Deploying and Configuring PCF through Ops Center

Feature Summary and Revision History

Summary Data

Table 1: 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 2: 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
      ```

MongoDB Authentication

Table 3: Feature History

Feature Name	Release Information	Description
MongoDB Authentication for Database Security	2025.03.0	The MongoDB authentication feature ensures robust end-to-end authentication and security for the MongoDB database used by the PCF. The authentication is qualified through CLI. MongoDB serves as the administrative database supporting critical functionalities such as Custom Reference Data (CRD) and Policy tracing, and also for inbuilt Subscriber Profile Repository (SPR) and Balance databases, enhancing operational efficiency. Command introduced: **db global-settings db-user-name** *MongoDB-username* **password** *password* **password** : Enables database authentication.

Introduction to MongoDB Authentication

The Policy Control Function (PCF) uses MongoDB as its administrative database, which supports critical functionalities such as Custom Reference Data (CRD) and Policy tracing. Additionally, PCF incorporates inbuilt Subscriber Profile Repository (SPR) and Balance databases to enhance its capabilities. To ensure robust security, PCF employs MongoDB authentication method, providing end-to-end protection for these databases.

Benefits

The key benefits are:

- Improved Security: Enabling authentication protects sensitive database information.
- Streamlined Configuration: Simple configuration options improve ease of use for MongoDB.
- Encryption Support: Encrypts passwords for secure storage and usage.

How MongoDB authentication works

Perform MongoDB authentication process in three stages:

- Authentication setup
- Password Encryption
- Database connection

Authentication setup

You can perform these authentication to:

- Enable MongoDB authentication using the Ops-Center configuration commands
- Configure the MongoDB username and passwords

Password encryption

To perform password encryption:

- Use the CLI utility to encrypt the password.
- Add the encrypted password as a JVM argument for the Engine, PB, and CRD processes.

Database connection

When you connect to the database:

- Ensure reachability between MongoDB replica members and PCF Engine pods.
- Access MongoDB using the mongo command with the actual username, and password.

Enable database authentication

Use these steps to enable MongoDB authentication for preserving subscriber-specific, balance data, and admin configuration data.

Before you begin

Ensure MongoDB authentication is enabled only in a system shutdown state.

Procedure

Step 1 Log into your Ops center configuration to configure and manage the applications and pods configuration.

Step 2 Choose **db global-settings** to specify the username and password for the Database.

```
config
  db
    global-settings
      db-user-name MongoDB_username
      password PlainTextPassword
    commit
```

Replace *MongoDB_username* and *PlainTextPassword* with actual username and password for authentication.

Example:

```
db global-settings db-user-name admin password abcxy@123
```

Step 3 Enter **commit** to enable the MongoDB authentication.

What to do next

For the engine to connect to the database configuration, configure the pcf engine, Policy builder (PB), and Custom Reference Data (CRD). For more information, see the *Configure PCF Engine, PB, and CRD section*.

Note In the **show running-config** command output the plain text password gets displayed in the encrypted format.

Configure PCF Engine, PB, and CRD

Use these steps to establish a database connection and ensure reachability between MongoDB replica members and the PCF Engine pod.

Procedure

Step 1 Log in to your Ops Center configuration.

The engine name appears based on the deployment configurations.

Step 2 Enter the **ops-center** command to encrypt and decrypt the plain text password:

a) Enter **db scdb execute password encrypt-password plain-text** *PlainTextPassword* that you spceified during enabling MongoDB authentication parameter.

Example:

```
[unknown] pcf# db scdb execute password encrypt-password plain-text abcxy@123
Mon Jun  16 08:14:14.138 UTC+00:00
output :
5F16DD813A2641A15B928FE70D2FDA18
```

b) Enter **db scdb execute password decrypt-password encrypted-text** *EncryptedPassword* to decrypt the plain text password.

Example:

```
[unknown] pcf# db scdb execute password decrypt-password encrypted-text
5F16DD813A2641A15B928FE70D2FDA18
Fri Jun 16 08:14:14.138 UTC+00:00
output :
abcxy@123
```

Step 3 Enter the properties needed to set the database username and encrypted password for the PCF engine, using these command:

```
config
       engine engine_name properties dbUsername value username
        engine engine_name properties dbpassword value encryptedpassword
        exit
```

Example:

```
engine pcf-green properties dbUsername value admin
engine pcf-green properties dbPassword value 5F16DD813A2641A15B928FE70D2FDA18
```

Step 4 Enter **commit** to commit the changes.

What to do next

After MongoDB authentication is enabled, complete the tasks:

- Verify the connectivity to MongoDB replica members from PCF engine pods.
- Access mongo replica set using the **mongo relica set**.

```
mongo primary_host_ip mongo_port -u admin -p password
```