



# HPE NFV Director

OpenStack Discovery 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.

# Contents

Notices .....	1
Preface .....	5
About this guide.....	5
Audience .....	5
Document history.....	5
Chapter 1 Introduction.....	6
1.1 Overview of resource discovery .....	6
1.1.1 Architectural View .....	6
1.1.2 NFV Discovery Process .....	7
1.2 NFV Director resources.....	7
1.2.1 Discovered.....	7
Chapter 2 Installation and Configuration .....	9
2.1 Installation .....	9
2.1.1 Prerequisites .....	9
2.1.2 Installation steps .....	9
2.2 Configuration.....	9
2.3 Enabling and Disabling of discovery process.....	11
2.3.1 Disable discovery even in fresh installation.....	11
2.3.2 Disable discovery temporarily.....	11
2.3.3 Enable Discovery.....	11
2.3.4 Manual Discovery trigger.....	11
2.3.5 Making changes in CA properties .....	12
2.3.6 Track Initial/Incremental Discovery completion.....	12
Chapter 3 DCN Integration.....	13
3.1 Prerequisites.....	13
3.2 Integrate DCN with NFV Director .....	13
3.2.1 Create the SDN Topology manually .....	13
3.2.2 Upload DCN resource (Manual “Discover”).....	15
3.2.3 Connect Datacenter with DCN resources .....	15
3.2.4 Replacement of Networking Artifacts.....	18
3.2.4.1 Replace NETWORKING:OPENSTACK Artifacts with NETWORKING:OPENSTACK:DCN.....	19
3.2.5 Create relationship between NETWORKING and DCN Artifacts .....	21

# List of tables

Table 1: Document history.....	5
--------------------------------	---

# List of figures

Figure 1: OpenStack Discovery Architecture .....	6
Figure 2: OpenStack Discovery NFV Director components.....	7
Figure 3: Discovered resources.....	8
Figure 4: Resources populated with default value .....	8
Figure 5: VIM to AUTHENTICATION relationship .....	10
Figure 6: DCN topology pictorial representation.....	15
Figure 7: Uploading DCN topology into fulfillment .....	15
Figure 8: Query ID of Datacenter .....	16
Figure 9: Response for Datacenter Query .....	16
Figure 10: Query ID of SHARED_NETRESOURCE:DCN .....	17
Figure 11: Response for SDN_CONTROLLER:DCN Query .....	17
Figure 12: Create Relationship .....	18
Figure 13: Query NETWORKING:OPENSTACK associated with Region .....	19
Figure 14: Query Response for NETWORKING:OPENSTACK associated with Region .....	20
Figure 15: REST operation to update NETWORKING:OPENSTACK.....	20
Figure 16: REST operation to create relationship between NETWORKING and DCN .....	22

# 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

<b>Edition</b>	<b>Date</b>	<b>Description</b>
1.0	July 30, 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.

### 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

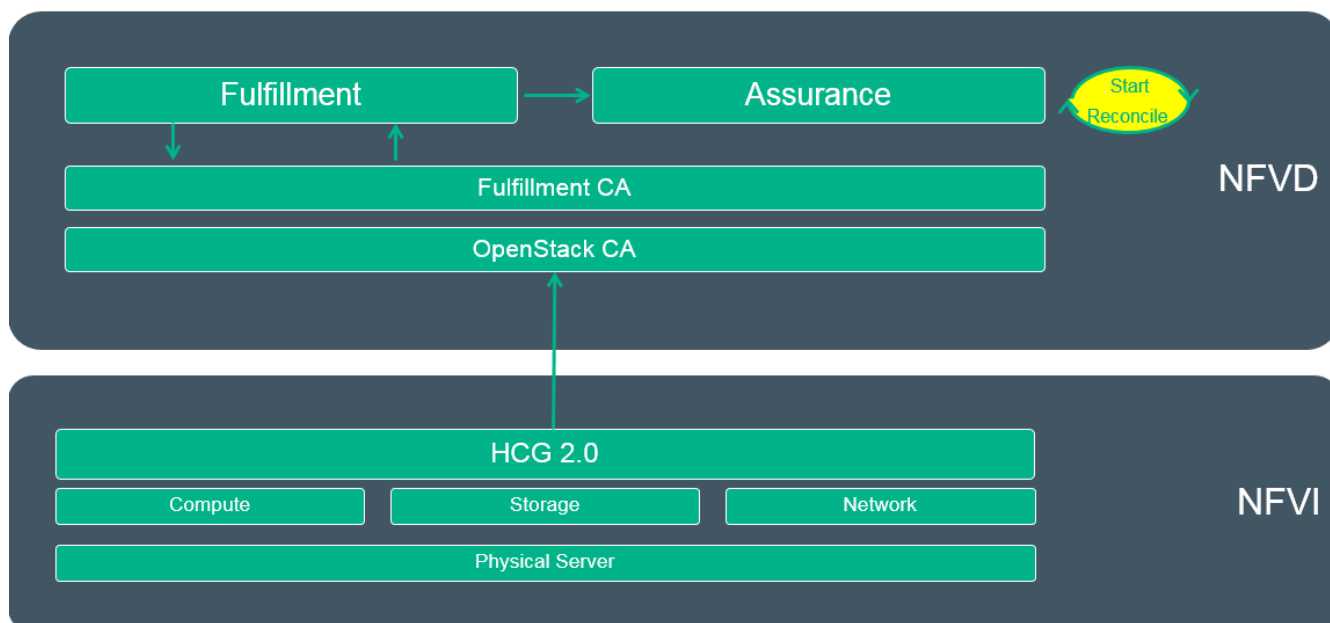


Figure 1: OpenStack Discovery Architecture

## 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 anytime, 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.

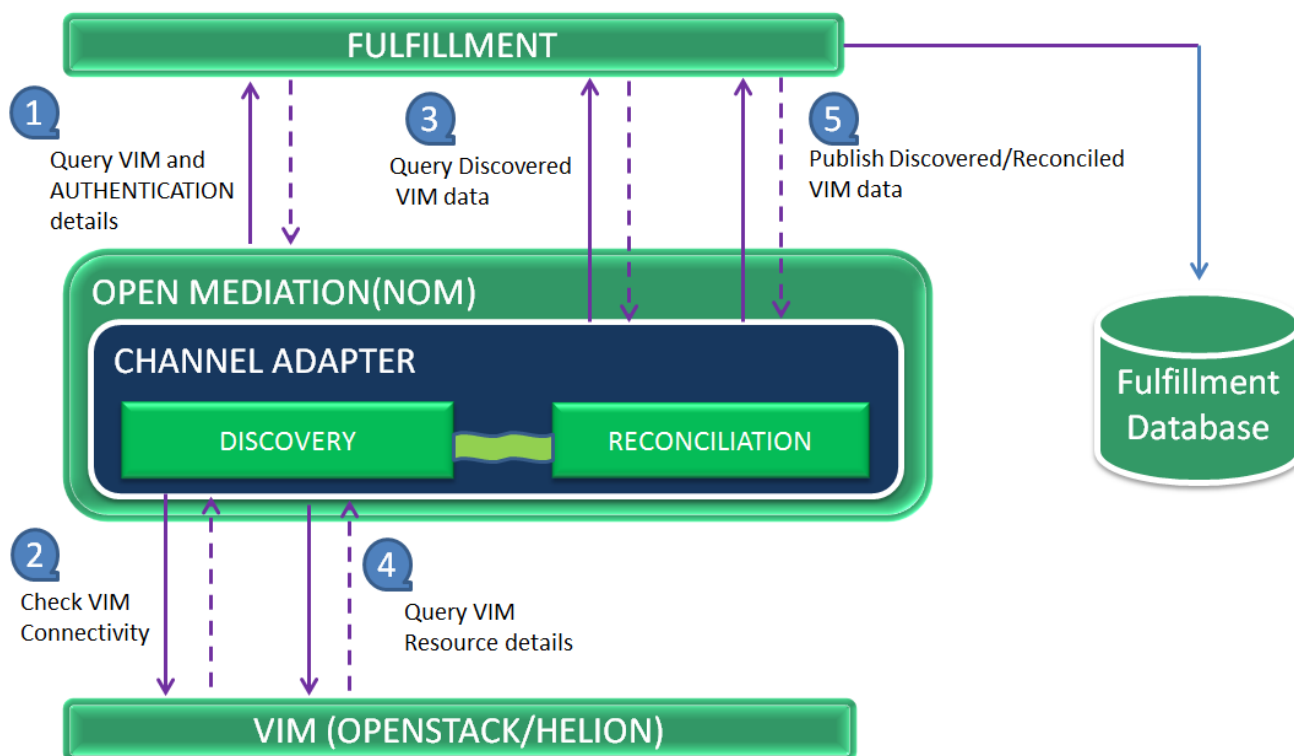


Figure 2: OpenStack Discovery NFV Director components

## 1.2 NFV Director resources

### 1.2.1 Discovered

Following resources are auto discovered from OpenStack or HCG



- ✓ Servers
  - CPU, Memory, Disk
  - Port, Interface
  - Total, used and available capacity
- ✓ Regions
- ✓ Tenant
- ✓ OpenStack Services
- ✓ Hypervisors
  - ESX, KVM, Bare metal
- ✓ Availability Zones
- ✓ Host Aggregates
- ✓ Networks
  - ✓ Subnetworks
    - IPAddress
  - ✓ Virtual Machines
    - vCPU, vMemory, vDisk
    - vPort
  - ✓ LUN/vLUN
  - ✓ Images
  - ✓ Flavor
  - ✓ Carrier Grade
    - NUMA, Huge pages
    - PCI-PT
    - SR-IOV

Figure 3: Discovered resources

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

Thus, VIM and AUTHENTICATION are manually provided as input.

Default values are populated into NFV Director for the following artifacts

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

Figure 4: Resources populated with default value

# Chapter 2 Installation and Configuration

## 2.1 Installation

### 2.1.1 Prerequisites

1. REST Client
2. openstack-ca-1.0.0.zip
3. fulfillment-ca-1.0.0.zip
4. 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
```



**NOTE:** Before running the above command to install the RPM, verify if the package is already installed by running:

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

If it returns the following output, skip the above step.  
nfvd-discovery-common-04.01.000-1.el6.noarch

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
```



**NOTE:** Before running the above command to install the channel adapters, verify if they are already installed by running:

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

If it returns the following output, skip the above steps.

```
INSTALLED    fulfillment-ca-10
INSTALLED    openstack-ca-10
```

Or if it returns the following output, skip the above steps.

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

3. Mandatory VIM and AUTHENTICATION artifact-relationship instances have to be uploaded to fulfillment for discovery to run.

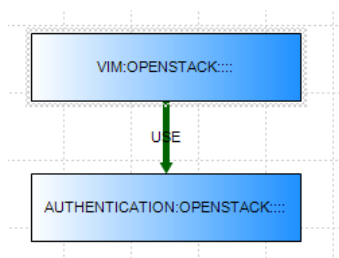


Figure 5: VIM to AUTHENTICATION relationship

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 <FF Host/IP>] [-port <FF Port>] [-vimname <VIM Name>] [-url <Keystone URL>] [-username <Admin User>] [-password <Admin Password>] [-tenantname <Tenant Name>] [-vimcategory <VIM Category>] [-discovertenant <Discover Tenant>] [-nominstance <NOM Instance Number>] [-nomhost <NOM Host Name>] [-vimtype <VIM Type>]
```

Where:

<**FF Host/ IP**>: Hostname or IPAddress of Fulfillment  
 <**FF Port**>: Fulfillment port (eg: 8080)  
 <**VIM Name**>: VIM Name  
 <**Keystone URL**>: VIM Identity/Keystone V2/V3 URL  
 (eg: V2-URL: http:0.0.0.0/5000/v2.0/tokens, V3-URL: http:0.0.0.0/5000/v3/auth/tokens)  
 <**Admin User**>: VIM user with administrator privileges on the tenants to be discovered  
 <**Admin Password**>: VIM Password for the admin user  
 <**Tenant Name**>: Tenant on which the above user has administrative privileges  
 <**VIM Category**>: VIM category, values are either OPENSTACK/HELION --optional  
 (default:OPENSTACK)  
 <**Discover Tenant**>: Comma separated list of tenants that needs to be discovered. --optional (default: blank,  
 discovers all tenants)  
 <**NOM Instance Number**>: NOM Instance number (Run discovery of this VIM on a specific NOM  
 Instance) –optional (default: blank)  
 <**NOM Host Name**>: NOM Hostname (Run discovery of this VIM on a specific NOM Host) --optional  
 (default: blank)  
 <**VIM Type**>: General type of the VIM, values is either OPENSTACK/HCG --optional

## 2.3 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>

### 2.3.1 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
```

### 2.3.2 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
```

### 2.3.3 Enable Discovery

Execute the below script.

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

### 2.3.4 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
```

## 2.3.5 Making changes in CA properties

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

1. disable discovery
2. undeploy 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

---

## 2.3.6 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 3 DCN Integration

## 3.1 Prerequisites

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. .

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.

## 3.2 Integrate DCN with NFV Director

### 3.2.1 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	<a href="https://172.19.244.225:8443/nuage/api/v3_2">https://172.19.244.225:8443/nuage/api/v3_2</a>

- 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 is the pictorial representation of DCN Topology

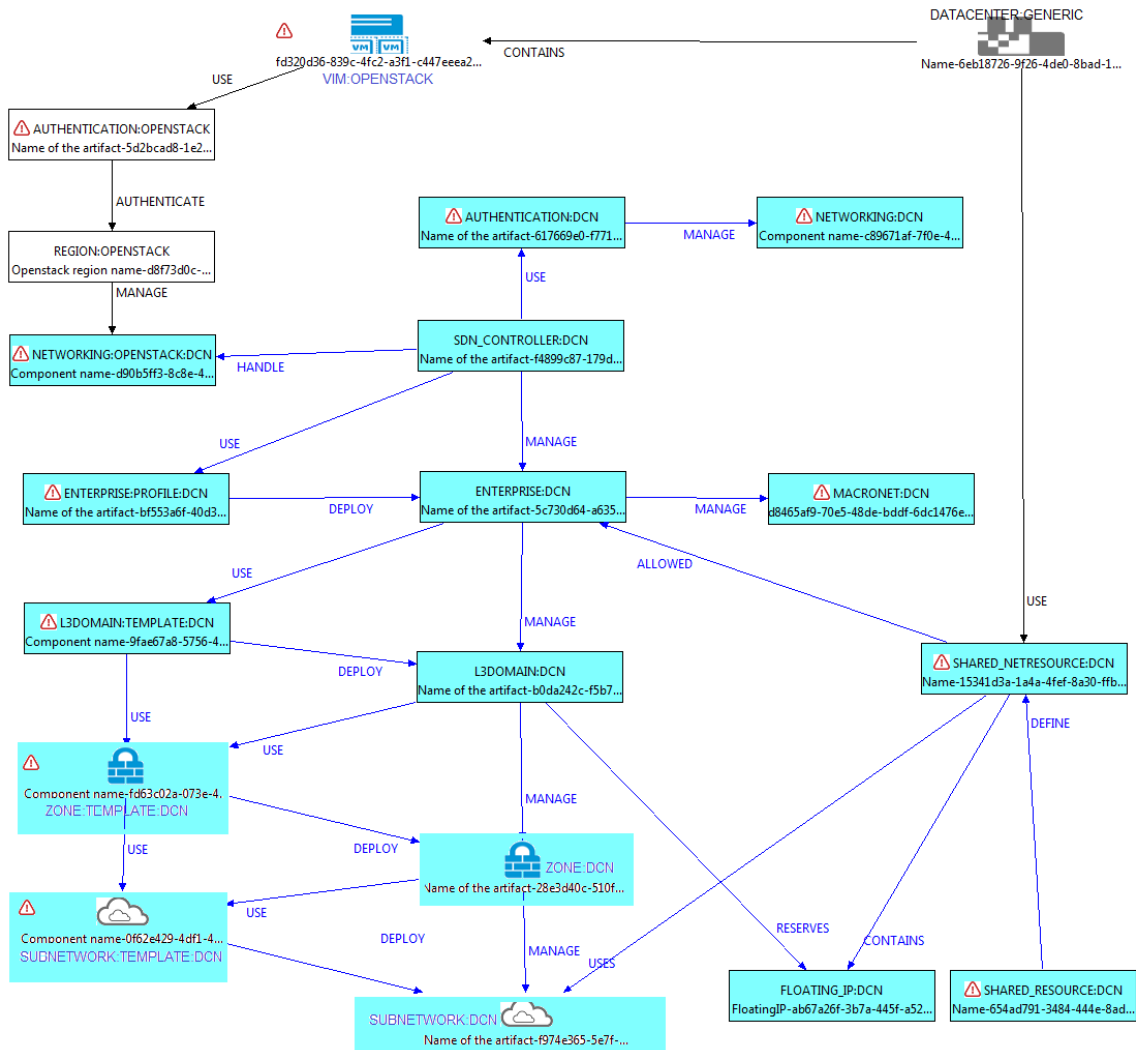


Figure 6: DCN topology pictorial representation

### 3.2.2 Upload DCN resource (Manual “Discover”)

- 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 the REST Client interface with the following configuration:

- Host:** `http://<FF_HOST_IP>:<FF_PORT>`
- Path:** `/nfvd/instance/upload`
- HTTP Method:** POST (selected)
- HTTP 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>
</instances>
```

A **SEND** button is located at the bottom right of the interface.

Figure 7: Uploading DCN topology into fulfillment

### 3.2.3 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:

- Host:** `http://<FF_HOST_IP>:<FF_PORT>`
- Path:** `/nfvd/instance/artifact/query/parameters`
- Query parameters:**
  - `definition`: DATACENTER:GENERIC
  - `attributeFilter`: GENERAL.Name=DC1
  - `exactMatching`: false
- HTTP Method:** GET (selected)
- Raw headers:**
  - `X-Auth-Token`: 3778fe88-e71d-4004-86bc-3188f7fd450b
  - `Content-Type`: application/xml

A blue **SEND** button is visible at the bottom right.

Figure 8: Query ID of Datacenter

The screenshot shows the response details for the query:

- Status:** 200: OK (Loading time: 200 ms)
- Response headers (4):**
  - Server: Apache-Coyote/1.1
  - Content-Type: application/xml
  - Transfer-Encoding: chunked
  - Date: Thu, 28 Apr 2016 13:51:08 GMT
- Request headers (2):** X-Auth-Token, Content-Type
- Redirects (0):** None
- Timings:** None

The response body is shown in XML format:

```
<?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>

```

The XML content is displayed in a 'Raw' view, with a red box highlighting the `<artifact-instance>` element.

Figure 9: 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.

c. INFO.DC.Name attribute filter is used to filter by Datacenter Name

The screenshot displays a REST client interface with the following details:

- 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
- Hash:**
  - Method: GET (selected)
  - Raw headers:
    - X-Auth-Token: 3778fe88-e71d-4004-86bc-3188f7fd450b
    - Content-Type: application/xml

Figure 10: Query ID of SHARED\_NETRESOURCE:DCN

The screenshot shows the response details for a query:

- Status:** 200: OK (Loading time: 200 ms)
- Response headers (4):**
  - 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 11: 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:

- Request:**
  - Host: `http://<FF_HOST_IP>:<FF_PORT>`
  - Path: `/nfvd/instance/relationship`
  - Query parameters: ADD
- Hash:**
  - Method:  GET  **POST**  PUT  DELETE Other methods
  - Content-Type: `application/xml`
- Raw headers:**
  - X-Auth-Token: `3778fe88-e71d-4004-86bc-3188f7fd450b`
  - 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 is highlighted.

Figure 12: Create Relationship

### 3.2.4 Replacement of Networking Artifacts

### 3.2.4.1 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

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

- Request** (Title)
- 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`
- Hash:** (Empty)
- Method:**  GET,  POST,  PUT,  DELETE, Other methods
- Raw headers:**
  - `X-Auth-Token: 3778fe88-e71d-4004-86bc-3188f7fd450b`
  - `Content-Type: application/xml`
- Buttons:** ADD, SEND

Figure 13: 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 14: 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 ▼ application/xml ▼

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 15: REST operation to update NETWORKING:OPENSTACK

## 3.2.5 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 application/xml

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 16: REST operation to create relationship between NETWORKING and DCN