

# Mellanox OpenStack Solution Reference

## Architecture

Rev 1.3 January 2014

www.mellanox.com

Mellanox Technologies

#### NOTE:

THIS HARDWARE, SOFTWARE OR TEST SUITE PRODUCT ("PRODUCT(S)") AND ITS RELATED DOCUMENTATION ARE PROVIDED BY MELLANOX TECHNOLOGIES "AS-IS" WITH ALL FAULTS OF ANY KIND AND SOLELY FOR THE PURPOSE OF AIDING THE CUSTOMER IN TESTING APPLICATIONS THAT USE THE PRODUCTS IN DESIGNATED SOLUTIONS. THE CUSTOMER'S MANUFACTURING TEST ENVIRONMENT HAS NOT MET THE STANDARDS SET BY MELLANOX TECHNOLOGIES TO FULLY QUALIFY THE PRODUCT(S) AND/OR THE SYSTEM USING IT. THEREFORE, MELLANOX TECHNOLOGIES CANNOT AND DOES NOT GUARANTEE OR WARRANT THAT THE PRODUCTS WILL OPERATE WITH THE HIGHEST QUALITY. ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NON INFRINGEMENT ARE DISCLAIMED. IN NO EVENT SHALL MELLANOX BE LIABLE TO CUSTOMER OR ANY THIRD PARTIES FOR ANY DIRECT, INDIRECT, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES OF ANY KIND (INCLUDING, BUT NOT LIMITED TO, PAYMENT FOR PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY FROM THE USE OF THE PRODUCT(S) AND RELATED DOCUMENTATION EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.



2

Mellanox Technologies 350 Oakmead Parkway Suite 100 Sunnyvale, CA 94085 U.S.A. www.mellanox.com Tel: (408) 970-3400 Fax: (408) 970-3403 Mellanox Technologies, Ltd. Beit Mellanox PO Box 586 Yokneam 20692 Israel www.mellanox.com Tel: +972 (0)74 723 7200 Fax: +972 (0)4 959 3245

© Copyright 2014. Mellanox Technologies. All Rights Reserved.

Mellanox®, Mellanox logo, BridgeX®, ConnectX®, CORE-Direct®, InfiniBridge®, InfiniHost®, InfiniScale®, MLNX-OS®, PhyX®, SwitchX®, UFM®, Virtual Protocol Interconnect® and Voltaire® are registered trademarks of Mellanox Technologies, Ltd.

Connect-IB<sup>TM</sup>, FabricIT<sup>TM</sup>, Mellanox Open Ethernet<sup>TM</sup>, Mellanox Virtual Modular Switch<sup>TM</sup>, MetroX<sup>TM</sup>, MetroDX<sup>TM</sup>, ScalableHPC<sup>TM</sup>, Unbreakable-Link<sup>TM</sup> are trademarks of Mellanox Technologies, Ltd.

All other trademarks are property of their respective owners

## Contents

1	Solut	ion Ove	rview7		
2	Acce	lerating	Storage9		
3	Netw	ork Virtu	alization on Mellanox Adapters11		
	3.1	Perform	ance Measurements 12		
	3.2	Seamle	ss Integration		
4	Setup	o and Ins	stallation14		
	4.1	Hardwa	re Requirements		
	4.3	Prerequ	iisites		
	4.4	OpenSt	ack Software Installation15		
	4.5	Trouble	shooting16		
5	Settir	ng Up th	e Network 17		
		5.1.1	Creating a Network 17		
		5.1.2	Creating a Para-Virtualized vNIC Instance		
		5.1.3	Creating an SR-IOV Instance		
		5.1.4	Creating a Volume		
		5.1.5	Attaching a Volume23		
	5.2	Verifica	5.1.2Creating a Para-Virtualized vNIC Instance185.1.3Creating an SR-IOV Instance205.1.4Creating a Volume225.1.5Attaching a Volume23Verification Examples235.2.1Instances Overview235.2.2Connectivity Check23		
		5.2.1	Instances Overview		
		5.2.2	Connectivity Check		
		5.2.3	Volume Check		

## List of Figures

Figure 1: Mellanox OpenStack Architecture	8
Figure 2: OpenStack Based IaaS Cloud POD Deployment Example	9
Figure 3: RDMA Acceleration	. 10
Figure 4: eSwitch Architecture	. 11
Figure 5: Latency Comparison	. 12
Figure 6: Network Virtualization	. 13
Figure 7: Mellanox MCX314A-BCBT, ConnectX-3 40GbE Adapter	. 14
Figure 8: Mellanox SX1036, 36x40GbE	. 15
Figure 9: Mellanox 40GbE, QSFP Copper Cable	. 15
Figure 10: OpenStack Dashboard Instances	. 18
Figure 11: OpenStack Dashboard, Launch Instance	. 19
Figure 12: OpenStack Dashboard, Launch Interface – Select Network	. 19
Figure 13: OpenStack Dashboard, Volumes	. 22
Figure 14: OpenStack Dashboard, Create Volumes	. 22
Figure 15: OpenStack Dashboard, Volumes	. 22
Figure 16: OpenStack Dashboard, Manage Volume Attachments	. 23
Figure 17: VM Overview	. 23
Figure 18: Remote Console Connectivity	. 24
Figure 19: OpenStack Dashboard, Volumes	. 24
Figure 20: OpenStack Dashboard, Console	. 24

## Preface

## About this Document

This reference design presents the value of using Mellanox interconnect products and describes how to integrate the OpenStack solution (Havana release or later) with the end-to-end Mellanox interconnect products.

## Audience

This reference design is intended for server and network administrators.

The reader must have experience with the basic OpenStack framework and installation.

## References

For additional information, see the following documents:

### Table 1: Related Documentation

Reference	Location			
Mellanox OFED User Manual	www.mellanox.com > Products > Adapter IB/VPI SW > Linux SW/Drivers			
	http://www.mellanox.com/content/pages.php?pg=products_dy n&product_family=26&menu_section=34			
Mellanox software source packages	https://github.com/mellanox-openstack			
OpenStack Website	www.openstack.org			
Mellanox OpenStack wiki page	https://wiki.openstack.org/wiki/Mellanox-OpenStack			
Mellanox Ethernet Switch Systems User Manual	http://www.mellanox.com/related-docs/user_manuals/SX10X X User Manual.pdf			
Mellanox Ethernet adapter cards	http://www.mellanox.com/page/ethernet_cards_overview			
Solutions space on Mellanox community	http://community.mellanox.com/community/support/solutions			
OpenStack RPM package	http://community.mellanox.com/docs/DOC-1187			
Mellanox eSwitchd Installation for OpenFlow and OpenStack	http://community.mellanox.com/docs/DOC-1126			

Reference	Location			
Troubleshooting	http://community.mellanox.com/docs/DOC-1127			
Mellanox OFED Driver Installation and Configuration for SR-IOV	http://community.mellanox.com/docs/DOC-1317			

## **Revision History**

Revision	Date	Changes
1.3	Jan. 2014	Removed OpenFlow sections. Added related topics for OpenStack Havana release.
1.2	Sep. 2013	Minor editing
1.1	June 2013	Added OpenFlow feature
1.0	May 2013	Initial revision

7

## **1** Solution Overview

Deploying and maintaining a private or public cloud is a complex task, with various vendors developing tools to address the different aspects of the cloud infrastructure, management, automation, and security. These tools tend to be expensive and create integration challenges for customers when they combine parts from different vendors. Traditional offerings suggest deploying multiple network and storage adapters to run management, storage, services, and tenant networks. These also require multiple switches, cabling, and management infrastructure, which increases both up front and maintenance costs.

Other, more advanced offerings provide a unified adapter and first level ToR switch, but still run multiple and independent core fabrics. Such offerings tend to suffer from low throughput because they do not provide the aggregate capacity required at the edge or in the core; and because they deliver poor application performance due to network congestion and lack of proper traffic isolation.

Several open source "cloud operating system" initiatives have been introduced to the market, but none has gained sufficient momentum to succeed. Recently OpenStack has managed to establish itself as the leading open source cloud operating system, with wide support from major system vendors, OS vendors, and service providers. OpenStack allows central management and provisioning of compute, networking, and storage resources, with integration and adaptation layers allowing vendors and/or users to provide their own plug-ins and enhancements.

Mellanox Technologies offers seamless integration between its products and OpenStack layers and provides unique functionality that includes application and storage acceleration, network provisioning, automation, hardware-based security, and isolation. Furthermore, using Mellanox interconnect products allows cloud providers to save significant capital and operational expenses through network and I/O consolidation and by increasing the number of virtual machines (VMs) per server.

Mellanox provides a variety of network interface cards (NICs) supporting one or two ports of 10GbE, 40GbE, or 56Gb/s InfiniBand. These adapters simultaneously run management, network, storage, messaging, and clustering traffic. Furthermore, these adapters create virtual domains within the network that deliver hardware-based isolation and prevent cross-domain traffic interference.

In addition, Mellanox Virtual Protocol Interconnect (VPI) switches deliver the industry's most cost-effective and highest capacity switches (supporting up to 36 ports of 56Gb/s). When deploying large-scale, high-density infrastructures, leveraging Mellanox converged network VPI solutions translates into fewer switching elements, far fewer optical cables, and simpler network design.

Mellanox plugins are inbox for Havana release. The Havana release includes out of the box support for InfiniBand and Ethernet Mellanox components for Nova, Cinder and Neutron.

#### Mellanox integration with OpenStack provides the following benefits:

- Cost-effective and scalable infrastructure that consolidates the network and storage to a highly efficient flat fabric, increases the VM density, commoditizes the storage infrastructure, and linearly scales to thousands of nodes
- Delivers the best application performance with hardware-based acceleration for messaging, network traffic, and storage
- Easy to manage via standard APIs. Native integration with OpenStack Neutron (network) and Cinder (storage) provisioning APIs
- Provides tenant and application security/isolation, end-to-end hardware-based traffic isolation, and security filtering
- Mellanox designed its end-to end OpenStack cloud solution to offer seamless integration between its products and OpenStack services.
- By using Mellanox 10/40GbE and FDR 56Gb/s adapters and switches with OpenStack Havana release, customers can gain significant improvement in block storage access performance with Cinder. In addition, customers can deploy an embedded virtual switch to run virtual machine traffic with bare-metal performance, provide hardened security and QoS, all with simple integration.
- Mellanox has partnered with the leading OpenStack distributions to allow customers to confidently deploy an OpenStack cloud, with proven interoperability and integrated support. For more information click <u>here</u>.



#### Figure 1: Mellanox OpenStack Architecture

## 2 Accelerating Storage

Data centers rely on communication between compute and storage nodes, as compute servers read and write data from the storage servers constantly. In order to maximize the server's application performance, communication between the compute and storage nodes must have the lowest possible latency, highest possible bandwidth, and lowest CPU utilization.



Figure 2: OpenStack Based IaaS Cloud POD Deployment Example

**Switching Fabric** 

Storage applications rely on iSCSI over TCP communications protocol stack continuously interrupt the processor in order to perform basic data movement tasks (packet sequence and reliability tests, re-ordering, acknowledgements, block level translations, memory buffer copying, etc). This causes data center applications that rely heavily on storage communication to suffer from reduced CPU efficiency, as the processor is busy sending data to and from the storage servers rather than performing application processing. The data path for applications and system processes must wait in line with protocols such as TCP, UDP, NFS, and iSCSI for their turn using the CPU. This not only slows down the network, but also uses system resources that could otherwise have been used for executing applications faster.

Mellanox OpenStack solution extends the Cinder project by adding iSCSI running over RDMA (iSER). Leveraging RDMA Mellanox OpenStack delivers 6X better data throughput (for example, increasing from 1GB/s to 5GB/s) and while simultaneously reducing CPU utilization by up to 80% (see Figure 3).

Mellanox ConnectX®-3 adapters bypass the operating system and CPU by using RDMA, allowing much more efficient data movement. iSER capabilities are used to accelerate hypervisor traffic, including storage access, VM migration, and data and VM replication. The use of RDMA shifts data movement processing to the Mellanox ConnectX-3 hardware, which provides zero-copy message transfers for SCSI packets to the application, producing significantly faster performance, lower network latency, lower access time, and lower CPU overhead. iSER can provide 6X faster performance than traditional TCP/IP based iSCSI. The

iSER protocol unifies the software development efforts of both Ethernet and InfiniBand communities, and reduces the number of storage protocols a user must learn and maintain.

RDMA bypass allows the application data path to effectively skip to the front of the line. Data is provided directly to the application immediately upon receipt without being subject to various delays due to CPU load-dependent software queues. This has three effects:

- There is no waiting, which means that the latency of transactions is incredibly low.
- Because there is no contention for resources, the latency is deterministic, which is essential for offering end users a guaranteed SLA.
- Bypassing the OS, using RDMA results in significant savings in CPU cycles. With a more efficient system in place, those saved CPU cycles can be used to accelerate application performance.

In the following diagram, it is clear that by performing hardware offload of the data transfers using the iSER protocol, the full capacity of the link is utilized to the maximum of the PCIe limit.

To summarize, network performance is a significant element in the overall delivery of data center services and benefits from high speed interconnects.Unfortunately the high CPU overhead associated with traditional storage adapters prevents systems from taking full advantage of these high speed interconnects. The iSER protocol uses RDMA to shift data movement tasks to the network adapter and thus frees up CPU cycles that would otherwise be consumed executing traditional TCP and iSCSI protocols. Hence, using RDMA-based fast interconnects significantly increases data center application performance levels.



#### Figure 3: RDMA Acceleration

## **3 Network Virtualization on Mellanox Adapters**

Single Root IO Virtualization (SR-IOV) allows a single physical PCIe device to present itself as multiple devices on the PCIe bus. Mellanox ConnectX®-3 adapters are capable of exposing up to 127 virtual instances called Virtual Functions (VFs). These virtual functions can then be provisioned separately. Each VF can be viewed as an additional device associated with the Physical Function. It shares the same resources with the Physical Function, and its number of ports equals those of the Physical Function.

SR-IOV is commonly used in conjunction with an SR-IOV enabled hypervisor to provide virtual machines with direct hardware access to network resources, thereby improving performance.

Mellanox ConnectX-3 adapters equipped with onboard embedded switch (eSwitch) are capable of performing layer-2 switching for the different VMs running on the server. Using the eSwitch will gain even higher performance levels and in addition improve security, and isolation.



#### Figure 4: eSwitch Architecture

Physical Ports (pPort)

### eSwitch main capabilities and characteristics:

- Virtual switching: creating multiple logical virtualized networks. The eSwitch offload engines handle all networking operations up to the VM, thereby dramatically reducing software overheads and costs.
- Performance: The switching is handled in hardware, as opposed to other applications that use a software-based switch. This enhances performance by reducing CPU overhead.

- Security: The eSwitch enables network isolation (using VLANs) and anti-MAC spoofing.
- Monitoring: Port counters are supported.

### 3.1 Performance Measurements

Many data center applications benefits from low latency network communication while others require deterministic latency. Using regular TCP connectivity between VMs can create high latency and unpredictable delay behavior.

<u>Figure 5</u> shows the dramatic difference (20X improvement) delivered by SR-IOV connectivity running RDMA compared to para-virtualized vNIC running a TCP stream.

Using the direct connection of the SR-IOV and the ConnectX-3 hardware eliminates the software processing that adds an unpredictable delay to packet data movement. The result is a consistently low latency that allows application software to rely on deterministic packet transfer times.



#### Figure 5: Latency Comparison

## 3.2 Seamless Integration

The eSwitch configuration is transparent to the OpenStack controller administrator. The installed eSwitch daemon on the server is responsible for hiding the low-level configuration. The administrator will use the standard OpenStack dashboard APIs REST interface for the fabric management.

The Neutron agent configures the eSwitch in the adapter card.



#### Figure 6: Network Virtualization

## 4 Setup and Installation

The OpenStack environment should be installed according to the OpenStack documentation package.

For OpenStack Havana release the following installation changes should be applied:

- A Neutron server should be installed with the Mellanox Neutron plugin.
- Mellanox Neutron agent, eSwitch daemon, and Nova VIF driver should be installed on the compute notes.

For OpenStack Grizzly release the following installation changes should be applied:

- A Neutron server should be installed with the Mellanox Neutron plugin.
- A Cinder patch should be applied to the storage servers (for iSER support).
- Mellanox Neutron agent, eSwitch daemon, and Nova VIF driver should be installed on the compute notes.

## 4.1 Hardware Requirements

- Mellanox ConnectX®-3 adapter cards
- 10GbE or 40GbE Ethernet switches
- Cables required for the ConnectX-3 card (typically using SFP+ connectors for 10GbE or QSFP connectors for 40GbE)
- Server nodes complying with OpenStack requirements
- Compute nodes with SR-IOV capability (BIOS and OS support)

In terms of adapters, cables, and switches, many variations are possible. Visit <u>www.mellanox.com</u> for more information.

#### Figure 7: Mellanox MCX314A-BCBT, ConnectX-3 40GbE Adapter



#### Figure 8: Mellanox SX1036, 36x40GbE



Figure 9: Mellanox 40GbE, QSFP Copper Cable



## 4.2 Software Requirements

- Supported OS
- RHEL 6.4 or higher
- Mellanox OFED 2.0.3 (SR-IOV support) or higher.
- KVM hypervisor complying with OpenStack requirements

### 4.3 **Prerequisites**

- 1. Hardware is set up.
  - To reduce the number of ports in the network, two different subnets can be mapped to the same physical interface on two different VLANs.
- 2. Mellanox OFED 2.0.3 (SR-IOV enabled) is installed on each of the network adapters.
  - For Mellanox OFED installation refer to Mellanox OFED User Manual (Installation chapter). http://www.mellanox.com/page/products\_dyn?product\_family=26
    - http://www.menanox.com/page/products\_dyn?product\_tammy=20
  - Visit Mellanox Community for verification options and adaptation. <u>http://community.mellanox.com/docs/DOC-1317</u>
- 3. The OpenStack packages are installed on all network elements.
- 4. EPEL repository is enabled. (<u>http://fedoraproject.org/wiki/EPEL</u>).

## 4.4 OpenStack Software Installation

For Mellanox OpenStack installation visit the Mellanox OpenStack wiki pages

at https://wiki.openstack.org/wiki/Mellanox-OpenStack.

Rev 1.3

## 4.5 Troubleshooting

Troubleshooting actions for OpenStack installation with Mellanox plugins can be found at <u>http://community.mellanox.com/docs/DOC-1127</u>.

## 5 Setting Up the Network

## 5.1 Configuration Examples

Once installation is completed, the network must be set up.

Setting up a network consists of the following steps:

- 1. Creating a network.
- 2. Creating a VM instance.

Two types of instances can be created:

- i. Para-virtualized vNIC.
- ii. SR-IOV direct path connection.
- 3. Creating disk volume.
- 4. Binding the disk volume to the instance created.

### 5.1.1 Creating a Network

Use the commands neutron net-create and neutron subnet-create to create a new network and a subnet ("net-example" in the example).

<pre>\$neutron net-create net-example Created a new network:</pre>							
Field	Value						
<pre>  admin_state_up   id   name   provider:network_type   provider:physical_network   provider:segmentation_id   shared   status   subnets</pre>	True                 16b790d6-4f5a-4739-a190-7598f331b696                 net-example                 vlan                 default                 4                 False                 ACTIVE						
tenant_id +	679545ff6c1e4401adcafa0857aefe2e   ++						

\$neutron subnet-create net-example 192.168.199.0/24 Created a new subnet:								
+	Value							
allocation_pools	' {"start": "192.168.199.2", "end": "192.168.199.254"}							
cidr	192.168.199.0/24							
dns_nameservers								
enable_dhcp	True							
gateway_ip	192.168.199.1							
host_routes								
id	3c9ff1ae-218d-4020-b065-a2991d23bb72							
ip_version	4							
name								
network_id	16b790d6-4f5a-4739-a190-7598f331b696							
tenant_id	679545ff6c1e4401adcafa0857aefe2e							

### 5.1.2 Creating a Para-Virtualized vNIC Instance

- 1. Using the OpenStack Dashboard, launch an VM instance using the Launch Instance button.
- 2. Insert all the required parameters and click Launch.

This operation creates a macvtap interface on top of a Virtual Function (VF).

#### Figure 10: OpenStack Dashboard Instances

	Instances						Logg	ged in as: admin Settings Sign C
openstack	Instances							+ Launch Instance
DASHBOARD	Instance Name	IP Address	Size	Keypair	Status	Task	Power State	Actions Launch Instan
Project Admin				No items to disp	vlay.			
	Displaying 0 items							
CURRENT PROJECT								
Aanage Compute								
Overview								
Instances								
Volumes								

Launch Instance		×
Details Access & Security Network	ing Volume Options Pos	st-Creation
Instance Source Image Image	in relation to the project	the resources used by this project
rh6.3	Flavor Details Name	m1.tiny
Instance Name	VCPUs	1
vm1	Root Disk	0 GB
Flavor	Ephemeral Disk	0 GB
m1.tiny	▼ Total Disk	0 GB
Instance Count	RAM	512 MB
1	Project Quotas Number of Instances	
	Total RAM (0 MB)	51,200 MB Available
		Cancel

Figure 11: OpenStack Dashboard, Launch Instance

3. Select the desired network for the vNIC ("net3" in the example).

Figure 12: OpenStack Dashboard, Launch Interface – Select Network



Rev 1.3

### 5.1.3 Creating an SR-IOV Instance

1. Use the command neutron port-create for the selected network ('net3' in the example) to create a port with 'vnic\_type=hostdev'.

```
$neutron port-create net-example --binding:profile type=dict vnic type=hostdev
Created a new port:
+------|
l Field
               | Value
                                                        +-----
                                                        - 1
| admin state up | True
| binding:capabilities | {"port_filter": false}
| binding:host id
               | binding:profile
               | {"physical_network": "default"}
| binding:vif_type | hostdev
| device id
                | device owner
                | {"subnet_id": "3c9ff1ae-218d-4020-b065-a2991d23bb72",
| fixed ips
                "ip address": "192.168.199.2"}
| id
               | a43d35f3-3870-4ae1-9a9d-d2d341b693d6
| mac address
            | fa:16:3e:67:ad:ef
               | name
              | 16b790d6-4f5a-4739-a190-7598f331b696
| network_id
| status
                | DOWN
                                                         T
               | 679545ff6c1e4401adcafa0857aefe2e
| tenant_id
                                                         1
```

2. Use the command nova boot to launch an instance with the created port attached.

\$nova bootflavor m1.smallimage rh6.4pnic port-id=a43d35f3-3870-4ae1-9a9d-d2d341b693d6 vm3										
++   Property   Value										
	•									
-	scheduling									
	rh6.4p									
	building									
	instance-00000042									
	None									
	m1.small									
id	161da6a9-6508-4e23-9f6f-881383461ab4									
security_groups	<pre>  [{u'name': u'default'}]</pre>									
user_id	b94edf2504c84223b58e254314528902									
OS-DCF:diskConfig	MANUAL									
accessIPv4	I									
accessIPv6	I									
progress	0									
OS-EXT-STS:power_state	0									
OS-EXT-AZ:availability_zone	nova									
config_drive	I									
status	BUILD									
updated	2013-12-19T07:32:42Z									
hostId	I									
OS-EXT-SRV-ATTR:host	None									
OS-SRV-USG:terminated_at	None									
key_name	None									
OS-EXT-SRV-ATTR:hypervisor_hostname	None									
name	vm3									
adminPass	tiTE37tQrNBn									
tenant id	679545ff6c1e4401adcafa0857aefe2e									
	2013-12-19T07:32:41Z									
os-extended-volumes:volumes attached	[]									
metadata	{}									

Rev 1.3

### 5.1.4 Creating a Volume

Create a volume using the Volumes tab on the OpenStack dashboard. Click the Create Volume button.

Figure 13: OpenStack Dashboard, Volumes

	Volumes					Logged	l in as: admin Settings	Sign Out
open <mark>stack</mark>	Volumes							e Volume
DASHBOARD	Name	Description	Size	Status	Туре	Attached To	Actions	
Project Admin				No items to display.				
	Displaying 0 items							
CURRENT PROJECT								

Create Volume	×
Volume Name	Description: Volumes are block devices that can be attached to instances.
Description	Volume Quotas
Additional information here	Total Gigabytes (0 GB) 1,000 GB Available
Type	Number of Volumes (0) 10 Available
Size (GB)	
5	
	Cancel Create Volume

Figure 14: OpenStack Dashboard, Create Volumes

Figure 15: OpenStack Dashboard, Volumes

Volumes						Info: Creating volume "vol1" ×		
Vol	Volumes				+	Create Volume		
	Name	Description	Size	Status	Туре	Attached To		Actions
	vol1		5GB	Available	-			Edit Attachments More *
Display	ying 1 item							

### 5.1.5 Attaching a Volume

Attach a volume to the desired instance.

The device name should be /dev/dv<letter>. E.g. "/dev/vdc"

#### Figure 16: OpenStack Dashboard, Manage Volume Attachments

Mana	age Volume Attachments				×
Att	achments				
	Instance	Device		Actions	
		No item	s to display.		
Display	ying 0 items				
	ch To Instance to Instance		Device Name		
vm1 (	55da5294-af09-4214-a056-6500e45f7d94	4) 💌	/dev/vdc		
				Cancel	Attach Volume

## 5.2 Verification Examples

#### 5.2.1 Instances Overview

Use the OpenStack Dashboard to view all configured instances.

Figure 17: VM Overview

	All	Instand	ces						Logg	ed in as: admin Settings Sign O	
openstack	Inst	Instances interventional intervention									
DASHBOARD		Project	Host	Name	IP Address	Size	Status	Task	Power State	Actions	
Project Admin		admin	xena027.mtr.labs.mlnx	vm3	192.168.203.5	m1.tiny   512MB RAM   1 VCPU   0 Disk	Active	None	Running	Edit Instance More *	
System Panel Oveniew		admin	xena019.mtr.labs.mlnx	vm2	192.168.203.4	m1.tiny   512MB RAM   1 VCPU   0 Disk	Active	None	Running	Edit Instance More *	
Instances		admin	xena027.mtr.labs.minx	vm1	192.168.203.2	m1.tiny   512MB RAM   1 VCPU   0 Disk	Active	None	Running	Edt Instance More *	
Volumes	Display	Daptarying 3 fams									
Elevere											

### 5.2.2 Connectivity Check

There are many options for checking connectivity between instances, one of which is to simply open a remote console and ping the required host.

To launch a remote console for a specific instance, select the Console tab and launch the console.

6	Instance Detail: vm1	Logged in as: admin Settings Sign O
openstack	Overview Log Console	
DASHEOARD	Instance Console	
Project Admin	If console is not responding to keyboard input: click the grey status bar below. <u>Click here to show only console</u>	
System Panel	Connected (unencrypted) to: QEMU (instance-00000026)	Send CtriAltDel
Overview	Red Hat Enterprise Linux Server release 6.3 (Santiago) Mernel 2.6.32−279.el6.x86_64 on an x86_64	
Instances	vm1 login: root Passuord:	
Volumes	Last login: Thu Apr 4 14:25:51 on ttyl (root0vml 71# ping 192.1608.203.4 PING 192.160.203.4 (192.166.203.4) 56(04) bytes of data.	
Flavors	64 bytes from 192.168.203.4: icmp_seq=1 ttl=64 time=1.78 ms 64 bytes from 192.168.203.4: icmp_seq=2 ttl=64 time=0.494 ms	
Images	^c 192.168.203.4 ping statistics 2 packets transmitted, 2 received, 0% packet loss, time 1330ms	
Projects	rtt min/aug/max/mdcu = 8.494/1.141/1.788/8.647 ms Troottvat =71m ping 192.166.283.5 PING 192.166.283.5 (192.168.283.5) 56(84) bytes of data.	
Users	64 bytes from 192.168.203.5: icmp_seq=1 ttl=64 time=0.969 ms 64 bytes from 192.168.203.5: icmp_seq=2 ttl=64 time=0.359 ms	
Routers	°C 192.168.203.5 ping statistics 2 packets transmitted, 2 received, 0% packet loss, time 1390ms	
System Info	rtt min/aug/max/mdev = 0.359/0.664/0.969/0.305 ms [rootQvm] "]#	

Figure 18: Remote Console Connectivity

#### 5.2.3 Volume Check

To verify that the created volume is attached to a specific instance, click the Volumes tab.

Figure 19: OpenStack Dashboard, Volumes

Volumes								Info: Attaching volume vol1 to instance		
Vol	Volumes							m1 on /dev/vdc. + Create Volume   Create Volume		
	Name	Description	Size	Status	Туре	Attached To	1	Actions		
	vol1		5GB	In-Use	-	Attached to vm1 on /dev/vdc		Edit Attachments		
Display	ving 1 item									

In addition, run the fdisk command from the instance console to see the volume details.

Figure 20: OpenStack Dashboard, Console

Instance Detail: vm1	
Overview Log Console	
Instance Console	
If console is not responding to keyboard	nput: click the grey status bar below. <u>Click here to show only console</u>
	Connected (unencrypted) to: QEMU (instance-00000037)
6	[root@192-168-203-3 ~]# fdisk -1
	Disk /dev/vda: 2147 MB, 2147483648 bytes 16 heads, 63 sectors/track, 4161 cylinders Units = cylinders of 1000 * 512 = 516096 bytes Sector size (Iogical/physical): 512 bytes / 512 bytes 1/0 size (minimum/optimal): 512 bytes / 512 bytes Disk identifier: 0x0000487c6 Device Boot Start End Blocks Id System /dev/vda1 3 409 204800 82 Linux swap / Solaris Partition 1 does not end on cylinder boundary. /dev/vda2 * 409 4162 1891328 83 Linux Partition 2 does not end on cylinder boundary.
	Disk /dev/vdb: 5368 MB, 5368709128 bytes 16 heads, 63 sectors/track, 104082 cylinders Units = cylinders of 1008 * 512 = 516096 bytes Sector size (Iogical/physical): 512 bytes > 512 bytes I/O size (minimum/optimal): 512 bytes / 512 bytes Disk identifier: 0x00000000 [root0192-168-203-3 ~]# _