Oracle® Communications

Policy and Charging Rules Function Installation and Upgrade Guide

Release 1.0
F20495-01
July 2019

Oracle Communications Policy and Charging Rules Function Installation and Upgrade Guide, Release 1.0

F20495-01

Contents

List of Tables

1 Introduction

The Oracle Communications Cloud Native Policy and Charging Rules Function (PCRF) solution incorporates new architecture with spring micro-service framework as backend support technology stack and Kubernetes Cloud Native Environment as running environment. The PCRF core service is the main functionality among PCRF micro services with the following enhancements when compared to legacy PCRF:

- Remove the MIA module from MPE, and let the MPE talks to with configuration server to save/load related data
- PCRF core service have integrated the MPE functionalities which are under legacy PCRF
- When PCRF Core needs to talk with any data source, these traffic shall go with the Diameter connector rather than from the PCRF core itself

For more information, see *Oracle Communications Policy and Charging Rules Function User's Guide*.

References

Refer to the following documents for more information about 5G cloud native Policy and Charging Rules Function.

- Cloud Native Environment Installation Document
- Policy and Charging Rules Function Cloud Native User's Guide

Acronyms

The following table provides information about the acronyms used in the document.

Table 1-1 Acronyms

Acronym	Definition
5GC	5G Core Network
5GS	5G System
5G-AN	5G Access Network
5G-EIR	5G-Equipment Identity Register
5G-GUTI	5G Globally Unique Temporary Identifier
5G-S-TMSI	5G S-Temporary Mobile Subscription Identifier
5QI	5G QoS Identifier
AF	Application Function
AMF	Access and Mobility Management Function
AS	Access Stratum
AUSF	Authentication Server Function
BSF	Binding Support Function

Table 1-1 (Cont.) Acronyms

Acronym	Definition
CAPIF	Common API Framework for 3GPP northbound APIs
CUSTOMER_REPO	The docker registry address in customer side, plus Port No. if registry has port attached
IMAGE_TAG	The image tag from release tar file is 1.1.0, You can decide to use any tag No. Then push related docker image with that specific tag to their registry
METALLB_ADDRESS_POOL	The address pool which configured on metallb to provide external IPs
MPS	Multimedia Priority Service
N3IWF	Non-3GPP InterWorking Function
NAI	Network Access Identifier
NEF	Network Exposure Function
NF	Network Function
NGAP	Next Generation Application Protocol
NR	New Radio
NRF	Network Repository Function
NSI ID	Network Slice Instance Identifier
NSSAI	Network Slice Selection Assistance Information
NSSF	Network Slice Selection Function
NSSP	Network Slice Selection Policy
NWDAF	Network Data Analytics Function
PCF	Policy Control Function
PDR	Packet Detection Rule
PEI	Permanent Equipment Identifier
PER	Packet Error Rate
PFD	Packet Flow Description
PPD	Paging Policy Differentiation
PPF	Paging Proceed Flag
PPI	Paging Policy Indicator
PSA	PDU Session Anchor
QFI	QoS Flow Identifier
QoE	Quality of Experience
(R)AN	(Radio) Access Network
RQA	Reflective QoS Attribute
RQI	Reflective QoS Indication
SA NR	Standalone New Radio
SBA	Service Based Architecture
SBI	Service Based Interface
SD	Slice Differentiator
SEAF	Security Anchor Functionality
SEPP	Security Edge Protection Proxy
SMF	Session Management Function

Table 1-1 (Cont.) Acronyms

Acronym	Definition
SMSF	Short Message Service Function
S-NSSAI	Single Network Slice Selection Assistance Information
SSC	Session and Service Continuity
SSCMSP	Session and Service Continuity Mode Selection Policy
UDM	Unified Data Management
UDR	Unified Data Repository
UDSF	Unstructured Data Storage Function

Locate Product Documentation on the Oracle Help Center Site

Oracle Communications customer documentation is available on the web at the Oracle Help Center (OHC) site, http://docs.oracle.com. You do not have to register to access these documents. Viewing these files requires Adobe Acrobat Reader, which can be downloaded at http://www.adobe.com.

1. Access the Oracle Help Center site at http://docs.oracle.com.
2. Click `Industries`.
3. Under the Oracle Communications subheading, click the `Oracle Communications documentation` link.

   The Communications Documentation page appears. Most products covered by these documentation sets will appear under the headings "Network Session Delivery and Control Infrastructure" or "Platforms."
4. Click on your Product and then the Release Number.

   A list of the entire documentation set for the selected product and release appears.
5. To download a file to your location, right-click the `PDF` link, select `Save target as` (or similar command based on your browser), and save to a local folder.

My Oracle Support

My Oracle Support (https://support.oracle.com) is your initial point of contact for all product support and training needs. A representative at Customer Access Support can assist you with My Oracle Support registration.

Call the Customer Access Support main number at 1-800-223-1711 (toll-free in the US), or call the Oracle Support hotline for your local country from the list at http://www.oracle.com/us/support/contact/index.html. When calling, make the selections in the sequence shown below on the Support telephone menu:

1. Select **2** for New Service Request.
2. Select **3** for Hardware, Networking and Solaris Operating System Support.
3. Select one of the following options:

- For Technical issues such as creating a new Service Request (SR), select **1**.
- For Non-technical issues such as registration or assistance with My Oracle Support, select **2**.

You are connected to a live agent who can assist you with My Oracle Support registration and opening a support ticket.

My Oracle Support is available 24 hours a day, 7 days a week, 365 days a year.

Customer Training

Oracle University offers training for service providers and enterprises. Visit our web site to view, and register for, Oracle Communications training at http://education.oracle.com/communication.

To obtain contact phone numbers for countries or regions, visit the Oracle University Education web site at www.oracle.com/education/contacts.

Emergency Response

In the event of a critical service situation, emergency response is offered by the Customer Access Support (CAS) main number at 1-800-223-1711 (toll-free in the US), or by calling the Oracle Support hotline for your local country from the list at http://www.oracle.com/us/support/contact/index.html. The emergency response provides immediate coverage, automatic escalation, and other features to ensure that the critical situation is resolved as rapidly as possible.

A critical situation is defined as a problem with the installed equipment that severely affects service, traffic, or maintenance capabilities, and requires immediate corrective action. Critical situations affect service and/or system operation resulting in one or several of these situations:

- A total system failure that results in loss of all transaction processing capability
- Significant reduction in system capacity or traffic handling capability
- Loss of the system's ability to perform automatic system reconfiguration
- Inability to restart a processor or the system
- Corruption of system databases that requires service affecting corrective actions
- Loss of access for maintenance or recovery operations
- Loss of the system ability to provide any required critical or major trouble notification

Any other problem severely affecting service, capacity/traffic, billing, and maintenance capabilities may be defined as critical by prior discussion and agreement with Oracle.

2

Installing Policy and Charging Rules Function (PCRF)

This section provides instruction for installing Policy and Charging Rules Function.

Installation Sequence

This section provides the order in which you shall perform the PCRF installation.

1. Create MySQL database. See Creating Database Account on MySQL Database
2. Download PCF package files and load them to the system. See Installation Preparation
3. Prepare all variables for Helm install command. See Variable Details
4. PCRF Deployment using Helm command. See Deploying Policy and Charging Rules Function
5. Verify PCRF Deployment. See Verifying PCRF Installation
6. Configure PCF. See Configuring Policy and Charging Rules Function (PCRF)
7. Verify PCRF configuration.

Pre-requisites

Following are the pre-requisites required for installing Policy and Charging Rules Function.

PCRF Software

PCRF software package includes:

- PCRF Helm Charts
- PCRF Docker Images

Software	Version
Kubernetes	v1.12.5
HELM	v2.11.0
MySQL	5.7 or later

Additional software that needs to be deployed as per the requirement of the services:

Software	Chart Version	Notes
elasticsearch	1.21.1	Needed for Logging Area
elastic-curator	1.2.1	Needed for Logging Area
elastic-exporter	1.1.2	Needed for Logging Area
logs	2.0.7	Needed for Logging Area

Software	Chart Version	Notes
kibana	1.5.2	Needed for Logging Area
grafana	2.2.0	Needed for Metrics Area
prometheus	8.8.0	Needed for Metrics Area
prometheus-node-exporter	1.3.0	Needed for Metrics Area
metallb	0.8.4	Needed for External IP
metrics-server	2.4.0	Needed for Metric Server
tracer	0.8.3	Needed for Tracing Area

Note:

In case any of the above services are needed and the respective software is not installed in CNE. Please install software before proceeding.

Note:

If you are using NRF, install it before proceeding with the PCRF installation.

Network access

The Kubernetes cluster hosts must have network access to:

- quay.io/datawire/ambassador docker image repository
- Local helm repository where the PCRF helm charts are available
- Local docker image repository where the PCRF images are available

Laptop/Desktop Client software

Following are the requirements for the laptop/desktop where the deployment commands shall be executed:

- Network access to the helm repository and docker image repository
- Network access to the Kubernetes cluster
- Necessary environment settings to run the 'kubectl' commands. The environment should have privileges to create namespace in the Kubernetes cluster.
- Helm client installed with the **push** plugin. The environment should be configured so that the `helm install` command deploys the software in the Kubernetes cluster.

Browser Support

It is recommend to use Firefox browser to access Kubernetes dashboard. The Configuration Management GUI page is accessed from different browsers.

Server or space requirements

For server and space requirements, refer to *Oracle Communications Cloud Native Environment Installation Guide*.

Creating Database Account on MySQL Database

To create an database account on MySQL database,

1. Navigate from SSH to MySQL database.
2. Login to database using the command,

   mysql - h<mysqlhost> -u<user> -p<password>

3. Execute the following command. The username and password is only provided as an example:

   CREATE USER 'pcrfusr'@'%' IDENTIFIED BY 'pcrfpasswd';

   GRANT ALL PRIVILEGES ON *.* TO 'pcrfusr'@'%';

Execute the following script to initial PCRF databases with above created database user. At first login to MySQL console via new user created above, `mysql -h<mysqlhost> -u<pcrfuser> -p<pcrfpassword>`

```
CREATE DATABASE IF NOT EXISTS `ocpm_config_server_pcrf`;

CREATE TABLE IF NOT EXISTS `ocpm_config_server_pcrf`.`topic_info` (
  `id` bigint(20) NOT NULL AUTO_INCREMENT,
  `description` varchar(255) COLLATE utf8_unicode_ci DEFAULT 'Default Topics.',
  `name` varchar(255) COLLATE utf8_unicode_ci DEFAULT NULL,
  `modify_date` datetime NOT NULL DEFAULT CURRENT_TIMESTAMP,
  `version` int(11) NOT NULL,
  PRIMARY KEY (`id`),
  UNIQUE KEY `UK_gd6b0a6mdpxc55qbibre2cldc` (`name`)
) AUTO_INCREMENT=3 DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci;
CREATE TABLE IF NOT EXISTS `ocpm_config_server_pcrf`.`configuration_item` (
  `id` bigint(20) NOT NULL AUTO_INCREMENT,
  `cfg_key` varchar(255) COLLATE utf8_unicode_ci DEFAULT NULL,
  `md5sum` varchar(255) COLLATE utf8_unicode_ci DEFAULT NULL,
  `cfg_value` mediumtext COLLATE utf8_unicode_ci,
  `version` int(11) NOT NULL,
  `topic_info_id` bigint(20) NOT NULL,
  PRIMARY KEY (`id`),
  KEY `FKdue8drxn6acrdt63iacireky1` (`topic_info_id`)
) DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci;

CREATE DATABASE IF NOT EXISTS `pcrf`;

insert into `ocpm_config_server_pcrf`.`topic_info` (name, version) values
('policy', 1);
insert into `ocpm_config_server_pcrf`.`topic_info` (name, version) values
('policySchema', 1);
insert into `ocpm_config_server_pcrf`.`topic_info` (name, version) values
('policyElement', 1);
insert into `ocpm_config_server_pcrf`.`topic_info` (name, version) values
('policyParam', 1);
insert into `ocpm_config_server_pcrf`.`topic_info` (name, version) values
('policygui', 1);

insert into `ocpm_config_server_pcrf`.`topic_info` (name, version) values
('amservice.system', 1);
insert into `ocpm_config_server_pcrf`.`topic_info` (name, version) values
('public.matchlist', 1);

insert into `ocpm_config_server_pcrf`.`topic_info` (name, version) values
```

```
('pe.serviceTag', 1);
insert into `ocpm_config_server_pcrf`.`topic_info` (name, version) values
('pe.policyTag', 1);
insert into `ocpm_config_server_pcrf`.`topic_info` (name, version) values
('pe.logLevel', 1);
insert into `ocpm_config_server_pcrf`.`topic_info` (name, version) values
('pcf.amservice.app', 1);
insert into `ocpm_config_server_pcrf`.`topic_info` (name, version) values
('nrfclient.cfg', 1);
insert into `ocpm_config_server_pcrf`.`topic_info` (name, version) values
('NRF.UDR', 1);
insert into `ocpm_config_server_pcrf`.`topic_info` (name, version) values
('NRF.BSF', 1);
insert into `ocpm_config_server_pcrf`.`topic_info` (name, version) values
('pcf.userservice.cfg', 1);
insert into `ocpm_config_server_pcrf`.`topic_info` (name, version) values
('NRF.CHF', 1);
insert into `ocpm_config_server_pcrf`.`topic_info` (name, version) values
('pcf.chfservice', 1);

insert into `ocpm_config_server_pcrf`.`topic_info` (name, version) values
('pcrf.public.trafficprofile.pccrule', 1);
insert into `ocpm_config_server_pcrf`.`topic_info` (name, version) values
('public.datamodel', 1);
insert into `ocpm_config_server_pcrf`.`topic_info` (name, version) values
('public.policy.test', 1);
insert into `ocpm_config_server_pcrf`.`topic_info` (name, version) values
('public.policy.project.content.pcrfCoreService', 1);
insert into `ocpm_config_server_pcrf`.`topic_info` (name, version) values
('pcrf.public.networkelement.pgw', 1);
insert into `ocpm_config_server_pcrf`.`topic_info` (name, version) values
('pcrf.public.networkelement.ggsn', 1);
insert into `ocpm_config_server_pcrf`.`topic_info` (name, version) values
('pcrf.public.retryprofile.pcc', 1);
```

Result: The database account is created. Execute the command, `show databases` to view database.

Installation Preparation

The following procedure describes the steps to download the PCRF Images and Helm files from OSDC.

1. Download the PCRF package file from Oracle Software Delivery Cloud (OSDC). Package is named as follows:
   `<nfname>-pkg-<marketing-release-number>.tgz`

   For example, `ocpcrf-pkg-12.5.2.0.0_1.0.0.tgz`
2. Extract the PCRF Package File via "tar".
   `tar -xvf <<nfname>-pkg-<marketing-releasenumber>>.tgz`

   This command results into `<<nfname>-pkg-<marketingrelease-number>>` directory.

   The directory consists of following:

   - **PCF Docker Images File:**
     `ocpcf-images-12.5.2.0.0_1.0.0.tar`
   - **Helm File:**

`ocpcf-12.5.2.0.0_1.0.0.tgz`

- **Readme txt File:**
  `Readme.txt` (Contains cksum and md5sum of tarballs)

3. Verify the checksums of tarballs mentioned in `Readme.txt`.

Deploying Policy and Charging Rules Function

The Policy and Charging Rules Function requires a MySQL database to store the configuration and run time data.

PCRF Software components as mentioned in pre-requisites section, can be extracted following the below steps.

1. Download the file, **ocpcrf-pkg-12.5.2.0.0_1.0.0.tgz**.
2. Untar **ocpcrf-pkg-12.5.2.0.0_1.0.0.tgz** to get PCRF docker image tar file then push it to customer docker registry.
3. Untar displays the following files:

```
ocpcrf-pkg-12.5.2.0.0_1.0.0.tgz
          |_ _ _ _ _ _ ocpcrf-images-12.5.2.0.0_1.0.0.tar
          |_ _ _ _ _ _ ocpcrf-12.5.2.0.0_1.0.0.tgz
          |_ _ _ _ _ _ Readme.txt
```

4. Run the following command to load **ocpcrf-images-12.5.2.0.0_1.0.0.tgz** to docker:

```
docker load --input /<IMAGE_PATH>/ocpcrf-images-12.5.2.0.0_1.0.0.tar
```

After executing above command, run "docker images" to view imported PCRF docker images, then create a new tag fo reach imported image and push it to customer docker registry

```
docker tag ocpcrf/diam-gateway:12.5.2.0.0_1.0.0 <CUSTOMER_REPO>/diam-
gateway:<IMAGE_TAG>
docker push <CUSTOMER_REPO>/diam-gateway:<IMAGE_TAG>

docker tag ocpcrf/pcrf_core:12.5.2.0.0_1.0.0 <CUSTOMER_REPO>/
pcrf_core:<IMAGE_TAG>
docker push <CUSTOMER_REPO>/pcrf_core:<IMAGE_TAG>


docker tag ocpcrf/ocpm_cm_service:12.5.2.0.0_1.0.0 <CUSTOMER_REPO>/
ocpm_cm_service:<IMAGE_TAG>
docker push <CUSTOMER_REPO>/ocpm_cm_service:<IMAGE_TAG>


docker tag ocpcrf/ocpm_config_server:12.5.2.0.0_1.0.0 <CUSTOMER_REPO>/
ocpm_config_server:<IMAGE_TAG>
docker push <CUSTOMER_REPO>/ocpm_config_server:<IMAGE_TAG>


docker tag ocpcrf/ocpm_pre:12.5.2.0.0_1.0.0 <CUSTOMER_REPO>/
ocpm_pre:<IMAGE_TAG>
docker push <CUSTOMER_REPO>/ocpm_pre:<IMAGE_TAG>


docker tag ocpcrf/diam-sim-server:12.5.2.0.0_1.0.0 <CUSTOMER_REPO>/diam-sim-
server:<IMAGE_TAG>
docker push <CUSTOMER_REPO>/diam-sim-server:<IMAGE_TAG>
```

```
docker tag ocpcrf/readiness-detector:latest <CUSTOMER_REPO>/diam-
gateway:latest
docker push <CUSTOMER_REPO>/diam-gateway:latest
```

> **Note:**
>
> User may need to configure docker certificate to access customer registry via HTTPS. Configure the certificate before executing the docker push command or the command may fail to execute.

After executing the above commands, below docker images should available for PCRF.

Table 2-1 Docker Image Name

Service Name	Docker Image Name	Service Category
Pcrf core Service	pcrf_core	PCRF
Diameter Gateway	diam-gateway	PCRF
Diameter Simulator	diam-sim-server	PCRF
Readiness check	readiness-detector	Common
CM Service	ocpm_cm_service	Common
Config Server Service	ocpm_config_server	Common
Policy Runtime Service	ocpm_pre	Common

Following table provides description of the Service Category:

Table 2-2 Service Category

Service Category	Description
Common	Common service module which would be consumed by PCRF as supporting service, such as GUI, policy engine
PCRF	Define PCRF services

5. Navigate to helm chart folder (refer to step 2 to get helm chart), and execute the following command to install PCRF:

> **Note:**
>
> It is mandatory to run following command under helm chart folder as the last line, `./<HELM_CHART_NAME_WITH_EXTENSION>` specifies that helm chart path is the current working path. If you want to run the command in another server, copy the helm chart file to that server folder, before executing the command from that server.

```
helm install --namespace=<PCRF_NAMESPACE>-pcrf --name=<PCRF_NAME>-pcrf \
--set
global.envMysqlHost=<MYSQL_HOST>,global.envMysqlUser=pcrfusr,global.envMysqlPa
ssword=pcrfpasswd \
```

```
--set global.envJaegerAgentHost=<JAEGER_SERVICE>.<JAEGER_SERVICE_NAMESPACE> \
--set global.envManageNF=PCF,global.envSystemName=PCRF
--set global.imageTag=<IMAGE_TAG>,global.dockerRegistry=<CUSTOMER_REPO> \
--set common.deploymentPcrfConfig.envMysqlDatabase=ocpm_config_server_pcrf \
./<HELM_CHART_NAME_WITH_EXTENSION>
```

> **Note:**
>
> Before proceeding with the installation, setup the variables. See Variable Details.

Table 2-3 Variable Details

Variable	Description	Notes
<PCRF_NAMESPACE>	Indicates deployment PCRF namespace used by helm command	Variable name can include uppercase and lower case alphabets, numbers, and special characters _, and -. Maximum allowed character length is 10.
<PCRF_NAME>	Indicates deployment PCRF name used by helm command	
<MYSQL_HOST>	MySQL host name or IP address	global.envMysqlUser and global.envMysqlPassword variables in above command from database section configured in previous step
<JAEGER_SERVICE> <JAEGER_SERVICE_NAMESPACE>	Both parameters could be found in same Kubernetes cluster.	Use the following format "<JAEGER_AGENT_SERVICE_NAME>.<JAEGER_NAMESPACE>" Such as "occne-tracer-jaeger-agent.occne-infra", occne-tracer-jaeger-agent is jaeger agent service name under jaeger deployment

Table 2-3 (Cont.) Variable Details

Variable	Description	Notes
<IMAGE_TAG>	The image tag used in customer docker registry. It is recommend to use the same image tag when pulling docker image to registry. If followed above steps to push docker image to customer docker registry then the <IMAGE_TAG> value should be 12.5.2.0.0_1.0.0	Each service deployment yaml file would use global.imageTag as image tag to fetch related docker image per helm chart design, if one service cannot use global image tag then please edit that service part under values.yaml, such as below: Before update: image: bsf_management_service imageTag: " After update: image: bsf_management_service imageTag: <IMAGE_TAG>
<CUSTOMER_REPO >	Indicates the docker registry address and the Port Number, if registry has port attached	If registry has port value, add port as well, such as "reg-1:5000"
<METALLB_ADDRESS_POOL>	The address pool which configured on metallb to provide external IPs	

Kubernetes provides the following three deployment types:

Table 2-4 PCRF Service Deployment Service Type

Service Type	Description
ClusterIP	Exposes the service on a cluster-internal IP. Choosing this value makes the service only reachable from within the cluster. This is the default ServiceType
NodePort	Exposes the service on each Node's IP at a static port (the NodePort). A ClusterIP service, to which the NodePort service routes, is automatically created. User will be able to contact the NodePort service, from outside the cluster, by requesting <NodeIP>:<NodePort>. Most PCRF service use NodePort to deploy.
LoadBalancer	Exposes the service externally using a cloud provider's load balancer. NodePort and ClusterIP services, to which the external load balancer will route, are automatically created. For GUI page and API gateway service, For GUI page and API gateway service, it is mandatory to use loadBalancer type. Given latest OCCNE already integrated METALLB, configure IP address to METALLB on OCCNE.

Verifying PCRF Installation

Run the following command to verify the PCRF installation:

```
kubectl get svc -n <PCRF-Namespace>
kubectl get pod -n <PCRF-Namespace>
```

If the installation is successful, all the pods should be in Running/Completed status. If any pod is found with error status, you can check pod log to view the error details.

3

Configuring Policy and Charging Rules Function (PCRF)

> **Note:**
>
> Before configuring, verify the installation.

Click the PODS to ensure all ports are active and running.

The following subsections provides the information for configuring PCRF.

Enabling LoadBalancer with MetalLB

Cloud Native Network already have MetalLB installed, and free external IPs are already configured under MetalLB.

Perform the following steps to enable LoadBalancer to specific services.

> **Note:**
>
> In PCRF namespace, only diam-Gateway service and cm service with GUI page requires load-balancer setting with accessible external IP.

Updating diam-gateway Service

To update diam-gateway service:

1. Login to Kubernetes cluster master node using ssh command.
2. Run the following command to edit svc yaml file for diam-gateway:

   ```
   kubectl edit svc <PCRF_NAME>-diam-gateway-service -n <PCRF_NAME_SPACE>
   ```

Table 3-1 Variables

Variable Name	Description
PCRF_NAME	The --name value used in helm install command
PCRF_NAME_SPACE	The --namespace value used in helm install command

Following is an sample content that displays in diam-gateway edit window.

```
1 # Please edit the object below. Lines beginning with a '#' will be ignored,
 2 # and an empty file will abort the edit. If an error occurs while saving
this file will be
```

```
 3 # reopened with the relevant failures.
 4 #
 5 apiVersion: v1
 6 kind: Service
 7 metadata:
 8   creationTimestamp: 2019-06-02T13:06:11Z
 9   labels:
10     category: common
11     io.kompose.service: <PCRF_NAME>-pcrf-diam-gateway-service
12   name: <PCRF_NAME>-pcrf-diam-gateway-service
13   namespace: <PCRF_NAME_SPACE>
14   resourceVersion: "21624671"
15   selfLink: /api/v1/namespaces/<PCRF_NAME_SPACE>/services/<PCRF_NAME>-
pcrf-diam-gateway-service
16   uid: 31a4b13f-8537-11e9-81c8-0010e08b3a8e
17 spec:
18   clusterIP: 10.20.37.37
19   externalTrafficPolicy: Cluster
20   ports:
21   - name: diameter
22     nodePort: 32592
23     port: 3868
24     protocol: TCP
25     targetPort: 3868
26   - name: http
27     nodePort: 31301
28     port: 8080
29     protocol: TCP
30     targetPort: 8080
31   selector:
32     io.kompose.service: <PCRF_NAME>-pcrf-diam-gateway-service
33   sessionAffinity: None
34   type: NodePort
35 status:
36   loadBalancer: {}
```

3. Add two new lines after line 7, after "metadata":

   `annotations:`

   `metallb.universe.tf/address-pool: <ADDRESS_POOL_NAME>`

   > **Note:**
   >
   > - As per user MetalLB setting, you should select an appropriate pool name to replace the variable, *<ADDRESS_POOL_NAME>*
   > - *annotation:* line must be kept vertical align with line 16, while following line, *metallb.universe.tf/address-pool: <ADDRESS_POOL_NAME>* must be kept vertical align with line 10. If vertical align restriction failed to follow this rule, the svc yaml file update may fail.

4. Replace line 34 text, **type: NodePort** with **type: LoadBalancer**.
   Following is the sample content after replacing the line 29:

```
1 # Please edit the object below. Lines beginning with a '#' will be ignored,
 2 # and an empty file will abort the edit. If an error occurs while saving
this file will be
 3 # reopened with the relevant failures.
```

```
 4 #
 5 apiVersion: v1
 6 kind: Service
 7 metadata:
 8   creationTimestamp: 2019-06-02T13:06:11Z
 9   labels:
10     category: common
11     io.kompose.service: <PCRF_NAME>-pcrf-diam-gateway-service
12   name: <PCRF_NAME>-pcrf-diam-gateway-service
13   namespace: <PCRF_NAME_SPACE>
14   resourceVersion: "21624671"
15   selfLink: /api/v1/namespaces/<PCRF_NAME_SPACE>/services/<PCRF_NAME>-
pcrf-diam-gateway-service
16   uid: 31a4b13f-8537-11e9-81c8-0010e08b3a8e
17 spec:
18   clusterIP: 10.20.37.37
19   externalTrafficPolicy: Cluster
20   ports:
21   - name: diameter
22     nodePort: 32592
23     port: 3868
24     protocol: TCP
25     targetPort: 3868
26   - name: http
27     nodePort: 31301
28     port: 8080
29     protocol: TCP
30     targetPort: 8080
31   selector:
32     io.kompose.service: <PCRF_NAME>-pcrf-diam-gateway-service
33   sessionAffinity: None
34   type: LoadBalancer
35 status:
36   loadBalancer: {}
```

5. Quit vim editor and save changes. A new diam-gateway pod starts up.
   a. In the new pod, following sample content displays. Note that if the EXTERNAL-IP is available, then the load balancer setting for diam-gateway service works.

```
NAME                          TYPE           CLUSTER-IP      EXTERNAL-IP
PORT(S)                                                        AGE
<PCRF_NAME>-diam-gateway-service       LoadBalancer   10.xxx.xx.xx
10.xxx.xxx.xx   3868:32592/TCP,8080:31301/TCP 4d
```

Updating cm-service

Follow the same process to update svc yaml for **<PCRF_NAME> -pcrf-cm-service**.

4

Upgrading Policy and Charging Rules Function (PCRF)

User can perform the helm upgrade command in the following scenarios.

- Update an existing parameter settings.
- Add more parameters as per the requirement

To upgrade PCRF:

```
helm upgrade <PCRF_NAME>-pcrf \
--set
global.envMysqlHost=<MYSQL_HOST>,global.envMysqlUser=pcrfusr,global.envMysqlPassw
ord=pcrfpasswd \
--set global.envJaegerAgentHost=<JAEGER_SERVICE>.<JAEGER_SERVICE_NAMESPACE> \
--set global.envManageNF=PCRF,global.envSystemName=PCRF\
--set global.imageTag=<IMAGE_TAG>,global.dockerRegistry=<CSTOMER_REPO> \
--set common.deploymentPcrfConfig.envMysqlDatabase=ocpm_config_server_pcrf \
./<HELM_CHART_NAME_WITH_EXTENSION>
```

> **Note:**
>
> <PCRF_NAME> must be same as helm install command

> **Note:**
>
> The upgrade command is similar to install command, because, if user do not specify the same parameters for both upgrade and install, then the settings applied by install command may lost and use default settings from **values.yaml** file for missing parameters in upgrade command.

For specific deployment, few parameters cannot be updated. Following table provides details of the parameters which cannot be updated.

Table 4-1 Parameters

Deploymnet	Parameter	Description
PCRF	global.envManageNF=PCRF,global.envSystemName=PCRF	Since GUI service is a common service which defined under common module, it requires to startup parameters to show which NF should be showed under specific deployment.

MetalLB Settings for Upgrade

After executing the helm upgrade command, the configured MetalLB settings may lost. User is required to update the settings manually by following the procedure in the Enabling LoadBalancer with MetalLB.

Verifying PCRF Upgrade

Run the following command to verify the PCRF upgrade:

```
kubectl get pod -n <PCRF-Namespace>
```

You can verify the below items from the output of above command:

- All pods under PCRF namespace should either in Running status or in Completed status. If any pod with error status found, check pod log to view the error details.
- For updated service per helm upgrade setting, check its RESTART output and AGE output, if specific had been updated then the old pod should be killed and a new pod should bring up. So, the related RESTART value should be 0 and AGE value should be 3-4 seconds if you check upgrade result with above command soon after helm upgrade .
- diam-gateway service and CM service should support metalLB setting.

5

Uninstalling Policy and Charging Rules Function (PCRF)

To uninstall or completely delete the Policy and Charging Rules Function deployment, execute the following command:

```
helm delete --purge <PCRF-Name>
kubectl delete namespace <PCRF-Namespace>
```

After executing the above commands, no PCRF deployment would be found under below commands output:

- helm list
- kubectl get namespace