



Brocade VCS plugin Deployment Guide
In Mirantis OpenStack Environment

Last Update: **9/3/2014**

Document Owner: SQA, Brocade Communications Systems, Inc.

Brocade VCS plugin Deployment Guide in Mirantis OpenStack Environment

Corporate Headquarters
San Jose, CA, USA
T: 1-408-333-8000
info@brocade.com

European Headquarters
Geneva, Switzerland
T: +41 22 799 56 40
emea-info@brocade.com

Asia Pacific Headquarters
Singapore
T: +65-6538-4700
apac-info@brocade.com

© 2014 Brocade Communications Systems, Inc. All Rights Reserved. 01/2014

Brocade, the Brocade B-weave logo, Fabric OS, File Lifecycle Manager, MyView, Secure Fabric OS, SilkWorm, and StorageX are registered trademarks and the Brocade B-wing symbol and Tapestry are trademarks of Brocade Communications Systems, Inc., in the United States and/or in other countries. FICON is a registered trademark of IBM Corporation in the U.S. and other countries. All other brands, products, or service names are or may be trademarks or service marks of, and are used to identify, products or services of their respective owners.

This document is for informational purposes only and does not set forth any warranty, expressed or implied, concerning any equipment, equipment feature, or service offered or to be offered by Brocade. Brocade reserves the right to make changes to this document at any time, without notice, and assumes no responsibility for its use. This informational document describes features that may not be currently available. Contact a Brocade sales office for information on feature and product availability. Export of technical data contained in this document may require an export license from the United States government.

Table of Contents

Brocade VCS Plugin Description:	4
APIs serviced by Brocade VCS Plugin:.....	4
Create_Network	4
Create_Port	4
Delete_Network	4
Delete_Port	4
Support Matrix.....	4
Network Topology:	5
Pre-Requisite:	5
Master Node Configuration:	6
Configuration Check – Brocade VCS Plugin code and ncclient:.....	8
Configuration Check – Neutron Configuration Files:	9
/etc/neutron/neutron.conf.....	9
/etc/neutron/plugins/openvswitch/ovs_neutron_plugin.ini	9
/etc/neutron/plugins/ml2/ml2_conf.ini	9
ovs configuration.....	9
Configuration Check – Neutron Database Configurations:.....	12
Configuration Check – Brocade VCS Device Configurations:	13
Eth1 Configurations:.....	13
Eth2 Configurations:.....	13
Eth3 Configurations:.....	13
Other Brocade VCS Commands:.....	13
Test Report	14
Health Check Results.....	15
Support Details	17

Brocade VCS Plugin Description:

Brocade provides a network service to OpenStack community by providing plugin to communicate with VCS/VDX devices. This Plugin supports management of L2 network on VCS/VDX devices. Plugin provides the services provided by OpenvSwitch to create the VLAN bridges on Compute Nodes. This plugin deploys VLAN configuration and AMPP (Automatic migration of Port Profiles) configurations in the VCS device.

APIs serviced by Brocade VCS Plugin:

List of API calls that are serviced by Brocade VCS plugin are –

Create_Network -> Using this API, Brocade Plugin will create a VLAN on VCS device

Create_Port -> Using this API, Brocade Plugin will create a Virtual machine on the compute node corresponding to the VLAN (selected network) on the VCS device

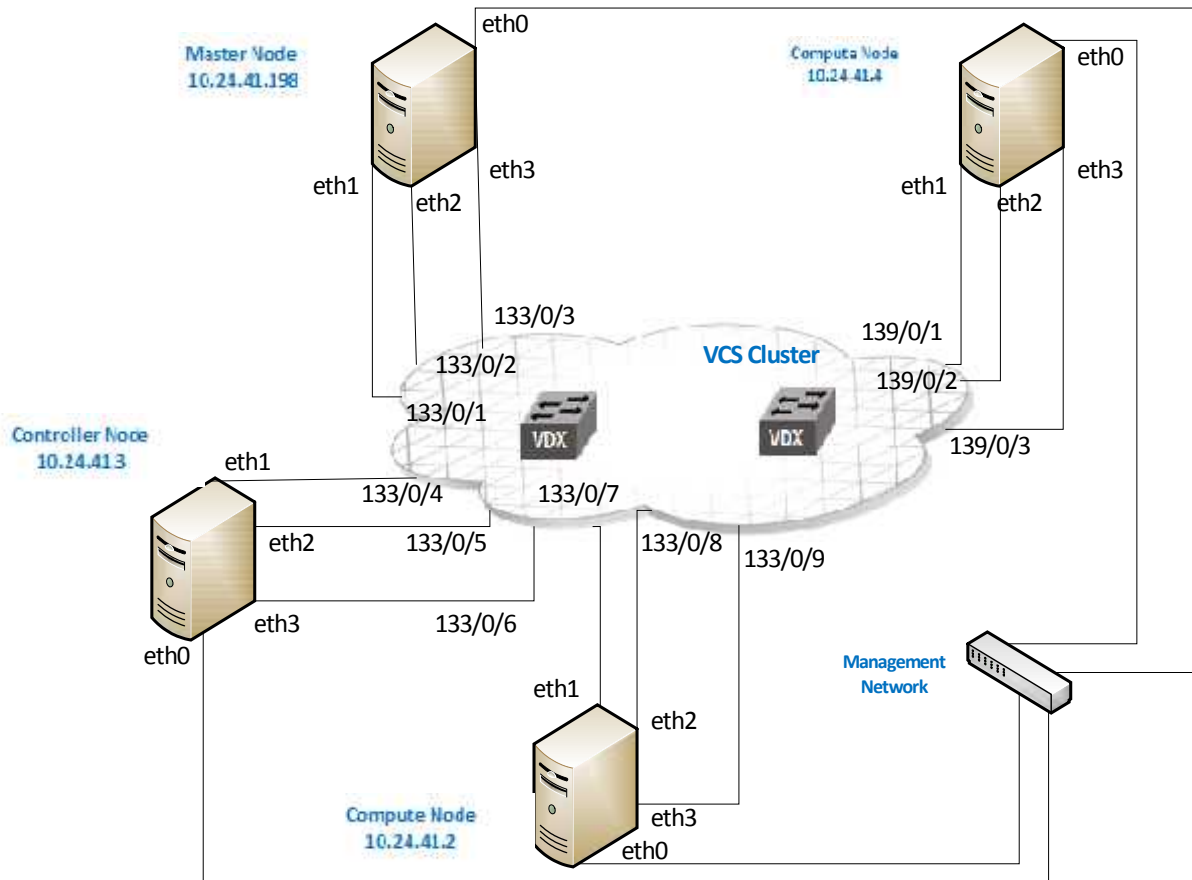
Delete_Network -> Using this API, Brocade Plugin will delete the VLAN on the VCS device corresponding to that Network

Delete_Port -> Using this API, Brocade Plugin will delete the Virtual machine on the compute node and clear the port association for the VLAN (selected network) on the VCS device

Support Matrix

Environment	Description
Supported VCS Devices	VDX6710, VDX6720, VDX6730 VDX8770, VDX6740, VDX6740T
Supported NOS Versions	NOS4.0.x
Supported VCS Cluster Mode	Logical Chassis
Supported OS Platform	Ubuntu
Upstream OpenStack Versions	2014.1 (Icehouse)
VLAN Limit/Range	1 - 4096
Mirantis OpenStack Version	5.0

Network Topology:



Eth0 – Connected to management network (Public network)
 Eth1 – Admin (PXE) – Access mode configuration in VCS device
 Eth2 – Private, Mgmt & Storage network – Trunk mode configuration in VCS device
 Eth3 – For Brocade Plugin – Port Profile mode configuration in VCS device

Pre-Requisite:

1. A user can download MOS version with or without purchasing a support contract (subscription) from Mirantis <http://software.mirantis.com/>
2. More details about System/Hardware requirements, recommended configurations and setting up the environment are available at <http://docs.mirantis.com/fuel/fuel-4.1/pre-install-guide.html>

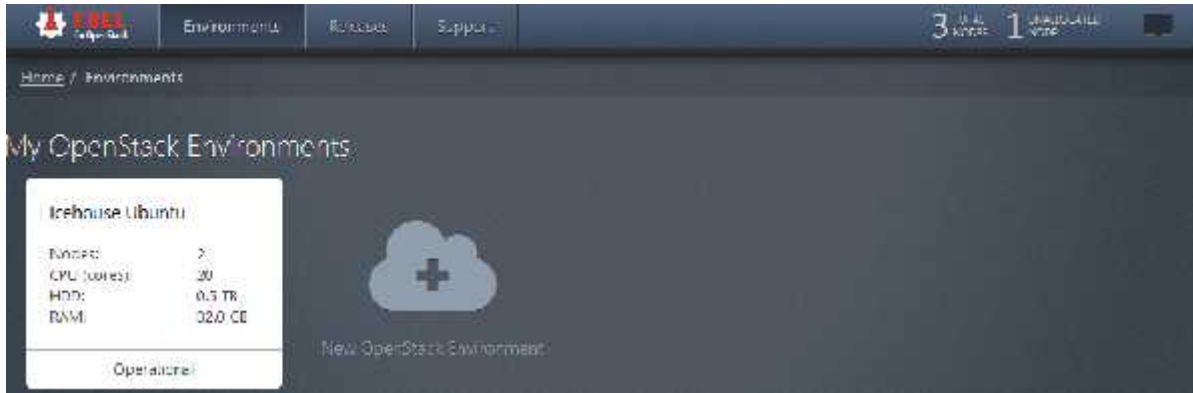
Master Node Configuration:

Master Node is the primary node, which manages all the Controller and Compute nodes and is responsible for

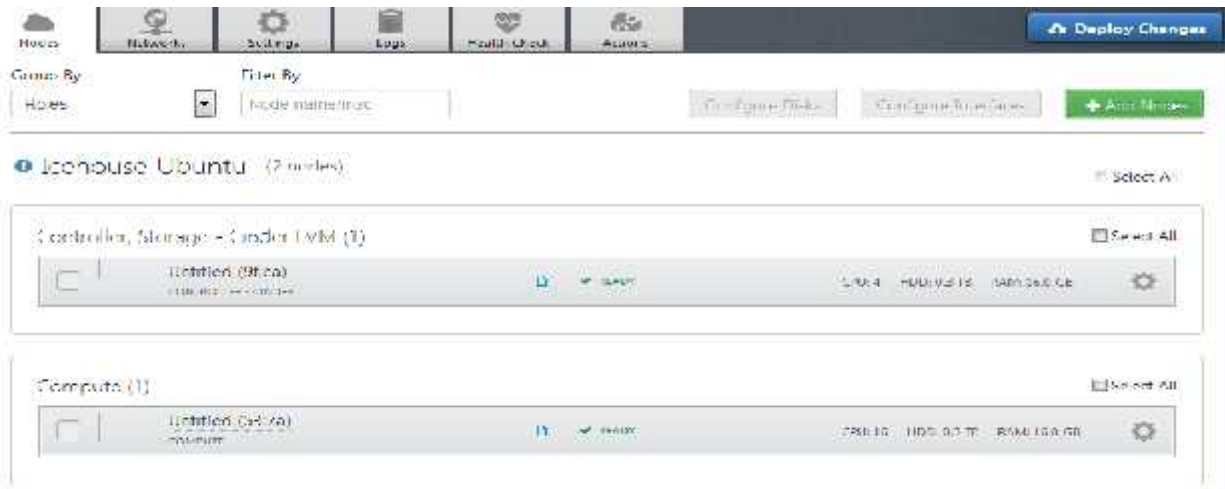
- Deploying the OS (Operating System) in the Controller and Compute nodes using PXE boot
- Deploying the Controller and Compute node configuration easily through Fuel UI
- Checking the status of the Controller and Compute nodes regularly

OpenStack deployment is made easier through intuitive UI called Fuel UI which is hosted in the Master Node. Fuel UI can also provide options for Network settings, Health Check and Logs tracking. Below are the step by step procedures for configuration and troubleshooting Controller and Compute nodes using Fuel UI.

1. Create Environment – click on the ‘New OpenStack Environment’ cloud icon. Environment can be able to Reset or Delete using ‘Actions’ tab



2. Make sure nodes are detected in Master Node. Click on ‘Add Nodes’, to assign the Roles to the discovered nodes.



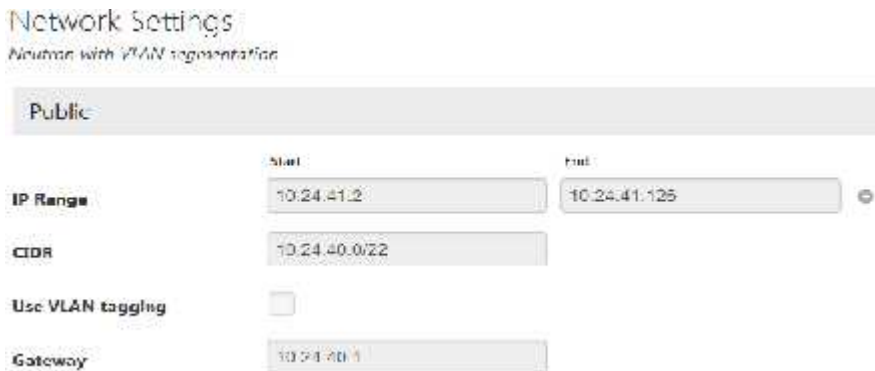
- Public, Admin, Private, Storage and Management networks can be associated to their respective Ethernet interfaces using 'Configure Interfaces' option.



- Disk allocation can be configured or viewed using 'Configure Disks' option.



- Verify Networks tab is populated with correct values



Neutron L2 Configuration

	Start	End	
VLAN ID range	<input type="text" value="100"/>	<input type="text" value="100"/>	
Base MAC address	<input type="text" value="fa:16:3e:00:00:00"/>		

Neutron L3 Configuration

Internal network CIDR	<input type="text" value="192.168.111.0/24"/>		
Internal network gateway	<input type="text" value="192.168.111.1"/>		
	Start	End	
Floating IP ranges	<input type="text" value="10.24.41.130"/>	<input type="text" value="10.24.41.254"/>	⊖
DNS Servers	<input type="text" value="8.8.8.8"/>	<input type="text" value="8.8.8.8"/>	

- Horizon credentials can be configured in Master node as –

OpenStack Settings

Access

username	<input type="text" value="admin"/>	Username for Administrator
password	<input type="password" value="****"/>	Password for Administrator
tenant	<input type="text" value="admin"/>	Tenant (project) name for Administrator
email	<input type="text" value="admin@example.org"/>	Email address for Administrator

- Click on 'Deploy Changes' button

Configuration Check – Brocade VCS Plugin code and ncclient:

- Verify the presence of Brocade VCS ML2 plugin code at location `/usr/lib/python2.7/dist-packages/neutron/plugins/ml2/drivers/brocade` and compare the files with Ice House upstream. Make sure the below code is available at `nosdriver.py` file

`/usr/lib/python2.7/dist-packages/neutron/plugins/ml2/drivers/brocade/nos/nosdriver.py`

```
from neutron.plugins.ml2.drivers.brocade.nos import nctemplates as template
#from neutron.plugins.brocade.nos import nctemplates as template
```

- On the controller node, install the netconf client (ncclient) which is required to communicate with the Brocade VCS cluster

```
# apt-get install git
# git clone https://code.gnnet.gr/git/ncclient
# cd ncclient && python setup.py install
```


Configuration Check – Neutron Configuration Files:

Make sure below mentioned configurations are done in respective configuration files in Controller and Compute nodes

/etc/neutron/neutron.conf

```
[DEFAULT]
core_plugin = neutron.plugins.ml2.plugin.Ml2Plugin
[database]
connection =
mysql://neutron:password@192.168.0.3:3306/neutron_ml2?read_timeout=60
```

/etc/neutron/plugins/openvswitch/ovs_neutron_plugin.ini

```
[ovs]
tenant_network_type = vlan
network_vlan_ranges =physnet1:400:600
integration_bridge = br-int
bridge_mappings =physnet1:br-eth3
```

/etc/neutron/plugins/ml2/ml2_conf.ini

```
[ml2]
tenant_network_types = vlan
type_drivers = vlan
mechanism_drivers = openvswitch,brocade
[ml2_type_vlan]
network_vlan_ranges = physnet1:400:500
[database]
connection =
mysql://neutron:password@192.168.0.3:3306/neutron_ml2?read_timeout=60
[ovs]
tenant_network_type = vlan
network_vlan_ranges = physnet1:400:500
bridge_mappings = physnet1:br-eth3
[ml2_brocade]
username = admin
password = password
address = 10.25.225.133
ostype = NOS
physical_networks = physnet1
```

ovs configuration

Make sure ovs-vsctl lists the required bridge configurations as below -

```
root@node-18:~# ovs-vsctl show
583bdc4f-52d0-493a-8d51-a613a4da6c9a
    Bridge "br-eth2"
        Port "br-eth2"
            Interface "br-eth2"
                type: internal
        Port "br-eth2--br-storage"
            tag: 102
            Interface "br-eth2--br-storage"
```

```

        type: patch
        options: {peer="br-storage--br-eth2"}
Port "br-eth2--br-mgmt"
    tag: 101
    Interface "br-eth2--br-mgmt"
        type: patch
        options: {peer="br-mgmt--br-eth2"}
Port "eth2"
    Interface "eth2"
Port "br-eth2--br-prv"
    Interface "br-eth2--br-prv"
        type: patch
        options: {peer="br-prv--br-eth2"}
Bridge br-mgmt
    Port "br-mgmt--br-eth2"
        Interface "br-mgmt--br-eth2"
            type: patch
            options: {peer="br-eth2--br-mgmt"}
    Port br-mgmt
        Interface br-mgmt
            type: internal
Bridge "br-eth0"
    Port "br-eth0"
        Interface "br-eth0"
            type: internal
    Port "br-eth0--br-ex"
        trunks: [0]
        Interface "br-eth0--br-ex"
            type: patch
            options: {peer="br-ex--br-eth0"}
    Port "eth0"
        Interface "eth0"
Bridge "br-eth1"
    Port "br-eth1--br-fw-admin"
        trunks: [0]
        Interface "br-eth1--br-fw-admin"
            type: patch
            options: {peer="br-fw-admin--br-eth1"}
    Port "eth1"
        Interface "eth1"
    Port "br-eth1"
        Interface "br-eth1"
            type: internal
Bridge br-ex
    Port "br-ex--br-eth0"
        trunks: [0]
        Interface "br-ex--br-eth0"
            type: patch
            options: {peer="br-eth0--br-ex"}
    Port br-ex
        Interface br-ex
            type: internal
    Port "qg-83437e93-e0"
        Interface "qg-83437e93-e0"
            type: internal
    Port phy-br-ex

```

```

        Interface phy-br-ex
    Bridge br-int
        Port "int-br-eth3"
            Interface "int-br-eth3"
        Port int-br-prv
            Interface int-br-prv
        Port br-int
            Interface br-int
                type: internal
        Port int-br-ex
            Interface int-br-ex
        Port "tap373e4404-77"
            tag: 6
            Interface "tap373e4404-77"
                type: internal
    Bridge br-storage
        Port br-storage
            Interface br-storage
                type: internal
        Port "br-storage--br-eth2"
            Interface "br-storage--br-eth2"
                type: patch
                options: {peer="br-eth2--br-storage"}
    Bridge br-prv
        Port phy-br-prv
            Interface phy-br-prv
        Port "br-prv--br-eth2"
            Interface "br-prv--br-eth2"
                type: patch
                options: {peer="br-eth2--br-prv"}
        Port br-prv
            Interface br-prv
                type: internal
    Bridge br-fw-admin
        Port "br-fw-admin--br-eth1"
            trunks: [0]
            Interface "br-fw-admin--br-eth1"
                type: patch
                options: {peer="br-eth1--br-fw-admin"}
        Port br-fw-admin
            Interface br-fw-admin
                type: internal
    Bridge "br-eth3"
        Port "phy-br-eth3"
            Interface "phy-br-eth3"
        Port "eth3"
            Interface "eth3"
        Port "br-eth3"
            Interface "br-eth3"
                type: internal
    ovs_version: "1.10.1"

```

Configuration Check – Neutron Database Configurations:

By default, in mysql, ovs_neutron database will be available. Delete the existing ovs_neutron database and create neutron_ml2 database which is the recommended database name since we are intended to use ML2 plugin.

Create neutron_ml2 databases and provide privileges as below -

```
Create database neutron_ml2;
GRANT ALL PRIVILEGES ON neutron_ml2.* TO 'neutron'@'localhost' \
IDENTIFIED BY 'password';
GRANT ALL PRIVILEGES ON neutron_ml2.* TO 'neutron'@'<compute node IP>' \
IDENTIFIED BY 'password';
FLUSH PRIVILEGES;
```

Restart neutron-server and mysql services using below commands –

```
service neutron-server restart
service mysql restart
```

Make sure below tables are available at neutron_ml2 database which includes brocade tables

```
mysql> show tables;
```

```
+-----+
| Tables_in_neutron_ml2 |
+-----+
| agents                 |
| allowedaddresspairs    |
| dnsnameservers         |
| externalnetworks      |
| extradhcpopts          |
| ipallocationpools      |
| ipallocations          |
| ipavailabilityranges   |
| ml2_brocadenetworks    |
| ml2_brocadeports       |
| ml2_flat_allocations   |
| ml2_gre_allocations    |
| ml2_gre_endpoints      |
| ml2_network_segments  |
| ml2_port_bindings      |
| ml2_vlan_allocations   |
| ml2_vxlan_allocations  |
| ml2_vxlan_endpoints    |
| networkdhcpagentbindings |
| networks               |
| ports                  |
| quotas                 |
| securitygroupportbindings |
| securitygrouprules    |
| securitygroups         |
| subnetroutes           |
| subnets               |
+-----+
27 rows in set (0.00 sec)
```

Configuration Check – Brocade VCS Device Configurations:

Eth1 Configurations:

VCS ports which are connected to eth1 (Admin PXE) of the servers should be configured with Access mode configurations as below –

```
interface TenGigabitEthernet 133/0/1
fabric isl enable
fabric trunk enable
switchport
switchport mode access
switchport access vlan 1
spanning-tree shutdown
no shutdown
```

Eth2 Configurations:

VCS ports which are connected to eth2 (Private, Management and Storage networks) of the servers should be configured with Trunk mode configurations as below –

```
interface TenGigabitEthernet 133/0/2
fabric isl enable
fabric trunk enable
switchport
switchport mode trunk
switchport trunk allowed vlan add 101-103
switchport trunk tag native-vlan
spanning-tree shutdown
no shutdown
```

Eth3 Configurations:

VCS ports which are connected to eth3 of the servers should be configured with Profile mode and this is the Brocade plugin's functionality

```
interface TenGigabitEthernet 133/0/3
fabric isl enable
fabric trunk enable
port-profile-port
no shutdown
```

Other Brocade VCS Commands:

Below are the VCS commands to verify the VLAN and Port Profile configurations

```
#show vlan brief
#show port-profile status
#show mac-address-table
```

Test Report

Below are the functionality test cases that are tested as part of Brocade VCS Plugin in the Mirantis OpenStack Environment.

S.No	Test Case Title	Test Case Description	Result
1	Create Network	Verify create network is successful and VLAN is created on VCS device	Pass
2	Create Multiple Networks	Verify multiple VLANs are created on the VCS device	Pass
3	Delete Network	Verify the VLAN is deleted from the VCS device	Pass
4	Delete Multiple Networks	Verify multiple VLANs are deleted from the VCS device	Pass
5	Create Network when VLAN range exceeds	Verify VLANs are not created on VCS devices	Pass
6	Launch instance	Verify Create Instance is successful <ul style="list-style-type: none"> • Verify the MAC address assigned to the VM by launching VM console • Verify the same MAC address is listing in MAC address table 	Pass
7	Create Multiple Instances	Verify Create Instances are successful <ul style="list-style-type: none"> • Verify the MAC addresses assigned to the VMs by launching VM console • Verify the multiple MAC addresses are listing in MAC address table 	Pass
8	Delete instance	Verify the MAC address of the deleted VM is not listing in MAC address table	Pass
9	Delete multiple instances	Verify the MAC addresses of the deleted VMs are not listing in MAC address table	Pass
10	Reboot of VCS Device	Verify even after the reboot of VCS device, created network (VLAN) and MAC addresses are listing properly	Pass
11	Ping between the VMs on same host.	Verify ping between the VMs present in same host is successful	Pass
12	Ping between the VMs on different host	Verify ping between the VMs present in different hosts is successful	Pass
13	Ping between the VMs on another Network.	Verify ping should not happen between VMs present in different networks	Pass
14	Create Duplicate Network with the existing subnet	Verify VLAN is not created on VCS devices	Pass
15	Disturb switch to server connection when Ping is happening between VMs.	Verify the following - <ul style="list-style-type: none"> • Verify the packet is getting dropped and ping is not successful • Verify the ping is getting successful again 	Pass
16	Create Network and Create Port when switch is not reachable.	Verify the creation is getting failed	Pass

Health Check Results

Health Check has been run after configuring the Brocade VCS Plugin in the Mirantis Open Stack environment and below are the snapshots -

<input checked="" type="checkbox"/>	Sanity tests. Duration 30 sec - 2 min	Expected Duration	Actual Duration	Status
<input checked="" type="checkbox"/>	Request flavor list	20 s	0.2 s	
<input checked="" type="checkbox"/>	Request image list	20 s	0.1 s	
<input checked="" type="checkbox"/>	Request instance list	20 s	0.2 s	
<input checked="" type="checkbox"/>	Request absolute limits list	20 s	0.0 s	
<input checked="" type="checkbox"/>	Request snapshot list	20 s	0.2 s	
<input checked="" type="checkbox"/>	Request volume list	20 s	0.0 s	
<input checked="" type="checkbox"/>	Request stack list	20 s	0.0 s	
<input checked="" type="checkbox"/>	Request active services list	20 s	0.3 s	
<input checked="" type="checkbox"/>	Request user list	20 s	0.1 s	
<input checked="" type="checkbox"/>	Check that required services are running	100 s	2.0 s	
<input checked="" type="checkbox"/>	Check internet connectivity from a compute	100 s	2.6 s	
<input checked="" type="checkbox"/>	Check DNS resolution on compute node	120 s	1.3 s	
<input checked="" type="checkbox"/>	Request list of networks	20 s	0.7 s	

✓ Functional tests. Duration 3 min - 14 min	Expected Duration	Actual Duration	Status
✓ Create instance flavor	30 s.	0.3 s.	✓
✓ Create volume and attach it to instance	150 s.	54.1 s.	✓
✓ Create keypair	25 s.	0.4 s.	✓
✓ Create security group	25 s.	0.3 s.	✓
✓ Check network parameters	50 s.	0.1 s.	✓
✓ Launch instance	700 s.	21.3 s.	✓
✓ Check network connectivity from instance via floating IP No available floating IP found. Please refer to OpenStack logs for more details.	300 s.	10.9 s.	✗
✓ Launch instance, create snapshot, launch instance from snapshot	300 s.	45.6 s.	✓
✓ Create user and authenticate with it to Horizon	80 s.	0.3 s.	✓
✓ Platform services functional tests. Duration 3 min - 80 min	Expected Duration	Actual Duration	Status
✓ Typical stack actions: create, update, delete, show details, etc.	440 s.	34.8 s.	✓
✓ Check stack auto-scaling Image with cfnutils package wasn't imported into Glance, please check http://docs.mirantis.com/openstack/fuel/fuel-5.0/user-guide.html#platform-tests-description. Please refer to OpenStack logs for more details.	2600 s.	0.1 s.	✗
✓ Check stack rollback	140 s.	12.7 s.	✓



Support Details

Customers with valid Mirantis and Brocade support contracts can contact Mirantis for any Open Stack related issues and Brocade for any VDX/NOS and Plug-in related issues.

Below are the valid Brocade support contacts-

Brocade Contact: <https://www.brocade.com/service-support/index.html>

Brocade Direct Support SLA: <http://www.brocade.com/services-support/support-plans/direct-support/index.page>