



HPE NFV Director

VIM Integration Guide

Release 4.1

First Edition

Notices

Legal notice

© Copyright 2016 Hewlett Packard Enterprise Development LP

Confidential computer software. Valid license from HPE required for possession, use or copying. Consistent with FAR 12.211 and 12.212, Commercial Computer Software, Computer Software Documentation, and Technical Data for Commercial Items are licensed to the U.S. Government under vendor's standard commercial license.

The information contained herein is subject to change without notice. The only warranties for HPE products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. HPE shall not be liable for technical or editorial errors or omissions contained herein.

Printed in the US

Trademarks

Linux is the registered trademark of Linus Torvalds in the U.S. and other countries.

Oracle and Java are registered trademarks of Oracle and/or its affiliates.

Adobe®, Acrobat® and PostScript® are trademarks of Adobe Systems Incorporated.

Microsoft®, Internet Explorer, Windows®, Windows Server 2007®, Windows XP®, and Windows 7® are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

Firefox® is a registered trademark of the Mozilla Foundation.

Google Chrome® is a trademark of Google Inc.

EnterpriseDB® is a registered trademark of EnterpriseDB.

Postgres Plus® Advanced Server is a registered U.S. trademark of EnterpriseDB.

UNIX® is a registered trademark of The Open Group.

X/Open® is a registered trademark, and the X device is a trademark of X/Open Company Ltd. in the UK and other countries.

Red Hat® is a registered trademark of the Red Hat Company.

Apache CouchDB, CouchDB, and the project logo are trademarks of The Apache Software Foundation.

Node.js project. Joyent® and Joyent's logo are registered trademarks of Joyent, Inc.

Neo4j is a trademark of Neo Technology.

VMware ESX, VMWare ESXi, VMWare vCenter and VMWare vSphere are either registered trademarks or trademarks of VMware, Inc. in the United States and other jurisdictions.

Contents

Notices	1
Preface	6
About this guide.....	6
Audience	6
Document history	6
Chapter 1 Introduction.....	7
1.1 Overview of resource discovery	7
1.1.1 Architectural View	7
1.1.2 NFV Discovery Process	8
1.2 NFV Director resources.....	8
1.2.1 Discovered resources	8
1.2.2 Resources with default value	9
Chapter 2 Installation and Configuration	10
2.1 Installation	10
2.1.1 Prerequisites.....	10
2.1.2 Installation steps	10
2.2 Configuration	10
Chapter 3 Triggering Discovery process	12
3.1 Upload VIM and AUTHENTICATION instances.....	12
3.2 Triggering discovery.....	13
3.3 Resources with default value.....	13
Chapter 4 Discovery utilities	15
4.1 Enabling and Disabling of discovery process	15
4.1.1 Disable discovery even in fresh installation	15
4.1.2 Disable discovery temporarily	15
4.1.3 Enable Discovery	15
4.1.4 Manual Discovery trigger	15
4.1.5 Making changes in CA properties.....	15
4.1.6 Track Initial/Incremental Discovery completion	15
Chapter 5 Multi VIM duplicate compute hostname scenario.....	17
5.1 Run amend duplicate compute hostnames tool	17
5.1.1 Track amend topology script completion.....	17
5.1.2 Enable Discovery	17
5.1.3 Rerun Discovery.....	17
Chapter 6 VIM Certificates	18
6.1 Importing VIM certificate to SiteScope	18
6.2 Importing Ceilometer certificate to SiteScope	18
Chapter 7 DCN Integration.....	19
7.1 Prerequisites	19
7.2 Integrate DCN with NFV Director	19
7.2.1 Create the SDN Topology manually.....	19

7.2.2 Upload DCN resource	21
7.2.3 Connect Datacenter with DCN resources.....	21
7.2.4 Replacement of Networking Artifacts	24
7.2.4.1 Replace NETWORKING:OPENSTACK Artifacts with NETWORKING:OPENSTACK:DCN ...	24
7.2.5 Create relationship between NETWORKING and DCN Artifacts	26

List of tables

Table 1: Document history..... 6

List of figures

Figure 1: OpenStack Discovery Architecture	7
Figure 2: OpenStack Discovery NFV Director components.....	8
Figure 3: Discovered resources.....	9
Figure 4: Resources populated with default value	9
Figure 5: DCN topology pictorial representation.....	20
Figure 6: Uploading DCN topology into fulfillment.....	21
Figure 7: Query ID of Datacenter	22
Figure 8: Response for Datacenter Query	22
Figure 9: Query ID of SHARED_NETRESOURCE:DCN.....	23
Figure 10: Response for SDN_CONTROLLER:DCN Query.....	23
Figure 11: Create Relationship.....	24
Figure 12: Query NETWORKING:OPENSTACK associated with Region.....	25
Figure 13: Query Response for NETWORKING:OPENSTACK associated with Region.....	25
Figure 14: REST operation to update NETWORKING:OPENSTACK.....	26
Figure 15: REST operation to create relationship between NETWORKING and DCN.....	27

Preface

About this guide

This document explains the procedure to install and configure the OpenStack Kilo Discovery procedure. This document also gives an overview of resource discovery, architectural view, enabling/disabling of discovery process, and list of objects discovered.

By following the procedures in this document, Helion CG and OpenStack Kilo resources can be discovered.

Audience

This document is any stakeholder requiring to perform resource discovery using the NFV Director. Pre requisite is to have knowledge of NFV Director Concepts, and an understanding of the NFV Director resource model.

Document history

Table 1: Document history

Edition	Date	Description
1.0	October 14, 2016	First edition.

Chapter 1

Introduction

The aim of this document is to provide

- Purpose and Overview of resource discovery using VIM.
- Architectural view.
- Installation and configuration of Discovery Process.
- Enabling and Disabling of Discovery Process.
- The list of resources discovered.
- Integrating DCN with NFV Director discovered Data Center

1.1 Overview of resource discovery

NFV Director is responsible for managing the lifecycle of VNF and it's important for NFV Director to know the complete topology of the OpenStack resources.

The complete list of OpenStack resource topology is described below.

The Discovery process described in this document helps in automatic discovery of OpenStack resources and their inter-relationship.

It is an optional component in the NFV Director.

1.1.1 Architectural View

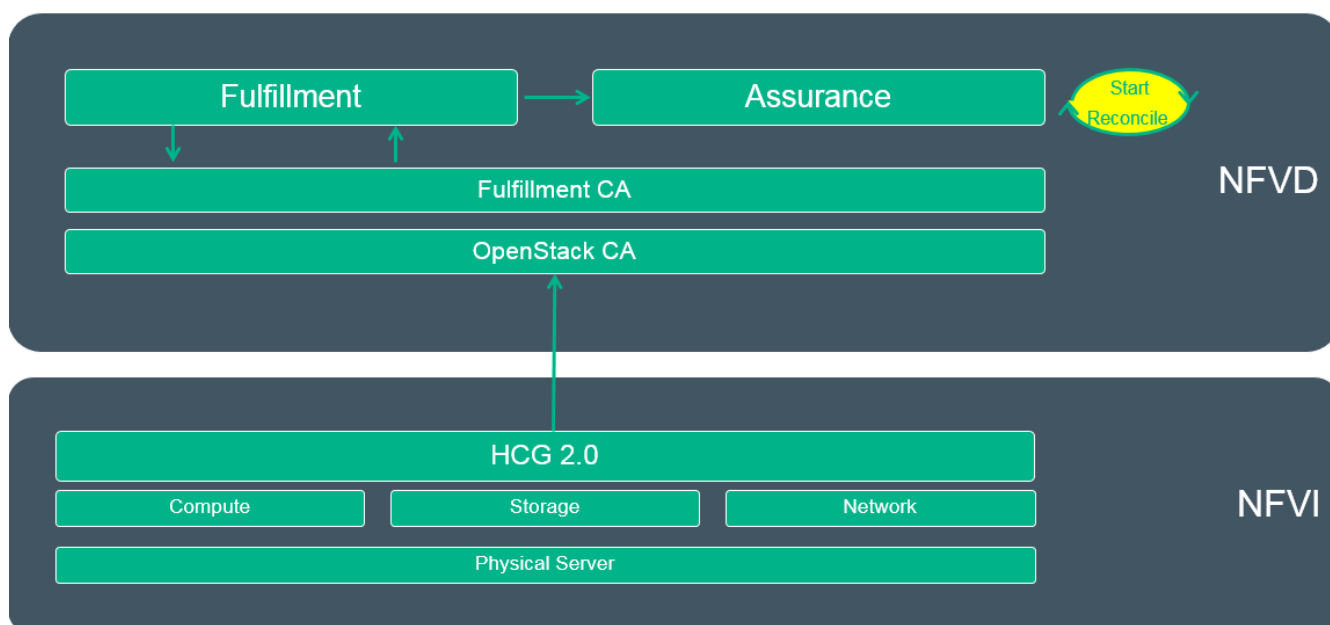


Figure 1: OpenStack Discovery Architecture

1.1.2 NFV Discovery Process

NFV Director Discovery is an optional component that provides the resource tree information for NFV-Director. It consists of VIM specific channel adapter such as openstack-ca and a reconciliation channel adapter that can be deployed on Open Mediation.

As soon as the discovery components are installed and configured, discovery process is triggered. Manual discovery can be triggered any time. At any time, only one instance of discovery runs.

Discovery process consists of two modules:

Discovery Module: Interacts with VIM such as OpenStack and queries for resource information and parse the JSON response into artifact-relationship model.

Reconciliation Module: Channel adapter will fetch OpenStack VIM & Authentication details from fulfillment via REST API and trigger discovery module by connecting to the VIM using provided credentials.

Reconciliation module will query fulfillment to get the existing discovered information at the start. Discovery module will send discovered data in artifact-relationship instance model. Once the discovered data is received, then Reconciliation module formulates logic and builds delta information to reconcile. The final data will be prepared and persisted to fulfillment via REST API's.

Below is the pictorial diagram that explains the design approach of NFV Director Discovery.

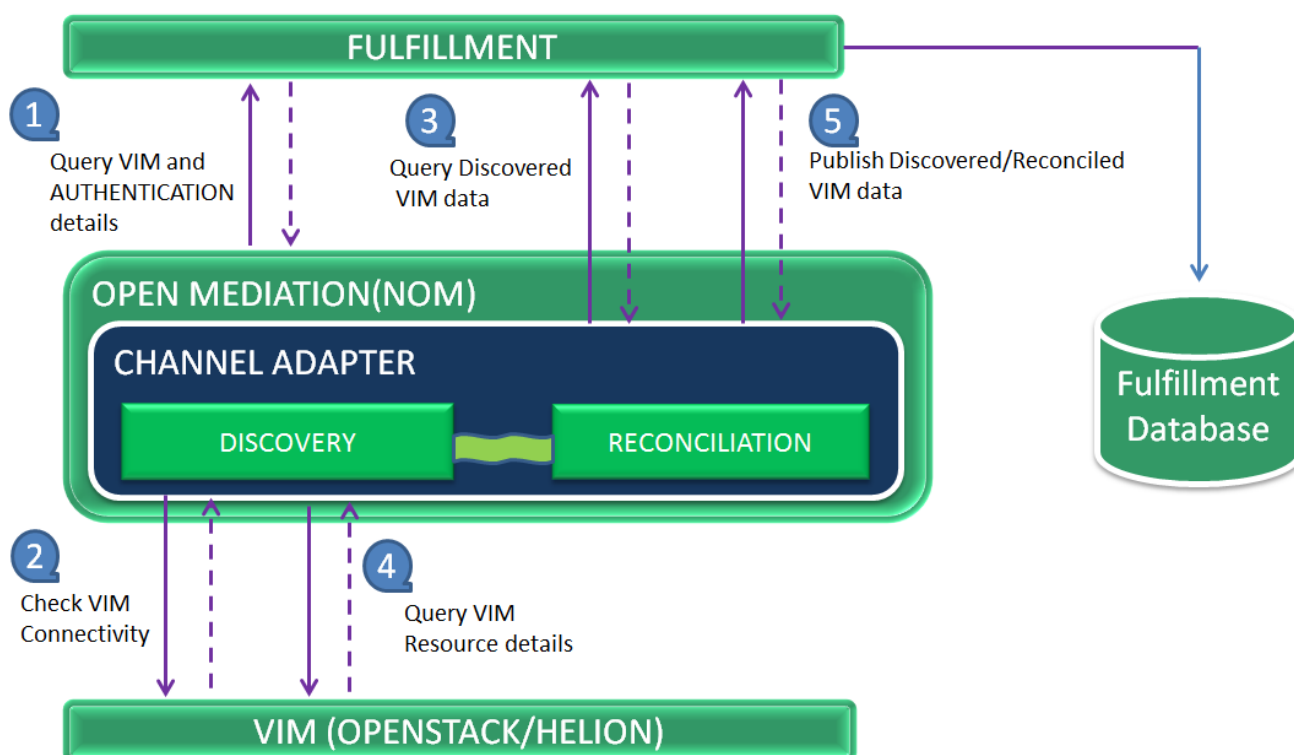


Figure 2: OpenStack Discovery NFV Director components

1.2 NFV Director resources

1.2.1 Discovered resources

Following resources are auto discovered from OpenStack or HCG

- | | |
|--|--|
| <ul style="list-style-type: none"> ✓ Servers <ul style="list-style-type: none"> • CPU, Memory, Disk • Port, Interface • Total, used and available capacity ✓ Regions ✓ Tenant ✓ OpenStack Services ✓ Hypervisors <ul style="list-style-type: none"> • ESX, KVM, Bare metal ✓ Availability Zones ✓ Host Aggregates | <ul style="list-style-type: none"> ✓ Networks ✓ Subnetworks <ul style="list-style-type: none"> • IPAddress ✓ Virtual Machines <ul style="list-style-type: none"> • vCPU, vMemory, vDisk • vPort ✓ LUN/vLUN ✓ Images ✓ Flavor ✓ Carrier Grade <ul style="list-style-type: none"> • NUMA, Huge pages • PCI-PT • SR-IOV |
|--|--|

Figure 3: Discovered resources

For the discovery of resources from VIM, VIM and its AUTHENTICATION details must be already present in the NFV Director system.

Thus, VIM and AUTHENTICATION are manually provided as input.

1.2.2 Resources with default value



NOTE: Default values are populated into NFV Director for the following artifacts as they cannot be discovered today from underlying VIM. Once Discovery is completed, these values have to configure in NFVD GUI by the user. Refer to 3.3 Resources with default value for more details.

- ✓ Policy
- ✓ LUN amount
- ✓ Server Class
- ✓ Server/CPU usage mode
- ✓ Data center
 - Rack
 - Enclosure

Figure 4: Resources populated with default value

Default values for these items populated by Discovery module.

Over Subscription Policy – 1

LUN Amount – 0

Server Class – Class_A

Server/CPU usage mode – shared

Datacenter –static data

Rack – static data

Enclosure – static data

Card -- static data.

Chapter 2 Installation and Configuration

Note: Sections in this chapter are only for reference. NFV Director install procedure would have already installed and configured the Openstack Discovery components in the <AA_HOST>.

1. To verify if the RPM package is already installed, run the following command on <AA_HOST>:

```
rpm -qa | grep nfvd-discovery-common
```

Output will be in the following format:

```
nfvd-discovery-common-<product version>.el6.noarch
```

2. To verify if required Channel Adapters are already deployed on <AA_HOST>:

```
/opt/openmediation-70/bin/nom_admin --list-ip-in-container | grep -e fulfillment -e openstack
```

It must return the following output:

```
DEPLOYED    fulfillment-ca-10
```

```
DEPLOYED    openstack-ca-10
```

2.1 Installation

2.1.1 Prerequisites

1. Rest Client
2. RPM package nfvd-discovery-common-<product version>.el6.noarch.rpm
 - a. openstack-ca-1.0.0.zip
 - b. fulfillment-ca-1.0.0.zip
3. DCN (Nuage) v3.2.1.1, if external DCN is used.

2.1.2 Installation steps

1. Install the discovery rpm in the <AA_HOST>, where we have the Open Mediation installed.

```
rpm -ivh nfvd-discovery-common-04.01.000-1.el6.noarch.rpm
```

2. Install openstack-ca and fulfillment-ca

```
unzip -d /opt/openmediation-70/ips/ /opt/HP/nfvd/discovery/common/openstack-ca-1.0.0.zip
nom_admin --install-ip openstack-ca-10
nom_admin --install-ip-in-container 0 openstack-ca-10
```

```
unzip -d /opt/openmediation-70/ips/ /opt/HP/nfvd/discovery/common/fulfillment-ca-1.0.0.zip
nom_admin --install-ip fulfillment-ca-10
nom_admin --install-ip-in-container 0 fulfillment-ca-10
```

2.2 Configuration

1. Edit the below file for configurations of rest end point

```
/var/opt/openmediation-70/containers/instance-0/ips/fulfillment-ca-10/etc/config/
reconciliation-endpoints.properties
```

```
#Fulfillment rest endpoint ipaddress/hostname
rest.endpoint=<FF_HOST_IP>

#Fulfillment rest endpoint port
rest.port=<FF_PORT>
```



NOTE: If the rest.endpoint and rest.port is reflecting the correct <FF_HOST_IP> and <FF_PORT>, skip this step.

2. Deploy openstack-ca and fulfillment -ca, so that properties changes are taken into effect.

```
/opt/open-mediation-70/bin/nom_admin --deploy-ip-in-container 0 openstack-ca-10
/opt/open-mediation-70/bin/nom_admin --deploy-ip-in-container 0 fulfillment-ca-10
```



NOTE: If any attribute is changed in the property file, the channel adapters must be un-deployed and redeployed, as shown in the commands below:



IMPORTANT: Deploying the fulfillment-ca will trigger discovery.

```
/opt/open-mediation-70/bin/nom_admin --undeploy-ip-in-container 0 openstack-ca-10
/opt/open-mediation-70/bin/nom_admin --undeploy-ip-in-container 0 fulfillment-ca-10
```

Chapter 3 Triggering Discovery process

Triggering the discovery involves two steps:

- Uploading VIM and AUTHENTICATION artifacts and relationship instances to NFV Director
- Triggering discovery

3.1 Upload VIM and AUTHENTICATION instances

VIM and AUTHENTICATION details of the OpenStack must be populated into NFV Director. By doing this, NFV Director becomes aware of the OpenStack URL, credentials and tenant details.

On <AA_HOST>, run the following script to populate the VIM and AUTHENTICATION details

```
/opt/HPE/nfvd/discovery/scripts/nfvd_createVIM.sh
```

Usage: nfvd_createVIM.sh

```
Usage: ./nfvd_createVIM.sh [-host <host>] [-port <port>] [-vimname <vimname>] [-url <url>] [-username <username>] [-password <password>] [-tenantname <tenantname>] [-discovertenant <discovertenant>] [-nominstance <nominstance>] [-nomhost <nomhost>] [-vimcategory <vimcategory>] [-vimtype <vimtype>][[-authversion <authversion>]]
```

Example:

```
./nfvd_createVIM.sh -host 10.206.254.14 -port 8080 -vimname FC33 -url https://10.207.114.100:5000/v2.0/tokens -username admin -password YMCtbGCT9 -tenantname admin -vimcategory HELION -vimtype HCG
```

Where:

MANDATORY:

-host	Hostname or IPAddress of Fulfillment
-port	Fulfillment Port (eg: 8080)
-vimname	VIM Name eg. vim-helion
-url	VIM Authentication URL(Keystone V2/V3 URL) Note: In case of V2 keystone, you need to append /tokens at the end of keytone URL Ex: <a href="https://<IP>:5000/v2.0/tokens">https://<IP>:5000/v2.0/tokens In case V3, you need to append "/auth/tokens" Ex: <a href="https://<IP>:5000/v3/auth/tokens">https://<IP>:5000/v3/auth/tokens
-username	VIM user with administrator privileges on the tenants to be discovered
-password	VIM password for the user
-tenantname	Tenant on which the user has administrative privileges

OPTIONAL:

-discovertenant	Comma separated list of specific tenants that needs to be discovered (default:blank), if not provided all tenants will be discovered
-nominstance	NOM instance number (default:blank)
-nomhost	NOM hostname (default:blank)
-vimcategory	VIM category , values are either OPENSTACK/HELION (default:OPENSTACK)
-vimtype	General type of the VIM (default:OPENSTACK for vimcategory OPENSTACK, HCG for HELION)
-authversion	VIM Authentication keystone version (default:V2)



NOTE: If there are multiple OpenStack instances to be discovered, their respective VIM and AUTHENTICATION instances must be uploaded.

3.2 Triggering discovery

Run the following script on <AA_HOST> to trigger discovery of the VIM instances uploaded in the previous step:

```
cd /opt/HPE/nfvd/discovery/scripts/
./trigger_reconciliation.sh
```



NOTE: Discovery logs will be available in Open Mediation Service Mix log.
Default location: /var/opt/openmediation-70/containers/instance-0/data/log



NOTE: For various discovery utilities, refer to the next chapter.

3.3 Resources with default value

This section is only applicable, when NFVD wants to deploy VNF on Storage.

Once discovery operation is complete, some resources are stored in NFVD with default value, e.g, LUN Amount. Refer to section 1.2.2 for resources that are stored with default value.

In order to update the default values to reflect actual values, follow the below steps:

- Query the DATACENTER artifact ID from FF server using REST client.

```
> http://<FF_HOST_IP>:8080/nfvd/instance/artifact?family=DATACENTER
```

GET POST PUT DELETE Other methods

Raw headers

```
X-Auth-Token: 3778fe88-e71d-4004-86bc-3188f7fd450b
Content-Type: application/xml
```

- Pick the DATACENTER artifact ID from the response body, for which you want to modify the resources.
- Perform a start of data load by executing the REST request “/nfvd/discovery/<datacenter_artifact_id>/start”.

```
> http://<FF_HOST_IP>:8080/nfvd/discovery/<datacenter_artifact_id>/start
```

GET POST PUT DELETE Other methods application/xml

Raw headers

Headers form

```
X-Auth-Token: 3778fe88-e71d-4004-86bc-3188f7fd450b
Content-Type: application/xml
```

- Edit the resources using GUI for which default values were populated by Discovery module. Browse to the respective resource from Instance Menu in GUI and select Edit option from Actions. Sample shown for LUN instance below:

Raw Attributes

Attribute Name	Value	Unit	Description
ID	LUN ID	TEXT	
Name	Kvm-Baremetal-Quality-A	TEXT	
Type	Kvm-Baremetal-Quality-A	TEXT	LUN type on base of Hypervisor (KVM, ESX, IRONIC)
Amount	200000	GB	

Parents

Name	Type	Category
Component name	STORAGE	OPENSTACK

Edit attributes: Kvm-Baremetal-Quality-A

INFO STATUS INTEGRATION

ID:
LUN ID

Name:
Kvm-Baremetal-Quality-A

Type:
Kvm-Baremetal-Quality-A

Amount:
200000

Update Cancel

- Stop the data load by executing the REST request `"/nfvd/discovery/<datacenter_artifact_id>/stop"`.

> `http://<FF_HOST_IP>:8080/nfvd/discovery/<datacenter_artifact_id>/stop`

GET
 POST
 PUT
 DELETE
 Other methods
 application/xml

Raw headers Headers form

```
X-Auth-Token: 3778fe88-e71d-4004-86bc-3188f7fd450b
Content-Type: application/xml
```

- Quota calculation will take some time to complete based on number of DATACENTER resources.
- Login with an Organization or VDC level user in the GUI, and the changes should be available in Quota management windows.
- User can now modify the quotas accordingly.

Chapter 4 Discovery utilities

4.1 Enabling and Disabling of discovery process

By default discovery is enabled, when NFV Director Discovery components are installed. The following utilities can be run on the <AA_HOST>

4.1.1 Disable discovery even in fresh installation

Execute the below script when you install the fulfillment-ca, before deploying it.

```
cd /opt/HPE/nfvd/discovery/scripts/  
sh disable_discovery.sh
```

4.1.2 Disable discovery temporarily

Execute the below script. Once disabled subsequent Discovery runs will not be triggered. Disabling while discovery in progress will not impact the current run.

```
cd /opt/HPE/nfvd/discovery/scripts/  
sh disable_discovery.sh
```

4.1.3 Enable Discovery

Execute the below script.

```
cd /opt/HPE/nfvd/discovery/scripts/  
sh enable_discovery.sh
```

4.1.4 Manual Discovery trigger

Manual discovery can be triggered any time. It will not get triggered when another instance of Discovery is already running. Run the following script to trigger manual discovery:

```
cd /opt/HPE/nfvd/discovery/scripts/  
./trigger_reconciliation.sh
```

4.1.5 Making changes in CA properties

Following is the procedure, if you want to make changes in channel adapter properties:

1. disable discovery
2. un-deploy the CA
3. make changes to properties
4. deploy CA
5. enable discovery

**NOTE:**

See 2.3.2 for instructions to disable discovery
See 2.2 for instructions to undeploy and deploy the CA
See 2.3.3 for instructions to enable discovery

4.1.6 Track Initial/Incremental Discovery completion

Open Mediation log file will have a status message of Discovery:

/var/opt/openmediation-70/containers/instance/data/log/servicemix-info.log

***** [FF-CA] Initial/Incremental Discovery Service has been completed successfully, Quota Calculation is in Progress

Chapter 5 Multi VIM duplicate compute hostname scenario

This section addresses the scenario to resolve the discovered topology, where in two VIM's have same compute node names across them.

5.1 Run amend duplicate compute hostnames tool

The script has to be used only in case the setup matches the below scenario:

- 1) VIM-1 has been discovered using previous kits and few VNF deployments have been made.
- 2) New VIM-2 has been added and discovered and one of its compute nodes has the same name as of VIM-1 compute node.

Once the kits have been installed disable discovery by referring to Note below.

Run the following script on <AA_HOST> to amend the topology:

```
cd /opt/HPE/nfvd/discovery/scripts/
./amend_duplicate_compute_topology.sh
```



NOTE: Discovery has to be disabled manually using the script as soon as the discovery kits have been installed. Refer to section Disable discovery temporarily

5.1.1 Track amend topology script completion

Open Mediation log file will have below messages:

```
/var/opt/openmediation-70/containers/instance/data/log/servicemix-info.log

[AMEND-DISCOVERY] || Processing amend discovered topology trigger ||

[AMEND-DISCOVERY] || Successfully completed amend topology for duplicate compute nodes ||
```

5.1.2 Enable Discovery

Refer to section 4.1.3 Enable Discovery to enable discovery again.

5.1.3 Rerun Discovery

Refer to section 4.1.4 Manual Discovery trigger for triggering discovery again.

Once discovery has been completed the Topology will have the proper data.

Chapter 6 VIM Certificates

If the VIM (vCenter, RHOS, pure OpenStack, HCG) services are https enabled, it is mandatory to import the VIM certificate into SiteScope.

6.1 Importing VIM certificate to SiteScope

In order to import VIM certificate into SiteScope, following is the process:

1. Go to SiteScope Preferences > Certificate Management
2. Click on “Import Certificates” option.
3. Provide the Host IP where Keystone service is installed and the Port. e.g. for OpenStack, the Keystone port is 5000.

In case of OpenStack, check keystone URL in “Access and Security” → “API Access” via horizon dashboard

4. Click on the ‘Load’ button to load the certificate.
5. Now select the loaded certificate and click on ‘Import’.

Note: Make sure that VIM certificates valid, i.e. they generated for correct VIM IP addresses

6.2 Importing Ceilometer certificate to SiteScope

1. Go to SiteScope Preferences > Certificate Management
2. Click on “Import Certificates” option.
3. Provide the Host IP where Ceilometer component is installed, and the Port. E.g. for OpenStack, the Ceilometer port is 8777.

In case of OpenStack, check ceilometer URL in “Access and Security” → “API Access” via horizon dashboard

4. Click on the ‘Load’ button to load the certificate.
5. Now select the loaded certificate and click on ‘Import’.

Note: Make sure that VIM certificates valid, i.e. they generated for correct VIM IP addresses

Chapter 7 DCN Integration

DCN Integration with NFV Director is an optional step. This would be required in case an external DCN has to be used for Networking. In current release DCN (Nuage) is supported.

DCN Topology has to be attached manually once Discovery has been completed.

7.1 Prerequisites

Below section explains the procedure to be followed to integrate DCN with NFV Director.

1. DCN (Nuage) v3.2.1.1, if external DCN is used.
2. DCN_Topology.xml → SDN Topology manually created.

7.2 Integrate DCN with NFV Director

7.2.1 Create the SDN Topology manually



Attachment file 'DCN_Topology.xml' contains the default SDN topology

In the DCN_Topology.xml, edit the following attributes:

- AUTHENTICATION > CREDENTIALS > Url

Value	Example
https://<nuage_ip>:<port>/nuage/api/v3_2	https://172.19.244.225:8443/nuage/api/v3_2

- AUTHENTICATION > CREDENTIALS > Login
- AUTHENTICATION > CREDENTIALS > Password
- AUTHENTICATION > CREDENTIALS > Admin_enterprise
- L3DOMAIN > DOMAIN > RouteDistinguisher

Value	Example
RD Value	65534:12538

- L3DOMAIN > DOMAIN > RouteTarget

Value	Example
RD Value	65534:56825

- L3DOMAIN > DOMAIN > BackHaulRouteDistinguisher

Value	Example
RD Value	65534:62251

- L3DOMAIN > DOMAIN > BackHaulRouteTarget
- L3DOMAIN > DOMAIN > ExportRouteTarget
- L3DOMAIN > DOMAIN > ImportRouteTarget

Value	Example
RT Value	65534:32060

- L3DOMAIN > DOMAIN > BackHaulVNID

Value	Example
VPN ID	314849

- L3DOMAIN > DOMAIN > BackHaulVNID

Value	Example
VPN ID	314849

- L3DOMAIN > DOMAIN > TunnelType

Value	Example
TunnelType	VXLAN

- MACRONET > MACRONET > address
- MACRONET > MACRONET > netmask
- SHARED_NETRESOURCE > RESOURCE > Address
- SHARED_NETRESOURCE > RESOURCE > Netmask
- SHARED_NETRESOURCE > RESOURCE > DomainRouteDistinguisher
- SHARED_NETRESOURCE > RESOURCE > DomainRouteTarget
- SHARED_NETRESOURCE > RESOURCE > Gateway

Below figure depicts the pictorial representation of DCN topology

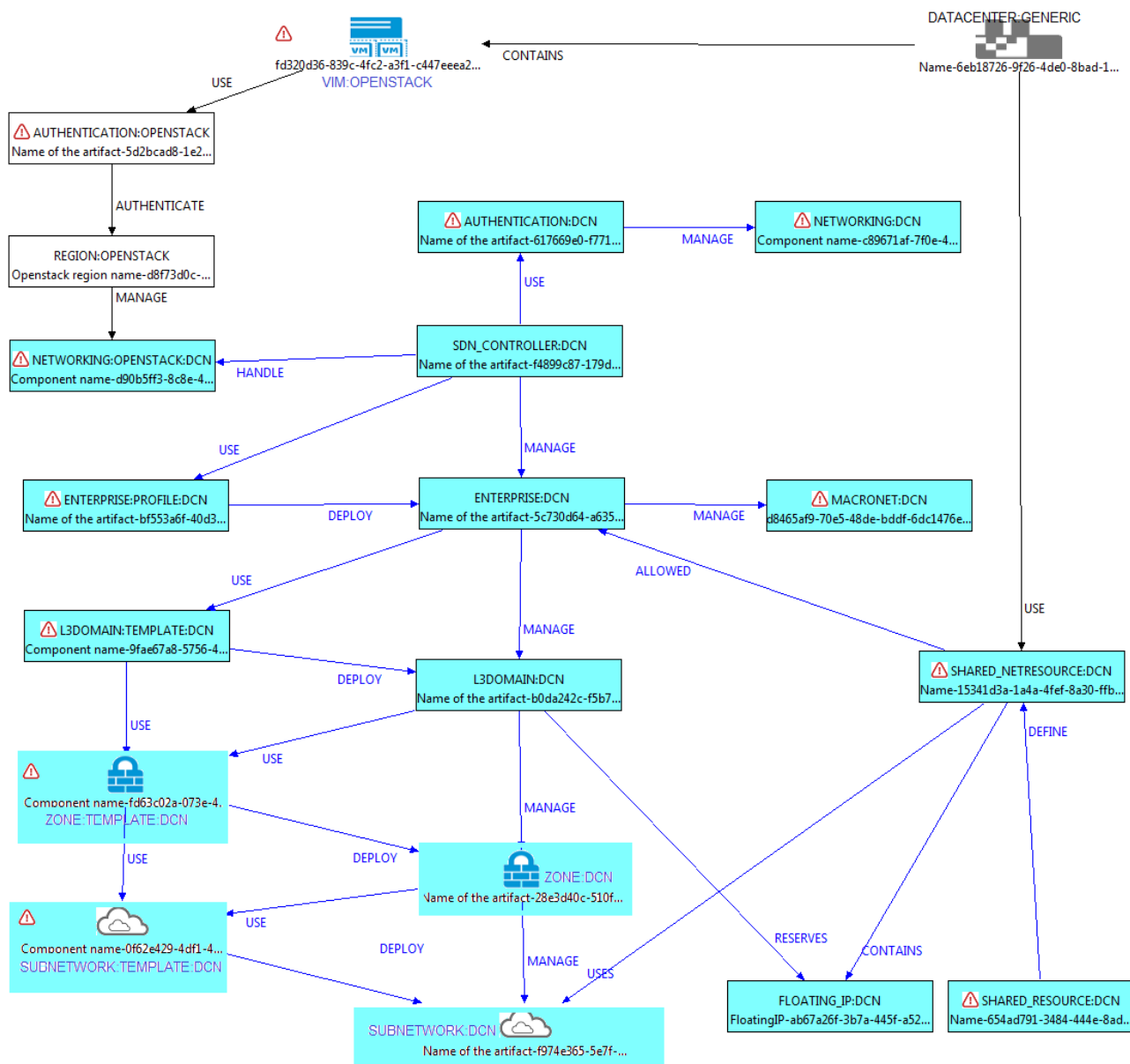


Figure 5: DCN topology pictorial representation

7.2.2 Upload DCN resource

- 1 Open REST Client.
- 2 Provide FF_HOST_IP and FF_PORT details in the REST URL. Select POST HTTP Operation.
- 3 Copy the content of file DCN_Topology.xml inside payload section.



IMPORTANT: For all Rest operations add the below headers:

Content-Type: application/xml

X-Auth-Token: 3778fe88-e71d-4004-86bc-3188f7fd450b.

The screenshot shows a REST client interface with the following configuration:

- Request:**
 - Host: `http://<FF_HOST_IP>:<FF_PORT>`
 - Path: `/nfvd/instance/upload`
 - Query parameters: ADD
 - Hash: (empty)
- Method:** POST (selected)
- Headers:**
 - X-Auth-Token: 3778fe88-e71d-4004-86bc-3188f7fd450b
 - Content-Type: application/xml
- Raw payload:**

```
<?xml version="1.0" encoding="utf-8"?>
<instances xmlns="http://www.hp.com/nfvd">
  <instance-trees>
    <instance>
      <id>07559026-75ae-4c2e-99a1-ab44201e5eb6</id>
      <name>DCN</name>
      <type>sdn_controller</type>
      <description>DCN</description>
      <artifact-instances>
        <artifact-instance>
          <id>f5612797-8983-460a-a494-0e974fa463f6</id>
        </artifact-instance>
      </artifact-instances>
      .....
      <relationship-instances>
        <relationship-instance>
          .....
        </relationship-instance>
      </relationship-instances>
    </instance>
  </instance-trees>
  <elements>
    .....
  </elements>
</instances>
```
- Buttons:** SEND

Figure 6: Uploading DCN topology into fulfillment

7.2.3 Connect Datacenter with DCN resources



CAUTION: Execute it per DC.

- 1 Query Datacenter ID
 - Provide FF_HOST_IP and FF_PORT details in the REST URL. Select GET HTTP Operation.

- Enter the Path and Query parameters and Headers as shown in sample below.
- GENERAL.Name attribute filter is used to filter by Datacenter Name DC1 or DC2 (Name of your Datacenter)

The screenshot shows a REST client interface with the following configuration:

- Request:**
 - Host: `http://<FF_HOST_IP>:<FF_PORT>`
 - Path: `/nfvd/instance/artifact/query/parameters`
 - Query parameters:

definition	DATACENTER:GENERIC	×
attributeFilter	GENERAL.Name=DC1	×
exactMatching	false	×
- Hash:**
 - Method: GET POST PUT DELETE Other methods
 - Raw headers / Headers form
 - HTTP headers:

X-Auth-Token	3778fe88-e71d-4004-86bc-3188f7fd450b	×
Content-Type	application/xml	✎ ×

Buttons: ADD, SEND

Figure 7: Query ID of Datacenter

Status: 200: OK ? Loading time: 200 ms

Response headers (4) Request headers (2) Redirects (0) Timings

Server: Apache-Coyote/1.1
Content-Type: application/xml
Transfer-Encoding: chunked
Date: Thu, 28 Apr 2016 13:51:08 GMT

Raw XML

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<artifact-instances xmlns="http://www.hp.com/nfvd">
  <artifact-instance internal-id="8f2e6b76-a367-4fa2-8444-637aab6ff73f" uri="/nfvd/instance/artifact/8f2e6b76-a367-4fa2-8444-637aab6ff73f">
    <artifact-definition>
      <category>GENERIC</category>
      <family>DATACENTER</family>
    </artifact-definition>
    <status><enabled>true</enabled>
    <label>ENABLED</label>
    <visible-label>ENABLED</visible-label>
  </status>
  <categories>
    <category>
```

Figure 8: Response for Datacenter Query

2 Query SHARED_NETRESOURCE:DCN ID

- Provide FF_HOST_IP and FF_PORT details in the REST URL. Select GET HTTP Operation.
- Enter the Path and Query parameters and Headers as shown in sample below.
- INFO.DC.Name attribute filter is used to filter by Datacenter Name

Request

Host
http://<FF_HOST_IP>:<FF_PORT>

Path
/nfvd/instance/artifact/query/parameters

Query parameters

definition	SHARED_NETRESOURCE:DCN	×
attributeFilter	INFO.DCName=DC1	×
exactMatching	false	×

ADD

Hash

GET POST PUT DELETE Other methods

Raw headers Headers form

HTTP headers

X-Auth-Token	3778fe88-e71d-4004-86bc-3188f7fd450b	×
Content-Type	application/xml	✎ ×

ADD

SEND

Figure 9: Query ID of SHARED_NETRESOURCE:DCN

Status: 200: OK ? Loading time: 200 ms

Response headers (4) Request headers (2) Redirects (0) Timings

Server: Apache-Coyote/1.1
Content-Type: application/xml
Transfer-Encoding: chunked
Date: Thu, 28 Apr 2016 13:51:08 GMT

Raw XML

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<artifact-instances xmlns="http://www.hp.com/nfvd">
  <artifact-instance internal-id="a2d4aee3-25a4-4e3c-a353-a9679046f7a9" uri="/nfvd/instance/artifact/a2d4aee3-25a4-4e3c-a353-a9679046f7a9">
    <artifact-definition>
      <category>DCN</category>
      <family>SHARED_NETRESOURCE</family>
    </artifact-definition>
  </artifact-instance>
  <artifact-instance internal-id="bea8c469-299b-446d-bd72-8cf2c5c0af60" uri="/nfvd/instance/artifact/bea8c469-299b-446d-bd72-8cf2c5c0af60">
    <artifact-definition>
      <category>DCN</category>
      <family>SHARED_NETRESOURCE</family>
    </artifact-definition>
  </artifact-instance>
</artifact-instances>
```

Figure 10: Response for SDN_CONTROLLER:DCN Query

- 3 Create Relationship between DATACENTER and each SHARED_NETRESOURCE retrieved from response above. Relationship sample is shown below:
 - a. parent-artifact-id: DC ID returned from "Query Datacenter ID" step
 - b. child-artifact-id: Shared_NetResource ID returned from "Query SHARED_NETRESOURCE:DCN ID" step

For the above example,

```
<relationship-instances xmlns="http://www.hp.com/nfvd">
  <relationship-instance>
    <categories/>
    <parent-artifact-id>8f2e6b76-a367-4fa2-8444-637aab6ff73f</parent-artifact-id>
```



```

<child-artifact-id>a2d4aee3-25a4-4e3c-a353-a9679046f7a9</child-artifact-id>
<status>
  <enabled>>true</enabled>
  <label>ENABLED</label>
  <visible-label>ENABLED</visible-label>
</status>
<relationship-type>USE</relationship-type>
</relationship-instance>
</relationship-instances>

```

Use the above block as payload section in the Rest client, as shown below.

The screenshot shows a REST client interface with the following configuration:

- Host:** http://<FF_HOST_IP>:<FF_PORT>
- Path:** /nfvd/instance/relationship
- Method:** POST
- Content-Type:** application/xml
- Raw payload:**

```

?xml version="1.0" encoding="utf-8"?>
<relationship-instances xmlns="http://www.hp.com/nfvd">
  <relationship-instance>
    <categories/>
    <parent-artifact-id>8f2e6b76-a367-4fa2-8444-637aab6ff73f</parent-artifact-id>
    <child-artifact-id>a2d4aee3-25a4-4e3c-a353-a9679046f7a9</child-artifact-id>
    <status>
      <enabled>true</enabled>
      <label>ENABLED</label>
      <visible-label>ENABLED</visible-label>
    </status>
    <relationship-type>USE</relationship-type>
  </relationship-instance>
</relationship-instances>

```
- SEND button:** Highlighted in orange.

Figure 11: Create Relationship

7.2.4 Replacement of Networking Artifacts

7.2.4.1 Replace NETWORKING:OPENSTACK Artifacts with NETWORKING:OPENSTACK:DCN

1. Query NETWORKING:OPENSTACK associated with each Region of the Datacenter

IMPORTANT: Execute the below steps for each region, **sacramento** region is used as an example.

- a. Provide FF_HOST_IP, FF_PORT details in REST URL.
- b. Select GET HTTP operation.

- c. Provide headers, path and query parameters as shown in below sample.

id: DC ID returned from “Query Datacenter ID” step

expression:

DATACENTER>VIM>AUTHENTICATION>REGION#GENERAL.Name=sacramento>NETWORKING

Request

Host
http://<FF_HOST_IP>:<FF_PORT>

Path
/nfvd/instance/artifact/query/path

Query parameters

id	8f2e6b76-a367-4fa2-8444-637aab6ff73f	×
expression	DATACENTER>VIM>AUTHENTICATION>REGION#GENERAL.Name=sacramento>NETWORKING	×

ADD

Hash

GET POST PUT DELETE Other methods

Raw headers Headers form Headers sets

X-Auth-Token: 3778fe88-e71d-4004-86bc-3188f7fd450b
Content-Type: application/xml

SEND

Figure 12: Query NETWORKING:OPENSTACK associated with Region
Below is the response received.

Status: 200 OK Loading time: 206 ms

Response headers (4) Request headers (2) Redirects (0) Timings

Server: Apache-Coyote/1.1
Content-Type: application/xml
Content-Length: 3921
Date: Fri, 29 Apr 2016 07:06:43 GMT

Raw XML

```
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
<artifact-instances xmlns="http://www.hp.com/nfvd">
  <artifact-instance internal-id="a7fc5aa9-e10d-3380-a3be-da9ed6f213c3" uri="/nfvd/instance/artifact/a7fc5aa9-e10d-3380-a3be-da9ed6f213c3">
    <artifact-version>1</artifact-version>
    <artifact-definition>
      <category>OPENSTACK</category>
      <family>NETWORKING</family>
    </artifact-definition>
    .....
  </artifact-instance>
</artifact-instances>
```

Figure 13: Query Response for NETWORKING:OPENSTACK associated with Region

1. Replace NETWORKING:OPENSTACK with NETWORKING:OPENSTACK:DCN for each Region
 - a. Copy the Response received and paste it in Payload section of REST Client.
 - b. Provide FF_HOST_IP, FF_PORT in the REST URL. Provide the Headers and Path as shown below.
 - c. Change the artifact group to DCN as shown in below sample.
 - d. Perform PUT HTTP operation.

Request

Host
http://<FF_HOST_IP>:<FF_PORT>

Path
/nfvd/instance/artifact

Query parameters

Hash

GET POST PUT DELETE Other methods

Raw headers Headers form Headers sets

X-Auth-Token: 3778fe88-e71d-4004-86bc-3188f7fd450b
Content-Type: application/xml

Raw payload Data form Files (0)

```
<artifact-instances xmlns="http://www.hp.com/nfvd">
  <artifact-instance internal-id="a7fc5aa9-e10d-3380-a3be-da9ed6f213c3" uri="/nfvd/instance/artifact/a7fc5aa9-e10d-3380-a3be-da9ed6f213c3">
    <artifact-version>1</artifact-version>
    <artifact-definition>
      <category>OPENSTACK</category>
      <family>NETWORKING</family>
      <group>DCN</group>
    </artifact-definition>
    .....
  </artifact-instance>
</artifact-instances>
```

Figure 14: REST operation to update NETWORKING:OPENSTACK

7.2.5 Create relationship between NETWORKING and DCN Artifacts



IMPORTANT: Execute the below steps for each region.

- 1 Create Relationship between NETWORKING_OPENSTACK:DCN and SDN_CONTROLLER:DCN. Relationship sample is shown below:
 - a. parent-artifact-id: Id of SDN_CONTROLLER:DCN Artifact. If you use the DCN_Template.xml, the value MUST be 94c80294-2175-4011-bdf2-78db5c689158
 - b. child-artifact-id: NETWORKING:OPENSTACK:DCN Id's returned from "Query NETWORKING:OPENSTACK associated with each Region of the Datacenter" step for each region.

For our example:

```
<relationship-instances xmlns="http://www.hp.com/nfvd">
  <relationship-instance>
    <categories/>
    <parent-artifact-id>94c80294-2175-4011-bdf2-78db5c689158</parent-artifact-id>
    <child-artifact-id>a7fc5aa9-e10d-3380-a3be-da9ed6f213c3</child-artifact-id>
    <status>
      <enabled>>true</enabled>
      <label>ENABLED</label>
      <visible-label>ENABLED</visible-label>
    </status>
    <relationship-type>HANDLE</relationship-type>
  </relationship-instance>
</relationship-instances>
```

Paste the above content in the payload section of the REST client.

Request

Host
http://<FF_HOST_IP>:<FF_PORT>

Path
/nfvd/instance/relationship

Query parameters

Hash

GET POST PUT DELETE Other methods

Raw headers Headers form Headers sets

X-Auth-Token: 3778fe88-e71d-4004-86bc-3188f7fd450b
Content-Type: application/xml

Raw payload Data form Files (0)

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<relationship-instances xmlns="http://www.hp.com/nfvd">
  <relationship-instance>
    <categories/>
    <parent-artifact-id>94c80294-2175-4011-bdf2-78db5c689158</parent-artifact-id>
    <child-artifact-id>a7fc5aa9-e10d-3380-a3be-da9ed6f213c3</child-artifact-id>
    <status>
      <enabled>true</enabled>
      <label>ENABLED</label>
      <visible-label>ENABLED</visible-label>
    </status>
    <relationship-type>HANDLE</relationship-type>
  </relationship-instance>
</relationship-instances>
```

Figure 15: REST operation to create relationship between NETWORKING and DCN