Technical Report

Red Hat Enterprise Linux 6, KVM, and NetApp Storage: Deployment Guide

Jon Benedict, NetApp

April 2012 | TR-4034

SUMMARY

NetApp® technology enables data centers to extend their virtual infrastructures to include the benefits of advanced storage virtualization. NetApp unified storage platforms provide industry-leading technologies in the areas of storage efficiencies, instantaneous virtual machine (VM) and datastore cloning for virtual servers, and virtual data center backup and business continuance solutions.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1 RHEL KVM on NetApp

1.1 Overview

NetApp® technology enables data centers to extend their virtual infrastructures to include the benefits of advanced storage virtualization. NetApp unified storage platforms provide industry-leading technologies in the areas of storage efficiencies, instantaneous virtual machine (VM) and datastore cloning for virtual servers, and virtual data center backup and business continuance solutions.

Red Hat, although a relative newcomer to the virtualization market, provides support for the high-performance and open source Kernel-Based Virtual Machine (KVM) hypervisor.

Red Hat Enterprise Linux (RHEL) also offers the benefits of flexible deployment:

- It can be deployed as a bare-metal operating system, hypervisor, or virtual guest operating system.
- From a storage perspective, RHEL KVM supports both storage area network (SAN: iSCSI, Fibre Channel [FC], Fibre Channel over Ethernet [FCoE]) and network-attached storage (NAS: Network File System [NFS]) for shared virtual machine storage.

NetApp and Red Hat maintain a long-term strategic alliance that includes end-to-end solution testing between Red Hat products and NetApp storage. As a result of this testing, NetApp has developed operational guidelines and best practices for storage arrays running NetApp Data ONTAP® in support of Red Hat Enterprise Linux®. These guidelines have been extended to include RHEL-based KVM virtualization.

The following sections provide the steps necessary to deploy RHEL 6, KVM, and NetApp storage together in accordance with TR-3848: RHEL 6, KVM, and NetApp Storage: Best Practices Guide. The steps provided are specific to where Red Hat and NetApp technologies intersect. That is, because Red Hat and NetApp provide excellent product documentation, this document focuses on topics that are different and not covered by existing documents.

1.2 Intended Audience

This document addresses the needs of system architects, system administrators, and storage administrators who deploy KVM with RHEL 6 on NetApp storage.

1.3 Topics Out of Scope

Deployment steps associated with IP and Fibre Channel networks are not covered in this document. However, a good understanding of these topics is necessary for properly configuring items such as virtual local area networks (VLANs), switched fabrics, and other related technologies.

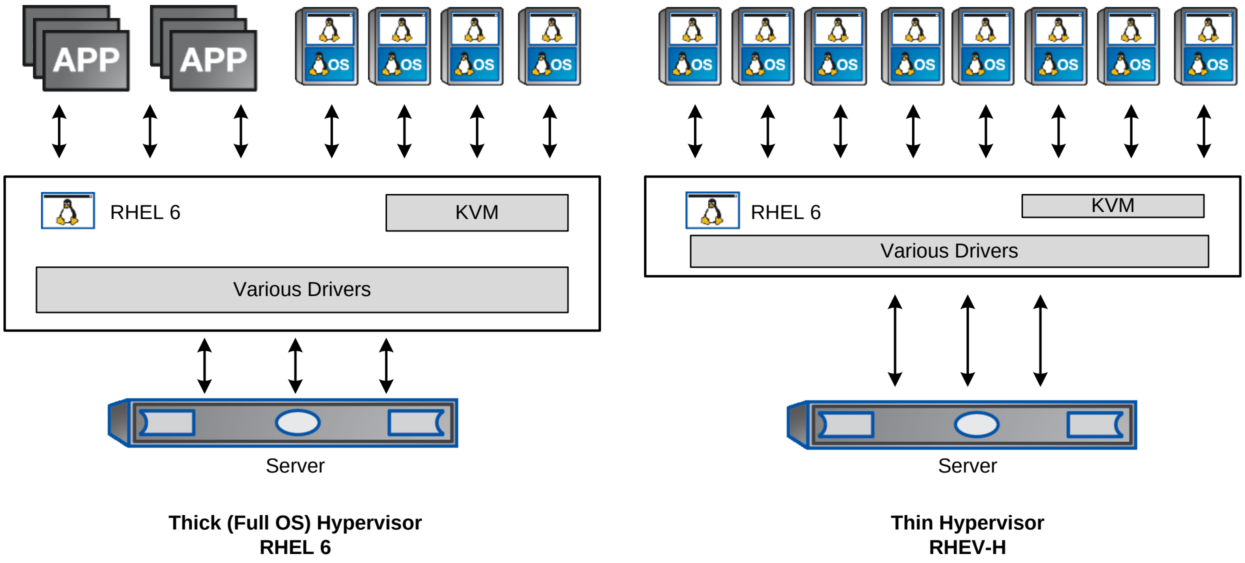
1.4 Best Practices and Supportability

Following the steps in this document are fully supported by both NetApp and Red Hat for their respective products. However, the ways and means described here cannot possibly cover every scenario or business need. Rather, this deployment guide should serve as a starting point. Should your environment require things not covered in this document, technical resources from Red Hat and NetApp are available to provide guidance that is compliant with support.

1.5 Thick and Thin Hypervisors

As described in "TR-3848: RHEL 6, KVM, and NetApp Storage: Best Practices Guide," Red Hat supports both thick and thin hypervisors. This document focuses solely on the thick or full operating system hypervisor available with Red Hat Enterprise Linux 6.

Figure 1) Thick and thin hypervisors.



2 Deploying NetApp Storage Best Practices for Red Hat KVM

It is assumed that a NetApp controller is already configured, at least from a base standpoint. That is to say that a NetApp FAS controller running Data ONTAP 7.3.5 or higher or 8.0.1 7-Mode is already deployed and accessible. Data ONTAP Cluster-Mode is not covered in this technical report.

Here are the items that should already be configured:

- Data ONTAP installed on its own root FlexVol® volume and aggregate
- Basic networking to include a primary interface, management interface, and host name that is fully resolvable in DNS
- A second NetApp FAS controller configured in active-active configuration (or MetroCluster™)
- At least one large disk aggregate that is separate from the root aggregate to be used for data storage

2.1 NetApp Storage Security Best Practices

1. Run the following commands to enhance security on the storage controller:

```
options rsh.access none
options rsh.enable off
options webdav.enable off
options security.passwd.rules.everyone on
options security.passwd.rules.minimum 8
options security.passwd.rules.maximum 14
options security.passwd.rules.minimum.alphabetic 2
options security.passwd.rules.minimum.digit 1
options security.passwd.rules.minimum.symbol 1
options security.passwd.rules.history 6
options security.passwd.lockout.numtries 6
options security.passwd.firstlogin.enable on
options telnet.enable off
options autologout.console.timeout 5
options autologout.console.enable on
options autologout.telnet.timeout 5
options autologout.telnet.enable on
options httpd.enable off
options httpd.timeout 300
options ssh.idle.timeout 60
```

2. Create a new administrator account to replace the root login:

```
useradmin user add newadmin -g administrators
```

3. Log in with the new administrator account to verify that it works.
4. Disable the root account from the new administrator account:

```
options security.passwd.rootaccess.enable off
```

5. Run the following commands per existing user:

```
useradmin user modify <acct> -g <group> -M 90
useradmin user modify <acct> -g <group> -m 1
```

Note: Secure Shell (SSH) protocol must be configured and enabled. Use the following one-time command to generate host keys:

```
secureadmin setup -q ssh 768 512 768
```

6. Use the following options to configure and enable SSH:

```
options ssh.access *
options ssh.enable on
options ssh.idle.timeout 60
options ssh.passwd_auth.enable on
options ssh.pubkey_auth.enable on
options ssh1.enable off
options ssh2.enable on
```

2.2 Licenses for Storage Features

If this is a new NetApp FAS controller, specific licenses must be obtained and installed prior to deploying storage.

The most relevant licenses to this deployment guide include, but are not limited to:

- FCP (includes FCoE)
- iSCSI
- NFS
- Deduplication
- FlexClone®
- SnapRestore®
- SnapMirror®
- SyncMirror® local (if using MetroCluster)
- Cluster remote (if using MetroCluster)

Multiple licenses can be installed as follows from the NetApp command line:

```
license add <first_license> <second_license> <third_license>
```

2.3 Enabling Storage Protocols

NFSv3

1. Add a license for NFS:

```
license add <LICENSE_KEY>
```

2. Set the following recommended options that enable NFS version 3:

```
options nfs.tcp.enable on
options nfs.udp.enable off
options nfs.v3.enable on
```

3. Enable NFS:

```
nfs on
```

Note: If this is part of an active-active pair, run these steps against both controllers.

FCP and FCoE

1. License FCP by logging into the NetApp controller and entering the following command:

```
license add <LICENSE_KEY>
```

2. Start the FCP service by entering the following command:

```
fcp start
```

3. Record the WWPN or FC port name for later use by entering the following command:

```
fcp show adapters
```

4. Check whether the Fibre Channel ports are targets or initiators by entering the following command:

```
fcadmin config
```

5. Make a Fibre Channel port into a target.

   Note: Only Fibre Channel ports that are configured as targets can be used to connect to initiator hosts on the SAN.

   For example, make a port called `0b` into a target port run by entering the following command:

```
fcadmin config -t target 0b
```

Note: If an initiator port is made into a target port, a reboot will be required. NetApp recommends rebooting after completing the entire configuration, because other configuration steps might also require a reboot.

Note: If this is part of an active-active pair, run these steps on both controllers.

iSCSI

The following steps configure the iSCSI service on a storage system.

Note: These steps do not include any host-specific configuration tasks.

1. From the storage controller CLI, license the iSCSI protocol:

Note: license add <LICENSE_KEY>

The following commands can all be run in the vFiler® context.

2. Start the iSCSI service:

```
iscsi start
```

3. Enable or disable the appropriate storage system interfaces for use with iSCSI.

   This example enables the iSCSI protocol on interfaces `e0a` and `e0c` and disables iSCSI on interfaces `e0b`, `e0d`, and `e0M`.

   Note: By default, all storage system interfaces are enabled for iSCSI when the service is started.

```
iscsi interface enable e0a e0c
iscsi interface disable e0b e0d e0M
```

4. Set the default iSCSI authentication method to `deny`:

```
iscsi security default –s deny
```

Note: If this is part of an active-active pair, run these steps on both controllers.

2.4 Provisioning NetApp Storage for RHEL 6 and KVM

Prior to creating NFS exports or LUNs, one or more FlexVol volumes need to be created. This in turn requires that at least one disk aggregate be created based on NetApp best practices.

2.5 Creating a FlexVol Volume

The following information is required to create a flexible volume: the volume's name and size and the aggregate on which it will exist. For example, to create a volume called `kvm_vol` on aggregate `aggr1` with a size of 10GB, run the following commands:

```
vol create kvm_vol01 aggr1 10g
vol options kvm_vol01 create_ucode on
vol options kvm_vol01 convert_ucode on
vol options kvm_vol01 nosnap on
```

2.6 Creating a LUN

LUNs are created within a FlexVol volume.

1. To create a 2GB LUN in volume kvm_vol called kvm_lun, run the following command:

```
lun create –s 2g –t linux /vol/kvm_vol01/kvm_lun01
```

2. Create an igroup:

For FCP or FCoE, an igroup named kvm_igroup01 is created:

```
igroup create -f -t linux kvm_igroup01 20:00:00:e0:8b:9d:a1:4d
```

   Note: The `–t` option specifies the OS type, and `–f` specifies FC.

   Note: Change the WWPN to reflect your environment.

For iSCSI, an igroup named kvm_igroup02 is created with an initiator name of `iqn.2012-01.com:server.host02:`

```
igroup create -i -t linux kvm_igroup02 iqn.2012-01.com:server.host02
```

   Note: The `–t` option specifies the OS type, and `–i` specifies iSCSI.

   Note: Change the IQN to reflect your environment. The IQN can be found on the HBA BIOS (if using a hardware-based initiator) or in `/etc/iscsi/initiatorname.iscsi` file (if using the native RHEL software-based initiator).

3. Map the LUN used to boot the igroup (in this example to the iSCSI igroup):

```
lun map /vol/kvm_vol01/kvm_lun01 kvm_igroup02
```

4. Mount and format the LUN, if using for data, or install the operating system if it is a boot LUN.

Note: Depending on the operating system, the concept of file system alignment might need to be addressed. Refer to section 5.2, “File system alignment.”

2.7 Creating a Thin-Provisioned NFSv3 Export

1. Create a new FlexVol volume, such as `kvm_vol02`.
2. Run the following commands to modify the volume options and the volume Snapshot™ options. These commands set the FlexVol volume `kvm_vol02` to be thin provisioned:

```
vol options kvm_vol02 guarantee none
vol options kvm_vol02 fractional_reserve 0
vol autosize kvm_vol02 on
vol autosize kvm_vol02 -m <2*SIZE>g
vol options kvm_vol02 try_first volume_grow
snap reserve kvm_vol02 0
snap autodelete kvm_vol02 on
```

Note: The autodelete setting differs from the one presented in RA-0007: Storage Efficiency Every Day. In RA-0007, autodelete is set to `off`. Here it is set to `on` to allow the storage to be imported into Provisioning Manager.

Note: Setting the autosize option to two times the original size of the volume allows the volume to double its original size by using more space from the aggregate. Be certain to understand the consequences of this setting and to monitor free space in all aggregates.

3. Export a new volume read/write to host IP 192.168.27.40 and put the entry permanently into the `/etc/exports` file:

```
exportfs -p rw=192.168.27.40 /vol/kvm_vol02
```

The volume should now be accessible from the host.

4. The `/etc/exports` file can also be modified manually. When this modification is complete, run the following command to export all existing entries in the `/etc/exports` file:

```
exportfs -a
```

Note: The `exportfs` command has many options. For more information, refer to now.netapp.com/NOW/knowledge/docs/ontap/rel733/html/ontap/cmdref/man1/na_exportfs.1.html.

2.8 Thin-Provisioning SAN

1. Run the following commands to modify the volume options and the volume Snapshot options. These commands set the FlexVol volume `kvm_vol01` and LUN `kvm_lun01` to be thin provisioned:

```
vol options kvm_vol01 guarantee none
vol options kvm_vol01 fractional_reserve 0
vol autosize kvm_vol01 on
vol autosize kvm_vol01 -m <2*SIZE>g
vol options kvm_vol01 try_first volume_grow
snap reserve kvm_vol01 0
snap autodelete kvm_vol01 on
lun set reservation /vol/kvm_vol01/kvm_lun01 disable
```

Note: The autodelete setting differs from the one presented in RA-0007: Storage Efficiency Every Day. In RA-0007, autodelete is set to `off`. Here it is set to `on` to allow the storage to be imported into Provisioning Manager.

Note: Setting the autosize option to two times the original size of the volume allows the volume to double its original size by using more space from the aggregate. Be certain to understand the consequences of this setting and to monitor free space in all aggregates.

2.9 Enabling Deduplication

1. Enable deduplication on FlexVol volume `/vol/kvm_vol01` by running:

```
sis on /vol/kvm_vol01
```

2. Start the initial scan (which should be run only once).

   Note: The initial scan of the data is most resource intensive; therefore, verify whether the time is acceptable before running it.

```
sis start -f -s /vol/kvm_vol01
```

3. Answer `yes` to the question.
4. Change the default deduplication schedule by running:

```
sis config -s SCHEDULE /vol/kvm_vol01
```

   Where `SCHEDULE` can be in one of these formats:

   - `[day_list][@hour_list]`
   - `[hour_list][@day_list]`
   - `auto`

Note: The `auto` setting runs only when 20% of the data in the volume has changed.

3 Storage Network Best Practices for Red Hat KVM

3.1 Storage Architecture Concepts

The examples explaining deployment in the subsequent section assume that redundant switches are set up and that Link Aggregate Control Protocol (LACP) has been enabled and configured on the relevant switch ports.

3.2 IFGRP LACP

Since this type of interface group requires multiple Ethernet interfaces and a switch that supports LACP, make sure that the switch is configured properly.

The commands are the same for Gigabit Ethernet (GbE) or 10 Gigabit Ethernet (10GbE).

Run the following command on the command line and also add it to the `/etc/rc` file, so it is activated upon boot. This example assumes that there are two network interfaces called `e0a` and `e0b` and that an interface group called `vif01` is being created:

```
ifgrp create lacp vif01 –b ip e0a e0b
wrfile -a /etc/rc "ifgrp create lacp vif01 –b ip e0a e0b"
```

Note: All interfaces must be in `down` status before being added to an interface group.

3.3 VLAN

Run the following commands to create tagged VLANs. This example assumes that there is a network ifgrp called `vif01` and that VLANs are being created that are tagged with numbers 10, 20, and 30. For failover to work properly, the commands must also be run on the other controller in an active-active pair.

```
vlan create vif01 10 20 30
wrfile -a /etc/rc "vlan create vif01 10 20 30"
```

Note: Although this example uses an ifgrp network interface, a non-ifgrp (e0a, e0b) can be used instead.

3.4 IP Config

Run the following commands from the command line.

Note: This example assumes that the user has a network interface called vif01-10 with IP address 192.168.10.10 and default router 192.168.10.1. If routing is not needed, skip the commands starting with `route add`.

```
ifconfig vif01-10 192.168.10.10 netmask 255.255.255.0 mtusize 1500 flowcontrol none
wrfile -a /etc/rc "ifconfig vif01-10 192.168.10.10 netmask 255.255.255.0 mtusize 1500 flowcontrol
none"
route add default 192.168.10.1 1
wrfile -a /etc/rc "route add default 192.168.10.1 1"
routed on
wrfile -a /etc/rc "routed on"
```

Note:

- The interface can be either physical (for example, e0a), an ifgrp (VIF), or a VLAN.
- Run this routine once for every physical or virtual interface for which an IP address is needed.
- If MultiStore® is used, routing is common to every vFiler unit in an IP space.
- If a 10Gb per second interface is not used, remove the `flowcontrol none` option.
- If jumbo frames are needed, change the `mtusize` option to 9000.

4 KVM Host Node Installation and Configuration

4.1 Hardware Requirements for Red Hat KVM

For all hardware requirements, refer to the RHEL 6 Installation Guide and the RHEL 6 Virtualization Guide. They can also be found in "TR-3848: RHEL 6, KVM, and NetApp Storage: Best Practices Guide."

4.2 Boot from SAN

Booting from iSCSI (HBA)

Although RHEL 6 supports the use of a software initiator for boot, an iSCSI HBA or multiprotocol Ethernet adapter is recommended for iSCSI-based boot LUNs. Specific configuration for hardware initiators differs from vendor to vendor, but generally complies with the following workflow:

1. Boot the server to the BIOS (onboard multiprotocol Ethernet device) or HBA BIOS (iSCSI HBA).
2. Enable the device as a boot device (might need to enable the BIOS on an iSCSI HBA).
3. Make note of the iSCSI initiator name or edit it to match a predetermined initiator name.
4. If necessary, edit the NetApp igroup to include the initiator name if it has not been entered yet.
5. Scan the bus for devices.
6. Select the device from which to boot.
7. Reboot.
8. RHEL 6 recognizes the device on reboot and during the install process.

Booting from FC (FCP or FCoE HBA)

An FCP or FCoE is required for FC-based boot LUNs. Specific configuration for HBAs differs from vendor to vendor, but generally complies with the following workflow:

1. Boot the server to the HBA BIOS.
2. Enable the device as a boot device (might need to enable the BIOS).
3. Make note of the WWPN.
4. If necessary, edit the NetApp igroup to include the WWPN(s).
5. Scan the bus for devices.
6. Select the device from which to boot.
7. Reboot.
8. RHEL 6 recognizes the device on reboot and during the install process.

Disk Layout for RHEL 6 KVM Host

Aside from following Red Hat best practices, disk layout is not critical. However, it is recommended to provide separate partitions for the following directories:

- /boot
- /var
- /opt (if using the Snap Creator™ backup framework)

Package Selection

If performing an interactive installation, select the Virtualization package group and all of the subgroups associated with it. If installing an RHEL 6 KVM host using Kickstart, select the following package groups:

- `@virtualization`
- `@virtualization-client`
- `@virtualization-platform`
- `@virtualization-tools`

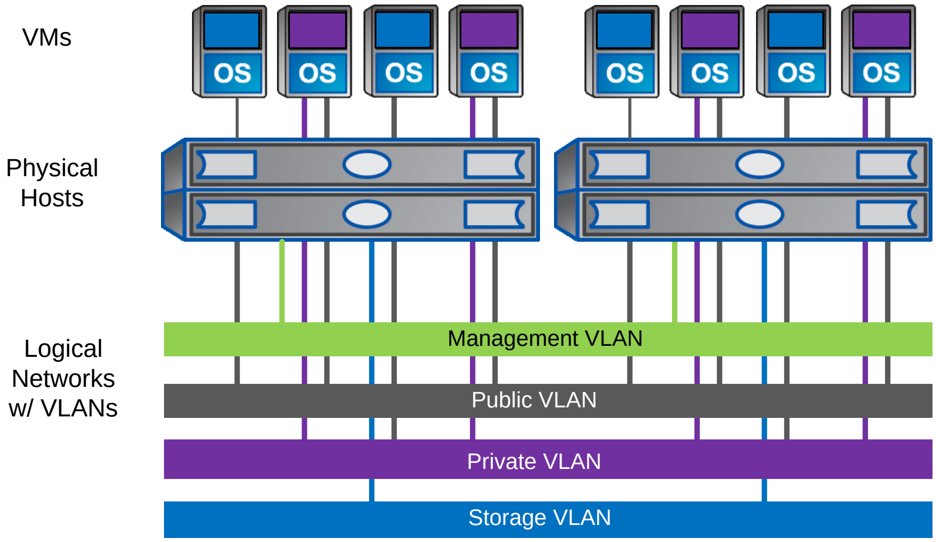
Caution

Do not install development packages, network sniffers, graphical desktop, or other server types on a hypervisor node. If a graphical desktop environment is needed to help manage the hypervisors, see section on setting up a remote administration host.

4.3 RHEL 6 Network Configuration

The RHEL 6 KVM hosts requires multiple network interfaces. The following examples explain the process of creating logical interfaces, VLAN tagging, and combining the two of them.

Figure 2) Example of VLAN layout.



Creating Logical Network Interfaces

Create a Channel Bond

1. Log in to the RHEL host to be configured with channel bonding.
2. Determine the physical network adapters to be bonded, for example, Eth0 and Eth1.
3. Determine the bonding mode to be utilized.
4. Disable the NetworkManager service by typing the following commands:

```
service NetworkManager stop
chkconfig NetworkManager off
```

Note: If the NetworkManager service is already off or not installed, the preceding service command fails. This is both expected and fine.

5. Create the file /etc/sysconfig/network-scripts/ifcfg-bond0 with the following entries:

```
DEVICE=bond0
TYPE=bond
USERCTL=no
BOOTPROTO=static
IPADDR=<IP_address>
NETMASK=<network_mask>
ONBOOT=yes
BONDING_OPTS="mode=0 miimon=100"
```

Note: This assumes that the bonding mode selected is mode 0.

Note: If jumbo frames are needed, add the line MTUSIZE=9000.

6. Edit the file /etc/sysconfig/network-scripts/ifcfg-eth0 to match the following entries:

```
DEVICE=eth0
BOOTPROTO=none
```

```
HWADDR=(MAC address of eth0)
ONBOOT=yes
MASTER=bond0
SLAVE=yes
USERCTL=no
```

7. Edit the file /etc/sysconfig/network-scripts/ifcfg-eth1 to match the following entries:

```
DEVICE=eth1
BOOTPROTO=none
HWADDR=(MAC address of eth1)
ONBOOT=yes
MASTER=bond0
SLAVE=yes
USERCTL=no
```

8. Create a new file called bonding.conf in the /etc/modprobe.d directory with the following line:

```
alias bond0 bonding
```

Note: Each additional bond will require an additional line (bond1, bond2, and so on).

9. Restart the networking service by typing the following command:

```
service network restart
```

Creating a Network Interface with VLAN Tagging

1. Log in to the RHEL host to be configured with VLAN tagging.
2. Determine the physical network adapters to be tagged, for example, Eth0.
3. Determine the VLAN tag to be used, for example, 100.
4. Load the VLAN tag module by typing the following command:

```
modprobe 8021q
```

5. Disable the NetworkManager service by typing the following commands:

```
service NetworkManager stop
chkconfig NetworkManager off
```

Note: If the NetworkManager service is already off or not installed, the preceding commands fail. This is both expected and fine.

6. Create the file `/etc/sysconfig/network-scripts/ifcfg-eth0.100` to match the following entries:

```
DEVICE=eth0.100
BOOTPROTO=static
ONBOOT=yes
IPADDR=<IP_address>
NETMASK=<network_mask>
VLAN=yes
USERCTL=no
```

7. Bring the interface up by typing the following command:

```
ifup eth0.100
```

Creating VLANs from Bonded Interfaces

1. Log in to host that is to be configured with the combination of bonded interfaces and VLAN tagging.
2. Determine the VLAN tag to be used, such as 100.
3. Follow the instructions for bonding two interfaces, such as Eth0 and Eth1, with one exception: create the `/etc/sysconfig/network-scripts/ifcfg-bond0` file as follows:

```
DEVICE=bond0
TYPE=bond
USERCTL=no
BOOTPROTO=none
ONBOOT=yes
BONDING_OPTS="mode=0 miimon=100"
```

Note: The ifcfg-eth0 and ifcfg-eth1 files and the /etc/modprobe.d/bonding.conf files are still created as originally described.

4. Create the file ifcfg-bond0.100 in the /etc/sysconfig/network-scripts directory with the following content:

```
DEVICE=bond0.100
VLAN=yes
IPADDR=<IP_address>
NETMASK=<network_mask>
ONBOOT=yes
BOOTPROTO=static
TYPE=ETHERNET
```

5. Restart the network service by typing the following command:

```
service network restart
```

6. To load the module automatically on boot, create the file `/etc/sysconfig/modules/vlan.modules` with permissions of 755 and the following contents:

```
#!/bin/sh

if [ `lsmod | grep -c 8021q`  -eq 0 ] ; then
    exec /sbin/modprobe -b 8021q >/dev/null 2>&1
fi
```

Configuring the libvirt Daemon

The libvirt daemon must be configured to start automatically on boot.

1. To configure the libvirt daemon to start automatically on boot, enter the following command:

```
chkconfig libvirtd on
```

2. Check to see that it is running with the following command:

```
service libvirtd status
```

3. If the libvirt daemon needs to be started, enter the following command:

```
service libvirtd start
```

Configuring Time for KVM Hosts

To avoid issues caused by incorrect time, it is paramount to enable and use NTP. This assumes that there is an operational NTP server accessible on the network.

1. The NTP service should be enabled and running on all RHEL servers, including KVM hosts:

```
service ntpd stop
ntpdate <ip_of_NTP_server>
echo “server <ip_of_NTP_server>” >> /etc/ntp.conf
chkconfig ntpd on
service ntpd start
```

Note: If the service ntpd stop command fails, it is likely because it was not running in the first place. This is both expected and fine.

Other Required Services

There are services that should also be enabled and started on an RHEL 6 KVM host. These include but are not limited to (depending on requirements):

- iptables
- ip6tables (if using IPv6)
- iscsi and iscsid (if using software-based iSCSI initiator)
- ksm and ksmtuned (if using shared memory)
- libvirtd and libvirt-guests
- multipathd
- netfs
- network
- ntpd
- sshd

4.4 Registering an RHEL 6 Host to Red Hat Network

To receive updates, patches, and fixes, an RHEL host must be registered to the Red Hat Network, or RHN. This requires a valid subscription to Red Hat Network (RHN).

1. Log in to the host that needs to be registered to Red Hat Network.
2. Type the following command to register the host to RHN:

```
subscription-manager register --username <rhn_login> --password
<rhn_password>
```

3. Type the following command to subscribe the host to all software channels to which it is entitled:

```
subscription-manager subscribe –auto
```

4.5 Creating Dedicated IP Access for Virtual Machines

The following steps provide the necessary guidance to allow virtual machines to have two-way network connectivity through a virtual bridge. In the following examples, the virtual bridge created is `br0`, but it could also be named `br_nfs`, `br-iscsi`, or any other meaningful name if necessary.

1. Determine which physical Ethernet device will be dedicated for use by virtual machines (the following example uses Ethernet device eth2).

2. Edit the ifcfg-eth2 file in the `/etc/sysconfig/network-scripts` directory to match the following contents:

```
DEVICE="eth2"
HWADDR="<mac_address>"
NM_CONTROLLED="no"
ONBOOT="yes"
BRIDGE=br0
BOOTPROTO=none
```

   Note: Replace <mac_address> with the actual value.

3. Create a file in the /etc/sysconfig/network-scripts directory (the name has to match the BRIDGE variable from the preceding file, for example, ifcfg-br0). The br0 matches the BRIDGE variable. The file should have the following contents:

```
DEVICE=br0
TYPE=Bridge
BOOTPROTO=static
IPADDR=<ip_address>
NETMASK=<netmask>
ONBOOT=yes
```

   Note: Replace the <ip_address> and <netmask> with actual values.

4. Restart networking on the server:

```
service network restart
```

5. Allow traffic bound for virtual machines to pass through the KVM host firewall by typing the following commands:

```
iptables -A INPUT -i br0 -j accept
service iptables save
```

Note: Any additional virtual bridges (br1, br2, and so on) that are created also require an IPtables rule to be entered and saved.

4.6 General RHEL 6 KVM Host Security Deployment

1. Create separate logins for each required user.
2. Enforce the use of strong passwords based on the Red Hat Knowledge Base article at access.redhat.com/kb/docs/DOC-9128.
3. List out all services that are configured to start on boot using the following command:

```
chkconfig –list | grep 3:on | awk {‘print $1’} | pr –T –columns=3
```

4. Shut down and disable any unneeded services, for example:

```
service NetworkManager stop; chkconfig NetworkManager off
```

5. Remove unneeded services where possible, for example:

```
yum –y remove postfix
```

6. Disable/uninstall RSH, telnet, and FTP in favor of SSH, SCP, and SFTP (or other secure FTP server).
7. The host /etc/fstab file should not use disk labels from which to boot.

   Note: If using a disk label as part of a cloning procedure, include a script to switch back to booting from a UUID when the cloned host comes back up.

8. Register all Red Hat KVM hosts to RHN to provide access to the latest security patches and bug fixes.

Configuring Secure Remote Access Between Hypervisors

The following steps provide the necessary guidance to configure the use of SSH key pairs for secure remote access to KVM hosts as well as between KVM hosts. The following example uses two KVM hosts, host1 and host2.

1. Create the key pairs.
   a. On host1, type:

```
cd /root
ssh-keygen -t rsa
```

   b. Accept the defaults and press Enter when prompted for a passphrase.
   c. On host2, type:

```
cd /root
ssh-keygen -t rsa
```

   d. Accept the defaults and press Enter when prompted for a passphrase.

   Note: This process creates two files on each host: id_rsa and id_rsa.pub.

2. Next, distribute the keys between the two hosts.
   a. On host1, type:

```
cd /root
scp .ssh/id_rsa.pub host2:/root/host1.pub
```

   b. On host2, type:

```
scp .ssh/id_rsa.pub host1:/root/host2.pub
```

   c. On host1, type:

```
cd /root
touch .ssh/authorized_keys
chmod 600 .ssh/authorized_keys
cat /root/host2.pub >> .ssh/authorized_keys
```

   d. On host2, type:

```
cd /root
touch .ssh/authorized_keys
chmod 600 .ssh/authorized_keys
cat /root/host2.pub >> .ssh/authorized_keys
```

3. The two RHEL 6 KVM hosts can now communicate through SSH without passwords, allowing for seamless virtual machine migration.

Using the Native RHEL 6 Firewall (IPtables)

Do not disable the IPtables firewall. Determine the ports that need to be opened and allow them using the following procedures. See the appendix for a list of ports to allow.

Note: If an RHEL KVM guest needs access to a specific port, then it will need to be opened on both the guest firewall and the host firewall.

1. Determine what ports are already allowed through the firewall by typing the following command:

```
iptables -L
```

2. To configure an individual port, such as SSH, to pass through the firewall, enter the following command:

```
iptables -I INPUT -m state --state NEW -p tcp --dport 22 -j ACCEPT
```

3. To configure a range of ports to pass through the firewall, such as VNC consoles, enter the following command:

```
iptables -I INPUT -m state --state NEW -p tcp --dport 5900:5910 -j ACCEPT
```

4. After making changes to the firewall, enter the following command to save the configuration:

```
# service iptables save
```

Using SELinux with KVM

Security Enhanced Linux (SELinux) provides an additional layer of security. By default, SELinux is enabled, and it should not be disabled.

1. To make sure that SELinux is enabled, type the following command:

```
getenforce
```

The response should be `Enforcing`.

2. NFS mounts used for virtual machine storage might need to have SELinux enabled. To do so, enter the following command:

```
setsebool -P virt_use_nfs=on
```

3. Block devices created postinstall, such as for VM storage, will need to be configured for use with SELinux. The following example assumes that a LUN is mounted on the default images directory of `/var/lib/libvirt/images.` Type the following commands to change the security context:

```
semanage fcontext -a -t virt_image_t “/var/lib/libvirt/images(/.*)?”
restorecon -R -v /var/lib/libvirt/images
```

Creating a Red Hat KVM Host Template

Creating an RHEL 6 KVM host template allows for rapid deployment of hypervisors by using NetApp FlexClone. After the template is created, FlexClone can clone the template much faster than installing from scratch or even network installs.

Preparing a Base Image

5. Install and create an RHEL 6 KVM host that includes all of the packages, security, and network configuration as described in the preceding sections.
6. Register the RHEL 6 KVM host to RHN and update all packages.

Making the Base Image Generic

1. Strip out all static configuration artifacts:

   Note: These items need to be reconfigured when the RHEL 6 KVM host comes back up, either manually or by script. Depending on what is configured on the host, there might be additional files to edit.

   – Strip host name, gateway, and IP information (/etc/hosts, /etc/sysconfig/network, /etc/sysconfig/network-scripts/ifcfg-{eth*,br*,bond*})

   – Strip MAC addresses from Ethernet configuration files (/etc/sysconfig/network-scripts/ifcfg-{eth*,br*,bond*})

   – If registered to RHN (including satellite and proxy), strip the systemid (/etc/sysconfig/rhn/systemid)

   – Strip the iSCSI initiator name in /etc/iscsi/initiatorname.conf (can be regenerated with iscsi-name command)

   - Replace LUN WWID with either a NetApp friendly wildcard or full wildcard in the multipath configuration file, etc/multipath.conf. `wwid 360a98000572d4273685a664462667a36 becomes wwid 360a9*`
   - Rebuild initramfs using `dracut`
   - Clear out multipath bindings (RHEL 6) (/etc/multipath/bindings)
   - Label boot device and direct system to boot from the label and not a UUID (RHEL 6) or path (RHEL 5) (`e2label` and /etc/fstab)
2. Strip out all dynamic configuration artifacts

   Note: These items will be recreated automatically when the RHEL 6 KVM host reboots. Depending on what is configured on the host, there might be additional files to edit.

   - Clear out LVM cache (/etc/lvm/cache/*)
   - Remove UDEV rule for Ethernet device assignment (RHEL 6) (/etc/udev/rules.d/70-persistent-net.rules)
   - Remove remaining persistent UDEV rules (RHEL 6) (/etc/udev/rules.d/*-persistent-*.rules)
   - Remove SSH host keys (/etc/ssh/ssh_host*)

Cloning the Boot LUN

Note: This requires the FlexClone and deduplication licenses to be added.

1. Create an RHEL 6 KVM host template that boots from a NetApp LUN as described in the previous section.
2. Log in to the NetApp controller, and enter the following command to clone the boot LUN:

```
clone start /vol/vol_name/LUN_tmplt /vol/vol_name/clone_name -n -l
```

3. Create a new igroup for the new RHEL 6 KVM host (see the section on provisioning NetApp storage).
4. Map the newly cloned boot LUN to the new igroup (see the section on provisioning NetApp storage).
5. Boot the newly cloned server.

4.7 Attaching an RHEL 6 KVM Host to NetApp Storage

Mounting NFS For VM Storage

The following steps provide the necessary guidance to mount NFS-based storage for use as a virtual machine datastore. This assumes that an NFS export was created on the NetApp controller using NetApp best practices.

4.8 Configuring RHEL 6 KVM Host for Predictable NFS Access

1. Configure the NFS client to use predictable ports, uncomment the following two lines in /etc/sysconfig/nfs:

```
LOCKD_TCPPORT=32803
STATD_PORT=662
```

2. Allow ports 32803 and 662 through the IPtables firewall.
3. Restart the host for the changes to take effect:

```
init 6
```

4. When the host comes back up, make sure that the host can see the available NFS storage by entering the following command:

```
showmount -e <IP_of_NetApp_NFS> | grep <path_of_export>
```

5. Add the NFS entry to the /etc/fstab file by entering the following command:

```
echo “<IP_of_NetApp_NFS>:<path_of_export> /var/lib/libvirt/images nfs
_netdev,defaults 0 0” >> /etc/fstab
```

6. Mount the NFS export as configured in /etc/fstab by typing the following command:

```
mount -a
```

Configuring Software-Based iSCSI Initiator

Note: If using a hardware-based initiator, all of this information can be added to the HBA BIOS.

1. Verify that the iSCSI package is installed:

```
rpm -q iscsi-initiator-utils
```

2. Change the iSCSI initiator nodename to match the system’s host name. Edit the `/etc/iscsi/initiatorname.iscsi` file and either note the value or edit the value to meet your needs. For example:

```
InitiatorName=iqn.2005-03.com.RedHat:linuxhost01
```

3. Add CHAP user names and passwords to the `/etc/iscsi/iscsid.conf` file:

```
node.session.auth.authmethod = CHAP
node.session.auth.username = iqn. 2005-03.com.RedHat:linuxhost01
node.session.auth.password = Password1234
discovery.sendtargets.auth.authmethod = CHAP
discovery.sendtargets.auth.username = iqn.2005-03.com.RedHat:linuxhost01
discovery.sendtargets.auth.password = Password1234
```

   Note: If using DM-Multipath, then change the timeout parameter to 5.

4. Start the iSCSI service:

```
service iscsid start
```

5. The target must be manually discovered once. Do the initial discovery for a specific IP address, for example:

```
iscsiadm -m discovery -t st -p 192.168.1.10
```

6. To automatically log in to all discovered nodes upon the next startup, add the following line to the `/etc/iscsi/iscsid.conf` file:

```
node.startup = automatic
```

   Note: This setting configures the host to only send network traffic to those already manually discovered by the host.

7. Configure the iSCSI service to start automatically at startup:

```
chkconfig iscsi on
```

8. On the storage controller, add a CHAP authentication entry for the server IQN that was obtained in step 2.

   Note: The password must be greater than or equal to 12 bytes.

   Note: If preexisting naming standards are not in place, the iSCSI initiator IQN can be used for the iSCSI CHAP user name as shown in the following example:

```
iscsi security add -i iqn.2005-03.com.RedHat:linuxhost01 -s CHAP -p Password1234 -n
iqn.1994-05.com.redhat:linuxhost01
```

The `-n` specifies the user name, which in this case is identical to the IQN, and the `-p` is the password used to authenticate the host to the storage controller.

9. On the storage controller, add the appropriate interfaces to the iSCSI interface access list for the IQN.

   Note: NetApp recommends this optional step. Replace the interfaces or ifgrps with appropriate values.

```
iscsi interface accesslist add iqn.2005-03.com.RedHat:linuxhost01 e0a e0c vif01
```

10. Create an igroup for the server if one has not been already created:

```
igroup create -i -t linux linuxhost01 iqn.2005-03.com.RedHat:linuxhost01
```

Detecting LUNs

This requires that at least one LUN has been properly created, zoned, and mapped to a NetApp igroup. This is relevant to all LUN-based storage protocols.

1. From the RHEL 6 host, type the following command to view all disks – local and remote:

```
cat /proc/partitions
```

If you follow the best practice of using multiple paths to the storage, the new LUN(s) will show up as multiple devices.

2. Follow the steps in “Configure RHEL 6 DM-Multipath” to complete the steps.

Configuring RHEL 6 DM-Multipath

This is relevant to all LUN-based storage protocols.

1. Verify that the DM-Multipath package is installed by running the following commands:

```
rpm -q device-mapper
rpm -q device-mapper-multipath
```

2. Replace the content of the `/etc/multipath.conf` file with the following:

```
defaults
{
        user_friendly_names yes
        max_fds 4096
        queue_without_daemon no
}
#blacklist
#{
#       wwid “*”
#       devnode "^(ram|raw|loop|fd|md|dm-|sr|scd|st)[0-9]*"
#       devnode "^hd[a-z]"
#       devnode "^cciss!c[0-9]d[0-9]*[p[0-9]*]"
#}
devices
{
        device
        {
        vendor "NETAPP"
        product "LUN"
        getuid_callout "/sbin/scsi_id -g -u -s /block/%n"
        prio_callout "/sbin/mpath_prio_alua /dev/%n"
        features "1 queue_if_no_path"
        hardware_handler "1 alua"
        path_grouping_policy group_by_prio
        failback immediate
        rr_weight uniform
        rr_min_io 128
        path_checker directio
        flush_on_last_del yes
        }
}
```

Note:

- If configuring DM-Multipath on an iSCSI host, then set `path_grouping_policy` to `multibus.`
- If using Red Hat Enterprise Linux 5 Update 2 or earlier, then remove the `flush_on_last_del` line.
- If using Red Hat Enterprise Linux 5 Update 1 or earlier, then remove the `max_fds` line.
- If using Red Hat Enterprise Linux with no updates, then change `directio` to `readsector`0.
- If using ALUA, then change `/sbin/mpath_prio_alua` to `/sbin/mpath_prio_ontap` and `1 alua` to `0.`
- Blacklist any nonmultipathed devices.

3. Start the DM-Multipath service:

```
service multipathd start
```

4. Configure the multipath devices by running the `multipath` command:

```
multipath -ll
```

5. Configure the DM-Multipath service to start automatically at boot:

```
chkconfig multipathd on
```

Mounting a Multipathed LUN to RHEL 6

After running the `multipath –ll` command mentioned earlier, the multipathed device is ready to have a file system installed on it and mounted.

1. Configure the newly created multipath device with LVM. In the next example, the entire disk is consumed by LVM:

```
pvcreate /dev/mapper/mpatha
vgcreate vg_kvm /dev/mapper/mpatha
lvcreate –l 100%FREE –n lv_kvm vg_kvm
```

2. Create the file system by typing the following command:

```
mke2fs –t ext4 /dev/vg_kvm/lv_kvm
```

3. Mount the newly created file system, in this case, to the default location for VM storage:

```
mount /dev/vg_kvm/lv_kvm /var/lib/libvirt/images
```

4. Make the mount automatic at boot time by typing the following command:

```
echo “/dev/vg_kvm/lv_kvm /var/lib/libvirt/images ext4 defaults 0 0” >> /etc/fstab
```

Note: If the multipathed device is an iSCSI LUN, change the `defaults` option to `_netdev,defaults`.

5 KVM Guest Configuration

Virtual guests running earlier versions of RHEL or Microsoft® Windows® might require the installation of the VirtIO drivers. For additional details, refer to Red Hat Enterprise Linux 6 Virtualization Guide.

5.1 Provisioning Virtual Machines in RHEL 6 KVM (CLI)

The following steps provide the necessary guidance for provisioning virtual machines on the KVM hypervisor in RHEL 6 using the command line.

The following steps assume that there is a Kickstart server available on the network as well as the configuration of bridged networking on the KVM host. Also, the virtual machine is being created on the local KVM host that uses NetApp shared storage mounted on the default location of /var/lib/libvirt/images.

Using the qemu-img Tool

1. Create a thick disk image for the virtual machine to use (assume the default directory for images):

```
qemu-img create -f raw /var/lib/libvirt/images/<name_of_vm>.img 16g
```

2. Provision an RHEL 6 virtual machine with 2 CPUs, 2GB of RAM, with 16GB of disk, that uses the bridged network `br0`. Type the following commands (lines that end in \ signify one long command):

```
virt-install --accelerate --hvm --connect qemu:///system \
       --network bridge:br0 \
       --name <name_of_vm> --vcpus=2 --ram=2048 \
       --file=/var/lib/libvirt/images/<name_of_vm>.img \
       --file-size=16 --vnc --os-variant=rhel6 \
       --location=http://<ip_of_repo_server/<repo_path> -x \
       ks=http://<ip_ks_server>/<path_to_ks_file>
```

Note: Additional parameters are discussed in detail in the main page for virt-install as well as the "Virtualization Administration Guide" available at http://redhat.com.

3. When the VM finishes installing, it will reboot.
4. The VM can then be put into production or configured as a template.
5. If the guest operating system is an RHEL version earlier to version 6 or a Microsoft operating system released prior to 2008, additional deployment steps need to be followed for proper file system alignment. See the next section.

5.2 File System Alignment

RHEL 6 KVM guests support both RHEL and Microsoft operating systems. However, the following versions require additional configuration to avoid misalignment between the virtual disk and the underlying storage:

- RHEL versions 3, 4, and 5
- All Microsoft Windows prior to Windows Vista, Windows Server® 2008, and Windows 7

Note: See TR-3747: Best Practices for File System Alignment in Virtual Environments for a full explanation of file system alignment and the implications of misalignment.

Using Kickstart for File System Alignment

The quickest, most consistent way to properly align the file systems in an RHEL 6 KVM environment is to use Kickstart. For complete information on setting up and using Kickstart, see the Red Hat Enterprise Linux 6 Deployment Guide.

For RHEL Guest Virtual Machines

1. In the Kickstart file that is to be used to create virtual machines and/or virtual machine templates, add the following lines at the end of the file:

```
%pre
 parted /dev/sda mklabel msdos
 parted /dev/sda mkpart primary ext3 64s 208718s
 parted /dev/sda mkpart primary 208720s 100%
 parted /dev/sda set 2 lvm on
```

2. In the main section of the Kickstart file, edit the disk layout as follows:

```
zerombr yes
##clearpart --linux --drives=sda  ## comment out or remove
part /boot --fstype ext3 --onpart sda1
part pv.2 —onpart sda2
volgroup VolGroup00 --pesize=32768 pv.2
logvol swap --fstype swap --name=LogVol01 --vgname=VolGroup00 --size=1008 --grow --maxsize=2016
logvol / --fstype ext3 --name=LogVol00 --vgname=VolGroup00 --size=1024 --grow
```

3. This will result in both partitions on the virtual disk to be properly aligned with the underlying NetApp storage.

For Windows Guest Virtual Machines

An altered Kickstart file can be used to properly align a virtual disk in preparation for a Windows guest installation. Kickstart cannot actually perform the Windows install, only the alignment.

1. Create a typical Kickstart file (copy a known good Kickstart file) that contains valid information and sections. It does not matter what the content is, but Anaconda must be able to parse the Kickstart file.
2. In the Kickstart file that is to be used to create Windows virtual machines and/or Windows virtual machine templates, add the following lines at the end of the file:

```
%pre
parted /dev/sda mklabel msdos
parted /dev/sda mkpart primary NTFS 128s 100%
chvt 3
echo "###################################"
echo "# Reboot with Windows Install DVD #"
echo "###################################"
sleep 5
exit
```

3. The Kickstart process will parse and execute the partition creation in the %pre section, but exit the Kickstart process prior to attempting to install anything.
4. When the Reboot message comes up, reboot the server with the Windows install DVD. When the Windows installer comes up it will recognize the NTFS partition and will allow installation to it.

Security Configuration for RHEL 6 KVM Guests

RHEL Guests

These procedures are similar in scope to those provided for the RHEL 6 KVM host.

1. Create separate logins for each required user, preferably using LDAP.
2. Enforce the use of strong passwords based on the Red Hat Knowledge Base article at access.redhat.com/kb/docs/DOC-9128.
3. List out all services that are configured to start on boot using the following command:

```
chkconfig --list | grep 3:on | awk {‘print $1’} | pr –T --columns=3
```

4. Shut down and disable any unneeded services, for example:

```
service NetworkManager stop; chkconfig NetworkManager off
```

5. Remove unneeded services where possible, for example:

```
yum -y remove postfix
```

6. Disable/uninstall RSH, telnet, and FTP in favor of SSH, SCP, and SFTP (or other secure FTP server).
7. The host /etc/fstab file should not use disk labels from which to boot.

   Note: If using a disk label as part of a cloning procedure, include a script to switch back to booting from a UUID when the cloned host comes back up.

8. Register all Red Hat KVM guests to RHN to provide access to the latest security patches and bug fixes.

Windows Guests

Create separate logins for each required user, preferably using LDAP or Active Directory®.

1. Enforce the use of strong passwords based on the Red Hat Knowledge Base article at access.redhat.com/kb/docs/DOC-9128.
2. List out all services that are running and enabled. Disable all unnecessary services.
3. Uninstall any unnecessary or insecure services.
4. All Windows guests should have access to Windows Update for package updates and security patches.

Using the Native Guest Firewall

RHEL 6 Guests

IPtables is a stateful packet filter that is enabled by default.

Caution

Do not disable IPtables; instead enable additional ports as necessary.

See the appendix for the list of ports to allow.

Note: If an RHEL KVM guest needs access to a specific port, then it will be required to be opened on both the guest firewall and the host firewall.

6. Determine what ports are already allowed through the firewall by typing the following command:

```
iptables -L
```

7. To configure an individual port, such as SSH, to pass through the firewall, enter the following command:

```
iptables -I INPUT -m state --state NEW -p tcp --dport 22 -j ACCEPT
```

8. To configure a range of ports to pass through the firewall, such as VNC consoles, enter the following command:

```
iptables -I INPUT -m state --state NEW -p tcp --dport 5900:5910 -j ACCEPT
```

9. After making changes to the firewall, enter the following command to save the configuration:

```
# service iptables save
```

10. Any ports that need to be opened on a guest need to be opened on the RHEL 6 KVM host as well.

Using SELinux with RHEL 6 Guests

SELinux provides an additional layer of security. By default, SELinux is enabled, and it should not be disabled.

1. To make sure that SELinux is enabled, type the following command:

```
getenforce
```

The response should be `Enforcing`.

2. For additional information on SELinux, refer to the Red Hat Enterprise Linux 6 Security-Enhanced Linux User Guide.

Configuring Time for KVM Guests

To avoid issues around virtual machine migration, SSL certificates, Web sessions, and other problems, guest timing needs to be addressed.

RHEL Guests

The NTP service should be enabled and running on all RHEL KVM guests.

1. Run the following commands to enable, configure, and start NTP (all versions of RHEL):

```
service ntpd stop
ntpdate -q name_of_ntp_server.com
echo “server <name_of_ntp_server.com>” >> /etc/ntp.conf
chkconfig ntpd on
service ntpd start
```

Note: If the service ntpd stop command fails, it is likely because it was not running in the first place. This is both expected and fine.

Windows Vista, Windows Server 2008, and Windows 7 Guests

Launch the Command Prompt application (use the Run as Administrator option) and run the following command:

```
C:\Windows\system32>bcdedit /set {default}

USEPLATFORMCLOCK on The operation completed successfully
```

For additional information regarding KVM guest time, review the chapter titled “KVM guest timing management” in the Virtualization Guide at docs.redhat.com.

Creating a KVM Guest Template

5.3 RHEL Guests

1. Install and create an RHEL 6 KVM guest that includes all of the package, security, and network configuration as described in the preceding sections.
2. Register the RHEL 6 KVM guest to RHN and update all packages.

Making the Base Image Generic

1. Strip out all static configuration artifacts:

   Note: These items need to be reconfigured when the RHEL 6 KVM guest comes back up, either manually or by script. Depending on what is configured on the host, there might be additional files to edit.

   a. Strip host name, gateway, and IP information (/etc/hosts, /etc/sysconfig/network, /etc/sysconfig/network-scripts/ifcfg-{eth*,br*,bond*})
   b. Strip MAC addresses from Ethernet configuration files (/etc/sysconfig/network-scripts/ifcfg-{eth*,br*,bond*})
   c. If registered to RHN (including satellite and proxy), strip the systemid (/etc/sysconfig/rhn/systemid)
   d. Strip the iSCSI initiator name in /etc/iscsi/initiatorname.conf (can be regenerated with iscsi-name command)
   e. Rebuild initrd (RHEL 5) or initramfs (RHEL 6) (mkinitrd for RHEL 5, dracut for RHEL 6)
   f. Label boot device and direct system to boot from the label and not a UUID (RHEL 6) or path (RHEL 5) (`e2label` and /etc/fstab)

2. Strip out all dynamic configuration artifacts.

   Note: These items will be recreated automatically when the RHEL 6 KVM host reboots. Depending on what is configured on the host, there might be additional files to edit.

   a. Clear out LVM cache (/etc/lvm/cache/*)
   b. Remove UDEV rule for Ethernet device assignment (RHEL 6) (/etc/udev/rules.d/70-persistent-net.rules)
   c. Remove remaining persistent UDEV rules (RHEL 6) (/etc/udev/rules.d/*-persistent-*.rules)
   d. Remove SSH host keys (/etc/ssh/ssh_host*)

Windows Guests

1. Install and create a Windows guest virtual machine that includes all of the package, security, and network configuration required.
2. Enable the system to use Windows Update.
3. Use the Microsoft SysPrep tool to make the guest generic.

NFS-Based RHEL 6 KVM Cloning

Cloning NFS-based virtual machines from the NetApp storage is preferable and faster than cloning from the RHEL 6 KVM host.

1. Create a template using the best practices from the previous section.
2. Shut down the virtual machine template.
3. From the NetApp controller, run the following command to clone the template:

```
clone start /vol/kvm_nfs_vol/rhel_base.img /vol/kvm_nfs_vol/rhel_clone.img -n –l
```

   Note: This creates a new virtual machine named rhel_clone.img from a template named rhel_base.img.

4. From the RHEL 6 KVM host, run the following command to create a new XML descriptor file, while preserving the newly created clone disk image:

```
virt-clone --original-xml=/etc/libvirt/qemu/rhel_base.xml --preserve-data --
file=/var/lib/libvirt/images/rhel_clone.img -n rhel_clone
```

LUN-Based RHEL 6 KVM Cloning

Although NetApp FlexClone can be used to clone a FlexVol volume or a LUN, it cannot be used to clone individual virtual machines in a LUN. There is, however, a native cloning tool available on the RHEL 6 KVM host. However, it will likely be faster to use Kickstart to deploy new RHEL 6 KVM guests rather than use the native `virt-clone` tool.

Kickstart

For complete explanation and deployment steps to deploy and use Kickstart, refer to the Red Hat Enterprise Linux 6 Deployment Guide.

6 Management Best Practices

Any management servers in this section can all be virtualized on a separate group of infrastructure hosts. By virtualizing the management servers, they gain the same benefits as the production virtual machines such as mobility, availability, and centralized data management on the NetApp controller(s).

6.1 libvirt

The libvirt library can be accessed using command line wrapper in two methods:

1. Using the virsh command prior to a directive, such as:

```
virsh start <guest_vm>
```

2. Using virsh as a shell environment:

```
virsh
```

   Note: Running `virsh` without any options will result in a virtualization specific shell.

   Note: Additional information on `virsh` can be found in the RHEL 6 Virtualization Guide.

3. The libvirt library can also be used as a portable API. Additional information can be found at ibvirt.org.

6.2 Virtual Machine Manager

1. To launch the graphical Virtual Machine Manager, type the following command:

```
virt-manager &
```

6.3 Configuring Remote Administration Host (Optional)

This section discusses the steps necessary to configure a remote administration host.

1. Deploy a desktop, virtual machine, or lightweight server that follows the steps and requirements of an RHEL 6 KVM host, with the following exceptions:
   - The libvirtd daemon does not need to be enabled or started as it will not be used as a hypervisor.
   - The addition of the Gnome Desktop Environment packages (or other GUI environment).
   - With the exception of the CPU extensions, the hardware requirements are much lighter.
2. Follow the security best practices provided for RHEL 6 KVM hosts, including disabling unnecessary services, avoiding insecure communication protocols, and setting up host keys.

6.4 RHN and RHN Satellite

Red Hat Network Satellite can be deployed as a virtual machine within the RHEL 6 KVM environment.

For detailed information on Red Hat Network, visit:
access.redhat.com/knowledge/docs/Red_Hat_Network/.

For detailed information, including deployment procedures, for Red Hat Network Satellite server, refer to the documents available at access.redhat.com/knowledge/docs/Red_Hat_Network_Satellite/.

6.5 NetApp Operations Manager

Operations Manager

NetApp Operations Manager can be deployed as a virtual machine within the RHEL 6 KVM environment. For detailed deployment instructions and best practices, see the Operation Manager, Provisioning Manager, and Protection Manager documents referenced in the appendix.

Kickstart Server

The Kickstart server can be deployed as a virtual machine within the RHEL 6 KVM environment.

See the Red Hat Enterprise Linux 6 Deployment Guide for information on setting up and using Kickstart.

Scaling Out and Separating the Environment

In this section, MultiStore and vFiler are used to provide scalable and mobile virtual storage controllers. The following procedure creates an IP space and assigns a designated interface to that IP space. All commands in this procedure are run from the Data ONTAP CLI.

1. Install the MultiStore license.
2. Create an IP space for a vFiler unit or multiple vFiler units, for example, `kvm_ipspace01`:

```
ipspace create kvm_ipspace01
```

3. Assign an interface that will be used for a vFiler unit to the newly created IP space:

```
ipspace assign kvm_ipspace01 vif03
```

The following procedure provisions one nondefault vFiler unit with a dedicated IP space, and one additional data volume also needs to be assigned to the vFiler unit.

4. Create a primary storage unit for the vFiler unit:

```
vol create <vfiler01_rootvol> <aggr_name> 256m
```

5. Create a vFiler unit and specify the root volume and IP space that will be used.

   Note: The IP address used by the vFiler unit must not be configured when the vFiler unit is created.

   Note: Quotas must be turned off before assigning a qtree or volume to a vFiler unit; they may be turned back on after the resource is assigned.

   Note: After assigning an IP space to a vFiler unit, the IP space cannot be changed without destroying the vFiler unit.

```
vfiler create kvm_vfiler01 -n -s kvm_ipspace01 -i vif03 /vol/var_vfiler01_rootvol
```

6. Disallow rsh and any other protocols not needed by the newly created vFiler unit by using the vfiler disallow command.

The following command disables rsh, cifs, iscsi, http, and ftp on the vFiler unit:

```
vfiler disallow kvm_vfiler01 proto=rsh proto=cifs proto=iscsi proto=http proto=ftp
```

7. Use the ifconfig command on the hosting storage system to configure the interface as Up with the IP address specified during the creation of the vFiler unit.

   Note: The IP address used here must be the same address used in the vfiler create command.

```
ifconfig vif03 <ip> netmask <netmask> partner <partner_interface>
```

8. Modify the routing table of the IP space the vFiler unit is using with the vfiler run command. This command adds a default route for the newly created IP space that the Vfiler unit is using:

```
vfiler run kvm_vfiler01 route add default <gateway_ip> 1
```

9. Add an additional data volume to the new vFiler unit:

```
vfiler add kvm_vfiler01 /vol/kvm_vfiler01_datavol
```

10. Apply the security best practices to the vFiler unit.

7 Deploying a Backup Framework for Red Hat KVM

7.1 Configuring Snapshot

The following procedure configures automatic Snapshot copies on a designated FlexVol volume. This example uses a FlexVol volume named `vol01`.

Note: These procedures might not be appropriate for creating Snapshot copies of volumes that contain LUNs. Use SnapDrive® and/or one of the SnapManager® products to create Snapshot copies of LUN data to make sure that the LUN is in a recoverable state before the Snapshot copy is created.

1. Turn off the nosnap option on the volume:

```
vol options <volume_name> nosnap off
```

2. Set the Snapshot reserve to an appropriate value. This example sets the Snapshot reserve to 20% of the total size of vol01:

```
snap reserve <volume_name> 20
```

Note: This reserve should be set based on the estimated rate of change for the volume.

Note: Set the reserve to 0 if this volume contains LUNs.

3. Create an automatic Snapshot schedule on the volume as necessary.
4. Use the `snap sched` command to assign an automatic Snapshot schedule. Snapshot schedule and retention requirements should be determined with the customer or business user:

```
snap sched [-A | -V] [<vol-name> [weeks [days [hours[@<list>]]]]]
```

   For example, the following command schedules automatic Snapshot copies on `vol01` to retain 4 weekly Snapshot copies, 7 daily Snapshot copies, and 8 hourly Snapshot copies that occur at 8 a.m., 12 p.m., 4 p.m., and 8 p.m.:

```
snap sched <volume_name> 4 7 8@8,12,16,20
```

Note: Weekly Snapshot copies occur at midnight on Sundays, daily Snapshot copies occur at midnight every day, and hourly Snapshot copies occur either on the hour or as configured by the optional <list> argument as seen in the previous example. For additional information, refer to the “Data Protection Online Backup and Recovery Guide” for your version of Data ONTAP at the NetApp Support (formerly NOW®) site.

Accounting for XML Descriptor Files in the Backup Strategy

Backing up the XML descriptor files is a critical piece in backup, disaster recovery, and site failover situations. By default, the XML descriptor files are located in /etc/libvirt/qemu, which is not part of the shared storage. Although there are several ways to overcome this, perhaps the easiest is to simply create a cron job that copies the contents of /etc/libvirt/qemu to a directory on the shared storage.

7.2 Snap Creator 3.x Preinstallation

Downloading the NetApp Snap Creator Framework

NetApp Snap Creator Framework can be downloaded from the NOW site at now.netapp.com.

1. Snap Creator is listed in the software download section under Snap Creator Framework.
2. Select the version of Snap Creator to be downloaded by selecting View & Download.
3. At the bottom of the page under the Software download instructions, click CONTINUE.
4. Read and accept the end-user license agreement (EULA) by clicking Accept.
5. Finally, select the operating system and bit level of the software package(s) to be downloaded.

Creating a Data ONTAP User for Snap Creator

NetApp recommends creating a new role, group, and user for the use of the NetApp Snap Creator Framework. The structure is that the role will be assigned to the group, and the group will contain the user. To complete this process, the following steps need to be completed using the Data ONTAP command line interface.

Note: Copying and pasting commands from this document might result in incorrectly transferred characters caused by end-of-line breaks or carriage returns. Such errors cause the commands to fail. It is recommended that you copy and paste the given commands into a text editor prior to entering on the array, so the characters can be verified and corrected before being pasted into the NetApp CLI console.

1. Create a role defining the rights required for Snap Creator on the storage system. Although it is possible to further restrict the user based on the features that will be used (for example, if SnapMirror will not be used, then api-snapmirror-* will not be needed), the following commands include all of the API roles used by Snap Creator. Execute the following commands on the command line:

```
useradmin role add <rolename> -a login-*,api-snapshot-*,api-system-*,api-ems-*,api-snapvault-
*,api-snapmirror-*,api-volume-*,api-lun-*,api-cg-*,api-nfs-*,api-filer-*,api-file-*,api-license-*
```

Where `<rolename>` is the name of the new role.

2. Create a new group on the storage system and assign the previously created role to the group. Enter the following command on each storage controller:

```
useradmin group add <groupname> -r <rolename>
```

Where `<groupname>` is the name of the new group that is being added.

3. Create a user account in the previously created group that will be used for Snap Creator. Enter the following command on each storage controller:

```
useradmin user add <username> -g <groupname>
```

   Note: After the user add command is entered, the system will prompt for a password for the account.

4. Use this restricted account when creating configuration files for Snap Creator.

Note: These steps need to be performed once for each controller where Snap Creator will be used.

Installing/Verifying Java Installation on Snap Creator Server

Java® Runtime Environment (JRE) 1.6 or higher must be installed on the Snap Creator server. A Red Hat–compliant JRE can be downloaded from Red Hat Network. The version of Java installed needs to match the version of Snap Creator installed.

1. The version of java can be verified by typing the following at a command prompt:

```
java –version
```

2. The output of the preceding command will list the installed version of Java. If the Java installation is 64-bit, that will be listed as well. If there is no mention of the bit level, then the installation is 32-bit.

7.3 Snap Creator 3.x Server

This section covers setting up and starting the Snap Creator server and the associated graphical user interface (GUI) for the Snap Creator Framework.

Note: Although the Snap Creator server can be deployed on a virtual machine, it needs to exist on a server outside of the infrastructure that it will back up. Otherwise, it is possible that the Snap Creator server could quiesce the server or virtual machine on which it is running without the ability to resume itself.

Snap Creator Server Setup on Red Hat Enterprise Linux 6

1. Configure the Snap Creator server. This can be done by changing directories into the scServer<version#> subdirectory, then executing the following command:

```
./snapcreator --profile setup
```

Note: The Snap Creator executable should already be configured upon extraction with proper permissions to be executed. If for some reason the preceding command does not work, the permissions might need to be added. This can be done by executing the following command:

```
chmod 755 snapcreator
```

Executing the preceding command will start the Snap Creator setup. The first screen will display the end-user license agreement (EULA). At the end of the EULA a prompt will appear asking for EULA acceptance. The following console block is indicative of what might be displayed (without the EULA text):

```
Do you accept the End User License Agreement (y|n):
```

2. Type y to accept the EULA and press Enter to continue with the setup.

Note: In the next step, you need to give confirmation for setting up Snap Creator. The following console block is indicative of what might be displayed:

```
Setup NetApp Snap Creator Framework 3.5.0 Server (y|n):
```

3. Type y and press Enter to continue with the setup.

A prompt will appear asking for the serial number of the storage system that will be used with Snap Creator. This is an optional field, but it is helpful later when sending support requests or looking at log files. This prompt looks similar to the following console block:

```
Enter serial number:
```

4. Enter the storage serial number if desired and press Enter to continue.

A prompt will ask if the GUI job monitor should be enabled. The GUI job monitor is new as of Snap Creator 3.5.0. The job monitor lists all of the jobs that have been run on the Snap Creator server and provides an easy way to determine if jobs are completing properly. The prompt looks similar to the following console block:

```
Enable GUI job monitor (Y|N):
```

5. Select either y to enable the job monitor or n to disable the job monitor, then press Enter to continue.

The next prompt only appears if the job monitor is enabled. When the job monitor is enabled, a prompt is presented to choose the size of the job monitor or how many jobs remain in the list. Windows defaults to 100 jobs. The prompt looks similar to the following prompt in the console block:

```
Enter job monitor size, how many jobs to allow:
```

6. Enter the number of jobs for the job monitor, then press Enter to continue.

The next prompt looks for the user name for the administrative user for the GUI. The prompt looks similar to the prompt in the following console block:

```
Please Enter GUI Administrator Username:
```

7. Type the user name and press Enter to continue.

The next prompt looks for the password for the administrative GUI user. The prompt looks similar to the prompt in the following console block:

```
Please Enter password for admin:
```

8. Enter the password for the GUI administrative user and press Enter to continue.

   Note: When typing the password, no characters appear on screen. This is by design.

The final prompt looks for confirming the password typed in the previous step. The prompt looks similar to the prompt in the following console block:

```
Please Confirm password for admin:
```

9. Retype the password for the GUI administrative user and press Enter to continue.
10. Open firewall port 8080 on the Snap Creator server IPtables firewall.

Starting the Snap Creator Web Console

The following steps describe how to start the Snap Creator Web console.

1. Change directories to `<path>/<to>/<scServer_v#>/gui`. For example, if Snap Creator is installed at /SC_3.5/scServer3.5.0c, use the cd command to change directories to `/SC_3.5/scServer3.5.0c/gui`.
2. Start the GUI by executing the following command:

```
java -jar snapcreator.jar
```

3. To open the Snap Creator GUI, open a Web browser and point the browser to http://hostname.com:8080, where:
   - `HostName` is the host name or IP address of the Snap Creator server.
   - `Port` is the port number where the Snap Creator server is running. By default this is port 8080.
4. Log in to Snap Creator using the credentials that were supplied during the setup process.
5. It is also recommended to create a start/stop script for the Snap Creator GUI. See the appendix for an example.

7.4 Snap Creator 3.x Agent

This section covers configuring and starting the Snap Creator agent.

Snap Creator Agent Configuration

1. Log in to the hypervisor or virtual machine that is running the agent.
2. Edit the `/opt/scAgent*/config/agent.conf file` and add each precommand and postcommand that the agent is authorized to run, one per line. For example:

```
command:sync
command:virsh *
command:/path/to/some/application
command:/path/to/some/script
```

The commands can be RHEL commands, application-specific commands, or even custom scripts.

Hosts

By default the Snap Creator Framework agent allows communications with any Snap Creator Framework server, but communications can be limited to a particular server. This is done by changing the host line in the agent.conf file. The default host entry in the agent.conf file is:

```
host: scServer@*
```

The wildcard entry, *, instructs Snap Creator to allow anything. The wildcard can be replaced with a host name or IP address to restrict communications to a particular Snap Creator server.

Starting the Snap Creator Agent on Windows

When installing Snap Creator on Windows, options are provided to install the Snap Creator agent and start it as a service. If the appropriate option was selected at install, then the Snap Creator service will already be started. The port that the Snap Creator agent will use to communicate with the Snap Creator server was selected at install. In the event that the Snap Creator agent service needs to be managed through the Windows Services plug-in, the installed name is SnapCreatorAgentService.

Starting the Snap Creator Agent on Red Hat Enterprise Linux

To start the Snap Creator agent on RHEL, execute the following commands for the initial setup:

1. To configure the Snap Creator agent change directories into the /<path>/<to>/scAgent_v<#> subdirectory, execute the following command:

```
./snapcreator --profile setup
```

Note: The Snap Creator executable should already be configured upon extraction with the proper permissions to be executed. If for some reason the preceding command does not work, the permissions might need to be added. This can be done by executing the following command:

```
chmod 755 snapcreator
```

Note: Executing the preceding command will start the Snap Creator setup. The first screen will display the end-user license agreement (EULA). At the end of the EULA a prompt will display for EULA acceptance. The following console block is indicative of what might be displayed (without the EULA text):

```
Do you accept the End User License Agreement (y|n):
```

2. Type y to accept the EULA and press Enter to continue with setup.

Note: In the next step, you need to give confirmation for setting up the Snap Creator server. In this instance the agent needs to be configured, not the server. The following console block is indicative of what might be displayed:

```
Setup NetApp Snap Creator Framework 3.5.0 Server (y|n):
```

3. Type n and press Enter to continue with the setup.

   Note: In the next step, you need to give confirmation for setting up the Snap Creator agent. The following console block is indicative of what might be displayed:

```
Setup NetApp Snap Creator Framework 3.5.0 Agent (y|n):
```

4. Type y and press Enter to continue with the setup.

   Note: This updates the environmental variables so that the Snap Creator agent scripts will work properly. The usage information for the agent will appear on screen.

5. Follow the on-screen information to start the Snap Creator agent. For reference, common commands are:

To start the Snap Creator agent:

```
/path/to/scAgent_v<#>/bin/scAgent start
```

To stop the Snap Creator agent:

```
/path/to/scAgent_v<#>/bin/scAgent stop
```

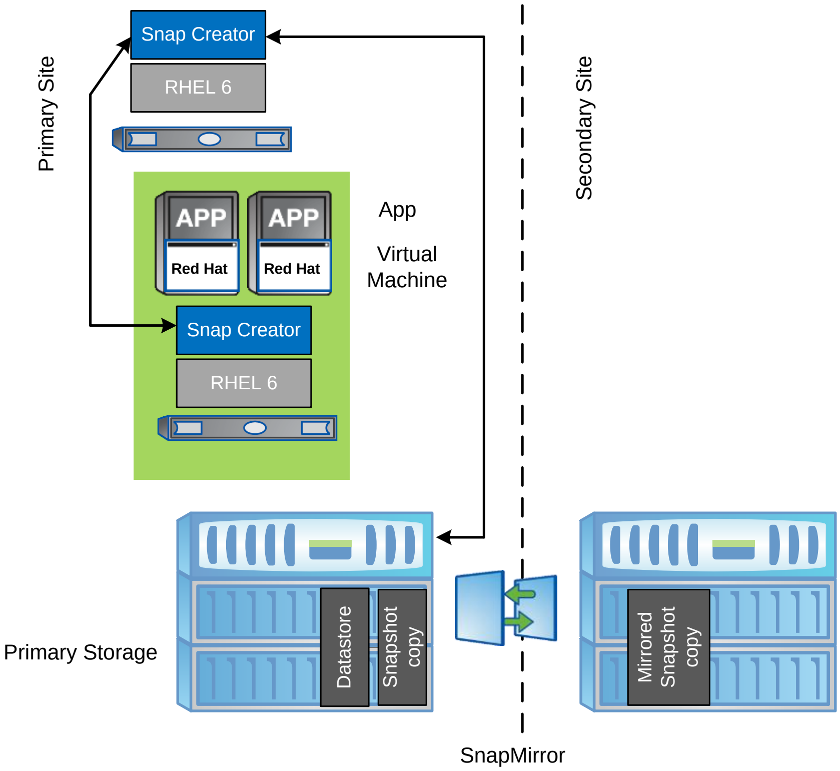
To check the status of the Snap Creator agent:

```
/path/to/scAgent_v<#>/bin/scAgent status
```

6. Open the IPtables firewall on the Snap Creator server to allow access to port 9090. If the Snap Creator server is a virtual machine, then the port needs to be opened on the virtual machine and the hypervisor.

Note: It is also recommended to create a script to start and stop the Snap Creator agent automatically. An example is provided in the appendix.

Figure 3) Example of Snap Creator server and agent layout.



7.5 Configuring a Snap Creator Profile for RHEL 6 KVM

Base Configuration for Red Hat KVM

1. Open a Web browser to the following URL: http://myserver.mydomain.com:8080 (be sure to replace myserver.mydomain with your Snap Creator portal URL). Log in to the site with the user name and password created when the server profile was created.
2. Select Management -> Configurations, then click the + button under Backup Profiles, and enter a name for the profile.
3. Under Backup Profiles, right-click the newly created profile and select New Configuration to launch the configuration wizard. Click Next.
4. Enter a Config Name and select both Password Encryption and Advanced Options. Click Next.
5. Select Standard Configuration. (Plug-In Configuration is not yet supported for the KVM plug-in.) Click Next.
6. Select None. Click Next.
7. Enter the IP address, the agent port (9090), and a timeout of 10. Click Test agent connection. When the test succeeds, click ok. Click Next.
8. Enter the IP address of the NetApp controller, enter login information, and select the transport protocol. If secure Web access is enabled on the NetApp controller, select HTTPS; otherwise, select HTTP. Click Next.
9. Select the volume(s) to be backed up by the profile. Click Next.

10. Enter a name for the Snapshot copies, a policy name, and select either Recent or Timestamp for naming convention. Select the number of Snapshot copies to keep and how many days to keep each Snapshot copy. Click Next.
11. Select Operations Manager Alert. Enter the IP and login information for the Operations Manager server. Click Next.
12. Click Finish.

Configuration Before and After Creating Snapshot Copies for Red Hat KVM

1. Open a Web browser to http://myserver.mydomain.com:8080 (be sure to replace myserver.mydomain with your Snap Creator portal URL) replacing the host and domain with appropriate values.
2. Log in to the site with the user name and password created when the server profile was created.
3. Select Management>Configurations and then select the profile and configuration to be edited.
4. Scroll to the bottom of the configuration to the PRE/POST section.
5. To quiesce an application and/or virtual machine prior to the Snapshot copy, click the + button below the Application Quiesce Command subsection. Enter the command(s) to be run. If there are multiple commands to be run, they need to be listed in proper order.

   Note: Make note of the command(s) to be run. Any command(s) will need to be added to the list of allowed actions for the Snap Creator agent in the next section.
6. To resume an application and/or virtual machine after the Snapshot copy, click the + button below the Application Un-Quiesce Command subsection. Enter the command(s) to be run. If there are multiple commands to be run, they need to be listed in proper order.
7. Make note of the command(s) to be run. Any command(s) will need to be added to the list of allowed actions for the Snap Creator agent in the next section.
8. Scroll to the top of the configuration and click the diskette icon to save the updated configuration.

Traditional Backup Methods

Although the use of Snapshot copies, Snap Creator, and SnapMirror is a best practice, it is not a requirement for Red Hat enterprise virtualization. However, NetApp recommends using some form of data backup as a key foundation piece of enterprise data protection.

NetApp also provides two means of data backup that are included in Data ONTAP and that do not require any additional licenses. The `dump` and `ndmpcopy` tools are available to replicate data to tape drives or to other storage, respectively. This will satisfy the requirement for backup utilities in Red Hat virtualization.

8 Deploying and Configuring Disaster Recovery for RHEL 6 KVM

NetApp SnapMirror is used in conjunction with NetApp Snap Creator for disaster recovery.

Verify that the following SnapMirror prerequisites have been met:

- A valid SnapMirror license must be available. If the SnapMirror source and destination are on different systems, then license SnapMirror on each system.
- For SnapMirror volume replication, the destination system must use a version of Data ONTAP that is the same as or later than the SnapMirror source system. To configure volume SnapMirror to support replication for disaster recovery, both the source and destination system must use the same version of Data ONTAP.
- The name and IP address of the source system must be in the `/etc/hosts` file of the destination system or must be resolvable through the DNS or by using the `yp` command.

- For optimal SnapMirror volume replication performance, when using traditional volumes, make sure that the SnapMirror source volume and destination volume contain disks of the same size, organized in the same RAID configuration.

8.1 Installing and Configuring SnapMirror

The following procedure enables and configures SnapMirror on two controllers and authorizes the destination controller to receive SnapMirror updates from the source controller.

1. Verify that both the source and destination controller names and IP addresses are in the `/etc/hosts` file on each controller participating in replication.
2. Add the SnapMirror license on both the source and destination storage systems:

```
license add <LICENSE_KEY>
```

3. Enable SnapMirror on both the source and destination storage systems:

```
options snapmirror.enable on
```

4. Configure SnapMirror access from the SnapMirror source storage controller. On the source system, authorize the destination systems that will receive the SnapMirror transfers. In this example, netapp02.example.com is the destination storage controller:

```
options snapmirror.access host=netapp02.example.com
```

Note: This procedure assumes that both of the source and destination volumes have already been created.

5. From the SnapMirror destination system, restrict the destination volume using the `vol restrict` command:

```
vol restrict /vol/dest_flexvol
```

6. For each source volume from the SnapMirror destination system, perform an initial baseline transfer using the `snapmirror initialize` command:

Note: This command constantly runs from the destination storage system.

```
snapmirror initialize –S netapp01:src_flexvol netapp02:dest_flexvol
```

Note: This command does not schedule automatic updates. It only allows manual updates to the volume.

7. From the SnapMirror destination system, verify the status of the initial transfer from the destination storage system using the `snapmirror status` command**:**

```
snapmirror status
```

8. From the SnapMirror destination system, edit or create the `/etc/snapmirror.conf` file and run the following command to allow further updates to the volume:

Note: This command does not schedule automatic updates. It only allows manual updates to the volume.

```
# /etc/snapmirror.conf file
netapp01:src_flexvol     netapp02:dest_flexvol - 0 22 * *
```

Note: The preceding example runs the SnapMirror for the src_flexvol volume every night at 10 p.m.

The schedule consists of four space-separated fields in the following order:

- Minute can be a value from 0 through 59.
- Hour can be a value from 0 through 23.

- Day of month can be a value from 1 through 31.
- Day of week can be a value from 0 (Sunday) through 6 (Saturday).

Note: A single dash (-) in any field means never and prevents the schedule entry from executing.

For a complete description of SnapMirror, including the `snapmirror.conf` syntax with examples and available arguments, see the Data Protection Online Backup and Recovery Guide.

9 Site Failover for RHEL 6 KVM and NetApp Storage

9.1 Deploying SnapMirror Async for Site Failover

The following procedures assume that the primary site has already been set up, the secondary NetApp FAS controller has been set up, and a SnapMirror relationship (DR) has also been configured between the two NetApp FAS controllers. This configuration can be done at the same time as the primary site or easily added on at a later date.

Procedures for Deploying SnapMirror for Site Failover

1. Configure the network in such a manner that both sites have the same subnets, IP spaces, and VLANs. If possible, the sites should have full Layer 2 and Layer 3 connectivity, but it is only a requirement for the SnapMirror relationship.
2. Configure the RHEL 6 KVM hosts at the secondary site. The virtual bridges, VLANs, subnets, and IP spaces must be identical to the primary site.
3. Configure failover targets (IP network). This means that the secondary controller must have the means to host the same IP addresses as the primary controller to support the same storage targets. This can typically be done by way of creating aliases on the existing ifgrps. These failover targets will remain down until they are needed in a site failover. After establishing the SnapMirror relationship, be sure to copy over NFS export permissions. Similar to the failover IPs, these exports should not be active except in the case of a site failover.

   Note: If the secondary controller carries its own production workload in addition to the role of secondary site RHEL 6 KVM storage, then additional NICs might need to be added to the NetApp storage controller. Otherwise, configure aliases on the existing NICs to handle the IP traffic in a failover situation. In either case, these failover IPs will remain dormant until a failover situation arises.

4. Configure zoning and igroups. Unlike IP networking, FC zoning is not shared between the sites. RHEL 6 KVM hosts at the primary site are to remain zoned for the primary NetApp FAS controller. RHEL 6 KVM hosts at the secondary site should only be zoned for the secondary NetApp FAS controller. The same rules apply to igroups for FC and iSCSI. However, be sure to properly map the secondary igroups to the right LUNs.
5. Query and log all LUN serial numbers on the primary and secondary controllers. When a volume is copied with SnapMirror, the LUNs at the secondary site will have a different serial number. In a site failover situation, they need to be changed. Prior to giveback, the original serial numbers need to be returned.
6. Run the following command to query the serial number for a given LUN:

```
lun serial /vol/InfraVMVol/vmlun
```

   Note: This is a critical step in the process. Log the serial numbers for all LUNs and keep copies at the secondary site.

   Note: It is crucial that the XML descriptor files (guest configuration) in /etc/libvirt/qemu be backed up. A simple way to do this would be to create an XML subdirectory under /var/lib/libvirt/images and schedule a cron job to copy the configuration files over. Then they will be part of the NetApp Snapshot copies and SnapMirror syncs. This will facilitate that in a failover situation, the infrastructure VMs can be properly and quickly imported.

Note: This is a critical step in the process.

7. The secondary hypervisor nodes must have the same IP connectivity (bridges, VLANs, and so on) as the ones at the primary site.

Procedures for Site Failover with SnapMirror Async

Note: Primary site is declared down.

1. Isolate IP traffic from primary site in case it comes back online while the DR site is running.
2. Break the SnapMirror relationship on the secondary NetApp FAS controller to make the volumes read-write:

```
snapmirror break name_of_flexvol
```

3. If using LUNs, bring the LUN(s) offline and make note of the original serial number. Change serial numbers on LUNs to the serial numbers from the LUNs at the primary site. Finally, bring it back online:

```
lun offline /vol/InfraVMVol/vmlun
lun serial /vol/InfraVMVol/vmlun
             Serial#: P3OoG4WVLFoa
lun serial /vol/InfraVMVol/vmvol <new_serial_#>
lun online /vol/InfraVMVol/vmlun
```

4. Bring up failover ifgrps on the secondary NetApp controller.
5. Bring up the RHEL 6 KVM hosts at the secondary site.
6. Attach the NetApp storage to the RHEL 6 KVM hosts and confirm that it is read-write and that all of the virtual machines are visible. Copy all of the saved XML descriptor files back to their default location of /etc/libvirt/qemu.
7. Import (define) each of the virtual machines into the secondary RHEL 6 KVM hosts by running the following command:

```
virsh define /etc/libvirt/qemu/<name_of_guest>.xml
```

   Note: This can easily be scripted into a for loop to quickly iterate through a large number of virtual machines quickly.

8. Start the RHEL 6 KVM guests:

```
virsh start <name_of_guest>
```

Note: This can easily be scripted into a for loop in order to quickly iterate through a large number of virtual machines quickly.

Note: Failover is complete.

Procedures for Site Giveback with SnapMirror Async

Note: When the primary site is back up and ready to operate again, be sure that it is still isolated.

1. Bring down the RHEL 6 KVM guests.
2. Unmount the NetApp storage from the RHEL 6 KVM hosts at the secondary site.
3. Bring down failover IPs and ifgrps on the NetApp controller at the secondary site.
4. If using LUNs, change serial numbers on LUNs back to original:

```
lun offline /vol/InfraVMVol/vmlun
lun serial /vol/InfraVMVol/vmvol <orig_serial_#>
lun online /vol/InfraVMVol/vmlun
```

5. Next, resync the volumes. From the NetApp controller at the primary site, run the following command: from the primary controller:

```
snapmirror resync -S ice3170-3b:InfraVMVol ice3170-3a:InfraVMVol
```

From the NetApp controller at the secondary site, run the following command:

```
snapmirror resync -S ice3170-3a:InfraVMVol ice3170-3b:InfraVMVol
```

6. Bring the full network back online.
7. Attach the NetApp storage to the RHEL 6 KVM hosts at the primary site and verify that they are read-write.
8. If any new RHEL 6 KVM guests were created at the secondary site, they need to be imported (defined) at the primary site.
9. Start the RHEL 6 KVM guests.

Note: Giveback is complete.

9.2 Deploying MetroCluster for Site Failover

MetroCluster is fairly straightforward to deploy when an environment is first being set up, but can be challenging to deploy as part of an existing environment.

Configuring MetroCluster

1. Configure the network. All Layer 2 and Layer 3 networking between the sites must be shared. That is to say that IP spaces, VLANs, and VIFs should be accessible between the sites.
2. Configure the hardware. MetroCluster has specific hardware requirements and specific hardware configuration that are outside of the scope of this document. It is likely that a NetApp professional services consultant will implement the hardware portion of the configuration.
3. Configure partner IPs on each NetApp FAS controller for each network interface, VLAN, and ifgrp.
4. Configure zoning and igroups. Zoning and igroups do not need special configuration. Storage zoning and NetApp igroups should be site-specific.
5. Create a mirrored aggregate:

```
aggr create aggr1 -m 12
```

This will create a mirrored aggregate named aggr1 with 12 disks. Six disks will be on the primary controller, and 6 disks will be on the secondary controller.

6. Create FlexVol volumes, NFS exports, and LUNs using standard NetApp best practices.
7. Disable the changing of LUN serial numbers on forced takeover:

```
options cf.takeover.change_fsid off
```

This will alleviate the need to change the LUN serial numbers.

8. Create RHEL 6 KVM hosts as prescribed in this document.
9. Create RHEL 6 KVM guests as prescribed in this document.
10. RHEL 6 KVM hosts and guests can operate at both sites.
11. It is crucial that the XML descriptor files (guest configuration) in /etc/libvirt/qemu be backed up. A simple way to do this would be to create an XML subdirectory under /var/lib/libvirt/images and schedule a cron job to copy the configuration files over. Then they will be part of the NetApp Snapshot copies and SnapMirror syncs. This will make sure that in a failover situation the infrastructure VMs can be properly and quickly imported.

Note: This is a critical step in the process.

Procedures for Site Failover with MetroCluster

8. The primary site fails (total loss of site power, catastrophe, and so on) and the secondary FAS controller recognizes that it no longer has connectivity with any part of the primary storage.
9. Isolate the primary site controller from the secondary site in case it comes back up. The easiest way to do this is to turn off power to the controller (leaving power on to the shelves). Otherwise, disconnect the cluster interconnect at the secondary site.

   Note: This is a critical step that has serious consequences if not followed. If the primary storage comes back up, it is required to service storage requests at the same time as the secondary controller.

10. Initiate a forced (manual) takeover on the secondary storage:

```
cf forcetakeover -d
```

   Note: If the RHEL 6 KVM hosts are still operational, then they can still reach the secondary storage. However, if the distance is great, then latency might be an issue. For this reason, the following procedures assume that everything is being failed over.

11. Any RHEL 6 KVM guests that were brought down when the primary site failed can be brought up at the secondary site without having to import them.

Note: The failover is complete.

Procedures for Site Giveback with MetroCluster

Caution

When the primary site is brought back online, do not bring the primary NetApp controller up yet, only the disk shelves.

1. Save the state of all RHEL 6 KVM hosts and guests.
2. Pause (suspend) all RHEL 6 KVM guests.
3. Detach the NetApp storage from the RHEL 6 KVM hosts.
4. Bring up the primary disk shelves and properly reestablish aggregate mirrors and make sure that they are synced. Only after this has been confirmed can the primary controller be brought back up.
   a. Confirm that the remote storage can be seen from the secondary controller:

```
aggr status -r
```

   b. Go into partner mode on the secondary controller:

```
partner
```

   c. From within partner mode, determine which aggregates are at what site and which aggregates are out of sync:

```
aggr status -r
```

   d. Before resyncing aggregates, take the secondary site aggregates offline:

```
aggr offline aggr1
```

   e. Resync the aggregates:

```
aggr mirror aggr1 -v aggr1
```

   Note: Before continuing, be sure ALL aggregates are resynced.

5. The primary controller will pause its own boot procedure until a giveback is initiated from the DR controller.

Caution
Do not initiate the giveback until the following steps are completed.

6. Initiate the giveback from the secondary NetApp FAS controller:

```
cf giveback
```

Note: The secondary NetApp FAS controller will reboot.

7. Bring up the RHEL 6 KVM hosts at the primary site and mount the storage.
8. Verify that all networking is restored properly.
9. Start the RHEL 6 KVM guests.

Note: Giveback is complete.

Appendixes

Ports to Allow Through Firewall

Table 1 describes the ports that need to be allowed through the firewall on an RHEL 6 KVM.

Table 1) Ports allowed through firewall.

Port	Protocol	Description
22	TCP	SSH
80, 443	TCP	HTTP, HTTPS
111	TCP, UDP	Portmap
123	TCP	NTP
16514	TCP	libvirt
3260	TCP,UDP	iSCSI (optional)
53	TCP, UDP	DNS
5353	TCP, UDP	mDNS
54321	TCP	KVM Inter-host Communication
5900-5910 (range can be increased)	TCP	VNC consoles (optional)
32803, 662	TCP	NFS Client
49152-49216	TCP	KVM Migration
67, 68	TCP, UDP	DHCP
8080	TCP	Snap Creator & Operations Manager Portals
8088	TCP	NetApp Management Console
8443	TCP	Secure Operations Manager Console
8488	TCP	Secure NetApp Management Console
9090	TCP	Snap Creator Agent
N/A	N/A	ICMP

Sample Start/Stop Script for Snap Creator Portal

1. Create the file sc_server in /etc/init.d with the permissions 755.
2. Run the following command to enable the script on boot:

```
chkconfig –add sc_portal; chkconfig sc_portal on
```

3. Copy the console block, provided in the following box, into the file.

```
#!/bin/sh
#
# scportal:    NetApp Snap Creator GUI
#
# chkconfig:   2345 99 1
# description: A script to start the java jetty server (GUI) Snap Creator Server
#

# Source function library.
. /etc/rc.d/init.d/functions
# edit as necessary
SC_GUI_PATH=/opt/scServer3.4.0/gui
JAR_FILE=snapcreator.jar

start()
{
        echo -n $"Starting scportal: "
        cd $SC_GUI_PATH
        daemon /usr/bin/java -jar $JAR_FILE --port 8080

        touch /var/lock/subsys/scportal
        echo
}

stop()
{
        echo -n $"Shutting down scportal: "
        kill -kill `ps aux | grep snapcreator.jar | grep -v "grep" | awk {'print $2'}`

        rm -f  /var/lock/subsys/scportal
        echo
}

# See how we were called.
case "$1" in
  start)
        start
        ;;
  stop)
        stop
        ;;
  restart|reload)
        stop
        start
        ;;
  status)
        status hidd
        ;;
  *)
        echo $"Usage: $0 {start|stop|restart|status}"
        exit 1
esac

exit 0
```

Sample Start/Stop Script for Snap Creator Agent

1. Create the file sc_agent in /etc/init.d with the permissions 755.
2. Enable the script to start automatically at boot by typing the following command:

```
chkconfig –add sc_agent; chkconfig sc_agent on
```

3. Copy the console block, provided in the following box, into the file.

```
#!/bin/sh
#
# scagent:     NetApp Snap Creator Agent
#
# chkconfig:   2345 99 1
# description: an agent to assist in creating & syncing NetApp Snapshot copies
#

# Source function library.
. /etc/rc.d/init.d/functions
# edit as necessary
SC_AGENT_PATH=/opt/scServer3.4.0/gui

start()
{
        echo -n $"Starting scagent: "
        daemon $SC_AGENT_PATH/scAgent start

        touch /var/lock/subsys/scagent
        echo
}

stop()
{
        echo -n $"Shutting down scagent: "
        $SC_AGENT_PATH/scAgent stop

        rm -f  /var/lock/subsys/scagent
        echo
}

# See how we were called.
case "$1" in
  start)
        start
        ;;
  stop)
        stop
        ;;
  restart|reload)
        stop
        start
        ;;
  status)
        status hidd
        ;;
  *)
        echo $"Usage: $0 {start|stop|restart|status}"
        exit 1
esac

exit 0
```

References

- Home page for KVM
  www.linux-kvm.org
- Red Hat and Microsoft Virtualization Interoperability
  www.redhat.com/promo/svvp/
- KVM: Kernel-Based Virtual Machine
  www.redhat.com/f/pdf/rhev/DOC-KVM.pdf
- Red Hat Enterprise Linux 6 Virtualization Guide
  docs.redhat.com/docs/en-US/Red_Hat_Enterprise_Linux/6/html/Virtualization_Administration_Guide/index.html
- Red Hat Enterprise Linux 6 Deployment Guide
  docs.redhat.com/docs/en-US/Red_Hat_Enterprise_Linux/6/html/Deployment_Guide/index.html
- Red Hat Enterprise Linux 6 Installation Guide
  docs.redhat.com/docs/en-US/Red_Hat_Enterprise_Linux/6/html/Installation_Guide/index.html
- Red Hat Enterprise Linux 6 Security-Enhanced Linux User Guide
  docs.redhat.com/docs/en-US/Red_Hat_Enterprise_Linux/6/html/Security-Enhanced_Linux/index.html
- TR-3747: Best Practices for File System Alignment in Virtual Environments
  media.netapp.com/documents/tr-3747.pdf
- TR-3427: Storage Best Practices and Resiliency Guide
  media.netapp.com/documents/tr-3437.pdf
- TR-3505: NetApp Deduplication for FAS and V-Series Deployment and Implementation Guide
  media.netapp.com/documents/tr-3505.pdf
- TR-3446: SnapMirror Async Overview and Best Practices Guide
  media.netapp.com/documents/tr-3446.pdf
- TR-3440: Operation Manager, Protection Manager, and Provisioning Manager Sizing Guide
  media.netapp.com/documents/tr-3440.pdf
- TR-3710: Operations Manager, Provisioning Manager, and Protection Manager Best Practices Guide
  media.netapp.com/documents/tr-3710.pdf

NetApp provides no representations or warranties regarding the accuracy, reliability or serviceability of any information or recommendations provided in this publication, or with respect to any results that may be obtained by the use of the information or observance of any recommendations provided herein. The information in this document is distributed AS IS, and the use of this information or the implementation of any recommendations or techniques herein is a customer's responsibility and depends on the customer's ability to evaluate and integrate them into the customer's operational environment. This document and the information contained herein may be used solely in connection with the NetApp products discussed in this document.

Go further, faster®

NetApp®

www.netapp.com