Oracle® Communications

InterWorking and Mediation Function (IWF) Cloud Native Installation Guide

Release 1.4
F31795-01
May 2020

Oracle Communications InterWorking and Mediation Function (IWF) Cloud Native Installation Guide, Release 1.4

F31795-01

Contents

A Appendix IWF Yaml Files

List of Tables

1 OCIWF Installation Overview

This sections provides overview information on IWF.

The Oracle Communications 5G InterWorking and Mediation Function (IWF) enables 5G Core NFs to communicate with EPC network elements. IWF is a Cloud-Native solution based on micro-services architecture that is deployed as an independent network function in the 5GC network or as an independent micro service within the 5GC NF, as a part of Oracle 5GC NFs like NRF, SEPP, SCP, and so on.

IWF includes the following features:

- Protocol Translation:
  - IWF allows 5GC NF to interwork with the EPC network elements or vice versa.
  - IWF supports protocol conversion capabilities from Diameter to HTTP/2 and vice versa.
- Message Mediation:
  - This feature allows API transformation to resolve inter-NF inter operational issues.
  - Allows users to create policy rules to execute mediation transformation.
  - It supports mediation on HTTP 1.1 and 2 protocol.
  - It supports Forward proxy, Reverse proxy and end-point.

References

- CNE Install Guide
- IWF User's Guide

Acronyms and Terminologies

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

Table 1-1 Acronyms and Terminologies

Field	Description
5GC	5G Core Network
API	Application Program Interface
CNE	Cloud Native Environment
DWR	Device Watchdog Request
EPC	Evolved Packet Core
IWF	InterWorking and Mediation Function

Table 1-1 (Cont.) Acronyms and Terminologies

Field	Description
NEF	Network Exposure Function
NF	Network Function
NRF	Network Repository Function
OSDC	Oracle Software Delivery Cloud
SCP	Service Communication Proxy
SEPP	Security Edge Protection Proxy
UDP	User Defined Protocol

Documentation Admonishments

Admonishments are icons and text throughout this manual that alert the reader to assure personal safety, to minimize possible service interruptions, and to warn of the potential for equipment damage.

Table 1-2 Admonishments

Icon	Description
DANGER	Danger: (This icon and text indicate the possibility of personal injury.)
WARNING	Warning: (This icon and text indicate the possibility of equipment damage.)
CAUTION	Caution: (This icon and text indicate the possibility of service interruption.)

Locate Product Documentation on the Oracle Help Center Site

Oracle Communications customer documentation is available on the web at the Oracle Help Center 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 **Oracle Communications documentation** link.

   The Communications Documentation page displays.

4. Click on your product and then the release number.

   A list of the documentation set for the selected product and release displays.

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.

Customer Training

Oracle University offers training for service providers and enterprises. Visit our web site to view, and register for, Oracle Communications training:

http://education.oracle.com/communication

To obtain contact phone numbers for countries or regions, visit the Oracle University Education web site:

www.oracle.com/education/contacts

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.

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

IWF Pre-requisites

This section includes information about the required environment for IWF deployment.

IWF Software

The IWF software includes:

- IWF Helm chart
- IWF docker images

The following software must be installed:

Software	Version
Kubernetes	v 1.15.3
HELM	v 2.14.3

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
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:**
>
> Install the specified software items before proceeding if any of the above services are needed and the respective software is not already installed in CNE.

Network access

The Kubernetes cluster hosts must have network access to:

- Local docker image repository where the IWF images are available.
- Local helm repository where the IWF helm charts are available.

> **Note:**
>
> All the kubectl and helm related commands that are used in this document must be executed on a system depending on the infrastructure of the deployment. It could be a client machine, VM, server, local desktop and so on.

Client machine requirement

There are some requirements for the client machine where the deployment commands need to be executed:

- It should have network access to the helm repository and docker image repository.
- Helm repository must be configured on the client.
- It should have network access to the Kubernetes cluster.
- It should have necessary environment settings to run the `kubectl` commands. The environment should have privileges to create namespace in the Kubernetes cluster.
- It should have helm client installed. The environment should be configured so that the `helm install` command deploys the software in the Kubernetes cluster.
- Master/Management node must have the `jq` tool installed to be used by configuration map content parsing tools.
- Operator must create MySQL NDB database cluster to store the configuration and run time data.

3 IWF Installation Sequence

This section informs about the installation sequence for IWF installation.

Table 3-1   IWF Installation Sequence

SL No.	Installation sequence
1 □	Installation Preparation
2 □	Configure `custom_values.yaml` file. This includes configuring the following based on the deployment: 1. Repository path 2. IWF details.
3 □	Deploy IWF. IWF can be deployed in either of the following ways: 1. With HELM repository 2. With Local repository
4 □	Verify IWF deployment

IWF Images

Micro service	Image
`ociwf-iwf-mediation`	`ocmed-iwfmediation:1.4.0`
`ociwf-iwf-diameterproxy`	`ociwf-iwfdiamproxy:1.4.0`
`ociwf-iwf-d2h`	`ociwf-iwfd2h:1.4.0`
`ociwf-iwf-h2d`	`ociwf-iwfh2d:1.4.0`
`ociwf-iwf-nrfclient`	`ociwf-iwfnrfclient:1.4.0`
`ociwf-nf-mediation`	`oocmed-nfmediation:1.4.0`
`ociwf-pcf-diam-gateway`	`diam-gateway:1.4.0`
`ociwf-iwf-pcfdiscovery`	`oociwf-iwfpcfdiscovery:1.4.0`
`iwf-mediation-test`	`ocmed-iwfmediation:1.4.0`
`nf-mediation-test`	`ocmed-nfmediation:1.4.0`

4
IWF Installation Preparation

This section includes information about the preparation required before IWF installation.

For more information about configuring docker image and registry, see chapter OCCNE Docker Image Registry Configuration in OCCNE Installation Guide.

Table 4-1 IWF Installation Preparation

Step	Procedure	Description

1
□
	Download the OCIWF package	Customers are required to download the OCIWF package from Oracle Software Delivery Cloud (OSDC). Package is named as follows:
		`<nfname>-pkg-<marketing-release-number>.tgz`
		`For example:ociwf-pkg-1.4.0.0.0.tgz`

2 □	Untar the OCIWF Package	Untar the OCIWF package into a specific repository: `tar -xvf <<nfname>-pkg-<marketing-release-number>>` The package file consists of following: 1. OCIWF Docker Images tar ociwf-images-1.4.0.tar 2. Helm File ociwf-1.4.0.tgz 3. Readme txt file `Readme.txt` (Contains cksum and md5sum of tarballs)
3 □	Check the checksums	Check the checksums of tarballs mentioned in `Readme.txt`. Refer to the `Readme.txt` file for commands and checksum details.
4 □	Load the tarball to system	Execute the following command to push the Docker images to docker registry: `docker load --input ociwf-images-1.4.0.tar`

Table 4-1 (Cont.) IWF Installation Preparation

Step	Procedure	Description
5 □	Check if all the images are loaded	Execute the following command to check: `docker images` Refer table **IWF Images** in section IWF Installation Sequence for the list of images.

6
□
	Push docker images to docker registry	Execute the following commands to push the docker images to docker registry:
		`docker tag <image-name>:<image-tag> <docker-repo>/<image-name>:<image-tag>`
		`docker push <docker-repo>/<image-name>:<image-tag>`

7 □	Untar Helm Files	Untar the helm files: `tar -xvzf ociwf-1.4.0.tgz`
8 □	Download the InterWorking and Mediation Function (IWF) Custom Template ZIP file	Download the **InterWorking and Mediation Function (IWF) Custom Template** ZIP file from OHC: • Go to the URL, docs.oracle.com • Navigate to **Industries->Communications->Diameter Signaling Router->Cloud Native Network Elements** • Click the **InterWorking and Mediaiton Function (IWF) Custom Template** link to download the zip file. • Unzip the template to get `ociwf-custom-values-1.4.0.yaml` file.

5 IWF Installation

This section includes information about the OCIWF installation procedures.

IWF Deployment Procedure

The following procedure guides you through installation of OCIWF on CNE.

Table 5-1 OCIWF Deployment

Step #	Procedure	Description

1
□
	Create Database User/Group	Create User with permission to access the tables on all the SQL nodes present in the NDB cluster, by executing:
		Note:
		• The OCIWF uses a MySQL database to store the configuration and run time data.
		• The OCIWF deployment using MySQL NDB cluster requires the database administrator to create user in MYSQL DB and to provide the user with necessary permissions to access the tables in the NDB cluster.
		1. Login to the server where the ssh keys are stored and SQL nodes are accessible.
		2. Connect to the SQL nodes.
		`ssh <USERNAME>@<HOSTNAME>`
		3. Login to the MYSQL as a root user:
		`/usr/local/mysql/bin/mysql -h 127.0.0.1 -u root -p <password>`
		4. Create MYSQL user:
		`CREATE USER '<USERNAME>'@'%' IDENTIFIED BY '<PASSWORD>';`
		`DROP DATABASE if exists diameter;`
		`CREATE DATABASE diameter CHARACTER SET utf8;`
		`GRANT SELECT, INSERT, CREATE, ALTER, DROP, LOCK TABLES,`
		`CREATE TEMPORARY TABLES, DELETE, UPDATE,`
		`EXECUTE ON diameter.* TO '<USERNAME>'@'%';`
		5. Execute the following commands on one of the NDB SQL node:
		a. Log into the MYSQL user created in the previous step:
		`/usr/local/mysql/bin/mysql -h 127.0.0.1 -u <USERNAME> -p <PASSWORD>`
		b. Create MYSQL table:
		`USE diameter;`
		`CREATE TABLE IF NOT EXISTS SESSION_CORRELATION (`
		`SESSION_ID varchar(255) NOT NULL,`
		`RESOURCE_ID varchar(255) NOT NULL UNIQUE,`
		`PEER_IDENTITY varchar(255) NOT NULL,`
		`PEER_REALM varchar(255) NOT NULL,`
		`REQUEST_COUNT int(11),`
		`PRIMARY KEY (SESSION_ID)`
		`) ENGINE=NDBCLUSTER DEFAULT CHARSET=utf8;`

Table 5-1 (Cont.) OCIWF Deployment

Step #	Procedure	Description
		Note: The `<username>` and `<password>` is created by the Database Administrator. 6. Exit from database and logout from SQL node.
2 □	Customize `ociwf-custom-values.yaml` file	Customize `ociwf-custom-values.yaml` file as per the deployment requirement: Update service ports accordingly. For more information, see IWF Installation Preparation. To configure the parameters, see section IWF Configuration or, The `ociwf-custom-values-1.4.0.yaml` template can be downloaded from OHC. Download the InterWorking and Mediation Function (IWF) Custom Template ZIP file and Unzip to get `ociwf-custom-values-1.4.0.yaml` file.
3 □	Perform the Diameter configuration	Configure diameter peer(s) in the following file: `ociwf/charts/pcf/templates/configmap-pcf-diam-gateway-service-diameter.yaml` Refer, **IWF User guide** for diameter peer configuration details.
4 □	Deploy IWF from Helm repository	To deploy IWF from helm repository, execute: `helm install ociwf/ -f <ociwf-custom-values.yaml> --name <helm-release> --namespace <k8s namespace> --version <ociwf version>` **For example**: `helm install ociwf-helm-repo/ociwf -f ociwf-custom-values.yaml --name ociwf --namespace iwfsvc --version <ociwf version>` or,
5 □	Deploy IWF from local repository	To deploy IWF from local repository, execute: `helm install ociwf -f <ociwf-custom-values.yaml> --name <helm-release> --namespace <k8s namespace>` **For example**: `helm install ociwf -f ociwf-custom-values.yaml --name ociwf --namespace iwfsvc`
6 □	Check status of the services	Execute the following command: `kubectl get services -n <namespace>` **For example**: `kubectl get services -n iwfsvc` **Note:** If `metallb` is used, `EXTERNAL-IP` is assigned to `ociwf-endpoint`.

Table 5-1 (Cont.) OCIWF Deployment

<table>
<tr><th>Step #</th><th>Procedure</th><th>Description</th></tr>
<tr><td>7
□</td><td>Check status of the pods</td><td>

Execute the following command:

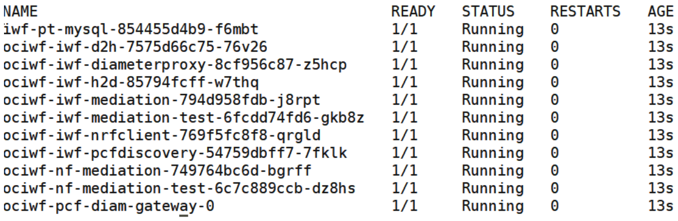
`kubectl get pods -n <ociwf_namespace>`

Status column of all the pods should be 'Running'.

Ready column of all the pods should be n/n, where n is number of containers in the pod.

For example:

`kubectl get pods -n iwfsvc`



```
NAME                                          READY   STATUS    RESTARTS   AGE
iwf-pt-mysql-854455d4b9-f6mbt                 1/1     Running   0          13s
ociwf-iwf-d2h-7575d66c75-76v26                1/1     Running   0          13s
ociwf-iwf-diameterproxy-8cf956c87-z5hcp       1/1     Running   0          13s
ociwf-iwf-h2d-85794fcff-w7thq                 1/1     Running   0          13s
ociwf-iwf-mediation-794d958fdb-j8rpt          1/1     Running   0          13s
ociwf-iwf-mediation-test-6fcdd74fd6-gkb8z     1/1     Running   0          13s
ociwf-iwf-nrfclient-769f5fc8f8-qrgld          1/1     Running   0          13s
ociwf-iwf-pcfdiscovery-54759dbff7-7fklk       1/1     Running   0          13s
ociwf-nf-mediation-749764bc6d-bgrff           1/1     Running   0          13s
ociwf-nf-mediation-test-6c7c889ccb-dz8hs      1/1     Running   0          13s
ociwf-pcf-diam-gateway-0                      1/1     Running   0          13s
```

</td></tr>
</table>

6 IWF Uninstallation

Deleting the IWF deployment

To completely delete or remove the IWF deployment, execute:

```
helm delete --purge <helm-release>
```

For example:

```
helm delete --purge ociwf
```

Delete kubernetes namespace

```
kubectl delete namespace <ocnssf kubernetes namespace>
```

For example:

```
kubectl delete namespace iwfsvc
```

7 IWF Upgrade

This section includes information about upgrading the existing IWF deployment.

Note: The `ociwf_customized_values.yaml` must be updated with the required details for upgrade.

The `ociwf_customized_values.yaml` can be downloaded from the OHC.

To download the **InterWorking and Mediation Function (IWF) Custom Template** ZIP file from OHC:

- Go to the URL, docs.oracle.com
- Navigate to **Industries->Communications->Diameter Signaling Router->Cloud Native Network Elements**
- Click the **InterWorking and Mediation Function (IWF) Custom Template** link to download the zip file.
- Unzip the template to get `ociwf-custom-values-1.4.0.yaml` file.

To upgrade an existing IWF deployment, execute:

```
helm upgrade <release> -f <ociwf-custom-values-1.4.0.yaml>
```

In case of backout:

1. Check the history of helm deployment:

   ```
   helm history <helm_release>
   ```

2. Rollback to the required revision:

   ```
   helm rollback <release name> <revision number>
   ```

8

IWF Configuration

This section includes configuration options supported for IWF.

For detailed information about configuring parameters, see IWF Configurable Parameters.

Download the InterWorking and Mediation Function (IWF) Custom Template ZIP file from OHC and Unzip the template to get `ociwf-custom-values-1.4.0.yaml` file.

Updating Mediation Rules

- A script file is available in `tools/` directory, which downloads the required rule files. It needs namespace as well as configmap name. These rules then can be changed accordingly.
- After changing the rules, the user must reload the changes by executing: `"kubectl create configmap <configmapname> -n <namespace> --dry-run -o yaml --from-file="<filename>" | kubectl replace -f -"`
- The test rules can also be modified similarly.

Mediation Test Mode configuration

IWF Mediation can be configured to send request and response to IWF Mediation Test Service by setting `test-mode-enable` to `enable` in environment variable in the IWF Mediation deployment chart.

MYSQL Configuration

The MYSQL allows the operator to configure the database and store the association between the N5 and Rx session.

- Configurable options supported through custom values file `ociwf-custom-values-1.4.0.yaml`.
  1. Using CNE Mysql cluster
     a. Edit `iwf-mysql:enabled`: false (Disables the mysql pod)
     b. Edit `iwf-diameterproxy`:
        Enter the IP of the primary and secondary mysql nodes in the mysql cluster deployed in CNE.

        ```
        dpDBService1: xyz.xyz.xyz.xyz
        ```

        ```
        dpDBService2: xyz.xyz.xyz.xyz
        ```

        > **Note:**
        >
        > Along with IP, the DB cluster credentials should be configured.

  2. Local mysql pod

> **Note:**
>
> These steps are only applicable for deployment and testing scenario.

a. Edit `iwf-mysql:enabled`: true (It brings up the mysql local pod deployment)

b. Edit `iwf-diameterproxy`:

```
dpDBService1: iwf-pt-mysql-svc
```

```
dpDBService2: iwf-pt-mysql-svc
```

This enables the mysql deployment as your database source instead of the cluster.

Reloading ConfigMap

Reloading the IWF-Mediation ConfigMap, if in case any updates were made in the Rules.

The tool `ociwf-rule-download_tool.sh` can be used to download the rule configmap in a folder (folder name will be configmap name). It needs namespace as well as required configmap name. These rules then can be changed accordingly.

The tool `ociwf-rule-upload_tool.sh` can be used to upload the rule config map from the existing config map folder. It needs namespace as well as required configmap name to be uploaded.

- If rules were changed on iwf active mediation, then use `ociwf-iwf-mediation-config-active` as the name of configmap.
- If rules were changed on iwf test mediation then use `ociwf-iwf-mediation-config-test` as the name of the configmap.

Reloading the NF-Mediation ConfigMap, if in case any updates were made in the Rules.

- The tool `ociwf-rule-download_tool.sh` can be used to download the rule configmap in a folder (folder name will be configmap name). It needs namespace as well as required configmap name. These rules then can be changed accordingly.
- The tool `ociwf-rule-upload_tool.sh` can be used to upload the rule config map from the existing config map folder. It needs namespace as well as required configmap name to be uploaded
- If rules were changed on nf active mediation, then execute `ociwf-nf-mediation-config-active` as the name of configmap
- If the rules were changed on nf test mediation, then execute: `ociwf-nf-mediation-config-test` as the name of the configmap.

IWF-Mediation Rule Configuration for PT

Add below rules to the mediation rules:

Note:

In the below rules, replace `PCF_loadBalancerIP_and_Port` and `IWF_loadBalancerIP_and_Port` with appropriate values.

```
rule "pt_d2h_AAR_rule_N5_To_Rx"
salience 20
when
m : IWFHttpRequest(header("pt_dest_uri") != null)
then
m.setUri(m.header("pt_dest_uri").replace("pcf.com","10.178.246.13:1000/
simulation"))   //"PCF_loadBalancerIP_and_Port"
m.header(m.DEL,"pt_dest_uri")
end

rule "pt1_h2d_RAR_rule_N5_To_Rx"
salience 20
when
  m : IWFHttpRequest(getUri() matches ".*(npcf-
policyauthorization)*(v1)*(notification)*(notify).*")
then
 m.forwardPath = IWFConsts.FORWARD_TO_H2D
 m.header(m.ADD,"diameterApplicationId","16777236")
 m.header(m.ADD,"diameterCommandCode","258")
 m.header(m.ADD,"original-req-uri",m.getUri())
end

rule "pt1-d2h_ASR_rule_N5_To_Rx"
salience 20
when
m : IWFHttpRequest(getUri() matches ".*(npcf-
policyauthorization)*(v1)*(notification)*(terminate).*")
then
m.forwardPath=IWFConsts.FORWARD_TO_H2D
m.header(m.ADD,"diameterApplicationId","16777236")
m.header(m.ADD,"diameterCommandCode","274")
m.header(m.ADD,"original-req-uri",m.getUri())
end

rule "pt_d2h_AAR_Update_Rx_To_N5"
salience 21
when
m : IWFHttpRequest(m.body("ascReqData:notifUri") != null)
then
m.body(m.UPDATE,"ascReqData:notifUri",update(m.body("ascReqData:notifUri"))
.replace("iwf.com","10.178.254.158:30079"))  //IWF_loadBalancerIP_and_Port
m.body(m.UPDATE,"ascReqData:evSubsc:notifUri",update(m.body("ascReqData:evS
ubsc:notifUri")).replace("iwf.com","10.178.254.158:30079"))  //
IWF_loadBalancerIP_and_Port
end

rule "pt_h2d_ccri_rule_N7_To_Gx"
salience 20
```

```
when
m : IWFHttpRequest(getUri() matches ".*(npcf-smpolicycontrol/v1/sm-
policies)(/$|$)")
then
m.forwardPath=IWFConsts.FORWARD_TO_H2D
m.header(m.ADD,"diameterApplicationId","16777238")
m.header(m.ADD,"diameterCommandCode","272")
m.header(m.ADD,"requestType","CREATE")
end

rule "pt_h2d_ccru_rule_N7_To_Gx"
salience 21
when
m : IWFHttpRequest(getUri() matches ".(npcf-smpolicycontrol/v1/sm-
policies/)(.)(/update)(/$|$)")
then
m.forwardPath=IWFConsts.FORWARD_TO_H2D
m.header(m.ADD,"diameterApplicationId","16777238")
m.header(m.ADD,"diameterCommandCode","272")
m.header(m.ADD,"requestType","UPDATE")
m.header(m.ADD,"original-req-uri",m.getUri())
end

rule "pt_h2d_ccrt_rule_N7_To_Gx"
salience 21
when
m : IWFHttpRequest(getUri() matches ".(npcf-smpolicycontrol/v1/sm-
policies/)(.)(/delete)(/$|$)")
then
m.forwardPath=IWFConsts.FORWARD_TO_H2D
m.header(m.ADD,"diameterApplicationId","16777238")
m.header(m.ADD,"diameterCommandCode","272")
m.header(m.ADD,"requestType","DELETE")
m.header(m.ADD,"original-req-uri",m.getUri())
end
```

NF-Mediation Rule Configuration

Add below rules to the mediation rules:

```
rule "default"
when
    m : NFHttpRequest(m.header("source") == "pcf")
then
    m.header(m.DEL, "custom-header-bad")

end

rule "scp-rule-header-1"
    salience 20
    agenda-group "scp-triggerpoint2"
when
    m : NFHttpRequest(m.header("custom-key") == "123")
then
```

```
    m.header(m.ADD, "custom-value", "pcf-nrf")
        m.header(m.UPDATE, "name", "nf","pcfnf")
end

rule "scp-rule-body-2"
    salience 21
    agenda-group "scp-triggerpoint2"
when
    m : NFHttpRequest( (m.body("nfType") == "PCF") &&
                       (m.body("fqdn") matches "^(cisco).*") )
then
    m.body(m.ADD, "pcfInfo:supiRangeList[1]:end", "2000000000")
    m.body(m.ADD, "pcfInfo:supiRangeList[1]:start", "1000000000")

    m.body(m.UPDATE, "nfServices[1]:ipEndPoints[1]:ipv4Address", 
"10.75.213.100")

    m.body(m.DEL, "nfServices[1]:apiPrefix")
end
```

9 IWF Configurable Parameters

This section includes information about configurable parameters required during IWF installation.

The following tables describes the configuration parameters for each micro service that is configured during IWF deployment using `ociwf-custom.values.yaml` file:

pcf-gateway Microservice

Table 9-1 pcf-gateway Microservice

Parameter	Description	Default Value
global.dockerRegistry	Image repo	cgbudocker.us.oracle.com :5655
global.imageTag	Image tag	latest
pcf.deploymentOcpmPcfDiamGateway.envGatewayMode	Mode of gateway	bsf
pcf.deploymentOcpmPcfDiamGateway.image	Name of the image	diam-gateway
pcf.deploymentOcpmPcfDiamGateway.imageTag	Tag of the Image	1.4.0
pcf.hostIp	Host IP	slave1=10.196.46.13

iwf-mediation

Table 9-2 IWF Mediation Configuration parameters

Parameter	Description	Default Value
image.name	Image name	ocmed-iwf-mediation
image.repository	Image repository name	reg-1:5000
image.tag	Tag of Image	1.4.0
service.active.ForwardToTest	Whether Trial rule test needs to be enabled or not	Disable
service.active.nodePortHttp	Http port to receive traffic	30079
service.active.nodePortHttps	Https port to receive traffic	30080

iwf-mediation-test Micro service

Table 9-3 iwf-mediation-test Micro service

Parameter	Description	Default Value
image.repository	Image repository name	reg-1:5000
Image name	Image name	ocmed-iwfmediation
image.tag	Tag of Image	1.4.0

nf-mediation

Table 9-4   NF Mediation Configuration parameters

Parameter	Description	Default Value
image.name	Image name	ocmed-nf-mediation
image.repository	Image repository name	reg-1:5000
image.tag	Tag of Image	1.4.0
service.active.ForwardToTest	Whether Trial rule test needs to be enabled or not	Disable
service.active.nodePortHttp	Http port to receive traffic	30081
service.active.nodePortHttps	Https port to receive traffic	30082

nf-mediation-test Micro service

Table 9-5   nf-mediation-test Micro service

Parameter	Description	Default Value
image.repository	Image repository name	reg-1:5000
Image name	Image name	ocmed-nfmediation
image.tag	Tag of Image	1.4.0

iwf-d2h

Table 9-6   IWF D2H Configuration parameters

Parameter	Description	Default Value
image.repository	Image repository name	reg-1:5000
image.name	Image name	ociwf-iwfd2h
image.tag	Tag of Image	1.4.0
opentracingHost	Kubernetes master node IP addres	127.0.0.1 (Customer must provide the correct IP address)
opentracingPort	UDP node port of Jaeger-Agent	0 (Customer must provide the correct port)

iwf-h2d

Table 9-7   IWF H2D Configuration parameters

Parameter	Description	Default Value
image.repository	Image repository name	reg-1:5000
image.name	Image name	ociwf-iwfh2d
image.tag	Tag of Image	1.4.0
opentracingHost	Kubernetes master node IP address	127.0.0.1 (Customer must provide the correct IP address)
opentracingPort	UDP node port of Jaeger-Agent	0 (Customer must provide the correct port)

iwf-diameterproxy

Table 9-8 IWF Diameter Proxy Configuration parameters

Parameter	Description	Default Value
image.repository	Image repository name	reg-1:5000
image.name	Image name	ociwf-iwfdiamproxy
image.tag	Tag of Image	1.4.0
DIAMETER_Realm	Diameter Realm of PT diameter node	Customer must provide the realm to be used
DIAMETER_Identity	FQDN of PT diameter node	Customer must provide the FQDN to be used
dpDBService1	MySQL cluster's node-1 IP address or MySQL K8s service name	iwf-pt-mysql-svc (customer must provide correct value)
dpDBService2	MySQL cluster's node-2IP addressor MySQL K8s service name	iwf-pt-mysql-svc (customer must provide correct value)
opentracing Host	Kubernetes master node IP address	127.0.0.1 (Customer must provide the correct IP address)
opentracing Port	UDP node port of Jaeger-Agent	0 (Customer must provide the correct port)
pcfDiscoveryMode	Flag which enables to switch modes(PDRA and D2H)	true
connectorMode	Mode of Diameter Connector	bsf

iwf-mysql

Table 9-9 IWF MYSQL Configuration parameters

Parameter	Description	Default Value	Notes
enabled	Option to provision local K8s MySQL pod	false	Customer needs to fill it. When set to true the local mysql pod is brought up (Note: This is only for testing purpose, not for production. Production environment is expected to use MySQL cluster)
mysqlUser	MySQL User name	iwf_diameter_usr	Customer needs to fill the user name to be used **Note**: This is only applicable when the above mentioned "enabled" option is set to true, else customer need not configure.
mysqlPassword	MySQL User password		Customer needs to fill the user password to be used (Note: This is only applicable when the above mentioned "enabled" option is set to true, else customer need not configure)

Table 9-9 (Cont.) IWF MYSQL Configuration parameters

Parameter	Description	Default Value	Notes
initializationFiles.iwf-db.sql	Mysql ddl commands to be run while deploying the Mysql pod	CREATE DATABASE IF NOT EXISTS diameter DEFAULT CHARACTER SET utf8 DEFAULT COLLATE utf8_general_ci;	Should Not be changed (Note: This is only applicable when the above mentioned "enabled" option is set to true, else customer need not configure)
initializationFiles.permission.sql	Mysql permission to the user	GRANT ALL PRIVILEGES ON *.* TO 'iwf_diameter_usr'@'%';	Customer needs to edit the "user name" in the command command, based on the value set to " mysqlUser" option. (Note: This is only applicable when the above mentioned "enabled" option is set to true, else customer need not configure)

iwf-nrfclient

Table 9-10 IWF NRF Client Configuration parameters

Parameter	Description	Default Value
image.repository	Image repository name	dsr-master0:5000
image.name	Image name	ociwf-iwfnrfclient
image.tag	Tag of Image	1.4.0

Table 9-11 iwf-pcfdiscovery Micro service

Parameter	Description	Default Value
image.repository	Image repository Name	reg-1:5000
image.name	Image Name	ociwf
image.tag	Tag or Image	1.4.0
opentracing Host	Kubernetes master node IP address	127.0.0.1 (Customer must provide the correct IP address)
opentracing Port	UDP node port of Jaeger-Agent	0 (Customer must provide the correct port)
bsfSvc	Service or IP of the BSF	
bsfPort	Port of the BSF	8080

Diameter Peer configuration

Peer nodes are configured in gateway in `configmap-pcf-diam-gateway-service-diameter.yaml` file in location of `chart pcf/templates`

The sample is provided below:

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: pcf-diam-gateway-config-peers
data:
  diameter-config-peers: |
    version: '0.3'
    kind: 'diameter-config'
    metadata:
      label: 'diameter-config-peers'
    setting:
      reconnectDelay: 3
      responseTimeout: 5
      connectionTimeOut: 3
      watchdogInterval: 6
      transport: 'TCP'
    # type: [af, dra]
    nodes:
      - name: 'P-CSCF'
        type: 'pcrf'
        responseOnly: true
        host: '10.75.215.205'
        port: 3880
        realm: 'ociwf.oracle.com'
        identity: 'pcrfsim.ociwf.oracle.com'
```

Parameters		Definitions
reconnectDelay		Time delay in seconds between successive peer connection establishment attempts
responseTimeout		Response timer value in seconds
connectionTimeOut		Connection timer value in seconds
watchdogInterval		Inactivity time in seconds after which DWR will be triggered
transport		Transport protocol type "TCP"
Nodes (list)	name	Name of the peer node
	response Only	Indicates the Diameter GW proxy client or server
	host	IP address of the peer node
	port	Port on which peer node listens for connections
	realm	Realm of the peer node
	identity	FQDN of the peer node

A

Appendix IWF Yaml Files

This section includes information about configurable parameters defined in IWF Yaml Files.

Sample `ociwf-custom-values-1.4.0.yaml` file:

```
# Copyright 2018 (C), Oracle and/or its affiliates. All rights reserved.
# Default values for iwf-pt.
# This is a YAML-formatted file.
# Declare variables to be passed into your templates

namespace: iwfsvc

#--------------------------------diam-gateway-----------------------------

pcf:
  global:
    dockerRegistry: cgbudocker.us.oracle.com:5655
    imageTag: staging-493384
  pcf:
    hostIp: slave1=10.196.46.13
    deploymentOcpmPcfDiamGateway:
      envGatewayMode: bsf
      image: diam-gateway
      imageTag: 1.4.0

#---------------------------------mysql-----------------------------------
iwf-mysql:
  enabled: true
  mysqlUser: iwf_diameter_usr
  mysqlPassword: Dukw1@m?
  initializationFiles:
   iwf-db.sql: |-
      CREATE DATABASE IF NOT EXISTS diameter DEFAULT CHARACTER SET utf8
DEFAULT COLLATE utf8_general_ci;
   permission.sql: |-
      GRANT ALL PRIVILEGES ON *.* TO 'iwf_diameter_usr'@'%';

#-----------------------------------dp----------------------------------

iwf-diameterproxy:
  image:
    repository: reg-1:5000
    name: ociwf-iwfdiamproxy
    tag: 1.4.0
  DIAMETER_Realm: ociwf.oracle.com
  DIAMETER_Identity: iwf.ociwf.oracle.com
  dpDBService1: iwf-pt-mysql-svc
  dpDBService2: iwf-pt-mysql-svc
  opentracingHost: 10.75.157.169
  opentracingPort: 32460
  mysqlUsername: iwf_diameter_usr
  mysqlPassword: Dukw1@m?
  pcfDiscoveryMode: true
  connectorMode: bsf

#----------------------------------d2h-----------------------------------

iwf-d2h:
  image:
    repository: reg-1:5000
    name: ociwf-iwfd2h
    tag: 1.4.0
  opentracingHost: 10.75.157.169
  opentracingPort: 32460


#----------------------------------h2d-----------------------------------

iwf-h2d:

  image:
    repository: reg-1:5000
    name: ociwf-iwfh2d
    tag: 1.4.0

  opentracingHost: 10.75.157.169
  opentracingPort: 32460
#----------------------------------
mediation-----------------------------------

iwf-mediation:

  image:
    repository: reg-1:5000
    name: ocmed-iwfmediation
    tag: 1.4.0

  service:
    active:
      nodePortHttp: 30079
      nodePortHttps: 30080
      forwardToTest: Disable
  opentracingHost: 10.75.157.169
  opentracingPort: 32460

#------------------------------mediation-----------------------------------
nf-mediation:
  enabled: true
  image:
    repository: reg-1:5000
    name: ocmed-nfmediation
    tag: 1.4.0
  service:
    active:
      nodePortHttp: 30081
      nodePortHttps: 30082
      forwardToTest: Disable
  opentracingHost: 10.75.157.169
  opentracingPort: 32460

#---------------------nrfclient-----------------------------------------
iwf-nrfclient:
  image:
    repository: reg-1:5000
    name: ociwf-iwfnrfclient
    tag: 1.4.0

#---------------------pcfDiscovery-----------------------------------

iwf-pcfdiscovery:
  image:
    repository: reg-1:5000
    name: ociwf-iwfpcfdiscovery
    tag: 1.4.0

  opentracingHost: 10.75.157.169
  opentracingPort: 32460
  bsfSvc: bsf-stub-service.default
  bsfPort: 8080
```