.

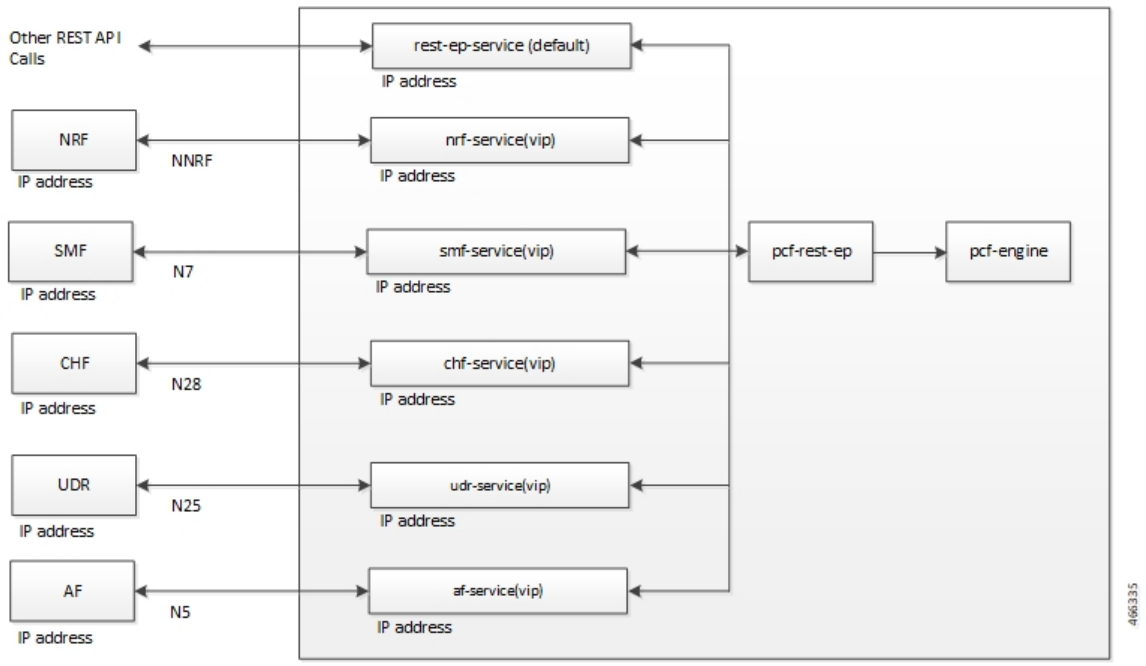
PCF supports both IPv4 and IPv6 connectivity on N5, N7, N28 and NNRF external interfaces/endpoints (inbound and outbound).

Architecture

This section describes how the network function components interact when the multiple VIP model is implemented.

The multiple VIP architecture focuses on high availability and load-balancing aspect of IP addresses in 5G. With relevance to the multiple VIP graphic, the Policy Engine invokes a new rest-ep service for a NF when you assign an IP address as an external endpoint. All the incoming requests from the network functions, such as NRF and SMF are routed to the rest-ep-service and the traffic is redirected to the pcf-rest-ep pod. The pod has a bilateral communication with the PCF Engine. The rest-ep-service operates as a load balancer.

Figure 29: Multiple VIPs



How it Works

This section describes how this feature works.

After the admin associates an IP address to a network function such as PCF, a new endpoint is linked to the network function through the rest-ep service. This service enables you to connect to the pcf-rest-ep pod.

You can configure multiple IP address for the N5, N7, N36, N28, and Nnrf interfaces. During this process, an individual K8 service resource of type Load Balancer is created for each interface that communicates with the rest-endpoint pod. These IP addresses get listed in the ExternalIP property of the K8 service.

PCF supports multiple IP service communications with one replica of the rest-endpoint pod.

Configuration Support for Multiple Virtual IP Address

The configuration of the multiple virtual IP address involves the following:

- Configuring the REST Endpoints
- Verifying the REST Endpoints Configuration

Configuring the REST Endpoints

This section describes how to configure the IP address, port numbers, and other attributes for a REST endpoint.

Note Configuration changes to the REST endpoint cause the endpoint to restart automatically. Cisco recommends making such changes only within the maintenance window.

Before configuring the external IP addresses for the PCF REST endpoints, make sure that you configure and deploy the IP addresses using the SMI Deployer.

For a single interface both IPv4 or IPv6 address can be used.

To configure REST endpoint, use the following configuration in the Policy Ops Center console:

```
config
   rest-endpoint
      interface [ n5 | n7 | n15 | n25 | n28 | nnrf ]
          ip
          ipv6 interface_ipv6_address
          port interface_port_number
      ips ip_address
      port port_number
      http-connection-limit maximum_inbound_connection_count
      http-idle-connection-timeout-on-server-seconds idle_connection_timeout
      replicas replica_count
      inbound-request-timeout-ms inbound_timeout
      outbound-request-timeout-ms outbound_timeout
      repository repository_address
      tracing-service-name tracing_service
      uri-scheme uri_scheme
      end
```

NOTES:

- For each REST endpoint, use a separate **rest-endpoint** *ip_address* command.
- **interface [n5 | n7 | n15 | n25 | n28 | nnrf]**—Specify the interface name and IP address that is configured for the external IP. *interface_name ip_address* must include the interface name such as N7, N36, N28, and NNRF.
  - **ip** *interface_ip_address*—Specify the IPv4 address that is assigned for the interface.
  - **ipv6** *interface_ipv6_address*—Specify the IPv6 address that is assigned for the interface.
  - **port** *interface_port_number*—Specify the port number for the interface.

    For example, to enable the N15 interface allocate resources such as IP and port number. Once the interface is configured, the PCF-AMF traffic can pass through N15.
- **ips** *ip_address*—Specify the IPv4 or IPv6 address that is assigned as a REST endpoint external IP address.
- **port** *port_number*—Specify the port number for the REST endpoint.
- **http-connection-limit** *maximum_inbound_connection_count*—Specify the maximum number of inbound HTTP connections that the REST endpoint server must accept. Default value is 200.
- **http-idle-connection-timeout-on-server-seconds** *idle_connection_timeout*—Specify the idle connection timeout for REST connection where PCF is acting as server. Default value is 60 seconds.

If the value is less than or equal to 0, the default value of 60 seconds is used.

- **replicas** *replica_count*—Specify the number of instances of the service-based interface.
- **inbound-request-timeout-ms** *inbound_timeout*—Specify the timeout period after which the inbound request expires. You can configure a single *inbound_timeout* value for all the configured interfaces or the single interface.
- **outbound-request-timeout-ms** *outbound_timeout*—Specify the timeout period after which the outbound request expires. You can configure a single *outbound_timeout* value for all the configured interfaces or the single interface.
- **repository** *repository_address*—Specify a repository that the network interface optimizes.
- **tracing-service-name** *tracing_service*—Specify the service that is used for tracing purpose.
- **uri-scheme** *uri_scheme*—Specify the URI scheme as HTTP or HTTPs.

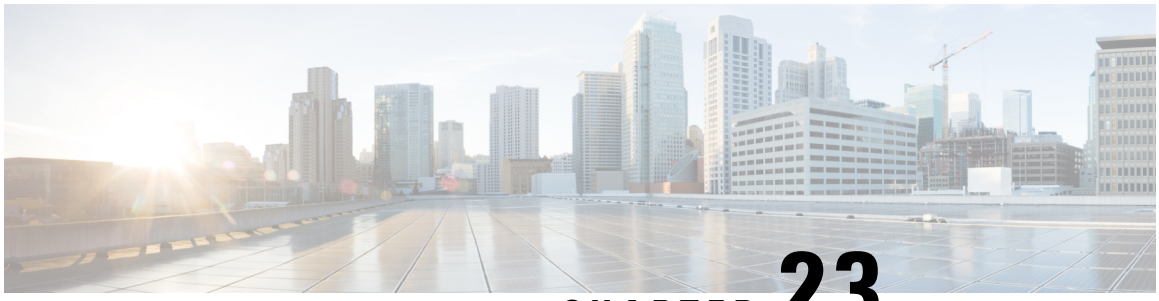
Note If the configured IP address is not accessible, then PCF fails to connect with the other NFs and reports an error message in the service as "Failed to allocate IP for "pcf/udr-rest-ep": no available IPs".

Verifying the REST Endpoints Configuration

This section describes how to verify the REST Endpoints configuration.

After an interface IP address is configured, you can observe a new service with the name as <*interface-name*>-rest-ep. The service type as ClusterIP gets created within the configured IP address. For example, n36-rest-ep.

If an IP address is not associated to an interface, then PCF considers an external IP address and associates it with the interface.



CHAPTER 23

N5 Authorization

Feature Summary and Revision History

Summary Data

Table 73: Summary Data

Applicable Product(s) or Functional Area	PCF
Applicable Platform(s)	SMI
Feature Default Setting	Disabled – Configuration required to enable
Related Documentation	Not Applicable

Revision History

Table 74: Revision History

Revision Details	Release
First introduced.	2022.02.0

Feature Description

PCF provides a method for the service providers to regulate the services available to individual subscribers. You can configure the bearer-level regulation through the customization and configuration of N5 Authorization.

The configuration handles the Video over NR (ViNR) authorization as per the subscriber attributes (SUPI, GPSI, and Throttling) to control the services available to each subscriber.

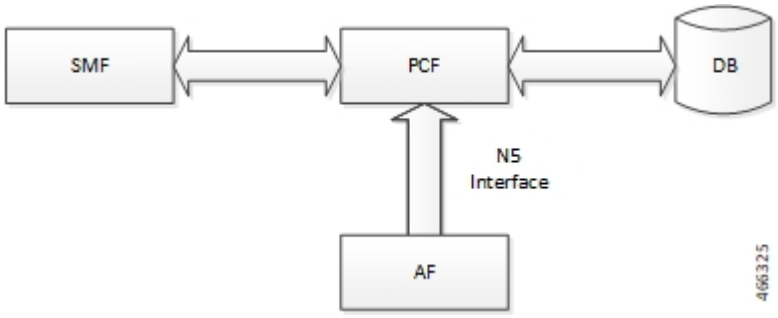
Architecture

This section depicts how the network function components interact during an N5 Authorization.

The SMF and PCF have a bilateral communication over the N7 interface. The AF sends an N5 Create/Update request to PCF. The PCF performs the N5 Authorization of the request by evaluating the message for the missing media type attribute and consults the value that is assigned to the Bearer-Authorization column in the STG table for the configured status as accept or reject. PCF fetches the STG information from the associated database. PCF communicates the evaluation result to the SMF and AF through REST requests.

The following figure illustrates how the NF interactions happen over the N5 interface.

Figure 30: NF Interactions



Components

This section describes the N5AuthorizationSTGConfiguration component in the N5 Authorization process.

The N5AuthorizationSTGConfiguration service configuration is used to evaluate the N5 Authorization table and obtain the configured output values. The N5AuthorizationSTGConfiguration service supports chained evaluation of Search Table Groups (STGs) which means multiple STGs are configured hierarchically in the service and outputs of one table is used as input keys for another table. The N5AuthorizationSTGConfiguration configuration evaluates all the bearers on receiving a Rest message and sends the appropriate Rest requests or responses depending on the bearer's authorization status provided the N5 session exists. The N5 Authorization table from which Bearer Authorization and Error Cause output values are received is configured as the last table in the list of chained STGs configured under N5AuthorizationSTGConfiguration.

How it Works

This section describes how this feature works.

At a high-level, PCF supports the N5-based authorization of bearers. The N5 authorization requires a Search Table Groups (STG), which enables logical grouping of multiple Customer Reference Data (CRD) tables. Within this STG, a CRD table that is dedicated to N5 Authorization is created in the Policy Builder. The input keys in the CRD signify the conditions based on which PCF determines the throttle limit for a bearer. The table has the following output columns:

- Bearer Authorization: Indicates whether to allow or reject a bearer.

- Error Cause: Specifies the Error-Message that is included in the N5 response, if necessary.

If PCF is configured to reject the N5 dedicated bearer when the associated Media-Type is missing, it rejects the bearer with the HTTP status code = 403 Forbidden, problem cause=REQUESTED_SERVICE_NOT_AUTHORIZED and, problem detail="Invalid service information, Media type is not specified" in response.

PCF is configured to reject a non-GBR bearer if the value for both, upload and download of the non-GBR bearer is set to 0. PCF determines if the bearer is non-GBR with 0-bit rate after consulting the NON-GBR QCI and ZERO BIT RATE QoS input columns in the N5 Authorization table. If Bearer-Authorization value is set to REJECT, then PCF rejects the bearer with HTTP status code=403 Forbidden, problem cause=REQUESTED_SERVICE_NOT_AUTHORIZED and, problem detail="BLOCKED" in response.

If PCF receives a N5 Create/Update request with multiple media components, and it rejects one of the media component after assessing for N5 Authorization, PCF sends a successful response for the accepted media components. For the rejected media components, PCF creates a scheduled event for sending a delayed N5 Notify request. You can configure the duration between the rejection and the time when scheduling the delayed message happens. The default value is set to 500 milliseconds.

Note In case, PCF rejects multiple media components with cause=REQUESTED_SERVICE_NOT_AUTHORIZED, the error resulting from the last rejected media component is set as problem detail in the response.

For existing bearers in an N5 session, PCF evaluates them for N5 Authorization when an event occurs such as LDAP refresh, N28 NOTIFY, and N7_NOTIFY. In situations where all the media components that are stored in the N5 sessions are rejected, then PCF sends a N7 Notify Terminate request to Application Function (AF).

Note You may observe a degradation in the performance of the PCF system when the N5AuthorizationSTGConfiguration service is added. The level of degradation corresponds to the number of STGs configured for the chained evaluation in the N5AuthorizationSTGConfiguration service and the number of bearers the service has evaluated.

Call Flows

This section describes the key call flows for this feature.

All Bearers Are Rejected Call Flow

This section describes the All Bearers Are Rejected call flow.

Figure 31: All Bearers Are Rejected Call Flow

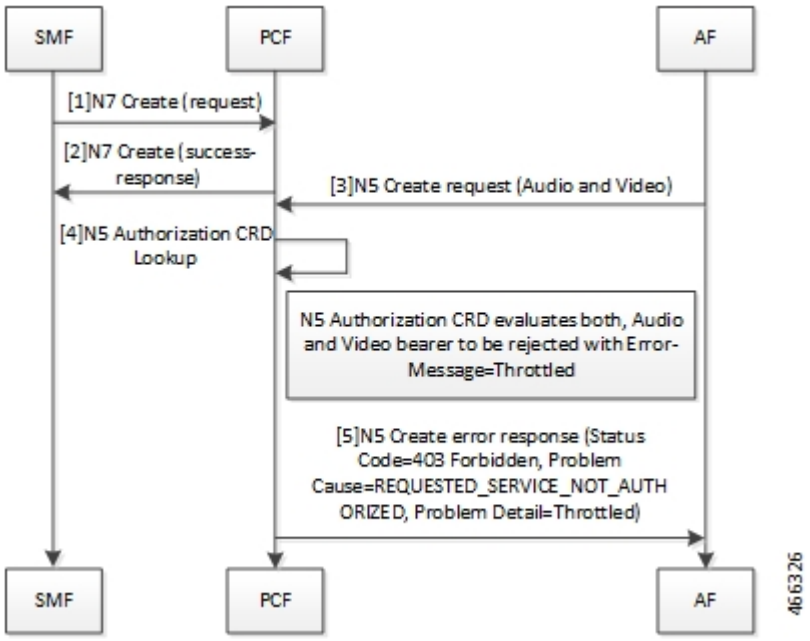


Table 75: All Bearers Are Rejected Call Flow Description

Step	Description
1	The SMF sends a N7 Create request to the PCF.
2	The PCF responds to the SMF with the success response.
3	The AF sends an N5 Create request (Audio and Video) message to the PCF.
4	The PCF performs the N5 Authorization CRD lookup.
5	The N5 Authorization CRD evaluates both, audio and video bearer. If there is a missing MediaType IE, PCF rejects the bearer. PCF validates all the bearers for Bearer-Authorization=REJECT. The bearers are classified as unauthorized and are not installed on the SMF. If all bearers received in the AAR are rejected, PCF sends a N5 Create error response with Status Code=403 Forbidden, Problem Cause=REQUESTED_SERVICE_NOT_AUTHORIZED, Problem Detail=Throttled to the AF

Few Bearers Are Rejected Call Flow

This section describes the Few Bearers are Rejected call flow.

Figure 32: Few Bearers Are Rejected Call Flow

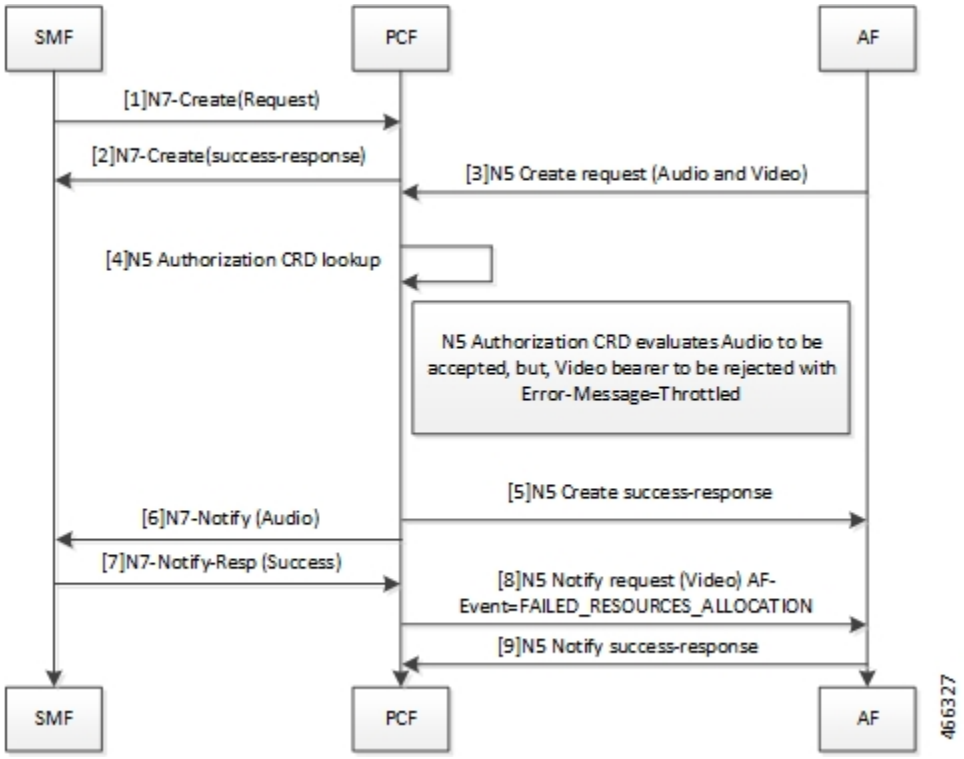


Table 76: Few Bearers Are Rejected Call Flow Description

Step	Description
1	The SMF sends a N7 Create request to the PCF.
2	The PCF responds to SMF with success response.
3	The AF sends an N5 Create request (Audio and Video) message to the PCF.
4	The PCF performs the N5 Authorization CRD lookup.
5	The N5 Authorization CRD evaluates both the audio and video bearers. The audio bearers that contain the required MediaType IE are tagged as accepted. Video bearers with the missing MediaType IE are rejected. Bearers evaluated to Bearer-Authorization=ACCEPT are authorized and installed on the SMF. PCF responds to the accepted audio bearers with N5 Create success response.
6	The PCF sends N7 Notify (Audio) to the SMF.
7	The SMF responds to the PCF with a N7 Notify-Resp (Success).
8	Bearers evaluated to Bearer-Authorization=REJECT are marked as unauthorized and are not installed at the SMF. The PCF sends N5 Notify request (Video) AF-Event=FAILED_RESOURCES_ALLOCATION to AF.

Step	Description
9	The AF responds with N5 Notify success-response to the PCF.

Existing Bearers Are Rejected Call Flow

This section describes the Existing Bearers Are Rejected call flow.

Figure 33: Existing Bearers Are Rejected Call Flow

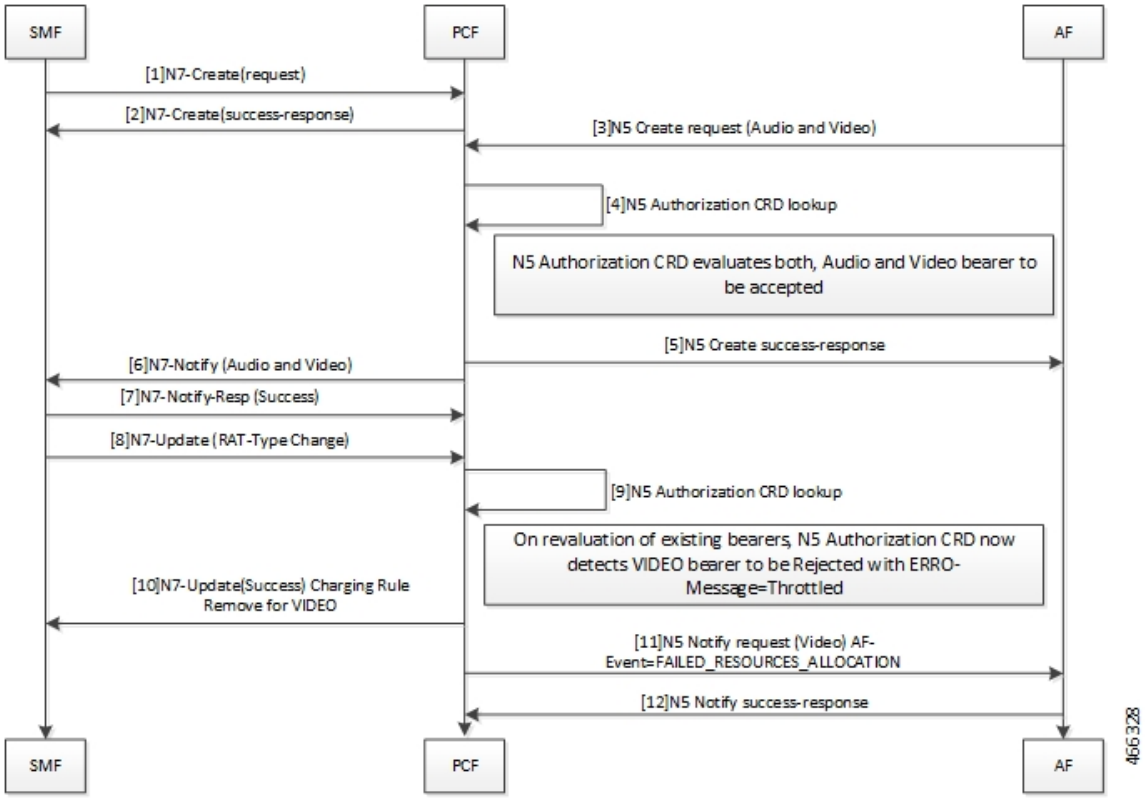


Table 77: All Bearers Are Rejected Call Flow Description

Step	Description
1	The SMF sends a N7 Create request to the PCF.
2	The PCF responds to the SMF with a N7 Create Success response.
3	The AF sends N5 Create request (Audio and Video) message to the PCF.
4	The PCF performs the N5 Authorization CRD lookup.
5	The N5 Authorization CRD evaluates both, the audio and video bearers. If successful authorization, PCF sends N5 Create success response to AF.
6	The PCF sends N7 Notify (Audio and Video) message.

Step	Description
7	The SMF responds with N7-Notify-Resp (Success) to the PCF.
8	The SMF sends N7 Update (RAT-Type Change).
9	The PCF performs the N5 Authorization CRD lookup.
10	When PCF reevaluates the existing bearer and the N5 Authorization CRD detects a VIDEO bearer with the Bearer-Authorization=REJECT, PCF rejects the bearer with Error-Message=Throttled. The PCF sends N7-UPDATE (Success) Charging Rule Remove for VIDEO to the SMF.
11	The PCF sends N5 Notify request (Video) AF-Event=FAILED_RESOURCES_ALLOCATION to the AF
12	The AF responds with N5 Notify success-response to the PCF.

Considerations

The following considerations apply when you configure the N5 Authorization:

- The STG names that are configured in the N5AuthorizationSTGConfiguration should be unique.
- The IE names for the output columns that are configured in the N5AuthorizationSTGConfiguration service should be unique.
- The chained evaluation keys should have the same IE name for the output column in the source table, and the input column in the destination table.
- The result of the N5AuthorizationSTGConfiguration service is available in the last table that is defined in the list. The table includes the output columns with the following mandatory IE names: Bearer-Authorization and Error-Message.
- The Bearer-Authorization column can be configured to accept the fixed values that are Accept and Reject.
- Perform the configurations that are required for defining and mapping the CRD tables as per the requirement.
- The Policy Server evaluates the mapped source output IEs (result column of the STG) through the CRD which it has created. If PCF has not created the CRD, then it cannot query the corresponding chained input key which further limits it from verifying the N5 Authorization.
- 1:1 mapping must exist between a chained pair of output IE and the input key.

Limitations

This feature has the following limitations in this release:

- When N5 Authorization fails, PCF sends an N5 Notify request only if the AF has subscribed to AF-Event=FAILED_RESOURCES_ALLOCATION in N5 Create request.
- The N5 Authorization is performed only against MediaComponent IE in the request. This indicates that the attributes from N5 Create/Update messages that are used as input for the CRD table evaluation should be from MediaComponent IE only. PCF does not evaluate the MediaSubComponent IE.

- If using the PolicyState or Session data retrievers that are bound to the input keys, then PCF retrieves the data for the input keys if it is inserted into the session data.

Feature Configuration

This section describes how to configure N5 Authorization.

The configuration of the N5 Authorization capability in PCF involves the following steps:

1. Creating the STG Tables
2. Adding the N5AuthorizationSTGConfiguration Service
3. Configuring the Service Chaining
4. Rejecting N5 Create with Missing MediaType IE
5. Setting Up the Delayed Message Schedule

Creating the STG Tables

This section describes how to create the STG column in Policy Builder.

To configure the STG column, use the following configuration:

1. Log in to Policy Builder.
2. Click the **Reference Data** tab, and from the left pane click **Custom Reference Data Tables** to view the options.
3. On the left pane, click the **Search Table Groups** folder.
4. In the **Search Table Group Summary** pane, click **Search Table Group**. A default STG gets created under the **Search Table Groups** folder.
5. Click the new STG and in the **Search Table Groups** pane rename the STG with a unique name.
6. Click **Customer Reference Data Table**. A new table gets created on the left pane.
7. Click the new table to open the **Customer Reference Data Table** pane. Rename the table with a unique name.
8. Navigate to the Columns section and click **Add**. A default column gets added to the Columns section.
9. Click the newly created column heading and rename it. Select the options in the corresponding row as applicable to your environment.

Note If the **Key** option is selected for a specific column, then it indicates as the input column.

10. Save the changes.

Adding the N5AuthorizationSTGConfiguration Service

This section describes how to add the N5AuthorizationSTGConfiguration service.

To configure the N5AuthorizationSTGConfiguration service, use the following configuration:

1. Log in to Policy Builder.
2. Choose the **Services** tab, and from the left pane click **Use Case Templates** to create a new service.
3. On the left pane, click **Summary** to open the **Summary** pane.
4. Under **Actions**, click **Use Case Template**.
5. In the **Use Case Template** pane, specify the name for the template.
6. Click the **Actions** tab and select **Add**.
7. In the **Select Service Configuration** dialog box, select the N5AuthorizationSTGConfiguration and click **OK**. The Use Case template with the specified name is created.
8. In the left pane, click **Services > Service Options** to view the options. The newly created service appears in the **Service Options**.
9. Select the service that you have created.
10. Under **Service Configurations**, click **Add** to open the **Select Service Configuration** dialog box.
11. Under **Service Configurations**, select **N5AuthorizationSTGConfiguration**, then click **OK**.

Configuring the Service Chaining

This section describes how to configure the service chaining for N5 Authorization.

Before configuring the service chaining, ensure that you have created the use case templates and added the N5AuthorizationSTGConfiguration service. Use case templates are the building blocks of the PCF architecture. The use case templates allow you to define the Service Configuration objects to be set by a Service Option.

To configure service chaining, use the following configuration:

1. Log in to Policy Builder.
2. Click the **Services** tab, and from the left pane click **Service Options** to view the options.
3. Expand the new service that you have created, and select the child.
4. In the **Service Option** pane, select **N5_AuthorizationSTGConfiguration** service under **Service Configurations** and specify the N5_AuthorizationSTGConfiguration parameters.
5. Expand the **List Of Input Column Avp Pairs (List) > ColumnAndAvpPair**, and enter the appropriate information.
6. Expand the **List Of Output Column Avp Pairs (List) > ColumnAndAvpPair**, and enter the Avp Name as Bearer-Authorization. Similarly, in another **ColumnAndAvpPair > Avp Name** field specify Error-Message.
7. Save the changes.

Rejecting the N5 Create Request with Missing MediaType IE

This section describes how to enable PCF to reject the N5 Create Request with Missing MediaType IE.

To configure PCF to reject the N5 Create Request, use the following configuration:

1. Log in to Policy Builder.
2. Click the **Reference Data** tab.
3. In the left pane, click **SBA Profiles** > **N5 Profiles**.
4. Click **N5 Profile**.
5. In the **N5 Profile** pane, select the **Reject AAR with missing Media Type** check box.
6. Save the changes.

Setting Up the Delayed Message Schedule

This section describes how to set up the duration after which PCF sends the delayed message to the AF.

To configure the delayed message schedule through the Policy Builder, use the following configuration:

1. Log in to Policy Builder.
2. Click the **Reference Data** tab.
3. In the left pane, click **SBA Profiles** > **N5 Profile**.
4. Click **N5 Profile**.
5. In the **N5 Profile** pane, specify the duration in the **Sending Delayed Message Wait Time (In millisec)** field. If you do not specify the period, then PCF considers the default period of 500 milliseconds.

N5 Profile

This section describes the parameters, which you can configure for the N5 Profile.

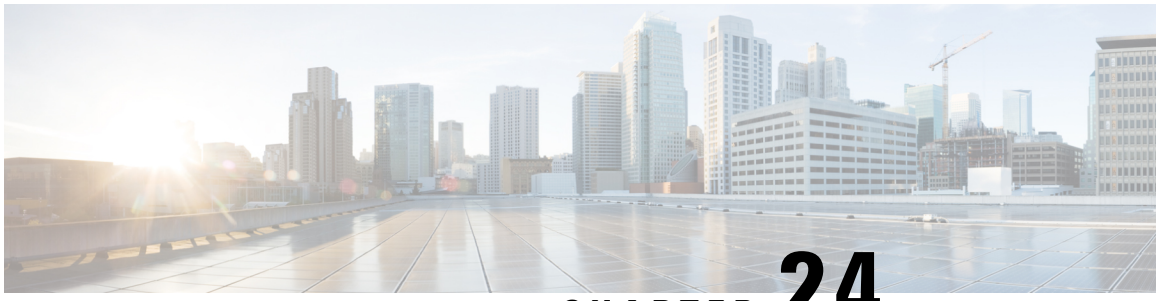
Before setting the service parameters, ensure that you create a use case template and add a service for this configuration. For details, see Configuring the Use Case Template, on page 293 and Adding a Service, on page 292.

The following table describes the N5 Profile service parameters:

Table 78: N5 Client Parameters

Parameter	Description
Reject AAR with missing Media Type	Enables PCF to reject the N5 Create/Update requests when MediaComponent have MediaType IE unspecified. PCF rejects the request with HTTP Status Code=403 Forbidden, Problem Cause=REQUESTED_SERVICE_NOT_AUTHORIZED To enable the parameter, select the check box available in the **SBA Profiles** > **N5 Profiles**.

Parameter	Description
Delayed Message Wait Time	Allows you to specify the duration after which PCF sends a delayed message. The default value is 500 milliseconds. To define the duration, specify the period in the text field available in **SBA Profiles > N5 Profiles**.



CHAPTER 24

Network Repository Function Subscription to Notifications

Feature Summary and Revision History

Summary Data

Table 79: Summary Data

Applicable Product(s) or Functional Area	PCF
Applicable Platform(s)	SMI
Feature Default Setting	Enabled – Configuration required to disable
Related Documentation	Not Applicable

Revision History

Table 80: Revision History

Revision Details	Release
First introduced.	2020.01.0

Feature Description

PCF supports the following functions for the Network Repository Function (NRF) Subscription to Notifications feature:

- The NRF supports the 3GPP December 2018 specification for interface discovery, registration for renaming NRF, change type, and removal or addition of new API attributes. PCF supports the notification subscription from NRF. The notifications are for profile changes that are based on the service name.
- PCF supports the subscription of notifications from NRF. This support includes the following functions:
  - Use the NRFManagement service for subscriptions for changes in network function instances that are based on the subscribed service name.
  - Implement notifications callback URL for PCF to handle the notifications from NRF for subscribed service names.
  - Allow the resubscription during the validity subscription time.
  - Support unsubscription based on the subscribed ID.
  - Prioritize NF profiles from NRF over preconfigurations or configured local set for an NF type.
- Supports the following repository functions:
  - Allow the repository configuration with multiple endpoints, which are primary endpoints, secondary endpoints, and tertiary endpoints.
  - Allow configuration of the profile discovery, which is based on service name and other parameters.
  - Allow configuration of the registration repository.
- Handles notifications from NRF for the subscribed service name.
- Provides statistics and metrics to track the following tasks:
  - Manage notifications of NFProfile from NRF for a specific service name.
  - Manage subscriptions for a specific service name.
  - Manage resubscriptions for a specific service name.
  - Manage unsubscriptions and deletions for a specific service name.

Standards Compliance

This feature complies with the following standards specifications:

- *3GPP TS 29.510 V15.2.0 (2018-12) "Network Function Repository Services"*

Configuration Support for the NRF Subscription to Notifications

The configuration of NRF subscription to notifications involves performing the followings steps:

1. Configuring NRF with Multiple Base URLs
2. Configuring NRF for Registration
3. Configuring NRF for Discovery of Network Function

Configuring NRF with Multiple Base URLs

This section describes how to configure NRF that has multiple base URLs.

To configure NRF with multiple base URLs, use the following configuration:

```
config
   nf-services nf_services_name
      repository repository
         name repository_name
         services services_name
         api-version-in-uri api_version_uri_name
         base-urls
          primary primary_endpoints_url
          secondary secondary_endpoints_url
          tertiary tertiary_endpoints_url
          end
```

NOTES:

- **nf-services** *nf_services_name*—Specify network functions, such as registration, repository, and service discovery details.
- **repository** *repository*—Specify a repository for the network function services.
- **name** *repository_name*—Specify the repository with the name you specify.
- **services** *services_name*—Specify a service for the repository name that you configured. Select one of these options — **nchf-spendinglimitcontrol**, **nnrf-disc**, **nnrf-nfm**, and **nudr-dr**.
- **api-version-in-uri** *api_version_uri_name*—Specify a version for the API version in URI for discovery and subscription of service to NRF.
- **base-urls**—Specify the primary, secondary, or tertiary endpoint as the base URL.
  - **primary** *primary_endpoints_url*—Specify the base URL for the primary endpoint.
  - **secondary** *secondary_endpoints_url*—Specify the base URL for the secondary endpoint when the primary endpoint is unavailable.
  - **tertiary** *tertiary_endpoints_url*—Specify the base URL for the tertiary endpoint when both the primary and the secondary endpoints are unavailable.

Configuring NRF for Registration

This section describes how to enable NRF for registering the NFs.

To configure NRF for registration, use the following configuration:

```
config
  nf-services nf_services_name
    registration
    service-repository  service_repository_name
    heartbeat
     failure-threshold failure_threshold_in_secs
```

```
interval-in-secs interval_in_secs
end
```

NOTES:

- **nf-services** *nf_services_name*—Specify the network function service configuration mode. From this mode, you can configure the services such as registration, repository, and service discovery details.
- **registration** – Enters the registration configuration mode.
- **service-repository** *service_repository_name*—Specify the name of the repository from the repository configuration.
- **heartbeat** – Enters the heartbeat configuration mode.
- **failure-threshold** *failure_threshold_in_secs*—Specify the value for the number of failures before confirming the heartbeat failure. The acceptable value is an integer in the range of 1-3.
- **interval-in-secs** *interval_in_secs*—Specify the interval between two heartbeats in seconds. The acceptable value is an integer.

Configuring NRF for Discovery of Network Function

This section describes how to configure NRF to enable discovery of an NF.

To configure NRF for discovering an NF, use the following configuration:

```
config
  nf-services nf_services_name
   discovery [ nchf-spendinglimitcontrol | nudr-dr ]
  service-repository service_repository
    cache-forever [ true | false ]
    disable-subscription [ true | false ]
    subscription-extension-in-minutes subscription_extension
    end
```

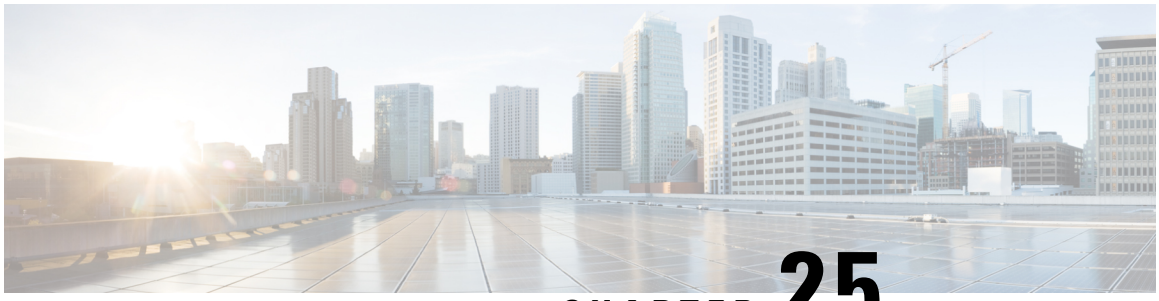
NOTES:

- **nf-services** *nf_services_name*—Specify network functions, such as registration, repository, and service discovery details.
- **discovery**—Enters the discovery configuration mode.
- **service-repository** *service_repository*—Specify the name of the repository that you configured in repositories.
- **cache-forever [true | false]**—Specify the discovery of services as "true" or "false" value. If this parameter is set to "true", then the discovered NFProfile cache does not expire at PCF.
- **disable-subscription [true | false]** —Specify the services as "true" or "false" to disable a subscription. If this is set to "true", then no subscription request is sent to NRF for the NF profile type.
- **subscription-extension-in-minutes** *subscription_extension*—Specify the duration by when you want to extend the subscription. PCF shows this value as validityTime in resubscription when the subscription validity time expires.

Troubleshooting Information

For message routing failures, check the datastore pod health and the logs for any issues.

For more information on how to check the pod health and logs, see Troubleshooting Information, on page 385.



CHAPTER 25

Network Slicing

Feature Summary and Revision History

Summary Data

Table 81: Summary Data

Applicable Product(s) or Functional Area	PCF
Applicable Platform(s)	SMI
Feature Default Setting	Enabled – Configuration required to disable
Related Documentation	Not Applicable

Revision History

Table 82: Revision History

Revision Details	Release
Enhancement introduced. Configuration updated for N5 interface service.	2022.02.0
First introduced.	2021.04.0

Feature Description

The network slicing solution allows the service providers to partition the 5G physical network into multiple virtual network slices.

PCF implements network virtualization by registering the Single–Network Slice Selection Assistance Information (S-NSSAIs) with the NRF. The S-NSSAI enables PCF to identify a network slice. After the registration is complete, SMF and AMF can discover the PCF instances serving the specific slices.

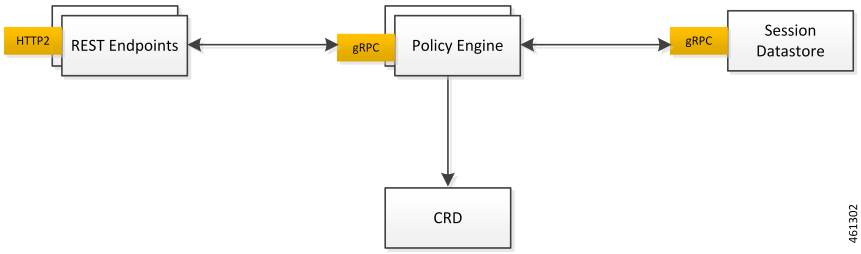
Note PCF supports only soft slicing, slice-based policy control, without isolating the system resources belonging to different slices.

Architecture

The REST endpoint performs the slice validation based on the requests from the client using HTTP2. The REST endpoint interacts with the Policy Engine to retrieve the policy status and the slice information over gRPC.

Slice information associated with the PDU session can be bound to CRD to generate the slice-specific policies.

Figure 34: Network Slice Architecture



How it Works

This section describes how this feature works.

Call Flows

This section describes the key call flows for this feature.

Slice Validation and Slice-Specific Policy Generation Call Flow

This section describes the Slice Validation and Slice-Specific Policy Generation call flow.

Figure 35: Slice Validation and Slice-Specific Policy Generation Call Flow

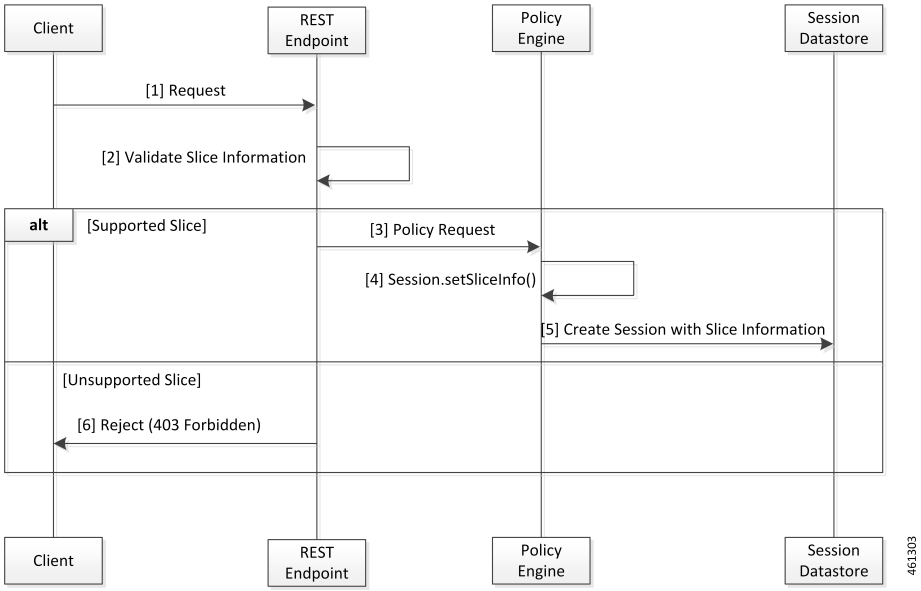


Table 83: Slice Validation and Slice-Specific Policy Generation Call Flow Description

Step	Description
1	The Client sends a request to validate the slice information to the REST endpoint.
2	The REST endpoint validates the slice information.
3	If the slice validation is successful, the REST endpoint sends a policy request to Policy Engine.
4	Policy Engine processes the request with the Session.setSliceInfo() message.
5	Policy Engine sends the Create Session request with the slice information to the Session Datastore.
6	If the slice validation is unsuccessful, the REST endpoint sends the Reject (403 Forbidden) message to the Client.

Configuring the Network Slicing Feature

Configuring this feature involves the following steps:

Configuring the Reject Requests Capability

This section describes how to enable the capability to reject requests from a slice that PCF does not support.

To enable PCF to reject requests, use the following configuration:

```
config
   advance-tuning slicing access-control [ enabled | disabled ]
   end
```

NOTES:

- **slicing access-control [enabled | disabled]**—Enable or disable PCF to reject the requests from the unsupported slices with the HTTP error code.

Configuring the Custom Error Codes

This section describes how to configure the error codes for the requests that PCF rejects.

To configure the custom error codes, use the following configuration:

```
config
   advance-tuning slice-access-control rejection-status-code error_code
   end
```

NOTES:

- **advance-tuning slice-access-control rejection-status-code** *error_code*—Specify the error code that must be displayed when PCF rejects a request. It must be an integer in the range of 100-599.
- If the error code is not configured, the default error code is 403.

Configuring the Allowed NSSAIs

This section describes how to configure the allowed NSSAIs in the PCF Registration Profile.

To configure allowed-NSSAIs, use the following configuration:

```
config
  service-registration
    profile
      allowed-nssais snssai_name sst sst_value [ sd sd_value ]
    services
      afService
        allowed-nssais snssai_name sst sst_value [ sd sd_value ]
      smfService
        allowed-nssais snssai_name sst sst_value [ sd sd_value ]
        end
```

NOTES:

- **allowed-nssais** *snssai_name* **sst** *sst_value* **[sd** *sd_value* **]**—Configures the SNSSAI. The *snssai_name* name is a logical identifier that is local to PCF. This name is not used in the PCF NFProfile when registering with NRF.

  To configure multiple slices per service, configure SNSSAI with same SST and different SD values.

  The **allowed-nssais** configured for smfService takes precedence over the allowed-nssais value configured at the profile-level.

Note Ensure to configure the allowed-nssais at the profile-level.

Configuration changes to the allowed-nssai of services do not affect the PDU sessions that are created before the configuration is modified.

Configuration Example

The following is an example configuration.

```
service-registration profile snssais embb-1
   sst 1
exit
service-registration profile snssais embb-2 sst 1
   sd 0000a1
exit
service-registration profile allowed-nssais name embb-1
   sst 1
exit
service-registration profile allowed-nssais name embb-2
   sst 1
   sd 0000a1
exit
service-registration services smfService
  allowed-nssais name embb-2 sst 1
     sd 0000a1
    exit
exit
```

Network Slicing OA&M Support

This section describes operations, administration, and maintenance information for this feature.

Statistics

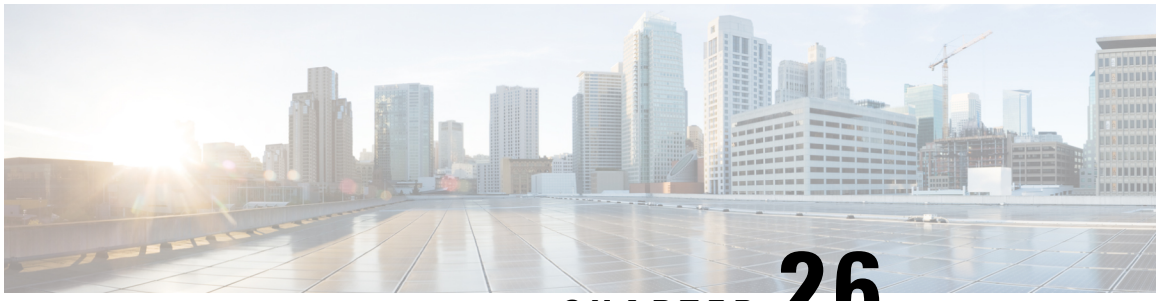
This section provides the counter that gets generated for the network slicing scenarios.

- inbound_request_slice_rejected: Captures the requests initiated for specific slices and the requests rejected for the slices that PCF does not support. The inbound_request_slice_rejected counter monitors requests that contain the slice information (Npcf_SMPolicyControl_Create).

  Note The inbound_request_slice_rejected does not determine the traffic on the slice.

  The inbound_request_slice_rejected counter supports the following labels:

  - interface_name—Indicates the name of the Service Based Interface (SBI) such as N7.
  - service_name—Indicates the name of the service such as npcf-smpolicycontrol.
  - operation_name—Indicates the name of the service operation such as Npcf_SMPolicyControl_Create.
  - command—Indicates the command type such as Create.
  - slice—Indicates the S-NSSAI that corresponds to the slice such as 1:0000ab.



CHAPTER 26

NRF Interface

Feature Summary and Revision History

Summary Data

Table 84: Summary Data

Applicable Product(s) or Functional Area	PCF
Applicable Platform(s)	SMI
Default Setting	Disabled – Configuration required to enable
Related Documentation	Not Applicable

Revision History

Table 85: Revision History

Revision Details	Release
Enhancement introduced. Configuration updated for N5 interface service.	2022.01.0
Enhancement introduced. PCF supports dual stack (IPv4 and IPv6) connectivity on all NRF external interfaces/endpoints.	2022.01.0

Revision Details	Release
Enhancement introduced. PCF supports IPv6 connectivity on all NRF external interfaces/endpoints.	2021.04.0
Enhancement introduced. Added new PCF attributes - priority and capacity	2020.02.0
Enhancement introduced. Introduced instructions on how to register an NF profile with NRF.	2020.01.0
First introduced.	Pre 2020.01.0

Feature Description

The NRF provides a fabric for all the NFs to register their profile and the supported services which facilitate in discovering each other. The registration enables the NFs to discover the other NFs based on the NF Type, Instance ID, and other conditions. In a broader view, this enables the NFs to exchange information that is required to carry out the diversified service requirements outlined for each NF.

PCF supports both IPv4 and IPv6 connectivity on its external endpoints (inbound and outbound).

With compliance to the *3GPP December 2018 29.510v15.2.0* specification, NF is equipped to use the NRF management and discovery services. These services allow you to invoke the following service operations:

NRF Management Services (nnrf-nfm)

- PCF uses the NFRegister service to register its profile and other parameters with the NRF. The registration process involves of PCF registering the npcf-am-policy-control and npcf-smpolicycontrol services with the NRF along with the list of services that the PCF instances expose.

  Note PCF endpoint registers with the NRF only if there is a reachable pcf-engine. Registration is complete when the heartbeat between the endpoint and engine is successful. If the heartbeat fails, the deregistration process is initiated.

- PCF uses the NFDeregister service to deregister its NF profile and the services that it has registered in the NRF. The NFDeregister service is initiated during a graceful endpoint shutdown.
- PCF applies the NFStatusSubscribe service to subscribe to the notifications when the NF_REGISTERED, NF_DEREGISTERED, and NF_PROFILE_CHANGED events occur on the individual NF instance. The instance is associated with the registered service, such as nchf-spendinglimitcontrol (CHF) and nudr-dr (UDR).
- The NFStatusNotify service enables the NRF to notify the subscribed PCF when the status of the individual NF instance change.
- PCF uses the NFStatusUnsubscribe service to unsubscribe to the notifications that are invoked when the status of an NF instance changes.

- PCF that is registered in NRF periodically contacts the NRF by sending a heartbeat. PCF attempts the contact by invoking the NFUpdate service operation to indicate that it is still operative.
- PCF monitors the NF profile (NFProfile) by periodically polling the NFProfile configuration to determine the modified parameters. If it detects a modified parameter, then PCF informs NRF about the update by sending a PATCH request containing the details of the modified parameter.
- When the PCF's registration status changes from REGISTERED to UNDISOCVERABLE or conversely in the NFStatus, PCF sends a PATCH request to NRF for the new status.

NRF Discovery Service (nnrf-disc)

- PCF uses the discovery service to discover the CHF and UDR NFs that support the nchf-spendinglimitcontrol and nudr-dr services.

How it Works

This section describes how this feature works.

At the startup, PCF registers its profile with the NRF endpoint of the highest priority. After the registration is complete, it periodically sends a heartbeat to the NRF along with its profile.

When PCF requires a service of another NF, it checks for the profile of that service in the cache. If PCF detects the NFProfile (profile), then it uses the information to consume the service. If the NFProfile is not found in the cache, PCF uses the configured NRF endpoints to discover the NF to which the service belongs. The information that is fetched by the discovery service is stored in the cache and reused until the validity period is met. If PCF does not find the NRF endpoint for discovery or receives an invalid response, it falls back on the local configuration looking for the required service.

After discovering the service from the NRF endpoint, PCF subscribes to the NRF for changes that happen in the NF profile. In response, a notification URI is called back for the event notification.

PCF updates the cache when NRF notifies it about the changes such as registration, deregistration, and modifications that happen in the NFPofile.

PCF periodically polls the NFProfile to determine the updated NF parameters. If it detects a modified parameter, PCF updates the configuration that is running. If PCF is registered to an NRF, then it sends a PATCH request to that NRF containing the details of the modified parameter in the payload. For example, `[{"op":"replace","path":"/capacity","value":33}]`.

If PCF determines that it is not registered (or deregistered) to an NRF, then it does not start the NRF Update Request.

The endpoint selection of the NF and NRF endpoints for registration and discovery is based on the probabilistic load-balancing algorithm (IETF RFC 2782) that uses priority and capacity parameters. In addition, for the discovery service, the locality of the NF is used in the algorithm as:

1. The first set of NFs is from the preferred locality which are sorted based on the locality for priority or capacity in the profile and endpoint.
2. The second set of NFs is from the geo-server locality which are sorted among the locality for priority or capacity in the profile and endpoint.
3. The third set of NFs is from the discovered NFs. These NFs are not part of the first and second set.
4. The fourth set contains the locally configured NFs.

Note Before PCF is shut down, it unregisters its profile and unsubscribes to the events that it has subscribed to.

Standards Compliance

This feature complies with the following standards specifications:

- *3GPP TS 29.510 version 15.0.0 (2018-06) "Network Function Repository Services"*
- *3GPP TS 29.510 version 15.2.0 (2018-12) "Network Function Repository Services"*
- *3GPP TS 29.510 CR#124 "Network Function Repository Services"*
- *3GPP TS 29.571 version 15.2.0 "Common Data Types for Service Based Interfaces"*

Configuring the PCF Profile

This section describes how to configure the PCF profile with NRF.

PCF registration involves associating the PCF profile with the NRF and registering the services such as npcf-am-policy-control and npcf-smpolicycontrol with the NRF. If you do not register any service, then the smfService is registered as the default service.

A PCF instance is discoverable by other NFs only after the PCF profile successfully registers with NRF. The PCF invokes the NFRegister service to complete the profile registration with the NRF.

Note Each NF Profile has a mapped Instance ID which the other NFs use to determine the profile.

To configure a PCF profile, use the following configuration in the Policy Ops Center console:

```
config
  service-registration
    profile
      allowed-plmns [ mcc mnc ]
        mcc mcc
        mnc mnc
      capacity pcf_capacity
      instance-id  instance_id
      locality locality_string
      pcf-info
        dnn-list dnn_list_name
        supi-ranges [ supi-range-id ]
          supi-range-id supi_range_id
            start start_integer
            end end_integer
            pattern regular_expression
      plmn-list [ mcc mnc ]
        mcc mcc
```

```
      mnc mnc
    priority pcf_priority
    snssais [ sst sd ]
      sst sst
      sd sd
  services
    [afService | smfService]
      allowed-nssais [ sst sd ]
       sst sst
       sd sd
      allowed-plmns [ mcc mnc ]
       mcc mcc
       mnc mnc
      api-version [ 1.0.0 | 1.0.2 ]
      end
```

NOTES:

- **service-registration**—Enters the service registration configuration mode.
- **profile** —Enter the profile configuration mode.
- **allowed-plmns [mcc mnc]**—Specify the PLMN code which is identified by a globally unique. The PLMN consists of Mobile Country Code (MCC) and Mobile Network Code (MNC). Typically, it is a 5 – 6 integers that identify a country, and a mobile network operator in that country represented in the form 001-01 or 001-001.
- **mcc** *mcc*—Specify the MCC value. Comprises of 3 integers.
- **mnc** *mnc*—Specify the MNC value. Comprises of 2–3 integers.
- **capacity** *pcf_capacity*—Specify the PCF profile's capacity. *pcf_capacity* must be an integer in the range is 0-65535.
- **instance-id** *instance_id*—Specify the service registration ID of the profile instance.
- **locality** *locality*—Specify the location of the NF instance such as geographic location and data center.
- **pcf-info**—Configures the PCF information such as Data Network Name and SUPI information.
- **dnn-list** *dnn_list_name*—Specify the Data Network Name (DNN) list name.
- **supi-ranges** *supi_range*—Specify the ranges of SUPIs, which the AUSF instance serves. If you do not specify a SUPI range, the AUSF instance determines a SUPI to serve.
- **supi-range-id** *supi_range_id*—Specify the SUPI range identifier.
- **start** *start*—Specify the initial value of a SUPI range. This value permits integers such as IMSI range.
- **end** *end*—Specify the last value of the SUPI range. This value permits integers such as IMSI range.
- **pattern** *pattern*—Specify a regular expression according to the ECMA-262 dialect that represents the set of SUPIs belonging to the specified range.
- **plmn-list [mcc mnc]** —Configures the PLMN code of the network function. Specifies the PLMN code which is a unique code. The PLMN consists of MCC and MNC. Typically, it is a 5–6 integers that identify a country, and a mobile network operator in that country represented in the form 001-01 or 001-001.

- **priority** *pcf_priority*—Specify the PCF profile's priority order. *pcf_priority* must be an integer in the range is 0-65535.
- **snssais [sst sd]**—Configures the S-NSSAIs of the network function.
- **sst** *sst*—Specify the Slice or Service Type to signify the expected Network Slice behaviour in terms of features and services. The acceptable range is 0–255.
- **sd** *sd*—Specify complements one or more Slice or Service Types to allow differentiation among multiple Network Slices of the same Slice or Service Type. Specifies the Slice Differentiator in a hexadecimal representation.
- **services** —Enters the services configuration mode.
- **allowed-nssais [sst sd]**—The Serving PLMN provides the NSSAI during the registration procedure. The NSSAI consists of the S-NSSAI values, which the UE uses in the serving PLMN for the current registration.
- **api-version** *api_version*—Specify the API version of the services that are deployed. The default version is 1.0.0.

Defining the PCF Registration Status

This section describes how to configure the PCF's registration status.

The registration status of PCF reflects its capability to transact with NRF and other NFs. The PCF instance that is registered with an NRF periodically contacts that NRF by invoking the NFUpdate service operation to indicate that it is operative.

You can now define the registration status as UNDISCOVERABLE. The UNDISCOVERABLE status is typically assigned when you want to perform preventive maintenance, or operations and maintenance activities. During this period, PCF would be in a dormant state, which means all the operations involving the PCF instance are suspended.

The feature to modify the registration status is compliant with *3GPP TS 29.510 CR 124*.

To configure the registration state as UNDISCOVERABLE, use the following configuration in the Policy Ops Center console:

```
config
   service-registration profile nf-status
   [ REGISTERED | UNDISCOVERABLE ]
   end
```

NOTES:

- **service-registration profile nf-status [REGISTERED | UNDISCOVERABLE]** —Configures the network function's registration status. The default NFStatus is REGISTERED.

Configuring the NRF Endpoint for Management Services

This section describes the configurations that you must perform to enable the NRF's management services.

1. Configuring the NRF Endpoint Group

2. Configuring the Management Service

Configuring the NRF Endpoint Group

This section describes how to configure the NRF Groups.

To configure the nnrf-nfm service for enabling the management service, use the following configuration in the Policy Ops Center console:

```
config
  group
   nrf
    mgmt [ name ]
     name nrf_group_name
     service
     type service_type
      nrf [ nrf-service-name ]
       nrf-service-name nrf_service_name
       endpoint-profile [ name ]
         name endpoint_profile_name
         capacity endpoint_capacity
         priority  endpoint_priority
         api-uri-prefix uri_prefix
         api-root  api
         uri-scheme  uri_scheme
         version
       uri-version [ name ]
        name version_name
        full-version full_version
       endpoint-profile [name]
        name endpoint_name
        priority endpoint_priority
        capacity endpoint_capacity
        primary ip-address
          ipv4 ipv4_address
          ipv6ipv6_address
          fqdn fqdn
          port port_number
        secondary ip-address
          ipv4 ipv4_address
          ipv6  ipv6_address
          fqdn fqdn
          port port_number
        tertiary ip-address
          ipv4 ipv4_address
          ipv6 ipv6_address
          fqdn fqdn
          port port_number
          end
```

- **group**—Enters the group configuration mode.

- **nrf**—Enters the NRF configuration mode.
- **mgmt [name]** —Enters the management configuration mode.
- **name** *nrf_group_name*—Specify the name of the nrf group.
- **service**—Enters the service configuration mode.
- **type** *service_type*—Specify the configured NF service types. The service types vary depending on the configured service. The PCF service supports the nnrf-nfm service.
- **nrf-service-name** *nrf_service_name*—Specify the NRF service name.
- **endpoint-profile [name]**—Enters the endpoint profile configuration mode.
- **name** *endpoint_profile_name*—Specify the name of the endpoint profile.
- **api-uri-prefix** *uri_prefix*—Specify the apiName. If not configured, it takes the standard API name for the service as per the specification.
- **api-root** *api*—Specify the deployment-specific service API prefix that is used within the apiRoot.
- **uri-scheme** *uri_scheme*—Specify the URI scheme as HTTP or HTTPs.
- **uri-version**—Specify the api/Version and the version number. The full version format is <Major-version>.<Minor-version>.<patch-version>.[alpha-<draftnumber>].
- **endpoint-name**—Specify the endpoint name and priority for the service to select the appropriate profile using the load-balancing logic. The priority must be an integer in the range of 0-65535. Capacity denotes the node capacity for the endpoint. It must be an integer in the range of 0-65535.
- **primary ip-address**—Specify the IP address, FQDN, and Port for the primary endpoint.
- **secondary ip-address**—Specify the IP address, FQDN, and port number for the secondary endpoint.
- **tertiary ip-address** —Specify the IP address, FQDN, and port number for the tertiary endpoint.

NOTES:

Configuring the Management Service

This section describes how to enable the management service for the NRF.

To configure the NRF Management service, PCF locality, and associating them to the NRF Endpoint, use the following configuration in the Policy Ops Center console:

```
config
  group
   nf-mgmt [ name ]
     name nf_management_group_name
     nrf-mgmt-group  -> /group/nrf/mgmt/name
      locality locality
     failover
       sla
         reconnect
           interval interval
           end
```

NOTES:

- **group**—Enters the group configuration mode.
- **nf-mgmt [name]**—Specify the management group that is associated to a network function.
- **locality** *locality*—Specify the NF locality.
- **failover**—Enters the failover configuration mode.
- **sla**—Enters the sla configuration mode.
- **reconnect**—Enters the reconnect configuration mode.
- **interval** *interval*—Specify the time interval after which NF must attempt a reconnect operation.

Configuring the NRF Endpoint for Discovery Service

This section describes the configurations that you must perform to enable NRF's discovery services.

1. Configuring the NRF Endpoint Group
2. Configuring the Discovery Service
3. Configuring the Local NF Endpoint

Configuring the NRF Endpoint Group

This section describes how to configure the NRF endpoint groups for the discovery of different NFs using the discovery (nnrf-disc) service.

To enable discovery of the NRF groups, use the following configuration in the Policy Ops Center console:

```
config
    profile
    nrf
    discovery [ name ]
     name discovery_group_name
     service
      type service_type
       nrf [ nrf-service-name ]
       nrf-service-name nrf_service_name
       endpoint-profile [ name ]
        name endpoint_profile_name
        capacity endpoint_capacity
        priority endpoint_priority
        api-uri-prefix uri_prefix_string
        api-root api
        uri-scheme uri_scheme
        version
         uri-version [ name ]
          name version_name
          full-version full_version
```

```
endpoint-name
 name endpoint_name
 priority endpoint_priority
 capacity endpoint_capacity
primary ip-address
 ipv4 ipv4_address
 ipv6 ipv6_address
 fqdn fqdn
 port port_number
secondary ip-address
 ipv4 ipv4_address
 ipv6 ipv6_address
 fqdn fqdn
 port port_number
tertiary ip-address
 ipv4 ipv4_address
 ipv6 ipv6_address
 fqdn fqdn
 port port_number
 end
```

NOTES:

- **profile**—Enters the **profile** configuration mode.
- **nrf**—Enters the **nrf** configuration mode.
- **discovery [name]**—Enters the **discovery [name]** configuration mode.
- **name** *discovery_group_name*—Specify the name of the discovery group. Discovery group is the logical link to the NRF endpoint groups (nrf-group). For each NF type, you can associate a discovery group and the locality information.
- **type** *service_type*—Specify the configured NF service types. The service types vary depending on the configured service. The PCF service supports the nnrf-disc service.
- **nrf-service-name** *nrf_service_name*—Specify the NRF service name.
- **endpoint-name** —Specify the endpoint's name and priority for the service to select the appropriate profile using the load-balancing logic. The priority must be an integer in the range of 0-65535. Capacity denotes the node capacity for the endpoint. It must be an integer in the range of 0-65535.
- **api-uri-prefix** *uri_prefix_string*—Specify the {apiName}. If not configured, it takes the standard API name for the service as per the specification.
- **api-root** *api*—Specify the deployment-specific service API prefix that is used within the apiRoot.
- **uri-scheme** *uri_scheme*—Specify the URI scheme as HTTP or HTTPs.
- **uri-version { name** *version_name* **| full-version** *full_version***}**—Specify the api/Version and the version number. The full version format is <Major-version>.<Minor-version>.<patch-version>.[alpha-<draftnumber>].
- **primary ip-address**—Specify the IP address, FQDN, and port number for the primary endpoint.
- **secondary ip-address**—Specify the IP address, FQDN, and port number for the secondary endpoint.

- **tertiary ip-address**—Specify the IP address, FQDN, and port number for the tertiary endpoint.

Configuring the Discovery Service

This section describes how to enable the discovery service for the NRF.

To configure the NRF Discovery and PCF locality and associating them to the NRF Endpoint, use the following configuration in the Policy Ops Center console:

```
config
    profile
      nf-pair
        nf-type [ type ]
          type nf_type
        nrf-discovery-group -> /group/nrf/discovery/name
        subscription-enabled subscription_status
        subscription-extension extension_value
        locality
          client -> /service-registration/profile/locality
          preferred-server server_name
          geo-server geo_server
          end
```

NOTES:

- **type** *nf_type*—Specify one or more NF types such as AMF, CHF, PCF, and UDM as the network element profile.
- **subscription-enabled** *subscription_status*—Specify if PCF is enabled to subscribe to notifications related to the discovered service.
- **subscription-extension** *extension_value*—Specify the duration (in minutes) for which the subscription is extended.
- **preferred-server** *server_name*—Specify the preferred server locality information. Preferred server locality is the locality that is considered as the locality of preference during the corresponding NF discovery.
- **geo-server** *geo_server*—Specify the geo-server locality information. Geo-server locality is a geo redundant site for the preferred locality and is used as the next suitable server locality after preferred locality, during NF discovery.

Configuring the Local NF Endpoint

This section describes how to configure the local NF endpoint.

The PCF becomes aware of the various NFs in the 5G fabric through the NF discovery service that is exposed by the NRF or through the CLI configuration. If the NRF is unavailable, then PCF relies on the local configuration of the NF endpoints to discover the NFs.

To configure the local configuration for the NF services that PCF uses, use the following configuration in the Policy Ops Center console:

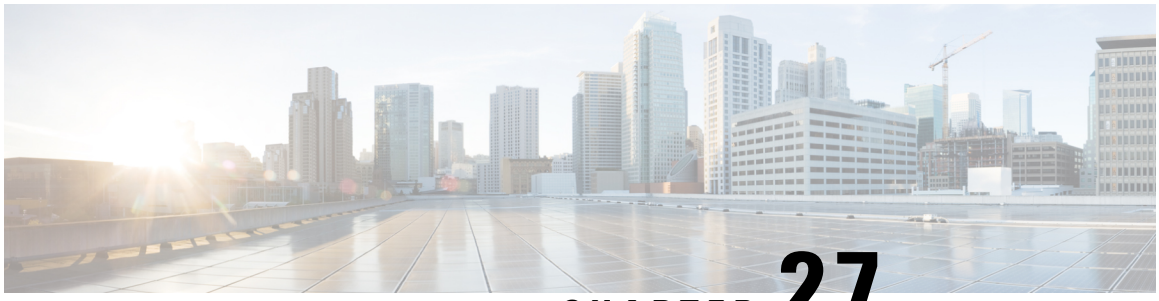
```
config
  profile
  nf-client
   nf-type
    udr
    udr-profile [ name ]
      name udr_profile_name
      locality
        name udr_locality_name
        priority priority
     sevice
      name service_name
      type [ type ]
       type service_type
       endpoint-profile [ name ]
         name endpoint_profile_name
         capacity endpoint_capacity
         priority endpoint_priority
         api-uri-prefix uri_prefix_string
         api-root api
         uri-scheme uri_scheme
         version
           uri-version [ name ]
             name version_name
             full-version full_version
       endpoint-profile [ name ]
         name endpoint_name
         priority endpoint_priority
         capacity endpoint_capacity
       primary ip-address
         ipv4 ipv4_address
         ipv6 ipv6_address
         port port_number
       secondary ip-address
         ipv4 ipv4_address
         ipv6 ipv6_address
         port port_number
       tertiary ip-address
         ipv4 ipv4_address
         ipv6 ipv6_address
         port port_number
    chf-profile [ name ]
     name chf_profile_name
     locality [ name ]
      name locality_name
      priority priority
     service
      name service_name
      type [ type ]
       type service_type
     endpoint-profile [ name ]
```

```
endpoint-profile [ name ]
  name endpoint_profile_name
  capacity endpoint_capacity
  priority endpoint_priority
  api-uri-prefix uri_prefix_string
  api-root api
  uri-scheme uri_scheme
  version
    uri-version [ name ]
      name version_name
      full-version full_version
endpoint-profile [ name ]
  name endpoint_name
  priority endpoint_priority
  capacity endpoint_capacity
primary ip-address
  ipv4 ipv4_address
  ipv6 ipv6_address
  port port_number
secondary ip-address
  ipv4 ipv4_address
  ipv6 ipv6_address
  port port_number
tertiary ip-address
  ipv4 ipv4_address
  ipv6 ipv6_address
  port port_number
  end
```

NOTES:

- **udr-profile [name]**—Enter the UDR profile configuration mode.
- **name** *udr_profile_name*—Specify the name of the UDR profile.
- **type** *service_type*—Specify the configured NF service types. The service types vary depending on the configured service.
- **nrf-service-name** *nrf_service_name*—Specify the NRF service name.
- **api-uri-prefix** *uri_prefix_string*—Specify the apiName. If not configured, it takes the standard API name for the service as per the specification.
- **api**—Specify the deployment-specific service API prefix that is used within the apiRoot.
- *uri_scheme*—Specify the URI scheme as HTTP or HTTPs.
- **uri-version**—Specify the API/version and the version number. The full version format is <Major-version>.<Minor-version>.<patch-version>.[alpha-<draftnumber>].
- **endpoint-name**—Specify the endpoint name and priority for the service to select the appropriate profile using the load-balancing logic. The priority must be an integer in the range of 0-65535. Capacity denotes the node capacity for the endpoint. It must be an integer in the range of 0-65535.
- **primary ip-address**—Specify the IP address, FQDN, and port number for the primary endpoint.

- **secondary ip-address**—Specify the IP address, FQDN, and port number for the secondary endpoint.
- **tertiary ip-address**—Specify the IP address, FQDN, and port number for the tertiary endpoint.



CHAPTER 27

N28 Interface

Feature Summary and Revision History

Summary Data

Table 86: Summary Data

Applicable Product(s) or Functional Area	PCF
Applicable Platform(s)	SMI
Feature Default Setting	Enabled – Always-on
Related Documentation	Not Applicable

Revision History

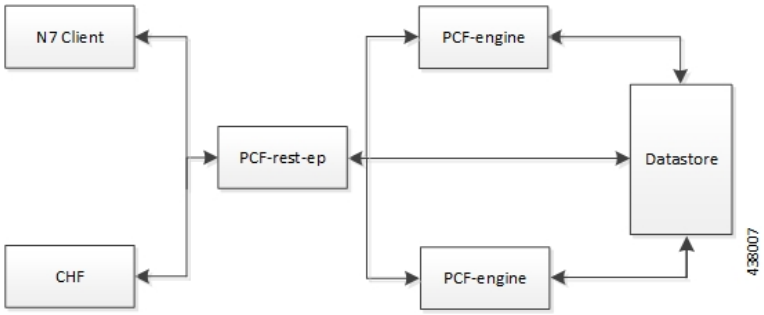
Table 87: Revision History

Revision Details	Release
First introduced.	2020.01.0

Feature Description

The N28 interface supports the key charging and quota handling scenarios.

Figure 36: N28 Interface



PCF performs the following capabilities through the N28 interface:

- Retrieving or subscribing to policy counter information from Charging Function (CHF) over N28 for use in policy decisions over N7 only. This includes subscription to specific counters or all.
- Support for receiving notifications for policy counter information changes from CHF and using the information for policy decisions.
- Support for using the retrieved counters in policy decisions through Virtual Services (VS).
- This includes subscription to specific counters or all Support for Service Based Architecture (SBA) interface toward CHF. Currently, it supports:
  - Initial Subscribe toward CHF on N7 session creation (if enabled).
  - Notify from CHF.
  - Unsubscribe toward CHF on N7 session termination.

Note Intermediate Subscribe and CHF driven termination is currently not supported.

- NRF discovery of CHF:
  - PCF also supports the local configuration for CHF endpoints.
  - If CHF endpoints are configured locally, the configured endpoints are used, and discovery may be skipped.
  - Currently discovery is only supported by NFType and does not support any criteria.
- Endpoints caching - Locally cache and reuse of the discovered CHF endpoints for sending N28 messages.
- Random Load Balancing for cached Endpoints.

How it Works

This section describes how this feature works.

The Nchf_SpendingLimitControl service enables the NF service consumer to retrieve policy counter status information per UE from the CHF by subscribing to spending limit reporting (that is notifications of policy counter status changes).

If the spending limit reporting is no more required, the Nchf_SpendingLimitControl service enables the NF service consumer to unsubscribe from the reporting.

On receiving an N7 Create a Session request, if the N28 lookup or counter subscription is configured, PCF Engine triggers a session creation and subscription toward CHF. PCF then retrieves the counter information from the CHF response and generates virtual services for each counter which are used for making policy decisions.

If the errors or timeouts policy decisions continue without N28 counter information or policy (N7 response is success but excludes N28 based policy), then the N7 session and N28 session terminate.

The interface or model details for the N28 interface are provided in *3GPP TS29.594*.

Call Flows

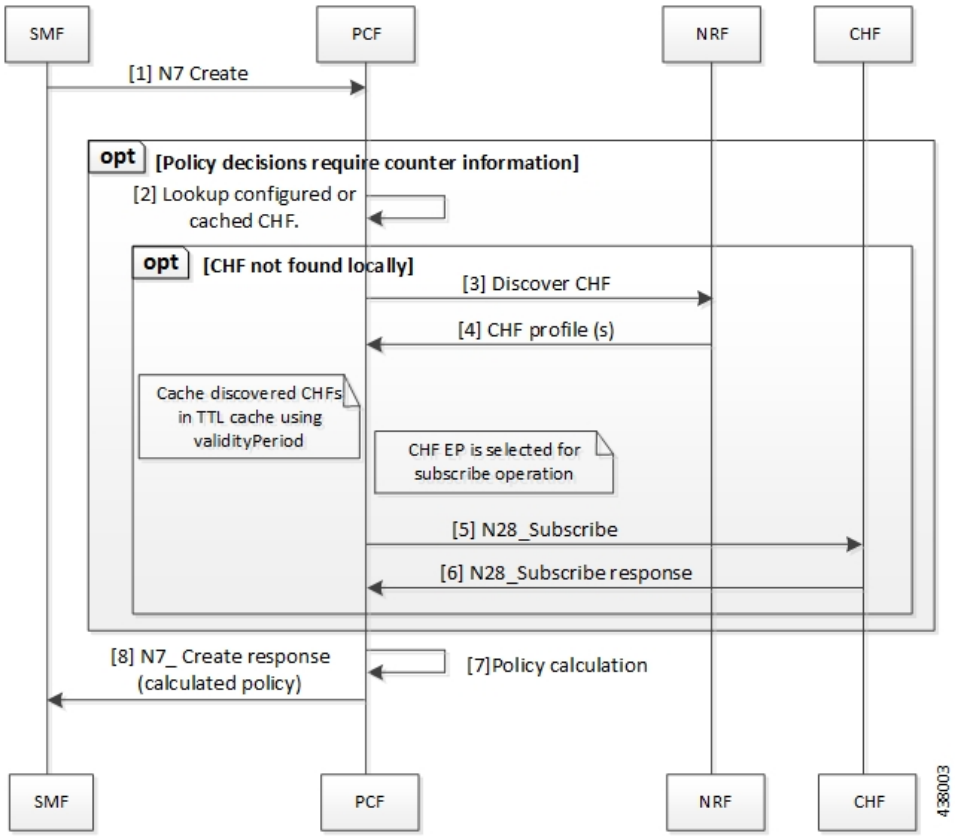
This section describes the key call flows for this feature.

Counter Subscription/Retrieval (N28 Session Creation)

This section describes the Counter Subscription/Retrieval (N28 Session Creation) call flow.

The decision to subscribe to N28 counters is determined based on the presence of the SpendingLimitRequest service configuration. If this service configuration is present in the policy, then the Policy Engine triggers the N28 session creation.

Figure 37: N28 Subscribe (N7 Create) Call Flow



Note

- **Counter retrieval**:
  - Generating the list of counters to subscribe is based on the SpendingLimitSubscription and RequestPolicyCounters service configuration: SpendingLimitSubscription also includes a list of counters to subscribe to. The RequestPolicyCounters service also satisfies the same role (providing a list of counters to subscribe to) and is expected to be used in cases where counters can come from different sources. For example, specific counters per LDAP attribute.
  - Policy Engine sends the subscribe request to PCF REST EP. The REST EP in turn attempts to lookup a CHF (based either on local configuration or via NRF discovery).
    - If no endpoint is available, error response is generated towards the engine.
    - If local endpoint is available, the REST EP invokes the Nchf_SpendingLimitControl_Subscribe operation towards the CHF.
      - If no response/error response is received, an error response is generated towards the engine for further action.
      - On success response, the counter information is forwarded to engine for further action.
    - If discovery is performed and endpoint is available, the REST EP invokes the Nchf_SpendingLimitControl_Subscribe operation towards the CHF which is handled as mentioned above.
      - The discovered CHF EP is also cached locally (in a TTL cache) so that it can be used for subsequent N28 operations.
      - For subsequent operations, the one of the locally cached EPs is randomly selected (that matches the selection criteria). Currently, only the NFType is supported as selected criteria.
      - The cached NFs are expired from local cache based on the ValidityPeriod provided by NRF in discovery response. Any subsequent operation that requires the EP will then result in a fresh discovery.
      - The discovered NF profile caching is generic and currently applicable for both CHF and UDR.

Table 88: N28 Subscribe (N7 Create) Call Flow Description

Step	Description
1	The SMF sends a N7 Create request to the PCF.
2	Based on the routing configuration, PCF configures the lookup or caches the CHF.
3	If CHF is not found locally, then the PCF sends a Discover CH request to NRF.
4	The NRF responds with the CHF profiles with the PCF.
5	The PCF sends a N28 Subscribe request to the CHF.

Step	Description
6	The CHF responds with the N28 Subscribe result to the PCF.
7	The PCF performs the policy calculation.
8	The PCF sends the N7 Create response after calculating the policy to the SMF.

Unsubscribe Counters (N28 Session Termination)

This section describes the Unsubscribe Counters (N28 Session Termination) call flow.

On receiving an N7 terminate request, PCF triggers an N28 Nchf_SpendingLimitControl_Unsubscribe request towards CHF for unsubscribing for changes in N28 counter information.

Figure 38: N28 Unsubscribe (N7 terminate) Call Flow

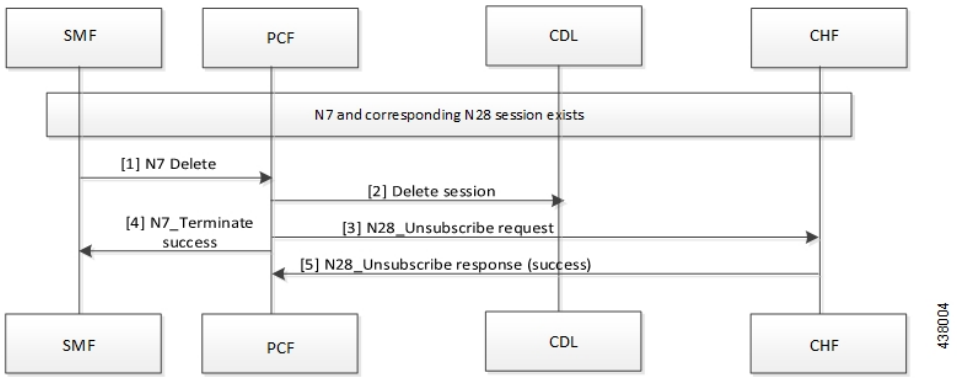


Table 89: N28 Unsubscribe (N7 terminate) Call Flow Description

Step	Description
1	The SMF sends a N7 Delete request to the PCF.
2	The PCF sends a Delete Session request to the CDL.
3	The PCF sends a N28 Unsubcribe request to the CHF.
4	The PCF forwards the N7 Terminate Success message to SMF.
5	If the unsubscribe request is successful, then the CHF sends the N28 Unsubscribe response to the CHF.

N28 Counter-Based Policy

Similar to existing Diameter Sy implementation, the counters retrieved from CHF are made available for policy decisions via Virtual Services (VS).

A Virtual Service (VS) is created per policy counter with counter ID and status as AVPs. This allows the binding of the counters to CRD tables for VS evaluation.

N28 Virtual Service Details

The N28 Virtual Service details are as follows:

- VS Name: Name will be of the format: CounterId-CounterStatus
- VS AVPs: Following AVPs will be added to the VS:
  - Code: counter-id, Value: the counter ID value
  - Code: counter-status, Value: the counter status

Notification of Counter Changes from CHF

This section describes the Notification of Counter Changes from CHF call flow.

In case of changes in the subscribed policy counters, the CHF notifies PCF using the Nchf_SpendingLimitControl_Notify operation. The PCF supports this operation through the PCF REST endpoint.

On receiving the notification, the REST EP performs a datastore lookup to determine the route and then forward the notification message to the selected engine group.

On the PCF Engine, the existing session is updated with the new counter information and policy is recalculated (using the new VS) and applicable decisions are pushed on the N7 interface towards SMF via N7 Notify operation.

Figure 39: N28 Notify

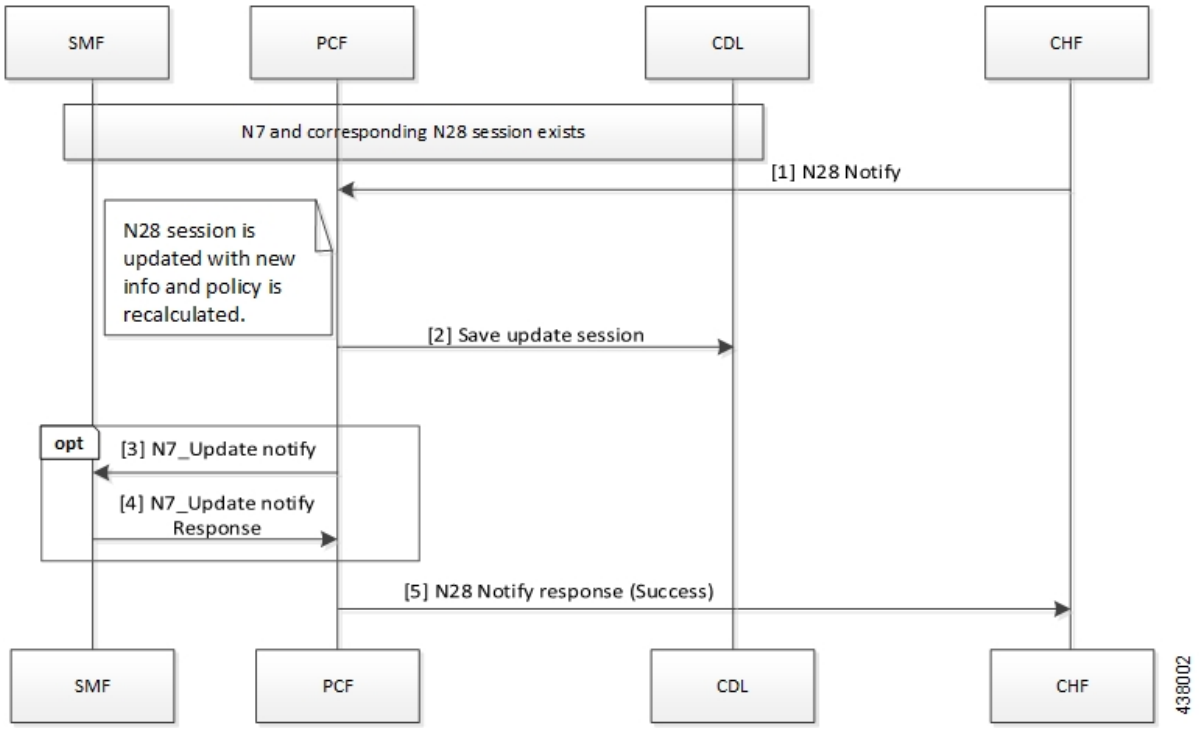


Table 90: N28 Notify Call Flow Description

Step	Description
1	If the N7 and N28 sessions are available, then CHF sends a N28 Notify request to the PCF.
2	After the N28 session is updated with the new information, the policy is recalculated and the updated session is saved in the CDL.
3	The PCF sends a N7 Update Notification message to the SMF.
4	The SMF sends response for the N7 Update Notification message to the PCF.
5	The PCF sends a N28 Notification Success response to the CHF.

Configuration Support for the N28 Interface

This section describes how to configure support for the N28 interface using the following services.

- SpendingLimitSubscription
- RequestPolicyCounters
- AvpServiceConfiguration

SpendingLimitSubscription

If SpendingLimitSubscription is configured in a policy, then the N28 session creation or subscription is triggered on session create. Only one instance of this configuration is allowed or else any random instance is picked.

The configuration includes subscriber identifiers (Subscriber SUPI and GPSI) and a list of Counter Ids to subscribe. The counters can be directly configured or can be pulled from other sources with the "Pull Value from…" configuration.

Before setting the service parameters, ensure that you create a use case template and add a service for this configuration. For details, see Configuring the Use Case Template, on page 293 and Adding a Service, on page 292.

RequestPolicyCounters

This section describes the parameters for the RequestPolicyCounters configuration.

Use this configuration to add counters in the subscription list while generating the N28 Subscribe request. Multiple instances of this configuration can exist. The application collects all instances and includes counters from all in the final CounterIds list (to subscribe).

Before setting the service parameters, ensure that you create a use case template and add a service for this configuration. For details, see Configuring the Use Case Template, on page 293 and Adding a Service, on page 292.

The following table describes the RequestPolicyCounters service parameters.

Table 91: RequestPolicyCounters Configuration Parameters

Parameters	Description
Priority	The priority of the message for processing. The higher the number, the higher the priority.
Policy Counter Group	Represents a logical name for the counter set included in the service configuration. The field pulls value from the OfferGroup column.
Policy Counter Id	Specifies the policy counter identifier name.

AvpServiceConfiguration

This section describes the parameters for the AvpServiceConfiguration configuration.

Before setting the service parameters, ensure that you create a use case template and add a service for this configuration. For details, see Configuring the Use Case Template, on page 293 and Adding a Service, on page 292.

The following table describes the AvpServiceConfiguration service parameters.

Table 92: AvpServiceConfiguration Configuration Parameters

Parameters	Description
Priority	The priority of the message for processing. The higher the number, the higher the priority.
Group Name	Specifies a group name. Only 1 per "Group Name" is allowed to be active. If multiple configurations are added highest priority per "Group Name" is used.
Code	Specifies a code for the AVP.
Value	Specifies a value for the AVP.

Troubleshooting

Perform the following when the message routing fails:

- Ensure that the SpendingLimitRequest service configuration is available and enabled in the subscribed service list in Policy Builder.
- If the CHF is configured locally, ensure that the URL is specified in the correct format. For CHFs that are not configured locally, make sure to enable the NF discovery.
- If discovery is enabled, ensure that the NRF URL is configured locally and is valid.
- Enable the DEBUG level for com.cisco.pcf.endpoint.routing and review the pcf-rest-ep logs for any issues.
- Review the data store pod health and the logs for information about the issues.

Configuring NF or Logical Groups

This section describes how to configure the NF locally or logical groups of the NFs.

To configure the NF or logical groups of the NFs, use following configuration in the Policy Ops Center console:

```
config
  network-function logical_group_name
  nf-info nf_type
  service-version version_in_uri
  http-endpoint list_of_base_urls
  end
```

NOTES:

- **network-function** *logical_group_name* —Specify the name for a logical group of NFs
- **nf-info** *nf_type* —Specify the type of NF that is configured. Currently, only NRF, CHF, and UDR are supported.
- **service-version** *version_in_uri* —Specify the version field in the resource URI for accessing the NF services.
- **http-endpoint** *list_of_base_urls* —Specify the base-urls that are used to consume services that are provided by the configured NF.

OAM Support

This section describes operations, administration, and maintenance support for this feature.

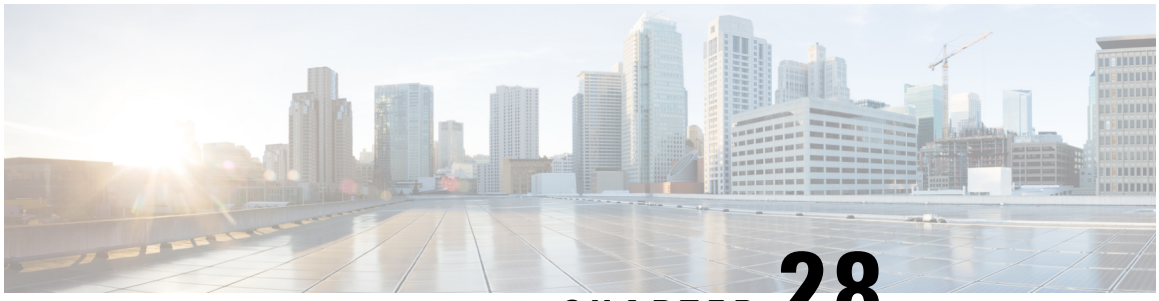
Statistics

This section provides the list of statistics and counters that are generated for the charging and quota handling scenarios.

The following metrics track the counter information:

- async_svc_runnable_total: Captures the total count of the async service runnable count.
- async_svc_runnable_total_seconds: Captures the total duration (in seconds) to process the async service runnable count.

For information on statistics, see *Ultra Cloud Core 5G Policy Control Function Statistics Reference*.



CHAPTER 28

Online Charging Enablement over N7 to SMF

Feature Summary and Revision History

Summary Data

Table 93: Summary Data

Applicable Product(s) or Functional Area	PCF
Applicable Platform(s)	SMI
Feature Default Setting	Disabled – Configuration required to enable
Related Documentation	Not Applicable

Revision History

Table 94: Revision History

Revision Details	Release
First introduced.	2020.01.0

Feature Description

PCF supports converged online and offline charging. As part of this support, PCF sends the CHF address to the SMF over the N7 interface. This allows the SMF to connect to the specified CHF for converged online and offline charging. In addition, PCF sends charging-specific attributes (charging decision attributes) in the PCC rules to SMF over the N7 interface.

How it Works

This section describes how this feature works.

The ability to send CHF addresses in "ChargingInformation" in SM policy create response is added to PCF. In the subsequent SM policy updates, the same address is sent to the SMF. Similarly, the ability to send charging decision attributes in the PCC rules is available in PCF.

The charging information includes primary and secondary CHF addresses. The charging decisions include the following attributes- chgId, meteringMethod, offline, online, ratingGroup, reportingLevel, serviceId, sponsorId, appSvcProvId, and afChargingIdentifier.

Note The charging decisions are supported only for a table-driven PCC and dynamic PCC rules.

Charging Information

- After the SM create control request is received, the PCF reads the charging information service configuration and adds the charging information in the PCF session (if it is not already added).
- PCF uses the charging information in the PCF session and sends the ChargingInformation field in "ChgDecs" in response.

Charging Data

- After the SM create control request is received, PCF retrieves the PCC rules using "TableDrivenDynamicPccRule" or "DynamicPccRule" service configurations.
- The PCC retrieves "ChgIds" (it can either be single chgid value or multiple based on comma separated values) from the "TableDrivenDynamicPccRule" and "ChgID" "from the DynamicPccRule" service configurations.
- PCF queries the Charging Data CRD table and retrieves the list of charging data to be sent, after the charging ids are found and "TableDrivenChargingDecisions" is configured.
- PCF creates response by adding all charging data under "ChgDecs" and also adds the reference in PCC rules by specifying the "refChgData" array.

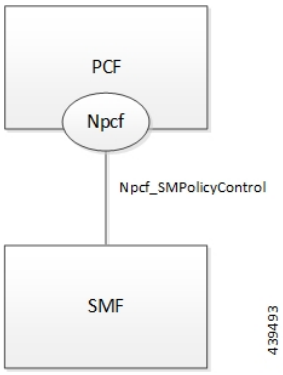
Call Flows

This section describes the key call flows for this feature.

Online and Offline Charging over N7 to SMF

This section describes the Online and Offline Charging over N7 to SMF call flow.

Figure 40: Charging over N7 Call Flow



Creating SM Policy

This section describes the Creating SM Policy call flow.

Figure 41: Create SM Policy Call Flow

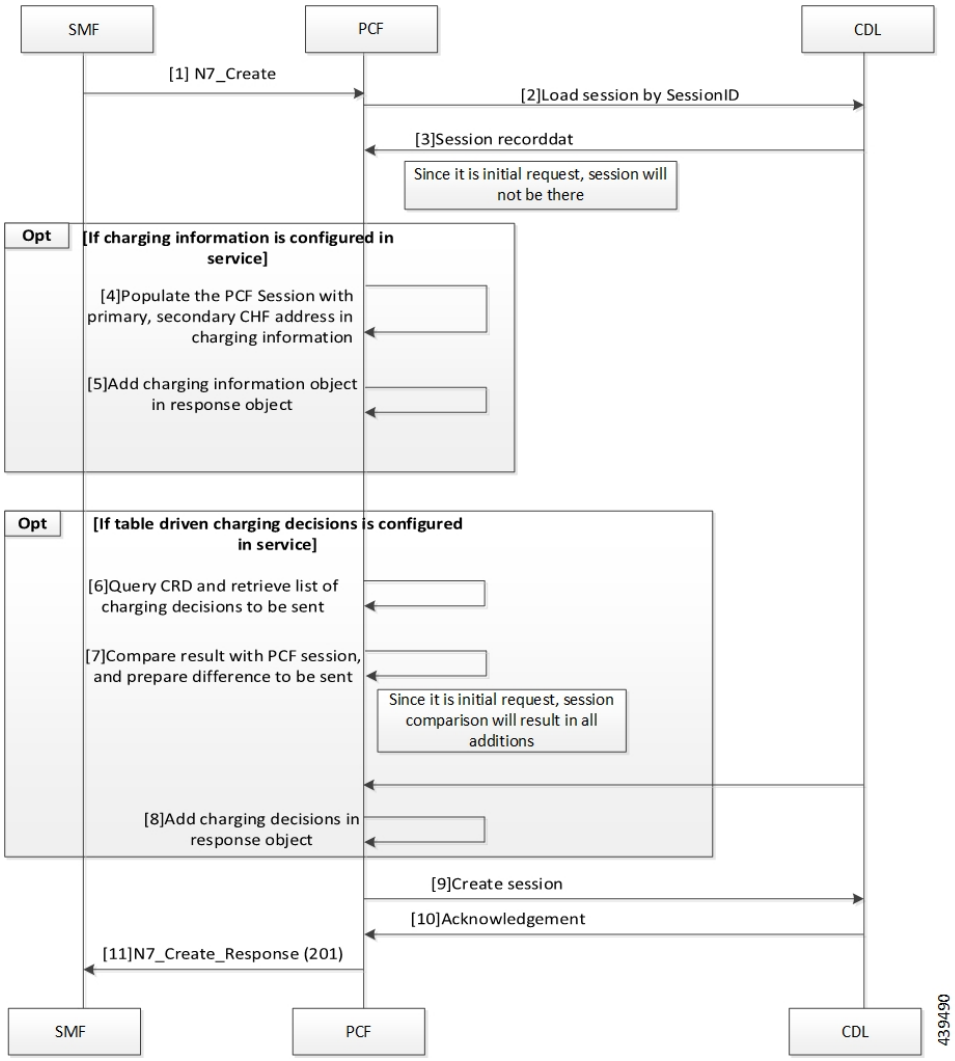


Table 95: Create SM Policy Call Flow Description

Step	Description
1	The SMF sends a N7 Create request to the PCF.
2	The PCF sends a Load Session request to the CDL.
3	The CDL sends a Session Record request to the PCF.
4	If the charging information is configured in the service, then PCF populates the PCF session with primary and secondary CHF address.
5	The PCF adds the charging information in the response object.
6	If the table-driven charging decision is configured in the service, then PCF queries the CRD to retrieve the list of charging decisions.

Step	Description
7	The PCF compares the results with the PCF session and identifies the differences.
8	The PCF adds charging decisions in the response object.
9	The PCF sends a Create Session request to the CDL.
10	In response, the CDL sends an acknowledgment to the PCF.
11	The PCF sends an N7 Create response to the SMF.

Updating SM Policy

This section describes the Updating SM Policy call flow.

Figure 42: Update SM Policy Call Flow

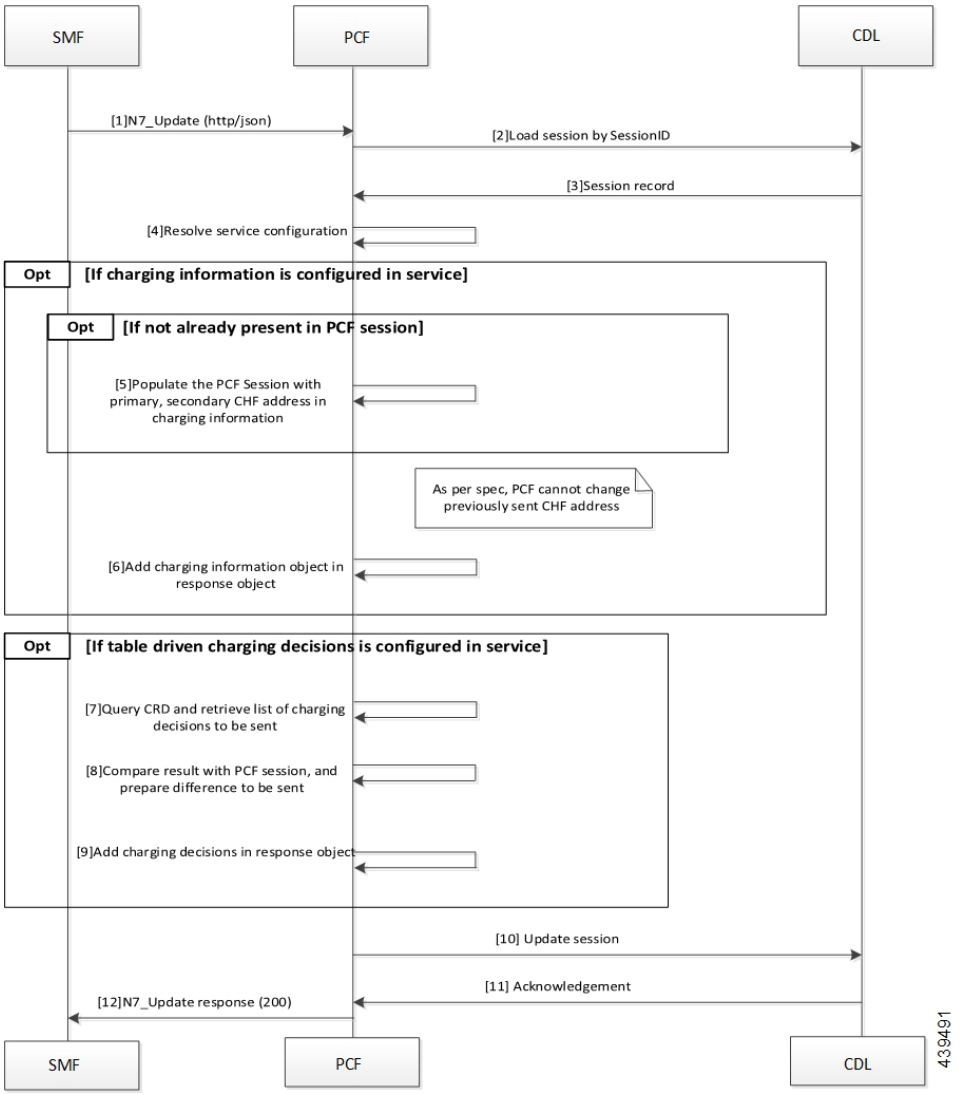


Table 96: Update SM Policy Call Flow Description

Step	Description
1	The SMF sends an N7 Update request to the PCF.
2	The PCF sends a load session by SessionID to the CDL.
3	In response, the CDL sends the Session Record to the PCF.
4	The PCF resolves the service configuration.
5	If the charging information is not available in the PCF session, then PCF populates the session with the primary and secondary CHF address.
6	If the charging information is configured in the service, then PCF adds the charging information object in the response object.
7	If the table driven charging decision is configured in the service, then PCF queries CRD to retrieve the list of charging decision.
8	The PCF compares the result with the PCF session and identifies the delta information.
9	The PCF adds the charging decision in the response object.
10	The PCF sends the Update Session request to the CDL.
11	The CDL acknowledges the update request by sending an acknowledgment to the PCF.
12	The PCF sends an N7 Update response to the SMF.

Updating Notify SM Policy

This section describes the Updating Notify SM Policy call flow.

Figure 43: Update Notify SM Policy Call Flow

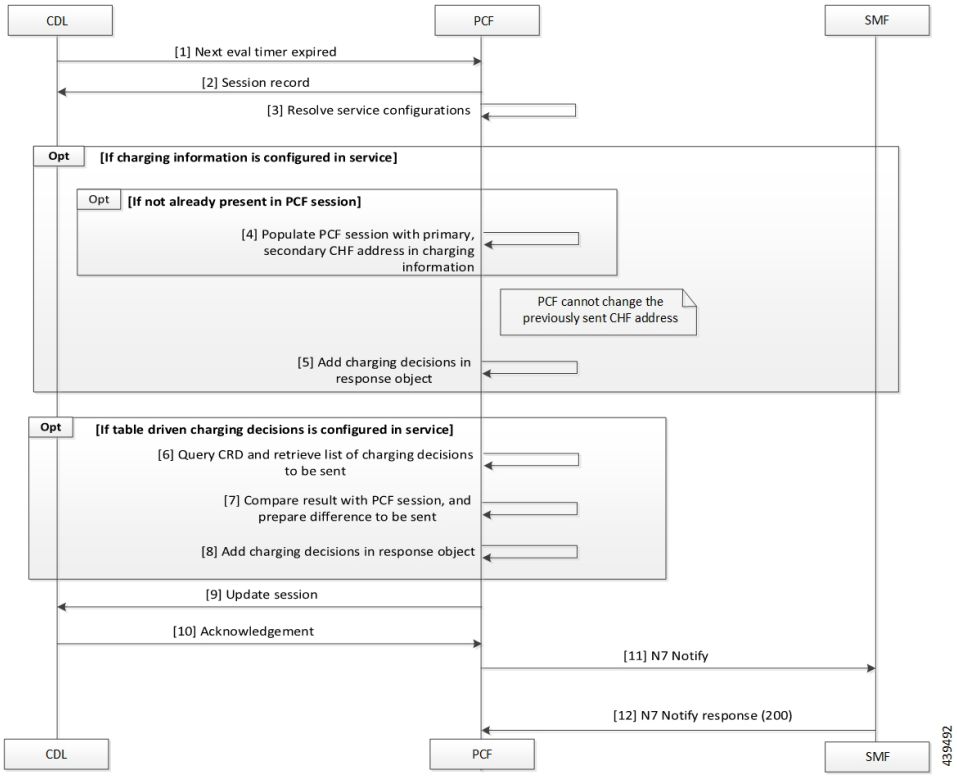


Table 97: Update Notify SM Policy Call Flow Description

Step	Description
1	The CDL sends a next evaluation timer request to the PCF.
2	The PCF sends the Session Record in response to the CDL.
3	The PCF resolves the service configuration.
4	If the charging information is not available in the PCF session, PCF populates the PCF session with primary and secondary CHF address in the charging information.
5	If the charging information is configured in the service, PCF adds the charging information in the response object.
6	If the table driven charging decision is configured in the service, PCF queries the CRD to retrieve the list pf charging decisions.
7	The PCF compares the results with the PCF session to identify the delta.
8	The PCF adds the charging decisions in the response object.
9	The PCF sends the Update Session request to the CDL.
10	In response, the CDL sends an acknowledgment to the PCF.

Step	Description
11	The PCF sends a N7 Notify request to the SMF.
12	The SMF sends a N7 Notify response to the PCF.

Configuration Support for Online Charging

The configuration of online charging enablement over N7 to SMF involves the following steps:

1. ChargingInformation
2. TableDrivenChargingDecision

ChargingInformation

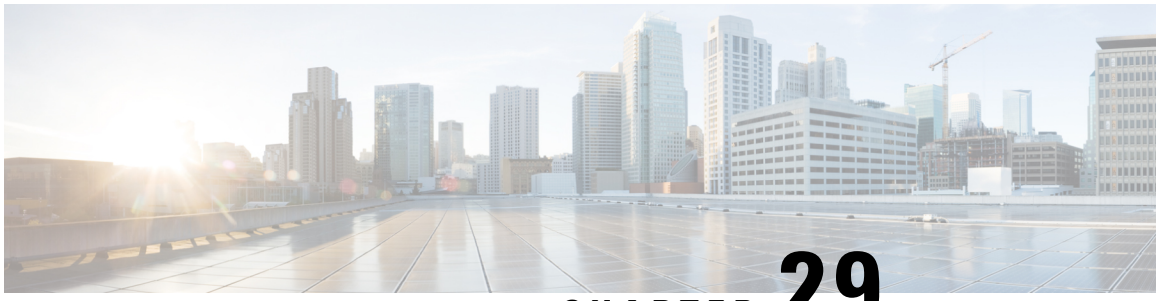
This section describes how to configure the ChargingInformation service.

1. Log in to Policy Builder and navigate to **Services** tab > **Use Case Templates**.
2. Under **Actions > Create Child**, click **Use Case Template**, and add **ChargingInformation** in Service Configuration.
3. Navigate to **Services > Service Option** (for that use case template).
4. Attach the service option to the service.

TableDrivenChargingDecision

This section describes how to configure the TableDrivenChargingDecision service.

1. Log in to Policy Builder and navigate to **Custom Reference Data Table**, and create a search table group for the charging decision table.
2. Navigate to **Services > Use Case Templates**.
3. Under **Actions > Create Child**, click **Use Case Template**, and add **TableDrivenChargingDecision** in Service Configuration.
4. Navigate to **Services > Service Option** (for that use case template).
5. Attach the Service Option to the service.
6. Map the source field to Custom Reference Data (CRD) table created in **Step 1**.



CHAPTER 29

PCF Integration with Access and Mobility Function

Feature Summary and Revision History

Summary Data

Table 98: Summary Data

Applicable Product(s) or Functional Area	AMF
Applicable Platform(s)	CN-CEE
Default Setting	Enabled – Configuration required to disable
Related Documentation	Not Applicable

Revision History

Table 99: Revision History

Revision Details	Release
Enhancement introduced. Added information on how to remove the stale sessions.	2020.05.01
Enhancement introduced. Introduced procedure to configure the N15 Access and Mobility Policies.	2020.02.0

Revision Details	Release
First introduced.	2020.01.0

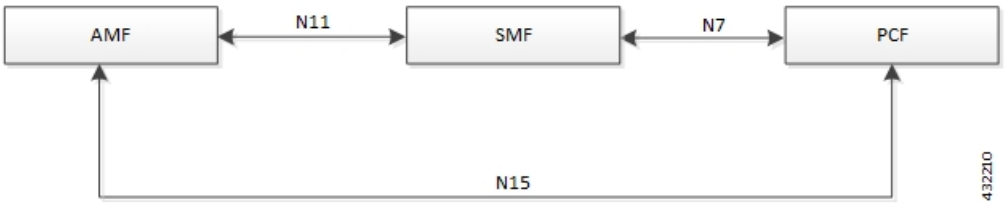
Feature Description

PCF integrates with AMF through the Access and Mobility Policy Control Service by transmitting the access control and mobility management-related policies to the AMF. With this integration, PCF, and AMF interact and exchange information through the following procedures:

- The PCF creates and updates the policies, and deletes the policy association depending on the request that it receives from AMF during the UE registration.
- The PCF notifies the AMF when a policy that AMF has subscribed to is updated. Similarly, AMF is also notified when a policy context is deleted for a UE.
- Depending on the event triggers that PCF has subscribed to, AMF takes the appropriate actions such as update the location procedure when the Service Area Restriction change triggers occur. The Service Area Restriction change is triggered only when a location change happens or the UE is changed in the Presence Reporting Area (PRA).
- During the PCF-AMF communication, if the PCF accumulates session information that is stale which means AMF has a more recent version of the session, or the session in PCF is no longer valid, then PCF purges the stale sessions.

In a reference point representation, a point-to-point reference point defines the interactions between the NFs. The PCF communicates with AMF over N15, and with SMF over N7.

Figure 44: Interfaces in a Non-Roaming 5G System Architecture



Th PCF-AMF framework is compliant with the definitions of *3GPP TS 23.502 [3]*, *3GPP TS 23.503 [4]*, and *3GPP TS 29.507*.

How it Works

This section describes how this feature works.

This section provides a summary of how the PCF and AMF work.

Call Flows

This section describes the key call flows for this feature.

Create Policy Association

This section describes the Create Policy Association call flow.

Figure 45: Create Policy Association Call Flow

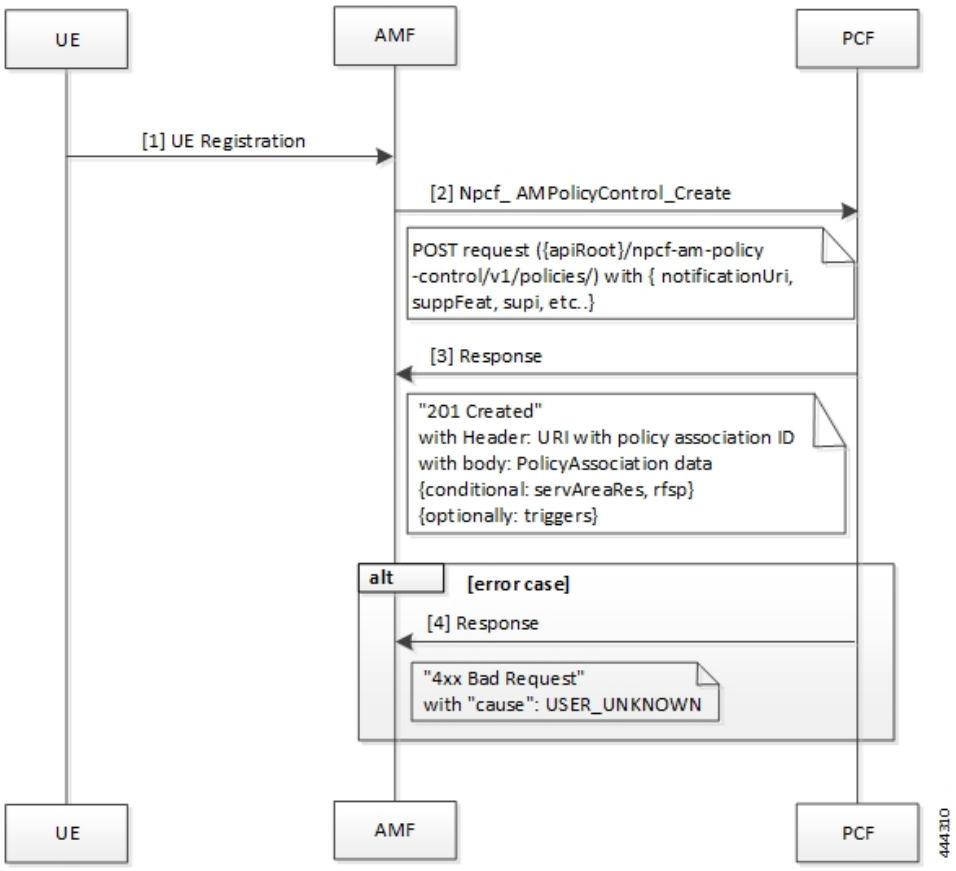


Table 100: Create Policy Association Call Flow Description

Step	Description
1	The User Equipment (UE) sends a UE Registration request to AMF.
2	The AMF forwards the UE Registration request in the form of a Npcf_ AMPolicyControl_Create request to the PCF.
3	If the registration is successful, then PCF responds to AMF with a header and policy ID details.
4	In case of registration failure, PCF responds to AMF with an error indicating that the request was not completed and the issue that caused the failure.

Update Policy Association

This section describes the Update Policy Association call flow.

Figure 46: Update a Policy Association Call Flow

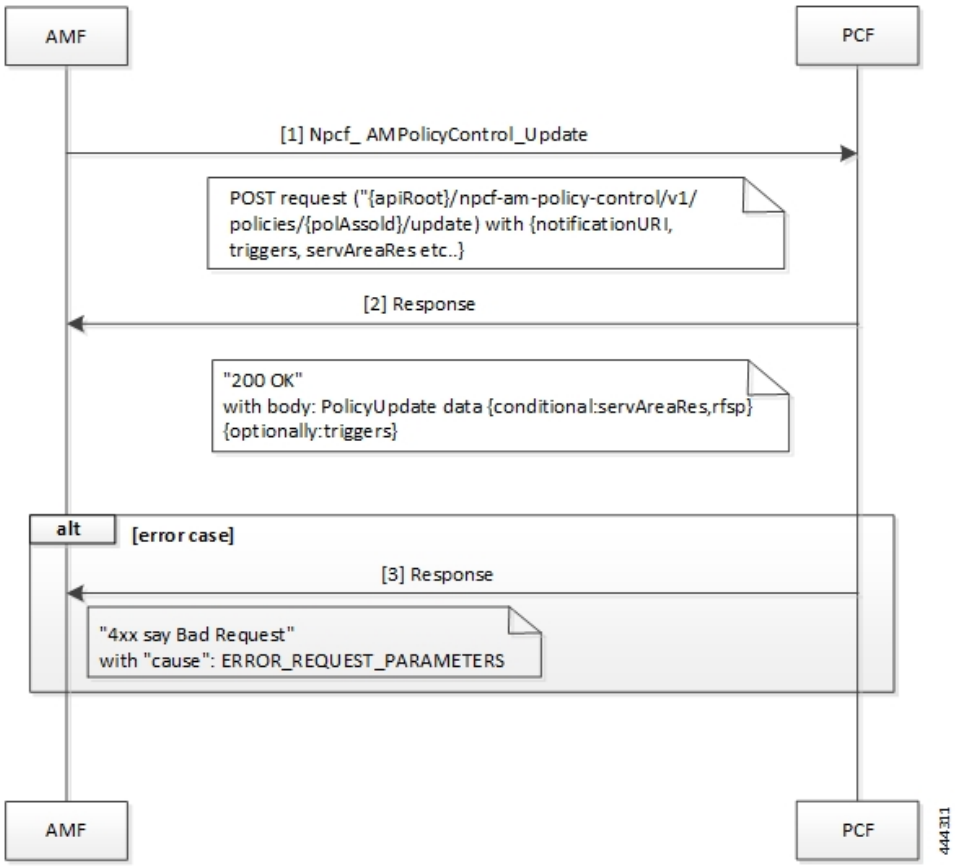


Table 101: Update a Policy Association Call Flow Description

Step	Description
1	When AMF is relocated and the new AMF instance prefers to maintain the policy association, the AMF sends the Npcf_ AMPolicyControl_Update request to PCF.
2	The PCF registers and subscribes to the triggers for the service area restriction changes and responds to AMF with the trigger details.
3	In case of registration failure, PCF responds to AMF with an error indicating that the request is not completed and details of the issue that caused the failure.

Delete Policy Association

This section describes the Delete Policy Association call flow.

Figure 47: Delete Policy Association Call Flow

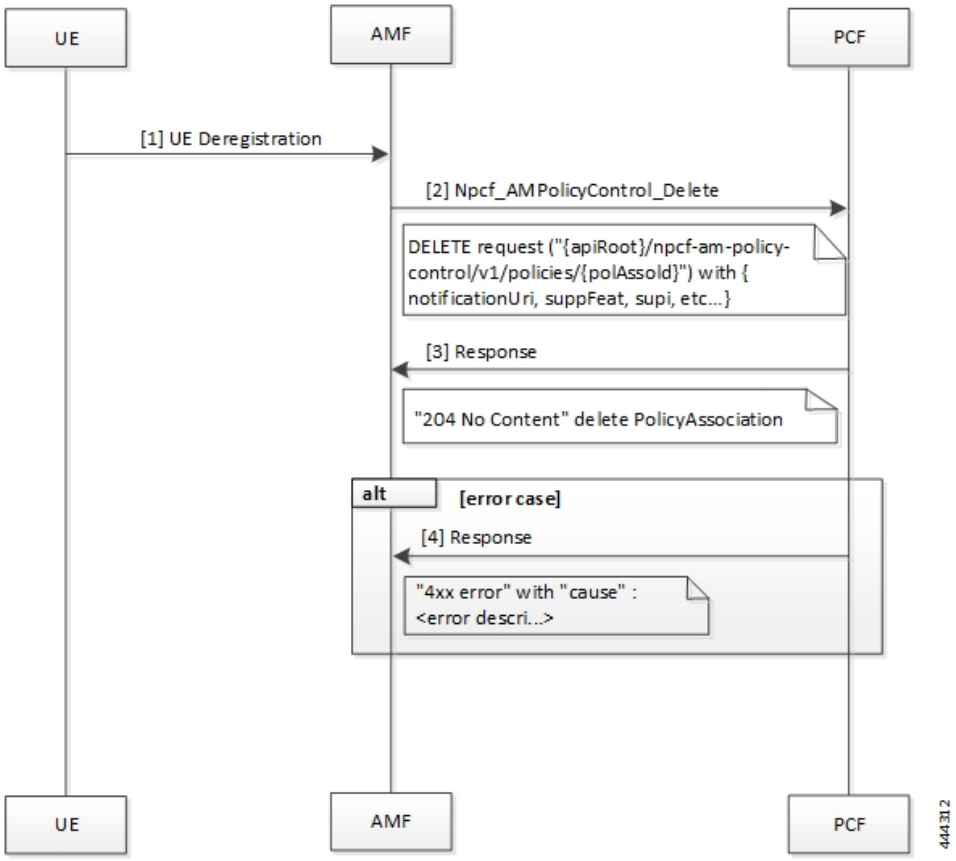


Table 102: Delete Policy Association Call Flow Description

Step	Description
1	In a situation where a policy association must be deleted, the UE sends a Deregistration request to AMF.
2	The AMF sends a Npcf_AMPolicyControl_Delete request to PCF.
3	On successful deletion, PCF sends a response to AMF with the confirmation.
4	In case the deletion was unsuccessful, PCF responds to AMF with an error indicating the deletion failure and the appropriate cause.

Terminate Policy Association

This section describes the Terminate Policy Association call flow.

Figure 48: Terminate Policy Association Call Flow

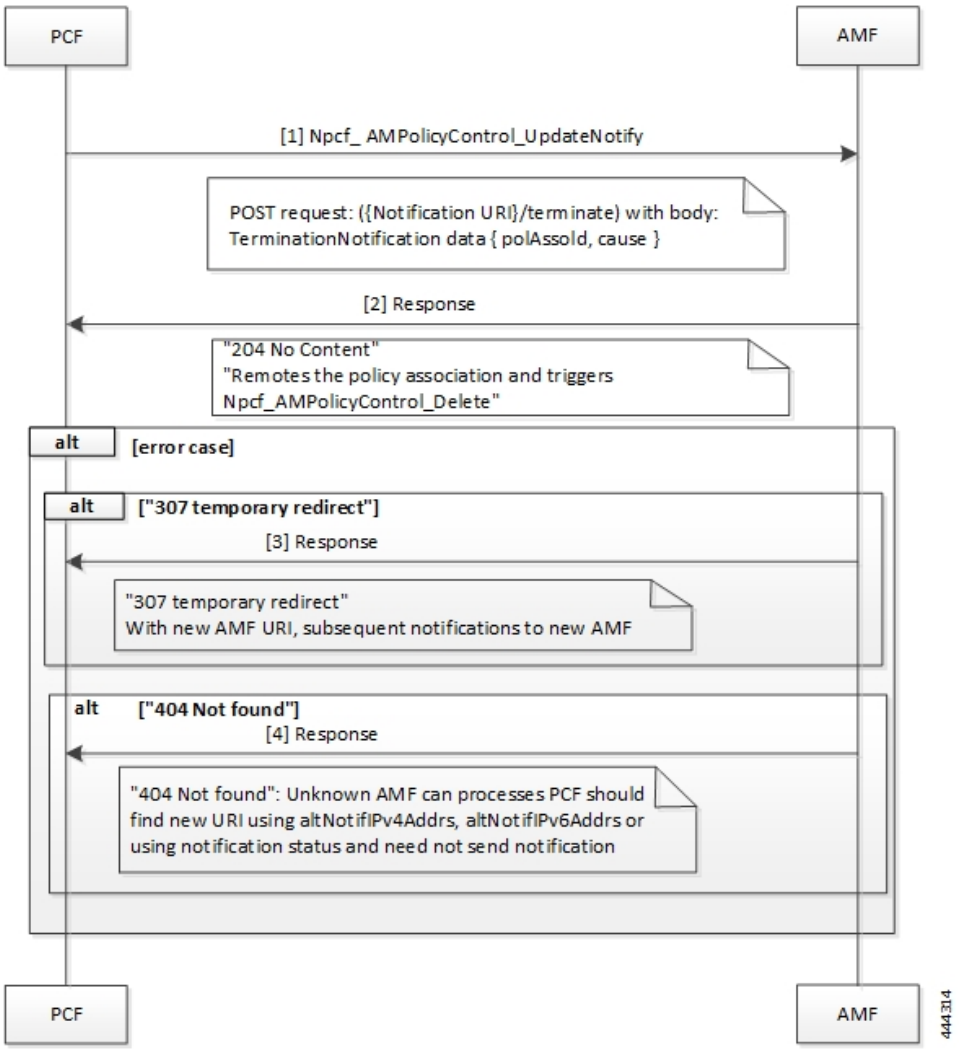


Table 103: Terminate Policy Association Call Flow Description

Step	Description
1	When PCF terminates the policy association, it initiates a terminate notification by sending the Npcf_ AMPolicyControl_UpdateNotify request to AMF.
2	The AMF responds to PCF with the confirmation indicating that Npcf_AMPolicyControl_Delete is initiated. Depending on the termination notification, AMF removes the policy association and initiates delete request.
3	In case the update policy enforcement was unsuccessful, the AMF redirects the subsequent notification to the new AMF.
4	In case of 404 error, AMF responds to PCF stating that it must search for a new URI using the IPv4 or IPv6 address, or refrain from sending notifications to the original AMF.

Update Notification Call Flow

This section describes the Update Notification call flow.

Figure 49: Update Notification Call Flow

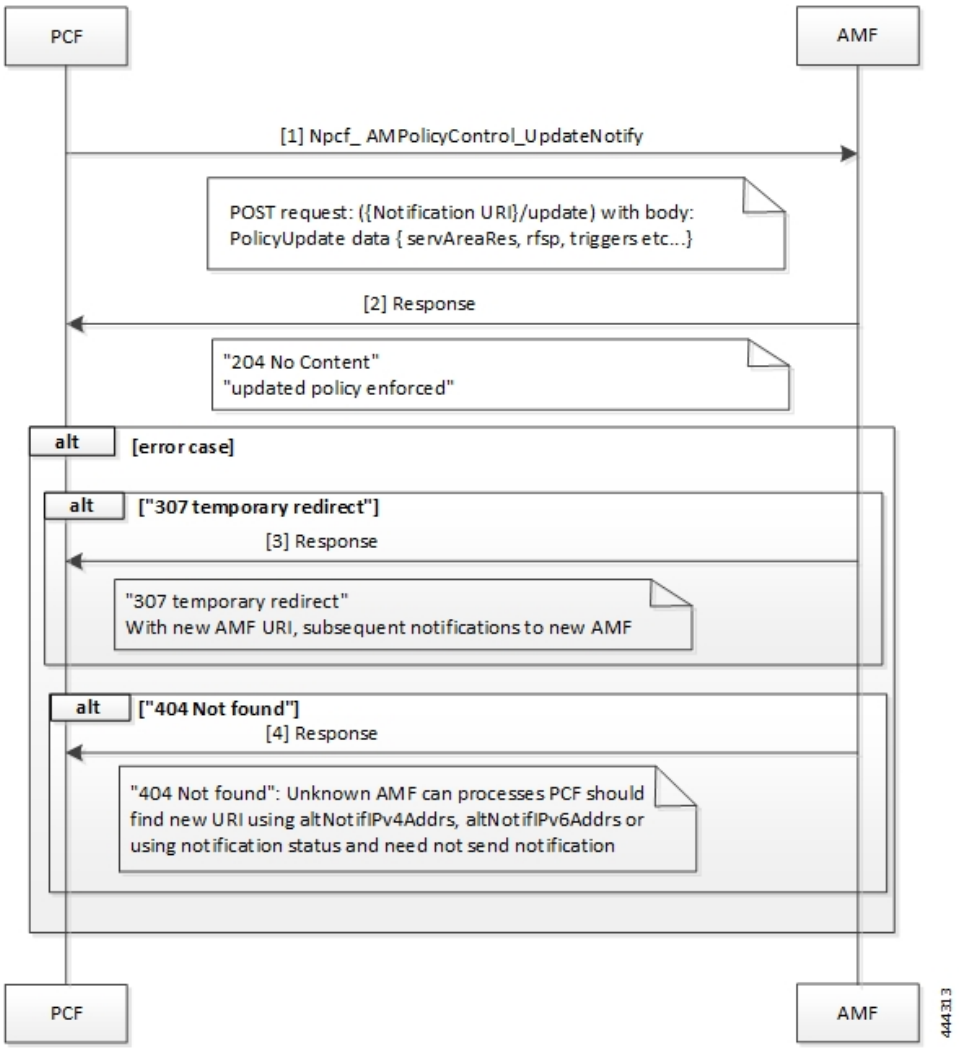


Table 104: Update Notification Call Flow Description

Step	Description
1	When PCF must change the policy, it initiates an update notification by sending the Npcf_ AMPolicyControl_UpdateNotify request to AMF.
2	The AMF responds to PCF with the confirmation indicating that update policy is enforced.
3	In case the update policy enforcement was unsuccessful, the AMF redirects the subsequent notification to the new AMF.

Step	Description
4	In case of 404 error, AMF responds to PCF stating that it must search for a new URI using the IPv4 or IPv6 address, or refrain from sends notifications to the original AMF.

Standards Compliance

This feature complies with the following standards specifications:

- *3GPP TS 29.510 Release 15.2.0 December 2018 "Network Function Repository Services"*
- *3GPP TS 29.571 [11] "Common Data Types for Service Based Interfaces"*

Limitations

This feature has the following limitations in this release:

- The PCF does not support PRA_CH trigger and related use cases.

Configuration Support for the N15 Access and Mobility Policies

This section describes how to configure the N15 access and mobility policies using the following services:

1. Configure the N15 interface using the information documented at Configuring the REST Endpoints, on page 163.
2. Configuring the N15 Policy Service
   - Configuring the N15 Policy Retrievers
   - Configuring the N15 Policy Triggers

Configuring the N15 Policy Service

This section describes the parameters for the N15 policy configuration.

The N15 policy service configuration object is used to configure the Service Area Restriction capability. The configuration involves mapping the N15 policy attributes and the Service Area Restriction CRD table that derives data from the bilateral exchange of requests between AMF and PCF. A one-to-many relation is supported between this service configuration object and the associated CRD table.

Before configuring the N15 policy service, ensure that you have created the use case template and added the required service.

For information on how to create a use case template and add a service for this configuration, see Configuring the Use Case Template, on page 293 and Adding a Service, on page 292.

Table 105: N15 Policy Parameters

Parameters	Description
Priority	Indicates the priority of the message for processing. The higher the number, the higher the priority. Default for most settings: 0
RAT Frequency Selection Priority	Indicates the "rfsp" attributes that PCF receives in the request. The Radio Access Network (RAN) uses this parameter to derive the UE-specific cell reselection priorities to control the idle mode camping, and to decide on redirecting the active mode UEs to different frequency layers or RATs.
UE Policy	The UE policy consists of the UE Access Network discovery and selection policies.
Area Code	The area code is required only when the TAC information is unavailable. This code is operator-specific.
Tac Value	TACs are required only when the area code is unavailable. Indicates a tracking area code that has a hexadecimal representation. Each character in the string shall take a value of "0" to "9" or "A" to "F" and represents 4 bits. The most significant character representing the 4 most significant bits of the TAC shall appear first in the string, and the character representing the 4 least significant bit of the TAC appears last in the string. Examples: A legacy TAC 0x4305 is encoded as "4305". An extended TAC 0x63F84B is encoded as "63F84B"
Restriction Type	Provides the options to configure the type of restriction attribute that you want to configure: • ALLOWED_AREAS: Indicates the area where the restriction can be applied. • NOT_ALLOWED_AREAS: Indicates the area where the restriction cannot be applied. • NO_RESTRICTION: Indicates the areas that do not have any restriction applied.
Max Num Of T As	Denotes the maximum number of allowed tracking areas for use when the restriction is set to "ALLOWED_AREAS". This attribute is unavailable when the Restriction Type takes the value as "NOT_ALLOWED_AREAS". **Note** The Max Num Of T As value cannot be lower than the number of TAIs included in the "tacs" attribute.

Parameters	Description
Max Num Of T As For Not Allowed Areas	Denotes the maximum number of allowed tracking areas for use when Restriction Type is set to "NOT_ALLOWED_AREAS". This attribute is unavailable when the Restriction Type takes the value as "ALLOWED_AREAS".

Configuring the N15 Policy Triggers

This section describes how to configure the N15 policy event triggers.

You can configure the event triggers through the Custom Reference Data (CRD) table. The triggers are a group of conditions used to evaluate a table. PCF subscribes to the configured triggers from the AMF. When the configured triggers are violated, AMF notifies PCF and sends the trigger information.

To configure the N15 policy event triggers, use the following configuration:

1. Log in to Policy Builder.
2. Click the **Reference Data** tab.
3. In the left pane, choose **Custom Reference Data Tables > Custom Reference Data Triggers**.
4. Select the service for which you want to create the trigger.
5. In the right pane, enter the following trigger parameter:

Parameter	Description
Priority	Indicates the priority of the event triggers that must be used in case multiple service initiator conditions match.
Trigger	Specifies the trigger against which the N15 policy object is evaluated. You can configure the following triggers: • LOC_CH: Location change. This trigger is issued when the tracking area of the UE is changed. • RFSP_CH: Change in the RAT Frequency Selection Priority. The UDM notifies the AMF when the subscribed RFSP index is changed. • SERV_AREA_CH: Change in the Service Area Restrictions. The UDM notifies the AMF when the subscribed service area restriction information has modified.

Configuring the N15 Policy Retrievers

This section describes how to configure the retrievers for the N15 policy configuration object.

You can add the retrievers through the CRD table or Service Configuration.

For information on how to add the retrievers through CRD, see Configuring Retrievers through Custom Reference Data Table, on page 326.

For information on how to add the retrievers through Service Configuration pane, see Configuring Retrievers through Service Configuration, on page 327.

You can configure the following parameters under N15 policy retrievers:

- N15 Access Type
- N15 AMF Id
- N15 AreaCode
- N15 Cell Global Identifier
- N15 GPSI
- N15 GroupID
- N15 MaxNumOfTAs
- N15 MaxNumOfTAsForNotAllowedAreas
- N15 MCC (SUPI Based)
- N15 MNC (SUPI Based)
- N15 Permanent Equipment Identifier
- N15 RAT Type
- N15 Restriction Type
- N15 Serving Plmn
- N15 ServiveName
- N15 SliceInformation
- N15 SUPI
- N15 Tracking Area Identifier

Configuring the Stale Session Timer

This section describes how to configure the stale session timer.

Stale session builds up due to events such as network and timeout issues. As a result, PCF starts rejecting new sessions due to capacity or session license limit. The stale session timer configuration lets you set a timer after which PCF revalidates the stale sessions by sending a N7Notify request. If the N7Notify request gets an error response with code 404, then the session is deleted.

To configure the stale session timer for N7 and N15, use the following configuration:

1. Log in into Policy Builder.
2. Select **Reference Data** tab.
3. From the left pane, select **Systems**.
4. Select and expand your system name.

5. Select **PCF Configuration**.
6. In the right pane, configure the following parameters depending on the interface:

Parameter	Description
N7 Stale Session Timer in Minutes	The stale session maps to a session that is not available on the peer. The configured timeout value determines the duration for which a N7 session can remain idle before PCF revalidates it using the N7Notify request. If the response returned for the request contains an error code 404, then the session gets deleted. Default value is 180 minutes. **Note** The stale session timer value should be less than the session expiration time. For information on how to configure the session expiration hours/minutes, see Adding a System, on page 56.
N15 Stale Session Timer in Minutes	The stale session maps to a session that is not available on the peer. The configured timeout value determines the duration for which a N15 session can remain idle before PCF revalidates it using the N7Notify request. If the response returned for the request contains an error code 404, then the session gets deleted. Default value is 180 minutes. **Note** The stale session timer value should be less than the session expiration time. For information on how to configure the session expiration hours/minutes, see Adding a System, on page 56.
Preferred Bit Rate	Defines the value of the bitrate that is sent in the N7 policies. The bitrate is automatically converted as per the configured preferred bitrate.

Removing Stale Sessions

This section describes how to remove stale sessions for an SMF instance.

When the SMF issuing the sessions is unavailable, the sessions become stale after a period of inactivity. These sessions expire based on the duration that you defined in the Stale Session Timer configuration. In the case of a large number of sessions, the system takes longer to delete the session.

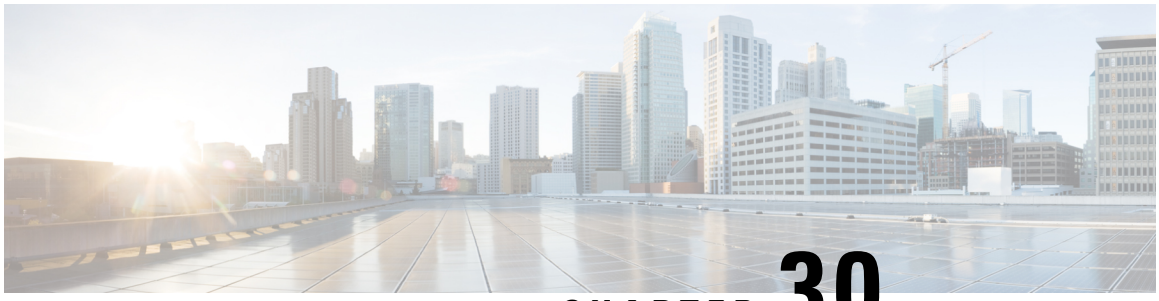
Important We recommend removing the stale sessions only when SMF is unavailable. If SMF is active and has active sessions on PCF, then executing the **cdl clear sessions** command may remove the active sessions.

To delete the sessions in bulk, use the following command:

```
cdl clear sessions filter { key smfInstanceIdKey:SMF_instance_ip_address
condition match }
```

NOTES:

- The **cdl clear sessions** command performs a hard delete of the sessions without generating termination request for the child sessions such as Rx and N28 sessions.
- *SMF_instance_ip_address*—Specify the instance ID of SMF, which is derived from the notification URL sent by the SMF.



CHAPTER 30

Diameter Peer Load Rebalancing

Feature Summary and Revision History

Summary Data

Table 106: Summary Data

Applicable Product(s) or Functional Area	PCF
Applicable Platform(s)	SMI
Feature Default Setting	Disabled – Configuration required to enable
Related Documentation	Not Applicable

Revision History

Table 107: Revision History

Revision Details	Release
First introduced.	2022.02.0

Feature Description

PCF supports diameter peer load rebalancing.

How it Works

This section describes how this feature works.

- CLI is implemented to show the diameter peer connections and its mapping to the individual pods.
- CLI is also used to initiate disconnection of a peer identified by its peer fqdn and realm details. The peer disconnect request is redirected to the respective diameter stack where the connection termination is managed in graceful manner.

Feature Configuration

To configure this feature, use the following configurations:

- View the diameter peer connections per pod.
- Diameter peer disconnection.

View the Diameter Peer Connections Per Pod

This section describes how to view the diameter peer connections per pod.

To view the diameter peer connections per pod, use the following configuration in the Policy Ops Center console:

show diameter peer-status

The output of this command displays the peer hostname, peer realm, pod IP and the status.

The following is a sample output of the **show diameter peer-status** command.

```
pcf# show diameter peer-status
PEER HOSTNAME         PEER REALM              POD IP          STATUS

------------------------------------------------------------------------
site-host-rx1   site-rx-client-cisco.com   192.168.174.137   Connected
site-host-rx2   site-rx-client-cisco.com   192.168.174.137   Connected
```

Diameter Peer Disconnection

This section describes how to disconnect the diameter peer connection.

To disconnect the diameter peer connection, use the following configuration in the Policy Ops Center console:

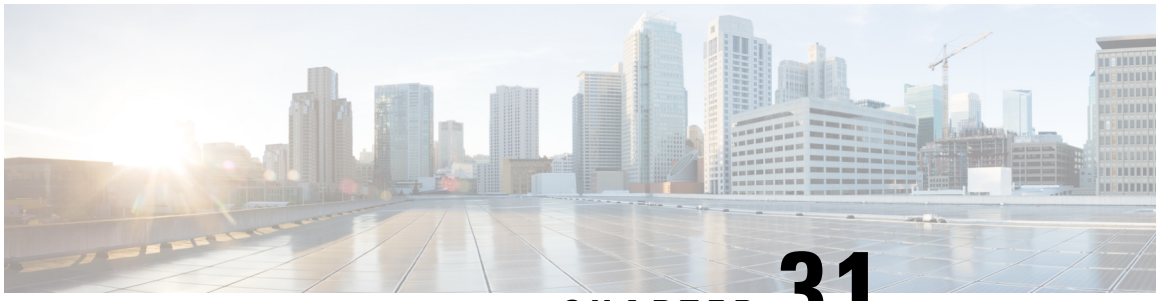
diameter-peer disconnect fqdn <peer-fqdn> realm <peer-realm>

The following is a sample connection success output of the **diameter-peer disconnect fqdn site-host-rx1 realm site-rx-client-cisco.com** command.

```
pcf# diameter-peer disconnect fqdn site-host-rx1 realm site-rx-client-cisco.com
```

The following is a sample connection failure output of the **diameter-peer disconnect fqdn site-host-rx1 realm site-rx-client-tmo.com** command.

```
pcf# diameter-peer disconnect fqdn ecscf-client-s111 realm cscf.mnc010.mcc100.3gppnetwork.org
```



CHAPTER 31

Persistent Storage for Policy Configuration

Feature Summary and Revision History

Summary Data

Table 108: Summary Data

Applicable Products or Functional Area	PCF
Applicable Platform(s)	SMI
Feature Default Setting	Enabled – Configuration required to disable
Related Documentation	Not Applicable

Revision History

Table 109: Revision History

Revision Details	Release
Enhancement introduced. Added the procedure to assign the storage volume as persistent storage.	2020.03.0
First introduced.	2020.02.0

Feature Description

Persistent storage is a storage solution that retains the data after the power and network resources are disconnected.

The PCF provides various storage technologies for managing the configuration data. The PCF has pre-defined storage such as the OpenStack Cinder volume used for storing the CRD data. PCF optionally stores the CRD data in shared storage such as OpenStack Cinder (default) or local storage. In the case of deployment on bare metal servers, PCF uses the local storage class along with the default storage layer as the persistent storage.

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

Restore Capability

The Subversion repository stores the policy-specific configuration data in the XMI format. This repository resides in an SVN pod. If the SVN pod is restarted, the repository experiences a data loss. In such scenarios, you must reimport the configuration files to the SVN pod.

A new restore mechanism is introduced to protect the configuration data and maintain its integrity when the SVN pod restarts.

How it Works

This section describes how this feature works.

The PCF implements the Kubernetes Persistent Volume (PV) framework, which lets the administrators allocate persistent storage for a cluster. Regardless of the storage tier, you can use the Persistent Volume Claims (PVCs) to request PV resources. You must enable persistent volume claim and assign storage that represents local storage. The data residing on the local storage is intact in situations where the associated node or pod restarts.

Restore Capability

The restore capability maintains the continuity of the policy configuration files in conditions where the SVN pod is restarted.

The policy configuration files are in the XMI format. Each SVN repository contains XMI files that are represented in a configMap. The configMap is updated whenever a policy configuration is modified and committed into an SVN repository. When the SVN pod is restarted, it verifies if the configMap is available and the corresponding XMI files are loaded to the repository.

The restore capability is managed through the following configMaps:

- Monitor-svn-configmap-pcf: Contains configuration data in key-value pairs that represent the repository name and policy hash.
- Policy-svn-persistence-configmap: Contains the configured value of the policy-configuration-restore configMap.

Configuring Persistent Storage

This section describes how to configure persistent storage.

Configuring the persistent storage in PCF involves the following steps:

1. Enabling Support for Persistent Storage
2. Assigning Persistent Storage

Enabling Support for Persistent Storage

This section describes how to enable persistent volume claim to configure persistent storage.

1. To enable persistent volume claim, use the following configuration:

```
config
  k8s
  use-volume-claims [ true | false ]
  end
```

NOTES:

- **config**—Enters the configuration terminal.
- **k8s**—Enters the Kubernetes configuration mode.
- **use-volume-claims [true | false]**—Configures using the volume claims during the NF deployment. When set to true, the default storage class such as OpenStack Cinder is enabled. If the **use-volume-claims** is set to false, then the data gets stored in the memory that is susceptible to lose on a pod restart.

Assigning Persistent Storage

This section describes how to assign a storage volume as the persistent storage.

Before configuring the persistent storage, ensure that use-volume-claims is enabled.

1. To assign persistent storage, use the following configuration:

```
config
 db
   global-settings
      volume-storage-class [ default | local ]
      end
```

NOTES:

- **config** – Enters the configuration terminal.
- **db** – Enters the database configuration mode.
- **global-settings** – Configures the database global settings.
- **volume-storage-class [default | local]** – Configures the storage that gets assigned as the persistent storage. Specify **default** to indicate the default storage volume. For example, Cinder. To indicate local-storage volume, specify **local**. If you do not specify any value, the PCF uses the default storage volume.

Configuring the Restore Capability

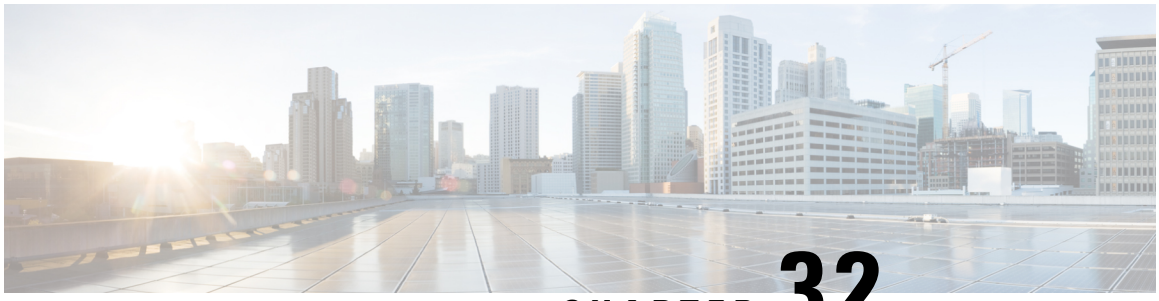
This section describes how to configure the restore capability.

To configure the restore capability that ensures the persistency of policy configuration file, use the following configuration in the Policy Ops Center console:

```
config
  engine engine_name
    pcf policy-configuration-restore [ true | false ]
    end
```

NOTES:

- **engine** *engine_name*—Specify the engine for which the restore capability must be configured
- **pcf policy-configuration-restore [true | false]**—Configures the capability that is responsible for restoring the configMap. The default value for this parameter is true.



CHAPTER 32

Pods and Services

Feature Summary and Revision History

Summary Data

Table 110: Summary Data

Applicable Product(s) or Functional Area	PCF
Applicable Platform(s)	SMI
Feature Default Setting	Enabled – Always-on
Related Documentation	Not Applicable

Revision History

Table 111: Revision History

Revision Details	Release
First introduced.	2020.01.0

Feature Description

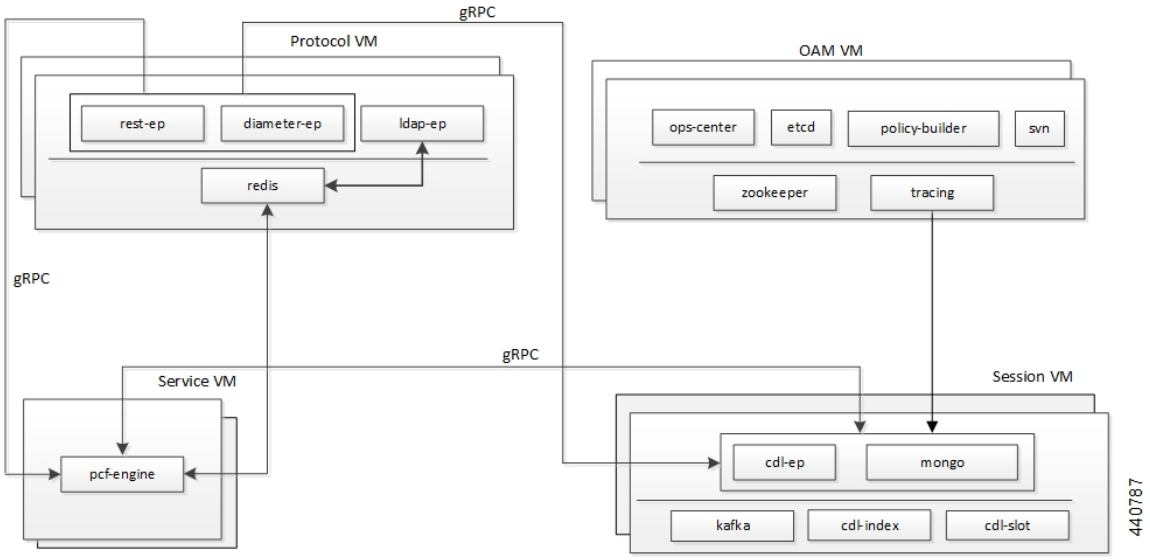
The PCF is built on the Kubernetes cluster strategy, which implies that it has adopted the native concepts of containerization, high availability, scalability, modularity, and ease of deployment. To achieve the benefits offered by Kubernetes, PCF uses the construct that includes the components such as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

Depending on your deployment environment, PCF deploys the pods on the virtual machines that you have configured. Pods operate through the services that are responsible for the intrapod communications. If the

machine hosting the pods fail or experiences network disruption, the pods are terminated or deleted. However, this situation is transient and PCF 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 50: Communication Workflow of Pods



The Protocol VM hosts the rest-ep, diameter-ep, and ldap-ep pod that governs the ingress (incoming) and egress (outgoing) traffic on the interfaces. The pods responsible for the operations and management processes reside in the OAM VM and, the Service VM hosts the pcf-engine. The session VMs hosts the pods that operate as the databases to store the data accessed by the pods. The illustration also depicts the services which the pods use to channel the interactions. The pods communicate over the gRPC interface.

Note Typically, multiple instances of the Protocol and OAM VMs are created to ensure resiliency.

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

For performing the maintenance activities, you can use the **kubectl drain** command to withdraw a node voluntarily. This command prepares the node by evicting or assigning the associated pods to another node with sufficient resources. You can run the **kubectl drain** on individual or multiple nodes concurrently.

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

For more information on the Kubernetes components in PCF, see the following.

- Pods, on page 247
- Services, on page 249

Pods

Pod is a process that runs on your Kubernetes cluster. Pod encapsulates a granular unit known as a container. A pod can contain 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 table lists the 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 252.

Table 112: PCF Pods

Pod Name	Description	Host Name
admin-db	Acts as the MongoDB router pod for the Admin database.	Session
api-pcf-ops-center	Functions as the confD API pod for the PCF Ops Center.	OAM
cdl-ep-session-c1	Provides an interface to the CDL. **Note** Configuration changes to the CDL endpoint cause the endpoint to restart automatically. Cisco recommends making such changes only within the maintenance window.	Session
cdl-index-session	Preserves mapping information of the keys to the session pods.	Session
cdl-slot-session-c1	Operates as the CDL Session pod to store the session data.	Session
cps-license-manager	Acts as the PCF License Manager.	OAM
crd-api-pcf-pcf-engine-app-pcf-<*n*>-mjgxp	Hosts the CRD APIs.	Protocol
db-admin	Acts as the replica set pod for the Admin database.	Session
db-admin-config	Acts as the replica set pod that stores the Admin database configuration.	Session
db-spr-config	Operates as the replica set pod that stores the SPR database configuration.	Session
db-spr1	Functions as the replica set pod that preserves the SPR database.	Session

Pod Name	Description	Host Name
diameter-ep-rx-rx	Contains the Diameter stack details and acts as the endpoint. **Note** Configuration changes to the diameter endpoint cause the endpoint to restart automatically. Cisco recommends making such changes only within the maintenance window.	Protocol
documentation	Contains the documentation.	OAM
etcd-pcf-etcd-cluster	Hosts the etc-d for the PCF application.	OAM
grafana-dashboard-cdl	Contains the Grafana metrics for CDL.	OAM
grafana-dashboard-pcf	Contains the Grafana metrics for PCF.	OAM
kafka	Hosts the Kafka details for the CDL replication.	Protocol
ldap-ep	Operates as an LDAP client to establish communication with an external LDAP server. **Note** Configuration changes to the LDAP endpoint cause the endpoint to restart automatically. Cisco recommends making such changes only within the maintenance window.	Protocol
network-query	Operates as the utility pod to determine the route IP for the Diameter outbound messages.	OAM
ops-center-pcf-ops-center	Acts as the PCF Ops Center.	OAM
patch-server-pcf-cnat-cps-infrastructure	Operates as the utility pod for patching the PCF JAR files.	OAM
pcf-day0-config-pcf-pcf -engine-<*n*>-rchg2	Dedicated for performing the Day-0 configuration for PCF.	OAM
pcf-engine-pcf-pcf-engine-app-pcf	Operates as the PCF Engine. **Note** Configuration changes to the PCF Engine endpoint cause the endpoint to restart automatically. Cisco recommends making such changes only within the maintenance window.	Service

Pod Name	Description	Host Name
pcf-rest-ep	Operates as a REST endpoint for PCF. **Note** Configuration changes to the REST endpoint cause the endpoint to restart automatically. Cisco recommends making such changes only within the maintenance window.	Protocol
policy-builder-pcf-pcf-engine-app	Operates as the Policy Builder for PCF.	OAM
redis-keystore	Operates as the REDIS Index.	Protocol
redis-queue	Operates as the REDIS IPC.	Protocol
rs-controller-admin	Responsible for the replication controller for Admin database.	Session
rs-controller-admin-config	Operates as a replication controller for the Admin database configuration.	Session
rs-controller-spr-config	Operates as a replication controller for SPR database configuration.	Session
rs-controller-spr1	Operates as a replication controller for the SPR database.	Session
smart-agent-pcf-ops-center	Operates as the utility pod for the PCF Ops Center.	OAM
svn	Stores all the PCF XMI configuration files.	OAM
svn-ldap	Stores the LDAP endpoint configuration which is configured through the ops-center.	Protocol
swift-pcf-ops-center	Operates as the utility pod for the PCF Ops Center.	OAM
traceid-pcf-pcf-engine	Stores the subscriber tracing details.	OAM
zookeeper	Assigned for the Zookeeper.	OAM

Services

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

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

Table 113: PCF Services and Pods

Service Name	Pod Name	Description
admin-db	admin-db-0	Serves to process the MongoDB-specific router messages.
cps-diameter-inbound-rx-rx-rx	cps-diameter-ep	Transmits the Rx messages to the Diameter endpoint. You can set an external IP address for the service.
crd-api-pcf-pcf-engine-app-pcf	crd-api	Processes the CRD API calls.
datastore-ep	datastore-ep	Processes the CDL endpoint calls.
datastore-ep-session	ngn-datastore-ep	Responsible for the CDL session.
datastore-notification-ep	pcf-engine	Responsible for sending the notifications from the CDL to the engine.
diameter-engine	pcf-engine	Acts as the Diameter endpoint to pcf-engine.
documentation	documentation	Processes the documentation API calls.
etcd	pcf-etcd-cluster	Processes the etc-d API.
etcd-pcf-etcd-cluster-<n>	pcf-etcd-cluster	Processes the etc-d stateful sets.
grafana-dashboard-cdl	grafana-dashboard-cdl	Responsible for managing the Grafana dashboard for inputs from the CDL.
grafana-dashboard-pcf	grafana-dashboard-pcf	Manages the Grafana dashboard for PCF.
helm-api-pcf-ops-center	helm-api	Manages the Ops Center API.
kafka	kafka	Processes the Kafka messages.
mongo-admin-<n>	db-admin-0	Responsible for the Admin database stateful sets.
mongo-admin-config-<n>	db-admin-config-0	Responsible for the Admin database configuration stateful sets.
mongo-spr-config-<n>	db-spr-config-0	Responsible for the SPR database configuration stateful sets.
mongo-spr1-<n>	db-spr1-0	Responsible for the SPR database stateful sets
ops-center-pcf-ops-center	ops-center	Manages the PCF Ops Center.
patch-server-pcf-cnat-cps-infrastructure	patch-server	Maintains the patch repository.

Service Name	Pod Name	Description
pcf-day0-config-pcf-pcf-engine-app-pcf	pcf-day0-config	Manages the Day-0 configuration.
pcf-engine	pcf-engine	Processes the API calls to pcf-engine.
pcf-rest-ep	pcf-rest-ep	Acts as the http2 request/response to the REST endpoint. You can set an external IP address for the service.
policy-builder-pcf-pcf-engine-app-pcf	policy-builder	Manages the Policy Builder's request/response messages.
redis-keystore-<n>	redis-keystore-0	Manages the REDIS keystore stateful set.
redis-queue-<n>	redis-queue-0	Processes the REDIS queue stateful set.
rs-admin	replica-set admin	Manages the replica set for Admin database.
rs-admin-config	replica-set admin-config	Manages the replica set for the Admin database configuration.
rs-spr-config	replica-set spr-conifg	Manages the replica set for the SPR configuration.
rs-spr1	replica-set sp1	Manages the replica set for the SPR database.
smart-agent-pcf-ops-center	smart-agent-pcf-ops-center	Responsible for the Ops Center API.
svn	cps-subversion	Responsible for the SVN API calls.
swift-pcf-ops-center	swift-pcf-ops-center	Responsible for the Ops Center API.

Ports and Services

PCF uses different ports for communication purposes. The following table describes the default ports and the associated services.

Table 114: Ports and Services

Port	Service	Usage
22	SSH	SMI uses this port to communicate with the virtual machines.
80	HTTP	SMI uses this port for providing Web access to CLI, Documentation, and TAC.
443	SSL/HTTP	SMI uses this port for providing Web access to CLI, Documentation, and TAC.

Port	Service	Usage
2024	SSH	SMI accesses the ConfdD CLI through this port.
3868	TCP	PCF uses this port as the default Diameter Endpoint on a public port.
6443	HTTP	SMI uses this port to communicate with the Kubernetes API server.
8080	HTTP	PCF uses this port to communicate with the Keep Alive API Interface on a public network.
9082	HTTP	PCF uses this port to access the SBI Interface on a public network. The Keepalive monitors the health of the container on this port. If the port is not accessible, then the kubectl restarts the container to restore the service.
9299	HTTP	SMI uses this port to communicate with the Prometheus Service.
9885	TCP	Default port that operates as the PCF Service gRPC endpoint on a private network.
10250	SSL/HTTP	SMI uses this port to communicate with Kubelet.
10256	HTTP	SMI uses this port to interact with the Kube proxy.

Limitations

This feature has the following limitations in this release:

When removing a node using the **kubectl drain** command, the pods managing the inbound traffic such as pcf-rest-ep, pcf-ldapserver-ep, and diameter-ep-rx-protocol cannot be assigned to another node. The workload of these pods' cannot be scheduled to another node since the traffic is routed through persistent connections that do not support load balance. As a result, the Grafana dashboard does not display the Transaction Per Second (TPS) for these pods.

Configuration Support for Pods and Services

This section describes how to associate pods to node and view the pod-related information using the following steps:

1. Associating Pods to the Nodes
2. Viewing the Pod Details and Status

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 get deployed on the nodes that match the key-value pair.

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

```
config
  label
    cdl-layer
      key key_value
      value value
    oam-layer
      key key_value
      value value
    protocol-layer
      key key_value
      value value
    service-layer
      key key_value
      value value
      end
```

NOTES:

- If you opt not to configure the labels, then PCF assumes the labels with the default key-value pair.
- **cdl-layer**—Configures the key-value pair parameters for the CDL.
- **oam-layer**—Configures the key-value pair parameters for the OAM layer.
- **protocol-layer**—Configures the key-value pair parameters for the protocol layer.
- **service-layer**—Configures the key-value pair parameters for the service layer.

Viewing the Pod Details and Status

This section describes how to view the pod details.

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

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

- To view the comprehensive pod details, use the following configuration:

  ```
  kubectl get pods -n pcf 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.

- To view the summary of the pod details, use the following configuration:

```
kp -get pods -o wide
```

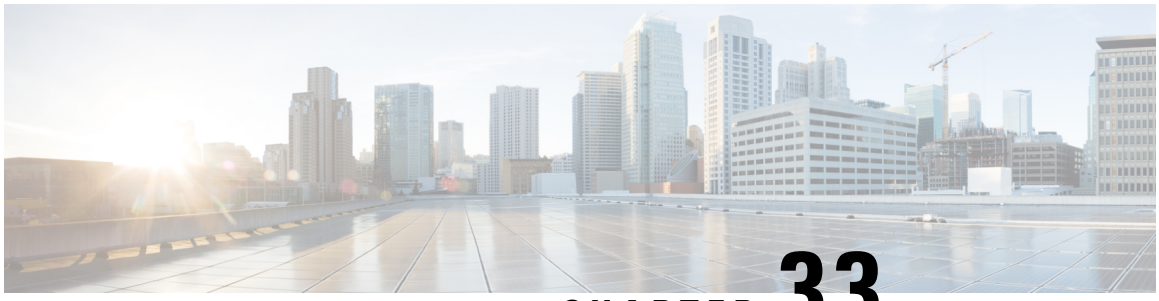
States

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 115: 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.



CHAPTER 33

Policy Tracing and Execution Analyzer

Feature Summary and Revision History

Summary Data

Table 116: Summary Data

Applicable Products or Functional Area	PCF
Applicable Platform(s)	SMI
Feature Default Setting	Enabled – Configuration required to disable
Related Documentation	Not Applicable

Revision History

Table 117: Revision History

Revision Details	Release
First introduced.	2020.01.0

Feature Description

PCF comes with a set of utilities to actively monitor and trace policy execution. These utilities interact with the core Policy Server and the Mongo database to trigger and store traces for specific conditions.

Architecture

Cisco PCF comes with a trace pod to actively monitor and trace the policy execution. The utilities in this pod interact with the Policy Engine pods and the Mongo database pods to trigger and store traces for specific conditions.

The policy tracing and execution analyzer is a three-tier architecture:

- Tier 1—Command-line utilities to manage the policy trace generation and extract policy traces.
- Tier 2—Policy server creation of policy traces using triggers that are defined in Tier 1.
- Tier 3—Storage of the policy traces in a MongoDB.

How it Works

This section describes how this feature works.

Configuration Support for the Policy Traces

This section describes how you configure the policy traces.

Configuration support of the policy traces involves the following steps:

1. Setting Up the Trace Database
2. Configuring the Trace Microservice Pod
3. Executing the Tracing Scripts

Setting Up the Trace Database

This section describes how to configure the database and port where you want to store the traces.

1. Log in to Policy Builder.
2. From left pane, select your system and click the appropriate cluster.
3. From right pane, select the check box for **Trace Database**.

   The following table provides the parameter descriptions under **Trace Database** check box.

Table 118: Trace Database Parameters

Parameter	Description
Primary Database IP Address	The name of the Mongo database cluster that holds the trace information which allows debugging of specific sessions and subscribers based on the unique primary keys.

Parameter	Description
Secondary Database IP Address	The IP address of the database that provides fail over support for the primary database. This is the mirror of the database that is specified in the Primary IP Address field. Use this only for replication or replica pairs architecture. This field is present but deprecated to maintain the downward compatibility.
Database Port	Port number of the database that stores the trace data. Default value is 27017.

Configuring the Trace Microservice Pod

PCF hosts the tracing-specific commands on the trace microservice pod that is available under the /usr/local/bin directory.

To determine the trace pod, use the following configuration:

```
config
  kubectl -n pcf namespace [ get pods | grep trace ]
  end
```

Sample output of the command:

```
luser@for-cn-dev-10c-masterb92844ec32:~$ kubectl -n pcf get pods | grep trace
traceid-pcf-pcf-engine-app-pcf-75b6dc6c4-hc7qc      1/1     Running   0          40m
luser@for-cn-dev-10c-masterb92844ec32:~$
```

Executing the Tracing Scripts

Tracing logs assist you in backtracking the steps that you or the system has performed to accomplish a task. This information is useful when you want to conduct forensics of the unexpected outcomes.

PCF provides two scripts that let you obtain the tracing information:

- trace_ids.sh: Manages the rules for activating and deactivating traces within the system.
- trace.sh: Allows retrieval of the real-time and historical traces.

The execution of the tracing scripts involves the following steps:

1. Managing the Trace Rules
2. Managing the Trace Results

Managing the Trace Rules

The **trace_ids.sh** script fetches the real-time and historical traces. This script resides in `/usr/local/bin/` of the Tracing Pod that you have configured.

See Configuring the Trace Microservice Pod, on page 257 for procedure to set up a Pod.

The Execute the **trace_ids.sh** script with *-h* arguments produces a help text describing the capabilities of the script.

The **trace_ids.sh** script starts a selective trace and outputs it to a standard out.

1. To specify the audit ID tracing, use the following configuration:

   ```
   kubectl -n pcf exec -it traceid-pcf-pcf-engine-app-pcf-75b6dc6c4-hc7qc
   -- trace_ids.sh -i specific id
   ```

2. To remove trace for specific audit ID, use the following configuration:

   ```
   kubectl -n pcf exec -it traceid-pcf-pcf-engine-app-pcf-75b6dc6c4-hc7qc

   -- trace_ids.sh -r specific id
   ```

3. To remove trace for all IDs, use the following configuration:

   ```
   kubectl -n pcf exec -it traceid-pcf-pcf-engine-app-pcf-75b6dc6c4-hc7qc
    -- trace_ids.sh -x
   ```

4. To list all the IDs under trace, use the following configuration:

   ```
   kubectl -n pcf exec -it traceid-pcf-pcf-engine-app-pcf-75b6dc6c4-hc7qc
    -- trace_ids.sh -l
   ```

   Adding a specific audit ID for tracing requires running the command with the -i argument and passing in a specific ID. The Policy Server matches the incoming session with the ID provided and compares this against the following network session attributes:

   - Credential ID
   - Framed IPv6 Prefix
   - IMSI
   - MAC Address
   - MSISDN
   - User ID

   If an exact match is found, then the transactions are traced.

   Note Spaces and special characters are not supported in the audit IDs.

   - Removing a specific audit ID from active tracing requires specifying the -r argument with ID to remove.
   - Removing all IDs requires sending in the -x argument. This step purges all the IDs from the database.
   - Listing all IDs requires sending in the -l argument.

   Example output:

   ```
   kubectl -n pcf exec -it traceid-pcf-pcf-engine-app-pcf-75b6dc6c4-hc7qc
    -- trace_ids.sh
   ```

-s mongo-admin-0 -p 27017 -t admin -d policy_trace -i 2001

Run the **trace_ids.sh** with *-h* arguments produces a help text describing the capabilities of the script as follows:

```
kubectl -n pcf exec -it traceid-pcf-pcf-engine-app-pcf-75b6dc6c4-hc7qc -- trace_ids.sh
-h
/usr/local/bin/trace_ids.sh: option requires an argument -- h
usage:
/usr/local/bin/trace_ids.sh -i specific id
    /usr/local/bin/trace_ids.sh -r specific id
    /usr/local/bin/trace_ids.sh -x
    /usr/local/bin/trace_ids.sh -l
    /usr/local/bin/trace_ids.sh -s mongo service name
    /usr/local/bin/trace_ids.sh -p mongo service port
    /usr/local/bin/trace_ids.sh -t mongo replica set
    /usr/local/bin/trace_ids.sh -d mongo database name

This script starts a selectve trace and outputs it to standard out.
1. Add Specific Audit Id Tracing  /usr/local/bin/trace_ids.sh -i specific id
2. Remove Trace for Specific Audit Id  /usr/local/bin/trace_ids.sh -r specific id
3. Remove Trace for All Ids /usr/local/bin/trace_ids.sh -x
4. List All Ids under Trace /usr/local/bin/trace_ids.sh -l
5. K8 mongo service name -s (default: mongo-admin-0)
6. Mongo port -p (default: 27017)
7. Replica set name -t (default: admin)
8. Trace database name -d (default: policy_trace)
9. /usr/local/bin/trace_ids.sh -h displays this help
```

Managing the Trace Results

The **trace.sh** script that initiates selective trace process resides in `/usr/local/bin/` of the Tracing Pod that you have configured.

See Configuring the Trace Microservice Pod, on page 257 for procedure to set up a pod.

1. To specify the audit ID tracing, use the following configuration:

   kubectl -n pcf exec -it traceid-pcf-pcf-engine-app-pcf-75b6dc6c4-hc7qc -- trace.sh -i *specific_id*

   Specifying the *-i* argument for a specific ID causes a real-time policy trace to be generated while the script is running. You can redirect this to a specific output file using standard Linux commands.

2. To dump all traces for the specific audit ID, use the following configuration:

   kubectl -n pcf exec -it traceid-pcf-pcf-engine-app-pcf-75b6dc6c4-hc7qc -- trace.sh -x *specific_id*

   Specifying the *-x* argument with a specific ID, dumps all historical traces for a given ID. You can redirect this to a specific output file using standard Linux commands.

3. To trace all, use the following configuration:

   kubectl -n pcf exec -it traceid-pcf-pcf-engine-app-pcf-75b6dc6c4-hc7qc – trace.sh -a

   Specifying the *-a* argument causes all traces to output in the real-time policy trace while the script is running. You can redirect this to a specific output file using standard Linux commands.

4. To trace all the errors, use the following configuration:

```
kubectl -n pcf exec -it traceid-pcf-pcf-engine-app-pcf-75b6dc6c4-hc7qc
 -- trace.sh -e
```

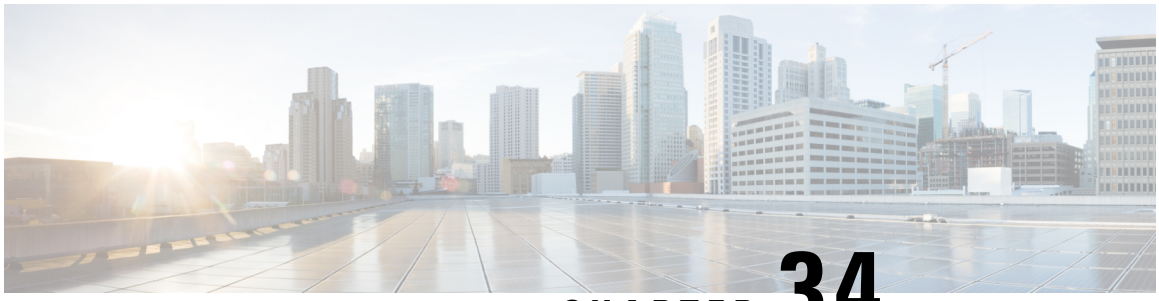
Specifying the *-e* argument causes all traces that are triggered by an error to output in real-time policy trace while the script is running. You can redirect this to a specific output file using standard Linux commands.

Example output:

```
kubectl -n pcf exec -it traceid-pcf-pcf-engine-app-pcf-75b6dc6c4-hc7qc
 -- trace.sh -s mongo-admin-0
-p 27017 -t admin -d policy_trace -x 1234567890
```

5. Execute the **trace.sh** script with *-h* arguments to produce a help text describing the capabilities of the script as follows:

```
kubectl -n pcf exec -it traceid-pcf-pcf-engine-app-pcf-75b6dc6c4-hc7qc -- trace.sh -h
/usr/local/bin/trace.sh: option requires an argument -- h usage:
    /usr/local/bin/trace.sh -i specific_id
    /usr/local/bin/trace.sh -x specific_id
    /usr/local/bin/trace.sh -a
    /usr/local/bin/trace.sh -e
    /usr/local/bin/trace.sh -s mongo_service_name
    /usr/local/bin/trace.sh -p mongo_service_port
    /usr/local/bin/trace.sh -t mongo_replica_set
    /usr/local/bin/trace.sh -d mongo_database_name
    /usr/local/bin/trace.sh -h
This script starts a selectve trace and outputs it to standard out.
1. Specific Audit Id Tracing  /usr/local/bin/trace.sh -i specific_id
2. Dump All Traces for Specific Audit Id  /usr/local/bin/trace.sh -x specific_id
3. Trace All /usr/local/bin/trace.sh -a
4. Trace All Errors /usr/local/bin/trace.sh -e
5. K8 mongo service name -s (default: mongo-admin-0)
6. Mongo port -p (default: 27017)
7. Replica set name -t (default: admin)
8. Trace database name -d (default: policy_trace)
9. /usr/local/bin/trace.sh -h displays this help
```



CHAPTER 34

Policy Control Request Triggers Over N7

Feature Summary and Revision History

Summary Data

Table 119: Summary Data

Applicable Products or Functional Area	PCF
Applicable Platform(s)	SMI
Feature Default Setting	Enabled – Configuration required to disable
Related Documentation	Not Applicable

Revision History

Table 120: Revision History

Revision Details	Release
First introduced.	2020.01.0

Feature Description

PCF can subscribe to the policy control request triggers on the SMF. This is done by including triggers in the response to smPolicyControl_Create request. PCF can modify the request triggers that are subscribed in the SMF using Npcf_SMPolicyControl_UpdateNotify request or in response to smPolicyControl_Update service operation.

Configuring the Policy Control Request Trigger Events over N7

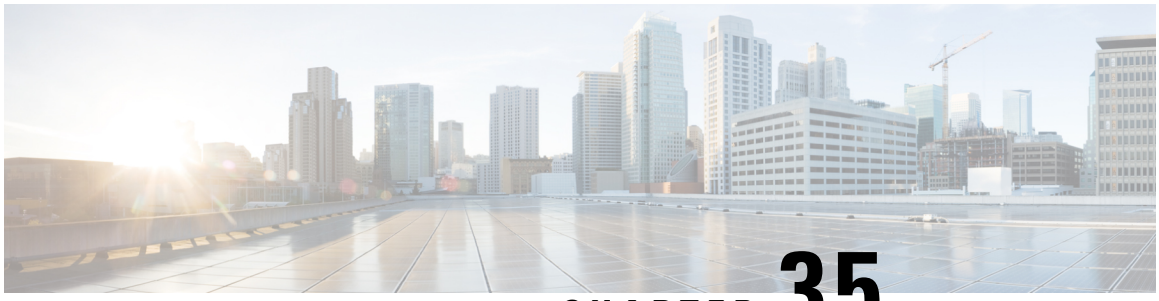
This section describes how to configure the Policy Control Request Trigger (EventTrigger) service to configure different events that are subscribed in response to smPolicyControl_Create request.

Before configuring the trigger service, ensure that you have created the use case templates and added the required service.

For information on how to create a use case template and add a service for this configuration, see Configuring the Use Case Template, on page 293 and Adding a Service, on page 292.

1. Under **Service Configurations**, click **Add** to open the **Select Service Configuration** window.
2. Choose **PCF > EventTrigger** and configure the required parameters.

   PCF can subscribe to the policy control request triggers in SMF. For the list of supported policy control triggers, see *3GPP TS 29.512 N7* specification.



CHAPTER 35

Predefined Rules and Rulebase

Feature Summaryand Revision History

Summary Data

Table 121: Summary Data

Applicable Products or Functional Area	PCF
Applicable Platform(s)	SMI
Feature Default Setting	Enabled – Always-on
Related Documentation	Not Applicable

Revision History

Table 122: Revision History

Revision Details	Release
First introduced.	2020.01.0

Feature Description

During session establishment and modification, the SMF communicates with PCF over the N7 interface. When a user equipment (UE) establishes a packet data unit (PDU) session, the UE requires policies for session management. PCF stores the policies as Policy and Charging Control (PCC) rule IDs in the Policy Builder application. When SMF receives the session establishment request, SMF requests PCF to provide policies, PCF then sends the PCC rule ID. SMF uses the PCC rules to configure the UPF for various data flow tasks, such as shaping, policing to provide bandwidth, and charging functions.

PCF supports configuration of PCC rule ID for Rule and Rulebase. Rulebase is the collection of charging rule names. PCF sends this PCC rule ID to the SMF, and as per the PCC rule ID, the SMF searches the definition of these rules.

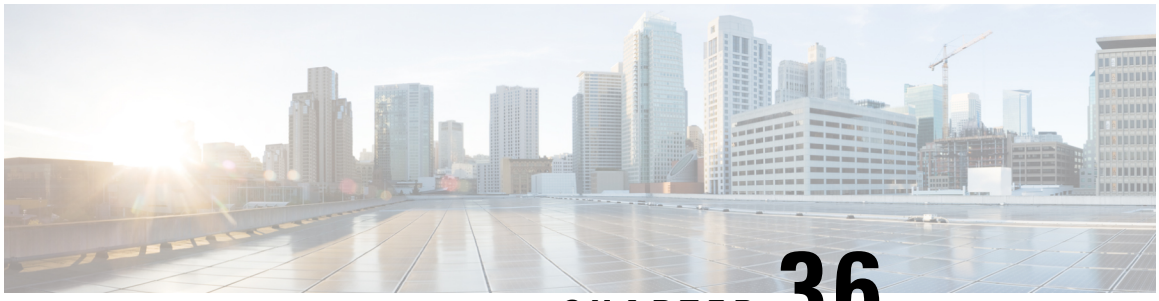
Note Rule and Rulebase Name is equivalent to Charging Rule Name and Charging Rulebase Name that were present in PCRF.

Configuration Support for Rule and Rulebase

This section describes how to configure a PCC rule ID for Rule and Rulebase rules.

1. Log in to the Policy Builder application.
2. In the **Service Configurations** pane, click **DynamicPccRule**.

   The dynamic PCC rule parameters appear.
3. Click the **PCC Rule Id** parameter and enter the **Pcc_Rule_Id** value.
4. Click **Save**.



CHAPTER 36

Rx Authorization

Feature Summary and Revision History

Summary Data

Table 123: Summary Data

Applicable Product(s) or Functional Area	PCF
Applicable Platform(s)	SMI
Feature Default Setting	Disabled – Configuration required to enable
Related Documentation	Not Applicable

Revision History

Table 124: Revision History

Revision Details	Release
First introduced.	2020.01.0

Feature Description

PCF provides a method for the service providers to regulate the services available to individual subscribers. You can configure the bearer-level regulation through the customization and configuration of Rx Authorization.

The configuration handles the Video over LTE (ViLTE) authorization as per the subscriber attributes (IMSI, MSISDN, and Throttling) to control the services available to each subscriber.

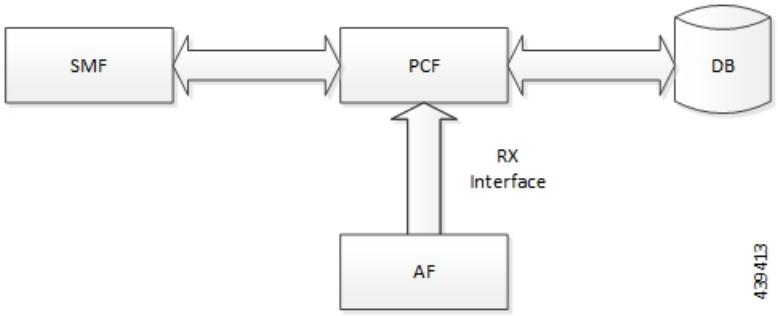
Architecture

This section depicts how the network function components interact during an Rx Authorization.

The SMF and PCF have a bilateral communication over the N7 interface. The AF sends an AAR request to PCF. The PCF performs the Rx Authorization of the request by evaluating the message for the missing Media-AVP and consults the value that is assigned to the Bearer-Authorization column in the STG table for the configured status as accept or reject. PCF fetches the STG information from the associated database. PCF communicates the evaluation result to the SMF through a REST request and AF through the outgoing Diameter messages.

The following figure illustrates how the NF interactions happen over the Rx interface.

Figure 51: NF Interactions



Components

This section describes the RxAuthorizationSTGConfiguration component in the Rx Authorization process.

The RxAuthorizationSTGConfiguration service configuration is used to evaluate the Rx Authorization table and obtain the configured output values. The RxAuthorizationSTGConfiguration service supports chained evaluation of Search Table Groups (STGs) which means multiple STGs are configured hierarchically in the service and outputs of one table is used as input keys for another table. The RxAuthorizationSTGConfiguration configuration evaluates all the bearers on receiving a Diameter message and sends the appropriate Diameter requests or responses depending on the bearer's authorization status provided the Rx session exists. The Rx Authorization table from which Bearer Authorization and Error Cause output values are received is configured as the last table in the list of chained STGs configured under RxAuthorizationSTGConfiguration.

How it Works

This section describes how this feature works.

At a high-level, PCF supports the Rx-based authorization of bearers. The Rx authorization requires a Search Table Groups (STG), which enables logical grouping of multiple Customer Reference Data (CRD) tables. Within this STG, a CRD table that is dedicated to Rx Authorization is created in the Policy Builder. The input keys in the CRD signify the conditions based on which PCF determines the throttle limit for a bearer. The table has the following output columns:

- Bearer Authorization: Indicates whether to allow or reject a bearer.
- Error Cause: Specifies the Error-Message that is included in the AAA Diameter message, if necessary.

If PCF is configured to reject the Rx dedicated bearer when the associated Media-Type is missing, it rejects the bearer with the Experimental-Result-Code=INVALID_SERVICE_INFORMATION (5061) in AAA.

PCF is configured to reject a non-GBR bearer if the value for both, upload and download of the non-GBR bearer is set to 0. PCF determines if the bearer is non-GBR with 0-bit rate after consulting the NON-GBR QCI and ZERO BIT RATE QoS input columns in the Rx Authorization table. If PCF rejects the bearer, then its Bearer-Authorization value is set to REJECT with Result-Code=DIAMETER_AUTHORIZATION_REJECTED (5003) AVP and the Error-Message="BLOCKED (0)" in AAA.

If PCF receives an AAR message with multiple Media-Component-Descriptions AVPs, and it rejects one of the AVPs after assessing for Rx Authorization. PCF sends a successful AAA message for the accepted AVPs. For the rejected media component, PCF creates a scheduled event for sending a delayed Rx RAR. You can configure the duration between the rejection and the time when scheduling of the delayed message happens. The default value is set to 500 milliseconds.

Note In case PCF rejects multiple Media-Component-Descriptions AVPs with AAA 5003, the Error-Message resulting from the last evaluated rejected AVP is sent in the AAA message.

For existing bearers in an Rx session, PCF evaluates them for Rx Authorization when an event occurs such as LDAP refresh, N28 NOTIFY, and N7_NOTIFY. In situations where all the Media-Component-Descriptions that are stored in the Rx sessions are rejected, then PCF sends an Rx Abort-Session-Request (ASR) to Application Function (AF).

Note You may observe a degradation in the performance of the PCF system when the RxAuthorizationSTGConfiguration service is added. The level of degradation corresponds to the number of STGs configured for the chained evaluation in the RxAuthorizationSTGConfiguration service and the number of bearers the service has evaluated.

Call Flows

This section describes the key call flows for this feature.

All Bearers Are Rejected Call Flow

This section describes the All Bearers Are Rejected call flow.

Figure 52: All Bearers Are Rejected Call Flow

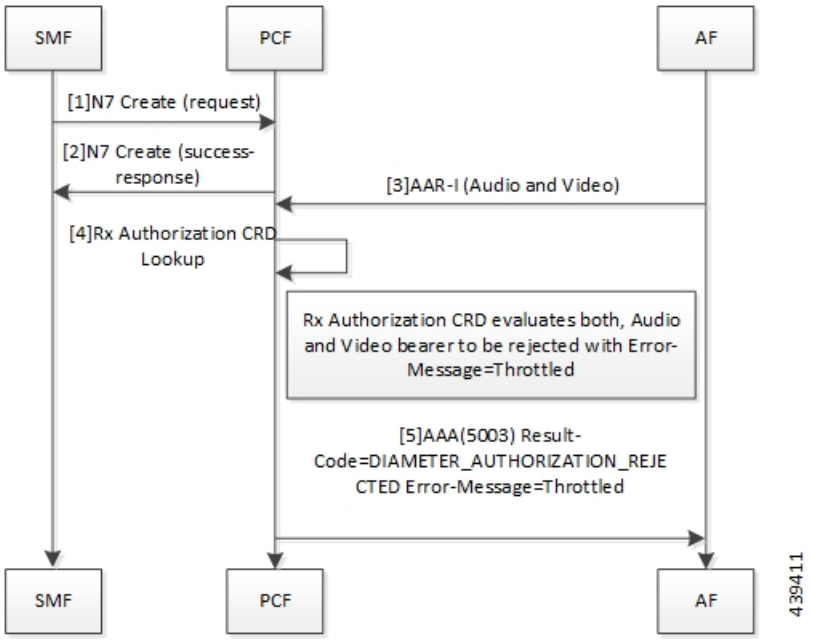


Table 125: All Bearers Are Rejected Call Flow Description

Step	Description
1	The SMF sends a N7 Create request to the PCF.
2	The PCF responds to the SMF with the success response.
3	The AF sends an AAR-I (Audio and Video) message to the PCF.
4	The PCF performs the Rx Authorization CRD lookup.
5	The Rx Authorization CRD evaluates both, audio and video bearer. If there is a missing Media-Type AVP, PCF rejects the bearer. PCF validates all the bearers for Bearer-Authorization=REJECT. The bearers are classified as unauthorized and are not installed on the SMF. If all bearers received in the AAR are rejected, PCF sends a AAA (5003) Result-Code=DIAMETER_AUTHORIZATION_REJECTED Error-Message=Throttled to the AF.

Few Bearers Are Rejected Call Flow

This section describes the Few Bearers are Rejected call flow.

Figure 53: Few Bearers Are Rejected Call Flow

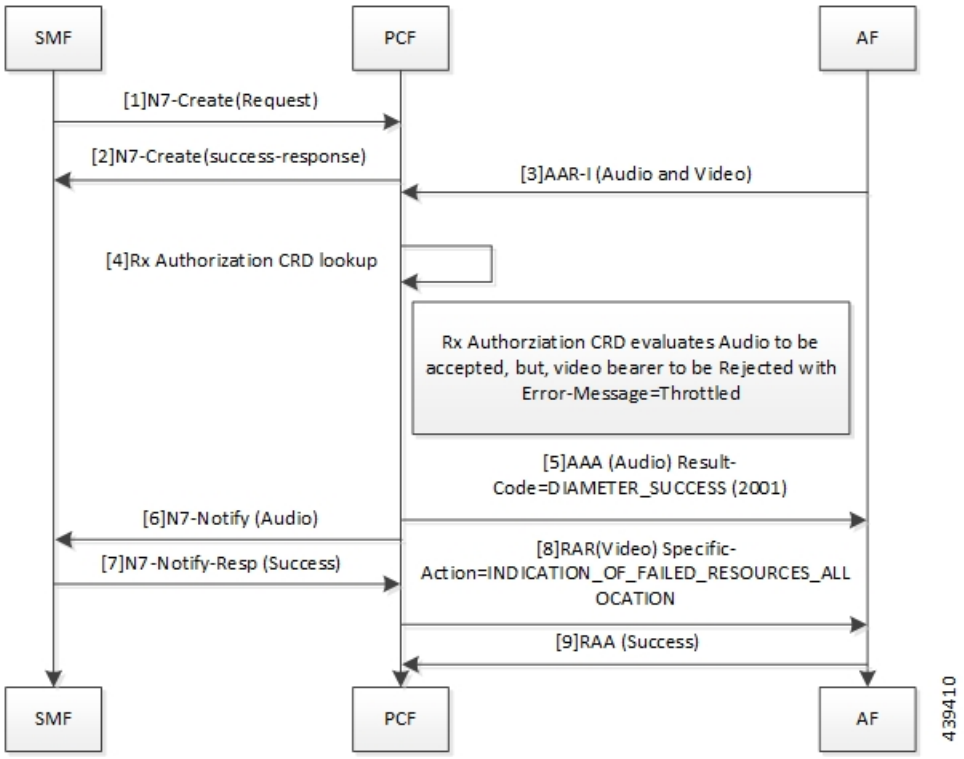


Table 126: Few Bearers Are Rejected Call Flow Description

Step	Description
1	The SMF sends a N7 Create request to the PCF.
2	The PCF responds to SMF with success response.
3	The AF sends an AAR-I (Audio and Video) message to the PCF.
4	The PCF performs the Rx Authorization CRD lookup.
5	The Rx Authorization CRD evaluates both the audio and video bearers. The audio bearers that contain the required Media-Type AVP are tagged as accepted. Video bearers with the missing Media-Type AVP are rejected. Bearers evaluated to Bearer-Authorization=ACCEPT are authorized and installed on the SMF. PCF responds to the accepted audio bearers with AAA (Audio) Result-Code=DIAMETER_SUCCESS (2001).
6	The PCF sends N7 Notify (Audio) to the SMF.
7	The SMF responds to the PCF with a N7 Notify-Resp (Success).

Step	Description
8	Bearers evaluated to Bearer-Authorization=REJECT are marked as unauthorized and are not installed at the SMF. The PCF sends RAR (Video) Specific-Action=INDICATION_OF_FAILED_RESOURCES_ALLOCATION message to AF.
9	The AF sends RAA (Success) message to PCF.

Existing Bearers Are Rejected Call Flow

This section describes the Existing Bearers Are Rejected call flow.

Figure 54: Existing Bearers Are Rejected Call Flow

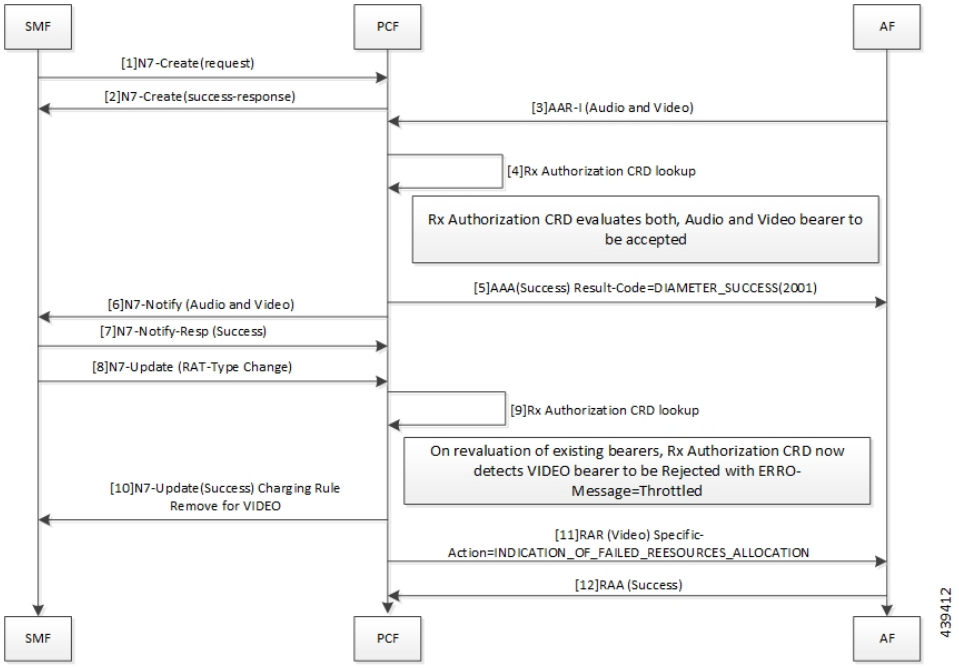


Table 127: All Bearers Are Rejected Call Flow Description

Step	Description
1	The SMF sends a N7 Create request to the PCF.
2	The PCF responds to the SMF with a N7 Create Success response.
3	The AF sends an AAR-I (Audio and Video) message to the PCF.
4	The PCF performs the Rx Authorization CRD lookup.
5	The Rx Authorization CRD evaluates both, the audio and video bearers. If successful authorization, PCF sends AAA (Success) Result-Code=DIAMETER_SUCCESS(2001) to AF.

Step	Description
6	The PCF sends N7 Notify (Audio and Video) message.
7	The SMF responds with N7-Notify-Resp (Success) to the PCF.
8	The SMF sends N7 Update (RAT-Type Change).
9	The PCF performs the Rx Authorization CRD lookup.
10	When PCF reevaluates the existing bearer and the Rx Authorization CRD detects a VIDEO bearer with the missing AVP, PCF rejects the bearer with Error-Message=Throttled. The PCF sends N7-UPDATE (Success) Charging Rule Remove for VIDEO to the SMF.
11	The PCF sends RAR (VIDEO) Specific-Action=INDICATION_OF_FAILED_RESOURCES_ALLOCATION to the AF.
12	The AF responds with RAA (Success) to the PCF.

Considerations

The following considerations apply when you configure the Rx Authorization:

- The STG names that are configured in the RxAuthorizationSTGConfiguration should be unique.
- The AVP names for the output columns that are configured in the RxAuthorizationSTGConfiguration service should be unique.
- The chained evaluation keys should have the same AVP name for the output column in the source table, and the input column in the destination table.
- The result of the RxAuthorizationSTGConfiguration service is available in the last table that is defined in the list. The table includes the output columns with the following mandatory AVP names: Bearer-Authorization and Error-Message.
- The Bearer-Authorization column can be configured to accept the fixed values that are Accept and Reject.
- Perform the configurations that are required for defining and mapping the CRD tables as per the requirement.
- The Policy Server evaluates the mapped source output AVPs (result column of the STG) through the CRD which it has created. If PCF has not created the CRD, then it cannot query the corresponding chained input key which further limits it from verifying the Rx Authorization.
- 1:1 mapping must exist between a chained pair of output AVP and the input key.

Limitations

This feature has the following limitations in this release:

- When an Rx Authorization fails, PCF sends an Rx_RAR request only if the Specific-Action= INDICATION_OF_FAILED_RESOURCES_ALLOCATION is armed in the AAR message.

- The Rx authorization is performed only at the Media-Component-Description AVP level. This indicates that the AVPs from the AAR message that are used as input for the CRD table evaluation should be from Media-Component-Description AVP only. PCF does not evaluate of the Media-Sub-Component AVP.
- If using the PolicyState or Session data retrievers that are bound to the input keys, then PCF retrieves the data for the input keys if it is inserted into the session data.

Configuration Support for Rx Authorization

This section describes how to configure Rx Authorization.

The configuration of the Rx Authorization capability in PCF involves the following steps:

1. Creating the STG Tables
2. Adding the RxAuthorizationSTGConfiguration Service
3. Configuring the Service Chaining
4. Rejecting the AAR with the Missing Media-Type AVP
5. Setting Up the Delayed Message Schedule

Creating the STG Tables

This section describes how to create the STG column in Policy Builder.

To configure the STG column, use the following configuration:

1. Log in to Policy Builder.
2. Click the **Reference Data** tab, and from the left pane click **Custom Reference Data Tables** to view the options.
3. On the left pane, click the **Search Table Groups** folder.
4. In the **Search Table Group Summary** pane, click **Search Table Group**. A default STG gets created under the **Search Table Groups** folder.
5. Click the new STG and in the **Search Table Groups** pane rename the STG with a unique name.
6. Click **Customer Reference Data Table**. A new table gets created on the left pane.
7. Click the new table to open the **Customer Reference Data Table** pane. Rename the table with a unique name.
8. Navigate to the Columns section and click **Add**. A default column gets added to the Columns section.
9. Click the newly created column heading and rename it. Select the options in the corresponding row as applicable to your environment.

Note If the **Key** option is selected for a specific column, then it indicates as the input column.

10. Save the changes.

Adding the RxAuthorizationSTGConfiguration Service

This section describes how to add the RxAuthorizationSTGConfiguration service.

To configure the RxAuthorizationSTGConfiguration service, use the following configuration:

1. Log in to Policy Builder.
2. Choose the **Services** tab, and from the left pane click **Use Case Templates** to create a new service.
3. On the left pane, click **Summary** to open the **Summary** pane.
4. Under **Actions**, click **Use Case Template**.
5. In the **Use Case Template** pane, specify the name for the template.
6. Click the **Actions** tab and select **Add**.
7. In the **Select Service Configuration** dialog box, select the RxAuthorizationSTGConfiguration and click **OK**. The Use Case template with the specified name is created.
8. In the left pane, click **Services > Service Options** to view the options. The newly created service appears in the **Service Options**.
9. Select the service that you have created.
10. Under **Service Configurations**, click **Add** to open the **Select Service Configuration** dialog box.
11. Under **Service Configurations**, select **RxAuthorizationSTGConfiguration**, then click **OK**.

Configuring the Service Chaining

This section describes how to configure the service chaining for Rx Authorization.

Before configuring the service chaining, ensure that you have created the use case templates and added the RxAuthorizationSTGConfiguration service. Use case templates are the building blocks of the PCF architecture. The use case templates allow you to define the Service Configuration objects to be set by a Service Option.

To configure service chaining, use the following configuration:

1. Log in to Policy Builder.
2. Click the **Services** tab, and from the left pane click **Service Options** to view the options.
3. Expand the new service that you have created, and select the child.
4. In the **Service Option** pane, select **Rx_AuthorizationSTGConfiguration** service under **Service Configurations** and specify the Rx_AuthorizationSTGConfiguration parameters.
5. Expand the **List Of Input Column Avp Pairs (List) > ColumnAndAvpPair**, and enter the appropriate information.
6. Expand the **List Of Output Column Avp Pairs (List) > ColumnAndAvpPair**, and enter the Avp Name as Bearer-Authorization. Similarly, in another **ColumnAndAvpPair > Avp Name** field specify Error-Message.

7. Save the changes.

Rejecting the AAR with the Missing Media-Type AVP

This section describes how to enable PCF to reject the AAR messages with missing Media-Type AVPs.

To configure PCF to reject the AAR messages, use the following configuration:

1. Log in to Policy Builder.
2. Click the **Reference Data** tab.
3. In the left pane, click **Diameter Clients** > **Rx Clients**.
4. Click **Rx-Client**.
5. In the **Rx Client** pane, select the **Reject AAR with Invalid Service Info for missing Media-Type** check box.
6. Save the changes.

Setting Up the Delayed Message Schedule

This section describes how to set up the duration after which PCF sends the delayed message to the AF.

To configure the delayed message schedule through the Policy Builder, use the following configuration:

1. Log in to Policy Builder.
2. Click the **Reference Data** tab.
3. In the left pane, click **Diameter Clients** > **Rx Clients**.
4. Click **Rx-Client**.
5. In the **Rx Client** pane, specify the duration in the **Sending Delayed Message Wait Time (In millisec)** field. If you do not specify the period, then PCF considers the default period of 500 milliseconds.

Rx Client

This section describes the parameters, which you can configure for the Rx client.

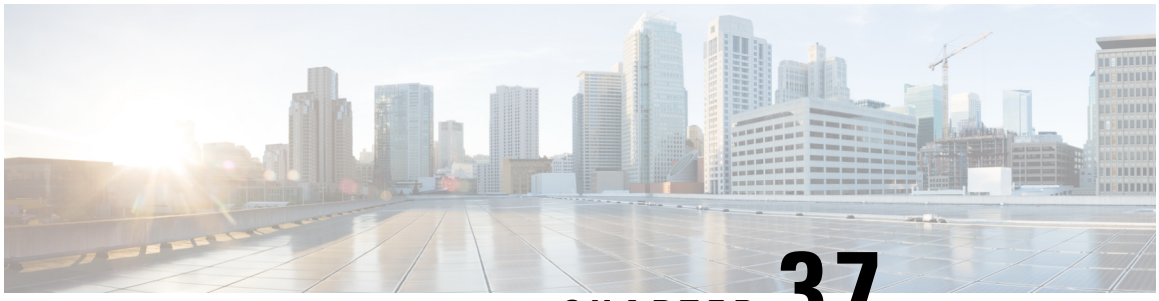
Use the Rx Client, which is a Diameter client object along with the Rx interface. You can add the Rx-specific features to the generic Diameter client.

Before setting the service parameters, ensure that you create a use case template and add a service for this configuration. For details, see Configuring the Use Case Template, on page 293 and Adding a Service, on page 292.

The following table describes the Rx Client service parameters:

Table 128: Rx Client Parameters

Parameter	Description
Reject AAR with Invalid Service Info for missing Media-Type	Enables PCF to reject the Rx_AAR message when Media-Component-Descriptions AVPs have the Media-Type AVP missing. PCF rejects the message with Experimental-Result-Code= INVALID_SERVICE_INFORMATION (5061). To enable the parameter, select the check box available in the **Diameter Clients > Rx Client**.
Delayed Message Wait Time	Allows you to specify the duration after which PCF sends a delayed message. The default value is 500 milliseconds. To define the duration, specify the period in the text field available in **Diameter Clients > Rx Client**.



CHAPTER 37

Rx Interface for 4G and 5G

Feature Summary and Revision History

Summary Data

Table 129: Summary Data

Applicable Product(s) or Functional Area	PCF
Applicable Platform(s)	SMI
Feature Default Setting	Enabled – Configuration required to disable
Related Documentation	Not Applicable

Revision History

Table 130: Revision History

Revision Details	Release
Behavior change introduced. Prior to the 2019.03 release, for the “3GPP Diameter Rx support on 5G PCF” feature, the “device-protocol-id configuration” option under Diameter application was required to be configured with the value 30. This option was configured using CLI for the Rx Diameter incoming calls for tagging them to RX_5G_TGPP device protocol. For the “Converged Rx Support for 5G/4G” feature, the session lookup is done at the PCRF + PCF engine to determine whether the incoming Rx request needs to be tagged to RX_TGPP device protocol or RX_5G_TGPP device protocol. Hence, for this feature the “device-protocol-id configuration” option under Diameter application is removed and is no longer configurable in the CLI. **Note** The update for this feature is not backward compatible and requires a fresh install.	2020.01.0
First introduced.	Pre 2020.01.0

Feature Description

The combined PCF-PCRF deployment architecture provides both PCRF and PCF capabilities. In this deployment, all the incoming Rx Diameter requests are sent to the PCRF or PCF Engine, where session lookup determines the session binding of the 4G and 5G sessions.

Relationships

This feature is an extension of the "3GPP Diameter Rx support on 5G PCF" feature, which was the implementation of N5 interface on PCF.

How it Works

This section describes how this feature works.

The incoming Rx requests are tagged to RX_5G_TGPP device protocol, if the respective N7 session is available. Else, the requests are tagged to RX_TGPP device protocol.

No CLI configuration is required for tagging these requests.

Routing the Rx Diameter Requests

This section describes the service configurations that enable routing of the Rx Diameter requests to PCF or PCRF.

- RxSTGConfiguration

Configuring RxSTGConfiguration AVP

This section describes the parameters that can be configured for RxSTGConfiguration.

The RxSTGConfiguration service configuration supports the following output AVPs that allow the dynamic value expression and their ranges to be defined.

Before setting the service parameters, ensure that you create a use case template and add a service for this configuration. For details, see Configuring the Use Case Template, on page 293 and Adding a Service, on page 292.

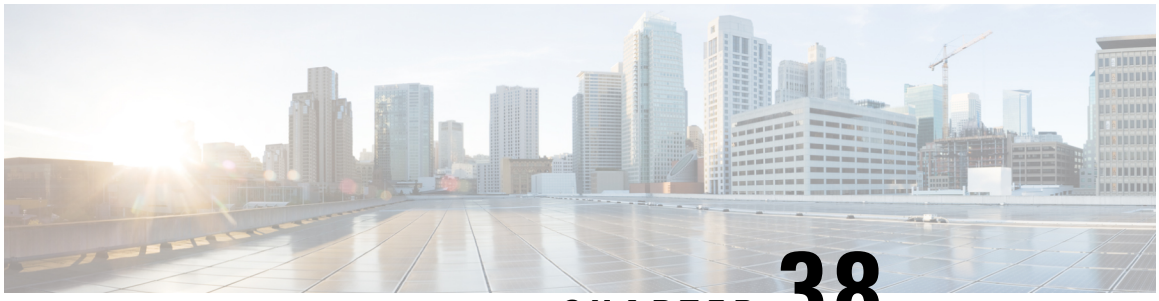
The following table describes the RxSTGConfiguration service parameters.

Table 131: RxSTGConfiguration Parameters

Parameters	Description
Dynamic-QoS-ARP-Priority-Level	**Note** This is a mandatory parameter if the Dynamic QoS ARP feature is enabled. This AVP is bound to the dynamic expression Priority-Level column. If the value is null/not configured, then Dynamic QoS ARP feature is disabled. If the value is configured, it overrides the integer PL value (if configured). The dynamic PL expression is either expected to match the java regex: `^[dD](\\s*([+-/*])\\s*([0-9]+))?$` or must be an offset value (of syntax: [+-][0-9]+). In case the value is provided in offset form, the “D” is implicit. Thus “+8” corresponds to “D+8” in expression form, “-5” corresponds to “D-5” and similarly, “0” corresponds to “D”.
Dynamic-QoS-ARP-Priority-Level-Default	If the default bearer doesn’t have a Priority-Level, this value is used as dedicated bearer PL. If the value is null/not configured, the default value (15) is used.
Dynamic-QoS-ARP-Priority-Level-Min	This output AVP provides upper/lower bound for the calculated PL value using the Dynamic expression provided under Dynamic-QoS-ARP-Priority-Level. If the value is null/not configured, the default value (1) is used.
Dynamic-QoS-ARP-Priority-Level-Max	The upper end of the valid PL range. If the value is null/not configured, the default value (15) is used.
Dynamic-QoS-Update-On-Change	This AVP controls whether the Rx rules must be updated on change in the dynamic PL value (for example, due to change in default bearer PL value). If value is null/not configured, the Rx rules aren’t updated with new dynamic PL value once installed.

Note

- Using the offset form may have minor performance gains as compared to a full expression.
- Range limits are not applied for the default dynamic values.
- Dynamic expression has an implicit “Enforce” QoS action. The Action column value is ignored.
- If dynamic expression configured for Priority-Level is invalid, PCF ignores the expression and does not include the ARP parameters (since PL is set as null) in the rule install. This is true even if absolute PL value is configured (absolute value is ignored).



CHAPTER 38

Site Isolation

Feature Summary and Revision History

Summary Data

Table 132: Summary Data

Applicable Product(s) or Functional Area	PCF
Applicable Platform(s)	SMI
Feature Default Setting	Enabled – Configuration required to disable
Related Documentation	Not Applicable

Revision History

Table 133: Revision History

Revision Details	Release
Enhancement introduced. Introduced instructions to configure the remote system ID in the secondary site while the primary site is undergoing a site isolation procedure.	2021.04.0
First introduced.	2020.02.0

Feature Description

Site isolation is segmenting your PCF environment to create silos of cluster or a standalone CDL instance in a Geographic Redundancy (GR) deployment. Each silo is self-sufficient with access to dedicated resources and network utilities. With this approach, you can upgrade or resolve network issues targeted towards the affected site without impacting any other site.

The site isolation strategy protects against data loss by replicating changes between the primary site and the secondary site. The secondary site takes over the primary site's traffic workload whenever the primary site is unavailable. After the maintenance activity is completed, you can bring up the primary site and reinstate it to the previous state to process the requests.

How it Works

This section describes how this feature works.

A site can be unavailable when it is undergoing maintenance level upgrade or experiencing a network issue. During this period, the site cannot manage the traffic that the client directs towards it. In such situations, you can isolate the site so that the traffic workload is switched from a primary site to a secondary site.

Configuring the PCF site isolation feature is a simplified process that involves issuing the commands from the PCF Ops Center console of the primary and secondary sites. The primary-secondary-primary switch includes the following:

1. In the PCF Ops Center of the primary site, set the PCF registration status to UNDISCOVEREABLE. If the primary site is unavailable, the client automatically contacts the secondary site. Similarly, when the primary site comes online, the client attempts to connect to the primary site. No manual intervention is required to bring up the secondary site.

   The primary and secondary sites are always synchronized, so the data integrity is maintained.

   To determine whether all the traffic requests are switched successfully to the secondary site, review the traffic status on the Grafana dashboard. Also, verify that the primary site has not received any SBA inbound traffic.

2. After the traffic is switched to a secondary site, you can bring down the primary site and take the required actions to upgrade or resolve the accessibility issues.

   Note If you intend to isolate the site without disrupting the GR replication system, do not shut down the primary site.

3. In the primary site, ensure that only the Ops Center-specific pods are running in the PCF product namespace. The rest of the pods must be terminated.

4. After the planned activities are completed on the primary site, and it is ready to be brought back to a consistent state, bring up the primary site.

5. Ensure that the sessions on the primary site are synchronized with the recent updates on the secondary site. You can verify the CDL changes and compare the CLD local session count on both the sites

Prerequisites

This section describes the prerequisites that must be met to configure the site isolation feature.

Before bringing down a site, ensure that all the in-progress traffic requests are completed.

Configuring the Site Isolation Feature

You can configure the site isolation feature from the PCF Ops Center.

Configuring the site isolation feature involves the following steps:

1. Configuring the PCF Registration Status
2. Bringing Down the Primary Site
3. Determining the Pod Status
4. Bringing Up the Primary Site
5. Verifying if the Sessions are Synchronized
6. Verifying if the Primary Site is Up

Configuring the PCF Registration Status

This section describes how to configure PCF as undiscoverable.

To configure the PCF registration status to undiscoverable, use the following configuration from the PCF Ops Center of the primary site:

```
config
  service-registration
   profile
    nf-status { REGISTERED | UNDISCOVERABLE }
    commit
    exit
```

NOTES:

- **config**—Enters the configuration mode.
- **service-registration**—Enters the service registration configuration mode.
- **profile**—Enters the profile configuration mode.
- **nf-status { REGISTERED | UNDISCOVERABLE }**—Enters the profile configuration mode.

Bringing Down the Primary Site

This section describes how to configure to bring the primary site down and the remote site for generating notification when primary site is isolated.

Note If you want to isolate the site without disrupting the GR replication system, do not bring down the primary site.

1. Configure the primary site to bring the primary site down on the PCF Ops Center:

   The secondary site takes over the primary site's traffic when the primary site is down or in the UNDISCOVERABLE state.

   ```
   config
     system mode shutdown
     commit
     end
   ```

   NOTES:

   - **config**—Enter the configuration mode.
   - **system mode shutdown**—Shut down the site.

2. Configure the remote system ID on the PCF Ops Center:

   After primary site is unavailable, configure the remote-system-id in the secondary site using the siteID of the primary site.

   ```
   config
     cdl
       datastore session
         slot notification remote-system-id [ siteID ]
       exit
        exit
   ```

   NOTES:

   - **config**—Enter the configuration mode.
   - **cdl**—Enter the CDL configuration mode.
   - **datastore session**—Enter the datastore session configuration.
   - **slot notification remote-system-id [siteID]**—Specify the siteID for the primary site. The SiteID is associated with the cdl remote-site system-id configuration in the YANG model.

   Sample Configuration

   The following is a sample configuration for specifying the siteID.

   ```
   cdl datastore session
   slot notification remote-system-id [ 1 ]. <- 1 is the siteID of site1
   exit
   ```

For more information on CDL components, see *Cisco Common Data Layer* documentation.

Determining the Pod Status

This section describes how to verify that only the PCF Ops Center-specific pod is running on the secondary site.

To verify if the Ops Center-specific pod is running in the PCF product namespace, use the following:

Use the following command in the CEE Ops Center of the secondary site:

```
show cluster pods | tab | nomore | include ops-center
```

Alternatively, on the master node, use the following command to display the pod status associated with a specific namespace.

```
kubectl get pods -n pcf_namespace
```

Bringing Up the Primary Site

This section describes how to bring up the primary site.

1. Configure the secondary site to remove siteID on the PCF Ops Center:

   Before bringing up the primary site, remove the primary site's siteID from the secondary site's "remote-system-id" list.

   ```
   no cdl datastore session slot notification remote-system-id
   ```

   Sample Configuration

   ```
   no cdl datastore session slot notification remote-system-id
   ```

2. Configure the primary site to bring up the primary site on the PCF Ops Center:

   ```
   config
     system mode running
     commit
     exit
   ```

   NOTES:

   - **config**—Enters the configuration mode.
   - **system mode running**—Configures the system mode as "running".

Verifying if the Sessions are Synchronized

This section describes how to verify if the sessions are synchronized between the sites.

The site isolation implementation requires that sessions are synchronized between the primary-secondary-primary sites. After the sites are switched, you can validate that synchronization is successful by reviewing the slots' state and indexes in both the sites. If the state of the slots and indexes is ONLINE, the synchronization status is a success. Another approach is to ensure that the local session count on both the sites match. The local session counts are synchronized between the primary and secondary site when the sessions are replicated.

To display the CDL status in the secondary site, use the following commands on the PCF Ops Center:

- To display the state of slots and indexes, run the following:

```
cdl show status
```

- To display the local session count details, run the following:

```
cdl show sessions count summary
```

Note

- Ensure the count mismatch between the sites are minimal, as the sessions count is updated with the live traffic.
- Ensure each slot and index instances has non-zero records and status shows "ONLINE" in "cdl show status" output.
- Alternatively, Grafana CDL dashboard can be used to view the total number of session counts, per slot and index records in respective panels.

Verifying if the Primary Site is Up

This section describes how to confirm if the primary site is brought up successfully.

To verify whether the primary site is up, review the deployment status and percentage usage using the following commands on the PCF Ops Center:

```
show system status deployed
show system status percent-ready
```

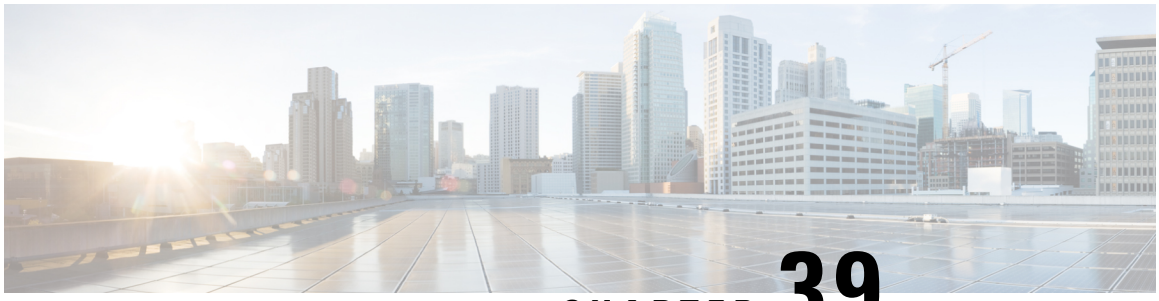
Example:

The following example displays the output of the **show system status deployed** and **show system status percent-ready** commands:

```
system status deployed true
system status percent-ready 100.0
```

NOTES:

- The deployment status of the system must be true.
- The percentage of the system must be 100.
- When the primary site is available, ensure to change the nf-status from UNDISCOVERABLE to REGISTERED to enable PCF to serve the SBI traffic. For information on how to change the nf-status, see Configuring the PCF Registration Status, on page 283.



CHAPTER 39

Simless Emergency Feature

Feature Summary and Revision History

Summary Data

Table 134: Summary Data

Applicable Product(s) or Functional Area	PCF
Applicable Platform(s)	SMI
Feature Default Setting	Disabled – Configuration required to enable
Related Documentation	Not Applicable

Revision History

Table 135: Revision History

Revision Details	Release
First introduced.	2022.01.0

Feature Description

Simless emergency feature allows the UE without a SIM to make an emergency call. Emergency calls are routed to the emergency services in accordance with national regulations to the subscriber location.

How it Works

This section describes how this feature works.

- Add the required DNN to the Emergency DNN list.
- To the added Emergency DNN, ensure that the Query LDAP is set to FALSE in DNN table to skip the LDAP Query.
- Ensure to add Is Emergency (Boolean) condition in the custom policies to process the emergency calls.

Feature Configuration

To configure this feature, use the following configuration:

1. Add DNN to the Emergency DNN List.
2. Update the DNN Table.
3. Add Is Emergency variable in the policies.

Add DNN to the Emergency DNN List

This section describes how to add DNN to the Emergency DNN List.

1. Log in into Policy Builder.
2. Click the **Reference Data** tab.
3. From the left pane, click **Systems**.
4. Click to expand your system name.
5. Click **PCF Configuration**.
6. In the right pane, add the required DNN to the **Emergency DNN List** for the emergency calls.

Update DNN Table

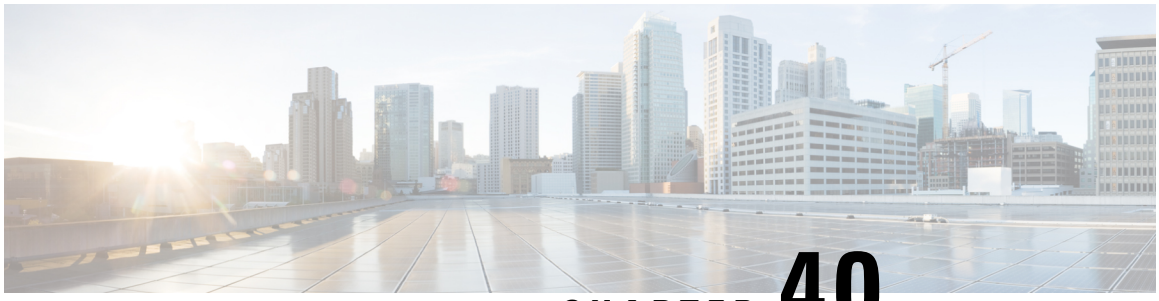
This section describes how to update the DNN table.

1. Log in to Policy Builder and navigate to Custom Reference Data.
2. Click the **Custom Reference Data**.
3. Click **Logical_dnn** in the Custom Reference Data Tables.
4. In the **Actions** column, click **Edit** symbol to change the query_ldap parameter.
5. Edit the **query_ldap** parameter to **false**.
6. Click **Close**, to close the **Logical_dnn** dialog box.

Add Is Emergency Variable in the Policy

This section describes how to add Is Emergency variable in the custom policy.

1. Log in into Policy Builder.
2. Click the **Tools** menu and choose **Preferences**.
3. Check the **Show Policies (custom configuration) editing options?** check box and then click **Ok**.
4. Click the **POLICIES** tab.
5. On the left pane, click to expand the **Initial Blueprint** folder under **Policies**.
6. Expand the **Setup network access policies**, and then click **Missing GPSI**.
7. On the right pane, click **A Policy N7 TGPP Session exists** under **Conditions** to add a new variable.
8. From the **Available Input Variables**, click **Add** to add the **Is Emergency (Boolean)** to the **Conditions** tab.
9. For the **Is Emergency (Boolean)** variable, choose **< >** from the **Operator** drop-down list and enter **true** in the **Value** column.
10. Save and publish the changes.



CHAPTER 40

Service

Feature Summary and Revision History

Summary Data

Table 136: Summary Data

Applicable Product(s) or Functional Area	PCF
Applicable Platform(s)	SMI
Default Setting	Enabled – Always-on
Related Documentation	Not Applicable

Revision History

Table 137: Revision History

Revision Details	Release
First introduced.	2020.01.0

Important Due to non-backward compatible changes in PCF operations center configuration model, a direct PCF upgrade is not possible. You must perform a fresh PCF installation after un-deploying the previous installation and clearing out the PCF configmaps from CNEE.

Feature Description

A service dictates the capabilities that are assigned to a subscriber (in USuM). An administrator assigns a service to a user through the service configurations. Depending on the service provider's requirements, PCF lets you flexibly map the service configuration with the policies.

For instance, a user with the GOLD account might get a high upload/download speed in comparison to a BRONZE user.

In a tier-based classification, if the quota is "y" then the users from the first tier are redirected to a portal and users belonging the second tier would only experience a downgrade in the speed.

Service

A service is effectively a "code" to label the service and a collection of Service Options which contain the definition of what a service is. Multiple services can be assigned to a single subscriber. If multiple services are assigned to a subscriber, the service options are combined between all assigned services.

Adding a Service

Before adding a service, ensure that you have created the corresponding Use Case template for the service that you intend to add. For information on how to create a use case template, see Configuring the Use Case Template, on page 293.

Use the following steps to add a service through Policy Builder.

1. Log in to Policy Builder.
2. Click the **Use Case Templates** from the left pane and select the template that you have created.
3. In the right pane, click **Add** to include a new service.
4. In the **Select Service Configuration** dialog box, click the appropriate entry to view the associated services.
5. Select the service and click **OK**. The selected service is added as a new service.
6. In the left pane, choose **Services** > **Service Options** to view the options.
7. Expand the service that you have created and select the child.

Note The service name resembles the name that you specified for the use case template.

8. In the **Service Option** pane, click the service under **Service Configurations** and specify the parameters referring to the relevant configuration.

Service Configuration

PCF uses the low-level configuration objects to drive a feature in the system. You can configure the Service Configuration objects from the **Service** > **Service Option** > **Use Case Template**.

Types of service configurations:

- PriorityConfiguration: Only one configuration is allowed to be active at a time. If multiples priority configurations are added, the configuration of the highest priority is used. These are used in cases where only a single value makes sense. For example, when sending an Accept message, only one template is required. Objects of this type always have a priority field. If multiple priority configurations are added, the highest priority object is used. For example, AccessAcceptConfiguration and RegisterMacAddress.
- GroupConfiguration (most common): Only 1 configuration per 'Group Name' is allowed to be active. If multiple configurations are added, the highest priority per Group Name is used. These configurations are used in cases where a configuration only makes sense for a single "group" (key). For example, to control the upload/download speed based on the network type (cell, Wi-Fi, and so on). A service configuration to control network speed with a group set for cell/Wi-Fi would allow multiple service configurations to be added. These objects always have a group field and a priority field. For each unique group value, the highest priority is used. For example, IsgServiceConfiguration, All Diameter Configurations, and OneTimeUsageCharge.
- ServiceConfiguration: Multiple configurations are allowed. If multiple configurations are added, all are used. For example, AutoChargeUpAccounts, AutoProvisionQuota, and BalanceRateConfiguration.

Note The Modify feature in PB for Use Case Options/Service Options can override the values conditionally.

Use Case Templates

Use case templates are the essential elements of the PCF architecture. The values that you define in the templates allow you to design and configure one or more services once and reuse them.

Only advanced users such as administrators are authorized to create a use case template.

On a higher-level, the use case template lets you:

- Define the Service Configuration objects to be set by a Service Option.
- Provide default values and hide values which the use case must not configure.
- Optionally, contains Initiators (Conditions) which define when the template is active.
- Makes Service Option and Service creation easier. For example, a use case template setup to create different upload or download speeds includes a DefaultBearer QoS Service Configuration object. The user creating a use case template can set default and hide the values for ARP and other values that are not directly related to upload or download speed. This allows the creation of the Service Option to be much simpler.
- A copy of the Use Case Options is created while copying a use case template.

Configuring the Use Case Template

This section describes how to configure the use case template.

Use the following steps to configure the use case template through Policy Builder.

1. Log in to Policy Builder.
2. Select the **Services** tab, and from the left pane click **Use Case Templates** to create a new service.
3. On the left pane, click **Summary** to open the **Summary** pane.
4. Under **Actions**, click **Use Case Template**.
5. In the **Use Case Template** pane, specify the name for the template.
6. Click the **Actions** tab and select **Add**.
7. In the **Select Service Configuration** dialog box, select the service and click **OK**. The **Use Case template** with the specified name is created.
8. In the left pane, click **Services > Service Options** to view the options. The newly created service appears in the **Service Options**.
9. Select the service that you have created.
10. Under **Service Configurations**, click **Add** to open the **Select Service Configuration** dialog box.
11. Under **Service Configurations**, select the service, then click **OK**.

GenericServiceConfiguration

This section describes the parameters for the GenericServiceConfiguration service configuration object.

Table 138: GenericServiceConfiguration Parameters

Parameters	Description
Priority	Denotes the priority of the message for processing. The higher the number, the higher the priority. Default for most settings: 0
Group Name	Specifies a group name. Only 1 per Group Name is allowed to be active. If multiple configurations are added highest priority per Group Name is used.
Code	Specifies a code for the AVP.
Value	Specifies a value for the AVP.
String Value	Specifies the string value.
Int Value	Indicates the integer value.
Long Value	Indicates the long value.
Boolean Value	Specifies the boolean value.

Parameters	Description
String Value to Override	Indicates whether overriding is required. For virtual services, if the value of “String Value” field matches exactly with the value of “String Value To Override”, then the value of “String Value” is over written with the “New String Value".
New String Value	The new string value that is used to overwrite the “String Value” if the value of “String Value” field matches exactly with the value of “String Value To Override”.
Precedence	Defines the second-level priority when the highest priority matches among the multiple generic service configurations.

Common Parameters

These parameters are common between many service configuration objects.

Table 139: Common Service Configuration Object Parameters

Parameter	Description
Apn Agg Max Bit Rate DL	Defines the total bandwidth usage for the downlink direction of non-GBR QCIs at the APN.
Apn Agg Max Bit Rate UL	Defines the total bandwidth usage for the uplink direction of non-GBR QCIs at the APN.
Arp	AllocationRetentionPriority • Priority Level – Priority-Level AVP value. • Preemption Capability – Preemption-Capability AVP value. • Preemption Vulnerability – Preemption-Vulnerability AVP value.
Balance Code	Indicates with which balance the quota is associated. You can subscribe to multiple balances, but the monitoring key is associated with one balance.
Diameter Client	The client configuration is used to apply different policies based on PCF type. To filter a service based on the Diameter client, specify which Diameter client you want the service to be applied to. Diameter clients are configured in the **Reference Data > Diameter Clients > Diameter Clients** section of the interface. This parameter is optional.
Dosage	How much quota to initially give the client (in bytes). Default: 0

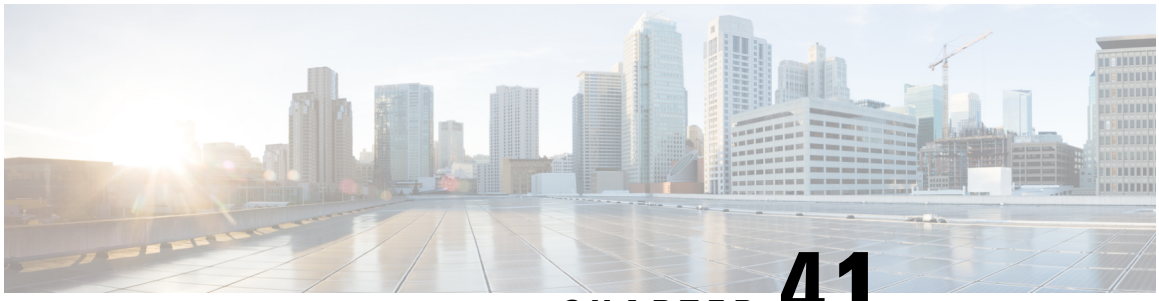
Parameter	Description
Dual Stack Session	Set to enable or disable the parameter. Default: disabled
Enable Resource Allocation Notification	Can be set to enabled or disabled. Default: disabled
Encoding Format	Can be set to true or false. If the Monitoring Key parameter is numeric, set this parameter to true. Default: false
Event Trigger	Used primarily to notify the starting and stopping of applications or to report usage. It is not used to rerequest rules.
Flow Status	Defines whether the service data flow is enabled or disabled.
Framed I P Type	Can be set to one of the following options: • ANY_ONE • BOTH • IPv4_ADDRESS • IPv6_ADDRESS Default: ANY_ONE
Guaranteed Bit Rate DL	Defines the guaranteed bit rate allowed for the downlink direction.
Guaranteed Bit Rate UL	Defines the guaranteed bit rate allowed for the uplink direction.
List of Input Column Avp Pairs (List)	Defines the mapping between the AVP Names and the key columns defined in the selected STG. These AVPs are used as inputs while evaluating the CRD table in STG. • Avp Name – The name of the Diameter AVP that is used as input for CRD table evaluation. For example: Flow-Number, Media-Component-Number, and so on. • Column – The key column in STG that corresponds to the specified AVP.
List Of Output Column Avp Pairs (List)	Defines the mapping between the AVP Names and the output columns defined in the selected STG. These mappings indicate how the output columns values are mapped to AVPs after the CRD is evaluated. • Avp Name – The name of the Diameter AVP to which the value of the output column is mapped while setting the charging parameters on the dynamic rule (for the Dedicated Bearer). For example: Rating-Group Service-Identifier. • Column – The output column defined in the selected STG.

Parameter	Description
Max Req Bandwidth DL	Defines the maximum bit rate allowed for the downlink direction.
Max Req Bandwidth UL	Defines the maximum bit rate allowed for the uplink direction.
Monitoring Key	Identifies a usage monitoring control instance. You can specify any value.
Monitoring Level	Can be set to one of the following values: • SESSION_LEVEL (0) • PCC_RULE_LEVEL (1) • ADC_RULE_LEVEL (2)
Mute Notification	Indicates whether notifications for application starts and stops are muted for ADC Rule by the TDF.
New String Value	The new string value that is used to overwrite the “String Value” if the value of “String Value” field matches exactly with the value of “String Value To Override”.
Online	Defines whether the online charging interface from PCF for the associated PCC rule is enabled. The default charging method provided by PCF takes precedence over any preconfigured default charging method at PCF. • Enable: Indicates that the online charging interface for the associated PCC rule is enabled. • Disable: Indicates that the online charging interface for the associated PCC rule is disabled.
Offline	Defines whether the offline charging interface from PCF for the associated PCC rule is enabled. The default charging method provided by PCF takes precedence over any preconfigured default charging method at PCF. • Enable: Indicates that the offline charging interface for the associated PCC rule is enabled. • Disable: Indicates that the offline charging interface for the associated PCC rule is disabled.
Precedence	Defines the second-level priority when the highest priority matches among the multiple generic service configurations.

Parameter	Description
Preemption Capability	When provided within the QoS-Information AVP, the AVP defines whether a service data flow can get resources that were already assigned to another service data flow that has a lower priority level. If it is provided within the Default-EPS-Bearer-QoS AVP, the AVP defines whether the Default Bearer can get resources that were already assigned to another bearer with a lower priority level. • 0: Indicates that the service data flow or bearer is allowed to get resources that were already assigned to another service data flow or bearer with a lower priority level. • 1: Indicates that the service data flow or bearer is not allowed to get resources that were already assigned to another service data flow or bearer with a lower priority level. This is the default value applicable if this AVP is not supplied.
Preemption Vulnerability	When provided within the QoS-Information AVP, the AVP defines whether a service data flow can lose the resources assigned to it in order to admit a service data flow that has a higher priority level. If it is provided within the Default-EPS-Bearer-QoS AVP, the AVP defines whether the Default Bearer can lose the resources assigned to it in order to admit a pre-emption capable bearer with a higher priority level. • 0: Indicates that the resources assigned to the service data flow or bearer can be pre-empted and allocated to a service data flow or bearer with a higher priority level. This is the default value applicable if this AVP is not supplied. • 1: Indicates that the resources assigned to the service data flow or bearer cannot be pre-empted and allocated to a service data flow or bearer with a higher priority level.
Priority	The priority of the message for processing. The higher the number, the higher the priority. Default for most settings: 0
Priority Levels	Used to decide whether a bearer establishment or modification request can be accepted, or rejected due to resource limitations (typically used for admission control of GBR traffic). The AVP can also be used to decide which existing bearers to pre-empt during resource limitations. The priority level defines the relative importance of a resource request. Values 1–15 are defined, with value 1 as the highest level of priority. • Values: 1–8 – Assigned for services that are authorized to receive Prioritized treatment within an operator domain. • Values: 9–15 – Assigned to resources that are authorized by the Home network and thus applicable when a UE is roaming.

Parameter	Description
Provision Default Bearer QoS	Must be bound to the appropriate column in the STG. The data contained in the STG column is of type True/False. If the value is True, the Default Bearer QoS information from the session is applied to the rule, while QoS information derived from the prior parameters in this STG is ignored.
Qci	The Quality of Service (QoS) Class Identifier. The QoS class identifier identifies a set of IP-CAN specific QoS parameters that define QoS, excluding the applicable bitrates and ARP. It is applicable both for uplink and downlink direction. The QCI values 0, 10–255 are divided for usage as follows: • 0: Reserved • 10-127: Reserved • 128-254: Operator specific • 255: Reserved
Rating Group	The charging key for the PCC rule used for rating purposes.
Realm	The destination realm where the message is sent from PCF.
Redirect Address	Indicates the target for redirected application traffic.
Redirect Address Type	Defines the address type of the address given in the Redirect-Server-Address AVP. Default: IPV4_ADDRESS
Redirect Server Address	Indicates the target for redirected application traffic.
Redirect Support	This value indicates that Redirection is enabled for a detected application's traffic.
Retry Profile	Indicates the Rule Retry Profile to be used. When PCF receives a Charging-Rule-Report indicating failure to install or to activate one or more rules, it evaluates the failed rules and takes further action.
Rule Group	Used to classify rules at PCF to change set of predefined rules based on policy. This parameter is optional.
Rule Name	A partial name configured in Policy Builder (as derived using AF-Application-Identifier and Media-Type values from the Custom dynamic rule name table in Gx Client). Default: AF

Parameter	Description
Scheduled Hour	Can be set to one of the following values: • Default: Turns off the Hour Boundary RAR enhancement feature for look-ahead rules installation at hour boundary. This causes rules to be installed at hour boundary as applicable. • CurrentHour: Rule activation time will be current time, deactivation time will be the next hour. • NextHour: Rule activation time will be the next hour, and deactivation time will be next-next hour.
Search Column	Must be bound to the Key column in the STG. The data contained in the STG column is of type Text.
Search Group	A constant value that PCF uses to search within the Search Table Group indicated by the Search Table parameter.
Search Table	The name of the table from which to perform a lookup.
String Value to Override	Indicates whether overriding is required. For virtual services, if the value of “String Value” field matches exactly with the value of “String Value To Override”, then the value of “String Value” is over written with the “New String Value".
Tdf Application Identifier	References the application detection filter (for example, its value may represent an application such as a list of URLs) to which the PCC rule for application detection and control in PCF applies.
ToD Schedule	Identifies the schedule for rule activation and deactivation.



CHAPTER 41

Session Queries over LDAP

Feature Summary and Revision History

Summary Data

Table 140: Summary Data

Applicable Product(s) or Functional Area	PCF
Applicable Platform(s)	SMI
Feature Default Setting	Disabled – Configuration required to enable
Related Documentation	Not Applicable

Revision History

Table 141: Revision History

Revision Details	Release
Enhancement introduced. Added procedural information to configure the LDAP Endpoint.	2020.02.0
First introduced.	2020.01.0

Feature Description

In the policy-based network, the SPR/LDAP initiates a NAP notification towards PCF to signify a profile change. Upon receiving the notification, the PCF refreshes the subscriber profile by requerying LDAP to receive information about the modified subscriber.

If the NAP endpoint terminates on PCRF, the PCRF forwards the NAP request to PCF when it does not find the session in the local database. In situations where the NAP endpoint terminates on PCF, the PCF requeries LDAP and CHF to refresh the subscriber details.

How it Works

This section describes how this feature works.

NAP Notifications

When you modify subscriber details, the NAP server, LDAP server, and PCF or PCRF perform the following operations:

NAP request termination on the PCRF

1. The LDAP server updates the NAP server with the modified details.
2. The NAP server broadcasts the Subscriber Change Notification message to the connected PCRF server. The message contains the unique identifier, and MSISDN or IMSI ID.
3. After receiving the message, the PCRF sends an acknowledgment to NAP. The PCRF then searches for the local session.
4. If the subscriber session is active on the PCRF, then PCRF requests the updated subscriber information from SPR or LDAP server. Depending upon the information it receives, PCRF updates the local session with the updated subscriber information and sends a Re-Auth-Request (RAR) for the Policy and Charging Rules Function (PCEF). For example, if PCRF identifies a session for the notification that contains the specified MSISDN in the PCRF then it triggers a Gx-RAR for the subscriber sessions.
5. If PCRF does not find the subscriber session locally, then the Policy Server forwards the Subscriber Change Notification to PCF. After receiving notification, PCF seeks the session locally and takes the appropriate action.

NAP request termination on PCF

When profile changes occur in NAP, it signifies that certain policies are added or modified. In this situation, the PCF performs the following:

1. Upon receiving a notification from NAP, the PCF initiates a requery or refresh request.
2. The PCF sends an N28 Subscribe Update request seeking the details of the policies that are added or updated.
3. After receiving the updates, the PCF reevaluates the policies to determine the updated policies and sends the Update_Notify message to SMF (over the N7 interface).

LDAP Queries

The Policy Server manages the 4G and 5G subscriber information in separate modules, which indicates that the PCRF continues to store the 4G-specific information, and PCF preserves the 5G-specific details. When the Policy Server receives a request seeking subscriber information, the LDAP with other components performs the following tasks:

1. The LDAP queries the MongoDB or Subscriber Profile Repository (SPR) by sending the "Get Subscriber Information" message.
2. After receiving the query, the Policy Server searches the subscriber information in the local MongoDB instance.
3. After receiving the search query, the Policy Server searches the subscriber information in the local MongoDB instance.
4. If the Policy Server discovers the subscriber information on PCRF, it sends the details to LDAP in the defined format. If the PCRF does not find the information, it forwards the request to PCF for further processing.
5. When PCF detects the information, it notifies PCRF with the subscriber information, which the PCRF forwards to the LDAP in the specified format.

Call Flows

This section describes the key call flows for this feature.

NAP Notification Call Flow

This section describes the NAP Notification call flow.

Figure 55: NAP Notification Call Flow

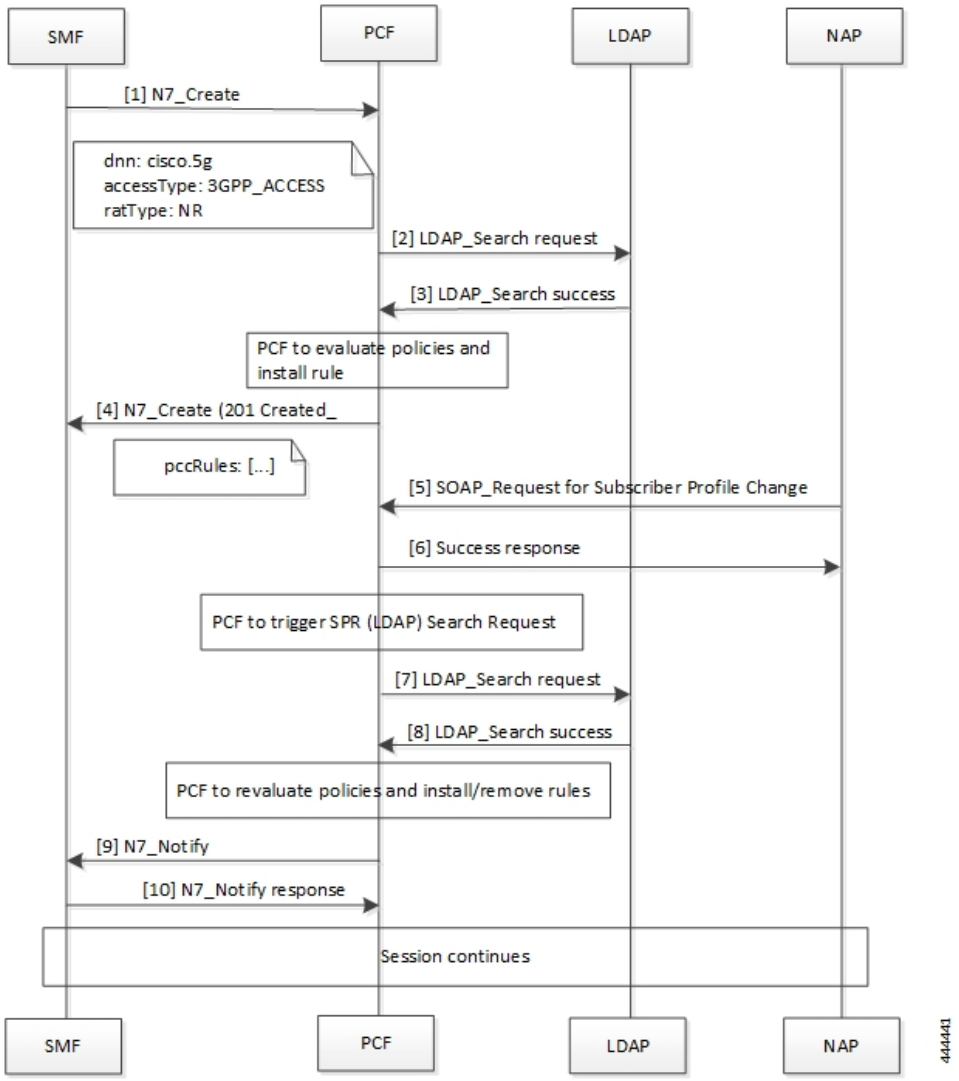


Table 142: NAP Notification Call Flow Description

Step	Description
1	The SMF sends an N7 Create request to the PCF requesting the policy details.
2	The PCF searches for the configured policies by sending the LDAP Search request towards LDAP.
3	The LDAP sends the response with search results in the LDAP Search success message to the PCF.
4	PCF evaluates the policies to determine the newly added or modified policies, and install the rules as required. The PCF responds with a set of pccRules to the original N7_Create request from the SMF with HTTP status 201.

Step	Description
5	The NAP sends a SOAP request for Subscriber Profile Change to the PCF.
6	In response to the request, PCF sends a Success response along with the requested subscriber information to NAP.
7	After PCF initiates a search request to LDAP, the PCF sends a LDAP Search request to LDAP.
8	The LDAP responds with LDAP_Search success message and the search results to the PCF.
9	PCF reevaluates the policies to determine the updated or modified policies, and installs or removes the policy rules as required. The PCF initiates an N7 Notify request to the SMF.
10	The SMF acknowledges the request with the N7 Notify response message towards the PCF.

LDAP Server Initialization Call Flow

This section describes the LDAP Server Initialization call flow.

Figure 56: LDAP Server Initialization Call Flow

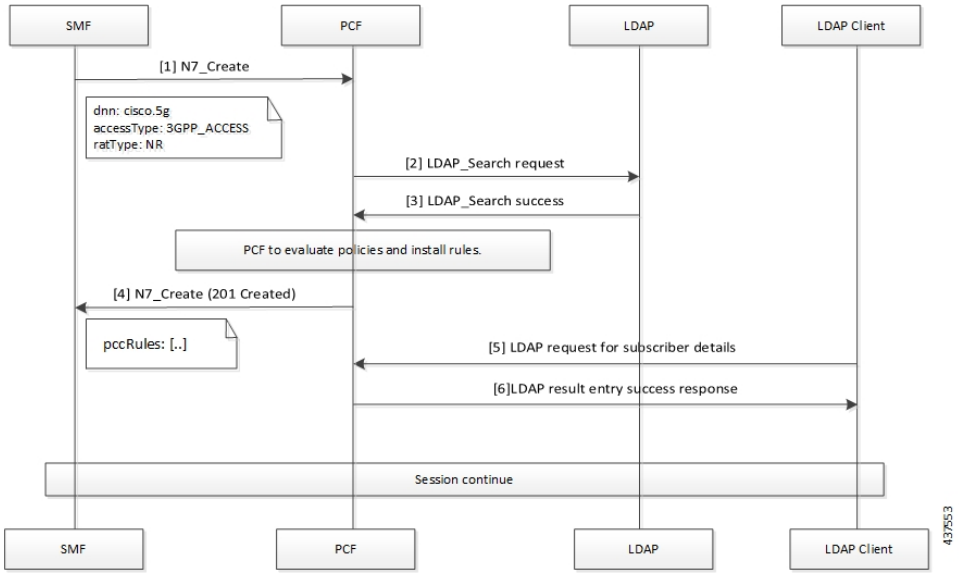


Table 143: LDAP Server Initialization Call Flow Description

Step	Description
1	The SMF sends an N7 Create request to the PCF requesting the policy details.
2	The PCF searches for the configured policies by sending the LDAP Search request towards LDAP.
3	The LDAP sends the response with search results in the LDAP Search Success message to the PCF.

Step	Description
4	PCF evaluates the policies to determine the newly added or modified policies, and install the rules as required. The PCF responds with a set of pccRules to the original N7 Create request from the SMF with the HTTP status 201.
5	The LDAP Client sends an LDAP request for Subscriber Profile Change to the PCF.
6	In response to the request, PCF sends a Success response along with the requested subscriber information to LDAP Client.

Enabling the Policy Server to Process the NAP and LDAP Queries

The configuration that enables the Policy Server to forward the NAP and LDAP queries to PCF or PCRF involves the following:

1. Configuring the gRPC Endpoint for PCF
2. Configuring the Forwarding Capability

Configuring the gRPC Endpoint for PCF

This section describes how to configure the gRPC endpoint to route the messages for PCF.

To set up the endpoint for gRPC, use the following configuration:

```
config
  engine engine_group_name
   grpc externalIPs external_ip
    port port_number
    end
```

For example,

```
engine magenta grpc externalIPs [192.0.2.18] port 8080
```

NOTES:

- **engine** *engine_group_name* —Specify the engine group name.
- **grpc externalIPs** *external_ip*—Specify the gRPC external IP address.
- **port** *port_number*—Specify the port number.

Configuring the Forwarding Capability

This section describes how to configure the forwarding capability.

For High Availability (HA) or Geographic Redundancy (GR) environments, ensure that the PCF Engine can access the Policy Server VMs. You can configure the capability responsible for routing the notification and queries by adding the following parameters to the qns.conf file.

The following table describes the application parameters.

Table 144: Application Parameters

Parameter Name	Description	Default Value	Possible Values	Example
-DsubmitToPCF	When set to true, PCRF sends NAP and LDAP requests to the PCF Engine. For HA or GR deployment, the external PCF Engine must be able to access the Policy Server VMs. Enable this feature on PCRF. This is an optional parameter.	False	True or False	-DsubmitToPCF=true
-Dpcf.host	Host or IP address of the PCF Engine on which PCRF sends the NAP and LDAP request. This parameter works when you set the submitToPCF parameter to true. Configuring this parameter is an optional step.	-	IP or host address	-Dpcf.host=192.0.2.19

Parameter Name	Description	Default Value	Possible Values	Example
-Dpcf.alternate.host	Host or IP address of the PCF Engine on which PCRF sends the NAP and LDAP requests. The NAP and LDAP requests are sent to the specified IP or host address when the address specified in the –Dpcf.host parameter is not accessible from the Policy Server. This parameter is usable only when you set the submitToPCF parameter to true. Configuring this parameter is an optional step.	-	The IP or host address	-Dpcf.alternate.host =192.0.2.20
-Dpcf.actions.sync.timeout Ms.default	The timeout period in milliseconds. Policy Server reports a timeout message when the PCRF sends a NAP and LDAP request and waits for the response until the specified interval is met. Configuring this parameter is an optional step.	350 (recommended value)	An integer value	-Dpcf.actions.sync.timeout Ms.default=350

Parameter Name	Description	Default Value	Possible Values	Example
-Dpcf.engine.port	The port number on which the PCF Engine is running. The NAP and LDAP requests are directed to this port number.	9884	An integer value	-Dpcf.engine.port=9884

Configuration Support for PCF-NAP Requests

This section describes the prerequisites and configurations that are required to support the PCF-NAP communication.

This configuration support involves the following:

- Prerequisites for PCF-NAP Requests

1. Configuring the Unified API
2. Setting a Limit on NAP Requests

Prerequisites for PCF-NAP Requests

This section describes the prerequisites that must be met for PCF-NAP communication.

For PCF-NAP interaction, make sure that the following configurations are available in your environment:

- N7 interface must be configured. For information on configuring the N7 interface, see Configuration Support for the N7 and N28 Interface, on page 317.
- LDAP must be configured to operate with PCF. For information on configuring the LDAP, see Configuring PCF to use LDAP, on page 137.

Configuring the Unified API

This section describes how to configure the unified API through the PCF Ops Center.

PCF receive NAP requests to requery the LDAP and reevaluate policies after receiving notification about profile change from NAP, so the new policies are applied. PCF receives the NAP requests through the unified API ingress endpoint.

To configure the unified API, use the following configuration in the Policy Ops Center console:

```
config
  api unified
    engine-group engine_group_name
    external-port external_ip
```

```
externalIPs external_ip
end
```

NOTES:

- **api unified**—Enter the unified API configuration mode.
- **engine-group** *engine_group_name* —Specify the PCF engine's group name.
- **external-port** *port_number*—(Optional) Specify the service to be accessed using an external IP instead of an Ingress endpoint. Specifies the external port number to expose the unified API endpoint.
- **externalIPs** *external_ip*—(Optional) Specify the service to be accessed using an external IP instead of an Ingress endpoint. Specifies the IP address for the external endpoint.

Setting a Limit on NAP Requests

This section describes how to set a limit on the number of NAP requests for PCF to process.

To configure the maximum number NAP requests TPS per PCF Engine deployment, use the following configuration in the Policy Ops Center console:

```
config
  engine engine_name
     properties broadcast.tps value tps
     end
```

NOTES:

- **engine** *engine_name* —Specify the engine name.
- **properties broadcast.tps value** *tps*—Specify the maximum number of NAP requests TPS that each PCF Engine must process. The default value is 20.

Configuration Support for LDAP Endpoint

This section describes how to configure the LDAP server endpoint that enables PCF to establish a connection with LDAP.

The configuration of the LDAP server endpoint involves the following steps:

1. Configuring the LDAP Endpoint
2. Setting a Limit on LDAP Search Request

Configuring the LDAP Endpoint

This section describes how to configure the LDAP server endpoint and the associated filter mappings.

Based on the LDAP endpoint configuration, the LDAP endpoint authenticates itself with PCF to retrieve the subscriber details through the search query.

Note Configuration changes to the LDAP endpoint cause the endpoint to restart automatically. Cisco recommends making such changes only within the maintenance window.

To configure the LDAP server endpoint, use the following configuration in the Policy Ops Center console:

```
config
  ldap-server-endpoint
  connect
    bind-ip ip_address
      port port_number
    binddn username
      password password
    request-timeout timeout
      replica replica_count
    max-transactions maximum_transaction
  health-check-attributes attribute_name
    valueattribute_value
  health-check-filter name attribute_name
    valueattribute_value
  ldap-clients client_name
    passwordpassword
  input-mapping filter_from_client
  internal-lookup-key [ IMSI | IP_ADDRESS | MSISDN ]
  output-mapping output_attribute_name
    input session_attribute_name
    end
```

NOTES:

- **ldap-server-endpoint**—Enters the LDAP server endpoint configuration mode.
- **connect**—Enters the LDAP connection configuration.
- **bind-ip** *ip_address* **port** *port_number* **request-timeout** *timeout*—Specify the external IP address and port number to which the LDAP client can connect to externally. The default port number is 9389.
- **binddn** *username* **password** *password*—Specify the user DN, for example: cn=manager, ou=account, so=profile, and password for connecting to the LDAP server.
- **request-timeout** *timeout_duration* —Specify the duration in milliseconds after which the request expires. The request awaits a response from the PCF engine. The default timeout value is 2000.
- **replica** *replica_count* —Specify the replica count for the LDAP server.
- **max-transactions** *maximum_transaction*—Specify the maximum number of transactions per second that each connection must process. The default value is 200.
- **health-check-attributes** *attribute_name* **value** *attribute_value*—Specify the attribute name and value that the client receives as a response to the health check request.
- **health-check-filter name** *attribute_name* **value** *attribute_value*—Specify the attribute name and value that distinguishes the health check request.

- **ldap-clients** *client_name* **password** *password*—Specify the configuration that PCF uses to configure multiple client authentication parameters.
- **input-mapping** *filter_from_client*—Specify the configuration to map the filter ID received from LDAP client and the internal-lookup-key. The accepted value must contain text string. For example, IMSI, MSISDN, framedIp, framedIpv6Prefix. You can configure the input mapping separately for frameIP, MSISDN, IMSI, and framedIpv6Prefix.
- **internal-lookup-key [IMSI | IP_ADDRESS | MSISDN]**—Configures the internal lookup key.
- **output-mapping** *output_attribute_name* **input** *session_attribute_name* —Specify the table that is used to defile the response attributes for the client. The response attribute name is mapped to the internal CPS session attributes for added flexibility.

Note PCF does not process the requests for which the output-mapping configuration is missing. The response attributes contain only those values that are configured in the output mapping as input key.

You can configure multiple supported keys only if they are available in the PCF session. The input keys can be duplicate but not the output values that you cannot configure two output-mappings with the same values.

Setting a Limit on LDAP Search Request

This section describes how to set the limit on the number of LDAP search requests for PCF to process.

To configure the maximum number LDAP requests TPS per replica, use the following configuration in the Policy Ops Center console:

```
config
  ldap-server-endpoint connect
  max-transactions max_tps
  end
```

NOTES:

- **max-transactions** *max_tps* —Specify the maximum number of LDAP requests TPS that each replica must process. The default value is 200.

OAM Support

This section describes operations, administration, and maintenance support for this feature.

Statistics

This section provides the list of statistics and counters that are involved when the Policy Server routes the LDAP queries and NAP notification to PCF or PCRF.

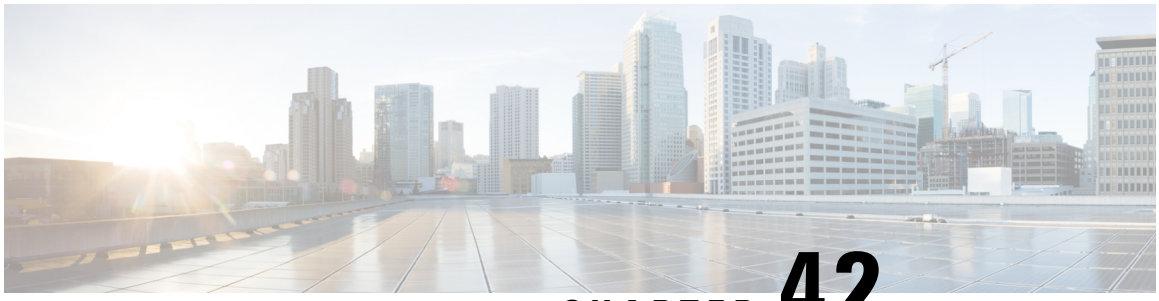
- PCF:

- inbound_request_total: Captures the total number of inbound LDAP search requests that PCF receives.
- incoming_request_total: Captures the total number of search results that contain the result code.
- LDAP_CHANGE-RES success: Invoked when the LDAP change message is successfully sent to the PCF Engine.
- LDAP_CHANGE-RES error: Invoked when the LDAP change message is not sent to the PCF Engine because of some exception.
- LDAP_SEARCH-RES success: Invoked when the LDAP query receives successful response from the PCF Engine.
- LDAP_SEARCH-RES error: Invoked when the LDAP queries fail to process due to an error or an exception.
- ldap_policy_request_total: Captures the total count of LDAP policy requests.
- message_total: Captures the total NAP requests such as total count of ldap_notify and ldap-change-message messages.

- PCRF:
  - ldap_change_success: Invoked when the PCRF receives success response from PCF for a NAP notification.
  - ldap_change_timeout: Invoked when the PCRF receives timeout response from PCF for a NAP notification.
  - ldap_change_<MessageType>: Invoked when the PCRF receives an error message from PCF for a NAP notification.
  - ldap_search_success: Invoked when the PCRF receives success response from the PCF for the LDAP queries.
  - ldap_search_timeout: Invoked when the PCRF receives timeout response from the PCF for the LDAP queries.
  - ldap_search_<MessageType>: Invoked when the PCRF receives an error message from the PCF for the LDAP queries.

- PCRF counters:
  - ldap_search_send: Captures the count of the cumulative number of the LDAP queries which the PCRF sends to the PCF.
  - ldap_change_send: Captures the count of the cumulative number of the NAP notifications that PCRF sends to the PCF.

For information on statistics, see *Ultra Cloud Core 5G Policy Control Function Statistics Reference*.



CHAPTER 42

Specification Compliance - N7 and N28

Feature Summary and Revision History

Summary Data

Table 145: Summary Data

Applicable Product(s) or Functional Area	PCF
Applicable Platform(s)	SMI
Feature Default Setting	Disabled – Configuration required to enable
Related Documentation	Not Applicable

Revision History

Table 146: Revision History

Revision Details	Release
Enhancement introduced. With this release, various service configurations are added and enhanced for compliance with the 3GPP December 2018 specification. The procedures to configure use case initiators and retrievers are also added.	2020.01.0
First introduced.	Pre 2020.01.0

Feature Description

PCF complies to the 3GPP December 2018 specification by supporting the N7 and N28 interfaces.

Relationships

The SMF should comply to 3GPP December 2018 specification so that PCF can apply the dedicated bearer rules to SMF.

Components

The following components are involved when PCF is configured to work with the N7 and N28 interface.

- N15 Interface
- N28 Interface
- N7 Interface
- N5 Interface

N15 Interface

The N15 interface complies with *3GPP TS 29.507 Release 15* specification.

N28 Interface

The N28 interface complies with the 3GPP December 2018 specification. Hence, no compliance changes are required on the N28 interface for this feature.

N7 Interface

The pcf-rest-ep and pcf-engine comply with the 3GPP December 2018 specification.

N5 Interface

The pcf-rest-ep and pcf-engine comply with the 3GPP December 2020 specification.

Rx Interface

With the N7 interface being 3GPP December 2018 specification compliant, PCF supports the notification of rule status of the dedicated bearer rules. You can configure this support with the new "ruleReports" parameter, which is sent in the SmPolicyUpdateRequest message. This parameter consists of the report of rule status as successful or failed.

When IMS initiates multimedia calls, PCF installs the dedicated bearer rules on the SMF nodes. SMF enforces these rules and provides the status of these rules to PCF in the ruleReports parameter.

Following are some of the subparameters of the ruleReports parameter that the feature supports:

- Array of pccRuleIds

- Rule Status
- Failure Code

The N5 (Rx) interface works in the following way:

- If the AAR message on the N5 interface receives the “INDICATION_OF_SUCCESSFUL_RESOURCES_ALLOCATION (8)” action, PCF sends the “SUCC_RES_ALLO” event trigger to SMF in the SmPolicyUpdateNotify message.
- SMF sends the status report in the SmPolicyUpdateRequest message. When PCF sends the “SUCC_RES_ALLO” event trigger, ruleReports parameter consists of the rules with the Rule Status as Active. Rule Status value can either be “Active” or “Inactive”. PCF sends these rule reports through Rx_RAR toward AF.
- The ruleReports parameter also consists of “Inactive” rules along with their Failure Status. If some rules are Inactive, PCF sends the failure report of these in Rx_RAR to AF. If all the rules are Inactive, PCF sends the failure report of these rules in Rx_ASR to AF.

Note PCF sends the failure report to AF if the Rx session has already requested the "INDICATION_OF_FAILED_RESOURCES_ALLOCATION(9)" specific action in AAR.

- PCF handles reporting of both Active and Inactive rule status in the same ruleReports along with the “SUCC_RES_ALLO” event trigger.
- If multiple Inactive rules exist under ruleReports with multiple failureCode, then all the rules are mapped to only single Specific-Action. Then, PCF sends these rules in the Rx_RAR (or abort-cause for Rx_ASR).

Configuration Support for the N7 and N28 Interface

This section describes how to configure the N7 and N28 interface by configuring the following services:

- SessionRule
- SessionRuleAction
- SessionRuleConditionData
- QosData
- TableDrivenQosDecision
- TableDrivenDynamicPccRule
- Use Case Initiators
- Retrievers

SessionRule

This section describes how to configure SessionRule Service.

The SessionRule service configuration consists of policy information elements that are associated with a PDU session. The object configures the QoS attributes for the default bearer.

Before setting the service parameters, ensure that you create a use case template and add a service for this configuration. For details, see Configuring the Use Case Template, on page 293 and Adding a Service, on page 292.

To configure the SessionRule service, use the following configuration:

1. Log in to Policy Builder.
2. Click the **Services** tab.
3. Configure the following parameters:
   a. From the **Use Case Template** pane, select the N7 service configuration object.
   b. From the **Service Configurations** pane, select **SessionRule**.
   c. Click **Add**.
4. From the N7 service configuration, click **SessionRule**.

   The service configuration parameters appear in the right pane.
5. Configure one or more of the following **SessionRule** parameters:
   - **Sess Rule Id**: Enter a unique value to identify the session rule within a PDU session.
   - **Uplink**: This parameter is the Aggregate Maximum Bit Rate (AMBR) for the uplink frequency. Configure the value of this parameter in BitRate.
   - **Downlink**: This parameter is the AMBR for the downlink frequency. Configure the value of this parameter in BitRate.
   - **5qi**: This 5G QoS Identifier identifies a specific QoS forwarding behavior for a 5G QoS flow. Configure a numeric value for this parameter.
   - **Arp**: Configure the following Allocation Retention Priority levels:
     - **Priority Level**: Configure this parameter to define the relative importance of a resource request.
     - **Preempt Cap**: Configure this parameter to define a service data flow to reassign the resources. These resources are already assigned to another service data flow with a lower priority level.
     - **Preempt Vuln**: Configure this parameter to define a service data flow to lose the assigned resources for admitting a service data flow with a higher priority level.
   - **Priority Level**: Configure this parameter to indicate the 5QI priority level. Enter a value from 1 through 127, where 1 implies the highest level, and 127 implies the lowest level.
   - **Aver Window**: Configure this parameter to indicate the Averaging Window, which is in milliseconds. Enter a value from 1 through 4095, where 1 implies the minimum averaging level and 4095 implies the maximum averaging level.

- **Max Data Burst Vol**: Configure this parameter to indicate the Maximum Data Burst Volume, which is in bytes. Enter a value from 1 through 4095, where 1 implies the minimum data burst volume and 4095 implies the maximum data burst volume.

6. Click **Add**.

SessionRuleAction

This section describes how to configure the SessionRuleAction service.

The SessionRuleAction service configures the values for the various SessionRuleAction attributes that are based on Policy or the configured SessionRule.

Before setting the service parameters, ensure that you create a use case template and add a service for this configuration. For details, see Configuring the Use Case Template, on page 293 and Adding a Service, on page 292.

To configure the SessionRuleAction service parameters, use the following configuration:

1. Log in to Policy Builder.
2. Click the **Services** tab.
3. Configure the following parameters:
   a. From the **Use Case Template** pane, select the N7 service configuration object.
   b. From the **Service Configurations** pane, select **SessionRuleAction**.
   c. Click **Add**.

   The **Select Service Configuration** window appears.
4. From the N7 service configuration, click **SessionRuleAction**.

   The service configuration parameters appear in the right pane.
5. For all the parameters of **SessionRuleAction** service, configure one of the following options:
   - **Mirror**: The value that is requested on the N7 interface is granted.
   - **Enforce**: The default bearer QoS value is granted. Enforce is the default value.
   - **Bound**: The minimum value between the configured value, and the requested value is granted and is sent back as a response.
6. Click **Add**.

SessionRuleConditionData

This section describes how to configure the SessionRuleConditionData service.

The PCF schema lets you set a threshold for an active SessionRules service by configuring the SessionRuleConditionData service. The value that you define in this service configuration indicates the period for which the SessionRules service remains active. After the set period is complete, the SessionRules service is deactivated. The SessionRuleConditionData service includes the Extend Deactivation parameter. If you

have configured the Extend Deactivation parameter and an event occurs within the configured interval, then PCF extends the waiting period. For example, if the deactivation time is set to 2 hours, and an event occurs after 1 hour, then the activation time is extended by 2 hours from the time the event happened. The service remains active for 3 hours.

Before setting the service parameters, ensure that you create a use case template and add a service for this configuration. For details, see Configuring the Use Case Template, on page 293 and Adding a Service, on page 292.

To configure the SessionRuleConditionData service parameters, use the following configuration:

1. Log in to Policy Builder.
2. Click the **Services** tab.
3. From the **Use Case Template** pane, select the N7 service configuration object.
4. From the **Service Configurations** pane, select **SessionRuleConditionData**, and click **Add**.

   The **Select Service Configuration** window appears.
5. From the N7 service configuration, click **SessionRuleConditionData**.

   The service configuration parameters appear in the right pane.
6. Configure one or more of the following **SessionRuleConditionData** parameters:
   - **Priority**: Enter an integer value to indicate the priority of the service configuration object. This value is considered in case multiple service initiator conditions match.
   - **Deactivation Time (In Minutes)**: Specify the time in minutes after which the SessionRule service is deactivated.
   - **Deactivation Time (In Hours)**: Enter the time in hours after which the SessionRule service is deactivated.
   - **Deactivation Time (In Seconds)**: Specify the time in seconds after which the SessionRule service is deactivated.

     Note PCF aggregates the values that you specify in hours, minutes, and seconds to determine the deactivation time.
   - **Extend Deactivation**: If set to true, then PCF extends the SessionRule service deactivation time for the period that you have specified for deactivation.

QosData

This section describes how to configure the QosData Service.

The QosData Service configuration is updated with the parameters to meet the 3GPP December 2018 specification compliance.

Before setting the service parameters, ensure that you create a use case template and add a service for this configuration. For details, see Configuring the Use Case Template, on page 293 and Adding a Service, on page 292.

To configure the QosData service from Policy Builder, use the following configuration:

1. Log in to Policy Builder.
2. Click the **Services** tab.
3. Configure the following parameters:
   a. From the **Use Case Template** pane, select the N7 service configuration object.
   b. From the **Service Configurations** pane, select **QosData**.
   c. Click **Add**.
4. From the N7 service configuration, click **QosData**.

   The **Select Service Configuration** window appears.
5. Configure one or more of the following **QosData** parameters:
   - **Qnc**: Enter a boolean value for this parameter. This parameter indicates whether notifications are requested from 3GPP NextGen RAN (NG-RAN) when the Guaranteed Flow Bit Rate (GFBR) is no longer available for a QoS Flow during the lifetime of the QoS Flow. By default, the value of this parameter is **False**.
   - **Priority Level**: Enter an integer value to indicate the scheduling of resources among QoS Flows.
   - **Aver Window**: Enter an integer value to indicate the duration for which the guaranteed, and maximum bitrate is to be calculated.
   - **Max Data Burst Vol**: Enter an integer value to indicate the maximum amount of data to be transferred for 5G-AN PDB.
   - **Reflective QoS**: Enter a boolean value to indicate whether the QoS information is reflective for the corresponding Service Data Flow. The default value of this parameter is **False**.
   - **Sharing Key Dl**: Configure this parameter to indicate the PCC rules that can share a resource in the downlink direction.
   - **Sharing Key Ul**: Configure this parameter to indicate the PCC rules that can share a resource in the uplink direction.
   - **Max Packet Loss Rate Dl**: Configure this parameter to indicate the downlink maximum rate for the lost packets that can be used for a service data flow.
   - **Max Packet Loss Rate Ul**: Configure this parameter to indicate the uplink maximum rate for the lost packets that can be used for a service data flow.
   - **Def Qos Flow Indication**: Enter a boolean value to indicate the binding of QoS Flow, which is associated with the default QoS rule, with the dynamic PCC rule. The default value of this parameter is **False**.
6. Click **Add**.

TableDrivenQosDecision

This section describes how to configure the TableDrivenQosDecision Service.

The TableDrivenQosDecision service configuration captures the data from the custom CRD table for the newly added fields. This table defines and associates the parameters in the table.

Before setting the service parameters, ensure that you create a use case template and add a service for this configuration. For details, see Configuring the Use Case Template, on page 293 and Adding a Service, on page 292.

To configure the TableDrivenQosDecision service parameters, use the following configuration:

1. Log in to Policy Builder.
2. Click the **Services** tab.
3. Configure the following parameters:
   a. From the **Use Case Template** pane, select the N7 service configuration object.
   b. From the **Service Configurations** pane, select **TableDrivenQosDecision**.
   c. Click **Add**.

   The **Select Service Configuration** window appears.
4. From the N7 service configuration, click **TableDrivenQosDecision**.

   The service configuration parameters appear in the right pane.
5. Configure one or more of the following **TableDrivenQosDecision** parameters:
   - **Qnc Source**: Enter the primary key value for the column configured under **Search Column**.
   - **Authorized QoS Priority Level**: Specify the priority that is used for scheduling the resources among the QoS Flows.
   - **Aver Window Source**: Specify the duration over which the guaranteed, and maximum bitrate is calculated.
   - **Max Data Burst Vol Source**: Enter the maximum amount of data that is required to be transferred within a period of 5G-AN PDB.
   - **Reflective QoS Source**: Specify the value that applies reflective QoS for the SDF.
   - **Sharing Key Dl Source**: Specify the value that indicates resource sharing in downlink direction with the service data flows having the same value in their PCC rule.
   - **Sharing Key Ul Source**: Specify the value that indicates resource sharing in an uplink direction with the service data flows having the same value in their PCC rule.
   - **Max Packet Loss Rate DI Source**: Specify the maximum rate for lost packets that can be tolerated in the downlink direction for the service data flow.
   - **Max Packet Loss Rate UI Source**: Enter the maximum rate for lost packets that can be tolerated in the uplink direction for the service data flow.
   - **Def QoS Flow Indication Source**: Enter the value that indicates the dynamic PCC rule that shall always have its binding with the default QoS Flow.

   Only the mandatory parameters are listed in this section. For the complete list of parameters, see TableDrivenQosDecision, on page 107.

6. Click **Add**.

TableDrivenDynamicPccRule

This section describes how to configure the TableDrivenDynamicPccRule service.

The TableDrivenDynamicPccRule service configuration shows one or more PCC rules that are available in the custom reference data table.

Before setting the service parameters, ensure that you create a use case template and add a service for this configuration. For details, see Configuring the Use Case Template, on page 293 and Adding a Service, on page 292.

To configure the TableDrivenDynamicPccRule service parameters, use the following configuration.

1. Log in to Policy Builder.
2. Click the **Services** tab.
3. Configure the following:
   a. From the **Use Case Template** pane, choose the N7 service configuration object.
   b. From the **Service Configurations** pane, choose **TableDrivenDynamicPccRule**.
   c. Click **Add**.

   The **Select Service Configuration** window appears.
4. From the N7 service configuration, click **TableDrivenDynamicPccRule**.

   The service configuration parameters appear in the right pane.
5. Configure one or more of the following **TableDrivenDynamicPccRule** parameters:
   - **Search Table**: Enter the name of the table that is used to perform a lookup.
   - **Search Column**: Enter a value, which is associated with the primary key column of the STG. The data that is contained in the STG column has a text value.
   - **Search Value**: Enter a value of the **Search Column** primary key to perform a lookup in **Search Table**.
   - **Input List (List)**: If the lookup requires extra key or value pairs, then configure this field using **InputColumn** under **Input List**.
     - **Column Name**: Enter a value that is associated with the additional key column of STG.
     - **Column Value**: Enter the value that you have entered for the **Column Name** field.
   - **Pcc Rule Id Source**: Specify a value for the key column that is associated with the PccRuleId column within the STG. The data that is contained in the STG column has a text value.
   - **Precedence Source**: Enter a value for this field that is associated with the Precedence column in STG. The data that is contained in the STG column has a numeric value. For the allowed values, see 3GPP specification 29.512.
   - **App Id Source**: Specify a value that is associated with the App Id column in the STG. The data that is contained in the STG column has a text value.

- **Qos Id Source**: Enter a value that is associated with the Qos Id column in the STG. The data that is contained in the STG column has a text value.
- **Chg Id Source**: Specify a value that is associated with the Chg Id column in the STG. The data that is contained in the STG column has a text value.
- **Flow Information Source**: Enter a value that is associated with the **Flow Information column** in the STG. The data that is contained in the STG column has a text value.

  Only the mandatory parameters are listed in this section. For the complete list of parameters, TableDrivenDynamicPccRule, on page 109.

Note

- Use the following format to add the **Flow Information Source** parameter for PCF to perform Flow Information grouped parameter mapping:

  Multiple Flow Information is separated by "," as delimiter, whereas each Flow Information the format is: <Flow Description1>;<packetFilterUsage1>;<Tos TrafficClass1>;<Spi1> ;<Flow Label1>;<Flow Direction1>, <Flow Description2>;<packetFilterUsage2>

- Using an incorrect format results in missing the Flow Information value.

6. Click **Add**.

Use Case Initiators

This section describes how to configure the Use Case Initiators.

Use case initiators are a group of conditions that indicate the time, event, or functionality for a specific service to be added for a subscriber. If no use case initiators are configured, then the service configuration objects are added.

1. Log in to Policy Builder.
2. Click the **Services** tab.
3. From the **Use Case Template** pane, select the N7 service configuration object, and click the **Use Case Initiators** tab.
4. In the **Service Initiators (OR Together)** field, configure multiple use case initiators, which activate the Use Case Template and its service configurations. Service Initiators are a group of conditions. If any one of the service initiators of the use case template is true then the service configuration of that use case template are used.

Note In the **Service Initiators (OR Together)** box, select the add icon to add a service initiator and close icon to remove a service initiator. Use the up or down arrow buttons to specify the order in which service initiators are to be evaluated.

5. In the **Initiator Name** field, enter a name for the group of conditions.
6. In the **Conditions (AND Together)** field, click **Add**.

   The condition box appears.
7. Select one or more conditions from the Select a Condition Phrase window from the PCF, N7, 3GPP, or N28 3GPP messages option. The conditions that you add are associated with the service initiator. Conditions are related to the messages session, subscriber information, balance information, or the message itself.

   The conditions that you select appear in the **Conditions (AND Together)** box.

   Note If multiple conditions exist for the **Conditions (AND Together)** box, then all the chosen conditions must be true for them to be configured.
8. In the **Conditions (AND Together)** box, click one or multiple conditions.

   The input variables for the selected condition appear in the right pane.
9. From the available input variables, add all or the required input variables. See the Conditions of Input Variables, on page 325 section for the conditions to configure for the input variables.

   The input variables are added.
10. Configure the values for the added variables.

    The use case initiators are configured.

Conditions of Input Variables

The table lists the conditions for the input variables.

Condition for Input Variable	Description
A Policy Request Message exists	An N7, N15, N28, or UDR Policy Request message exists in the system.
A Policy Request Message does not exist	An N7, N15, N28, or UDR Policy Request does not exist in the system.
A Policy Message exists	An N7, N15, N28, or UDR Policy Request or Response message exists in the system.
A Policy Message does not exist	An N7, N15, N28, or UDR Policy Request or Response message does not exist in the system.
A Policy Response Message exists	An N7, N15, N28, or UDR Policy Response message exists in the system.
A Policy Response Message does not exist	An N7, N15, N28, or UDR Policy Response message does not exist in the system.
A Policy N7 TGPP Session exists	A valid N7 TGPP session exists in the Policy Builder configuration for a subscriber.

Condition for Input Variable	Description
A Policy N7 TGPP Session does not exist	A valid N7 TGPP session does not exist in the Policy Builder configuration for a subscriber.
A N28 TGPP Session exists	A valid N28 TGPP session exists in the Policy Builder configuration for a subscriber.
A N28 TGPP Session does not exist	A valid N28 TGPP session does not exist in the Policy Builder configuration for a subscriber.

Retrievers

Retrievers are the values that are retrieved for the key columns from the custom reference data table. You can retrieve the value of retrievers in the following ways in Policy Builder.

- Using the Service Configuration option
- Using the custom reference data table

Following is the list of the available session-level retrievers in PCF:

- N7 Access Type
- N7 Cell Global Identifier
- N7 DNN
- N7 GPSI
- N7 MCC (SUPI Based)
- N7 MNC (SUPI Based)
- N7 Permanent Equipment Identifier
- N7 RAT Type
- N7 Serving Network
- N7 SliceInformation
- N7 SUPI
- N7 Tracking Area Identifier

Note These retrievers are session-level retrievers and not the message-level retrievers. It implies that as a request is received, a session is created in the database for a subscriber. Then, the value of these retrievers is retrieved from those sessions.

Configuring Retrievers through Custom Reference Data Table

This section describes how to configure the retrievers through the custom reference data table.

1. Log in to Policy Builder.
2. From the **Custom Reference Data Tables** pane, select a rule.

   The parameters of the selected custom reference table appear in the right pane.
3. In the **Columns** box, select a key column.
4. Click the **Bind to Session/Policy State** option and click **Select**.

   A dialog box with the list of available retrievers appears.
5. In the text box, enter **N7** or **N15**to view all the newly added retrievers.
6. Select a retriever and click **OK**.

Configuring Retrievers through Service Configuration

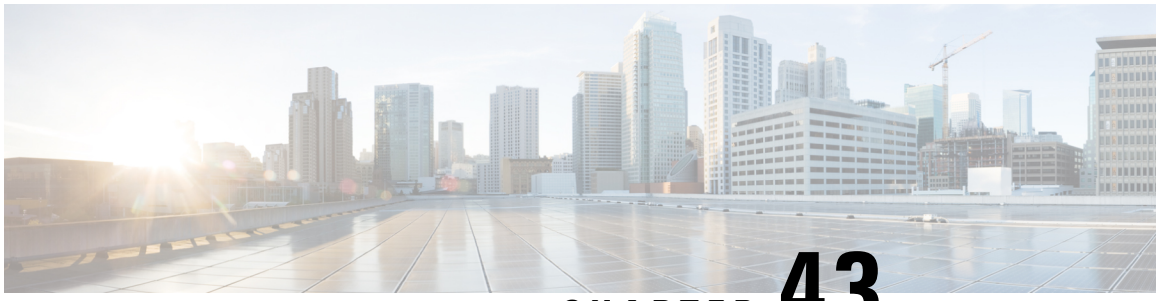
This section describes how to configure the retrievers through the service configuration.

1. Log in to Policy Builder.
2. From the Services pane, click the N7/N15 service-compliant configuration.
3. Select a service configuration from **Service Configurations** box.

   The parameters of the selected service parameters appear.
4. Choose a parameter and click the pull value icon to pull a value for the parameter.

   The **Dynamically pull this value from** window appears.
5. Click the **Bind to Session/Policy State** option and click **Select**.

   A window showing the list of available retrievers appears.
6. In the text box, enter **N7** or **N15** to view all the newly added retrievers.
7. Select a retriever and click **OK**.



CHAPTER 43

Status Monitoring Using Commands

Feature Summary and Revision History

Summary Data

Table 147: Summary Data

Applicable Product(s) or Functional Area	PCF
Applicable Platform(s)	SMI
Feature Default Setting	Enabled – Configuration required to disable
Related Documentation	Not Applicable

Revision History

Table 148: Revision History

Revision Details	Release
Enhancement introduced. Introduced procedure to view the NF connection status	2020.03.0
First introduced.	2020.01.0

Feature Description

PCF allows you to view the Diameter peer, LDAP connection, and NRF registration status using the PCF Ops Center. Alternatively, you can access this information from the Grafana dashboard. For information on Grafana, see the *Ultra Cloud Core 5G Policy Control Function Statistics Reference*.

PCF provides an in-depth information about the NF endpoint that it has discovered through NRF. PCF discovers an NF endpoint only when both, PCF and the NF are registered with the same NRF. PCF also lets you view the subscriber session details across the configured regions.

Viewing the Connection and Registration Status

This section describes how to view the Diameter peer, LDAP connection, and NRF registration status.

Based on the component that you want to view the status for, use the following configuration:

- To view the status of the configured Diameter peers, use the following configuration:

  `show diameter peer-status`

  The output of this command displays the peer hostname, peer realm, and the peer status.

  The following is a sample output of the **show diameter peer-status** command.

```
pcf# show diameter peer-status
PEER HOSTNAME   PEER REALM                      PEER
                                                     STATUS
-----------------------------------------------------------
site-host-rx2   site-rx-client-mob-1.com        Connected
site-host-rx1   site-rx-client-mob-2.com        Connected
```

- To view the status of the LDAP connection, use the following configuration:

  `show ldap connection-status`

  The output of this command displays the ServerSet, maximum number of available connections, and the number of available connections.

  The following is a sample output of the **show ldap connection-status** command.

```
pcf# show ldap connection-status
SERVERSET   METRIC                      VALUE
-------------------------------------------------
USD         MaximumAvailableConnections    10
USD         NumAvailableConnections        0
```

- To view the NRF registration status, use the following configuration:

  `show rest-endpoint registration-status`

  The output of this command displays the IP address of the pod, registration status, and the registered NRF URI.

  The following is a sample output of the **show rest-endpoint registration-status** command.

```
pcf# show rest-endpoint registration-status
POD IP             REGISTERED    NRF URI
-------------------------------------------------------------
192.0.2.19:8486 Registered    http://192.0.2.12:8001/
```

Viewing the NFs Connected to PCF

This section describes how to view the NFs that are presently connected to PCF.

- To view the NFs that are currently connected to PCF, use the following configuration:

  show rest-endpoint peer-status

  The command displays the mapped port numbers only if you have configured the ports for the interface. After you configure the port and send a request to that port, then on running the **show rest-endpoint peer-status** command, the PCF displays the mapping details. If the request comes to the pcf-rest-ep external port, which is not associated with any configured interface port, then the NFName is displayed as "UNKNOWN".

  For information on how to configure the NFs and the corresponding port numbers with PCF, see *Configuring the External IP Address* section in the *Multiple Virtual IP Address* chapter.

  The output of this command displays NF name, peer details, pod IP address, peer IP address, and the duration for which the connection is active.

```
pcf# show rest-endpoint peer-status
PEER  NF
PORT  NAME  POD IP        PEER IP     CONNECTION DURATION
-------------------------------------------------------------------------
8142  AMF   192.0.2.255  192.0.2.254  1 days 17 hours 44 minutes 38 seconds
8147  AMF   192.0.2.255  192.0.2.254  1 days 17 hours 44 minutes 13 seconds
6082  CHF   192.0.2.255  192.0.2.254  1 days 17 hours 44 minutes 22 seconds
8042  SMF   192.0.2.255  192.0.2.252  1 days 17 hours 44 minutes 17 seconds
8043  SMF   192.0.2.255  192.0.2.252  1 days 17 hours 44 minutes 38 seconds
8044  SMF   192.0.2.255  192.0.2.252  1 days 17 hours 44 minutes 17 seconds
8045  SMF   192.0.2.255  192.0.2.252  1 days 17 hours 44 minutes 17 seconds
8046  SMF   192.0.2.255  192.0.2.252  1 days 17 hours 44 minutes 17 seconds
8047  SMF   192.0.2.255  192.0.2.252  1 days 17 hours 44 minutes 17 seconds
```

Viewing the Discovered Endpoint

This section describes how to view the discovered endpoint details such as the NF type, IP address, and the port number of the endpoint.

Based on the endpoint that you want to view, use the following configuration:

- To view the discovered endpoints associated to the CHF service, use the following configuration:

  show rest-endpoint discover-cache chf

  The output of this command displays the NF type, NF instance ID, NF status, IPv4 address, port number, and the expiry date of the CHF's discovered profiles.

  The following is a sample output of the **show rest-endpoint discover-cache chf** command.

```
pcf# show rest-endpoint discover-cache chf
NF Type  NF INSTANCE ID                        NF STATUS   IPV4 ADDRESS  PORT    EXPIRY
CHF     6DDF833D6-b0c9-5503-9800=e806cff43941 Registered  192.0.2.18  4001.0  0.0
CHF     6DDF833D6-b0c9-5503-9800=e806cff43943 Registered  192.0.2.19  4003.0  0.0
CHF     6DDF833D6-b0c9-5503-9800=e806cff43942 Registered  192.0.2.20  4002.0  0.0
```

Note The NF Status indicates the discovered NFs registration status with NRF.

- To view the discovered endpoints associated to the UDR service, use the following configuration:

  show rest-endpoint discover-cache udr

  The output of this command displays the NF type, NF instance ID, NF status, IPv4 address, port number, and the expiry date of the UDR's discovered profiles.

  The following is a sample output of the **show rest-endpoint discover-cache udr** command.

```
pcf# show rest-endpoint discover-cache udr
NF Type  NF INSTANCE ID                        NF STATUS   IPV4 ADDRESS  PORT    EXPIRY
UDR      6DDF833D6-b0c9-5503-9800=e806cff43941 Registered  192.0.2.21  2001.0  0.0
UDR      6DDF833D6-b0c9-5503-9800=e806cff43943 Registered  192.0.2.22  2003.0  0.0
UDR      6DDF833D6-b0c9-5503-9800=e806cff43942 Registered  192.0.2.12  2002.0  0.0
```

- To view the consolidated list of endpoints that PCF has discovered, use the following configuration:

  show rest-endpoint discover-cache

  The output of this command displays the NF type, NF instance ID, NF status, IPv4 address, port number, and the expiry date of both, CHF and UDR discovered profiles.

  The following is a sample output of the **show rest-endpoint discover-cache** command.

```
pcf# show rest-endpoint discover-cache chf
NF Type  NF INSTANCE ID                        NF STATUS   IPV4 ADDRESS  PORT    EXPIRY
CHF      6DDF833D6-b0c9-5503-9800=e806cff43941 Registered  192.0.2.18  4001.0  0.0
CHF      6DDF833D6-b0c9-5503-9800=e806cff43943 Registered  192.0.2.19  4003.0  0.0
CHF      6DDF833D6-b0c9-5503-9800=e806cff43942 Registered  192.0.2.20  4002.0  0.0

pcf# show rest-endpoint discover-cache udr
NF Type  NF INSTANCE ID                        NF STATUS   IPV4 ADDRESS  PORT    EXPIRY
UDR      6DDF833D6-b0c9-5503-9800=e806cff43941 Registered  192.0.2.21  2001.0  0.0
UDR      6DDF833D6-b0c9-5503-9800=e806cff43943 Registered  192.0.2.22  2003.0  0.0
UDR      6DDF833D6-b0c9-5503-9800=e806cff43942 Registered  192.0.2.12  2002.0  0.0
```

Fetching the Subscriber Sessions

This section describes how to fetch the subscriber sessions that are configured across regions.

- Prerequisites for Fetching Subscriber Sessions
- Viewing the Subscriber Session Details

Prerequisites for Fetching Subscriber Sessions

This section describes the prerequisites configuration that you must configure before fetching the subscriber session data across the configured regions.

The prerequisite configuration involves the following step:

- Configuring the Configuration File, on page 333

Configuring the Configuration File

This section describes how to configure the cluster name, external IP address, and port number of the unified API service in the configuration file.

The PCF configuration file determines the application servers and their associated configurations.

To configure the cluster and external IP address for the unified API service, use the following configuration:

```
config
  deployment add config
    cluster-name cluster_name
    unified-api-external-ip external_ip
    port port_number
    end
```

NOTES:

- **cluster-name** *cluster_name*—Specify the cluster name where you want to deploy PCF.
- **unified-api-external-ip** *external_ip*—Specify the IP address on which the unified API listens.
- **port** *port_number*—Specify the port number on which the unified API is exposed.

Verifying the Contents of the Configuration File

This section describes how to verify the contents of the configuration file.

Use the **deployment show-config** command to view the deployment details.

The following is a sample output of the **deployment show-config** command.

```
pcf# deployment show-config
deploymentInfo:
west-coast, 192.0.2.18, http,9090
east-coast, 192.0.2.22, http,9090
```

Deleting the Configuration File

This section describes how to delete the configuration file.

You can delete a configuration file when you no longer require any of the configured resources and services.

To delete the configuration file that is currently configured in your deployment environment, use the following configuration:

```
deployment remove-config
```

Viewing the Subscriber Session Details

This section describes how to view the subscriber session information about the sessions configured.

Before proceeding with the configuration, make sure that you have configured the cluster and external IP for the unified API. For more information, see Configuring the Configuration File, on page 333.

- To view the sessions available on all the configured regions, use the following configuration:

  ```
  show subscriber [ imsi imsi_value | msisdn msisdn_value]
  ```

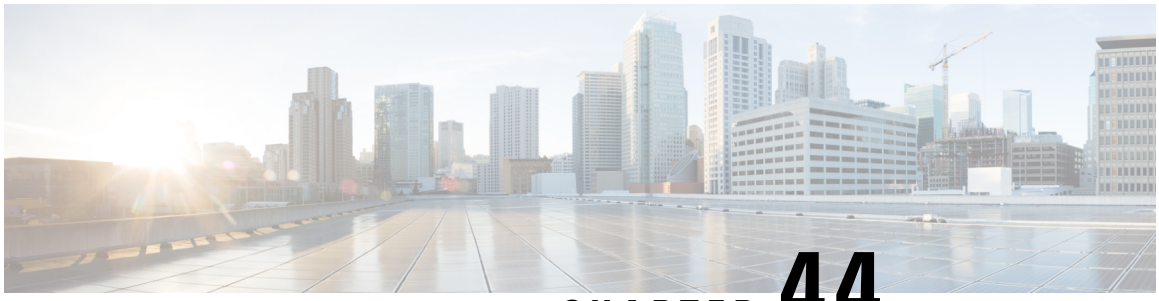
  NOTES:

- *imsi_value*—Displays the subscriber session associated to the specified IMSI value.
- *msisdn_value*—Displays the subscriber session associated to the specified MSISDN value.

The output of this command displays session details and the interfaces that are involved.

```
pcf# show subscriber imsi 100100222233266
SessionInfo:

+-------------------------------------------------------------------------------------+
| 1: development-session      : ism.3.imsi-10012121212123.133131313.1222.68881149 |
+-------------------------------------------------------------------------------------+
| Activity Timestamp                                                                  |
| ------------------------------------------------------------------------------------|
| Connect Time                : 06-03-2020 04:39:43 AM      |
| Expiration Time             : 29-03-2020 04:39:43 PM remaining: 23 days, 20:04.0|
|                                                                                     |
| Session Detail                                                                      |
| ------------------------------------------------------------------------------------|
|  Bearer Session Type        : STATIC_5G                                             |
|   External Profile          : No                                                    |
|-------------------------------------------------------------------------------------|
| LDAP                                                                                |
|-------------------------------------------------------------------------------------|
|   External Profile          : No                                                    |
|                                                                                     |
| N7                                                                                  |
|-------------------------------------------------------------------------------------|
| mcc,mnc                     : 100, 010                                              |
| dnn                         : static.one.5g                                         |
| rat-type                    : NR                                                    |
| access-type                 : 3GPP_Access                                           |
| pdu-session-id              : 5                                                     |
| supi                        : imsi-100101222233266                                  |
| gpsi                        : msisdn-11112433266                                    |
| framed-ip-v4                : 31.31.154.88                                          |
| framed-ip-v6                : 2710:ae00:d2f3:9a78                                   |
| update-notify-url           : http://192.0.2.12:7010/callbacks/v1/           |
|                               smPoliciesUpdateNotification/i msi-10001011111:5  |
```



CHAPTER 44

UDR Interface

Feature Summary and Revision History

Summary Data

Table 149: Summary Data

Applicable Product(s) or Functional Area	PCF
Applicable Platform(s)	SMI
Feature Default Setting	Enabled – Configuration required to disable
Related Documentation	Not Applicable

Revision History

Table 150: Revision History

Revision Details	Release
Enhancement introduced. Introduced changes related to the Nudr_DataRepository service.	2020.02.0
First introduced.	2020.01.0

Feature Description

The PCF interfaces with the User Data Repository (UDR) to receive subscriber-related policies for User Equipment (UE) attach and session establishment. When a UE attaches to the network, AMF requires AM policies of the subscriber from PCF for the UE. Similarly, when a UE makes a PDU Session, SMF requires policy rules from PCF. Subscriber attributes are stored in the UDR. PCF communicates with UDR to receive these attributes which are used in the evaluation of policies.

PCF invokes the Nudr_DataRepository service to retrieve the AM and SM Policy attributes. PCF carries out the discovery of UDR URL through the Nnrf_NFDiscovery service which is NRF service.

API Details

Description	API URI	Request Parameters	Response Parameters
Retrieves the access and mobility policy data for a subscriber.	{apiRoot}/nudr-dr/v1/policy-data/ues/{ueId}/am-data	Path and Query	200 Ok AmPolicyData
Retrieves the session management policy data for a subscriber.	{apiRoot}/nudr-dr/v1/policy-data/ues/{ueId}/sm-data	Path and Query	200 Ok SmPolicyData

Important The UE Id in the API represents the SUPI or GPSI.

Parameter Details

PCF supports the following parameters:

AMPolicy Query Parameters

Table 151: AMPolicy Query Parameters

Parameter Name	Parameter Type	Description
ueId	Path	SUPI or GPSI

AmPolicyData

Table 152: AmPolicyData

Parameter Name	Parameter Type	Description
ueId	Path	SUPI or GPSI
AmPolicyData	-	List of category identifiers associated with the subscriber.

SmPolicy Query Parameters

Table 153: SmPolicy Query Parameters

Parameter Name	Parameter Type	Description
ueId	Path	SUPI or GPSI
Snssai	Query	Identifies single network slice selection assistance information.
Dnn	Query	Identifies a Data Network Name.
Fields	Query	Attributes to be received.

SmPolicyData

Before configuring the SmPolicyData parameters, make sure that in the Policy Builder the policy subscriber AVP field has the following keys:

- allowedServices
- subscCats

Table 154: smPolicySnssaiData Object

Parameter Name	Description
smPolicySnssaiData	The Session Management Policy data per S-NSSAI for all the SNSSAIs of the subscriber. The key of the map is the S-NSSAI. **Note** The SmPolicySnssaiData parameter contains the Snssai and SmPolicyDnnData objects.

Table 155: Snssai Object

Parameter Name	Description
Snssai	Identifies a single network slice selection assistance information.

Table 156: SmPolicyDnnData objects

Parameter Name	Description
Dnn	Identifies the Data Network Name.
allowedServices	List of subscriber’s allowed service identifiers.
subscCats	List of categories associated with the subscriber.
gbrUl	Maximum aggregate UL bitrate that is provided across all GBR QoS Flows in the DNN.

Parameter Name	Description
gbrDl	Maximum aggregate DL bitrate that is provided across all GBR QoS Flows in the DNN.
adcSupport	Indicates whether application detection and control that is enabled for a subscriber.
subscSpendinglimits	Indicates whether PCF must enforce policies that are based on subscriber spending limits.
ipv4Index	Information that identifies which IP pool or external server that is used to allocate the IPv4 address.
ipv6Index	Information that identifies which IP pool or external server that is used to allocate the IPv6 address.
Offline	Indicates that the offline charging is applicable to the PDU session.
Online	Indicates that the online charging is applicable to the PDU session.
chfInfo	Address of the charging function.
usageMonDataLimits	Contains a list of usage monitoring profiles that are associated with the subscriber. The monitoring key that is used as the key in the map.
usageMonData	Contains the remaining allowed usage data that are associated with the subscriber. The monitoring key that is used as the key in the map.
mpsPriority	Indicates subscription to the MPS priority service. Priority applies to all traffic on the PDU Session.
imsSignallingprio	Indicates subscription to the IMS signaling priority service. Priority only applies to IMS signaling traffic.
mpsPrioritylevel	Relative priority level for the multimedia priority services.
smPolicySnssaiData	The Session Management Policy data per S-NSSAI for all the SNSSAIs of the subscriber. The key of the map is the S-NSSAI.

Note The PCF does not support the UDR-N36 response data fields such as umDataLimits and umData.

How it Works

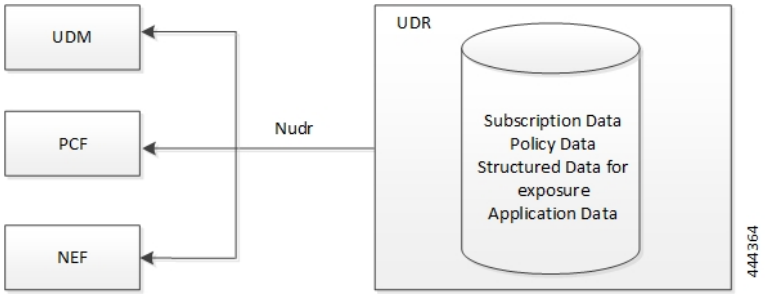
This section describes how this feature works.

The User Data Repository (UDR) provisions PCF to retrieve the data stored in the UDR through the Nudr_DataRepository service. The service is also responsible for enabling the NF to subscribe and unsubscribe to the data change notifications from UDR. In particular to PCF, the Nudr_DataRepository service provides the following retrieve services to access policy control-related subscription information and application-specific information that is stored in the UDR:

- Subscription to notifications from the UDR on changes in the policy control-related subscription information.
- Subscription to the UDR for the AF requests targeting a DNN and S-NSSAI or a group of UEs (roaming UEs for LBO case) that are identified by an Internal Group Identifier.
- Subscription to notifications from the UDR on the update of AF requests targeting a DNN and S-NSSAI or a group of UEs (roaming UEs for LBO case) identified by an Internal Group Identifier.

The N36 reference point is defined for the interactions between PCF and UDR in the following reference point representation.

Figure 57: N36 Reference Point



Call Flows

This section describes the key call flows for this feature.

AM Policy Subscription Call Flow

This section describes the AM Policy Subscription call flow .

Figure 58: AM Policy Subscription Call Flow

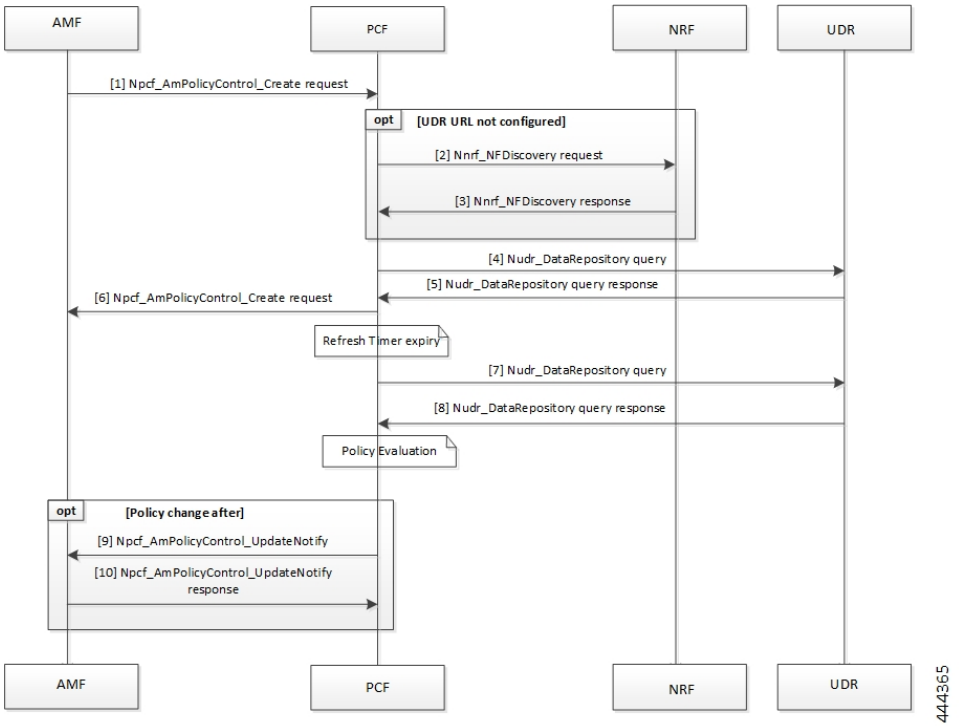


Table 157: AM Policy Subscription Call Flow Description

Step	Description
1	The AMF sends a Npcf_AmPolicyControl_Create request to the PCF.
2	If the UDR URL is not configured, the PCF sends the Nnrf_NFDiscovery request is sent to NRF.
3	In response, the NRF sends the Nnrf_NFDiscovery results to the PCF.
4	The PCF sends the Nudr_DataRespository query to the UDR.
5	In response, the UDR sends the repository details to the PCF.
6	The PCF sends the Npcf_AmPolicyControl_Create request to the AMF.
7	The PCF sends the Nudr_DataRepository query to the UDR.
8	The UDR responds to PCF with the repository information.
9	After PCF evaluates the response, if the policy has modified, then the PCF sends the Npcf_AmPolicyControl_UpdateNotify request to the AMF.
10	The AMF sends the Update Notification as a response to the PCF.

SM Policy Subscription Call Flow

This section describes the SM Policy Subscription call flow.

Figure 59: SM Policy Subscription Call Flow

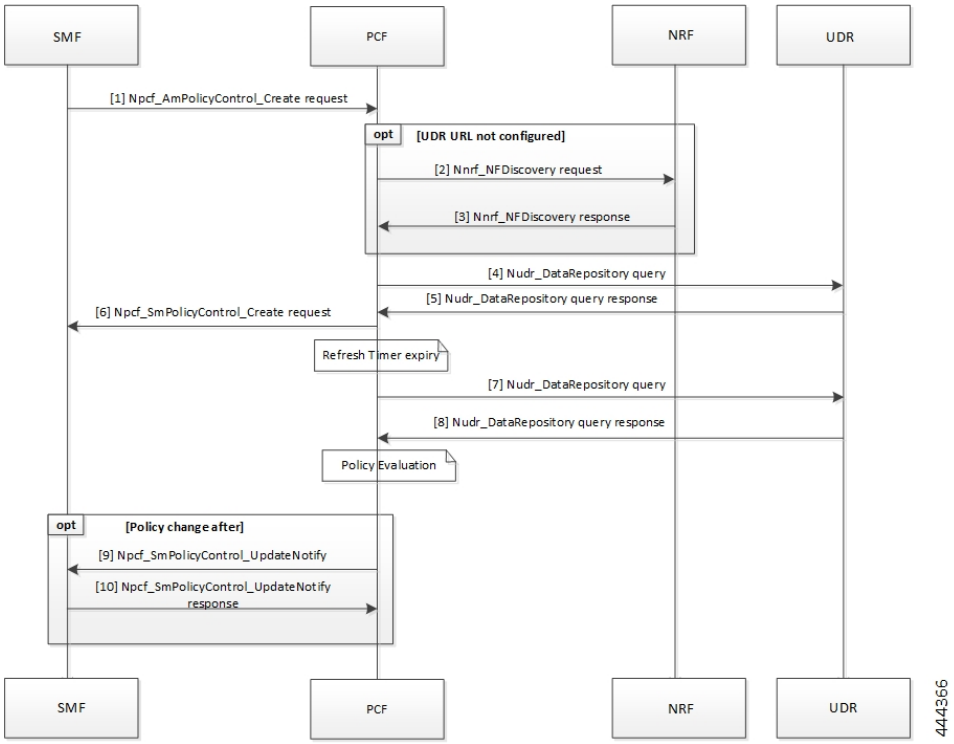


Table 158: SM Policy Subscription Call Flow Description

Step	Description
1	The SMF sends a Npcf_SmPolicyControl_Create request to the PCF.
2	If the UDR URL is not configured, the PCF sends the Nnrf_NFDiscovery request to NRF.
3	In response, the NRF sends the Nnrf_NFDiscovery results to the PCF.
4	The PCF sends the Nudr_DataRespository query to the UDR.
5	In response, the UDR sends the repository details to the PCF.
6	The PCF sends the Npcf_SmPolicyControl_Create request to the SMF.
7	The PCF sends the Nudr_DataRepository query to the UDR.
8	The UDR responds to PCF with the repository information.
9	After PCF evaluates the response, if the policy has modified, then the PCF sends the Npcf_SmPolicyControl_UpdateNotify request to the SMF.
10	The SMF sends the Update Notification as a response to the PCF.

Configuring the UDR Base URL

This section describes how to configure the UDR base URL discovery.

You must configure the UDR base URL for discovering profiles. The base URL aids in navigating to the mapped UDR. In the absence of the UDR base URL, the NRF's base URL is queried for the UDR base URL on the discovered profiles.

To configure the UDR base URL, use the following configuration in the Policy Ops Center console:

```
config
  nrfDiscovery
    nfType
      locality
        client client_name
        geoServer server_name
        preferredServer server_name
      nrfDiscoveryGroup
        subscriptionEnabled [ true | false ]
        subscriptionExtension extension_count
        type nf_type
        end
```

NOTES:

- **nrfDiscovery**—Enters the NRF discovery configuration mode.
- **locality**—Enters the locality configuration mode. The PCF REST endpoint considers the locality configuration.
- **client** *client_locality*—Specify the client locality of used by the NRF endpoint.
- **geoServer** *server_name*—Specify the geo redundant site of the preferred locality.
- **preferredServer** *preferred_locality* —Specify the server that is configured as the preferred server. Preferred locality takes precedence over the geo locality while using the service of the discovered NF (UDR).
- **nrfDiscoveryGroup**—Enters the NRF discovery group configuration.
- **subscriptionEnabled [true | false]** —Configures the subscription capability.
- **subscriptionExtension** *extension_count*—Specify the count for which the subscription can be extented.
- **type** *nf_type*—Specify the NF type. For this configuration, it should be UDR.

Standards Compliance

This feature complies with the following standards specifications:

- *3GPP TS 23.501 "System Architecture for the 5G System "*
- *3GPP TS 23.502 "Procedures for the 5G System (5GS)"*

- *3GPP TS 23.503 "Policy and charging control framework for the 5G System (5GS)"*
- *3GPP TS 29.508 "Session Management Event Exposure Service"*
- *3GPP TS 29.512 "Session Management Policy Control Service"*
- *3GPP TS 29.513 "Policy and Charging Control signalling flows and QoS parameter mapping"*
- *3GPP TS 29.519 V15.4.0 "Usage of the Unified Data Repository Service for Policy Data, Application Data and Structured Data for Exposure"*

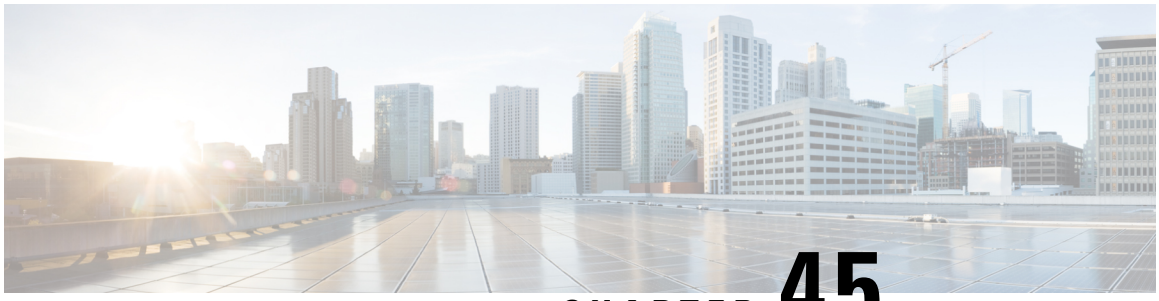
Filtering the Profile Data

PCF queries the subscriber attributes that are mapped in additional profiles from UDR for the N7 interface. Based on the UDR response, the attributes that are returned are mapped to the subscriber attribute. The allowed services that are returned can be mapped to the services associated with the subscriber.

For information on how to configure the additional profile data, see Setting Up Additional Profile Data, on page 137.

In Policy Builder, you can access the following filters under **Domain > Additional Profile Data > Filter**:

- SUPI
- GPSI



CHAPTER 45

Update Requests Toward CHF

Feature Summary and Revision History

Summary Data

Table 159: Summary Data

Applicable Product(s) or Functional Area	PCF
Applicable Platform(s)	SMI
Feature Default Setting	Enabled – Configuration required to disable
Related Documentation	Not Applicable

Revision History

Table 160: Revision History

Revision Details	Release
First introduced.	2020.01.0

Feature Description

The 4G CPS PCRF supports LDAP notification. In 5G, this support is available to PCF by using the same LDAP plugins which were deployed for 4G. PCF supports the following features:

- Requery Subscriber Profile Repository (SPR) on receiving the LDAP notification.
- Reevaluate the subscriber policies after receiving LDAP response.
- Based on the action derived from the CRD table, the corresponding action is performed over the N28 interface. The corresponding actions are to start, continue, update, reinitiate, and to terminate an N28 session.

How it Works

This section describes how this feature works.

The SMF sends the N7 Create Request to PCF. PCF then sends the query to LDAP to find the subscriber profile. The LDAP notifications from the LDAP client are supported in the following way:

1. For any changes in a subscriber profile, PCF receives an LDAP notification. PCF then sends a requery to find the updated subscriber profile.
2. On receiving the LDAP notification, PCF sends the N28 Subscribe Update on the Charging Function (CHF). PCF sends this update to receive the updated status of policy counters for the policy that the subscriber has subscribed.
3. After reevaluating the counters that PCF receives from CHF over the N28 interface, if policy has changed, then SMF is updated with the Update Notify message over the N7 interface.

Standards Compliance

This feature complies with the following standards specifications:

- *3GPP TS 29.510 V15.2.0 (2018-12) "Network Function Repository Services"*

Configuration Support for Setting up the Update Requests

Setting up the update requests toward CHF involves configuring the following services:

1. TableDrivenActionOverN28
2. SpendingLimitSubscription

TableDrivenActionOverN28

This section describes the parameters that are required to configure the TableDrivenActionOverN28 service.

The TableDrivenActionOverN28 service configuration evaluates and retrieves action on the received messages. During the policy evaluation, if the TableDrivenActionOverN28 service exists in policy, PCF evaluates the CRD table and determines the action for the implementation on the N28 interface.

Based on the action value, PCF performs the following tasks:

- If the action value is Update and if the N28 session exists, PCF initiates the Intermediate Spending Limit Report operation.
- On receiving the Subscription-Update response or timeout, PCF reevaluates the TableDrivenActionOverN28 configuration and determines the next action.
- If the action value is Reinitiate, PCF terminates the existing session and initiates a new session by sending a subscription request. After PCF terminates the existing session, sync N28 Unsubscribe is sent and the session gets deleted.

The TableDrivenActionOverN28 service configuration accepts only request message attributes for Input column binding. Hence, the attribute value pair (AVP) code must match the JSON path that is received in the message. In addition to message attributes, the following AVP codes are used as input AVP codes:

- Command-Code—The command code of request message.
- Application-Id—The Diameter Application Identifier to send the Diameter message.

Before setting the service parameters, ensure that you create a use case template and add a service for this configuration. For details, see Configuring the Use Case Template, on page 293 and Adding a Service, on page 292.

The following table describes the TableDrivenActionOverN28 service parameters.

Note Select the N28 service configuration object to configure this service.

Table 161: TableDrivenActionOverN28 Parameters

Parameters	Description
Search Table Group	Enter the table group that you want to search from the custom reference data table. **Note** The STG and the included CRD table can have key columns that may not refer to the message attributes in the inbound message. For example, output of other tables. Therefore, ensure to bind these columns correctly. • Input Column Binding—Under this parameter, select a value for the ColumnAndAvpPair parameter. • AVP Name—Select the name of the Request Attribute for input for evaluating the custom reference data table. • Column—Select the key column in the Search Table Group corresponding to the specified AVP.

Parameters	Description
Output Column Binding	Under this parameter, select a value for the ColumnAndAvpPair parameter. • Avp Name—Select the name of the Request Attribute to which the value of the output column is mapped. • Column—Select the CRD table column for the output AVP. **Note** The list of Output column to AVP bindings supports only one column or AVP binding for fixed AVP code Action-Over-N28.

SpendingLimitSubscription

This section describes the parameters that are required to configure the SpendingLimitSubscription service.

The SpendingLimitSubscription service configuration object is added for the N28 interface. With this configuration, you can perform the following tasks:

- Request and subscribe the policy counter status reporting from PCF to CHF.
- Unsubscribe from spending limit reports.
- Receive notifications of spending limit reports from CHF to PCF.

Note Select the N28 service configuration object to configure the SpendingLimitSubscription service.

Before setting the service parameters, ensure that you create a use case template and add a service for this configuration. For details, see Configuring the Use Case Template, on page 293 and Adding a Service, on page 292.

The following table describes the SpendingLimitSubscription service parameters.

Table 162: SpendingLimitSubscription Parameters

Parameter	Description
Supi	Pulls the value from the session and policy state retrievers, which are mapped to the N7 Subscription Permanent Identifier (SUPI) object. **Note** The default values of SUPI and GPSI are not configured and their values are always dynamically pulled from the session and policy state retrievers.
Gpsi	Pulls the value from the session and policy state retrievers, which are mapped to the N7 Generic Public Subscription Identifier (GPSI) object. **Note** The default values of SUPI and GPSI are not configurable, and their values are always dynamically pulled from the session and policy state retrievers.

Parameter	Description
Defaults On Failure	This is an optional parameter. • Under this parameter, configure DefaultSpendingLimitReport, which specifies the default list of Policy Counter Identifiers that are subscribed for failures. Configure the following parameters: • Failure Reason—From the drop-down list, select failure code for the default identifier. • Identifier—Enter the name for the policy counter identifier. • Status—Select the status of the policy counter identifier.

Use Case Template Actions

This section describes the actions that you can perform for the use case templates.

The N28 TGPP session exists condition indicates that a valid N28 session exists for the use case to become true.

The following table describes the condition input variables that you can configure in Policy Builder:

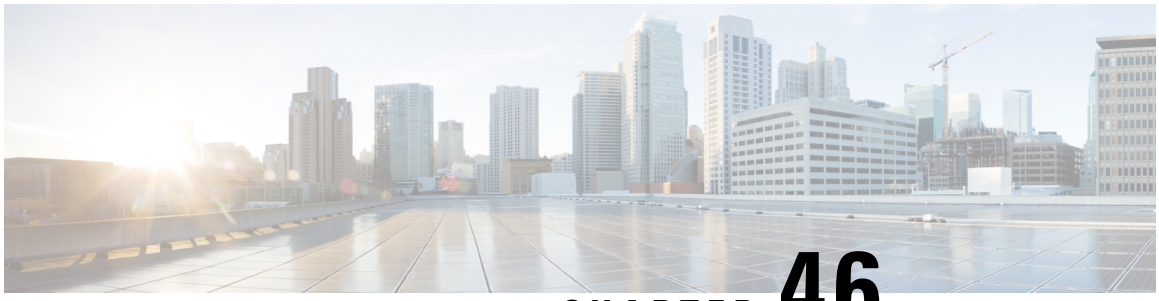
Table 163: Conditions and the AVP Descriptions

Condition Input Variable	AVP Used and Description
failureReason	Reason for failure when the N28 session is not established due to an error.
lastSubscriptionType	Last subscription request type, which can be Initial or Update.
failureReason	Failure reason of the last N28 session.
syCountersIdentifierAndStatus	N28 counter identifier and status.
subscriptionId	Unique ID of the N28 session.
Connected	Boolean value to indicate whether the N28 connection is established or not.

Troubleshooting Information

For message routing failures, check the datastore pod health and the logs for any issues.

For more information on how to check the pod health and logs, see Troubleshooting Information, on page 385.



CHAPTER 46

VoNR through the Rx Interface

Feature Summary and Revision History

Summary Data

Table 164: Summary Data

Applicable Products or Functional Area	PCF
Applicable Platform(s)	SMI
Feature Default Setting	Enabled – Configuration required to disable
Related Documentation	Not Applicable

Revision History

Table 165: Revision History

Revision Details	Release
Enhancement introduced. Revised the content to include the new and updated call flows that reflects support for the PCF to PCRF interaction which encourages the VoLTE calls. Also, added procedure to configure the gRPC interface.	2020.01.0
First introduced.	Pre 2020.01.0

Feature Description

PCF supports the full Diameter stack along with the standard Diameter interfaces like Rx. With this support, you can configure PCF to handle VoLTE calls for 4G and VoNR for 5G. In 4G scenarios, PCF serves as a proxy and performs the session binding lookups to re-route the 4G calls to the PCRF instances for processing. PCF supports the PCRF GR active/active mode for routing messages to the secondary PCRF site when the primary site is unavailable.

Prerequisites

To enable the interaction between PCF and PCRF for VoLTE/VoNR calls, ensure to perform the following on PCRF:

- Install the following features by adding them to the `/etc/broadhop/pcrf/features` file:
  - `com.cisco.bindingdb.feature`
  - `com.broadhop.diameter2.local.cnat.feature`
- On all the QNS nodes, ensure to add the Diameter endpoint gRPC service names in the `/etc/hosts` file.
- Include the etcd IPs in the `qns.conf: -Ddiameter.registry.etcd.hosts=`*`comma separated etcd-external IP`* file.
- Make sure that the routable network connectivity exists between the node where the PCF Diameter endpoint is hosted, the QNS engine VMs for site-local, and the remote (in GR mode) communication (over HTTP2).
- The Rx peer must be connected to both the local and remote PCF Diameter endpoints simultaneously with one or both the connections as active. For the active Rx connection from AF to PCF, the AF peer handles and receives the Rx requests from other PCF instance without any connectivity issues.
- For the PCRF and PCF components to communicate, the configured port numbers must be open on the firewall.

How it Works

This section describes how this feature works.

Call Flows

This section describes the following call flows.

Session Create, Update, and Terminate Call Flow

This section describes the call flow to create, update, or terminate a session.

Figure 60: Session Creation, Update, and Termination Call Flow

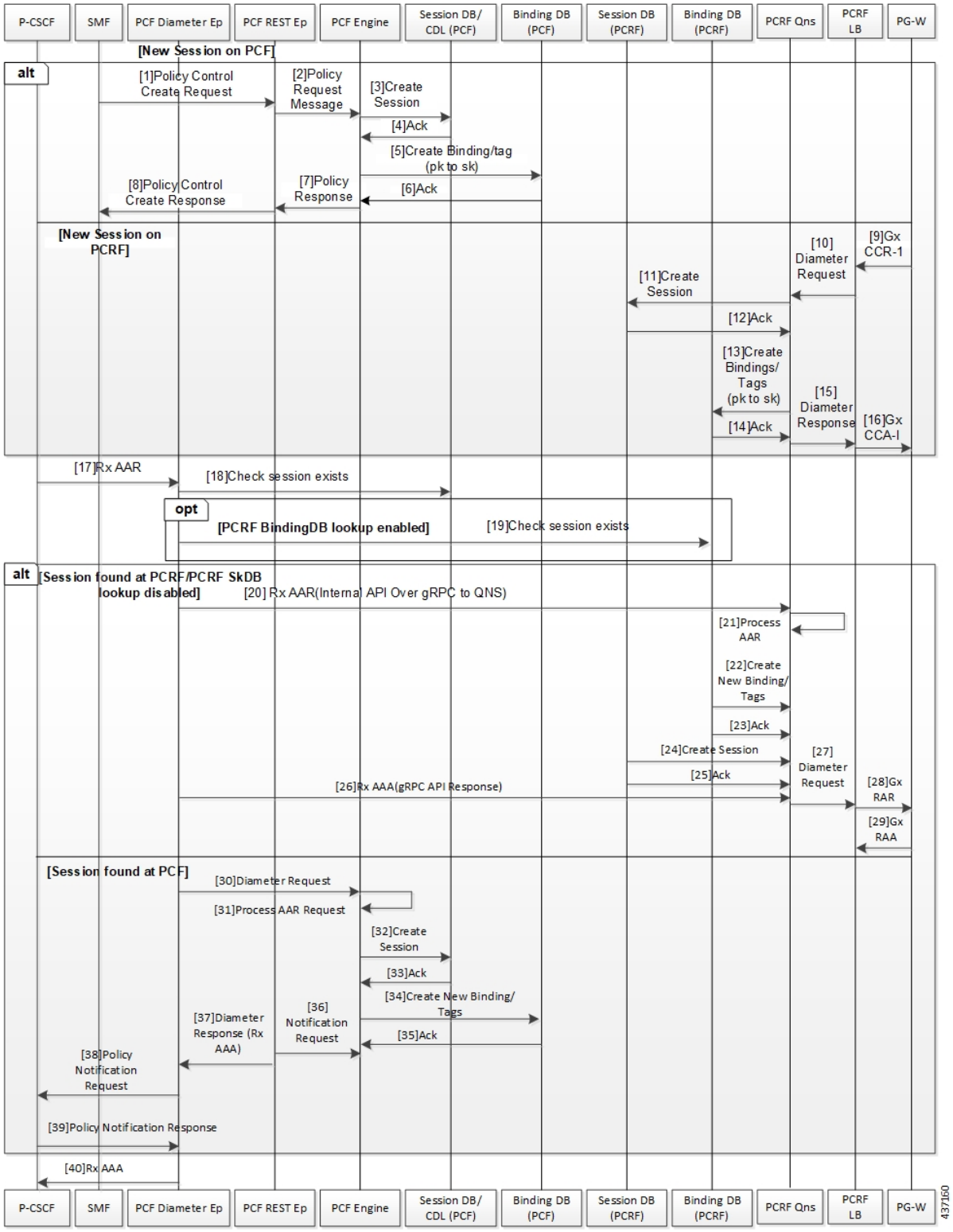


Table 166: Session Creation, Update, and Termination Call Flow Description

Step	Description
1	The SMF sends a Policy Control Create request to the PCF REST endpoint.
2	The PCF REST endpoint forwards the request to the PCF Engine.
3	The PCF Engine sends a Create Session request to Session DB/CDL.
4	The Session DB/CDL sends an acknowledgement to the PCF Engine for the Create Session request.
5	The PCF performs a binding or tagging the database and sends it to the Binding DB.
6	The PCF confirms that the binding or tagging of the database is complete by sending an acknowledgement to the PCF Engine.
7	The PCF Engine sends Policy response to the PCF REST endpoint.
8	The PCF REST endpoint creates a response and sends it to the SMF.
9	A new session is created on PCRF. The P-GW sends Gx CCR request to the PCRF LB.
10	The PCRF LB sends the Diameter request to the PCRF QNS.
11	The PCRF QNS creates a session in the session database and sends it to PCRF.
12	The session database sends an acknowledgement to the PCRF indicating that the session is created.
13	The PCRF QNS performs the binding and notifies the PCRF.
14	The PCRF sends an acknowledgement to the PCRF QNS indicating that the binding is created.
15	The PCRF QNS sends Diameter response to the PCRF LB.
16	The PCRF LB sends Gx CCA request to the P-GW.
17	The P-CSCF sends Rx AAR request to the PCF Diameter endpoint.
18	The PCF Diameter endpoint checks if the session exists on PCF.
19	If the PCRF BindingDB lookup is enabled, the PCF Diameter endpoint checks whether the session exists on PCRF.
20	In the existing session of the PCRF SkDB instance with lookup enabled, the PCF Diameter endpoint sends the Rx AAR request over the gRPC interface to the PCRF QNS.
21	The PCRF QNS processes the AAR request.
22	The PCRF QNS binds or tags the database and sends an acknowledgement to PCRF.
23	The PCRF acknowledges the binding or tagging of the database and notifies the PCRF QNS.
24	The PCRF QNS sends Create Session request to the PCRF.
25	The PCRF sends an acknowledgement for the request to the PCRF QNS.

Step	Description
26	The PCRF QNS sends Rx AAR request to the PCRF QNS.
27	The PCRF QNS sends Diameter request to the PCRF LB.
28	The PCRF LB sends Gx RAR request to the P-GW.
29	The P-GW sends the Gx RAA response to PCRF LB.
30	In the existing session of PCF, the PCF Diameter endpoint sends Diameter request to the PCF REST endpoint.
31	The PCF Engine processes the AAR request.
32	The PCF Engine sends a Create Session request to Session DB/CDL.
33	The Session DB/CDL sends an acknowledgement to PCF Engine.
34	PCF sends a Create New Binding/Tag request to the Binding DB.
35	The Binding DB sends an acknowledgement to PCF.
36	The PCF Engine sends Notification request to the PCF REST endpoint.
37	The PCF REST endpoint sends a Diameter response request to the PCF Diameter endpoint.
38	The PCF Diameter endpoint sends Policy Notification Request to the P-CSCF.
39	The P-CSCF sends the Policy Notification response to the PCF Diameter endpoint.
40	The PCF Diameter endpoint sends the Rx AAA request to the P-CSCF.

Binding Database Query Failures Call Flow

This section describes the PCF failover for the binding database query call flow.

Figure 61: PCF Failover for Binding Database Query Call Flow

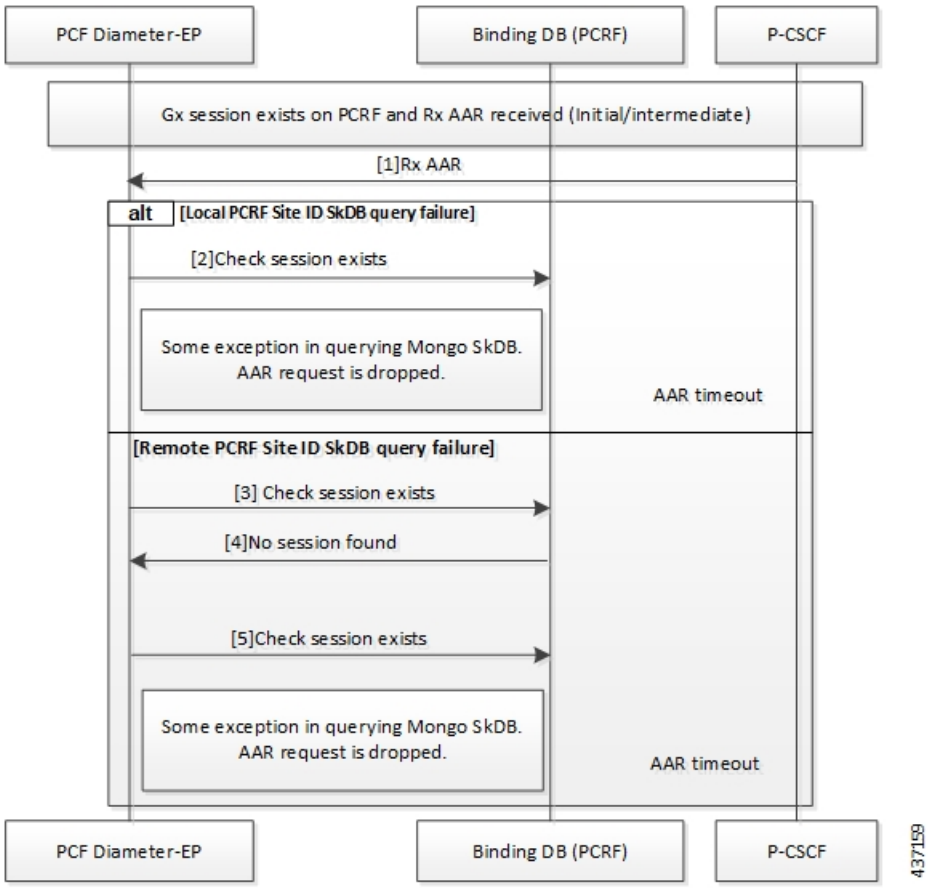


Table 167: PCF Failover for Binding Database Query Call Flow Description

Step	Description
1	When a Gx session exists on PCRF and Rx AAA response is received, P-CSCF sends the Rx AAA request to the PCF Diameter endpoint.
2	For a local PCRF failure, the PCF Diameter endpoint checks if the session exists on local PCRF.
3	If the PCRF query times out due to the local PCRF failure or network issue, PCF reattempts sending the query to the remote PCRF instance.
4	If the session is not found, PCRF notifies the PCF Diameter endpoint.
5	If the PCF Diameter endpoint query times out due to the local PCF or PCRF failure or network issue, the PCF Diameter endpoint reattempts to check if the session exists in the local PCRF.

Binding Database Query Call Flow

This section describes the PCF Rx rerouting and PCRF binding database call flow.

Figure 62: PCF Rx Rerouting and PCRF Binding Database Call Flow

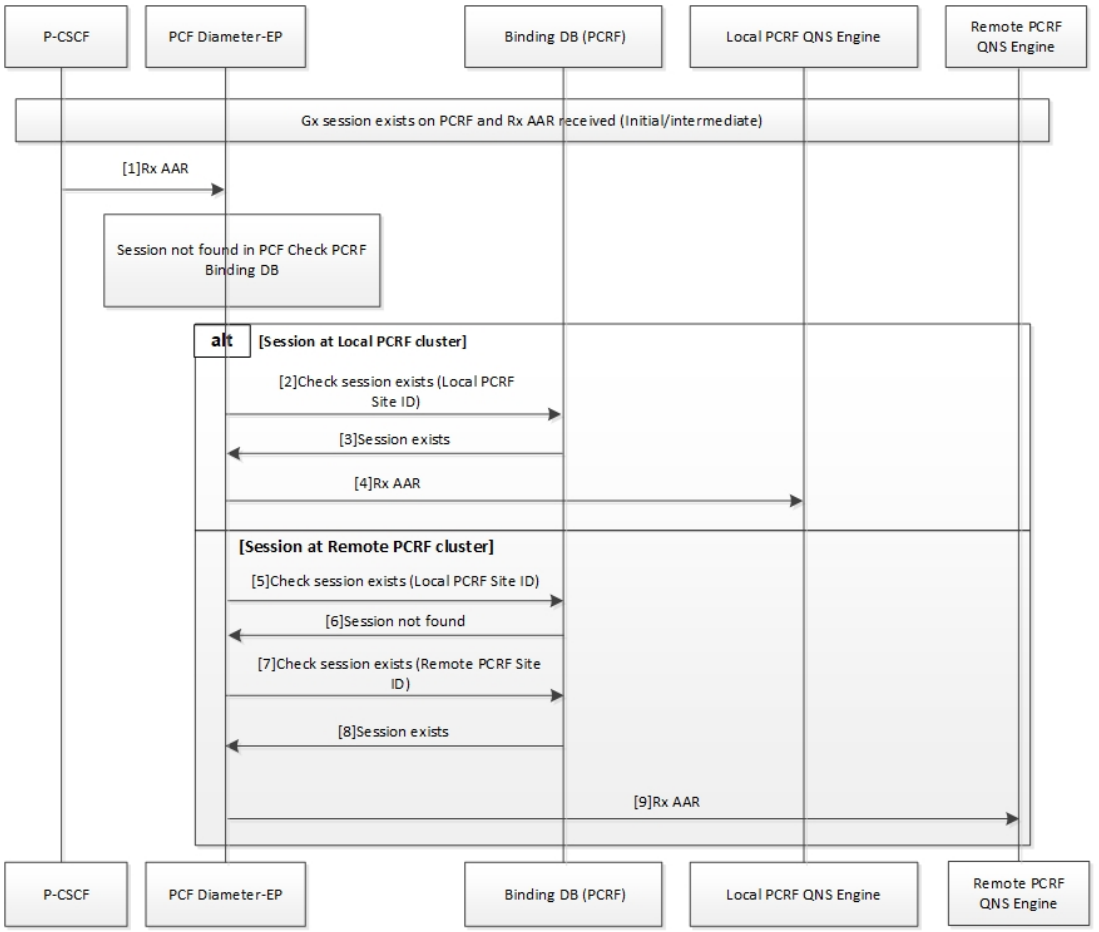


Table 168: PCF Rx Rerouting and PCRF Binding Database Call Flow Description

Step	Description
1	When a Gx session exists on the PCRF and the local PCF has received the original AAR Initialization message, the P-CSCF sends the Rx AAR request to the local PCF Diameter endpoint.
2	The PCF Diameter endpoint performs a parallel local data store lookup and a binding database query toward the site local PCRF SkDB instance. The PCF Diameter endpoint checks if the session exists on the PCRF.
3	The PCRF notifies the PCF Diameter endpoint that the session exists.
4	The PCF Diameter endpoint forwards the Rx AAR request to the PCRF QNS engine.
5	If the PCF Diameter endpoint query times out due to the local PCRF failure or network issue, the PCF reattempts sending the query to the local PCRF instance.
6	If the session is not found, then the PCRF notifies the PCF Diameter endpoint.
7	The PCF Diameter endpoint reattempts to check if the session exists in the remote PCRF instance.

Step	Description
8	If the session is found, the PCRF notifies the PCF Diameter endpoint that the session exists.
9	The PCF Diameter endpoint sends the Rx AAA response to the QNS Engine.

PCF Failover Call Flow

This section describes the PCF failover call flow.

Figure 63: PCF Failover Call Flow

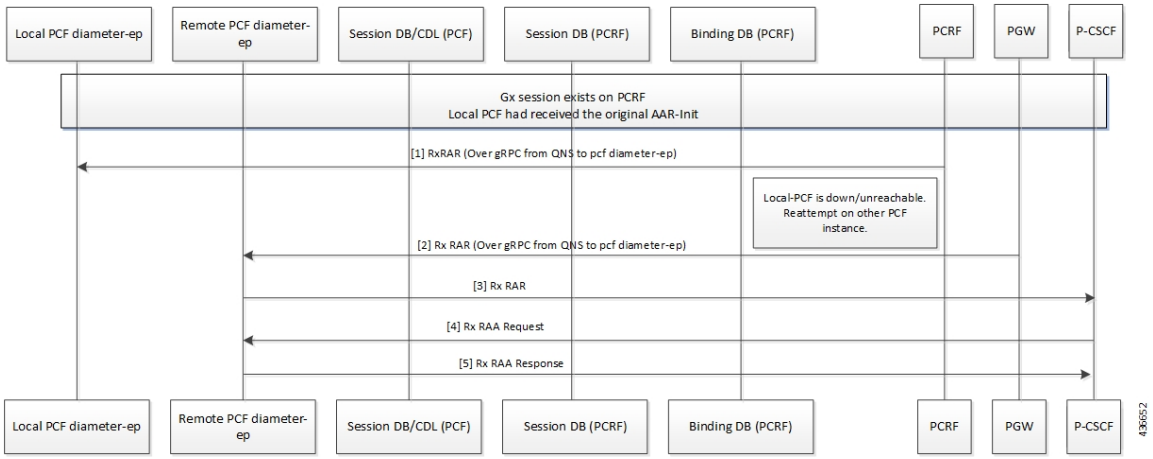


Table 169: PCF Failover Call Flow Description

Step	Description
1	When a Gx session exists on the PCRF and the local PCF has received the original AAR Initialization message, the PCRF sends the Rx RAR request over gRPC to the local PCF Diameter endpoint.
2	If the local PCF is unreachable or is inactive, the PCRF reattempts to send the Rx RAR request over gRPC to the remote PCF Diameter endpoint.
3	If the remote PCF Diameter endpoint is connected, it forwards the Rx RAR request to P-CSCF.
4	The P-CSCF sends the Rx RAA request to the remote PCF Diameter endpoint.
5	The Remote PCF Diameter endpoint sends the Rx RAA response to the PCRF.

Standards Compliance

This feature complies with the following standards specifications:

- *3GPP TS 29.510 V15.2.0 (2018-12) "Network Function Repository Services"*

Limitations

This feature has the following limitations in this release:

- The deployment or configuration of the PCF Binding database (SkDB) Mongo instances is not supported via PCF. The PCF binding database instances are deployed through the PCRF installation with PCF SkDB as separate "sites."
- The Gx interface traffic should not be configured on PCF because it supports only the Diameter Rx interface with the Rx rerouting enabled.
- The remote binding database lookup in PCRF for a session can be enabled only when the existence is configured.
- If the PCRF BindingDB lookup capability is disabled in PCF, then PCF forwards all the requests for which the associated sessions are not found in the PCF CDL database to the local PCRF.

Enabling Interaction Between PCF and PCRF for VoNR Calls

This section describes the configurations that you must perform to enable the interaction between PCF and PCRF.

- Configuring the Interface Between PCF and PCRF

Configuring the Interface Between PCF and PCRF

The gRPC interface is configured on PCRF to accept the Diameter request. You can configure the gRPC interface by adding the **com.broadhop.diameter2.local.cnat.feature** to the `/etc/broadhop/pcrf/features` file.

Note The **com.broadhop.diameter2.local.cnat.feature** capability is an additional requirement to the standard list of features that are required for 4G PCRF to be operational.

To enable the access, configure the following system properties using the existing "properties" CLI which is available through the engine node:

- **diameter grpc channel count** *integer*: Specifies the number of gRPC channels that each diameter-ep replica opens toward a diameter-engine.
- **diameter group** *group* **stack** *stack* **grpc ext-svc ip** *ip-address* **port** *port*: Indicates the externally accessible IP: Port for the diameter-ep service that belongs to the specified group and stack. This property enables the specified diameter-ep to receive incoming gRPC requests from diameter-engines outside the K8 cluster.
- **diameter group** *group* **stack** *stack* **diameter-engine alt-engines primary svc-name** *external-service-name* **port** *port*: Indicates the external-service-name that refers to the service defined under the root level external-services CLI node. The port number corresponds to the port number on which the alternate primary engine is listening (for gRPC requests). This port number must be included in the list of port numbers for the configured external-service.

- **diameter group** *group* **stack** *stack* **diameter-engine alt-engines secondary svc-name** *external-service-name* **port** *port*: Acts as the alternative for the primary diameter-engine that is configured for **diameter group** *group* **stack** *stack* **diameter-engine alt-engines primary svc-name** *external-service-name* **port** *port*. If PCF diameter-ep is unable to send the request to the primary engine, it reattempts sending the request on the secondary diameter-engine.
- **diameter group** *group* **stack** *stack* **diameter-engine alt-engines check-session-exists [true | false]** : Enables verification of the session's existence (using PCRF BindingDB/SkDB) before forwarding the request to the alternate engine. If this property is enabled and the session is not found on the alternate engines, then the message is sent to the local pcf-engine for error handling. By default, the property is configured as False.
- **external-services** *svc-name* **ips** *list of ip addresses* **ports** *list of ports*: Holds the external service definition for services that exist outside the K8 cluster. This property creates an Egress or external K8 service that comprises of the IP addresses/port numbers which allows access to services outside the cluster. For example, alternate Diameter engines.
- **etcd external-ips** *list of IP addresses*: Lists the externally accessible IP addresses for the etcd Diameter endpoint registry that enables access beyond the K8 cluster.

VoNR through Rx Interface OA&M Support

This section describes operations, administration, and maintenance information for this feature.

Statistics

This section provides the list of statistics and counters that enable you to track the flow of messages between PCRF and PCF:

- grpc_message_send_total: Collects the total count of messages that are sent over the gRPC toward the PCF Engine. This metric support the following tag and values:
  - command_code, application, remote_service: Counts the number of Diameter requests sent toward the Diameter Engines.

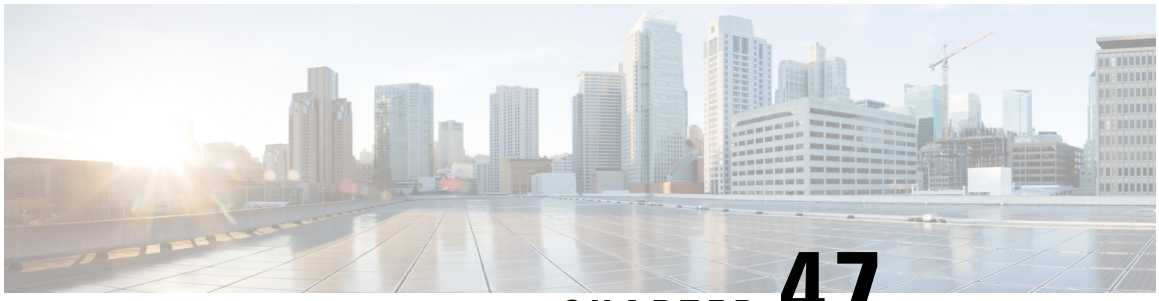
    The `command_code` represents the Diameter command code and the application represents the Diameter application to which the message belongs.

    The `remote_service` corresponds to the name of the Diameter engine service to which the request is forwarded to.

    For the PCF Engine, the metric corresponds to the Diameter Engine value.

    For alternate engines, it corresponds to the name of the external service selected.

For information on statistics, see *Ultra Cloud Core 5G Policy Control Function, Statistics Reference*.



CHAPTER 47

Advanced Tuning Parameters

Feature Summary and Revision History

Summary Data

Table 170: Summary Data

Applicable Product(s) or Functional Area	PCF
Applicable Platform(s)	SMI
Feature Default Setting	Disabled – Configuration required to enable
Related Documentation	Not Applicable

Revision History

Table 171: Revision History

Revision Details	Release
Enhancement introduced. PCF supports message threshold per endpoint.	2022.02.0
Enhancement introduced. Added procedure to configure the N7 stale session error codes.	2020.05.01
First introduced.	2020.01.0

Feature Description

The PCF Ops Center allows you to configure the advanced tuning parameters for PCF. The tuning parameters primarily consist of the async-threading and http2-threading parameters. These parameters provide the flexibility of the tuning threads responsible for PCF's incoming and outgoing requests over HTTP.

Note Configure the advanced tuning parameter values only if you have a strong understanding of the PCF deployment.

PCF supports the message threshold per endpoint.

Note Message threshold is applicable only for the configured message types in rest-endpoint

Configuration Support for the Advanced Tuning Parameters

This section describes how to configure the advanced tuning parameters using the CLI. The configuration of the advanced tuning parameters involves:

- Configuring the Async Threading Parameters
- Configuring the HTTP2 Threading Parameters
- Configuring the N7 Stale Session Error Codes
- Configuring the Message Threshold Per Endpoint

Configuring the Async Threading Parameters

This section describes how to fine tune the async threading parameters.

To configure the http2-threading parameters, use the following configuration in the Policy Ops Center console:

```
config
  advance-tuning
  async-threading
    default-priority default_priority
    default-worker-threads default_worker_threads
    default-queue-size default_queue_size
    default-processing-threads default_processing_threads
    default-drop-oldest-when-full [ true | false ]
    threading-config service_name
    priority priority
    queue-size queue_size
    threads number_threads
    end
```

NOTES:

- **advance-tuning**—Enters the advance tuning configuration mode.
- **async-threading**—Enters the async threading configuration mode.
- **default-priority** *default_priority*—Specify the default priority level.
- **default-worker-threads** *default_worker_threads*—Specify the default number of worker threads.
- **default-queue-size** *default_queue_size*—Specify the default size of the queue.
- **default-processing-threads** *default_processing_threads*—Specify the default number of threads used for processing.
- **default-drop-oldest-when-full [true | false]** —Indicates if the oldest message in the queue should be removed when the queue is full.
- **threading-config** *service_name*—Specify the service name for which the threading configuration is enabled.
- **priority** *priority*—Specify the priority of the thread.
- **queue-size** *queue_size*—Specify the queue size.
- **threads** *number_threads*—Specify the number of threads to be processed.

Configuring the HTTP2 Threading Parameters

This section describes how to refine the http2-threading parameters.

To configure the http2-threading parameters, use the following configuration in the Policy Ops Center console:

```
config
   http2-threading
   min-thread-pool-size min_thread_pool
   max-thread-pool-size max_thread_pool
     idle-thread-timeout-ms idle_thread_timeout
     max-queue-capacity max_queue_capacity
     disable-validation [ true | false ]
     end
```

NOTES:

- **http2-threading** *http2_threading*—Specify the parameters for inbound SBA requests that are received by PCF.
- **min-thread-pool-size** *min_thread_pool*—Specify the minimum number of threads for processing the inbound SBA request. The accepted range contains integers. Default value is 5.
- **max-thread-pool-size** *max_thread_pool*—Specify the maximum size of the thread pool.
- **idle-thread-timeout-ms** *idle_thread_timeout*—Specify the time in milliseconds that the thread can remain idle. *idle_thread_timeout* must contain only integers. Default value is 60000.
- **disable-validation [true | false]**—Disables the validation of the request sent to PCF. **[true | false]** must contain the value as true or false. Default value is false.
- **max-queue-capacity** *max_queue_capacity*—Specify the maximum number of requests that can wait in the queue for processing. *max_queue_capacity* must contain only integers. Default value is 5000.

- **max-thread-pool-size** *max_thread_pool*—Specify the maximum number of threads that PCF can accommodate in the pool. *max_thread_pool_size* must contain only integers. Default value is 20.

Configuring the N7 Stale Session Error Codes

This section describes how to configure the error codes for the N7 stale sessions.

To configure the n7-stale-session-error-codes parameters, use the following configuration in the Policy Ops Center console:

```
config
  advance-tuning
    n7-stale-session-error-codes error_codes
    end
```

NOTES:

- **n7-stale-session-error-codes** *error_codes*—Specify the error code values for the N7 sessions. When a session is idle, the PCF revalidates it by using the N7NotifyUpdate request. If the N7NotifyUpdate response includes any one or more specified error codes, then the session expiry time is reverted to original value.

  You can specify multiple error codes using comma-separated values.

Configuring the Message Threshold Per Endpoint

This section describes how to configure the message threshold enhancement.

To configure the message threshold enhancement, use the following configuration in the Policy Ops Center console:

```
config
   advance-tuning
      overload-control
         rest
            global
              action threshold-action { N7_CREATE | N7_DELETE | N7_UPDATE
 | N15_CREATE | N15_DELETE | N15_UPDATE}
                  discard-action { DROP | REJECT } threshold-count
threshold_count
                     end
```

NOTES:

- **discard-action { DROP | REJECT } threshold-count** *threshold_count*—Specify the type of discard-action when the message is received at endpoint. The threshold-count provides the maximum number of inbound messages for each threshold-action configured per endpoint. For example, N7_CREATE or N7_DELETE.

OAM Support

This section describes operations, administration, and maintenance information for this feature.

Bulk Statistics Support

This section provides the list of statistics and counters that are generated for the monitoring for message threshold enhancement.

Note The following values apply to all the statistics:

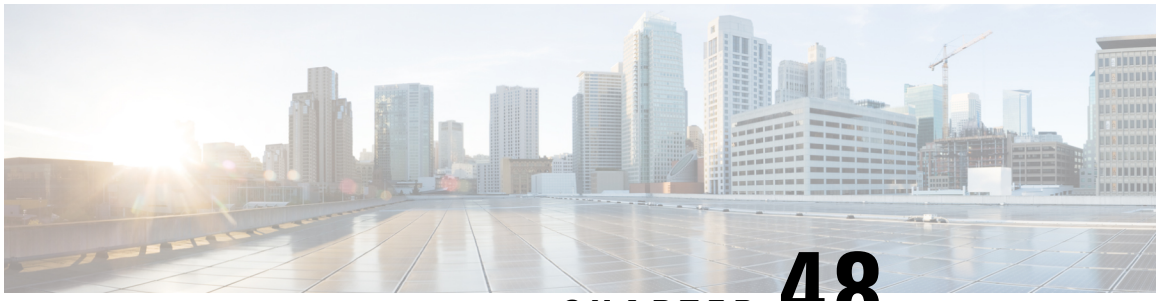
- Unit - Int64
- Type - Counter
- Nodes - Service

The following metrics track the counter information:

- inbound_request_threshold_exceeded_total - Captures the total count of the inbound threshold requests exceeded due to overload.

  The following labels are defined for this metric:

  - interface_name
  - service_name
  - operation_name
  - command
  - action



CHAPTER 48

PCF Application-Based Alerts

Feature Summary and Revision History

Summary Data

Table 172: Summary Data

Applicable Products or Functional Area	PCF
Applicable Platform(s)	SMI
Feature Default Setting	Disabled – Configuration required to enable
Related Documentation	Not Applicable

Revision History

Table 173: Revision History

Revision Details	Release
First introduced.	2020.01.0

Feature Description

When the system detects an anomaly, it generates an alert notification. The system statistics are the cause for these alert notifications. You can set an expression to trigger an alert when the expression becomes true.

How it Works

This section describes how this feature works.

The Common Execution Environment (CEE) uses the Prometheus Alert Manager for alerting operations. The CEE YANG model, accessible through CLI or API, allows you to view the active alerts, silenced alerts, and alert history. During the application installation or upgradation, the system adds a set of preset alerting rules. 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:

- **DefiningAlert 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.

Configuring Alert Rules

This section describes how to configure the alert rules.

To configure the alert rules, use the following configuration:

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

NOTES:

- **alerts rules**—Specify the Prometheus alerting rules.
- **group** *alert_group_name*—Specify the Prometheus alerting rule group. One alert group can have multiple lists of rules. *alert-group-name* is the name of the alert group. *alert_group_name* must be a string in the range of 0–64 characters.
- **rule** *rule_name*—Specify the alerting rule definition. *rule_name* is the name of the rule.
- **expression** *promql_expression*—Specify the PromQL alerting rule expression. *promql_expression* is the alert rule query expressed in PromQL syntax. The *promql_expression* must be a string in the range of 0–64 characters.
- **duration** *duration*—Specify the duration of a true condition before it is considered true. *duration* is the time interval before the alert is triggered.

- **severity** *severity_level*—Specify 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*—Specify 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*—Specify the annotation to attach to the alerts. *annotation_name* is the name of the annotation.
- **value** *annotation_value*—Specify the annotation value. *annotation_value* is the value of the annotation.

The following example configures an alert, which is triggered when the percentage of N7 responses is less than the specified threshold limit.

Example:

```
 configure terminal
  alerts rules group PCFN7chk_incr
  interval-seconds 300
  rule PCFN7chk_incr
  expression "sum(increase(inbound_request_total{interface_name=\"N7\",
result_code=~\"2..\"}[3m])) / sum(increase(inbound_request_total{interface_name=\"N7\"}[3m]))<
 0.95"
    severity major
    type "N7 Communications Alarm"
    annotation summary
    value "This alert is fired when the percentage of N7 responses is less than threshold"
    exit
  exit
exit
```

Viewing Alert Logger

The Alert Logger stores the alerts that PCF generates by defualt. You can view these alerts using the following command:

show alert history [filtering]

You can narrow down the result using the following filtering options:

- **annotations**—Specify the annotations of the alert.
- **endsAt**—Specify the end time of the alert.
- **labels**—Specify the additional labels of the alert.
- **severity**—Specify the severity of the alert.
- **source**—Specify the source of the alert.
- **startsAt**—Specify the start time of the alert.
- **type**—Specify the type of the alert.

You can view the active and silenced alerts with the **show alerts active** and **show alerts active** commands.

Example:

```
show running-config alerts
  interval-seconds 300
  rule PCFN7chk_incr
  expression "sum(increase(inbound_request_total{interface_name=\"N7\",
result_code=~\"2..\"}[3m])) / sum(increase(inbound_request_total{interface_name=\"N7\"}[3m]))<
 0.95"
    severity major
    type "N7 Communications Alarm"
    annotation summary
    value "This alert is fired when the percentage of N7 responses is less than threshold"
    exit
  exit
exit
```

The following example displays the history of the alerts configured in the system:

Example:

```
show alerts history
alerts active PCFN7chk_incr ac2a970ab621
state active
severity major
type "N7 Communications Alarm"
startsAt 2019-11-15T08:26:48.283Z
source System
annotations [ "summary:This alert is fired when the percentage of N7 responses is less than
 threshold.” ]
```

The following example displays the active alerts. The alerts remain active as long as the evaluated expression is true.

Example:

```
show alerts active
alerts active PCFN7chk_incr ac2a970ab621
state active
severity major
type "N7 Communications Alarm"
startsAt 2019-11-15T08:26:48.283Z
source System
annotations [ "summary:This alert is fired when the percentage of N7 responses is less than
 threshold.” ]
```

Sample Alerts Configuration

This section provides sample configurations that are defined in PCF.

Interface-Specific Alerts

N7 Interface Inbound

Use the following commands to configure alerts related to an inbound N7 interface.

```
alerts rules group PCFSvcStatus
   interval-seconds 300
   rule PCFN7Inbound
   expression sum(increase(inbound_request_total{interface_name=\”N7\”,
result_code=~\”2..\”}[5m])) /sum(increase(inbound_request_total{interface_name =\”N7\”}[5m]))
 <0.90
```

```
   severity major
   type Communications Alarm
   annotation summary
   value This alert is fired when the percentage of Success N7 responses sent is lesser
threshold.
   exit
exit
```

N7 Interface Outbound

Use the following commands to configure alerts related to an outbound N7 interface.

```
alerts rules group PCFSvcStatus
   interval-seconds 300
   rule PCFN27outbound
   expression sum(increase(outgoing_request_total{interface_name
=\"N7\",response_status=~\"2..\"}[5m])) /sum(increase(outgoing_request_total{interface_name
 =\"N7\"}[5m])) <0.90
   severity major
   type Communications Alarm
   annotation summary
  value This alert is fired when the percentage of Success N7 responses received is lesser
 threshold.
   exit
exit
```

N28 Interface Inbound

Use the following commands to configure alerts related to an inbound N28 interface.

```
alerts rules group PCFSvcStatus
   interval-seconds 300
   rule PCFN28Inbound
   expression
sum(increase(inbound_request_total{interface_name=\"N28\",response_status=~\"2..\"}[5m]))
/sum(increase(inbound_request_total{interface_name =\"N28\"}[5m])) <0.90
   severity major
   type Communications Alarm
   annotation summary
   value This alert is fired when the percentage of Success N28 responses sent is lesser
 threshold.
   exit
exit
```

N28 Interface Outbound

Use the following commands to configure alerts related to an outbound N28 interface.

```
alerts rules group PCFSvcStatus
   interval-seconds 300
   rule PCFN28outbound
   expression sum(increase(outgoing_request_total{interface_name
=\"N28\",response_status=~\"2..\"}[5m])) /sum(increase(outgoing_request_total{interface_name
 =\"N28\"}[5m])) <0.90
   severity major
```

```
   type Communications Alarm
   annotation summary
   value This alert is fired when the percentage of Success N28 responses received is
lesser threshold.
   exit
exit
```

Diameter Rx Interface Inbound

Use the following commands to configure alerts related to an inbound Diameter Rx interface.

```
alerts rules group PCFSvcStatus
   interval-seconds 300
   rule PCFNRxInbound
   expression
sum(increase(diameter_responses_total{command_code=\"AAA|STA\",response_status=~\"2001\"}[5m]))
 /sum(diameter_responses_total(outgoing_request_total{command_code=\"A AA|STA\"}[5m])) <
0.90
   severity major
   type Communications Alarm
   annotation summary
   value This alert is fired when the percentage of Success Rx responses Send is lesser
threshold.
   exit
exit
```

Diameter Rx Interface Outbound

Use the following commands to configure alerts related to an outbound Diameter Rx interface.

```
alerts rules group PCFSvcStatus
   interval-seconds 300
   rule PCFNRxOutbound
   expression
sum(increase(diameter_responses_total{command_code=\"RAA|ASA\",response_status=~\"2001\"}[5m]))
 /sum(diameter_responses_total(outgoing_request_total{command_code=\"AAA|STA\"}[5m])) <
0.90
   severity major
   type Communications Alarm
   annotation summary
  value This alert is fired when the percentage of Success Rx responses received is lesser
 threshold.
   exit
exit
```

Message-Level Alerts

N7 Create Request

Use the following commands to configure alerts related to N7 Create Request.

```
alerts rules group PCFSvcStatus
   interval-seconds 300
   rule N7CreateRequest
```

```
  expression sum(increase(inbound_request_total{interface_name=\"N7\", command=\"Create\",
 result_code=~\"2..\"}[5m])) / sum(increase(inbound_request_total{interface_name =\"N7\",
command=\"Create\"}[5m])) < 0.90
   severity major
   type Communications Alarm
   annotation summary
   value This alert is fired when the percentage of Success N7 Create responses sent is
lesser threshold.
   exit
exit
```

N7 Update Request

Use the following commands to configure alerts related to N7 Update Request.

```
alerts rules group PCFSvcStatus
   interval-seconds 300
   rule N7UpdateRequest
  expression sum(increase(inbound_request_total{interface_name=\"N7\", command=\"Update\",
 result_code=~\"2..\"}[5m])) /sum(increase(inbound_request_total{interface_name =\"N7\",
command=\"Update\"}[5m])) < 0.90
   severity major
   type Communications Alarm
   annotation summary
   value This alert is fired when the percentage of Success N7 Update responses sent is
lesser threshold.
   exit
exit
```

N7 Delete Request

Use the following commands to configure alerts related to N7 Delete Request.

```
alerts rules group PCFSvcStatus
   interval-seconds 300
   rule N7DeleteRequest
  expression sum(increase(inbound_request_total{interface_name=\"N7\",command=\"Delete\",
 result_code=~\"2..\"}[5m])) /sum(increase(inbound_request_total{interface_name
=\"N7\",command=\"Delete\"}[5m])) < 0.90
   severity major
   type Communications Alarm
   annotation summary
   value This alert is fired when the percentage of Success N7 Delete responses sent is
lesser threshold.
   exit
exit
```

N7 Notify Request

Use the following commands to configure alerts related to N7 Notify Request.

```
alerts rules group PCFSvcStatus
   interval-seconds 60
   rule N7NotifyUpdate
   expression sum(increase(outgoing_request_total{interface_name
=\"N7\",command=\"Notify\", response_status=~\"2..\"}[5m]))
```

```
/sum(increase(outgoing_request_total{interface_name =\"N7\",command=\"Notify\"}[5m])) <
0.90
   severity major
   type Communications Alarm
   annotation summary
   value This alert is fired when the percentage of Success N7 Notify responses received
 is lesser threshold.
   exit
exit
```

N28 Subscribe (Initial) Request

Use the following commands to configure alerts related to N28 Subscribe Request.

```
alerts rules group PCFSvcStatus
   interval-seconds 60
   rule N28Subscribe
   expression sum(increase(outgoing_request_total{interface_name
=\"N28\",command=\"Subscribe\", response_status=~\"2..\"}[5m]))
/sum(increase(outgoing_request_total{interface_name =\"N28\",command=\"Subscribe\"}[5m]))
< 0.90
   severity major
   type Communications Alarm
   annotation summary
   value This alert is fired when the percentage of Success N28 Subscribe (Initial)
responses received is lesser threshold.
   exit
exit
```

N28 Subscribe (Update) Request

Use the following commands to configure alerts related to N28 Subscribe Update Request.

```
alerts rules group PCFSvcStatus
   interval-seconds 60
   rule N28SubscribeUpdate
   expression sum(increase(outgoing_request_total{interface_name =\"N28\",
command=\"Subscribe_Update\", response_status=~\"2..\"}[5m])) /
sum(increase(outgoing_request_total{interface_name =\"N28\",
command=\"Subscribe_Update\"}[5m])) < 0.90
   severity major
   type Communications Alarm
   annotation summary
  value This alert is fired when the percentage of Success N28 Subscribe (Update) responses
 received is lesser threshold.
   exit
exit
```

N28 Notify Request

Use the following commands to configure alerts related to N28 Notify Request.

```
alerts rules group PCFSvcStatus
   interval-seconds 60
   rule N28Notify
   expression sum(increase(outgoing_request_total{interface_name =\"N28\",
```

```
command=\"Notify\", response_status=~\"2..\"}[5m])) /
sum(increase(outgoing_request_total{interface_name =\"N28\", command=\"Notify\"}[5m])) <
0.90
   severity major
   type Communications Alarm
   annotation summary
   value This alert is fired when the percentage of Success N28 Notify responses send is
 lesser threshold.
   exit
exit
```

Rx AAR Request

Use the following commands to configure alerts related to Rx AAR Request.

```
alerts rules group PCFSvcStatus
   interval-seconds 300
   rule RxAAR
   expression sum(increase(diameter_responses_total{command_code=\"AAA\",
response_status=~\"2001\"}[5m])) /
sum(diameter_responses_total(outgoing_request_total{command_code=\"AAA\"}[5m])) < 0.90"
   severity major
   type Communications Alarm
   annotation summary
  value This alert is fired when the percentage of Success Rx AAR responses send is lesser
 threshold.
   exit
exit
```

Rx STR Request

Use the following commands to configure alerts related to Rx STR Request.

```
alerts rules group PCFSvcStatus
   interval-seconds 300
   rule RxSTR
   expression
sum(increase(diameter_responses_total{command_code=\"STA\",response_status=~\"2001\"}[5m]))
 /sum(diameter_responses_total(outgoing_request_total{command_code=\"STA\"}[5m])) < 0.90
   severity major
   type Communications Alarm
   annotation summary
  value This alert is fired when the percentage of Success Rx STA responses send is lesser
 threshold.
   exit
exit
```

Rx RAR Request

Use the following commands to configure alerts related to Rx RAR Request.

```
alerts rules group PCFSvcStatus
   interval-seconds 300
   rule RxSTR
   expression sum(increase(diameter_responses_total{command_code=\"RAA\",
response_status=~\"2001\"}[5m]))
```

```
/sum(diameter_responses_total(outgoing_request_total{command_code=\"RAA\"}[5m])) < 0.90
   severity major
   type Communications Alarm
   annotation summary
   value This alert is fired when the percentage of Success Rx RAR responses Received is
 lesser threshold.
   exit
exit
```

Rx ASR Request

Use the following commands to configure alerts related to Rx ASR Request.

```
alerts rules group PCFSvcStatus
   interval-seconds 300
   rule RxASR
   expression
sum(increase(diameter_responses_total{command_code=\"ASA\",response_status=~\"2001\"}[5m]))
 /sum(diameter_responses_total(outgoing_request_total{command_code=\"ASA\"}[5m])) < 0.90
   severity major
   type Communications Alarm
   annotation summary
  value This alert is fired when the percentage of Success Rx ASR responses send is lesser
 threshold.
   exit
exit
```

Process-Level Alerts

CDL Endpoint Down

Use the following commands to configure alerts related to CDL endpoint down.

```
alerts rules group cdl-ep-change
   rule pod-down
   expression up{pod=~'cdl-ep.*'} == 0
   duration 1m
   severity major
   type Equipment Alarm
   annotation description
   value CDL EP Pod Down
   exit
exit
```

CDL Slot State Change

Use the following commands to configure alerts related to CDL slot state change.

```
alerts rules group cdl-slot-change
   rule pod-down
   expression up{pod=\"cdl-slot-session-c1-m1-0\"} == 0
   severity major
   type Equipment Alarm
```

```
   annotation description
   value CDL Pod Slot Change
   exit
exit
```

Diameter Endpoint State Change

Use the following commands to configure alerts related to Diameter endpoint state change.

```
alerts rules group diamter-ep-change
   rule pod-down
   expression up{pod=~'diameter-ep.*'} == 0
   duration 1m
   severity major
   type Equipment Alarm
   annotation description
   value Diameter EP Change
   exit
exit
```

ETCD State Change

Use the following commands to configure alerts related to etcd state change.

```
alerts rules group ep-mapping-change
   rule pod-down
   expression up{pod=~'etcd-pcf.*'} == 0
   duration 1m
   severity major
   type Equipment Alarm
   annotation description
   value EP Mapping Change
   exit
exit
```

Grafana Dashboard State Change

Use the following commands to configure alerts related to Grafana dashboard state change.

```
alerts rules group grafana-dashboard-change
   rule pod-down
   expression up{pod=~'grafana-dashboard.*'} == 0
   duration 1m
   severity major
   type Equipment Alarm
   annotation description
   value Grafana Dashboard Change
   exit
exit
```

Kafka State Change

Use the following commands to configure alerts related to Kafka state change.

```
alerts rules group kafka-change
   rule pod-down
   expression up{pod=~'kafka.*'} == 0
   duration 1m
   severity major
   type Equipment Alarm
   annotation description
   value Kafka Changed
   exit
exit
```

LDAP Endpoint State Change

Use the following commands to configure alerts related to LDAP endpoint state change.

```
alerts rules group ldap-change
   rule pod-down
   expression up{pod=~'ldap-pcf.*'} == 0
   duration 1m
   severity major
   type Equipment Alarm
   annotation description
   value LDAP Pod Changed
   exit
exit
```

PCF Engine State Change

Use the following commands to configure alerts related to PCF Engine state change.

```
alerts rules group pcf-engine-change
   rule pod-down
   expression up{pod=~'pcf-engine-pcf.*'} == 0
   duration 1m
   severity major
   type Equipment Alarm
   annotation description
   value PCF Engine Changed
   exit
exit
```

REST Endpoint State Change

Use the following commands to configure alerts related to REST endpoint state change.

```
alerts rules group pcf-rest-ep-change
   rule pod-down
   expression up{pod=~'pcf-rest-ep.*'} == 0
   duration 1m
   severity major
   type Equipment Alarm
   annotation description
   value PCF Rest EP Change
```

```
  exit
exit
```

Call Flow Procedure Alerts

LDAP Query Request

Use the following commands to configure alerts related to LDAP Query Request.

```
alerts rules group PCFProcStatus
  interval-seconds 300
  rule LDAPQuery
  expression sum(increase(message_total{type=~\".*_ldap_query\", status=\"success\"}[5m]))
 /sum(increase(message_total{type=~\".*_ldap_query\"}[5m])) < 0.90
  severity major
  type Communications Alarm
  annotation summary
  value This alert is fired when the success percentage of ldap query request is lesser
 threshold.
  exit
exit
```

LDAP Modify Request

Use the following commands to configure alerts related to LDAP Modify Request.

```
alerts rules group PCFProcStatus
  interval-seconds 300
  rule LDAPModify
  expression sum(increase(message_total{component=\"ldap-ep\", type=~\".*_ldap_modify\",
 status=\"success\"}[5m])) / sum(increase(message_total{component=\"ldap-ep\",
type=~\".*_ldap_modify\"}[5m])) < 0.90
  severity major
  type Communications Alarm
  annotation summary
  value This alert is fired when the success percentage of ldap modify request is lesser
 threshold.
  exit
exit
```

PLF Query Request

Use the following commands to configure alerts related to PLF Query Request.

```
alerts rules group PCFProcStatus
  interval-seconds 300
  rule PLFRequest
  expression
sum(increase(message_total{type=~\"ldap_search-res_success\",status=\"success\"}[5m]))
/sum(increase(message_total{type=~\"ldap_search-res_.*\"}[5m])) <0.90
  severity major
  type Communications Alarm
  annotation summary
  value This alert is fired when the success percentage of PLF request is lesser threshold.
```

```
   exit
exit
```

NAP Notification Request

Use the following commands to configure alerts related to NAP Notification Request.

```
alerts rules group PCFProcStatus
   interval-seconds 300
   rule NAPNotification
   expression sum(increase(message_total{type=~\"ldap_change-res_success\",
status=\"success\"}[5m])) /sum(increase(message_total{type=~\"ldap_change-res_.*\"}[5m]))
<0.90
   severity major
   type Communications Alarm
   annotation summary
   value This alert is fired when the success percentage of NAP request is lesser threshold.

   exit
exit
```

System Alerts

Disk Full Alert

Use the following commands to configure alerts related to disk full alert.

```
alerts rules group
   rule node-disk-running-full
   expression node_filesystem_usage > 0.0001
   duration 5m
   severity critical
   type Processing Error Alarm
   annotation disk_full
   value test
   exit
exit
```

VM Down Alert

Use the following commands to configure alerts related to virtual machine down alert.

```
alerts rules group vm-state-change
   rule vm-down
   expression up{pod=~\"node-expo.*\"} == 0
   duration 1m
   severity major
   type Equipment Alarm
   annotation summary
   value VM Down
   exit
exit
```

High Memory Usage

Use the following commands to configure alerts related to high memory usage.

```
alerts rules group memory-util-high
   rule mem-util-high
   expression avg(node_memory_MemAvailable_bytes /node_memory_MemTotal_bytes * 100) by
(hostname) < 20
   duration 1m
   severity critical
   type Processing Error Alarm
   annotation mem_util_high
   value Hig Memory Usage
   exit
exit
```

High Disk Usage

Use the following commands to configure alerts related to high disk usage alert.

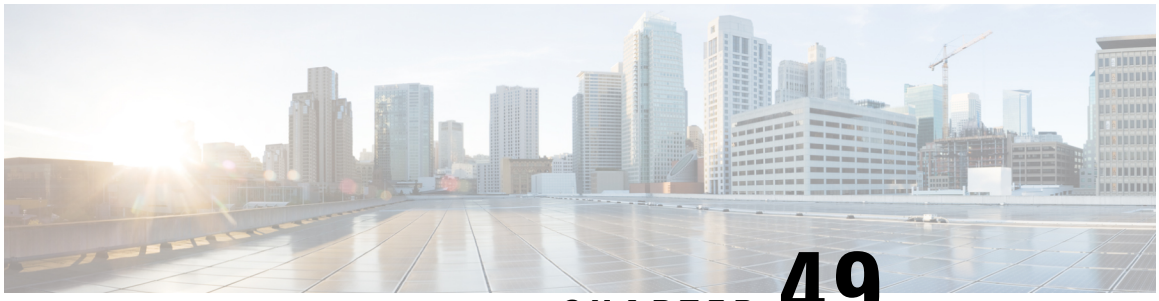
```
alerts rules group disk-util-high
   duration 1m
   rule disk-util-high
   expression avg (node_filesystem_avail_bytes{mountpoint =\"/\"}
/node_filesystem_size_bytes{mountpoint =\"/\"} *100) by (hostname) <20
   severity critical
   type Processing Error Alarm
   annotation description
   value Hig Memory Usage
   exit
exit
```

High CPU Usage

Use the following commands to configure alerts related to high CPU usage alert.

```
alerts rules group cpu-util-high
   rule cpu-util-idle
   duration 1m
   expression avg(rate(node_cpu_seconds_total{mode='idle'}[1m])) by (hostname) *100 < 50

   severity critical
   type Processing Error Alarm
   annotation description
   value Hig CPU
   exit
exit
```



CHAPTER 49

Event Logs

Feature Summary and Revision History

Summary Data

Table 174: Summary Data

Applicable Product(s) or Functional Area	PCF
Applicable Platform(s)	SMI
Feature Default Setting	Enabled – Always-on
Related Documentation	Not Applicable

Revision History

Table 175: Revision History

Revision Details	Release
First introduced.	2020.02.0

Feature Description

PCF provides a centralized view of the application logs that are consolidated from different containers. The unified view improves the efficiency as you can determine the issue faster instead of accessing the individual containers to view the logs. Collection of logs from the containers is enabled by default.

You can view the logs in the real time and offline mode. The real-time mode captures the current event activity that is performed on the container. In the offline mode, you have the flexibility to access the logs from a remote machine.

Logs are listed based on the timestamp at which they are generated.

How it Works

This section describes how this feature works.

The OAM node hosts the logs which different application containers generate. These containers include the pcf (engine), pcf-rest-ep, policy-builder, diameter-ep, ldap-ep, crd, and unifiedapi.

Viewing the Logs

This section describes how to view the consolidated application logs.

To view the consolidated logs, use the following command:

kubectl logs -n *namespace* **consolidated-logging-0**

NOTES:

- *namespace* – Specifies the namespace under which PCF is deployed.

Troubleshooting Information

This section provides information for troubleshooting any issues that may arise during the feature operation.

If the logs are not generated in the consolidated-logging-0 pod, then one of the following conditions may be causing the failure. To resolve the issue, make sure that you do the following:

- Verify the status of <*namespace*>-pcf-oam-app helm deployment. To view the configured helm charts and their status, use the following command:

  helm list

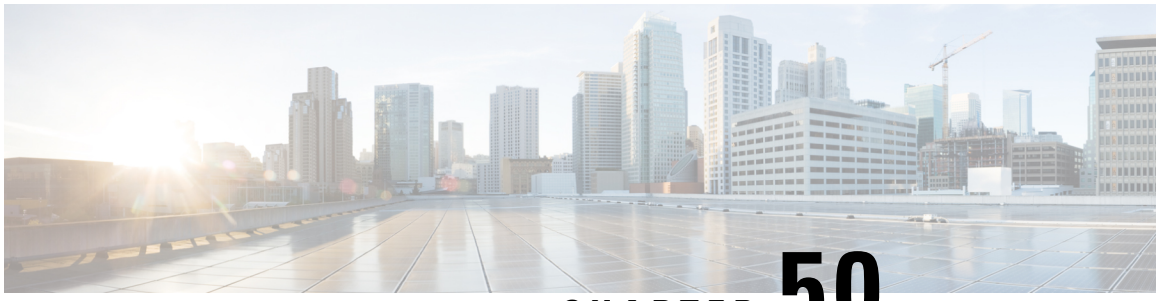
- Ensure that the gRPC stream appender is enabled by verifying the contents of cps-logback configMap. To verify the contents, use the following command:

  kubectl describe configmap -n *namespace* **cps-logback**

- Ensure that the consolidated-logging-0 pod is up and running. To check the pod status, use the following command:

  kubectl describe pod consolidated-logging-0 -n *namespace*

- Verify that the consolidated-logging-0 pod is accessible through the consolidated-logging service. To verify the connection, use the `nc` command.



CHAPTER 50

Troubleshooting Information

Feature Summary and Revision History

Summary Data

Table 176: Summary Data

Applicable Product(s) or Functional Area	SMI
Applicable Platform(s)	PCF
Feature Default Setting	Enabled – Configuration required to disable
Related Documentation	Not Applicable

Revision History

Table 177: Revision History

Revision Details	Release
First introduced.	2020.01.0

Debugging the PCF Deployment Issues

This section describes how to debug the issues that may occur when you deploy PCF through the SMI Deployer.

To debug the deployment issues, use the following checklist. If the checklist does not assist you in resolving the issue, analyze the diagnostic data that is available in the form of logs.

Table 178: Troubleshooting Checklist

Task	Resolution
Verify if the Ops Center is refreshing with the latest configurations	Manually verify if the configurations are refreshed. If the Ops Center is not refreshing or displaying the recent changes, then reinstall the helm charts. For information on reinstalling the charts, see Issue with Refreshing the PCF Ops Center, on page 387.
Validate if the external IPs and ports are accessible.	Use Telnet or any other application protocol and access the external IP address. This is to confirm that the IP address is accessible. If you are unsure of the IP address, run the following in the Kubernetes service to view the configured external IP addresses and port number: **`kubectl get services -n`** *`namespace`*
Ensure that the IP addresses and ports that are configured for PCF are open in the firewall.	Use the following command to open the ports: **`firewall-cmd –zone=public –add-port=`** *`port`***`/tcp –permanent`**
Confirm if PCF connects with the other NFs.	Use the following command on the master node to verify that a healthy connection is available between the NFs: **`nc -v`** Alternatively, from the proto VM, run the **nc -v** command on the Telnet CLI.

Task	Resolution
Validate that the successfully deployed helm chart is listed in the helm list.	Use the following steps to determine which helm chart is not listed in the helm list: 1. Run the following on the master node to view the list of deployed helm charts: **helm list** 2. If the helm chart is not found, run the following in the operational mode to view the charts irrespective of their deployment status. **show helm charts** 3. Review the pcf-ops-center logs to identify the helm chart which has the issue. Depending on the issue, take the appropriate action. Alternatively, you can review the consolidated set of logs, using the following command: **kubectl logs -n namespace consolidated-logging-0** For information about the event logs, see Event Logs, on page 383.

Issue with Refreshing the PCF Ops Center

This section describes how to refresh the PCF Ops Center to display the latest configurations.

Problem

The PCF Ops Center is not considering the recent configurations due to which you may observe stale data or not get the expected response.

Resolution

You can refresh the PCF Ops Center using the basic and advanced steps. Perform the advanced steps only when the basic steps do not resolve the issue.

Basic

1. Run the following to undeploy PCF from the Ops Center:

   system mode shutdown

2. Use the following to manually purge any pending deployments from the helm:

   helm delete --purge *helm_chart_name*

3. From the master node, run the following to delete the configMaps from the namespace where PCF is installed:

   kubectl delete cm *config_map_name* **-n** *namespace*

4. Run the following to delete the product-specific configMaps from the CNEE namespace.
   a. Use the following to list the available configMaps:

      kubectl get configmaps -n *namespace*

From the list, determine the configMap that you want to delete.

b. Run the following to delete the configMap:

Kubectl delete configmap *configmap_name* **-n** *namespace*

5. Use the following commands to reinstall the helm chart. Once the chart is installed, a new instance of the PCF Ops Center is available.

helm upgrade –install *release name addR/chart_name* **-f** *filenames* **--namespace** *namespace*

Advanced

1. Remove the cnee-ops-center.
2. Delete the configMaps from the namespace.

   For more information on step 1 and 2, see the **Basic** steps.
3. Install the PCF Ops Center. For information on how to PCF Ops Center, see Deploying and Accessing PCF, on page 16.

The recent configuration is not rendered because the responsible pods are not in a healthy state to process the refresh request. To investigate the issue at the pod level, review the pod's state.

Use the following command to view the pod's logs:

kubectl describe pod *pod_name* **-n** *namespace*

Alternatively, you can review the consolidated set of logs, using the following command:

kubectl logs -n namespace consolidated-logging-0

For information about the event logs, see Event Logs, on page 383.

In the logs, the values in the Status and Ready columns indicate the following:

- If the Status column displays the state as Running, and the Ready column has the same number of containers on both sides of the forward-slash (/), then the pod is healthy and operational. This implies that the issue is at the application level. To investigate the application issue, check the logs of all the containers residing within the pods to detect the issue. Or, log into the container and review the logs.
- If the Status column displays the state as Pending, Waiting, or CrashLoopBackOff, then run the following to review the details such as the messages, reasons, and other relevant information:

  kubectl describe pod *pod_name* **-n** *namespace*
- If the Status is init or ContainerCreating, it signifies that the pod is in the process of starting up.
- If the Status is Running, and in the Ready column the number of containers on both sides of forward-slash (/) are different, then the containers have issues.

  Run the following to view the details:

  kubectl describe pod *pod_name* **-n** *namespace*

  When reviewing the details, if the Ready column has the value as false then it indicates that the corresponding container has issues. Review the associated logs to understand the issue.

- If the Status and Ready columns, and logs of the container do not indicate any issue, then verify that the required ingress or the service that is required to reach the application is up and running.

Subscriber Not Found or Primary Key Not Found

This section describes how to resolve the issues that report the Subscriber Not Found or Primary Key Not Found messages.

Problem

When the NFs cannot find the subscriber details, they send the Subscriber Not Found or Primary Key Not Found to PCF.

Resolution

1. Analyse the logs of the PCF Engine and REST endpoint pod for the subscriber or primary key related issues.

   On the master node, run the following command to determine the engine and rest-ep pod.

   kubectl logs -n *namespace pod_name*

2. Navigate to the pods and review the subscriber availability status and the subscriber count in the database. Based on the subscriber's status, take the appropriate action to resolve the issue.

   cdl show session *count/summary*

Message Routing Issues

This section describes how to troubleshoot the message routing issues.

Problem

You may observe a message routing failure when a message from the PCF endpoint incorrectly routes a message from Canary to the PCF Engine. The issue occurs when the message is sent to an incorrect PCF group.

Resolution

The following conditions might be causing the message routing failure. Check for these conditions and correct them, if necessary.

- From the PCF Ops Center, manually verify that the routing rules are configured correctly and they match the incoming traffic.
- Ensure that the Istio proxy is injected in the pcf-rest-ep pod.
- Verify that the virtual services are generated using the **istioctl** command. For more information on the traffic routing logs, see Collecting the Troubleshooting Information, on page 390.
- Enable the DEBUG level for com.cisco.pcf.endpoint.routing and review the pcf-rest-ep logs for any issues. Use the following command to enable the DEBUG level:

```
debug logging logger com.cisco.pcf.endpoint.routing level debug
```

Collecting the Troubleshooting Information

If you encounter issues in your PCF environment, gather and analyse the information associated to the failed action or process. Having this information enables you to detect the component that experiences the failure and resolve the issue faster.

The following table covers the components which might experience an issue, and the logs that contain the information corresponding to the issue.

Table 179: Issues

Issue	Logs
Deployment errors	Review the following logs to determine the issue. These logs assist you in identifying the component that may be the source of the error. Use the following commands on the master node: • View the available pods and review the pod status: **kubectl logs -n** *namespace pod_name* Depending on the pod's state, perform the appropriate remediation actions. To understand the pod's states, see States, on page 254. • View the configured helm charts and their status: **helm list** • View the helm chart details for the REST endpoint: **helm get** *namespace* **-pcf-rest-ep**
Communication issues between the NFs	1. On the master node, run the following command to identify the pod that is responsible for the communication: **kubectl logs -n** *namespace pod_name* 2. Use the tcpdump utility to trace the packets.
Registration and deregistration issues	Use the following command to review the PCF REST endpoint logs: **helm get** *namespace* **-pcf-rest-ep**
Ops Center issues	Review the pod's log that hosts the Ops Center to determine the issue. **kubectl logs -n** *namespace pod_name* To resolve the issue, if you require the configuration information, then run one of the following commands: **show full-configuration** Or, **show running-config**

Issue	Logs
Traffic routing issues	To view the traffic routing-specific logs, use the following configuration: **kubectl get pod -o yaml -n** *namespace pcf-rest-ep pod_name* **istioctl get virtualservice -n** *namespace* **-o yaml** **istioctl get destinationrules -n** *namespace* **-o yaml** Also, review the logs of the following pods: • Pcf-rest-ep instance • Pcf-engine instance • Datastore or Session DB
Subscriber issues	Review the logs associated to the PCF Engine and REST endpoint to determine the issue. For additional information about the subscriber availability status and the subscriber count in the database, run the following command: **cdl show session** *count/summary*

Alerts

Alerts are notification messages that are generated when incidents requiring your attention or response occur. Review the historical and active alerts to determine the issue.

Alerts for PCF are generated through the CEE utility. To view these alerts, run the following command in the CEE Ops Center:

For active alerts:

```
show alerts active
```

For historical alerts:

```
show alerts history
```

Note You must have appropriate permission to view the alert details.

For information on application-based alerts, see PCF Application-Based Alerts, on page 367.

Interface Error Codes

This section describes the codes that PCF reports for the interface errors.

Interface codes are generated as part of the logs or captured in the statistics.

The following tables describes the error and the corresponding codes:

Table 180: N7 Error Codes

Error	Error Code	Description
USER_UNKNOWN	400 Bad Request	The HTTP request is rejected because the end user who is specified in the request is unknown to the PCF.
ERROR_INITIAL_PARAMETERS	400 Bad Request	The HTTP request is rejected. This error is reported when the set of session or subscriber information which PCF requires for a rule selection is incomplete, erroneous, or unavailable for decision making. For example, QoS, RAT type, and subscriber information.
ERROR_TRIGGER_EVENT	400 Bad Request	The HTTP request is rejected because the set of session information sends a message that originated due to a trigger is incoherent with the previous set of session information for the same session. For example, trigger met was RAT changed, and the RAT notified is the same as before.
TRAFFIC_MAPPING_ INFO_REJECTED	403 Forbidden	The HTTP request is rejected because the PCF doesn't accept one or more of the traffic mappings filters provided by the SMF in a PCC Request.
ERROR_CONFLICTING_REQUEST	403 Forbidden	The HTTP request is rejected because the PCF can't accept the UE-initiated resource request as a network initiated resource allocation is already in-progress. This resource allocation has packet filters that cover the packet filters in the received UE-initiated resource request. The SMF rejects the attempt for a UE-initiated resource request.
POLICY_CONTEXT_DENIED	403 Forbidden	The HTTP request is rejected because the PCF doesn't accept the SMF request due to operator policies and local configuration.

Table 181: N28 Error Codes

Error	Error Code	Description
USER_UNKNOWN	400 Bad Request	The subscriber that is specified in the request isn't known at the CHF and the subscription can't be created.

Error	Error Code	Description
NO_AVAILABLE_POLICY _COUNTERS	400 Bad Request	There are no policy counters available for the subscriber at the CHF.

Note The generic error codes are applicable for all the network interfaces.

Table 182: Generic Error Codes

Error	Error Code	Description
TIMEOUT	408 Request Timeout	The HTTP request to the server took longer than the period the server is configured to wait.
OVERLOAD	429 Too Many Requests	The server has received too many consecutive requests to process within a short interval.
INTERNAL_ERROR	500 Internal Server Error	The server has encountered an unprecedented condition, which does not have an appropriate message.
SERVICE_UNAVAILABLE	503 Service Unavailable	The server cannot process the request because it is either, overloaded or is unavailable due to scheduled maintenance. This is a transient state.

Forwarding logs to the Splunk Server

This section describes how to enable PCF to forward the logs to the Splunk server.

Splunk is a third-party monitoring application that stores the log files and provides index-based search capability. You can configure PCF to send the logs securely to a Splunk server which could be an external server.

Important The Splunk server is a third-party component. Cisco does not take the responsibility of installing, configuring, or maintaining this server.

Use the following configuration to forward the logs to the Splunk server.

```
config
  debug splunk
    batch-count no_events_batch
```

batch-interval-ms*batch_interval_ms*
batch-size-bytes *batch_size*
hec-token*hec_token*
hec-url *hec_url*
end

The following is an example configuration:

```
configure
 debug splunk hec-url https://splunk.10.86.73.80.nip.io:8088
 debug splunk hec-token 68a81ab4-eae9-4361-92ea-b948f31d26ef
 debug splunk batch-interval-ms 100
 debug splunk batch-count 10
 debug splunk batch-size-bytes 102400
 end
```

NOTES:

- **debug splunk**—Enters the configuration debug mode.
- **batch-count** *no_events_batch*—Specify the maximum number of events to be sent in each batch.
- **batch-interval-ms** *batch_interval_ms*—Specify the interval in milliseconds at which a batch event is sent.
- **batch-size-bytes** *batch_size*—Specify the maximum size in bytes of each batch of events.
- **hec-token** *hec_token*—Specify the HTTP Event Collector (HEC) token for the Splunk server.
- **hec-url** *hec_url*—Specify the protocol, hostname, and HTTP Event Collector port of the Splunk server. The default port is 8088.

Pods stop running when PCF is upgraded through the Rolling Upgrade process

This section describes how to ensure that the pods are running when PCF is upgraded.

Problem

When the PCF version is upgraded to the subsequent available version, some pods such as CRD and Policy Engine stop running.

Resolution

Whenever you configure PCF ensure that you configure the following parameters:

- **db global-settings db-replica** *replica_count*
- **db spr shard-count** *shard_count*
- **rest-endpoint ips** *ip_address1, ip_address2, ip_address3*
- **rest-endpoint port** *port_number*
- **engine** *engine_name*

  replicas *replica_count*

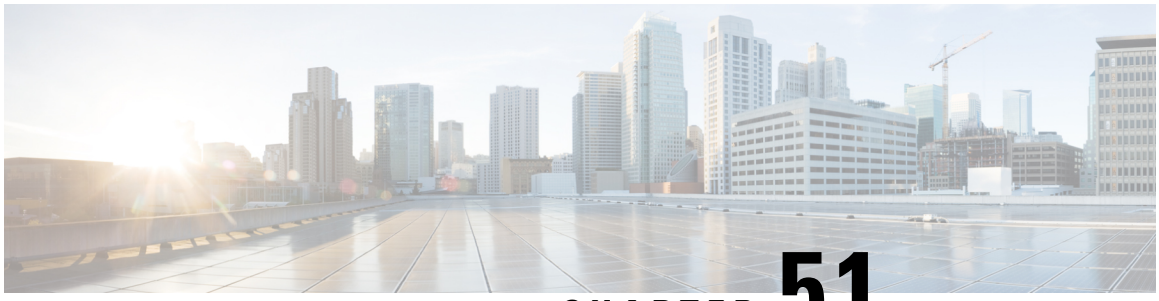
unified-api-replicas *api_replica_count*

subversion-run-url *repository_url*

subversion-config-url *configuration_url*

tracing-service-name *service_name*

- **service-registration profile locality** *profile_name*
- **service-registration profile plmn-list [mcc mnc]**
- **service-registration profile snssais [sst sd]**



CHAPTER 51

Sample PCF Configuration

Sample Configuration File

The following is only a sample configuration file provided solely for your reference. You must create and modify your own configuration file according to the specific needs of your deployment.

Important The mandatory parameters are required to ensure that the critical pods such as CRD and Policy Engine are in the running state.

```
config
datastore primary-endpoint connection-settings keep-alive keep-alive-time-ms 200
datastore primary-endpoint connection-settings channel count 4
datastore primary-endpoint connection-settings timeout-ms 500
datastore external-endpoints datastore
 connection-settings keep-alive keep-alive-time-ms 200
 connection-settings channel count 3
 connection-settings timeout-ms 500
exit
ldap replicas 2
ldap server-set USD
 search-user dn cn=sdcUser,dc=C-NTDB
 search-user password $8$yx0jELXTK0f7CJO2XklpJx+CpCUIX13B9C5oQ4NEnaI=
 health-check interval-ms 5000
 health-check dn cn=sdcUser,dc=C-NTDB
 health-check filter msisdn=918369110173
 health-check attributes napCustType
 initial-connections                              10
 max-connections                                  10
 retry-count                                      2
 retry-timer-ms                                   100
 max-failover-connection-age-ms                   60000
 binds-per-second                                 0.2
 number-consecutive-timeouts-for-bad-connection -1
 missing-attribute-result-code                    32
 connection 192.0.2.18 389
  priority        400
  connection-rule ROUND_ROBIN
  auto-reconnect  true
  timeout-ms      200
```

```
  bind-timeout-ms 3000
 exit
 connection 192.0.2.18 390
  priority        400
  connection-rule ROUND_ROBIN
  auto-reconnect  true
  timeout-ms      200
  bind-timeout-ms 3000
 exit
 connection 192.0.2.18 391
  priority        400
  connection-rule ROUND_ROBIN
  auto-reconnect  true
  timeout-ms      200
  bind-timeout-ms 3000
 exit
exit
//This is a mandatory parameter
db global-settings db-replica 3
//This is a mandatory parameter
db global-settings volume-storage-class local
db spr shard-count 1
db balance shard-count 1
debug tracing type DISABLED
debug logging default-level error
debug logging logger com.broadhop
 level warn
exit
debug logging logger com.broadhop.custrefdata.impl.dao.GenericDao
 level error
exit
debug logging logger com.broadhop.diameter2.policy.endpoints
 level error
exit
debug logging logger com.broadhop.ldap
 level error
exit
debug logging logger com.broadhop.microservices.control
 level error
exit
debug logging logger com.broadhop.utilities.queue.redis
 level error
exit
debug logging logger com.cisco
 level warn
exit
debug logging logger com.cisco.diameter
 level error
exit
debug logging logger com.cisco.diameter.endpoint
 level error
exit
debug logging logger com.cisco.pcf
 level debug
exit
debug logging logger com.cisco.pcf.endpoint.client
 level error

exit
debug logging logger com.cisco.pcf.endpoint.client.Http2JettyRequestAsync
 level error
exit
debug logging logger com.cisco.pcf.ldapserver
 level warn
```

```
exit
debug logging logger com.cisco.pcf.nf.cache.NfCache
 level warn
exit
debug logging logger io.prometheus.client
 level error
exit
debug logging logger policy.engine
 level debug
exit
debug logging logger rest.message
 level warn
exit
features patching ingress-enabled true
diameter settings timeouts-ms dpa 5000
diameter application rx
 application-id   16777236
 tgpp-application true
 vendor           [ 10415 ]
exit
diameter group rx-protocol-1
 mode server
 stack rx-protocol-1
  application rx
  replicas    2
  bind-ip      192.0.2.19
  bind-port   3868
  fqdn        pcf-rx-server-1
  realm       pcf.rx.server.cisco.com
  settings timeouts-ms request 5000
 exit
exit
ldap-server-endpoint connect bind-ip  192.0.2.20
ldap-server-endpoint connect binddn cn=plfuser
ldap-server-endpoint connect password $8$1eiow0TCw8sMRzP8czGABoog5Y1DxrD49EGWVmw3PoI=
ldap-server-endpoint connect port 1399
ldap-server-endpoint connect request-timeout 5000
ldap-server-endpoint connect replicas 2
ldap-server-endpoint connect max-transactions 200
ldap-server-endpoint health-check-filter name msisdn
ldap-server-endpoint health-check-filter value 11110100000
ldap-server-endpoint input-mapping framedipaddress
 internal-lookup-key IP_ADDRESS
exit
ldap-server-endpoint input-mapping imsi
 internal-lookup-key IMSI
exit
ldap-server-endpoint input-mapping msisdn
 internal-lookup-key MSISDN
exit
ldap-server-endpoint output-mapping ACCESS_TYPE
 input accessType
exit
ldap-server-endpoint output-mapping RAT_TYPE
 input ratType
exit
ldap-server-endpoint output-mapping calledstationid
 input dnn
exit
ldap-server-endpoint output-mapping callingstationid
 input msisdn
exit
ldap-server-endpoint output-mapping framedipv4
 input framedIp
```

```
exit
ldap-server-endpoint output-mapping framedipv6
 input framedIpv6Prefix
exit
ldap-server-endpoint output-mapping imsi
 input supi
exit
ldap-server-endpoint output-mapping offline_charging
 input offline
exit
ldap-server-endpoint output-mapping online_charging
 input online
exit
ldap-server-endpoint health-check-attributes msisdn
 value 11110100000
exit
//This is a mandatory parameter
rest-endpoint port   9082
rest-endpoint tracing-service-name pcf-rest-endpoint
rest-endpoint replicas 2
rest-endpoint interface n28
 ip [  192.0.3.20 ]
exit
rest-endpoint interface n7
 ip [  192.0.3.21 ]
exit
rest-endpoint interface n15
 ip   [ 10.102.3.218 ]
 port 9082
exit
rest-endpoint interface nnrf
 ip                          [  192.0.2.22 ]
 outbound-request-timeout-ms 500
exit
advance-tuning http2-threading min-thread-pool-size 10
advance-tuning http2-threading max-thread-pool-size 25
advance-tuning http2-threading disable-validation false
advance-tuning overload-control rest global limits max-requests-per-sec 9000
advance-tuning overload-control rest global action throttle-action REJECT
advance-tuning overload-control rest global action throttle-action N7_CREATE discard-action
 DROP threshold-count 3500

advance-tuning overload-control rest global action throttle-action N7_CREATE discard-action
 REJECT threshold-count 2000

advance-tuning overload-control diameter global limits max-requests-per-sec 9000
advance-tuning overload-control diameter global action throttle-action DROP
advance-tuning async-threading default-worker-threads 20
advance-tuning async-threading default-queue-size 100
advance-tuning async-threading default-processing-threads 20
advance-tuning async-threading http2-connect-timeout-ms 120
api unified engine-group pcf01production
//This is a mandatory parameter
api unified externalIPs [  192.0.2.23 ]
//This is a mandatory parameter
api unified external-port 8080
//This is a mandatory parameter
engine pcf01production
//This is a mandatory parameter
 replicas              2
//This is a mandatory parameter
 subversion-run-url    http://svn/repos/run
//This is a mandatory parameter
 subversion-config-url http://svn/repos/configuration
```

```
//This is a mandatory parameter
 tracing-service-name  pcf-engine
 properties broadcast.tps
  value 100
 exit
 properties ldap.retry.time.ms
  value 200
 exit
 properties loopback.delay
  value 20
 exit
 properties pcf.actions.sync.timeoutMs.default
  value 410
 exit
 properties useZlibCompression
  value true
 exit
 properties virtualservice.cache.enabled
  value true
 exit
 properties virtualservice.evaluate.defaultvs
  value true
 exit
 properties warmup.message.count
  value 20
 exitexit
label protocol-layer key smi.cisco.com/node-type-2
label protocol-layer value protocol
label service-layer key smi.cisco.com/node-type-3
label service-layer value service
label cdl-layer key smi.cisco.com/node-type-4
label cdl-layer value session
label oam-layer key smi.cisco.com/node-type
label oam-layer value oam
external-services datastore
 ips   [  192.0.2.24 ]
 ports [ 8882 ]
exit
profile nf-client nf-type udr
 udr-profile local-udr
  locality localudr
   priority 10000
   service name type nudr-dr
    endpoint-profile udr_profile_1
     capacity   10
     priority   30
     uri-scheme http
     version
      uri-version v2
      exit
     exit
     endpoint-name udr_ep1
      primary ip-address ipv4 10.102.4.151
      primary ip-address port 5182
     exit
    exit
    endpoint-profile udr_profile_2
     capacity   10
     priority   30
     uri-scheme http
     version
      uri-version v2
      exit
     exit
```

```
      endpoint-name udr_ep1
       primary ip-address ipv4 10.102.4.151
       primary ip-address port 5183
      exit
     exit
     endpoint-profile udr_profile_3
      capacity   10
      priority   30
      uri-scheme http
      version
       uri-version v2
       exit
     exit
      endpoint-name udr_ep1
       primary ip-address ipv4 10.102.4.151
       primary ip-address port 5184
      exit
     exit
    exit
   exit
  exit
 exit
profile nf-client nf-type chf
 chf-profile local-chf
  locality localchf
   priority 10000
   service name type nchf-spendinglimitcontrol
    endpoint-profile chf_profile_1
     capacity   50
     priority   30
     uri-scheme http
     version
      uri-version v1
      exit
     exit
     endpoint-name chf_ep1
      primary ip-address ipv4 10.102.1.151
      primary ip-address port 5082
     exit
    exit
    endpoint-profile chf_profile_2
     capacity   50
     priority   30
     uri-scheme http
     version
      uri-version v1
      exit
     exit
     endpoint-name chf_ep1
      primary ip-address ipv4 10.102.1.151
      primary ip-address port 5083
     exit
    exit
    endpoint-profile chf_profile_3
     capacity   50
     priority   30
     uri-scheme http
     version
      uri-version v1
      exit
     exit
     endpoint-name chf_ep1
      primary ip-address ipv4 10.102.1.151
      primary ip-address port 5084
```

```
     exit
    exit
   exit
  exit
 exit
exit
profile nf-pair nf-type UDR
 nrf-discovery-group    nrf-discovery-group
 subscription-enabled   true
 subscription-extension 3
 locality client  pcf01
 locality preferred-server loc1
 locality geo-server loc2
exit
profile nf-pair nf-type CHF
 nrf-discovery-group    nrf-discovery-group
 subscription-enabled   true
 subscription-extension 3
 locality client  pcf01
 locality preferred-server loc1
 locality geo-server loc2
exit
service-registration services amfService
exit
service-registration services smfService
exit
//This is a mandatory parameter
service-registration profile locality pcf01
//This is a mandatory parameter
service-registration profile capacity 20
//This is a mandatory parameter
service-registration profile priority 10
//This is a mandatory parameter
service-registration profile nf-status REGISTERED
//This is a mandatory parameter
service-registration profile plmn-list 100 010
exit
//This is a mandatory parameter
service-registration profile snssais
//This is a mandatory parameter
1 sd ABCDEF
exit
group nf-mgmt nf-mgmt-grpup
 nrf-mgmt-group nrf-register-group
 locality       pcf01
 failover sla 1200
 reconnect interval 100
exit
group nrf discovery nrf-discovery-group
 service type nrf nnrf-disc
  endpoint-profile nrf_disc_profile_1
   capacity   10
   priority   10
   uri-scheme http
   version
    uri-version v1
    exit
   exit
   endpoint-name nrf_disc_ep1
    priority 1
    capacity 10
    primary ip-address ipv4 192.0.2.26
    primary ip-address port 8183
    secondary ip-address ipv4 192.0.2.19
```

```
    secondary ip-address port 8184
    tertiary ip-address ipv4 192.0.2.15
    tertiary ip-address port 8185
   exit
  exit
 exit
exit
group nrf mgmt nrf-register-group
 service type nrf nnrf-nfm
  endpoint-profile nrf_regi_profile_1
   capacity   10
   priority   10
   uri-scheme http
   version
    uri-version v1
    exit
   exit
   endpoint-name nrf_regi_ep1
    priority 1
    capacity 10
    primary ip-address ipv4 192.0.1.15
    primary ip-address port 8183
    secondary ip-address ipv4 192.0.3.15
    secondary ip-address port 8184
    tertiary ip-address ipv4 192.0.2.12
    tertiary ip-address port 8185
   exit
  exit
 exit
exit
cdl system-id          1
cdl node-type          session
cdl enable-geo-replication true
cdl zookeeper data-storage-size 1
cdl zookeeper log-storage-size 1
cdl zookeeper replica 3
cdl remote-site 2
 db-endpoint host  192.0.2.24
 db-endpoint port 8882
 kafka-server 19.102.11.108 10091
 exit
 kafka-server 19.102.11.109 10092
 exit
 kafka-server 19.102.11.110 10093
 exit
exit
cdl label-config session
 endpoint key smi.cisco.com/node-type-4
 endpoint value session
 slot map 1
  key   smi.cisco.com/node-type-4
  value session
 exit
 slot map 2
  key   smi.cisco.com/node-type-4
  value session
 exit
 slot map 3
  key   smi.cisco.com/node-type-4
  value session
 exit
 slot map 4
  key   smi.cisco.com/node-type-4
  value session
```

```
 exit
 index map 1
  key   smi.cisco.com/node-type-4
  value session
 exit
 index map 2
  key   smi.cisco.com/node-type-4
  value session
 exit
exit
cdl logging logger datastore.ep.session
 level debug
exit
cdl logging logger datastore.index.session
 level debug
exit
cdl logging logger datastore.slot.session
 level debug
exit
cdl datastore session
 cluster-id      1
 label-config    session
 geo-remote-site [ 2 ]
 endpoint replica 2
 endpoint external-ip 10.102.11.218
 index replica 2
 index map 2
 slot replica 2
 slot map 4
 slot notification limit 25
exit
cdl kafka replica 3
cdl kafka storage 1
cdl kafka label-config key smi.cisco.com/node-type-4
cdl kafka label-config value session
cdl kafka external-ip 10.102.11.104 10091
exit
cdl kafka external-ip 10.102.11.105 10092
exit
cdl kafka external-ip 10.102.11.106 10093
exit
system mode running
helm default-repository base-repos
helm repository base-repos
 url https://charts.10.100.11.107.nip.io/pcf.2020.05.m0.i33
exit
k8s name          cl-hawaii-s1
k8s namespace     pcf-pcf-hawaii-s1
k8s nf-name       pcf
k8s registry      docker.10.100.11.107.nip.io/pcf.2020.05.m0.i26
k8s single-node   false
k8s use-volume-claims true
k8s ingress-host-name 10.84.102.200.nip.io
k8s nodes cl-hawaii-s1-master-1
 node-type   master
 worker-type master
exit
k8s nodes cl-hawaii-s1-master-2
 node-type   master
 worker-type master
exit
k8s nodes cl-hawaii-s1-master-3
 node-type   master
 worker-type master
```

```
exit
aaa authentication users user admin
 uid        1117
 gid        1117
 password   $1$ywmCvRqU$e1ho7HLAmgfQS5LT9HAXQ.
 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 groups group policy-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
```