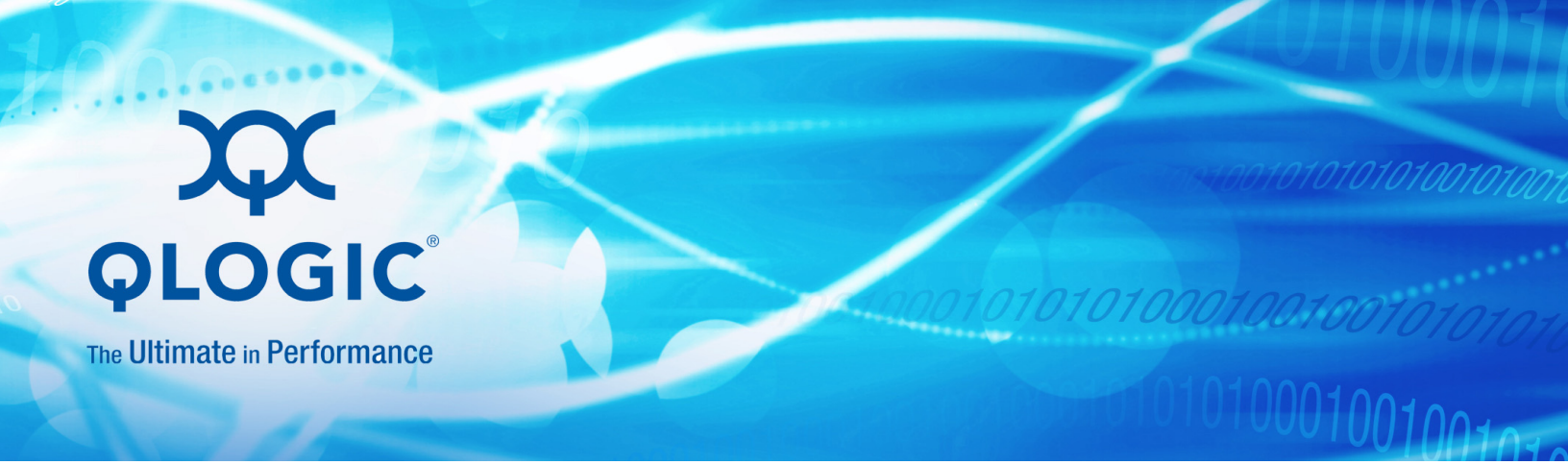


User's Guide

OpenStack® Deployment with VXLAN Configuration

QLogic® 3400 and 8400 Series Adapters

83840-546-03 02

Document Revision History

Rev 01, May 15, 2015

Rev 02, May 22, 2015

Changes	Sections Affected
Removed OpenStack release name.	Title page
Removed PRELIMINARY from footer.	All
Added OpenStack release names.	"Intended Audience" on page v "Audience" on page 1
In the third bullet, added web site for OpenStack (Kilo) configuration information.	"Prerequisites" on page 3
Added a new last paragraph about using the latest QLogic out-of-box drivers.	"VXLAN Configuration with the QLogic 3400/8400 Series Adapters" on page 4
In the second paragraph, removed the reference to OpenStack (Juno).	"Known Issue" on page 15

Table of Contents

Preface

Intended Audience

This guide is intended for OpenStack (Juno and Kilo) users who want to configure virtual extensible LAN (VXLAN) with the QLogic 3400/8400 Series Adapters.

What Is in This Guide

This preface specifies the intended audience, explains the typographic conventions used in this guide, lists related documents, and provides technical support and contact information.

The remainder of this guide is organized into the following chapters:

- Chapter 1 Introduction describes the audience for this guide and outlines the OpenStack architecture.
- Chapter 2 Example Architecture describes the OpenStack 4 node architecture.
- Chapter 3 Prerequisites describes what you need to use OpenStack with VXLAN configuration using the QLogic 3400/8400 Series Adapters.
- Chapter 4 VXLAN Configuration with the QLogic 3400/8400 Series Adapters describes how to install packages, configure the system, and configure VXLAN using the QLogic 3400/8400 Series Adapters.
- Chapter 5 Configuration Deployment with VXLAN describes how to deploy OpenStack with VXLAN.
- Chapter 6 Testing provides a sample testing procedure.
- Chapter 7 Known Issue describes a known issue with open virtual search (OVS) and OpenStack.

Related Materials

For information about downloading documentation from the QLogic Web site, see “Downloading Updates” on page viii.

Documentation Conventions

This guide uses the following documentation conventions:

- Text in blue font indicates a hyperlink (jump) to a figure, table, or section in this guide, and links to Web sites are shown in underlined blue. For example:
  - Table 9-2 lists problems related to the user interface and remote agent.
  - See "Installation Checklist" on page 6.
  - For more information, visit www.qlogic.com.
- Text in **bold** font indicates user interface elements such as a menu items, buttons, check boxes, or column headings. For example:
  - Click the **Start** button, point to **Programs**, point to **Accessories**, and then click **Command Prompt**.
  - Under **Notification Options**, select the **Warning Alarms** check box.
- Text in `Courier` font indicates a file name, directory path, or command line text. For example:
  - To return to the root directory from anywhere in the file structure: Type `cd /root` and press ENTER.
  - Enter the following command: `sh ./install.bin`
- Key names and key strokes are indicated with UPPERCASE:
  - Press CTRL+P.
  - Press the UP ARROW key.
- Text in *italics* indicates terms, emphasis, variables, or document titles. For example:
  - For a complete listing of license agreements, refer to the *QLogic Software End User License Agreement*.
  - What are *shortcut keys*?
  - To enter the date type *mm/dd/yyyy* (where *mm* is the month, *dd* is the day, and *yyyy* is the year).
- Topic titles between quotation marks identify related topics either within this manual or in the online help, which is also referred to as *the help system* throughout this document.

- Command line syntax conventions include the following:
  - Plain text indicates items that you must type as shown. For example:
    - `qaucli -pr nic -ei`
  - < > (angle brackets) indicate a variable whose value you must specify. For example:
    - `<serial_number>`
  - [ ] (square brackets) indicate an optional parameter. For example:
    - `[<file_name>]` means specify a file name, or omit it to select the default file name.
  - | (vertical bar) indicates mutually exclusive options; select one option only. For example:
    - `on|off`
    - `1|2|3|4`
  - ... (ellipsis) indicates that the preceding item may be repeated. For example:
    - `x...` means *one* or more instances of `x`.
    - `[x...]` means *zero* or more instances of `x`.
  - ( ) (parentheses) and { } (braces) are used to avoid logical ambiguity. For example:
    - `a|b c` is ambiguous
      `{(a|b) c}` means `a` or `b`, followed by `c`
      `{a|(b c)}` means either `a,` or `b c`

License Agreements

Refer to the *QLogic Software End User License Agreement* for a complete listing of all license agreements affecting the QLogic 3400/8400 Series Adapters.

Technical Support

Customers should contact their authorized maintenance provider for technical support of their QLogic products. QLogic-direct customers may contact QLogic Technical Support; others will be redirected to their authorized maintenance provider. Visit the QLogic support Web site listed in Contact Information for the latest firmware and software updates.

For details about available service plans, or for information about renewing and extending your service, visit the Service Program Web page at http://www.qlogic.com/Support/Pages/ServicePrograms.aspx.

Downloading Updates

The QLogic Web site provides periodic updates to product firmware, software, and documentation.

To download firmware, software, and documentation:

1. Go to the QLogic Downloads and Documentation page: http://driverdownloads.qlogic.com.
2. Type the QLogic model name in the search box.
3. In the search results list, locate and select the firmware, software, or documentation for your product.
4. View the product details Web page to ensure that you have the correct firmware, software, or documentation. For additional information, click **Read Me** and **Release Notes** under Support Files.
5. Click **Download Now**.
6. Save the file to your computer.
7. If you have downloaded firmware, software, drivers, or boot code, follow the installation instructions in the *Readme* file.

Instead of typing a model name in the search box, you can perform a guided search as follows:

1. Click the product type tab: **Adapters**, **Switches**, **Routers**, or **ASICs**.
2. Click the corresponding button to search by model or operating system.
3. Click an item in each selection column to define the search, and then click **Go**.
4. Locate the firmware, software, or document you need, and then click the item's name or icon to download or open the item.

Training

QLogic Global Training maintains a Web site at www.qlogictraining.com offering online and instructor-led training for all QLogic products. In addition, sales and technical professionals may obtain Associate and Specialist-level certifications to qualify for additional benefits from QLogic.

Contact Information

QLogic Technical Support for products under warranty is available during local standard working hours excluding QLogic Observed Holidays. For customers with extended service, consult your plan for available hours. For Support phone numbers, see the Contact Support link at support.qlogic.com.

Support Headquarters	QLogic Corporation 12701 Whitewater Drive Minnetonka, MN 55343 USA
QLogic Web Site	www.qlogic.com
Technical Support Web Site	http://support.qlogic.com
Technical Support E-mail	support@qlogic.com
Technical Training E-mail	training@qlogic.com

Knowledge Database

The QLogic knowledge database is an extensive collection of QLogic product information that you can search for specific solutions. QLogic is constantly adding to the collection of information in the database to provide answers to your most urgent questions. Access the database from the QLogic Support Center: http://support.qlogic.com.

1 Introduction

This chapter describes the audience for this guide and outlines the OpenStack architecture.

Audience

This guide is intended for OpenStack (Juno and Kilo) users who want to configure virtual extensible LAN (VXLAN) with the QLogic 3400/8400 Series Adapters.

What is OpenStack?

The OpenStack project is an open source cloud computing platform that supports all types of cloud environments and works as infrastructure as a service (IAAS).

The technology consists of a series of interrelated projects that control pools of processing, storage, and networking resources throughout a data center, which users manage through a web-based dashboard, command line tools, or representational state transfer (REST) APIs.

2 Example Architecture

This chapter describes the OpenStack 4 node architecture.

OpenStack Nodes

This guide is intended for use with OpenStack 4 node architecture, which consists of the following nodes:

- Controller
- Compute1
- Compute2
- Network

These nodes are described in the following sections.

Controller Node

In the example architecture used in this guide, the Controller node runs the Identity service (Keystone), Image Service (Glance), management portions of the Compute service (Nova Management) and Networking service (Neutron Server/modular layer 2 (ML2) plug-in), networking plug-in, and the dashboard (Horizon). The architecture also includes supporting services such as a database (mysql), message broker (Rabbitmq), and network time protocol (NTP).

Compute Node

There are two compute nodes (Compute1 and Compute2) to check VXLAN configuration. In this architecture, the compute nodes use a kernel-based virtual machine (KVM) as the hypervisor (KVM is the default hypervisor). The compute nodes run the Networking node plug-in (ML2) and layer 2 agent (OVS).

Network Node

The Network node runs the networking plug-in (ML2), layer 2 agent OVS, layer 3 agent, and DHCP agent. This node also handles external (Internet) connectivity for tenant virtual machines (VMs) or instances of Compute nodes.

3 Prerequisites

This chapter describes what you need to use OpenStack with VXLAN configuration using the QLogic 3400/8400 Series Adapters.

You need the following knowledge and equipment:

- One or more QLogic 3400/8400 Series Adapters
- An understanding of OpenStack deployment and experience with Neutron networking with flat mode.
- An understanding of OpenStack configuration on Red Hat® 7, as documented here:

  OpenStack (Juno):
  http://docs.openstack.org/juno/install-guide/install/yum/content/

  OpenStack (Kilo):
  http://docs.openstack.org/kilo/install-guide/install/yum/content/
- The Compute1 and Compute2 nodes must have one or more 3400/8400 Series Adapters present in the system.
- For each VM instance, the Red Hat 7 KVM Guest image tests the VXLAN configurations, which support the 3400/8400 Series Adapters' inbox driver, as described here:
  http://docs.openstack.org/image-guide/content/ch_obtaining_images.html

4 VXLAN Configuration with the QLogic 3400/8400 Series Adapters

This chapter describes how to install packages, configure the system, and configure VXLAN using the QLogic 3400/8400 Series Adapters.

QLogic FastLinQ™ 3400/8400 Series Adapters deliver assists and offloads for VXLAN in OpenStack deployments that enable efficient distribution of network transmit and receive processing for VXLAN traffic across servers with multiple CPU cores. With QLogic VXLAN Overlay acceleration, the adapters provide the ability to distribute workloads efficiently across all processor cores and deliver maximum performance without burdening the host CPU.

QLogic recommends updating the 3400/8400 Series networking driver to the latest available out-of-box driver (see “Downloading Updates” on page viii) for maximum performance in OpenStack VXLAN deployments.

Installing Required Packages and System Configurations

To install the required packages and configure the system:

1. Enable and start the `libvirtd` service on the Compute1 and Compute2 nodes by issuing the following command:

   ```
   # systemctl enable libvirtd && systemctl start libvirtd
   ```

2. Verify that KVM kernel modules are loaded by issuing the following commands:

   ```
   # lsmod | grep kvm
   kvm
   kvm_<arch>
   ```

3. Disable Selinux.

4. In the `/etc/selinux/config` file on the Network, Compute1, and Compute2 nodes, change `SELINUX=enforcing` to `SELINUX=permissive`.
5. Disable the firewall by issuing the following command:

   ```
   # systemctl disable firewalld && systemctl stop firewalld
   ```
6. Make sure the correct version of the `bnx2x` NIC drivers are loaded on the Compute1 and Compute2 nodes.

   You can use the Red Hat 7 inbox driver to check the version.
7. Set the MTU size to 1600 for the VXLAN header.

   This size avoids fragmentation, which can impact throughput.

VXLAN Configurations

For VXLAN configuration using OVS and the ML2 plug-in, make the configuration changes described in the following sections for all nodes (Controller, Network, Compute1, and Compute2).

Controller Node Configuration

To configure the Controller node:

1. In the Controller node, edit the `/etc/neutron/plugins/ml2/ml2_conf.ini` file as follows:

```
[ml2]
tenant_network_types = flat, vxlan
type_drivers = vxlan
mechanism_drivers = openvswitch.

[ml2_type_vxlan]
…
vni_ranges = 65537:69999

[securitygroup]
enable_security_group = True
firewall_driver =
neutron.agent.linux.iptables_firewall.OVSHybridIptablesFirewa
llDriver
enable_ipset = True
```

2. Restart the Neutron service and OpenStack Nova service by issuing the following commands:

```
# systemctl restart neutron-server.service
# systemctl restart openstack-nova-api.service
```

Network Node Configuration

To configure the Network node:

1. In the Network node, edit the `/etc/neutron/plugins/ml2/ml2_conf.ini` file as follows:

```
[ml2]
tenant_network_types = flat, vxlan
type_drivers = vxlan
mechanism_drivers = openvswitch

[ml2_type_flat]
flat_networks = external

[ml2_type_vxlan]
…
vni_ranges = 65537:69999


[securitygroup]
enable_security_group = True
firewall_driver =
neutron.agent.linux.iptables_firewall.OVSHybridIptablesFirewa
llDriver
enable_ipset = True

[ovs]
local_ip = <INSTANCE_TUNNEL_INTERFACE_IP_ADDRESS>
tunnel_type = vxlan
tunnel_bridge = br-tun
integration_bridge = br-int
tunnel_id_ranges = 65537:69999
tenant_network_type = vxlan
enable_tunneling = true

[agent]
```

```
root_helper = sudo neutron-rootwrap
/etc/neutron/rootwrap.conf
tunnel_types = vxlan
vxlan_udp_port = 4789
l2_population = False
```

2. Restart the Open vSwitch service and other Neutron agents services by issuing the following commands:

```
# service openvswitch-switch restart
# service neutron-plugin-openvswitch-agent restart
# service neutron-l3-agent restart
# service neutron-dhcp-agent restart
# service neutron-metadata-agent restart
```

Compute Node Configuration (Compute1 and Compute2)

To configure the Compute1 and Compute2 nodes:

1. In the Compute1 and Compute2 nodes, edit the `/etc/neutron//ml2/ml2_conf.ini` file as follows:

```
[ml2]
tenant_network_types = flat, vxlan
type_drivers = vxlan
mechanism_drivers = openvswitch


[ml2_type_vxlan]
…
vni_ranges = 65537:69999


[securitygroup]
enable_security_group = True
firewall_driver =
neutron.agent.linux.iptables_firewall.OVSHybridIptablesFirewa
llDriver
enable_ipset = True

[ovs]
local_ip = <INSTANCE_TUNNEL_INTERFACE_IP_ADDRESS> //
different ip for compute node1 and computenode2
tunnel_type = vxlan
```

```
tunnel_bridge = br-tun
integration_bridge = br-int
tunnel_id_ranges = 65537:69999
tenant_network_type = vxlan
enable_tunneling = true
[agent]
root_helper = sudo neutron-rootwrap
/etc/neutron/rootwrap.conf
tunnel_types = vxlan
vxlan_udp_port = 4789
l2_population = False
```

2. Restart Open vSwitch, Neutron agent, and Nova compute services by issuing the following commands:

```
# service openvswitch-switch restart
# service nova-compute restart
# service neutron-plugin-openvswitch-agent restart
```

Verify VXLAN Ports on Network and Compute Nodes

To verify VXLAN ports on the Network, Compute1, and Compute2 nodes:

1. Ensure that the Network node tunneling bridge output and associated VXLAN ports are configured correctly by issuing the following command:

```
# ovs-vsctl show
```

Following is a sample output.

```
Bridge br-tun
        Port br-tun
            Interface br-tun
                type: internal
        Port patch-int
            Interface patch-int
                type: patch
                options: {peer=patch-tun}
        Port "vxlan-0a00015c"
            Interface "vxlan-0a00015c"
                type: vxlan
                options: {df_default="true", in_key=flow,
local_ip="10.0.1.81", out_key=flow, remote_ip="10.0.1.92"}
        Port "vxlan-0a00015b"
```

```
            Interface "vxlan-0a00015b"
                type: vxlan
                options: {df_default="true", in_key=flow,
local_ip="10.0.1.81", out_key=flow, remote_ip="10.0.1.91"}
```

2. Ensure that the Compute1 and Compute2 nodes' tunneling bridge output and associated VXLAN ports are configured correctly by issuing the following command:

   ovs-vsctl show

   Following is a sample output.

```
Bridge br-tun
        Port "vxlan-0a00015c"
            Interface "vxlan-0a00015c"
                type: vxlan
options: {df_default="true", in_key=flow,
local_ip="10.0.1.91", out_key=flow, remote_ip="10.0.1.92"}
        Port "vxlan-0a000151"
            Interface "vxlan-0a000151"
                type: vxlan
options: {df_default="true", in_key=flow,
local_ip="10.0.1.91", out_key=flow, remote_ip="10.0.1.81"}
        Port br-tun
            Interface br-tun
                type: internal
        Port patch-int
            Interface patch-int
                type: patch
                options: {peer=patch-tun}
    ovs_version: "2.1.3"
```

5 Configuration Deployment with VXLAN

This chapter describes how to deploy OpenStack with VXLAN, as follows:

- “Creating a Tenant Network ”
- “Creating the vxlan_subnet Subnet ” on page 11
- “Creating the VM with the VXLAN Network” on page 11

Creating a Tenant Network

To create a tenant network:

1. Create a tenant network with the VXLAN network type by issuing the following command:

   ```
   # neutron net-create demo-net --provider:network_type=vxlan
   ```

2. Issue the following command to view the Neutron net list:

   ```
   # neutron net-list
   ```

   Following is a sample output.

```
[root@network1 plugins]# neutron net-list
+--------------------------------------+----------+------------------------------------------------------+
| id                                   | name     | subnets                                              |
+--------------------------------------+----------+------------------------------------------------------+
| c1ae0470-49f7-4ba7-ba17-5af90141eeb0 | demo-net | a969519f-9ded-4875-bd05-25d755672be0 13.0.0.0/24     |
| bf4abf7e-7ed9-44d0-8e9b-71c2b2486efb | ext-net  | 22b758c1-a10c-43c2-ae98-9254f11adb15 172.28.0.0/20 |
+--------------------------------------+----------+------------------------------------------------------+
```

The values in the `name` column are described in the following paragraphs.

`demo-net`	The VXLAN network for VM access
`ext-net`	An external network that provides Internet access for instances using NAT/floating IP address and a qualified security group

Creating the vxlan_subnet Subnet

To create the vxlan_subnet:

1. Create a subnet network for VXLAN by issuing the following command:

```
# neutron subnet-create demo-net --name vxlan_subnet
--gateway 13.0.0.1 13.0.0.0/24
```

2. Issue the following command to view the Neutron subnet list:

```
# neutron subnet-list
```

A sample output follows.

```
[root@network1 plugins]# neutron subnet-list
+--------------------------------------+-------------+--------------+--------------------------------------------------+
| id                                   | name        | cidr         | allocation_pools                                 |
+--------------------------------------+-------------+--------------+--------------------------------------------------+
| a969519f-9ded-4875-bd05-25d755672be0 | vxlan_subnet | 13.0.0.0/24   | {"start": "13.0.0.2", "end": "13.0.0.254"}       |
| 22b758c1-a10c-43c2-ae98-9254f11adb15 | ext-subnet   | 172.28.0.0/20 | {"start": "172.28.11.231", "end": "172.28.11.250"} |
+--------------------------------------+-------------+--------------+--------------------------------------------------+
```

Creating the VM with the VXLAN Network

The following instructions create a VM with the VXLAN network on two different Compute hosts. A VM named P1 is created on one host using the Compute1 node; a VM named P2 is created on a different host using the Compute2 node.

To create and verify the VMs:

1. Create two VMs with VXLAN by issuing the following commands:

```
# nova boot --flavor m1.small --image <KVM guest OS Image_ID>
--nic net-id=<demo net ID> --hint force_hosts=compute1
--security-group default  P1


# nova boot --flavor m1.small --image <KVM guest OS Image_ID>
--nic net-id=<demo net ID> --hint force_hosts=compute2
--security-group default  P2
```

Following is a sample output for the P2 VM on the Compute2 node.

```
[root@network1 plugins]# nova boot --flavor m1.small --image 6292c70f-f327-4f82-be36-547afb9febcb  --nic net-id=c1ae0470-49f7-4ba7-ba17-5af90141eeb0 --hint force_hosts=
10.0.0.92 --security-group default P2
+--------------------------------------+--------------------------------------+
| Property                             | Value                                |
+--------------------------------------+--------------------------------------+
| OS-DCF:diskConfig                    | MANUAL                               |
| OS-EXT-AZ:availability_zone          | nova                                 |
| OS-EXT-SRV-ATTR:host                 | -                                    |
| OS-EXT-SRV-ATTR:hypervisor_hostname  | -                                    |
| OS-EXT-SRV-ATTR:instance_name        | instance-00000092                    |
| OS-EXT-STS:power_state               | 0                                    |
| OS-EXT-STS:task_state                | scheduling                           |
| OS-EXT-STS:vm_state                  | building                             |
| OS-SRV-USG:launched_at               | -                                    |
| OS-SRV-USG:terminated_at             | -                                    |
| accessIPv4                           |                                      |
| accessIPv6                           |                                      |
| adminPass                            | zYhp4RKEaF3N                         |
| config_drive                         |                                      |
| created                              | 2015-03-26T08:35:20Z                 |
| flavor                               | m1.small (2)                         |
| hostId                               |                                      |
| id                                   | bf47b578-e3af-4a92-8daa-f76d5be46446 |
| image                                | rhx (6292c70f-f327-4f82-be36-547afb9febcb) |
| key_name                             | -                                    |
| metadata                             | {}                                   |
| name                                 | P2                                   |
| os-extended-volumes:volumes_attached | []                                   |
| progress                             | 0                                    |
| security_groups                      | default                              |
| status                               | BUILD                                |
| tenant_id                            | 28b7418d330c4fb08d753c84f10c4eec     |
| updated                              | 2015-03-26T08:35:20Z                 |
| user_id                              | 681a6563531b488a99bad094976a89f8     |
+--------------------------------------+--------------------------------------+
```

2. To ensure that the VM is up and running, issue the following command:

   `# nova list`

   Following is a sample output.

```
[root@network1 plugins]# nova list
+--------------------------------------+------+--------+------------+-------------+----------------------------------+
| ID                                   | Name | Status | Task State | Power State | Networks                         |
+--------------------------------------+------+--------+------------+-------------+----------------------------------+
| eb05450e-582f-474e-ad86-ae1a61df4275 | P1   | ACTIVE | -          | Running     | demo-net=13.0.0.5, 172.28.11.235 |
| bf47b578-e3af-4a92-8daa-f76d5be46446 | P2   | ACTIVE | -          | Running     | demo-net=13.0.0.6, 172.28.11.236 |
+--------------------------------------+------+--------+------------+-------------+----------------------------------+
```

6 Testing

This chapter contains a testing example and describes how to check the VMS on the host and compute host level.

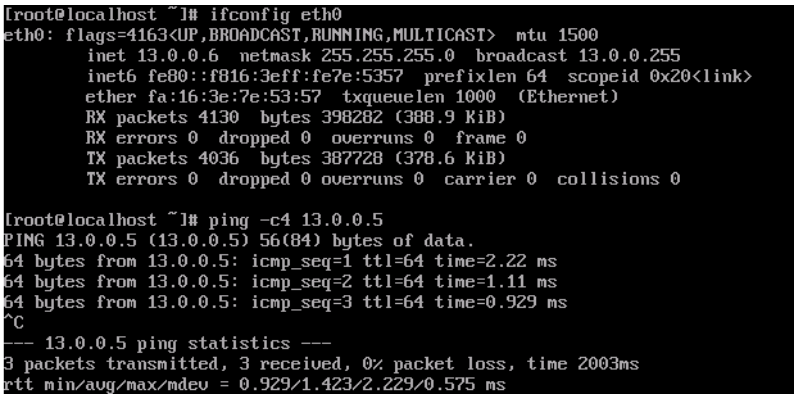
Example Test

In this example, two VMs are created. A VM named P1 is created on one host using the Compute1 node; a VM named P2 is created on a different host using the Compute2 node.

To ensure that the VMs are up and running:

1. Log in to the system using either a floating IP address or the Horizon dashboard Instance log in console.
2. Ping between the two VMs with an IP demo-net assigned IP address.

   Following is a sample output.



Testing VMs

To test VMs on the host level, issue the following command:

```
ovs-ofctl dump-ports br-tun
```

The command output shows that packets and bytes counts increase as the ping operations continue between the VMs.

To test VMs on the compute host level, issue the following command at the interface level:

```
# tcpdump  -i  ens5f0  -c 1000 –w /root/vxlan.pcap
```

This command captures a packet trace, runs WIRESHARK®, and decodes packets as VXLAN packets.

7 Known Issue

This chapter describes a known issue with OVS and OpenStack.

Sometimes OVS version 2.1.2-2 generates a segfault with the OpenStack (Kilo) release.

QLogic has installed latest version OVS-2.3.1 from the following location: http://openvswitch.org/releases/openvswitch-2.3.1.tar.gz

Following is an example of how to compile OVS.

To compile OVS on Red Hat 7.9:

1. Copy the distribution tar ball (`Openvswitch-2.3.1.tar.gz`) to the rpm source directory (`/root/rpmbuild/SOURCES`).
2. Install the following build prerequisites before compiling Open vSwitch:
   - ❑ `gcc`
   - ❑ `make`
   - ❑ `python-devel`
   - ❑ `openssl-devel`
   - ❑ `kernel-devel`
   - ❑ `graphviz`
   - ❑ `kernel-debug-devel`
   - ❑ `autoconf`
   - ❑ `automake`
   - ❑ `rpm-build`
   - ❑ `redhat-rpm-config`
   - ❑ `libtool`
3. Extract the spec file from `Openvswitch-2.3.1.tar.gz` (`Openvswitch-2.3.1/rhel/openvswitch.spec`) to the `/root/rpmbuild/SPECS/` folder.
4. Edit the `openvswitch.spec` file and remove the `Openvswitch-kmod` line from the Requires section.

5. Issue the following command:

   rpmbuild -bb /root/rpmbuild/SPECS/openvswitch.spec

   This command creates an Open vSwitch 64-bit rpm in the `/root/rpmbuild/RPMS/x86_64` location.

6. Issue the following command to install rpm:

   rpm -ivh openvswitch-2.3.1-1.x86_64.rpm

Corporate Headquarters QLogic Corporation 26650 Aliso Viejo Parkway Aliso Viejo, CA 92656 949.389.6000 **www.qlogic.com**

International Offices UK | Ireland | Germany | France | India | Japan | China | Hong Kong | Singapore | Taiwan | Israel