Cloud Training in a Box: Inexpensive and Reproducible Training Environments for Cloud Solutions

This Dell Technical White Paper provides a virtual environment for demonstrating, testing, and training on the use of Dell's deployment procedures for an OpenStack Cloud.

Dell Global Support and Deployment
Learning & Development

January 2012| Rev 1.0

Contents

Tables

Executive summary

Providing lab-based training on highly complex, multi-server solutions presents challenges.

Dell Global Support and Deployment Learning and Development sought to overcome these challenges by delivering training that:

- Provides a high student/lab station ratio
- Is not be prohibitively expensive
- Is reproducible for eventual use in regional training centers
- Is a model for training on a variety of Cloud Computing Solutions including OpenStack and Hadoop

In order to meet this need, Dell Global Support and Deployment leveraged existing available servers and the native virtualization capabilities of current Linux distributions to provide a 1:1 student/lab station ratio with no incremental cost to Dell. Additionally, this model is easily reproducible at any of Dell's training centers that have the capability of providing one virtualization-capable server per lab station.

The Challenge

When delivering hands-on lab-based training, a tension exists between the desire to provide every single student with the opportunity to perform the lab exercises and the need to control costs for hardware purchases. This tension exists even when training topics only require a single server per lab station. It is amplified when training solutions require multiple servers.

One response to this tension is to have students share a workstation and work together or take turns performing the lab exercises—but when forced to make that compromise, Dell knows from learner feedback that the "customer experience" is reduced. Also, the effectiveness of the training is likely reduced.

Dell needed to be able to deliver training on Dell Solution offerings of OpenStack and Hadoop. In order to replicate the minimal configuration of each solution's reference architecture, Dell needed to have at least six systems (and one or more network switches) in each lab station. Additionally, each reference architecture calls for different server models and for PowerEdge C-Series servers, which is often not as suitable for use in other training.

To provide each pair of students, in a class size of 8, with a lab station to share, requires 24 servers (48, if strictly following the reference architectures for those two solutions). These numbers reflect one implementation, which has to have remote access for classes in other regions. You have to double those server numbers if Dell wants to provide individual lab stations for each student.

Once Dell assembles this infrastructure, you can remotely access the infrastructure for classes in other regions, but it requires an administrative presence in the physical location in case the servers need attention. Many instructors have a strong preference for physical access to their class infrastructure in case something goes wrong during class—or at least to have confidence that someone will be available during their class hours, rather than having to wait for a remote administrator in another time zone.

Therefore, if Dell regional training centers prefer to have locally available lab infrastructure, Dell either has to ship hardware around the world, or buy more servers. Also, each training center has to have to have the skilled and available headcount, and the data-center space to set up this complex infrastructure.

Selecting a Platform

These considerations forced Dell Global Support and Deployment to consider the possibility of delivering these solutions using virtualization.

Is Virtualization a Possible Solution?

Two considerations persuaded Dell Global Support and Deployment that Dell can move to a virtualized solution:

- While elements of our OpenStack and Hadoop solutions offer enhanced functionality when deployed on PowerEdge C-series servers (such as the ability to update and configure BIOS and firmware), the basic functionality of each solution was hardware-agnostic within certain broad requirements. The solution can run, with compromises of some minor functionality, on virtual machines.
- While the reference architectures for these solutions call out specific Dell PowerEdge C-series servers, the target audience for training had already been trained on the support of those servers. These courses were only tasked with the training of the solutions and the deployment process for each; Support Services was not attempting to train the hardware as well.

Which Virtualization Platform?

Many virtualization platforms are currently available: VMware, HyperV, Xen, KVM, Virtual Box, and others. Dell Global Support and Deployment chose to go with the KVM virtualization that is natively available in recent Linux distributions for several reasons:

- The target audience is responsible for support of Linux/KVM virtualization as well, but gets less opportunity for hands-on experience with it than they do with VMware. By using KVM virtualization in these labs, learners gain experience in that skill set as well.
- Because the KVM tools are open source and exist in the package repositories native to each distribution, the deployment of this infrastructure is simplified and more easily scripted.
- Because the KVM tools are present in distributions that are free of charge (for example, CentOS) Dell Global Support and Deployment can easily document for customers how they can set up a similar testing/training environment at a very minimal cost.

What type of Remote Access?

Choosing a method of remote access to the virtualization hosts involved the following considerations:

- Because these solutions are built on top of a Linux OS, rather than MS Windows, access through RDP (Remote Desktop Protocol) is not a first choice.

- If students are going to access the systems from Linux-based client systems, then SSH with X-forwarding is an ideal solution; however, in most of Dell's classrooms, the existing student stations are running MS Windows.
- Access through a DRAC or BMC is a possibility (and remains so in the solution that Dell Global Support and Deployment ultimately chose—in case remote access is needed for access to system setup or other pre-OS configuration tasks) but often feels slow compared to more advanced remote access solutions.
- VNC (Virtual Network Computing) is a possibility, but generally does not offer secure access by default. Also, most of the VNC clients Dell considered required administrator privileges for installation, which can be a problem at some sites.

Ultimately, Dell chose to use the open source FreeNX server on the hosts, and to install the NoMachines nxclient (a free download) on the student systems, for the following reasons:

- It is encrypted and more tightly compressed than VNC.
- It offers secure access by default.
- The server can be installed from publicly-available repositories, making scripted installation easier.
- The client can be installed without requiring administrator privileges on the student systems.

What Hardware?

The hardware requirements are minimal. The Hadoop requirements are the more substantial of the two solutions and can be satisfied with 20 GB of disk space dedicated to the host, 20 GB of space for the virtual machine serving as the admin node for our cloud, and 40 GB of space for each of 5 other virtual machines. Each virtual machine functioned with 2 GB of RAM; Dell did not test for functionality at lower amounts because of the abundance of RAM in the available systems.

Dell Global Support and Deployment finds that each lab station can be provided with a single server that meets the following requirements:

- 16 GB of RAM
- 240 GB of storage
- Multicore CPUs with the VT extensions

These requirements are satisfied with a chassis of M605 blade servers already configured with RHEL6.

Implementation

Deployment of the training infrastructure involves the following steps:

1. Configuration of the Blade servers for remote access via their DRACs
2. Development of a Kickstart installation script (Appendix A: Sample Kickstart installation script) for the virtual hosts that accomplishes each of the following deployment tasks:
    - Installation of Red Hat Enterprise Linux 6 x86_64 with the appropriate virtualization packages configured by default

- Configuration of a local repository for post-install package installations
- Installation and configuration of SSH and the FreeNX server
- Download of the Cloud Solution software to be used
- Configuration of the virtual machines and their virtual storage
- Configuration of the isolated virtual networks to be used

3. Integration of the Blade Servers and the Kickstart scripts into Dell's local deployment Solution[1], enabling reinstallation with a simple PXE boot.

Once the initial deployment is complete, each server is remotely accessible to students through the NXclient or through the DRAC and is equipped to be a standalone lab station hosting multiple virtual machines in a cloud configuration.

Deployment of the solution software can be performed in either of two ways:

1. To fully simulate the field deployment process, an "Installer" VM is used temporarily to act as a PXE server for the installation of the admin node. This takes the place of the deployment laptop described in the *Crowbar Deployment Guide*[2] and allows students to simulate the process that will be used by Dell deployment teams.
2. The admin node can be configured for one-time boot from the Crowbar/OpenStack ISO. This installs the admin node OS and copies all necessary packages for the installation of Crowbar and the admin node services.

Outcome

Using this solution, Dell Global Support and Deployment is able to develop a combined course on OpenStack, Hadoop[3], and the Crowbar deployment tool used in both solutions. Support Services achieved this outcome without purchasing new hardware. This solution provided a reproducible model that Dell can use for future training in regional training centers globally.

After each class, the preparation process to refresh the solution completely for the next class takes approximately 45 minutes, but can be mostly unattended once the process is initiated.

Conclusion

This model is not applicable to all Dell courses because many of them are dependent on particular hardware requirements. But for courses that are focused on software training only, and on software that is largely hardware-agnostic, this project provides an incredibly cost-effective model for training

---

[1] This step is optional. Other sites may choose to deploy the Hosts using the Kickstart scripts as part of a manual installation instead of configuring a PXE server as we did.
[2] github.com/dellcloudedge/barclamp-crowbar/raw/master/crowbar_framework/public/crowbar_deployment_guide.pdf
[3] The Hadoop solution required some modification to this configuration and is not within the scope of this whitepaper.

complex solutions with better customer experience for our learners and more effective learning than a traditional hardware-based lab setup provides.

To Learn More

For more information on Dell Cloud Solutions, visit the following links:

- dell.com/cloud
- dell.com/crowbar

For the latest information on the development work that Dell teams are doing in the area of cloud computing, go to the following blogs:

- robhirschfeld.com
- bartongeorge.net

For information on or downloads of the latest open source release of Crowbar:

- github.com/dellcloudedge

White papers and spec sheets:

- content.dell.com/us/en/enterprise/by-need-it-productivity-data-center-change-response-openstack-cloud.aspx
- content.dell.com/us/en/enterprise/d/business~solutions~engineering-docs~en/Documents~crowbar-software-framework-specsheet.pdf.aspx
- content.dell.com/us/en/enterprise/d/business~solutions~whitepapers~en/Documents~cloud-ops-for-openstack.pdf.aspx
- content.dell.com/us/en/enterprise/spredir.ashx/business~solutions~whitepapers~en/Documents~revolutionary-approach-to-cloud-building.pdf
- content.dell.com/us/en/enterprise/spredir.ashx/business~solutions~whitepapers~en/Documents~private-and-public-clouds.pdf

Appendix A: Sample kickstart installation script

The kickstart file below illustrates some of the configuration tasks that were automated for easy redeployment. Some other tasks that were specific to our environment and would be neither generally applicable, nor generally needed, were removed. This is offered as an example only and would need to be modified by a knowledgeable administrator before use in another environment.

Table 1. Sample kickstart file

```
#################################################################
# Kickstart Deployment file for Cloud Training Lab Stations
#################################################################

# General Installation Options #
install
url --url=http://10.10.10.1/cobbler/ks_mirror/RHEL6Server_64
vnc
xconfig  --startxonboot
lang en_US.UTF-8
keyboard us
timezone --utc America/Chicago
network --bootproto static --ip $ip_address_eth0 --netmask 255.255.240.0 --gateway 10.10.10.254 \
   --hostname $hostname --nameserver $name_servers
authconfig --enableshadow --passalgo=sha512 --enablefingerprint
rootpw  --iscrypted $1$MCuMJ0$yn5m0QKtVeBdZeQrKqnQx.
firewall --disabled
selinux --permissive
bootloader --location=mbr  --driveorder=sdb --append="crashkernel=auto rhgb quiet"
clearpart --drives=sdb --initlabel
part /boot --fstype=ext4 --size=500
part pv.syspv --grow --size=1
volgroup sysvg --pesize=4096 pv.syspv
logvol / --fstype=ext4 --name=lv_root --vgname=sysvg --size=16440
logvol swap --name=lv_swap --vgname=sysvg --size=16436
firstboot --disabled
reboot

repo --name="Red Hat Enterprise Linux"  --
baseurl=http://10.10.10.1/cobbler/ks_mirror/RHEL6Server_64/ --cost=100
repo --name="High Availability"  --
baseurl=http://10.10.10.1/cobbler/ks_mirror/RHEL6Server_64/HighAvailability/ --cost=1000
repo --name="Resilient Storage"  --
baseurl=http://10.10.10.1/cobbler/ks_mirror/RHEL6Server_64/ResilientStorage/ --cost=1000
repo --name="Scalable Filesystem Support"  --
baseurl=http://10.10.10.1/cobbler/ks_mirror/RHEL6Server_64/ScalableFileSystem/ --cost=1000

key --skip

# Package Selection #
%packages
@base
@console-internet
@core
@debugging
@basic-desktop
@desktop-platform
@directory-client
@general-desktop
@hardware-monitoring
@ha
@ha-management
@java-platform
@large-systems
@network-file-system-client
@performance
@perl-runtime
@resilient-storage
```

```
@scalable-file-systems
@server-platform
@server-policy
@virtualization
@virtualization-client
@virtualization-platform
@virtualization-tools
@x11
mtools
pax
python-dmidecode
oddjob
sgpio
genisoimage
wodim
certmonger
pam_krb5
krb5-workstation
nscd
pam_ldap
nss-pam-ldapd
sg3_utils
perl-DBD-SQLite
expect
xorg-x11-apps
xorg-x11-fonts-misc
audiofile
libXpm
tcl
libXaw
firefox
lftp

# Post-Install Script #
%post

# Obtain and install a configuration file for a YUM repository valid at my location
wget -O /etc/yum.repos.d/rhel6-dell.repo ftp://10.10.10.1/pub/rhel6-dell.repo

# Optional – For remote access, install and configure freenx server
rpm -ivh ftp://10.10.10.1/pub/freenx-0.7.3-6.el6.ay.x86_64.rpm ftp://10.10.10.1/pub/nx-3.4.0-
6.el6.ay.x86_64.rpm
nxsetup --install --auto --setup-nomachine-key
nxsetup --test >> /root/nxsetup.results.txt

# Create a script to prep the system for the Cloud Solutions Class
cat > /root/CloudSolutionsPostInstallPrep.sh <<EOF1
#!/bin/bash

# Get the files needed.
mkdir /CloudSolutions/
cd /CloudSolutions/
wget  ftp://10.10.10.1/pub/CloudSolutions/*.xml
wget -O /var/lib/libvirt/images/openstack-0.9.5-9.531.iso \
   ftp://10.10.10.1/pub/CloudSolutions/openstack-0.9.5-9.531.iso
wget ftp://10.10.10.1/pub/CloudSolutions/openstack_installer.tgz

# Destroy and reconfigure the default networking for the virtual machines
virsh net-destroy default
virsh net-undefine default
virsh net-define defaultnet.xml
virsh net-start default
virsh net-autostart default

# Define the Public Network for the virtual cloud
virsh net-define publicnet.xml
virsh net-start public
virsh net-autostart public

# Define the Private Network for the virtual cloud
virsh net-define privatenet.xml
```

```
virsh net-start private
virsh net-autostart private

# Optional – Convert the Installer VM from a VMware image to a KVM image
tar xzvf openstack_installer.tgz
cd Openstack\ Installer/
virt-convert Openstack\ Installer.vmx vmdef.OpenstackInstaller.xml
mv Openstack_Installer.raw /var/lib/libvirt/images/

# Define a default storage pool
virsh pool-create-as default dir --target /var/lib/libvirt/images

# Create the disk images for the VMs
virsh vol-create-as default admin-hda.img 10G --allocation 256k
virsh vol-create-as default node1hda.img 10G --allocation 256k
virsh vol-create-as default node1hdb.img 10G --allocation 256k
virsh vol-create-as default node2hda.img 10G --allocation 256k
virsh vol-create-as default node2hdb.img 10G --allocation 256k

# Define the VMs from .xml files
virsh define vmdef.installer.xml
virsh define vmdef.admin.xml
virsh define vmdef.node1.xml
virsh define vmdef.node2.xml

EOF1

# backup the original rc.local
cp /etc/rc.local /root/rc.local

# add CloudSolutionsPostInstallPrep.sh to rc.local
cat >> /etc/rc.local <<EOF
if [ -f /root/CloudSolutionsPostInstallPrep.sh ]; then
cd /root
bash /root/CloudSolutionsPostInstallPrep.sh
mv /root/CloudSolutionsPostInstallPrep.sh /root/CloudSolutionsPostInstallPrep.sh.executed
fi

EOF
```

Appendix B: Network definition files

The files below are used in the kickstart file to define the desired network configuration for a virtual environment:

Table 2. defaultnet.xml: Revised definition for the default network

```
<network>
  <name>default</name>
  <forward mode='nat'/>
  <bridge name='virbr0' stp='on' delay='0' />
  <ip address='192.168.120.1' netmask='255.255.255.0'>
    <dhcp>
      <range start='192.168.120.2' end='192.168.120.254' />
    </dhcp>
  </ip>
</network>
```

Table 3. publicnet.xml: Definition for the public network

```
<network>
   <name>public</name>
   <bridge name="virbr2" />
   <ip address="192.168.123.1" netmask="255.255.255.0">
   </ip>
</network>
```

Table 4. privatenet.xml: Definition for the private network

```
<network>
   <name>private</name>
   <bridge name="virbr1" />
   <ip address="192.168.124.1" netmask="255.255.255.0">
   </ip>
</network>
```

Appendix C: Virtual machine definition files

The files below are used in the kickstart file to define the virtual machines used in the cloud solution:

Table 5. vmdef.admin.xml: Definition file for the Admin Node

```
<domain type='kvm'>
  <name>admin</name>
  <memory>2097152</memory>
  <currentMemory>2097152</currentMemory>
  <vcpu>2</vcpu>
  <os>
    <type arch='x86_64' machine='rhel6.0.0'>hvm</type>
    <boot dev='hd'/>
    <bootmenu enable='yes'/>
  </os>
  <features>
    <acpi/>
    <apic/>
    <pae/>
  </features>
  <clock offset='utc'/>
  <on_poweroff>destroy</on_poweroff>
  <on_reboot>restart</on_reboot>
  <on_crash>restart</on_crash>
  <devices>
    <emulator>/usr/libexec/qemu-kvm</emulator>
    <disk type='file' device='disk'>
      <driver name='qemu' type='raw' cache='none'/>
      <source file='/var/lib/libvirt/images/admin-hda.img'/>
      <target dev='hda' bus='ide'/>
      <alias name='ide0-0-0'/>
      <address type='drive' controller='0' bus='0' unit='0'/>
    </disk>
    <disk type='block' device='cdrom'>
      <driver name='qemu' type='raw'/>
      <target dev='hdc' bus='ide'/>
      <readonly/>
      <alias name='ide0-1-0'/>
      <address type='drive' controller='0' bus='1' unit='0'/>
    </disk>
    <controller type='ide' index='0'>
      <alias name='ide0'/>
      <address type='pci' domain='0x0000' bus='0x00' slot='0x01' function='0x1'/>
    </controller>
    <interface type='network'>
      <source network='private'/>
      <target dev='vnet0'/>
      <model type='e1000'/>
      <alias name='net0'/>
      <address type='pci' domain='0x0000' bus='0x00' slot='0x03' function='0x0'/>
    </interface>
    <interface type='network'>
      <source network='public'/>
      <target dev='vnet1'/>
      <model type='e1000'/>
      <alias name='net1'/>
      <address type='pci' domain='0x0000' bus='0x00' slot='0x04' function='0x0'/>
    </interface>
    <serial type='pty'>
      <source path='/dev/pts/2'/>
      <target port='0'/>
      <alias name='serial0'/>
    </serial>
    <console type='pty' tty='/dev/pts/2'>
      <source path='/dev/pts/2'/>
      <target type='serial' port='0'/>
      <alias name='serial0'/>
    </console>
```

```
    <input type='tablet' bus='usb'>
      <alias name='input0'/>
    </input>
    <input type='mouse' bus='ps2'/>
    <graphics type='vnc' port='5900' autoport='yes'/>
    <sound model='ac97'>
      <alias name='sound0'/>
      <address type='pci' domain='0x0000' bus='0x00' slot='0x05' function='0x0'/>
    </sound>
    <video>
      <model type='cirrus' vram='9216' heads='1'/>
      <alias name='video0'/>
      <address type='pci' domain='0x0000' bus='0x00' slot='0x02' function='0x0'/>
    </video>
    <memballoon model='virtio'>
      <alias name='balloon0'/>
      <address type='pci' domain='0x0000' bus='0x00' slot='0x06' function='0x0'/>
    </memballoon>
  </devices>
  <seclabel type='dynamic' model='selinux'>
    <label>system_u:system_r:svirt_t:s0:c25,c77</label>
    <imagelabel>system_u:object_r:svirt_image_t:s0:c25,c77</imagelabel>
  </seclabel>
</domain>
```

Table 6. vmdef.node1.xml: Definition file for the first Compute Node

```
<domain type='kvm'>
  <name>node1</name>
  <memory>1048576</memory>
  <currentMemory>1048576</currentMemory>
  <vcpu>2</vcpu>
  <os>
    <type arch='x86_64' machine='rhel6.0.0'>hvm</type>
    <boot dev='network'/>
    <boot dev='hd'/>
  </os>
  <features>
    <acpi/>
    <apic/>
    <pae/>
  </features>
  <clock offset='utc'/>
  <on_poweroff>destroy</on_poweroff>
  <on_reboot>restart</on_reboot>
  <on_crash>restart</on_crash>
  <devices>
    <emulator>/usr/libexec/qemu-kvm</emulator>
    <disk type='file' device='disk'>
      <driver name='qemu' type='raw' cache='none'/>
      <source file='/var/lib/libvirt/images/node1hda.img'/>
      <target dev='hda' bus='ide'/>
      <alias name='ide0-0-0'/>
      <address type='drive' controller='0' bus='0' unit='0'/>
    </disk>
    <disk type='file' device='disk'>
      <driver name='qemu' type='raw'/>
      <source file='/var/lib/libvirt/images/node1hdb.img'/>
      <target dev='vda' bus='virtio'/>
      <alias name='virtio-disk0'/>
      <address type='pci' domain='0x0000' bus='0x00' slot='0x07' function='0x0'/>
    </disk>
    <controller type='ide' index='0'>
      <alias name='ide0'/>
      <address type='pci' domain='0x0000' bus='0x00' slot='0x01' function='0x1'/>
    </controller>
    <interface type='network'>
      <source network='private'/>
```

```
      <target dev='vnet6'/>
      <model type='e1000'/>
      <alias name='net0'/>
      <address type='pci' domain='0x0000' bus='0x00' slot='0x03' function='0x0'/>
    </interface>
    <interface type='network'>
      <source network='public'/>
      <target dev='vnet7'/>
      <model type='e1000'/>
      <alias name='net1'/>
      <address type='pci' domain='0x0000' bus='0x00' slot='0x06' function='0x0'/>
    </interface>
    <serial type='pty'>
      <source path='/dev/pts/6'/>
      <target port='0'/>
      <alias name='serial0'/>
    </serial>
    <console type='pty' tty='/dev/pts/6'>
      <source path='/dev/pts/6'/>
      <target type='serial' port='0'/>
      <alias name='serial0'/>
    </console>
    <input type='mouse' bus='ps2'/>
    <graphics type='vnc' port='5903' autoport='yes'/>
    <sound model='ac97'>
      <alias name='sound0'/>
      <address type='pci' domain='0x0000' bus='0x00' slot='0x04' function='0x0'/>
    </sound>
    <video>
      <model type='cirrus' vram='9216' heads='1'/>
      <alias name='video0'/>
      <address type='pci' domain='0x0000' bus='0x00' slot='0x02' function='0x0'/>
    </video>
    <memballoon model='virtio'>
      <alias name='balloon0'/>
      <address type='pci' domain='0x0000' bus='0x00' slot='0x05' function='0x0'/>
    </memballoon>
  </devices>
  <seclabel type='dynamic' model='selinux'>
    <label>system_u:system_r:svirt_t:s0:c412,c512</label>
    <imagelabel>system_u:object_r:svirt_image_t:s0:c412,c512</imagelabel>
  </seclabel>
</domain>
```

Table 7. vmdef.node2.xml: Definition file for the second Compute Node

```
<domain type='kvm'>
  <name>node2</name>
  <memory>1048576</memory>
  <currentMemory>1048576</currentMemory>
  <vcpu>2</vcpu>
  <os>
    <type arch='x86_64' machine='rhel6.0.0'>hvm</type>
    <boot dev='network'/>
    <boot dev='hd'/>
  </os>
  <features>
    <acpi/>
    <apic/>
    <pae/>
  </features>
  <clock offset='utc'/>
  <on_poweroff>destroy</on_poweroff>
  <on_reboot>restart</on_reboot>
  <on_crash>restart</on_crash>
  <devices>
    <emulator>/usr/libexec/qemu-kvm</emulator>
```

```
    <disk type='file' device='disk'>
      <driver name='qemu' type='raw' cache='none'/>
      <source file='/var/lib/libvirt/images/node2hda.img'/>
      <target dev='hda' bus='ide'/>
      <alias name='ide0-0-0'/>
      <address type='drive' controller='0' bus='0' unit='0'/>
    </disk>
    <disk type='file' device='disk'>
      <driver name='qemu' type='raw'/>
      <source file='/var/lib/libvirt/images/node2hdb.img'/>
      <target dev='vda' bus='virtio'/>
      <alias name='virtio-disk0'/>
      <address type='pci' domain='0x0000' bus='0x00' slot='0x07' function='0x0'/>
    </disk>
    <controller type='ide' index='0'>
      <alias name='ide0'/>
      <address type='pci' domain='0x0000' bus='0x00' slot='0x01' function='0x1'/>
    </controller>
    <interface type='network'>
      <source network='private'/>
      <target dev='vnet4'/>
      <model type='e1000'/>
      <alias name='net0'/>
      <address type='pci' domain='0x0000' bus='0x00' slot='0x03' function='0x0'/>
    </interface>
    <interface type='network'>
      <source network='public'/>
      <target dev='vnet5'/>
      <model type='e1000'/>
      <alias name='net1'/>
      <address type='pci' domain='0x0000' bus='0x00' slot='0x06' function='0x0'/>
    </interface>
    <serial type='pty'>
      <source path='/dev/pts/5'/>
      <target port='0'/>
      <alias name='serial0'/>
    </serial>
    <console type='pty' tty='/dev/pts/5'>
      <source path='/dev/pts/5'/>
      <target type='serial' port='0'/>
      <alias name='serial0'/>
    </console>
    <input type='mouse' bus='ps2'/>
    <graphics type='vnc' port='5902' autoport='yes'/>
    <sound model='ac97'>
      <alias name='sound0'/>
      <address type='pci' domain='0x0000' bus='0x00' slot='0x04' function='0x0'/>
    </sound>
    <video>
      <model type='cirrus' vram='9216' heads='1'/>
      <alias name='video0'/>
      <address type='pci' domain='0x0000' bus='0x00' slot='0x02' function='0x0'/>
    </video>
    <memballoon model='virtio'>
      <alias name='balloon0'/>
      <address type='pci' domain='0x0000' bus='0x00' slot='0x05' function='0x0'/>
    </memballoon>
  </devices>
  <seclabel type='dynamic' model='selinux'>
    <label>system_u:system_r:svirt_t:s0:c616,c758</label>
    <imagelabel>system_u:object_r:svirt_image_t:s0:c616,c758</imagelabel>
  </seclabel>
</domain>
```

Table 8. vmdef.installer.xml: Definition file for the optional installer VM

```
<domain type='kvm'>
  <name>Openstack_Installer</name>
  <memory>524288</memory>
  <currentMemory>524288</currentMemory>
  <vcpu>2</vcpu>
  <os>
    <type arch='x86_64' machine='rhel6.0.0'>hvm</type>
    <boot dev='hd'/>
  </os>
  <features>
    <acpi/>
    <apic/>
    <pae/>
  </features>
  <clock offset='utc'/>
  <on_poweroff>destroy</on_poweroff>
  <on_reboot>restart</on_reboot>
  <on_crash>restart</on_crash>
  <devices>
    <emulator>/usr/libexec/qemu-kvm</emulator>
    <disk type='file' device='disk'>
      <driver name='qemu' type='raw' cache='none'/>
      <source file='/var/lib/libvirt/images/Openstack_Installer.raw'/>
      <target dev='hda' bus='ide'/>
      <alias name='ide0-0-0'/>
      <address type='drive' controller='0' bus='0' unit='0'/>
    </disk>
    <disk type='file' device='cdrom'>
      <driver name='qemu' type='raw'/>
      <source file='/var/lib/libvirt/images/openstack-0.9.5-9.531.iso'/>
      <target dev='hdc' bus='ide'/>
      <readonly/>
      <alias name='ide0-1-0'/>
      <address type='drive' controller='0' bus='1' unit='0'/>
    </disk>
    <controller type='ide' index='0'>
      <alias name='ide0'/>
      <address type='pci' domain='0x0000' bus='0x00' slot='0x01' function='0x1'/>
    </controller>
    <interface type='network'>
      <source network='private'/>
      <target dev='vnet0'/>
      <model type='e1000'/>
      <alias name='net0'/>
      <address type='pci' domain='0x0000' bus='0x00' slot='0x03' function='0x0'/>
    </interface>
    <interface type='network'>
      <source network='public'/>
      <target dev='vnet1'/>
      <model type='e1000'/>
      <alias name='net1'/>
      <address type='pci' domain='0x0000' bus='0x00' slot='0x04' function='0x0'/>
    </interface>
    <serial type='pty'>
      <source path='/dev/pts/2'/>
      <target port='0'/>
      <alias name='serial0'/>
    </serial>
    <console type='pty' tty='/dev/pts/2'>
      <source path='/dev/pts/2'/>
      <target type='serial' port='0'/>
      <alias name='serial0'/>
    </console>
    <input type='tablet' bus='usb'>
      <alias name='input0'/>
    </input>
    <input type='mouse' bus='ps2'/>
    <graphics type='vnc' port='5900' autoport='yes'/>
```

```
    <sound model='ac97'>
      <alias name='sound0'/>
      <address type='pci' domain='0x0000' bus='0x00' slot='0x05' function='0x0'/>
    </sound>
    <video>
      <model type='cirrus' vram='9216' heads='1'/>
      <alias name='video0'/>
      <address type='pci' domain='0x0000' bus='0x00' slot='0x02' function='0x0'/>
    </video>
    <memballoon model='virtio'>
      <alias name='balloon0'/>
      <address type='pci' domain='0x0000' bus='0x00' slot='0x06' function='0x0'/>
    </memballoon>
  </devices>
  <seclabel type='dynamic' model='selinux'>
    <label>system_u:system_r:svirt_t:s0:c25,c77</label>
    <imagelabel>system_u:object_r:svirt_image_t:s0:c25,c77</imagelabel>
  </seclabel>
</domain>
```