UCS RHEL 6 KVM and Palo Integration
# TABLE OF CONTENTS

1 INTRODUCTION .................................................................................................................. 3
2 TEST REQUIREMENTS ........................................................................................................ 3
3 TEST CASES ....................................................................................................................... 4
   3.1 Prerequisites and Caveats .............................................................................................. 4
   3.2 Creation of UCS Service Profile .................................................................................. 4
   3.3 Install RHEL 6.1 ......................................................................................................... 7
   3.4 Configure KVM ........................................................................................................... 9
   3.5 Setup UCSM to RHEL Integration .............................................................................. 10
   3.6 Verify Dynamic VNICS on Server ............................................................................. 12
   3.7 Create KVM Virtual Machines .................................................................................. 13
   3.8 Migrate the VM from NAT to Palo Dynamic VNIC ................................................... 16
   3.9 Start the VM and verify networking .......................................................................... 17
   3.10 Troubleshooting commands .................................................................................... 18
1 Introduction

The purpose of this document is to provide a step by step sample configuration on integrating UCS Dynamic VNICS with RHEL KVM hypervisor.

Similar to UCS integration with ESX it is now possible to present Dynamic VNICS to a Virtual Machine running on a KVM hypervisor. You get all the same benefits with KVM as you do with ESX. Improved network performance, network visibility, and enhanced network options can all be applied to a VM.

2 Test Requirements

- Software
  - UCS 2.x image
- Hardware Requirements
  - UCS Test system
    - At least one B series blade with a Cisco Palo card
  - RHEL 6 or 6.1 with KVM installed and enabled
3 Test Cases

3.1 Prerequisites and Caveats

Integration with KVM has only been tested with RHEL 6 and 6.1. It is highly recommended to use 6.1. Other linux distributions with KVM have not been tested.

UCS requires the 2.x code be loaded. All components in the UCS system should be upgraded the 2.x code. This includes the BIOS and Palo Adapters.

The UCS Service Profile needs to have Dynamic VNICs assigned. The number of Dynamic VNICs that can be assigned to a Service Profile is still tied to the number of Virtual Interfaces (VIFs) that can be supported per blade slot. The number of VIFs is still based off the number of IOM to Fabric Interconnect Uplinks. That equation is as follows

\[(\text{# of Uplinks} \times 15) - 2\]

This means the number of VIFs you can support with 1 uplink is 13, 2 uplinks is 28, and 4 uplinks is 58.

The running OS still needs to see at least one real VNIC for network connectivity though it is highly recommended to have two. One VNIC on fabric A and one VNIC on fabric B.

If you assume a minimum of 2 VNICs this means you can support a maximum number of Dynamic VNICs of 11 with 1 uplink, 26 with 2 uplinks, and 56 with 4 uplinks.

If you create virtual Fibre Channel interfaces that also requires a VIF and should be subtracted from the total number of available VIFs to create a dynamic VNIC from. The full equation is as follows

\[((\text{# of Uplinks} \times 15) - 2) - \#\text{VNICs} - \#\text{VFCs} = \#\text{of dynamic VNICs that can be created}\]

3.2 Creation of UCS Service Profile

A Service Profile for the RHEL host should be created. It should contain at least 2 VNICs (one on each fabric) and if desired 2 vHBAs (Also one on each Fabric).

To start we will define only 10 VNICs but this number can be increased later depending on the number of IOM uplinks that are active.

Here we create the vHBAs. Note we create two vHBAs using WWN pool. One vHBA on each fabric.
Next we create the dynamic vNIC policy. Here we name the policy and create 10 dynamic vNICS.
Next create the two VNICS. One on fabric A and the other on Fabric B. Do NOT select Fabric Failover. Make sure to add the appropriate VLANs to each VNIC.
Finish the Service Profile creation and apply it to a blade that has a Cisco VIC/Palo card installed. Because Dynamic VNICS are defined the association will fail if you attempt to assign the Service Profile to a blade without a VIC/Palo card.

Here you can see the service profile assigned to a blade. Note how the Dynamic VNICS are load balanced across the A and B fabrics.

3.3 Install RHEL 6.1

Now that the Service Profile is associated, load RHEL 6.1 on the blade. Make sure to install the virtualization components when you install RHEL.

This testplan will not detail the full RHEL installation but will show where to select KVM installation.
When you get to the RHEL server installation type, make sure to select “Customize now” radio button and select “Next”
Also make sure to select Virtualization and select all the virtualization services.

RHEL will start installing after you select next.

### 3.4 Configure KVM

Depending on your KVM configuration you may or may not need to modify qemu.conf file. In this testplan everything is done by root so we need to make a few changes to allow KVM to work with NFS shared storage.

Edit qemu.conf

```
[root@rhel-palo2 ~]# cd /etc/libvirt/
[root@rhel-palo2 libvirt]# vi qemu.conf
```

Change the following lines

```plaintext
# The default security driver is SELinux. If SELinux is disabled
# on the host, then the security driver will automatically disable
# itself. If you wish to disable QEMU SELinux security driver while
# leaving SELinux enabled for the host in general, then set this
# to 'none' instead
#
```
security_driver = "none"

# The user ID for QEMU processes run by the system instance
user = "root"

# The group ID for QEMU processes run by the system instance
group = "root"

# Whether libvirt should dynamically change file ownership
# to match the configured user/group above. Defaults to 1.
# Set to 0 to disable file ownership changes.
dynamic_ownership = 0

Now restart libvirtd with the following command
[root@rhel-palo2 libvirt]# service libvirtd restart
Stopping libvirtd daemon: [ OK ]
Starting libvirtd daemon: [ OK ]

3.5 Setup UCSM to RHEL Integration

In this step we will setup the integration between UCSM and the RHEL host. The setup between UCSM and RHEL is a little different than UCSM and VMware vCenter. Since RHEL does not have a vCenter equivalent we only need to create a port-profile and port-profile client. There is nothing else to create.
Create the port-profile in UCSM.

In UCSM browse to the VM tab, select “Port-profiles” from the tree and right click and create new port-profile.

Fill out the port-profile box to network settings you want a VM to have. In the example below we are creating an access port that will allow the VM to access vlan 10.

Note that most VMs expect a single VLAN to be presented. If you select multiple vlans you are essentially trunking to the VM.

Do not select “High Performance”. This option only works with VMware ESX 5.0.
Now that we have created the port-profile we need to create a port-profile client. The purpose of the client is to actually assign the port-profile to a host.

Select the port-profile, right click and select “create port-profile client”. Give the client a name and description and change the Distributed Virtual Switch to default. By setting it default you are telling UCSM that this is for KVM and not for VMware.

![Create Profile Client](image)

3.6 Verify Dynamic VNICs on Server

Linux sees dynamic VNICs just as it does regular VNICs. They show up as eth interfaces and can be viewable with “ifconfig” It can be difficult to tell which eth interfaces are dynamic and which are not. Generally Dynamic VNICs will not have MAC addresses assigned to them. So any eth interface without a MAC should be a Dynamic VNIC.

```
[root@rhel-palo2 log]# ifconfig -a | more
eth0   Link encap:Ethernet  HWaddr 00:25:B5:00:00:2E
       inet addr:172.18.217.195  Bcast:172.18.217.255
       Mask:255.255.255.0
       inet6 addr: fe80::225:b5ff:fe00:2e/64 Scope:Link
       UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
       RX packets:7054 errors:0 dropped:0 overruns:0 frame:0
       TX packets:737 errors:0 dropped:0 overruns:0 carrier:0
       collisions:0 txqueuelen:1000
       RX bytes:911661 (890.2 KiB)  TX bytes:240572 (234.9 KiB)
eth1   Link encap:Ethernet  HWaddr 00:25:B5:00:00:1F
       inet6 addr: fe80::225:b5ff:fe00:1f/64 Scope:Link
       UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
```
Above you can see that eth0 and eth1 both have MAC addresses that match the MAC addresses assigned through the Service Profile. They are the real VNICS. Eth2 does not have a MAC address at all so it is a Dynamic VNIC.

3.7 Create KVM Virtual Machines
This example will use Virtual Machine Manager GUI but you can create the VM via the CLI.

Start the VMM and create a Virtual Machine.
Choose your install media option.

![Create a new virtual machine](image)

Choose Memory and vCPU

![Create a new virtual machine](image)
Allocate the amount of disk space to use

Verify the settings and create the Virtual Machine. Currently VMM does not have the ability to assign Dynamic VNICs. We need to first create the VM and then change the settings.
3.8 Migrate the VM from NAT to Palo Dynamic VNIC

Once the OS is installed on the VM we migrate the VM from NAT based networking to an actual Dynamic VNIC hosted on the Palo card. Note that to do this we must modify the XML file that describes the VM we created. Currently this is the only way to move the VM to a Palo based Dynamic VNIC. You need to be comfortable with Linux CLI and editors.

Shutdown the VM.

Run the following command

```
[root@rhel-palo2 ~]# virsh list --all
```

<table>
<thead>
<tr>
<th>Id</th>
<th>Name</th>
<th>State</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>palo-vm-1</td>
<td>shut off</td>
</tr>
</tbody>
</table>

Now run `virsh edit “VM name”` of the VM you want to change

```
[root@rhel-palo2 ~]# virsh edit palo-vm-1
```

Move to the network section. You will have some lines that look like

```xml
<interface type='network'>
  <mac address='52:54:00:c1:17:0e'/>
  <source network='default'/>
  <model type='virtio'/>
  <address type='pci' domain='0x0000' bus='0x00' slot='0x03' function='0x0'/>
</interface>
```

We will edit this section to add
1. A MAC address – Currently you create the MAC. Use a common MAC scheme.
2. Port-profile – This should match the port-profile you created in UCSM
3. One of the dynamic eth interfaces – use one that has not been assigned. We will use eth8

Change the networking section to look like the following

```xml
<interface type='direct'>
  <mac address='00:16:36:84:14:ed'/>
  <source dev='eth8' mode='private'/>
  <virtualport type='802.1Qbh'>
    <parameters profileid='VM-vlan10'/>
  </virtualport>
  <model type='virtio'/>
  <address type='pci' domain='0x0000' bus='0x00' slot='0x03' function='0x0'/>
</interface>
```
Save the file and exit. If you get an error on exit go back and reedit the file and make sure there are no mistakes

3.9 Start the VM and verify networking

Start the VM and verify that networking works.

You can start the VM with virsh

```
[root@rtp-ucs-pod1-196 ~]# virsh start palo-vm-1
Domain palo-1 started
```

```
[root@rtp-ucs-pod1-196 ~]# virsh list --all

<table>
<thead>
<tr>
<th>Id</th>
<th>Name</th>
<th>State</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>rhel-palo2</td>
<td>running</td>
</tr>
<tr>
<td>2</td>
<td>rhel-palo3</td>
<td>running</td>
</tr>
<tr>
<td>4</td>
<td>palo-vm-1</td>
<td>running</td>
</tr>
</tbody>
</table>
```

You now need to get on the console of the VM using VMM or issuing the command virt-viewer palo-vm-1.

Once on the console login and verify that you have network connectivity.
Verify the eth interface you used for the VM now is up and has the MAC address you assigned.  
[root@rtp-ucs-pod1-196 ~]# ifconfig eth8
eth8      Link encap:Ethernet  HWaddr 00:16:36:84:14:ed
inet6 addr: fe80::6cfe:32ff:fee6:d45e/64 Scope:Link
UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
RX packets:1569 errors:0 dropped:0 overruns:0 frame:0
TX packets:18 errors:0 dropped:0 overruns:0 carrier:0
 collisions:0 txqueuelen:1000
RX bytes:143255 (139.8 KiB)  TX bytes:1368 (1.3 KiB)

Use ip link to verify  
[root@rtp-ucs-pod1-196 ~]# ip link show eth8
10: eth8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state
UP qlen 1000
       link/ether 00:16:36:84:14:ed brd ff:ff:ff:ff:ff:ff
ip link to verify as well

Note that the link state is up

3.10 Troubleshooting commands

From the CLI of UCSM

Use show service-profile circuit name to see which Dynamic VNICS are up and running from a UCSM standpoint. Look for VIFS that show as UP

Verify the port-profile config from NXOS
rtp-ucs-pod1-B(nxos)# show running-config port-profile

!Command: show running-config port-profile
!Time: Tue Jun 21 19:11:29 2011

version 5.0(3)N2(2.1)

port-profile type vethernet VM-vlan10
  switchport trunk native vlan 10
  switchport trunk allowed vlan 10
  switchport mode trunk
  pinning server pinning-failure link-down
  vmware port-group
  max-ports 64
  description vlan 10 connectivity for VM
  state enabled

port-profile type vethernet rhel-vlan10
  switchport trunk native vlan 10
  switchport trunk allowed vlan 10
  switchport mode trunk
  pinning server pinning-failure link-down
  vmware port-group
  max-ports 64
  state enabled

port-profile type vethernet vnic-A
  switchport trunk native vlan 2
  switchport trunk allowed vlan 1-2, 10, 150-155
  switchport mode trunk
  pinning server pinning-failure link-down
  vmware port-group
  max-ports 64
  state enabled

port-profile type vethernet vnic-B
  switchport trunk native vlan 1
  switchport trunk allowed vlan 1-2, 10, 150-155
  switchport mode trunk
  pinning server pinning-failure link-down
  vmware port-group
  max-ports 64
  state enabled

Use show system on NXOS

rtp-ucs-pod1-B(nxos)# show system internal vim info int Veth32770
interface Veth32770
  iod: 53
  if_index: 1c080010
  fsm_state: VIM_VETH_FSM_ST_CREATED
  num_attaches: 1
  detach_reason: never detached
  vim_down_reason: initializing
  curr_profile: rhel-vlan10
flags: veth=00000180, veth rt_info=0008
desc:
    pinning info:
        ds_id: (12)
        ds_port_uuid: 00:16:36:84:14:01 (3)
vNIC sam fixed: FALSE
vNIC ID: 0
hpp: 0
    connectee_mac_addr: 00:16:36:84:14:01
peer sync done : no
attach info:
    src: unknown
    status: active
    fsm_state: VIM_NIV_VIF_FSM_ST_UP_ACTIVE
    pp_alias: rhel-vlan10 (8)
        ds_id: (12)
        ds_port_uuid: 00:16:36:84:14:01 (3)
    conn_name:
    conn_dev:
    conn_uuid: ffff880660452ae1 (11)
    host_uuid: ffff880660452af1 (12)
    conn_mac_addr: 00:16:36:84:14:01