Brocade VCS ML2 & L3 plugin Deployment Guide

In Mirantis OpenStack v6.0 Environment

Last Update: 6/10/2015

Document Owner: SQA, Brocade Communications Systems, Inc.
# Table of Contents

Brocade VCS and SVI Plugin Description: ................................................................. 5

APIs serviced by Brocade VCS L2 Plugin: ................................................................. 5
  Create_Network ........................................................................................................ 5
  Create_Port ............................................................................................................... 5
  Delete_Network ........................................................................................................ 5
  Delete_Port ............................................................................................................... 5

APIs serviced by Brocade VCS L3 Plugin: ................................................................. 5
  Create_router, Update_router -> ................................................................. 5
  add_interface_to_router -> ............................................................................. 5
  Remove_interface_from_router -> ................................................................. 5
  Delete_router -> .................................................................................................. 5

Support Matrix .......................................................................................................... 6

Network Topology: .................................................................................................... 7

Pre-Requisite: ........................................................................................................... 7

Master Node Configuration: .................................................................................... 8

Configuration Check – Brocade VCS Device Configurations: ............................... 8
  Eth0 Configurations: ............................................................................................. 10
  Eth2 Configurations: ............................................................................................. 10
  Eth3 Configurations: ............................................................................................. 10
  Other Brocade VCS Commands: ......................................................................... 10

Configuration Check – Brocade VCS Plugin code and ncclient: ............................ 13

Configuration Check – Neutron Configuration Files: ............................................ 13
  /etc/neutron/neutron.conf .................................................................................. 13
  /etc/neutron/plugins/openvswitch/ovs_neutron_plugin.ini (Only on the compute nodes) ... 13
  /etc/neutron/plugins/ml2/ml2_conf.ini ............................................................. 14
  ovs configuration ............................................................................................... 14
SVI - L3 Networking driver. ................................................................. 16
Test Report ....................................................................................... 17
Health Check Results....................................................................... 18
Note : ............................................................................................. 20
Support Details ................................................................................ 21
Brocade VCS and SVI 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 and L3 router 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.

SVI support will be implemented using Openstack L3 extension API. The I3 service framework will be used to create the router, add the subnet to VLAN.

APIs serviced by Brocade VCS L2 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

APIs serviced by Brocade VCS L3 Plugin:
List of API calls that are serviced by Brocade VCS L3 plugin are –

Create Router, Update router -> create_router and update_router methods will set the admin_state state as down

add_interface_to_router -> Openstack subnet_id is provided in the request parameters. Using this API, Brocade Plugin will create ve interface and assign gateway ip of the subnet to the VCS device

Remove_interface_from_router -> Using this API, Brocade Plugin will remove ve interface along with assigned gateway ip of the subnet from VCS device

Delete_router -> This method is used to clean up the NETCONF, db connection and any caches.
Support Matrix

<table>
<thead>
<tr>
<th>Environment</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Supported VCS Devices</td>
<td>VDX6710, VDX6720, VDX6730</td>
</tr>
<tr>
<td></td>
<td>VDX8770, VDX6740, VDX6740T</td>
</tr>
<tr>
<td>Supported NOS Versions</td>
<td>NOS4.0.x /NOS5.0.x</td>
</tr>
<tr>
<td>Supported VCS Cluster Mode</td>
<td>Logical Chassis</td>
</tr>
<tr>
<td>Supported OS Platform</td>
<td>Ubuntu 12.04 LTS</td>
</tr>
<tr>
<td>Upstream OpenStack Versions</td>
<td>2014.2 (JUNO)</td>
</tr>
<tr>
<td>VLAN Limit/Range</td>
<td>1 - 4096</td>
</tr>
<tr>
<td>Mirantis OpenStack Version</td>
<td>6.x</td>
</tr>
</tbody>
</table>
Network Topology:

Eth0 -- Admin (PXE) – Access mode configuration in VCS

Eth1 – Connected to public network

Eth2 – Management and storage network – trunk mode configuration in VCS

Eth3 – Connected to VCS cluster – port profile mode configured in VCS

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/openstack/fuel/fuel-6.0/
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

Configuration Check – Brocade VCS Device Configurations:
Login to the switch 10.25.225.212 with username and password as admin/password.

Associate the vlan in the interfaces as follows,

VDX device connection to Fuel Master Node:

```
sw0(config)# interface TenGigabitEthernet 212/4/4
sw0(conf-if-te-212/4/4)# switchport
sw0(conf-if-te-212/4/4)# switchport mode access
sw0(conf-if-te-212/4/4)# switchport access vlan 100
sw0(conf-if-te-212/4/4)# no shutdown
sw0(conf-if-te-212/4/4)# exit

sw0(config)# interface TenGigabitEthernet 212/4/5
sw0(conf-if-te-212/4/5)# switchport
```
sw0(conf-if-te-212/4/5)#switchport mode trunk
sw0(conf-if-te-212/4/5)# switchport trunk allowed vlan add 101-102
sw0(conf-if-te-212/4/5)# switchport trunk tag native-vlan
sw0(conf-if-te-212/4/5)#no shutdown
sw0(conf-if-te-212/4/5)#exit

sw0(config)#interface TenGigabitEthernet 212/4/6
sw0(conf-if-te-212/4/6)#port-profile-port
sw0(conf-if-te-212/4/6)#no shutdown
sw0(conf-if-te-212/4/6)#exit

VDX device connection to Slave Node 1:

sw0(config)# interface TenGigabitEthernet 212/4/10
sw0(conf-if-te-212/4/10)# switchport
sw0(conf-if-te-212/4/10)# switchport mode access
sw0(conf-if-te-212/4/10)# switchport access vlan 100
sw0(conf-if-te-212/4/10)#no shutdown
sw0(conf-if-te-212/4/10)#exit

sw0(config)# interface TenGigabitEthernet 212/4/11
sw0(conf-if-te-212/4/11)# switchport
sw0(conf-if-te-212/4/11)# switchport mode trunk
sw0(conf-if-te-212/4/11)# switchport trunk allowed vlan add 101-102
sw0(conf-if-te-212/4/11)# switchport trunk tag native-vlan
sw0(conf-if-te-212/4/11)#no shutdown
sw0(conf-if-te-212/4/11)#exit

sw0(config)# interface TenGigabitEthernet 212/4/12
sw0(conf-if-te-212/4/12)#port-profile-port
sw0(conf-if-te-212/4/12)#no shutdown
sw0(conf-if-te-212/4/12)#exit

VDX device connection to Slave Node 2:

sw0(config)# interface TenGigabitEthernet 212/4/7
sw0(conf-if-te-212/4/7)# switchport
sw0(conf-if-te-212/4/7)# switchport mode access
sw0(conf-if-te-212/4/7)# switchport access vlan 100
sw0(conf-if-te-212/4/7)#no shutdown
sw0(conf-if-te-212/4/7)#exit

sw0(config)# interface TenGigabitEthernet 212/4/8
sw0(conf-if-te-212/4/8)# switchport
sw0(conf-if-te-212/4/8)# switchport mode trunk
sw0(conf-if-te-212/4/8)# switchport trunk allowed vlan add 101-102
sw0(conf-if-te-212/4/8)# switchport trunk tag native-vlan
sw0(conf-if-te-212/4/8)#no shutdown
sw0(conf-if-te-212/4/8)#exit

sw0(config)# interface TenGigabitEthernet 212/4/9
sw0(conf-if-te-212/4/9)#port-profile-port
sw0(conf-if-te-212/4/9)#no shutdown
sw0(conf-if-te-212/4/9)#exit
**Eth0 Configurations:**

VCS ports which are connected to eth0 (Admin PXE) of the servers should be configured with Access mode configurations as below – verify it by the command

```
sw0#show running-config interface TenGigabitEthernet 212/4/4
interface TenGigabitEthernet 212/4/4
  fabric isl enable
  fabric trunk enable
  switchport
  switchport mode access
  switchport access vlan 100
  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 – verify it by the command

```
sw0#show running-config interface TenGigabitEthernet 212/4/5
interface TenGigabitEthernet 212/4/5
  fabric isl enable
  fabric trunk enable
  switchport
  switchport mode trunk
  switchport trunk allowed vlan add 101-102
  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- verify it by the command

```
sw0#show running-config interface TenGigabitEthernet 212/4/6
interface TenGigabitEthernet 212/4/6
  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
```
2. Make sure nodes are detected in Master Node. Click on ‘Add Nodes’, to assign the Roles to the discovered nodes.

3. Public, Admin, Private, Storage and Management networks can be associated to their respective Ethernet interfaces using ‘Configure Interfaces’ option.
4. Verify Networks tab is populated with correct values

**Network Settings**

*Neutron with VLAN segmentation*

<table>
<thead>
<tr>
<th>Setting</th>
<th>Start</th>
<th>End</th>
</tr>
</thead>
<tbody>
<tr>
<td>IP Range</td>
<td>10.24.41.2</td>
<td>10.24.41.126</td>
</tr>
<tr>
<td>CIDR</td>
<td>10.24.40.0/22</td>
<td></td>
</tr>
<tr>
<td>Use VLAN tagging</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Gateway</td>
<td>10.24.40.1</td>
<td></td>
</tr>
</tbody>
</table>

**Neutron L2 Configuration**

<table>
<thead>
<tr>
<th>Setting</th>
<th>Start</th>
<th>End</th>
</tr>
</thead>
<tbody>
<tr>
<td>VLAN ID range</td>
<td>103</td>
<td>103</td>
</tr>
<tr>
<td>Base MAC address</td>
<td>fa:16:3e:00:00:00</td>
<td></td>
</tr>
</tbody>
</table>

**Neutron L3 Configuration**

<table>
<thead>
<tr>
<th>Setting</th>
<th>Start</th>
<th>End</th>
</tr>
</thead>
<tbody>
<tr>
<td>Internal network CIDR</td>
<td>192.168.111.0/24</td>
<td></td>
</tr>
<tr>
<td>Internal network gateway</td>
<td>192.168.111.1</td>
<td></td>
</tr>
<tr>
<td>Floating IP ranges</td>
<td>10.24.41.130</td>
<td>10.24.41.254</td>
</tr>
<tr>
<td>DNS Servers</td>
<td>8.8.4.4</td>
<td>8.8.8.8</td>
</tr>
</tbody>
</table>

5. Horizon credentials can be configured in Master node as –

**OpenStack Settings**

**Access**

<table>
<thead>
<tr>
<th>Setting</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>username</td>
<td>admin</td>
<td>Username for Administrator</td>
</tr>
<tr>
<td>password</td>
<td>••••</td>
<td>Password for Administrator</td>
</tr>
<tr>
<td>tenant</td>
<td>admin</td>
<td>Tenant (project) name for Administrator</td>
</tr>
<tr>
<td>email</td>
<td><a href="mailto:admin@example.org">admin@example.org</a></td>
<td>Email address for Administrator</td>
</tr>
</tbody>
</table>
6. Public network can be assigned to all node as

Public network assignment

- Assign public network to all nodes
- When disabled, public network will be assigned to controllers and zabbix-server only

7. Click on ‘Deploy Changes’ button

Configuration Check – Brocade VCS Plugin code and ncclient:

1. 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

2. 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.grnet.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
[database]
connection =
mysql://neutron:password@192.168.0.3:3306/neutron_ml2?read_timeout=60

/etc/neutron/plugins/openvswitch/ovs_neutron_plugin.ini (Only on the compute nodes)
[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, flat, local
type_drivers = vlan, flat, local
mechanism_drivers = openvswitch, brocade

[ml2_brocade]
# VCS cluster credential
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 -

```bash
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
SVI - L3 Networking driver.

This section describes how SVI feature can be leveraged to provide internetworking between networks configured using OpenStack.

Edit /etc/neutron/neutron.conf

```
service_plugins =
neutron.services.l3_router.brocade.l3_router_plugin.BrocadeSVIPlugin
```

A new field has been added to the existing fields in brocade.ini file.

Add the below configuration in both /etc/neutron/plugins/ml2/ml2_conf.ini and /etc/neutron/plugins/ml2/ml2_conf_brocade.ini

```
rbridge_id = <rbridge id of vcs device>
```

This field indicates the Rbridge on which the SVI interfaces would get created.
### Test Report

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

<table>
<thead>
<tr>
<th>S.No</th>
<th>Test Case Title</th>
<th>Test Case Description</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Create Network</td>
<td>Verify create network is successful and VLAN is created on VCS device</td>
<td>Pass</td>
</tr>
<tr>
<td>2</td>
<td>Create Multiple Networks</td>
<td>Verify multiple VLANs are created on the VCS device</td>
<td>Pass</td>
</tr>
<tr>
<td>3</td>
<td>Delete Network</td>
<td>Verify the VLAN is deleted from the VCS device</td>
<td>Pass</td>
</tr>
<tr>
<td>4</td>
<td>Delete Multiple Networks</td>
<td>Verify multiple VLANs are deleted from the VCS device</td>
<td>Pass</td>
</tr>
<tr>
<td>5</td>
<td>Create Network when VLAN range exceeds</td>
<td>Verify VLANs are not created on VCS devices</td>
<td>Pass</td>
</tr>
<tr>
<td>6</td>
<td>Launch instance</td>
<td>Verify Create Instance is successful&lt;br&gt;• Verify the MAC address assigned to the VM by launching VM console&lt;br&gt;• Verify the same MAC address is listing in MAC address table</td>
<td>Pass</td>
</tr>
<tr>
<td>7</td>
<td>Create Multiple Instances</td>
<td>Verify Create Instances are successful&lt;br&gt;• Verify the MAC addresses assigned to the VMs by launching VM console&lt;br&gt;• Verify the multiple MAC addresses are listing in MAC address table</td>
<td>Pass</td>
</tr>
<tr>
<td>8</td>
<td>Delete instance</td>
<td>Verify the MAC address of the deleted VM is not listing in MAC address table</td>
<td>Pass</td>
</tr>
<tr>
<td>9</td>
<td>Delete multiple instances</td>
<td>Verify the MAC addresses of the deleted VMs are not listing in MAC address table</td>
<td>Pass</td>
</tr>
<tr>
<td>10</td>
<td>Reboot of VCS Device</td>
<td>Verify even after the reboot of VCS device, created network (VLAN) and MAC addresses are listing properly</td>
<td>Pass</td>
</tr>
<tr>
<td>11</td>
<td>Ping between the VMs on same host.</td>
<td>Verify ping between the VMs present in same host is successful</td>
<td>Pass</td>
</tr>
<tr>
<td>12</td>
<td>Ping between the VMs on different host</td>
<td>Verify ping between the VMs present in different hosts is successful</td>
<td>Pass</td>
</tr>
<tr>
<td>13</td>
<td>Ping between the VMs on another Network.</td>
<td>Verify ping should not happen between VMs present in different networks</td>
<td>Pass</td>
</tr>
<tr>
<td>14</td>
<td>Create Duplicate Network with the existing subnet</td>
<td>Verify VLAN is not created on VCS devices</td>
<td>Pass</td>
</tr>
<tr>
<td>15</td>
<td>Disturb switch to server connection when Ping is happening between VMs.</td>
<td>Verify the following -&lt;br&gt;• Verify the packet is getting dropped and ping is not successful&lt;br&gt;• Verify the ping is getting successful again</td>
<td>Pass</td>
</tr>
<tr>
<td>16</td>
<td>Create Network and Create Port when switch is not reachable.</td>
<td>Verify the creation is getting failed</td>
<td>Pass</td>
</tr>
</tbody>
</table>
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 -
Check that required services are running
- 100 s. 1.9

Check internet connectivity from a compute
- 100 s. 0.2

Check DNS resolution on compute node
- 120 s. 0.4

Request list of networks
- 20 s. 0.1

**Functional tests, Duration 3 min - 14 min**

Create instance flavor
- 30 s. 0.2

Create volume and boot instance from it
- 350 s. 0.0

There are no cinder nodes or ceph storage for volume

Target component: Compute
Scenario:
1. Create a new small-size volume from image.
2. Wait for volume status to become "available".
3. Launch instance from created volume.
4. Wait for "Active" status.
5. Delete instance.
6. Delete volume.
7. Verify that volume deleted

Create volume and attach it to instance
- 350 s. 0.0

There are no cinder nodes or ceph storage for volume

Target component: Compute
Scenario:
1. Create a new small-size volume.
2. Wait for volume status to become "available".
3. Create new instance.
4. Wait for "Active" status.
5. Attach volume to an instance.
6. Check volume status is "in use".
7. Get information on the created volume by its id.
8. Detach volume from the instance.
9. Check volume has "available" status.
10. Delete volume.
11. Verify that volume deleted.
12. Delete server.

Check network connectivity from instance via floating IP
- 300 s. 0.7

Router can not be created, Please refer to OpenStack logs for more details.

Target component: Neutron
Scenario:
1. Create a new security group (if it doesn't exist yet).
2. Create router
3. Create network
4. Create subnet
5. Uplink subnet to router
6. Create an instance using the new security group
7. Create a new floating IP
8. Assign the new floating IP to the instance.
9. Check connectivity to the floating IP using ping command.
10. Check that public IP 8.8.8.8 can be pinged from instance.
11. Dissociate server floating ip.
12. Delete floating ip
15. Remove subnet.
16. Remove network.
Note:
Neutron network creation process might hang some times with netconf client timeout expiration error. Please refer the defect in the below location for more details
https://bugs.launchpad.net/neutron/+bug/1395976
Support Details

Customers with valid Mirantis 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](https://www.brocade.com/service-support/index.html)

**Brocade Direct Support SLA:** [http://www.brocade.com/services-support/support-plans/direct-support/index.page](http://www.brocade.com/services-support/support-plans/direct-support/index.page)