Installation runbook for Tintri Cinder Driver

Partner Name:	Tintri Inc.
Product Name:	Cinder driver
Product Version:	2.1.0.1
MOS Version:	7.0
OpenStack version:	Kilo
Product Type:	Cinder Driver

Contents

Document History
Introduction
    1.1 Objective
    1.2 Target Audience
2 Product Overview
3 Joint Reference Architecture
4 Networking
    4.1 Physical & Logical network topology
5 Installation and Configuration
    5.1 Overview of MOS installation steps
    5.2 MOS Installation in details
    5.3 Creation of OpenStack environment
    5.4 MOS Deployment
    5.5 Tintri Cinder Driver Installation steps
Testing
    6.1 Test tools
    6.2 Test cases
        6.2.1 Target Uses Case(s)
        6.2.2 Deployment modes and configuration options
        6.2.3 Functional testing
        6.2.4 Performance testing
        6.2.5 Negative testing

Document History

Version	Revision Date	Description
0.1	**01-25-2016**	**Initial Version**

1. Introduction

This document is to serve as a detailed Deployment Guide for Tintri Cinder driver. Tintri offers cinder driver storage solution that integrates Tintri Vmstore storage to Openstack cloud environment. This document describes the reference architecture, installation steps for certified MOS+Tintri Cinder driver, limitations and testing procedures.

1.1 Objective

The objective of Mirantis OpenStack certification is to provide Mirantis program partners with a consistent and unified approach for acceptance of their solution into the Mirantis Technology Partner Program.
Certification is designed within the context of Mirantis OpenStack infrastructure, including Mirantis Fuel deployment tool and supported cloud reference architectures

1.2 Target Audience

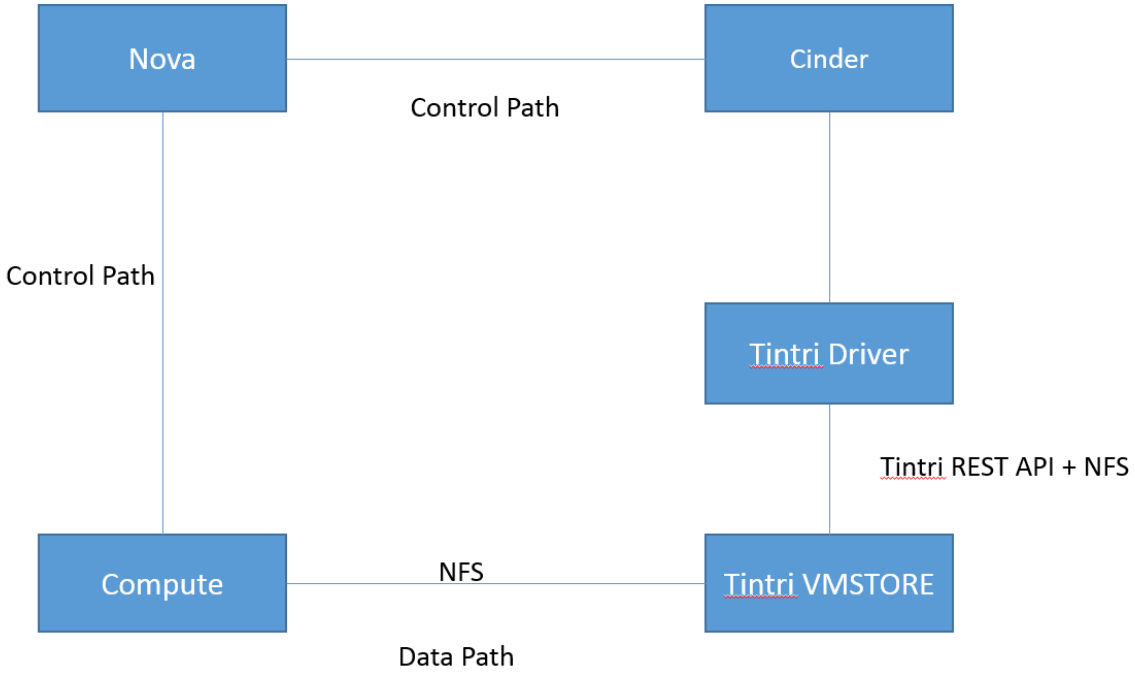
This document provides the details of the setup, configuration and tests run on Mirantis OpenStack with Tintri Cinder driver for the purpose of Unlocked OpenStack Driver Validation. To interpret the results, you must be familiar with:

- Mirantis OpenStack and Fuel Master
- Tempest tests
- Fuel Master UI Validation tests

2. Product Overview

The Tintri Cinder driver enables integration between Tintri storage and OpenStack cloud environments. With the Tintri Cinder driver and the Tintri VMstore appliance, OpenStack users gain unmatched visibility into their Cinder volumes and VMs.
Tintri challenges the storage quo with a system built specifically for virtualized and cloud environments. Tintri eliminates LUNs and volumes—the markers of out-of-date, physical storage—instead using individual virtual machines as the unit of management. With those opaque containers gone, many organizations, including 5 of the Fortune 15, have total visibility into their storage. They can manage it in 1/60th the time, improve performance 6x, and enjoy 10x the VM density. Tintri is smart storage to realize your virtualization vision.

3. Joint reference architecture



4. Networking

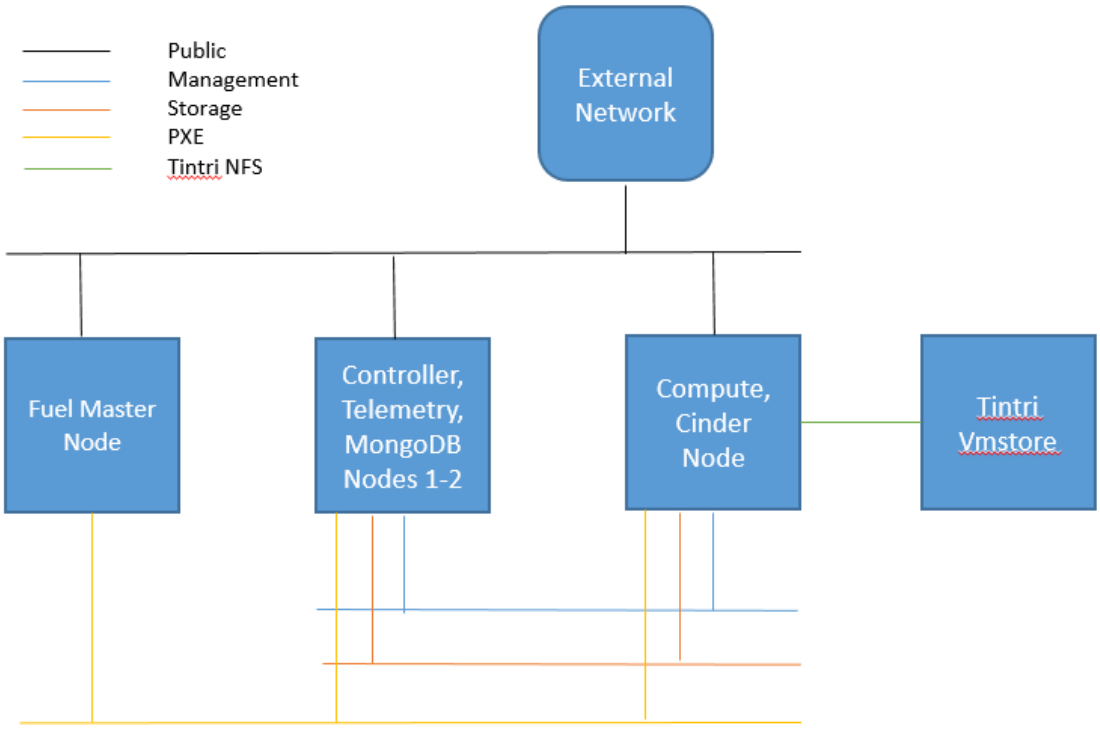
4.1 Physical & Logical network topology

Fuel Server: 2 NICs required:
NIC 1: Port for PXE network
NIC 2: Port for public network traffic

Controller Server: 4 NICs required:
NIC 1: Port for Management and Private network
NIC 2: Port for PXE network
NIC 3: Port for Storage network

NIC 4: Port for public network

Compute Server: 5 NICs required:
NIC 1: Port for Management and Private network
NIC 2: Port for PXE network
NIC 3: Port for Storage network
NIC 4: Port for public network
NIC 5: Port for Tintri VMstore network (NFS to mount volume)



- Fuel Master, Controller, and Compute nodes are connected to 1G public network.
- Fuel Master, Controller, and Compute nodes are connected to 1G PXE network. This network is private and is used for PXE booting the compute and controller nodes during installation.
- Controller and Compute nodes are connected to 1G Storage network. This network is private and is used to provide storage services such as replication traffic from Ceph.
- Controller and Compute nodes are connected to 1G Management network. This network is private and is primarily used for OpenStack Cloud management and for accessing OpenStack services (nova-api, OpenStack dashboard, etc.)
- Compute node and Tintri VMstore are connected to NFS network. This is a 10G network and is used to mount Tintri VMstore volumes on compute node. Tintri VMstore only needs NFS connection to compute node(s).

5. Installation and Configuration

5.1 Overview of MOS installation steps

Setup servers and networks.
Install the Fuel Master node.
Setup and configure controller nodes.
Setup and configure cinder nodes
Setup and configure nova nodes.
Install and configure Tintri Cinder driver.

5.2 MOS Installation in details

Please refer to Mirantis User Guide at https://docs.mirantis.com/openstack/fuel/fuel-7.0/user-guide.html for detail instructions on how to install Mirantis Openstack.

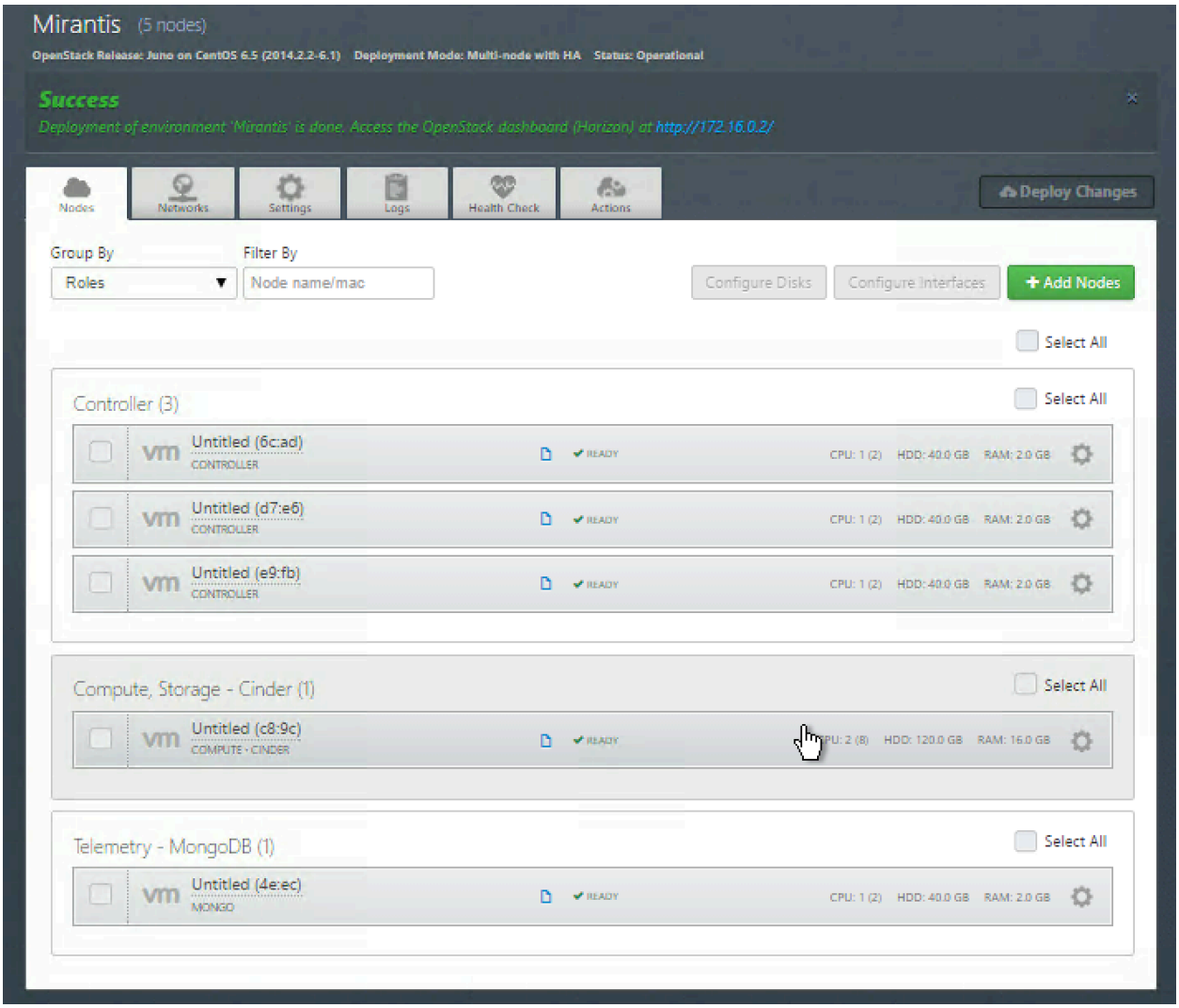
5.3 Creation of OpenStack environment

1. On the Fuel UI, click on "New OpenStack Environment".
2. When the wizard opens, enter the name and the desired OpenStack Release(Ubuntu).
3. Select the Compute for the Envirnment (KVM, Qemu)
4. Select the required Neutron Setup.
5. Under storage backends, leave the default option. Tintri Cinder driver can be installed after the openstack is deployed.
6. Select the additional services and click on finish.

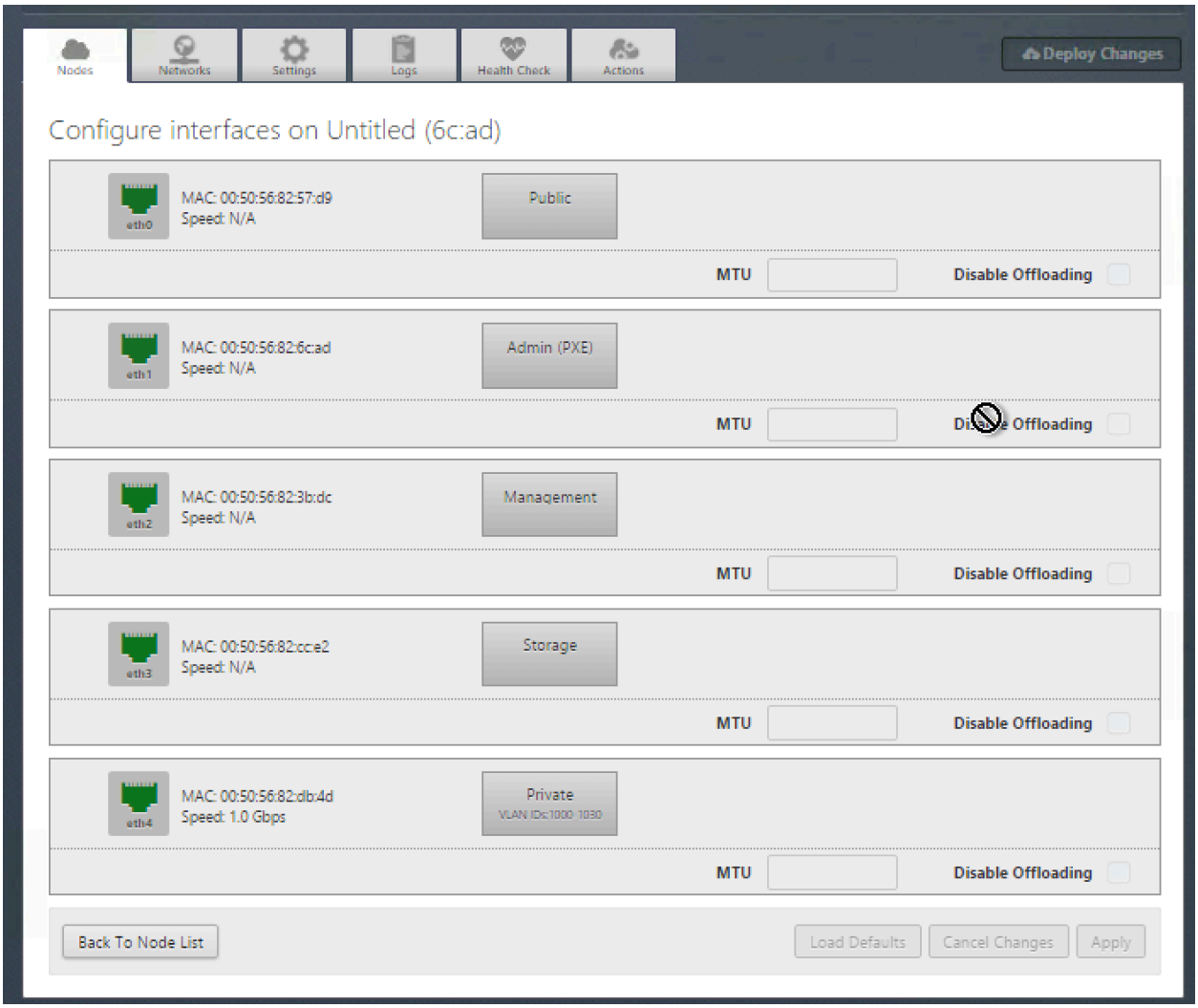
All the nodes (Physical or Virtual) should have access to the Mirantis PXE network(on the same isolated VLAN or switch). Boot the nodes and wait until they show up on the fuel UI.
Add the nodes to the environment and start the deployment.

5.4 MOS Deployment

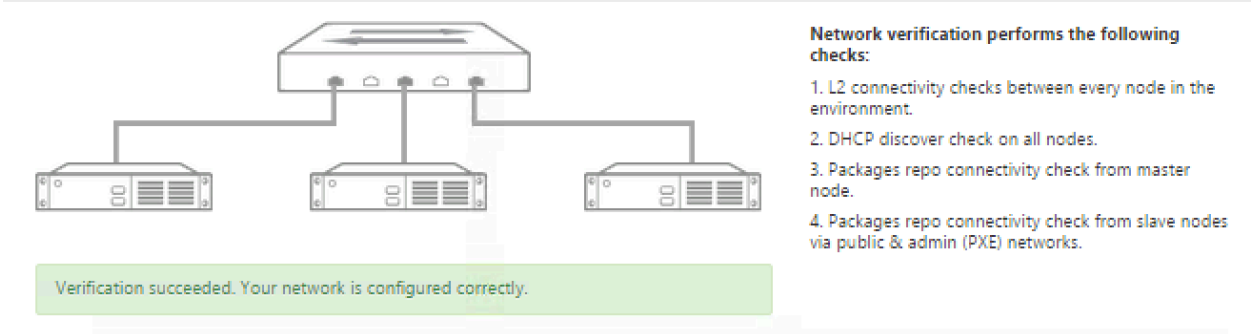
Once all the nodes have booted up through PXE, they appear on the Fuel UI. Configure controllers, cinder and compute. Have multiple controllers in the setup(3 controllers are recommended).



Configure interfaces on the machines. And set networks for Management, Storage, Private and Public networks.



After configuring the nodes, run “Verify Networks” under Networks tab. The test should pass.



After confirming the users, networks, passwords, configuration and settings, Click on “Deploy Changes” to deploy the OpenStack Environment.

5.5 Tintri Cinder Installation steps

Tintri Cinder driver can be downloaded from the Tintri Support portal.
http://support.tintri.com

The following is the procedure that can be followed to install the Tintri Cinder driver.

The below procedure should be performed on the cinder nodes to enable Tintri Cinder Volumes.
Unpack the Tintri Cinder driver tarball.
$ tar –xvzf Tintri_Cinder_Driver_2.1.0.1.tgz

Run the installation Script
$ ./install.sh

Edit /etc/cinder/cinder.conf and add the following configuration for all cinder backends you wish to create – In the below example (tintri5 and tintri6)

```
[cinder_tintri_5]
nfs_shares_config=/etc/cinder/tintri_shares_5
volume_driver=cinder.volume.drivers.tintri.TintriDriver
volume_backend_name=tintri5
nfs_mount_options=vers=3,lookupcache=none
tintri_server_hostname=<x.x.x.x>
tintri_server_username=<username>
tintri_server_password=<password>

[cinder_tintri_6]
nfs_shares_config=/etc/cinder/tintri_shares_6
volume_driver=cinder.volume.drivers.tintri.TintriDriver
volume_backend_name=tintri6
nfs_mount_options=vers=3,lookupcache=none
tintri_server_hostname=<x.x.x.x>
tintri_server_username=<username>
tintri_server_password=<password>
```

NOTE: Replace hostname, username, and password fields above with following values. Please contact your system administrator to obtain these values.

tintri_server_hostname : This is IP address of the admin network of tintri vmstore which can be obtained from hardware tab in vmstore administration UI.
tintri_server_username: This is username to login to admin network interface of tintri vmstore
tintri_server_password: This is password to login to admin network interface of tintri vmstore

Have the backends specified in "enabled_backends" under [DEFAULT]
enabled_backends=cinder_tintri_5,cinder_tintri_6

Create files mentioned in nfs_shares_config and add the location of the shares in the below format.
<Data IP of VMstore>:<Path to the share>
Example –
$ cat /etc/cinder/tintri_shares_5
10.200.161.18:/tintri/cinder5

NOTE: Replace data IP and Path fields above with following values. Please contact your system administrator to obtain these values.

Data IP of VMstore: This is IP address of data network of tintri vmstore which can be obtained from hardware tab in vmstore administration UI.

Path to the share: This is the path to share on tintri vmstore where volumes will be created. "/tintri" in example above is always a constant and need not be changed as tintri vmstore always exposes the share as "/tintri". "cinder5" in example above is a folder that must already exist inside the share. Please contact your system administrator to obtain values for these fields.

Restart the cinder-volume service.

NOTE: You can use the following command to restart cinder volume service:

service cinder-volume restart

On the Nova nodes where instances(that needs to access Tintri volumes) are hosted, make the following changes.
Note that most of the new nfs clients default to NFS4 which causes failure mounting NFS3. The NFS version should be specified in nova.conf under [libvirt] as follows.

nfs_mount_options=vers=3,proto=tcp

Restart the nova-compute service.

NOTE: you can use the following command to restart nova service:

service nova-compute restart

The following commands can be run on the cinder nodes to create volume types.
If multiple backends are present and the user wants to create cinder type, the following commands can be used to associate volume types to the backend.
$ cinder type-create backend5

$ cinder type-key backend5 set volume_backend_name=tintri5

6. Testing

6.1 Test tools

Fuel UI health check

Tempest (http://docs.openstack.org/developer/tempest/overview.html),

Equipment	Purpose
Fuel health check	Functional/acceptance testing
OpenStack Tempest test suite	Functional testing

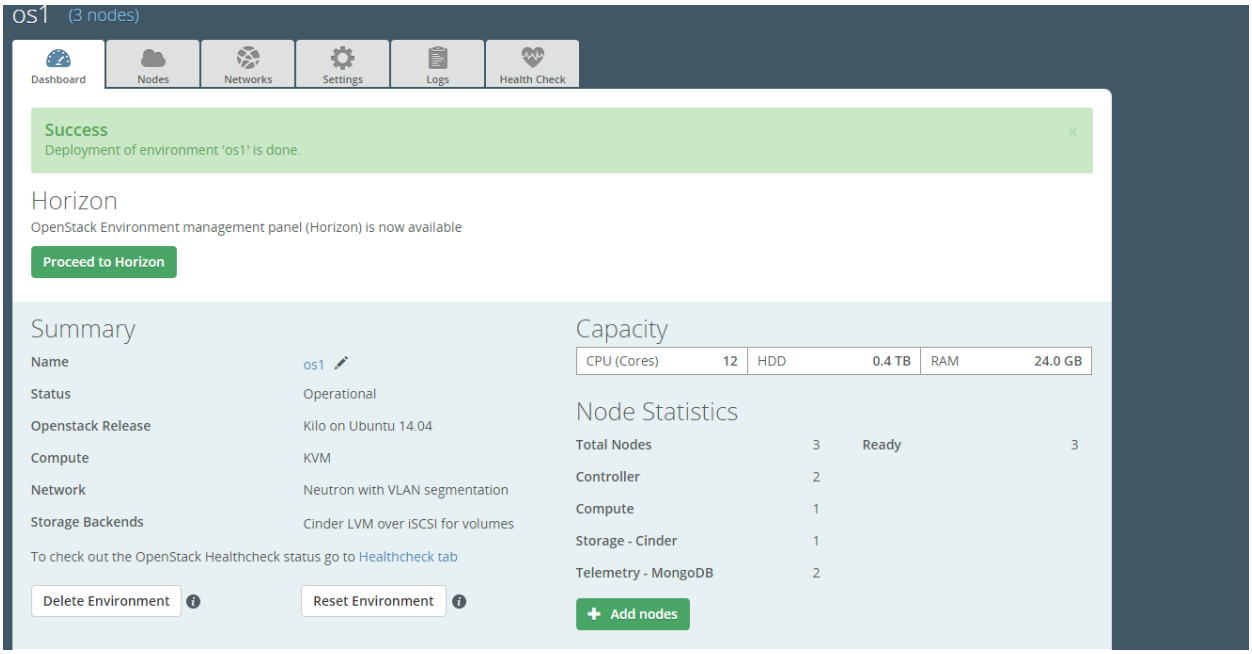
6.2 Test cases

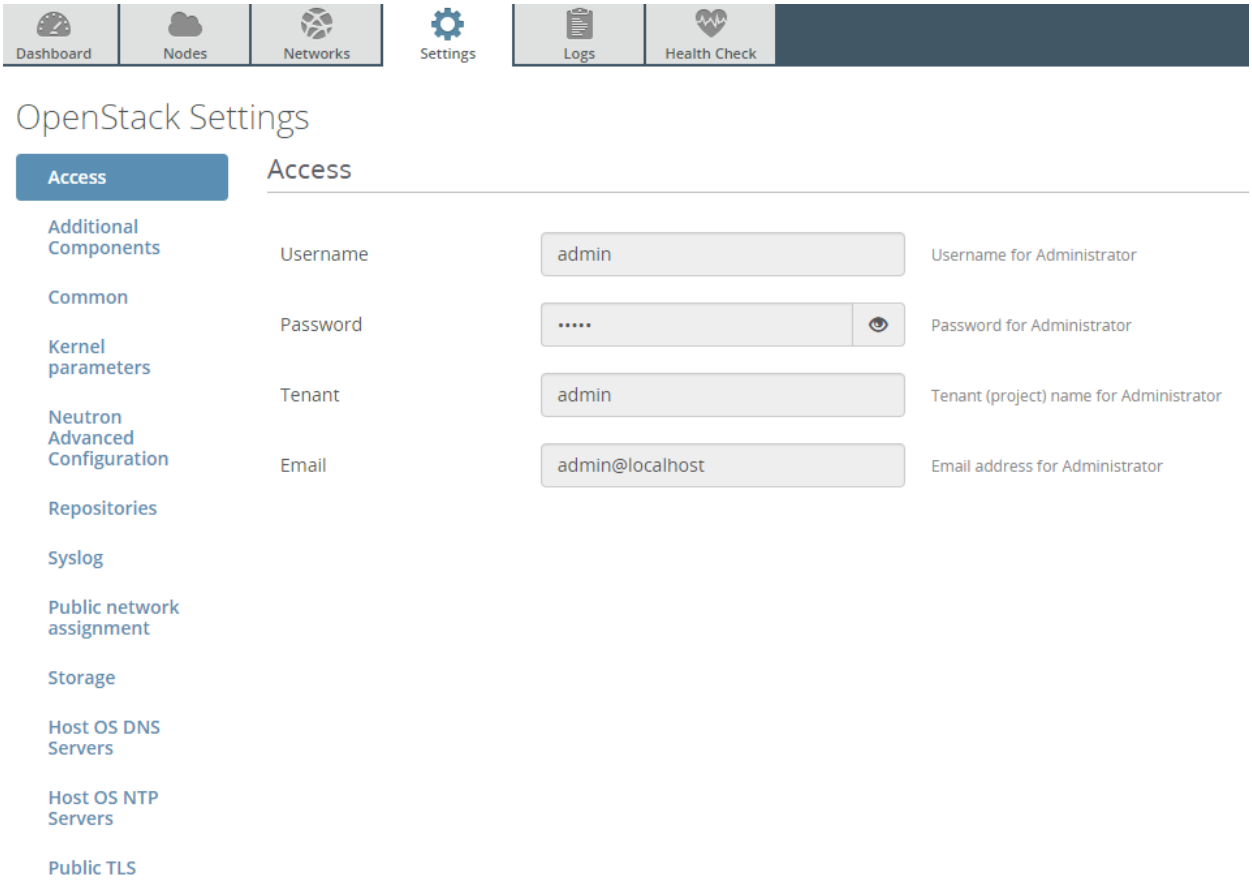
Tempest tests

All tempest volumes tests have passed.
Please refer to Openstack Tempest Testing Project site at
http://docs.openstack.org/developer/tempest/ for details on tempest and related information.

Note that cinder-backup functionality is not supported on Tintri cinder driver and hence these tests are excluded from the test runs.

Fuel UI Health Check Tests and Test Results


os1 (3 nodes)
Dashboard
Nodes
Networks
Settings
Logs
Health Check
Success
Deployment of environment 'os1' is done.
Horizon
OpenStack Environment management panel (Horizon) is now available
Proceed to Horizon
Summary
Name
os1
Status
Operational
Openstack Release
Kilo on Ubuntu 14.04
Compute
KVM
Network
Neutron with VLAN segmentation
Storage Backends
Cinder LVM over iSCSI for volumes
To check out the OpenStack Healthcheck status go to Healthcheck tab
Delete Environment
Reset Environment
Capacity
CPU (Cores) 12
HDD 0.4 TB
RAM 24.0 GB
Node Statistics
Total Nodes 3
Ready 3
Controller 2
Compute 1
Storage - Cinder 1
Telemetry - MongoDB 2
Add nodes


Dashboard
Nodes
Networks
Settings
Logs
Health Check
OpenStack Settings
Access
Additional Components
Common
Kernel parameters
Neutron Advanced Configuration
Repositories
Syslog
Public network assignment
Storage
Host OS DNS Servers
Host OS NTP Servers
Public TLS
Access
Username
admin
Username for Administrator
Password
Password for Administrator
Tenant
admin
Tenant (project) name for Administrator
Email
admin@localhost
Email address for Administrator

OpenStack Settings

- Access
- **Additional Components**
- Common
- Kernel parameters
- Neutron Advanced Configuration
- Repositories
- Syslog
- Public network assignment
- Storage
- Host OS DNS Servers
- Host OS NTP Servers
- Public TLS

Additional Components

- [ ] Install Sahara
  If selected, Sahara component will be installed
- [ ] Install Murano
  If selected, Murano component will be installed
- [x] Install Ceilometer
  If selected, Ceilometer component will be installed
- [ ] Use external Mongo DB
  If selected, You can use external Mongo DB as ceilometer backend

OpenStack Settings

- Access
- Additional Components
- **Common**
- Kernel parameters
- Neutron Advanced Configuration
- Repositories
- Syslog
- Public network assignment
- Storage
- Host OS DNS Servers
- Host OS NTP Servers
- Public TLS

Common

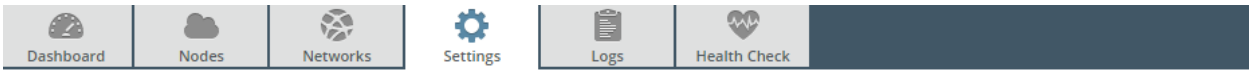
- [x] Puppet debug logging
  Debug puppet logging mode provides more information, but requires more disk space.
- [ ] OpenStack debug logging
  Debug logging mode provides more information, but requires more disk space.
- [ ] Nova quotas
  Quotas are used to limit CPU and memory usage for tenants. Enabling quotas will increase load on the Nova database.

Hypervisor type

- (•) KVM
  Choose this type of hypervisor if you run OpenStack on hardware
- ( ) QEMU
  Choose this type of hypervisor if you run OpenStack on virtual hosts.

- [x] Use qcow format for images
  For most cases you will want qcow format. If it's disabled, raw image format will be used to run VMs. OpenStack with raw format currently does not support snapshotting.
- [x] Resume guests state on host boot
  Whether to resume previous guests state when the host reboots. If enabled, this option causes guests assigned to the host to resume their previous state. If guest was running a restart will be attempted when nova-compute starts. If the guest was not running previously, a restart will not be attempted.

Public Key — Public key(s) to include in authorized_keys on deployed nodes



OpenStack Settings

- Access
- Additional Components
- Common
- **Kernel parameters**
- Neutron Advanced Configuration
- Repositories
- Syslog
- Public network assignment
- Storage
- Host OS DNS Servers
- Host OS NTP Servers
- Public TLS

Kernel parameters

Initial parameters: console=tty0 net.ifnames=0 biosdevnar

Default kernel parameters

OpenStack Settings

- Access
- Additional Components
- Common
- Kernel parameters
- **Neutron Advanced Configuration**
- Repositories
- Syslog
- Public network assignment
- Storage
- Host OS DNS Servers
- Host OS NTP Servers
- Public TLS

Neutron Advanced Configuration

☐ Neutron DVR

Enable Distributed Virtual Routers in Neutron

OpenStack Settings

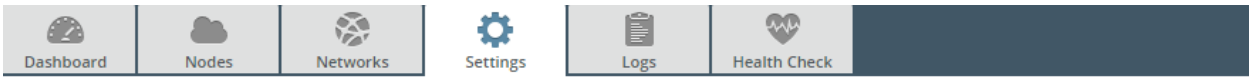
- Access
- Additional Components
- Common
- Kernel parameters
- Neutron Advanced Configuration
- **Repositories**
- Syslog
- Public network assignment
- Storage
- Host OS DNS Servers
- Host OS NTP Servers
- Public TLS

Repositories

Please note: the first repository will be considered the operating system mirror that will be used during node provisioning.
To create a local repository mirror on the Fuel master node, please follow the instructions provided by running "fuel-createmirror --help" on the Fuel
Please make sure your Fuel master node has Internet access to the repository before attempting to create a mirror.
For more details, please refer to the documentation (https://docs.mirantis.com/openstack/fuel/fuel-7.0/operations.html#external-ubuntu-ops).

Name	URI	Priority
ubuntu	deb http://archive.ubuntu.com/ubuntu/ tru	None
ubuntu-updates	deb http://archive.ubuntu.com/ubuntu/ tru	None
ubuntu-security	deb http://archive.ubuntu.com/ubuntu/ tru	None
mos	deb http://10.20.0.2:8080/2015.1.0-7.0/ubu	1050
mos-updates	deb http://mirror.fuel-infra.org/mos-repos/	1050
mos-security	deb http://mirror.fuel-infra.org/mos-repos/	1050
mos-holdback	deb http://mirror.fuel-infra.org/mos-repos/	1100
Auxiliary	deb http://10.20.0.2:8080/2015.1.0-7.0/ubu	1150

Add Extra Repo



OpenStack Settings

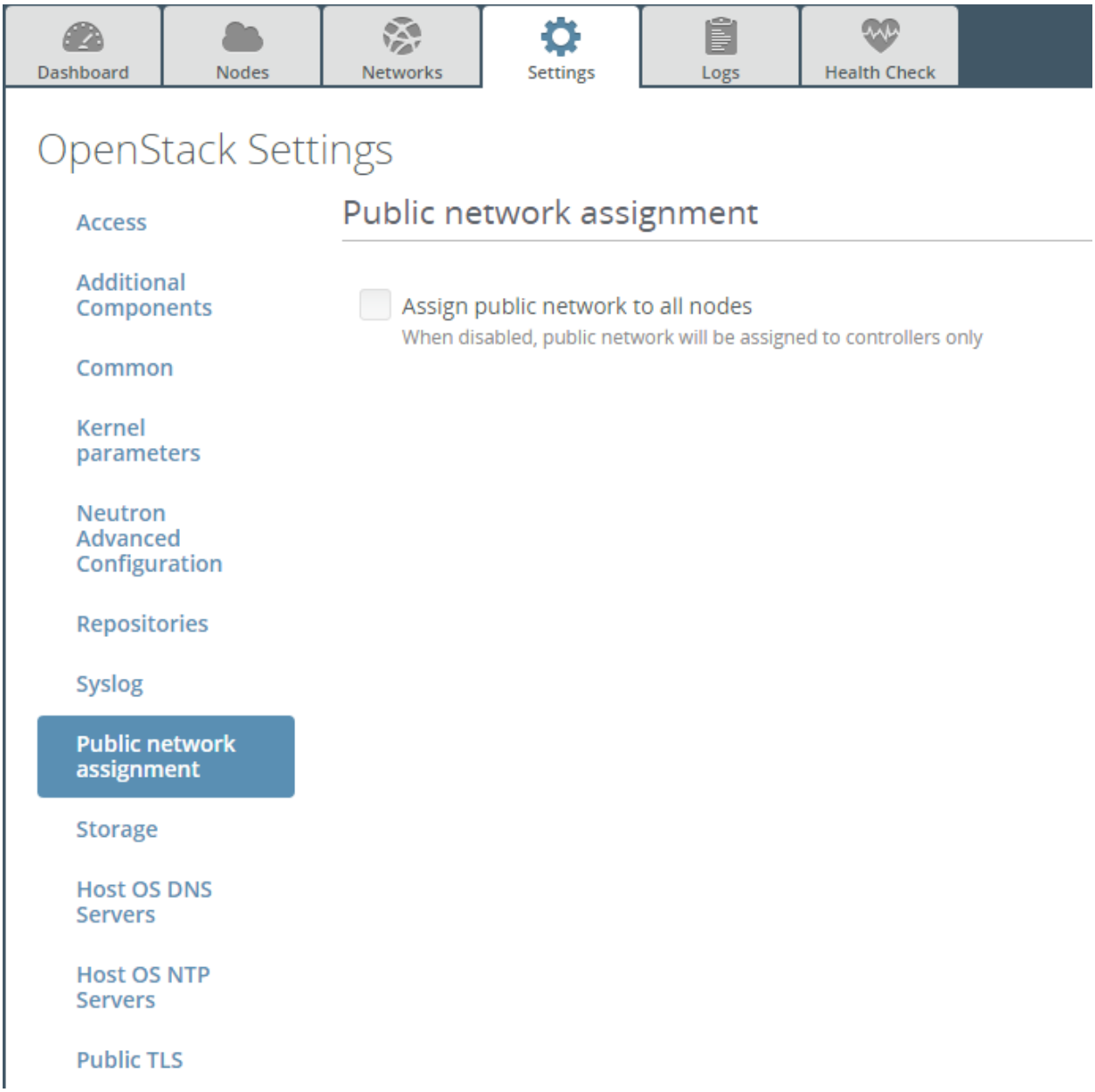
- Access
- Additional Components
- Common
- Kernel parameters
- Neutron Advanced Configuration
- Repositories
- **Syslog**
- Public network assignment
- Storage
- Host OS DNS Servers
- Host OS NTP Servers
- Public TLS

Syslog

Hostname		Remote syslog hostname
Port	514	Remote syslog port

Syslog transport protocol

- ( ) UDP
- (•) TCP


Dashboard
Nodes
Networks
Settings
Logs
Health Check
OpenStack Settings
Access
Additional Components
Common
Kernel parameters
Neutron Advanced Configuration
Repositories
Syslog
Public network assignment
Storage
Host OS DNS Servers
Host OS NTP Servers
Public TLS
Public network assignment
Assign public network to all nodes
When disabled, public network will be assigned to controllers only

OpenStack Settings

- Access
- Additional Components
- Common
- Kernel parameters
- Neutron Advanced Configuration
- Repositories
- Syslog
- Public network assignment
- Storage
- Host OS DNS Servers
- Host OS NTP Servers
- Public TLS

Storage

- [x] Cinder LVM over iSCSI for volumes
  It is recommended to have at least one Storage - Cinder LVM node.
- [ ] Ceph RBD for volumes (Cinder)
  Configures Cinder to store volumes in Ceph RBD images.
- [ ] Ceph RBD for images (Glance)
  Configures Glance to use the Ceph RBD backend to store images. If enabled, this option will prevent Swift from installing.
- [ ] Ceph RBD for ephemeral volumes (Nova)
  Configures Nova to store ephemeral volumes in RBD. This works best if Ceph is enabled for volumes and images, too. Enables live migration of all types of Ceph backed VMs (without this option, live migration will only work with VMs launched from Cinder volumes).
- [ ] Ceph RadosGW for objects (Swift API)
  Configures RadosGW front end for Ceph RBD. This exposes S3 and Swift API Interfaces. If enabled, this option will prevent Swift from installing.

Ceph object replication factor: 2
Configures the default number of object replicas in Ceph. This number must be equal to or lower than the number of deployed 'Storage - Ceph OSD' nodes.

OpenStack Settings

- Access
- Additional Components
- Common
- Kernel parameters
- Neutron Advanced Configuration
- Repositories
- Syslog
- Public network assignment
- Storage
- Host OS DNS Servers
- Host OS NTP Servers
- Public TLS

Host OS DNS Servers

DNS list: 8.8.8.8
List of upstream DNS servers, separated by comma

OpenStack Settings

- Access
- Additional Components
- Common
- Kernel parameters
- Neutron Advanced Configuration
- Repositories
- Syslog
- Public network assignment
- Storage
- Host OS DNS Servers
- **Host OS NTP Servers**
- Public TLS

Host OS NTP Servers

NTP server list: 0.fuel.pool.ntp.org, 1.fuel.pool.ntp.org,

List of upstream NTP servers, separated by comma

OpenStack Settings

- Access
- Additional Components
- Common
- Kernel parameters
- Neutron Advanced Configuration
- Repositories
- Syslog
- Public network assignment
- Storage
- Host OS DNS Servers
- Host OS NTP Servers
- **Public TLS**

Public TLS

- [x] HTTPS for Horizon
  Secure access to Horizon enabling HTTPS instead of HTTP
- [x] TLS for OpenStack public endpoints
  Enable TLS termination on HAProxy for OpenStack services

Select source for certificate

- (•) Self-signed
  Generate private key and certificate that will be signed by this key
- ( ) I have my own keypair with certificate
  Use pre-generated key and certificate

DNS hostname for public TLS endpoints: public.fuel.local

Your DNS entries should point to this name. Self-signed certificates also will use this hostname

Sanity tests

Test	Result
Ceilometer test to list meters, alarms and resources	Pass
Request flavor list	Pass
Request image list using Nova	Pass
Request instance list	Pass
Request absolute limits list	Pass
Request snapshot list	Pass
Request volume list	Pass
Request active services list	Pass
Request user list	Pass
Check that required services are running	Pass
Request list of networks	Pass

Functional tests. Duration

Test	Result
Create instance flavor	Pass
Check create, update and delete image actions using Glance v1	Pass
Check create, update and delete image actions using Glance v2	Pass
Create volume and boot instance from it	Pass
Create volume and attach it to instance	Pass
Check network connectivity from instance via floating IP	Pass
Create keypair	Pass
Create security group	Pass
Check network parameters	Pass
Launch instance	Pass
Launch instance with file injection	Pass
Launch instance, create snapshot, launch instance from snapshot	Pass
Create user and authenticate with it to Horizon	Pass

HA tests

Test	Result
Check data replication over mysql	Pass

Check amount of tables in databases is the same on each node	Pass
Check galera environment state	Pass
Check Pacemaker status	Pass
RabbitMQ availability	Pass
RabbitMQ replication	Pass

Platform services functional tests

Test	Result
Ceilometer test to check alarm state and get Nova metrics	Pass
Ceilometer test to check notifications from Glance	Pass
Ceilometer test to check notifications from Keystone	Pass
Ceilometer test to check notifications from Neutron	Pass
Ceilometer test to check notifications from Cinder	Pass
Ceilometer test to create, check and list samples	Pass
Ceilometer test to create, update, check and delete alarm	Pass
Typical stack actions: create, delete, show details, etc.	Pass
Advanced stack actions: suspend, resume and check	Pass
Check stack autoscaling	Pass
Check stack rollback	Pass
Update stack actions: inplace, replace and update whole template	Pass

Cloud validation tests

Test	Result
Check disk space outage on controller and compute nodes	Pass
Check log rotation configuration on all nodes	Pass

Configuration tests

Test	Result
Check usage of default credentials on master node	Pass

Check usage of default credentials for Openstack cluster	Pass

Troubleshooting:

For more details on troubleshooting health check test failures, refer to “What To Do When A Test Fails” section in Mirantis User guide at:

https://docs.mirantis.com/openstack/fuel/fuel-7.0/user-guide.html#user-guide

For more details on Mirantis health check tests, isolating and debugging failures, and other details, refer to “Details Of Health Checks” and “Troubleshooting” sections of Mirantis Operations Guide at:

https://docs.mirantis.com/openstack/fuel/fuel-7.0/operations.html#post-deployment-check-details

Openstack services (cinder, nova, neutron, etc.) log files are located in “/var/log” directory.

6.2.1 Target Use case(s)

1. Enterprise Private Cloud:
   Tintri offers VM-aware storage that is ideal for enterprise private clouds that leverage MOS with the Cinder service. Tintri offers high performance modern storage over NFS that provides consistent performance across multiple enterprise organizations and users.
2. Cloud Service Provider service offerings:
   CSPs leverage Tintri with Cinder for IaaS offerings. Multiple tenants co-exist on Tintri storage with no noisy neighbors. CSPs benefit from dramatically lower OPEX to manage their storage and reduced storage CAPEX due to Tintri’s high density and small datacenter footprint.
3. Higher level services:
   MOS with Tintri is leveraged with higher level PaaS and SaaS offerings that leverage the OpenStack API and Cinder services. Tintri provides Cinder persistent storage with consistent performance and visibility into Cinder VMs and volumes.

6.2.2 Deployment modes and configuration options

OS	Mode	HV	Network	Storage

			VLAN	Tintri Cinder
Ubuntu	HA	KVM	✓	✓

6.2.3 Functional testing

OS	Mode	HV	Network	Storage
			VLAN	Tintri Cinder
Ubuntu	HA	KVM	✓	✓

Note – Tempest could not be run on CentOS as python that came with CentOS was 2.6 and tempest is no longer supported on Python 2.6. The tempest logs with Ubuntu will be enclosed. Tintri does not support cinder backup and hence the cinder backup tests will be run with the Tintri Cinder driver.

6.2.4 Performance testing

OS	Mode	HV	Network	Storage
			VLAN	Tintri Cinder
Ubuntu	HA	KVM	✓	✓

6.2.5 Negative testing

OS	Mode	HV	Network	Storage
			VLAN	Tintri Cinder
Ubuntu	HA	KVM	✓	✓

No issues seen with manual negative tests. All tempest negative tests have passed.