PF6800 Ver. 5.0
Configuration Guide for Red Hat OpenStack

First Edition, October 2013

Legal Notices

Symbols

In this manual, the following two types of symbols are used. These symbols and their meanings are important for proper handling of the PFC.

CHECK:	Points that should be checked when operating devices or software.
NOTE:	Helpful, good-to-know information

Disclaimer

Unless explicitly set forth in a license agreement, the NEC Corporation makes no explicit or implicit guarantees regarding this product and the related documentation, including its merchantability or fitness for a particular purpose, and disclaims all liability pertaining to its handling, use, or attendant trade practices.

Contents

Chapter 1 Configuration of Red Hat OpenStack environment

In this manual the example of how to configure Red Hat OpenStack environment is given.

■CHECK:■ In this manual even though it is mentioned just OpenStack, it means Red Hat OpenStack.

■CHECK:■ OpenStack and PFC coordinate using OpenStack function of WebAPI that is a component of PFC. Please read “PF6800 Ver. 5.0 WebAPI User’s Guide” before configuring the OpenStack environment.

1.1 Summary of system configuration

Here the explanation about system configuration, software, hardware is given.

1.1.1 System configuration and explanation of terminology

Example of system configuration of OpenStack environment explained in this chapter is shown in Figure 1.1.1-1.

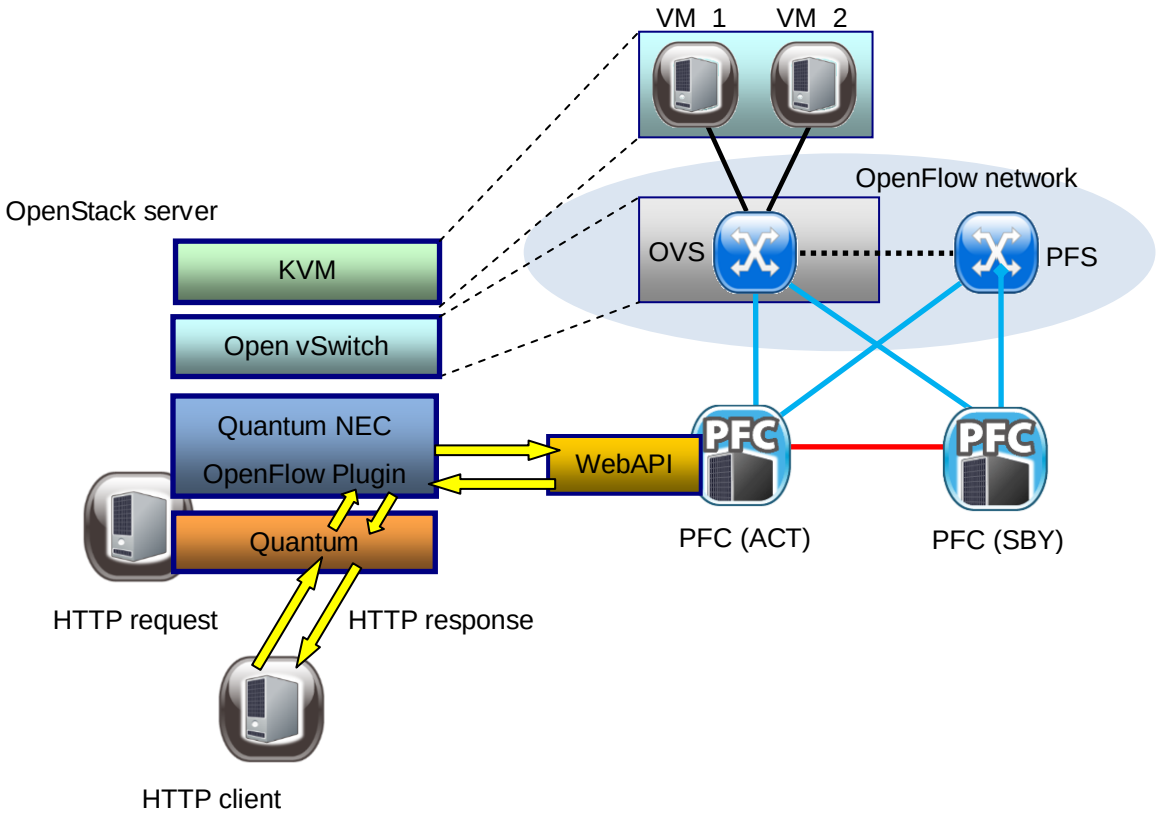

Figure 1.1.1-1 Example of OpenStack environment system configuration that uses WebAPI

Details of system configuration are given in Figure 1.1.1-2.

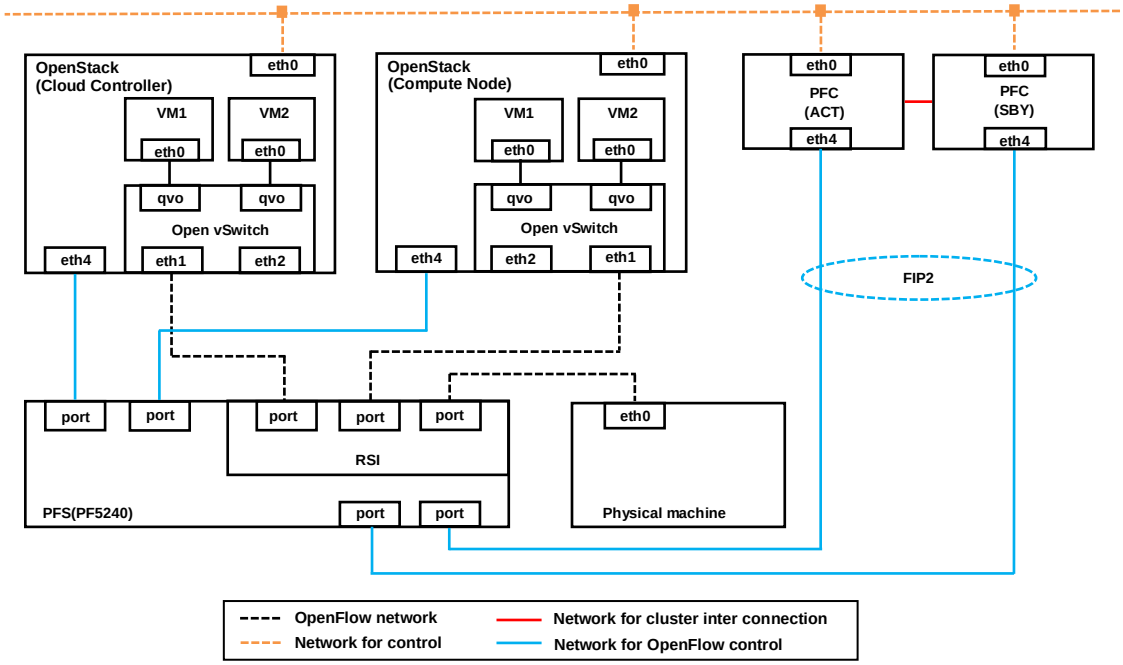

Figure 1.1.1-2 Details of System Configuration Diagram

The terminology used in the explanation of OpenStack environment is given in "Table 1.1.1-1 Terms used in OpenStack Environment".

Table 1.1.1-1 Terms used in OpenStack Environment

name of function	details
OpenStack (Red Hat OpenStack)	It is the core component of commercial product " Red Hat Enterprise Linux OpenStack Platform" based on the open source cloud platform management software " OpenStack"
Quantum	It is a generic name of network connection function provided by OpenStack project.
Quantum NEC OpenFlow Plugin	It is a plug in provided by NEC for Quantum. Coordination between Quantum and PFC (WebAPI) is possible through this plug in. Hereafter described as "NEC Plugin".
Keystone	It is a generic name of authentication function provided by OpenStack project.
Open vSwitch	It is open source virtual software switch. In the OpenStack environment it is installed on OpenStack server.

KVM	It is virtualization function that runs on Linux. This function needs to be run on OpenStack server in order to run VM in the OpenStack environment.

1.1.2 Details of hardware and software

This section explains the hardware and software that configures OpenStack environment.

1.1.2.1 Details of hardware

The hardware required to configure OpenStack environment is described below.

Table 1.1.2-1 Hardware

Name	Model	Remarks
OpenStack server	Use the model equivalent to (or above) Express5800 R120a-2	Installation of OpenStack and Open vSwitch.
PFC	Refer "1.3 Preparation for Installing PFC" of installation guide.	2 Machines are required as server for running PFC and WebAPI (ACT and SBY)
PFS	Use model of PF5xxx (PF5200, PF5800 etc)	None

■CHECK:■ In the explanation of subsequent sections, the expressions used in "Name" column of above table are used.

1.1.2.2 Software details

The software required to configure OpenStack environment is given below.

Table 1.1.2-2 Software

Name	version	Remarks
PFC	V5.0	None.
Red Hat Enterprise Linux Server	6.4 for 64-bit	Install as OS of OpenStack server.
Red Hat OpenStack	3.0 (Grizzly)	Includes Quantum.
NEC Plugin	Grizzly supported version	Install on OpenStack server.
Open vSwitch	1.2.2	Install on OpenStack server.

RESTClient	2.4 GUI	Install on client machines that access Quantum.

■CHECK:■ The versions mentioned in the above table may change because of version up of OpenStack and PFC.

1.2 Advance preparation

1.2.1 Setting PFC server

1.2.1.1 Installing PFC

For installation procedure of PFC, refer Configuration Guide.

Table 1.2.1-1 Setting IP address of PFC

Where to set	Name of the device	IP address
PFC1 (ACT)	eth0	Any global IP address
	eth4	Refer "Figure 1.2.1-1 Details of System Configuration Diagram" and specify the value that is adequate to configuration environment.
PFC2 (SBY)	eth0	Any global IP address
	eth4	Refer "Figure 1.2.1-2 Details of System Configuration Diagram" and specify the value that is adequate to configuration environment.
Floating IP for SecureChannel	FIP2	Refer "Figure 1.2.1-3 Details of System Configuration Diagram" and specify the value that is adequate to configuration environment.

1.2.1.2 Setting WebAPI

■CHECK:■ The settings explained here need to be performed for ACT , as well as for SBY.

1.2.1.2.1 Setting auto start of WebAPI

■CHECK:■ For details of settings, refer "2.1.2 Specifying the WebAPI Startup Parameters" of "PF6800 Ver. 5.0 WebAPI User's Guide".

1.2.1.2.2 Setting port of OpenStack function

■CHECK:■ For details of settings, refer "2.1.2 Specifying the WebAPI Startup Parameters" of "PF6800 Ver. 5.0 WebAPI User's Guide".

1.2.1.2.3 Restarting PFC

Restart both ACT and SBY PFC servers for the settings to become effective.

```
# pfc_stop_cluster                    (stops cluster)
# reboot                              (restarts)
```

1.2.1.3 Setting of auto saving startup-configuration

■CHECK:■ For setting details refer "1.2.6 Applying the Settings by WebAPI to startupconfiguration" of "PF6800 Ver. 5.0 WebAPI User's Guide".

1.2.2 Settings for OpenStack server

1.2.2.1 BIOS settings

To operate OpenStack, enable the system Virtualization Technology on the server on which OpenStack will run.

1. Power On the server on which OpenStack server will run.
2. Press F2 key when the server is starting (when NEC logo is being displayed) and display BIOS menu.
3. Select [Main]→[Processor Settings].
4. Change [Virtualization Technology:] from [Disabled] to [Enabled].
5. Press F10, save the settings and exit BIOS.

1.2.2.2 Installing OS

Install Red Hat Enterprise Linux Server 6.4 for 64-bit on the machine on which OpenStack server

will run.

1.2.2.3 Setting network

1.2.2.3.1 Setting network interface

Configure the /etc/sysconfig/network-scripts/ifcfg-eth0 of the server on which OpenStack server will run as shown below. (The items that are not described here should be kept as it is in it's default state.)

```
ONBOOT=yes
BOOTPROTO=none
IPADDR=[IP address to be allocated to eth0 of corresponding server]
PREFIX=[ Network prefix length corresponding to above IP address]
GATEWAY=[Appropriate gateway address corresponding to above IP address]
DNS1=[IP address of DNS server]
```

Configure /etc/sysconfig/network-scripts/ifcfg-eth4 as shown below.

```
ONBOOT=yes
BOOTPROTO=none
IPADDR=[IP address to be allocated to eth4 of corresponding server]
NETMASK=[subnet mask corresponding to above IP address]
```

Configure /etc/sysconfig/network of server on which OpenStack server will run as shown below.

```
NETWORKING=yes
HOSTNAME=[host name of the server on which OpenStack server will run]
GATEWAY=[IP address of default gateway]
```

1.2.2.3.2 Restarting network connection

Restart the network setting for setting to be reflected.

```
$ /etc/init.d/network restart
```

1.2.2.4 Proxy server settings (only when required)

Perform the settings for the proxy server when internet is to be accessed through proxy server

from the server that runs OpenStack server.

1.2.2.4.1 Setting environment variables

Set environment variables http_proxy, https_proxy, no_proxy.

```
http_proxy="http://[Address of proxy server]:[Port No of proxy server]/"
https_proxy="http://[Address of proxy server]:[Port No of proxy server]/"
no_proxy=[Address of server that will run keystone]
```

1.2.2.4.2 Settings of /etc/sudoers

To enable to inherit the settings of proxy server even at the time of execution of sudo, add following description in /etc/sudoers.

```
Defaults        env_keep = "http_proxy https_proxy no_proxy"
```

1.2.2.4.3 Settings of /etc/rhsm/rhsm.conf

Do the following settings to use subscription-manager.

/etc/rhsm/rhsm.conf

```
proxy_hostname = [Address of proxy server]
proxy_port = [Port No of proxy server]
```

1.2.2.4.4 Settings of yum

Do the following settings to install the package through yum command.

/etc/sysconfig/rhn/up2date

```
enableProxy=1
httpProxy=[Address of proxy server]:[Port No of proxy server]
```

/etc/yum.conf

```
proxy=http://[Address of proxy server]:[Port No of proxy server]/
```

1.2.2.5 Registration of subscription

Register the subscription using subscription-manager. For registration procedure, please refer the document of Red Hat Enterprise Linux.

1.3 Installing software

1.3.1 Installing OpenStack

Do the installation as per the instructions given the “Red Hat OpenStack Getting Started Guide”

```
https://access.redhat.com/site/documentation/en-US/Red_Hat_OpenStack/3/html/Getting_Started_Guide/index.html
```

Use environment configuration support tool called as PackStack and perform consolidated installation and settings. If various components that constitute OpenStack required to be installed and set individually then refer the following “Red Hat OpenStack Installation and Configuration Guide”

```
https://access.redhat.com/site/documentation/en-US/Red_Hat_OpenStack/3/html/Installation_and_Configuration_Guide/index.html
```

1.3.2 Installing NEC Plugin

1.3.2.1 Advance preparation

Stop Open vSwitch Plugin and cancel auto start setting.

```
$ service quantum-openvswitch-agent stop
$ chkconfig quantum-openvswitch-agent off
```

Stop Quantum Server

```
$ service quantum-server stop
```

1.3.2.2 Install the package

Install the package using yum command.

```
$ yum install -y openstack-quantum-nec
```

1.3.2.3 Enable NEC Plugin

Initialize the database that will be used by NEC Plugin.

```
$ mysql -u root -p -e "DROP DATABASE IF EXISTS quantum_nec"
$ mysql -u root -p -e "CREATE DATABASE IF NOT EXISTS quantum_nec"
$ mysql -u root -p -e "grant all on quantum_nec.* to 'quantum'@'%'"
$ mysql -u root -p -e "grant all on quantum_nec.* to 'quantum'@'localhost'"
```

Edit the configuration file（/etc/quantum/plugins/nec/nec.ini）of NEC Plugin.

Set the items of sql_connection of [DATABASE] section as below.

```
sql_connection = mysql://quantum:[quantum user password of MySQL]@[Host name or IP address of the host that installed MySQL]/quantum_nec?charset=utf8
```

■NOTE:■ If OpenStack is installed using PackStack, quantum user password of MySQL is mentioned in the Answer file generated by the PackStack.

Set the [OFC] section as below.

```
host = [IP address of PFC that provided the WebAPI service]
port = 8888
driver = pfc
```

Set the root_helper item of [AGENT] section as below.

```
root_helper = sudo quantum-rootwrap /etc/quantum/rootwrap.conf
```

Add [SECURITYGROUP] section and set as below.

```
# Firewall driver for realizing quantum security group function.
firewall_driver =
quantum.agent.linux.iptables_firewall.OVSHybridIptablesFirewallDriver
```

■CHECK:■ Items of firewall_driver are displayed spreading over 2 rows but actually mention those on 1 row.

Edit the configuration file of Quantum (/etc/quantum/quantum.conf)

Set the [DEFAULT] section core_plugin and api_extensions_path items as shown below.

```
core_plugin = quantum.plugins.nec.nec_plugin.NECPluginV2
api_extensions_path =
/usr/lib/python2.6/site-packages/quantum/plugins/nec/extensions/
```

■CHECK:■ Items of api_extensions_path are displayed spreading over 2 rows but actually mention those on 1 row.

Replace already allocated configuration file of Open vSwitch Plugin with configuration file of NEC Plugin.

```
$ rm -f /etc/quantum/plugin.ini
$ ln -s /etc/quantum/plugins/nec/nec.ini /etc/quantum/plugin.ini
```

Connect Open vSwitch with PFC.

```
$ ovs-vsctl set-controller br-int tcp:[IP address of PFC that is the connection destination of SecureChannel]
```

Start the services of Quantum Server, NEC Plugin.

```
$ service quantum-server start
$ service quantum-nec-agent start
$ chkconfig quantum-nec-agent on
```

1.4 Setting network

1.4.1 Setting PFS

Explanation regarding setting PFS is given.

1.4.1.1 Setting minicom

Explanation regarding installation and initial settings of minicom is given.

1.4.1.1.1 Installing minicom

Log in to the OpenStack server that is serially connected to PFS, and install minicom through apt-get command.

```
$ yum install -y minicom
```

1.4.1.1.2 Settings display screen of minicom

Display the settings screen of minicom.

```
$ minicom -s
```

1.4.1.1.3 Settings of minicom

Select "Serial port setup" of "configuration" menu, and do the settings as shown below.

```
A -    Serial Device      : /dev/ttyS0
B - Lockfile Location     : /var/lock
C -   Callin Program      :
D -  Callout Program      :
E -    Bps/Par/Bits       : 9600 8N1
F - Hardware Flow Control : Yes
G - Software Flow Control : No
```

■CHECK:■ Change the number at the end of /dev/ttyS0 to the value that corresponds to the port connected.

1.4.1.1.4 Saving the settings of minicom

Select "Exit" of [configuration] menu after selecting the "Save setup as dfl" of [configuration] menu.

1.4.1.2 Log in through serial

The method to log in to PFS through serial from OpenStack server serially connected to PFS is explained.

1.4.1.2.1 Starting minicom

Start minicom on the OpenStack server that is serially connected to PFS.

```
$ minicom
```

1.4.1.2.2 Log in

Use log in ID as "operator" and log in.

```
login: operator
```

1.4.1.3 Log in via LAN (only when required)

Do the settings to log in to PFS via LAN depending on the need.
For details refer "2.3 PF5240 Setup" of Quick Start Guide and "3.2.2 Initial settings of PF5240" of Configuration guide.

1.4.1.4 RSI settings

Do the setting to run PFS as RSI (Real Switch Instance).
For details refer "2.3 PF5240 Setup" of Quick Start Guide and "3.2.2 Initial Settings of PF5240" of Configuration guide.

1.4.2 Creating user and tenant

1.4.2.1 Setting the administrator information and updating it in the operation environment

To make the authentication as administrator simple, create keystonerc_admin file under home directory of user. Contents of the file are as below.

```
export OS_USERNAME=admin
export OS_PASSWORD=[Details of ADMIN_PASSWORD set by localrc]
export OS_TENANT=NAME=admin
export OS_AUTH_URL=http://[IP address of the machine on which keystone is running]:35357/v2.0/
```

■NOTE:■ If OpenStack is installed using PackStack, then keystonerc_admin file will be created automatically.

Update the contents of the file in the operation environment.

```
$ soruce ~/keystonerc_admin
```

1.4.2.2 Create general user

Create general user using the command given below. In the example the user name is taken as "username" and the password is taken as "secret".

```
$ keystone user-create --name username --pass secret
```

1.4.2.3 Create the role

Create the role using following command. The role name taken in the example is "user".

```
$ keystone role-create --name user
```

1.4.2.4 Create tenant

Create the tenant using following command. The tenant name taken in the example is "tenant".

```
$ keystone tenant-create --name tenant
```

1.4.2.5 Add the user to tenant

Add the user in tenant by specifying created general user, role, various IDs of tenant as arguments. The command is given below.

```
$ keystone user-role-add --user [ID displayed in user-create] ¥
                         --role [ID displayed in role-create] ¥
                         --tenant_id [ID displayed in tenant-create]
```

1.4.2.6 Setting user information

To make the authentication as user simple, create keystonerc_admin file under home directory of user. Contents of the file are as below.

```
export OS_USERNAME=username
export OS_PASSWORD=secret
export OS_TENANT=NAME=tenant
export OS_AUTH_URL=http://[IP address of the machine on which keystone is running]:35357/v2.0/
```

1.4.3 Settings of Quantum network

Explanation about setting Quantum network manually is given here.

1.4.3.1 Network settings

Explanation about the Quantum network settings in the server on which OpenStack server is running is given.

1.4.3.1.1 Checking the tenant ID

Check the tenant ID for which network is to be created. (In this explanation TenantA is taken as target)

```
$ keystone tenant-list
+----------------------------------+---------+---------+
|                id                |   name  | enabled |
+----------------------------------+---------+---------+
| 247e478c599f45b5bd297e8ddbbc9b6a | TenantA |   True  |
| 2b4fec24e62e4ff28a8445ad83150f9d | TenantC |   True  |
| 3719a4940bf24b5a8124b58c9b0a6ee6 | TenantB |   True  |
| 5fcfbc3283a142a5bb6978b549a511ac |   demo  |   True  |
| b7445f221cda4f4a8ac7db6b218b1339 |  admin  |   True  |
+----------------------------------+---------+---------+
```

1.4.3.1.2 Checking the user ID

Check the user ID to be associated with the VM to be created. (In this explanation UserA is taken as target)

```
$ keystone user-list
+----------------------------------+-------+---------+-------------------+
|                id                |  name | enabled |       email       |
+----------------------------------+-------+---------+-------------------+
| 5a9149ed991744fa85f71e4aa92eb7ec |  demo |   True  |                   |
| 5b419c74980d46a1ab184e7571a8154e | admin |   True  | admin@example.com |
| 8e37cb8193cb4873a35802d257348431 | UserC |   True  |                   |
| c11f6b09ed3c45c09c21cbbc23e93066 | UserB |   True  |                   |
| ca567c4f6c0942bdac0e011e97bddbe3 | UserA |   True  |                   |
+----------------------------------+-------+---------+-------------------+
```

1.4.3.1.3 Updating user information in operation environment

Update the contents of file set in "1.4.2.6 Setting user information" in operation environment.

```
$ soruce ~/keystonerc_username
```

1.4.3.1.4 Creating network

Create network using quantum net-create command. $TENANT_ID used in the example specifies the ID of TenantA.

```
$ quantum net-create net1
Created a new network:
+---------------------------+--------------------------------------+
| Field                     | Value                                |
+---------------------------+--------------------------------------+
| admin_state_up            | True                                 |
| id                        | 04457b44-e22a-4a5c-be54-a53a9b2818e7 |
| name                      | net1                                 |
| provider:network_type     | flat                                 |
| provider:physical_network | physnet1                             |
| provider:segmentation_id  |                                      |
| router:external           | False                                |
| shared                    | True                                 |
| status                    | ACTIVE                               |
| subnets                   |                                      |
| tenant_id                 | 247e478c599f45b5bd297e8ddbbc9b6a     |
+---------------------------+--------------------------------------+
```

1.4.3.1.5 Creating subnet

Crate the subnet for the abovementioned network using quantum subnet-create command.

```
$ quantum subnet-create net1 30.0.0.0/24
Created a new subnet:
+------------------+--------------------------------------------+
| Field            | Value                                      |
+------------------+--------------------------------------------+
| allocation_pools | {"start": "30.0.0.2", "end": "30.0.0.254"} |
| cidr             | 30.0.0.0/24                                |
| dns_nameservers  |                                            |
| enable_dhcp      | True                                       |
| gateway_ip       | 30.0.0.1                                   |
| host_routes      |                                            |
| id               | b8e9a88e-ded0-4e57-9474-e25fa87c5937       |
| ip_version       | 4                                          |
| name             |                                            |
| network_id       | 04457b44-e22a-4a5c-be54-a53a9b2818e7       |
| tenant_id        | 247e478c599f45b5bd297e8ddbbc9b6a           |
```

```
+------------------+--------------------------------------------+
```

1.4.3.2 Connecting network

Here the procedure to connect created VM to Quantum network is explained.

1.4.3.2.1 Starting VM and creating Quantum network

Start the registered VM image (called as TenantA_VM1 in the example below) using nova boot command. Also connect to the network at the same time.

```
$ nova boot --flavor m1.tiny --image tty-quantum ¥
  --nic net-id=04457b44-e22a-4a5c-be54-a53a9b2818e7 TenantA_VM1
```

1.4.3.2.2 Checking VM and network

Check that the VM and network are associated using nova list command.

```
$ nova list
+--------------------------------------+-------------+--------+--------------
-------+
| ID                                   | Name        | Status | Networks
       |
+--------------------------------------+-------------+--------+--------------
-------+
| 09923b39-050d-4400-99c7-e4b021cdc7c4 | TenantA_VM1 | ACTIVE | sharednet1=30
.0.0.3 |
+--------------------------------------+-------------+--------+--------------
-------+
```

1.4.3.3 Registration of port for connecting physical machine

The connection destination port needs to be associated with OpenStack in order to connect the physical machine to OpenStack network or legacy network. This section explains the procedure about the same.

1.4.3.3.1 Checking the connection destination port

Check the datapath-id and port No of connection destination port by show topology command of

physical network of PFC.

1.4.3.3.2 Creating Quantum Port

Create Quantum port using quantum port-create command.

```
$ quantum port-create --name PORT1 --admin-state-down ¥
  --mac-address 00:1b:21:58:cb:66 04457b44-e22a-4a5c-be54-a53a9b2818e7
Created a new port:
+----------------+--------------------------------------------------------------
---------------------------+
| Field          | Value
                            |
+----------------+--------------------------------------------------------------
---------------------------+
| admin_state_up | False
                            |
| device_id      |
                            |
| device_owner   |
                            |
| fixed_ips      | {"subnet_id": "d668ffc9-6ecf-486e-aeb4-f3804333002b", "ip
_address": "192.168.200.6"} |
| id             | 54d359e2-9e73-45ed-8386-48802668b279
                            |
| mac_address    | 00:1b:21:58:cb:66
                            |
| name           | baremetal
                            |
| network_id     | 04457b44-e22a-4a5c-be54-a53a9b2818e7
                            |
| status         | ACTIVE
                            |
| tenant_id      | 247e478c599f45b5bd297e8ddbbc9b6a
                            |
+----------------+--------------------------------------------------------------
---------------------------+
```

■CHECK:■	Do not forget --admin-state-down option.
■CHECK:■	When physical machine is to be connected, specify MAC address of NIC to be connected by --mac-address option. --mac-address option is not required when connecting to legacy network.
■CHECK:■	Specify UUID of Quantum network which is the connection destination in the last argument of quantum port-create command.

1.4.3.3.3 Registration of port information in OpenStack DB

Register port information in the database (mysql) of OpenStack.

```
$ mysql quantum_nec -e 'INSERT INTO portinfos VALUES ¥
  ("54d359e2-9e73-45ed-8386-48802668b279", "0x000100255c669d28", 2, 65535, ¥
  "00:1b:21:58:cb:66")';
```

■CHECK:■ quantum_nec and portinfos of argument are the fixed values defined in database.

■CHECK:■ Meaning of values within brackets that are there after VALUES is as below. Read a ppropriately based on the environment.

- 1st …UUID of Quantum port created in "1.4.3.3.2 Creation of Creating Quantum Port"
- 2nd, 3rd …datapath-id and port No confirmed by "1.4.3.3.1 Checking the connection destination port"
- 4th…VLAN ID (65535 fixed)
- 5th…MAC address of machine to be connected (not required if connecting to legacy network)

1.4.3.3.4 Activation of port

Activate Quantum port using quantum port-update command. .

```
$ quantum port-update 54d359e2-9e73-45ed-8386-48802668b279 ¥
  --admin_state_up True
Updated port: 54d359e2-9e73-45ed-8386-48802668b279
```

■CHECK:■ Specify the UUID of Quantum port created in "1.4.3.3.2 Creating Creating Quantum Port" immediately after the port-update.

1.4.3.4 Setting router

1.4.3.4.1 Creating router

Router can be created using quantum router-create command.

```
$ quantum router-create router-name
```

■CHECK:■ Specify the name of the router to be created just after router-create.

1.4.3.4.2 Deleting router

Router can be deleted by using quantum router-delete command.

```
$ quantum router-delete aefc810b-4b1e-4a31-b027-d6be8904179e
```

■CHECK:■ Specify the ID of the router to be deleted just after router-delete. Router ID can be checked by quantum router-list command.

1.4.3.4.3Adding router interface

Router interface can be added using quantum router-interface-add command.

```
$ quantum router-interface-add aefc810b-4b1e-4a31-b027-d6be8904179e ¥
  cd9764c9-d94d-456a-b6ce-de0562cf0405
```

■CHECK:■ Specify in the sequence of ID of target router, ID of target subnet just after the router-interface-add. Router ID can be checked by quantum router-list command and subnet ID can be checked by quantum subnet-list command.

1.4.3.4.4Deleting router interface

Interface of router can be deleted using quantum router-interface-delete command.

```
$ quantum router-interface-delete aefc810b-4b1e-4a31-b027-d6be8904179e ¥
  cd9764c9-d94d-456a-b6ce-de0562cf0405
```

■CHECK:■ Specify in the sequence of ID of target router, ID of target subnet just after the router-interface-delete. Router ID can be checked by quantum router-list command and subnet ID can be checked by quantum subnet-list command.

1.4.3.5 Setting static route of router

Static route can be set when the type of router is vrouter.

1.4.3.5.1 Adding static route of router

Static route can be added using quantum router-update command.

```
$ quantum router-update --route destination=20.9.1.0/24,nexthop=10.3.3.7 ¥
  router-name
```

■CHECK:■ Set the IP address of destination network and gateway IP address just after the –ro ute and specify the name of the router that sets corresponding path in the end.

When separate route information is to be added for the router for which route information is already added, it is required to specify already set route information once again. (the highlighted part in the below example is the information that was already set in the above example).

```
$ quantum router-update --route destination=20.9.1.0/24,nexthop=10.3.3.7 ¥
  --route destination=20.11.1.0/24,nexthop=10.4.3.9 router-name
```

1.4.3.5.2Deletion of static route of router

Route of the router can be deleted using quantum router-update command.

Information of all the routes can be deleted by specifying --no-routes option.

```
$ quantum router-update --no-routes router-name
```

Information of some routes can be deleted by re specifying the router information that is not in scope of deletion using --route option. (In the below example, highlighted part of "1.4.3.5.1 Adding static route of router") is deleted.

```
$ quantum router-update --route destination=20.11.1.0/24,nexthop=10.4.3.9 ¥
  router-name
```

1.5 Setting HTTP client

Settings of HTTP client are explained.

1.5.1 Setting HTTP client

1.5.1.1 Starting HTTP client

Starting method of RESTClient 2.4 GUI Edition to be used as HTTP client is explained.

1.5.1.1.1 Getting the RESTClient 2.4 GUI

RESTClient 2.4 GUI which is HTTP Client can be downloaded to any terminal (Windows of Linux etc) from following URL)

```
http://rest-client.googlecode.com/files/restclient-ui-2.4-jar-with-dependencies.jar
```

1.5.1.1.2 Starting RESTClient 2.4 GUI

Start the downloaded RESTClient 2.4 GUI.

```
$ java -jar restclient-ui-2.4-jar-with-dependencies.jar
```

1.5.2 Using QuantumAPI

The method to access PFC from RESTClient 2.4 GUI through Quantum and Quantum NEC OpenFlow plugin is explained here.

1.5.2.1 Sending Request to Quantum (Checking virtual network)

The method to check that the virtual network is created by sending Request to Quantum from RESTClient 2.4 GUI is explained.

1.5.2.1.1 Starting RESTClient 2.4 GUI

Start RESTClient 2.4 GUI.

```
$ java -jar restclient-ui-2.4-jar-with-dependencies.jar
```

1.5.2.1.2 Checking the creation of virtual network

Send Request from RESTClient 2.4 GUI to PFC ad check that the virtual network has been created.

1. Enter as below in the URL of HTTP Request of RESTClient 2.4 GUI, specify GET in the HTTP Method of Method tab and press Go button.

```
http://[IP address of Quantum]:9696/v1.0/tenants/
                    247e478c599f45b5bd297e8ddbbc9b6a/networks/detail.json
```

2. Check that the Status of HTTP Response under RESTClient 2.4 GUI is as below.

```
HTTP/1.1 200 OK
```

3. Check that he contents of Header tab of HTTP Response are as below.

```
<HTTP Header>      <Value>
Content-Type     : application/json
Content-Length   : 147
Date             : Tue, 24 Apr 2012 07:25:48 GMT (date is optional)
Connection       : keep-alive
```

4. Check that the contents of Body tag of HTTP Response are as below and check that virtual network is created.

```
{
  "networks" : [ {
    "name" : "public",
    "id" : "ad6ffda9-4e1e-40a2-8390-665381574e35"
  }, {
    "name" : "private",
    "id" : "304680a7-53af-4baa-b298-753c89c0f238"
```

```
   } ]
}
```

PF6800 Ver. 5.0

Configuration Guide for Red Hat OpenStack

First Edition, October 2013