NetApp
Go further, faster

Technical Report

Best Practices for KVM and Red Hat Enterprise Linux on NetApp Storage

Jon Benedict, NetApp
May 2010 | TR-3848

TABLE OF CONTENTS

1 PURPOSE OF THIS DOCUMENT

This technical report discusses the best prescriptive means of setting up a virtual server environment built around the Kernel-Based Virtual Machine (KVM) hypervisor on Red Hat Enterprise Linux® and NetApp® storage. Regardless of the application or applications to be supported, the KVM environment described in this technical report offers a solid foundation.

This technical report underscores the requirements of security, separation, and redundancy. These requirements are emphasized in the three major layers of the environment—server, network, and storage.

1.1 INTENDED AUDIENCE

This document addresses the needs of system architects, system administrators, and storage administrators who are investigating the use of KVM in the data center where NetApp is the intended back-end storage.

1.2 TERMINOLOGY

The following terms are used in this technical report:

- **Host node.** The physical server or servers that host one or more virtual servers.
- **Virtual server.** A guest instance that resides on a host node.
- **Shared storage.** A common pool of disk space, file- or LUN-based, available to two or more host nodes simultaneously.
- **KVM environment.** A general term that encompasses KVM, RHEL, network, and NetApp storage as described in this technical report.
- **Cluster.** A group of related host nodes that support the same virtual servers.
- **Virtual local access network (VLAN).** Useful at layer 2 switching to segregate broadcast domains and to ease the physical elements of managing a network.
- **Virtual interface (VIF).** A means of bonding two or more physical NICs for purposes of redundancy or aggregation.
- **Channel bond.** Red Hat's naming convention for bonding two or more physical NICs for purposes of redundancy or aggregation.

1.3 TOPICS OUT OF SCOPE

Best practices associated with IP and Fibre Channel networks are not covered in this document. However, a solid understanding of these topics is necessary to configure items like VLANs, switched fabrics, and so on.

2 KVM AND RED HAT ENTERPRISE VIRTUALIZATION

2.1 COMPARISON OF KVM AND RHEV

Red Hat Enterprise Virtualization (RHEV) is a full-featured suite of tools based around the KVM hypervisor. It includes a management portal, a small-footprint hypervisor, and tools to simplify the administration of a virtual environment.

In this technical report, KVM is a subset of RHEV. It does not have the management pieces offered by RHEV, but it is easily integrated into an existing environment.

2.2 EXPLANATION OF KVM

KVM is part of the Linux kernel and has been accepted in the upstream since 2007. Neither the host nor the guests require any modification, and the guests run as if they were on bare metal. Because it is part of the Linux kernel, KVM uses the same scheduler and memory management. This also means that as new features are added to the kernel, a KVM host and guest can take advantage immediately instead of having to wait for a specially modified kernel.

Guests under KVM operate as processes just like any other application, service, or script. This means that KVM administrators can use the traditional `top` command to monitor things like utilization and process state. Also, although KVM uses full virtualization, it includes paravirtualized drivers (virtio) for Windows® guests to boost block I/O and network performance.

Red Hat's implementation of KVM includes the KVM kernel module and a processor-specific module for Intel® or AMD, plus QEMU, which is a CPU emulator.

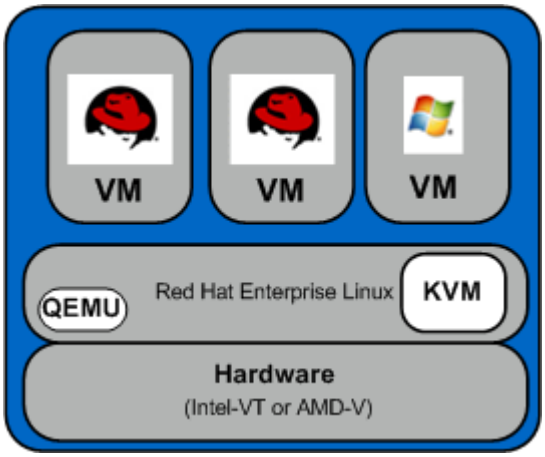

Figure 1) KVM abstraction.

3 SYSTEM REQUIREMENTS

Requirements to launch the hypervisor are conservative; however, overall system performance depends on the nature of the workload.

3.1 MINIMUM SYSTEM REQUIREMENTS

The following list specifies the minimum system requirements:

- 6GB free disk space
- 2GB RAM

3.2 RECOMMENDED SYSTEM CONSIDERATIONS

Although not required, the following list describes system considerations that NetApp strongly recommends:

- One processor core or hyper-thread for each virtualized CPU and one for the hypervisor
- 2GB RAM plus additional RAM virtualized guests
- Some type of 0ut-of-band management (IBM RSA, HP ILO, Dell DRAC, and so on)
- Multiple sets of at least 1GB NICs to separate traffic and allow bonding, or one pair of 10GB NICs to be bonded to carry all traffic
- Fibre Channel or iSCSI HBAs (if using hardware initiators and LUN-based storage)
- Redundant power

3.3 NETWORK REQUIREMENTS

The following list specifies the network requirements:

- Switches capable of VLAN segmentation
- Gigabit Ethernet (or 10GB Ethernet, if available)
- Multiple switches for channel bonding

3.4 KVM REQUIREMENTS

The KVM hypervisor requires a 64-bit Intel processor with the Intel VT extensions or a 64-bit AMD processor with the AMD-V extensions. It may be necessary to first enable the hardware virtualization support from the system BIOS.

Run the following command from within Linux to verify that the CPU virtualization extensions are available.

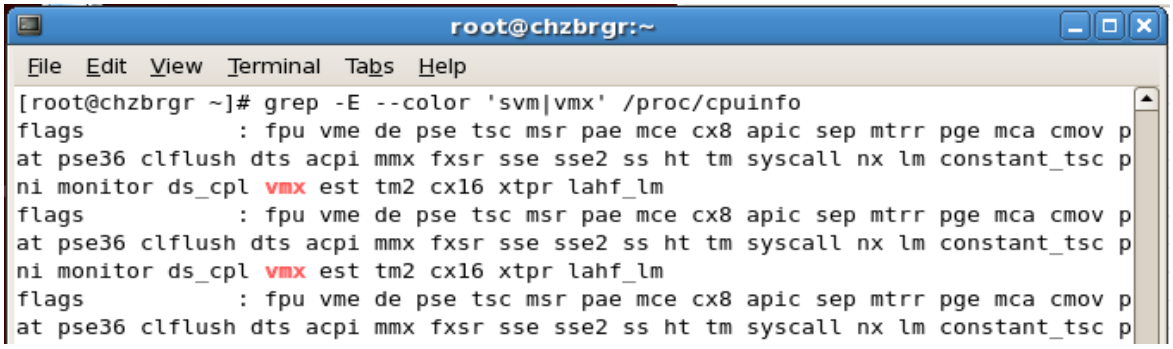

Figure 2) Verify availability of CPU virtualization extensions.

3.5 STORAGE REQUIREMENTS

Whether there are one or many host nodes hosting virtual machines, KVM requires a flexible means of storing virtual systems. KVM supports the following storage types:

- Direct-attached storage
- iSCSI or Fibre Channel LUNs, which may be shared in GFS or GFS2 configurations
- NFS mounted file system

Note: Only the last two storage types on the list can scale to support multiple hosts. Local storage is too easily limited by things like disk capacity, number of disks that can fit, and downtime required to add nodes and capacity.

When a NetApp FAS system is used, the underlying storage gains scalability, flexibility, and fault tolerance. In addition, the underlying storage gains a number of software features such as deduplication and FlexClone® technology, which provide efficient file, LUN, and volume level cloning.

3.6 SUPPORTED GUEST OPERATING SYSTEMS

The following guest operating systems are supported:

- RHEL 3, 4, and 5 (32-bit and 64-bit)
- Windows Server® 2003, Windows Server 2008 (32-bit and 64-bit)
- Windows XP

3.7 KVM HARDWARE LIMITATIONS

The following limitations apply to KVM:

- 256 CPUs per host node
- 16 virtual CPUs per guest
- 8 virtual NICs per guest
- 1TB RAM per host node
- 256GB RAM per guest

3.8 NETWORK ARCHITECTURE

Although this technical report does not discuss the specifics of setting up an Ethernet or switched fabric, the following considerations do need to be addressed.

MULTIPATHING

Regardless of how the network is laid out, the concept of multipathing should be incorporated. That is, if each host node has a public interface and a data interface, each interface should have two or more physical NICs managed by a virtual interface. Red Hat refers to this as a *channel bond.* On the NetApp side, this is referred to as a *VIF (virtual interface).* To extend the redundancy, each path should go to its own switch.

When using Fibre Channel Protocol (FCP), the same best practices apply. Multiple-fibre HBAs should be used on the host nodes. Each fibre path is directed to a separate fibre switch. The hosts manage the multiple paths with Device Mapper Multipath I/O (DM-MPIO), which is also supported for use with iSCSI.

Finally, if the different IP networks are to be combined on a 10GB network, there still need to be multiple 10GB NICs managed by a channel bond, and each path goes to its own switch. The separation is managed by VLAN segmentation, which is described in the following section.

SEPARATION

Keeping the different networks separated is important for both performance and security. The best way to provide this separation is by using VLAN segmentation. Simply put, VLAN segmentation allows a single switch to carry and separate several IP networks simultaneously.

In the case of a host node having separate channel bonds for public and data traffic, the primary paths for each could run to the same switch. In the case of the single 10GB channel bond, VLAN segmentation is the best way to separate the traffic on the same wire.

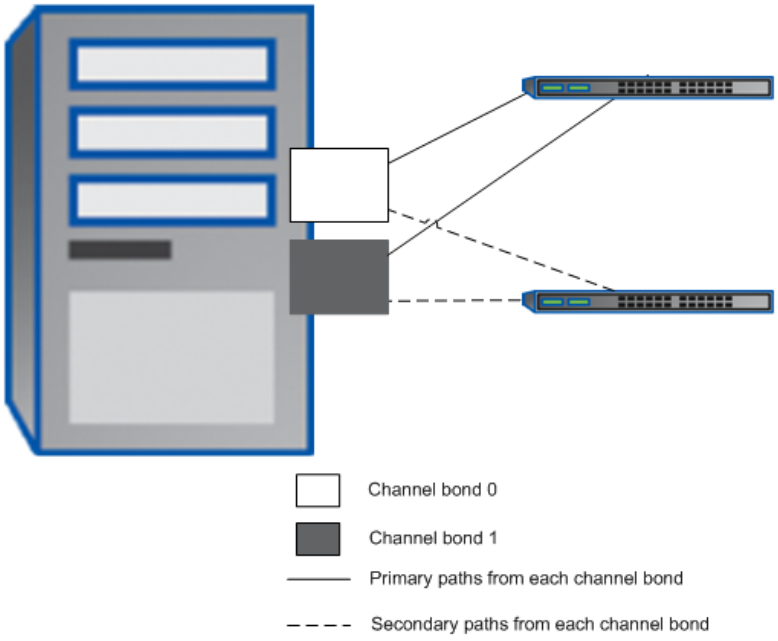

Figure 3) Channel bonds.

The server shown in Figure 3 has access to the public (primary) network as well as to the private (data) network. The NetApp FAS controllers are accessible only from the private network. In addition, there are redundant NICs (or HBAs) and paths for each network managed by a channel bond on the servers. Channel bonds allow multiple physical interfaces to be managed as one virtual interface for purposes of redundancy. The private (data) network in Figure 3 could represent NFS, iSCSI, or FCP.

Note: A bonded pair of 10GB NICs could carry all of the traffic. The separation of public and data traffic occurs by way of VLANs. This separation has the added benefit of faster throughput and fewer cables.

4 BEST PRACTICES FOR NETAPP FAS CONTROLLER CONFIGURATION

4.1 DATA ONTAP VERSION

The NetApp FAS controller should be running Data ONTAP® 7.3.2 or later, in a non-GX mode. For a highly available environment, the storage controllers should be clustered for failover.

Best practice is to use the latest version of Data ONTAP, available from NOW™ (NetApp on the Web). For the hardware, the latest firmware should also be applied to the storage controller, disk shelves, and disk drives.

4.2 AGGREGATES AND VOLUMES

An aggregate is the abstraction layer that NetApp uses to separate physical disks from flexible volumes. Flexible volumes are the logical data structures that contain both block-level and file-level storage. After a flexible volume is created, it can be used to store LUNs or NFS exports.

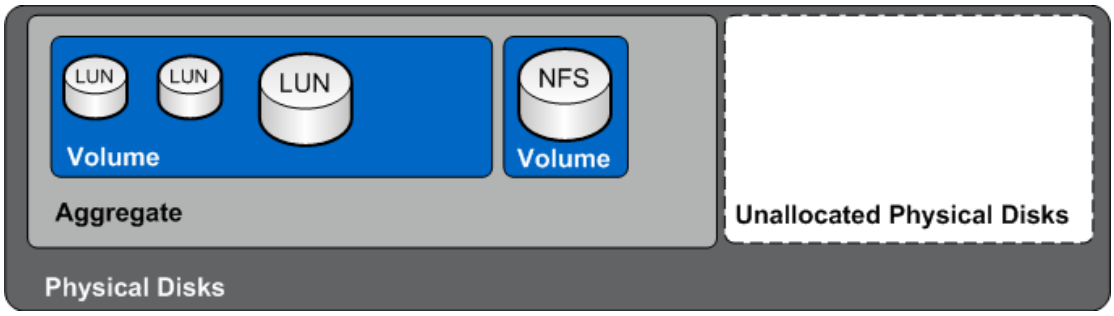

Figure 4) Aggregate layer.

The first disk aggregate, aggr0, should be running on the default three-disk group along with the default volume, vol0. Do not put any other user data in aggr0 or vol0. Create a separate aggregate for user data.

When creating aggregates, it is best to use the defaults for most items such as RAID groups and RAID level. The default RAID group size is 16 disks, used in a RAID-DP® configuration. RAID-DP is NetApp's high-performance implementation of RAID 6. Also, when creating aggregates and volumes, allow Data ONTAP to automatically choose disks and always maintain a hot spare in the storage array.

The data volumes should be named something meaningful, such as `kvm_vol` or `vol_kvm_store`. If multiple environments are to be stored on the same NetApp FAS controller, then extend the volume name to be even more descriptive, such as `vol_hr_nfs` or `vol_mrktg_fcp`.

In the example deployment described in this technical report, there are only two physical servers, which service a number of virtual guests. A pair of NetApp FAS controllers backs the environment. The group of two servers is referred to as a *cluster.* If there are multiple clusters, each one should have its own flexible volume. This allows a more secure approach to each cluster's shared storage.

NetApp Snapshot™ copies allow a point-in-time, read-only copy of a flexible volume that incurs no performance hit to the server or storage virtual environment. Further, it takes very little space, so it has little impact on storage consumption. A Snapshot copy usually takes less than a second to make, and up to 255 Snapshot copies can be stored per volume. In the context of the KVM virtual environment, the copy can be used to recover virtual servers and data affected by human error or software corruption. The use of separate flexible volumes for each KVM environment also makes more efficient use of the Snapshot copy.

4.3 SIZING

Sizing depends on the number of VMs to be deployed and on projected growth. To maximize storage efficiency, NetApp deduplication and thin provisioning should be employed on the volume and LUN, respectively. NetApp deduplication increases storage efficiency by folding identical blocks from different virtual machine images into a single instance on the storage controller. Thin provisioning allocates space for

a LUN without actually reserving the space all at once. When using NetApp deduplication with LUN-based storage, the LUN should be twice the size of the volume in order to see the full benefit of deduplication. Volumes allocated for NFS-based storage do not need the same consideration.

A single NetApp FAS controller can also be used to store multiple clusters. For instance, a single group of host nodes might support a number of databases while another group of host nodes supports application servers, all using the same NetApp FAS controller. By using VLAN segregation, they operate side by side without interacting with each other.

The secure separation can be further extended with the use of NetApp MultiStore® on the NetApp FAS controller. This allows the partitioning of a single storage device into multiple logical storage devices. For more information on MultiStore, refer to http://www.netapp.com/us/products/platform-os/multistore.html.

4.4 NETWORKING REQUIREMENTS FOR THE NETAPP FAS CONTROLLER

Virtual interfaces (VIFs) should also be configured on the storage controller. A VIF is a logical interface that adds an abstraction layer to two or more physical interfaces. Depending on the type of VIF, the virtual link afforded by the abstraction layer might provide failover capabilities or it might balance traffic across the physical interfaces.

After VIFs are configured, one or more VLANs should also be created on the VIFs to provide separate traffic for various storage protocols and environments. A VLAN is a logical separation of a switch, allowing it to carry multiple subnets simultaneously. In the context of this technical report, a single switch can carry the primary paths for the public and data networks. For example, instead of having a separate switch for public traffic—iSCSI traffic for one cluster and NFS traffic for another cluster—a single switch can be configured to carry the traffic for each.

For the KVM environment described in this technical report, separate VLANs are created for public traffic, iSCSI traffic, and NFS traffic. Each host has four NICs as well as a hardware-based iSCSI initiator with dual ports. Each traffic type uses a channel bond (Red Hat nomenclature for VIF). In addition, the paths (primary and secondary) go to separate switches. With VLANs, only two switches are needed. Without VLANS, this could require six switches.

This adds the requirement that each point along the network must be compatible with VLANs and also configured for use with VLANs. Each switch along the wire also needs to support and be configured for VLANs.

4.5 DATA RESILIENCY AND EFFICIENCY FOR THE STORED DATA

For the highest availability, NetApp FAS controllers should be set up in a multipath HA active-active configuration. This provides seamless failover for the back-end storage in the event of a hardware failure or planned maintenance window. The NetApp storage can also be replicated with SnapMirror® or SnapVault®, which should be used in disaster recovery (DR) planning. These products enable the use of remote locations for increased data protection. For more information on these products, see the "Storage Best Practices and Resiliency Guide" and "SnapMirror Async Overview and Best Practices Guide," referenced in Appendix C.

4.6 CHOOSING A STORAGE PROTOCOL

When selecting a protocol, consider speed, cost, and scalability. FCP, iSCSI, and NFS are all stable protocols, and this technical report does not recommend one over the others. When using a LUN-based protocol for a single group of KVM host nodes, a clustered file system such as GFS or GFS2 is required. Mounting the same ext3 file system simultaneously on several hosts leads to data corruption and instability because this file system is not designed for multiple hosts with read/write access.

NFS is the easiest and most affordable storage to deploy and attach the host nodes to because it usually involves the existing network infrastructure. It also scales very well in a virtual environment, which makes it easy to deploy KVM in test environments.

Like NFS, iSCSI requires only the existing Ethernet infrastructure. Although most deployments of iSCSI use the software-based initiator included in RHEL, it is also possible to use a hardware-based initiator such as

an iSCSI HBA. For a highly available host node environment, iSCSI can also be used in conjunction with GFS or GFS2.

For an environment that already includes Fibre Channel switches and the required cables and adapters, FCP may be an attractive choice. But for a data center that has not yet invested in FCP, the initial cost may outweigh the advantages. Like iSCSI, FCP can be used in conjunction with GFS or GFS2.

5 KVM DATA TYPES AND DISK ALIGNMENT

5.1 DATA AND DISK TYPES IN THE KVM ENVIRONMENT

The NetApp FAS controllers are extremely flexible in their operation. They can serve NFS, FCP, iSCSI, and CIFS traffic simultaneously.

Note: CIFS is not supported for the shared storage component; however, Windows guests should have no issue accessing CIFS shares.

Storage of a virtual environment includes four main types of files:

- Disk images
- Configuration files
- ISO images
- Golden images

To a virtual machine, a disk image is an abstraction that looks like and mimics a physical disk. The default type of disk image in a KVM virtual environment is a raw disk image. When a raw disk image is created, it is represented by a sparse file. That is, when an 8GB raw disk file is created, it looks like an 8GB file but it is not actually taking up any space. This is because only written sectors reserve space.

RHEL does not yet have an officially supported disk image format that supports thin provisioning. (This is only in the context of RHEL. NetApp supports thin provisioning of LUNs on the FAS controller.)

ISO images are byte-for-byte copies of CD or DVD media and can be used to install new virtual servers. Each different operating system version, architecture, and type requires an ISO image.

Configuration files store the metadata that KVM uses to run the virtual environment. The files are in XML format, and a configuration file exists for each virtual guest, network, and storage pool (if storage pools are configured). The default location for virtual guest configuration files is in `/etc/libvirt/qemu`, which is completely separate from the shared storage. For this reason, NetApp recommends making provisions for backing up this file. Another possibility is to create a symlink that has the data stored in the shared storage.

A golden image is simply a template. The concept is to build and configure a particular type of server (database, Web, application) once and then clone it when a new instance of that server is needed. The creation of a golden image is typically a manual process because it involves getting the server into a state where everything is ready to go. The process is discussed in section 13, “Creation and Configuration of Golden Images.”

5.2 DISK ALIGNMENT OVERVIEW

In any virtual environment, there are a number of layers of abstraction between physical disks and the VM's virtual disk. Each layer in turn is organized into blocks to make the most efficient use of storage. The focus is not the size of the block, but rather the starting offset. To avoid latency caused by extra reads and writes, the starting offset of a file system should line up with the start of the block at the next layer down.

This is in no way unique to NetApp; it applies to any storage vendor. It is a simple by-product of legacy partitioning schemes. For the full explanation of disk alignment in virtual environments, see TR-3747, "Best Practices for File System Alignment in Virtual Environments," which is referenced in Appendix C.

Traditionally, disk utilities such as GNU `fdisk` are used to alter the real geometry of a disk drive to partition the disk in a more efficient manner. A disk may only have 4 disk heads but may report 16 to the BIOS.

Unfortunately, this does not usually bode well because the default values used by tools like `fdisk` do not align the offsets properly for use in a virtual environment.

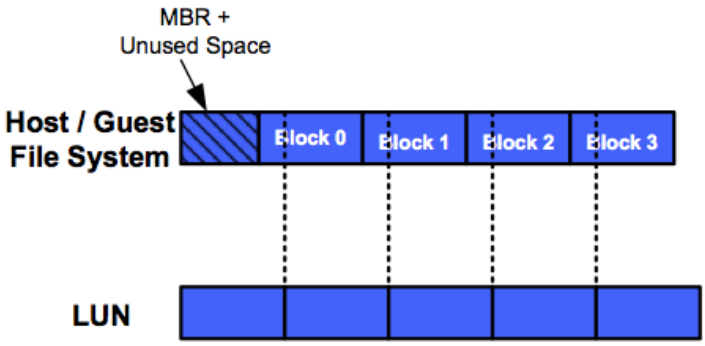

Figure 5) Misaligned blocks.

To quote TR-3747, “NetApp uses 4KB blocks (4 x 1,024 = 4,096 bytes) as its basic storage building block. Write operations can consume no less than a single 4KB block and can consume many 4KB blocks depending on the size of the write operation. Ideally, the guest/child OS should align its file system(s) such that writes are aligned to the storage device's logical blocks. The problem of unaligned LUN I/O occurs when the partitioning scheme used by the host OS doesn't match the block boundaries inside the LUN.”

Without this proper alignment, significant latency occurs because the storage controller has to perform additional reads and writes for the misaligned blocks. For example, most modern operating systems such as RHEL and Windows 2000 and 2003 use a starting offset of sector 63. Pushing the offset to sector 64 or sector 128 causes the blocks to align properly with the layers below.

Figure 6 shows proper alignment of the guest file system, through the host node file system, and down to the LUN on the NetApp FAS controller.

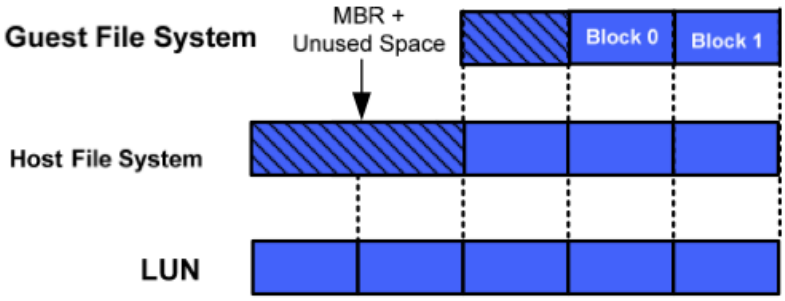

Figure 6) Properly aligned blocks.

Aligning the underlying storage is discussed in each of the storage cases later in this technical report. Aligning the disk images is discussed in more detail in the companion deployment guide, “Deployment Guide for KVM and Red Hat Enterprise Linux on NetApp Storage,” as well as in NetApp TR-3747. From a high level, it involves pushing the offset of the first disk partition to a number divisible by 8 sectors, with each subsequent partition aligning with a starting sector that is also divisible by 8.

5.3 BEST PRACTICES FOR NFS-BASED STORAGE

Best practices for NFS-based storage for KVM are centered on network and nomenclature. The network requirements are based on segregating NFS traffic from public traffic, as well as making use of jumbo frames. The host nodes have additional configuration requirements that are addressed later in this technical report. The nomenclature best practices are simply that naming should be meaningful and consistent.

Most important, the NFS traffic must be on a separate 1GB network minimum and must employ a separate VLAN. A 10GB network is preferable, in which case all traffic can exist on the same wire that carries the different VLANs. With the NFS traffic kept separate, the benefits of jumbo frames can be taken advantage of without affecting the other networks. A jumbo frame is a datagram that is larger than the default of 1500 bytes that standard Ethernet uses. This is in comparison to a typical NFS datagram, which is 8400 bytes in size.

When using the default datagram size of 1500 bytes, NFS datagrams get fragmented on the network and additional cycles are required to piece them back together, which can lead to serious performance degradation. By increasing the maximum transmission unit (MTU) to 9000 bytes, each NFS datagram can be sent across the wire in one piece.

On the NetApp controller, jumbo frames are best configured on a per VLAN basis. Following the networking best practice of setting up VIFs first, simply create a new VLAN to be associated with a VIF and adjust the MTU to 9000. From there, each switch port on the way to the host node should also be configured for jumbo frames in addition to the proper VLAN.

The naming portion of the NFS best practices is simple. The process of creating an NFS export on a NetApp FAS controller is straightforward:

1. Create a flexible volume.
2. Create a qtree on that volume (optional).
3. Export the volume or qtree.

When creating the volume for use as an NFS export, the space guarantee can be None and the Snapshot reserve should be left at 20% to account for the Snapshot copies. NetApp highly recommends using Snapshot copy. If Snapshot copy is not used, then enter 0% for the reserve.

A qtree is a special subdirectory of a volume that can be exported as an NFS share. The key benefits of qtrees are that they allow UNIX® and Linux style quotas and NetApp SnapVault and SnapMirror products can easily use them. In addition, a qtree can be assigned a security style that affects only its directories and files, not the entire volume. The use of qtrees is optional.

Give each volume or qtree a meaningful name, such as `/vol/kvm_vol/nfs_stor` or `/vol/vol_mktg/qt_nfs_storage`. The naming convention should be descriptive and consistent across volumes and exports. Storage and system administrators should be able to look at an export name and immediately recognize its type and use.

Finally, when creating the export, part of the process involves specifying which hosts and/or subnets have access to the NFS export. Be sure to allow access only to the specific hosts that need access. Don't give All Hosts access, and don't grant access to an entire subnet (unless all hosts on a subnet are going to mount the export).

DISK ALIGNMENT FOR NFS STORAGE

Disk alignment for NFS-based storage is needed only on the virtual disk images. NFS exports are aligned by default and require no additional consideration or configuration.

5.4 CONFIGURING NETAPP FOR ISCSI-BASED STORAGE

Like NFS-based storage, iSCSI-based storage best practices are best viewed from a network and naming aspect. A separate 1GB VLAN should be created to segregate the iSCSI traffic from the public traffic. A 10GB network is preferred if the proper NICs and switches are available. The next network step is to confine iSCSI traffic to the VIF so that the other interfaces on the NetApp FAS controller know not to accept iSCSI requests.

Like the naming for NFS-based storage, the naming for iSCSI-based storage should be meaningful. Create volumes and LUNs that are instantly recognizable. For example, using `/vol/vol_web/lun_java01` is an easy way to identify a Java™ application server.

An additional step in setting up a LUN, regardless of iSCSI or FCP, is to create an initiator group, or igroup. This mechanism allows a host or group of hosts to view a LUN by way of initiator address. After the igroup is

created, it is mapped to the LUN. Name igroups in the same meaningful way that LUNs and exports are named, using a name such as `ig_iscsi_kvm`.

DISK ALIGNMENT FOR ISCSI AND FCP-BASED STORAGE

The initial disk alignment is configured automatically when creating a LUN. LUN setup requires the selection of the specific operating system type.

Disk alignment for LUN-based storage can be accomplished in one of two ways. The first way is to use a partitioning tool such as `parted` to create a single large partition with the proper offset. Then use logical volume manager (LVM) to manage that partition.

The second way is to skip the step of creating any partitioning on the device and use LVM on the entire device. For example, instead of using the `parted` or `fdisk` partitioning utility to create `/dev/sdb1`, simply run the LVM utility against the entire device, `/dev/sdb`.

When creating the file system, be sure to create it with a 4096 block size. Then follow the steps in the deployment guide and NetApp TR-3747 for aligning the disk images.

5.5 CONFIGURING NETAPP FOR FCP-BASED STORAGE

An unused onboard fibre port can be converted for use as a target; that is, the onboard HBAs can function as either a target or an initiator, based on configuration. If there are no free onboard HBAs, HBA cards can be installed in open PCI slots. See now.netapp.com/NOW/knowledge/docs/san/fcp_iscsi_config/ for supported hardware. By default, the onboard HBAs are configured as initiators for use with the fibre-attached disk shelves that serve as the targets.

Once the HBA is installed and recognized, create a flexible volume and LUN. Also name the volume and LUN in a meaningful way.

Like the iSCSI LUN, FCP LUNs require the creation and use of an initiator group, or igroup. This is the mechanism that allows a host or group of hosts to view a LUN by way of WWPN. After the igroup is created, it is mapped to the LUN and is ready to be used. Igroups should also be named in the same meaningful way as LUNs and exports, using a name such as `ig_fcp_kvm`.

6 INSTALLATION AND BASE CONFIGURATION OF HOST NODES

The configuration of the host nodes is straightforward. The only tuning required is choosing the shared storage medium to host the virtual environment.

6.1 RHEL VERSION

To use KVM, RHEL version 5.4 or later is required. RHEL 5.3 and earlier versions do not include any of the required packages or libraries.

6.2 DISK LAYOUT

The disk layout should match Red Hat best practices as well as the needs of the data center hosting the virtual environment. Red Hat has recommendations for swap, depending on the size of the physical RAM. These recommendations are described in the deployment guide as well as on the Red Hat site. It is also a best practice to have `/boot` and `/var` on separate partitions. Separate partitions for `/home` and other major directories are not required for the host nodes.

Use of LVM is supported but might not be required on the host nodes. There is no performance hit with its continued use; however, if the host nodes are not running any applications aside from KVM, all of the growing file systems reside on the NetApp FAS controller. This means that NetApp flexible volumes and qtrees can take on the responsibilities normally relegated to LVM, such as Snapshot copies and online volume growth.

The one exception is in the use of GFS or GFS2, where using clustered LVM is a requirement.

6.3 PACKAGE SELECTION

The package selection should be particular and minimal for the host nodes. Do not install anything related to a purely desktop machine; games and office productivity packages have no place on a host node. The packages and package groups should be limited to items such as the base installation, text editors, and the KVM-related packages. As a case in point, the servers used to test and develop this document had the following packages listed in their `kickstart` files:

- **@admin-tools**. Group of Linux administrative tools
- **@base.** Group of base Linux packages
- **@core.** Group of packages required for the smallest installation
- **@editor.**: Group of text editors
- **@text-internet.** Group of nongraphical Internet tools
- **@gnome-desktop**. Group of packages for GUI (to demo GUI-based tools)
- **device-mapper-multipath.** Package for multipathing of LUN-based storage
- **Various packages associated with KVM and QEMU**
- **Various packages associated with Red Hat Cluster Suite and GFS2**

There are many other packages (805 in total) that are installed for dependencies, but these are very specific choices. Anything under 900 packages is considered streamlined.

The decision of whether to install the graphical packages is best left up to the business needs and requirements of the group responsible for maintaining the virtual environment. From a resource and security standpoint, it is better not to install any of the graphical packages; however, the graphical tool Virtual Machine Manager may appeal to many system administrators and engineers. One solution is to install and run only the graphical packages on one of the host nodes. This decision is best left to the maintainers of the environment.

6.4 SECURITY

Properly securing the host nodes is of paramount importance. This includes proper use of iptables for packet filtering and Security-Enhanced Linux (SELinux) for file-level security.

The firewall provided by iptables should allow only the ports needed to operate the virtual environment as well as communicate with the NetApp FAS controller. Under no circumstances should iptables be disabled. For a list of ports, see Appendix A: Ports to Allow in IPtables Firewall.”

SELinux was developed largely by the NSA (and later incorporated into the 2.6 Linux kernel in 2003) to comply with U.S. government computer security policy enforcement. SELinux is built into the kernel and provides a mandatory access control (MAC) mechanism, which allows the administrator to define the permissions for how all processes interact with items like files, devices, and processes.

For example, the default directory for disk images in a KVM environment is `/var/lib/libvirt/images`. SELinux has a default rule that gives that directory a security context consistent with virtualization. If someone or something creates a disk image in `/etc/root/kit`, for example, SELinux does not allow it to run without the proper security context. This provides a very granular level of security. For more information on SELinux, see the "Red Hat Enterprise Linux 5 Deployment Guide," referenced in Appendix C.

Unless Red Hat Cluster Suite is used, SELinux should remain in its default state of “enabled” and “targeted.” As of this writing, SELinux and Red Hat Cluster Suite are not supported by Red Hat when used together. Separately, however, they are fully supported by Red Hat.

The primary means of connecting to a virtual server and the virsh (virtual shell) console is by way of SSH and an SSH tunnel, respectively. It is also possible to configure communication to the virsh console with the use of TLS, but that is outside the scope of this technical report. For more information, see the "Red Hat Enterprise Linux 5 Virtualization Guide," referenced in Appendix C.

The default means of communicating with virsh is by way of an SSH tunnel. Essentially, a URI is called (`qemu+SSH://<host_node>/system`) and the tunnel is opened. Although it is possible to enter a password for each connection, the best practice is to use SSH keys. A key pair is created on the remote host, and the public key is distributed to each of the host nodes. This enables encrypted communication to

the host nodes without the use of passwords. This is very important when considering the automation of live migrating virtual servers.

6.5 UNNECESSARY AND INSECURE SERVICES

Many services are enabled by default on RHE that need to be shut off. For example, there is little need for a printing server or a mail server on a host node. Review all of the services that are configured to start on boot and disable any service that is not required to either support or protect the virtual environment.

Disable the use of RSH, telnet, FTP, and any other insecure service in favor of SSH, SCP, and SFTP.

6.6 REQUIRED SERVICES

Some required services are not enabled by default. Most notably, NTP must be enabled and configured on every host node and remote node in order to keep the time synchronized across the KVM environment. Also be sure that `libvirtd` is configured to start on boot. This is the service that allows remote access to the KVM hypervisor.

7 INSTALLATION AND CONFIGURATION OF KVM HOST NODES

The installation and configuration of the KVM is also straightforward. It consists of installing the KVM and QEMU packages, starting the `libvirtd` service, and choosing a shared storage medium.

7.1 SINGLE HOST NODE VERSUS MULTIPLE HOST NODES

The decision to use a single host node over multiple host nodes is easy to make. A single host node is useful in a lab environment or when testing certain features. There is no scalability, load balance, or redundancy with the use of a single node; however, a single node can make use of a LUN without requiring a cluster file system such as GFS or GFS2.

In contrast, an environment based on multiple host nodes is capable of scaling better, performing live migrations, and allowing server maintenance without virtual server downtime.

The best practice is to use two or more servers on which to base the virtual environment. As more virtual servers are created, they can be distributed across the host nodes to balance the use.

Having multiple host nodes also allows the virtual servers to be moved from one node to another without any interruption to the application they support. In the context of KVM, this is referred to as *live migration.* Live migration can also be used to move all virtual guests from a particular host node in order to perform hardware maintenance.

7.2 PACKAGE INSTALLATION

The best practice for installing packages involves having the host nodes properly registered with Red Hat. The servers are normally registered to the Red Hat network or to a Red Hat network satellite server. Both of these require proper support subscriptions. For secure networks that do not have access to either, it is necessary to set up a software repository that is local to the KVM infrastructure.

Having a properly configured repository (by way of Red Hat network, satellite, or local repository) is necessary to facilitate the proper fulfillment of package dependencies in Red Hat Enterprise Linux. For example, if the command to install a particular package is invoked, the package manager is intelligent enough to call for the dependent packages as well. If there is no repository, package installations fail on the dependencies, and installing packages piecemeal to satisfy dependencies is awkward.

After the KVM-related packages are installed, start the `libvirtd` service and configure it to start on boot.

Then attach the shared storage. KVM supports two methods of attaching shared storage—traditional and storage pools. In the traditional model, NFS exports and LUNs that are to be mounted automatically at boot are entered in the `/etc/fstab` file of the host nodes. The operating system manages the automatic mounting and unmounting of the storage. The environment built for this technical report uses the traditional method.

In the storage pool method, `libvirtd` handles the mounting and unmounting of the storage. Storage pools can be created and managed from the Virtual Machine Manager graphical tool as well as from the virsh command line tool. In both cases, an XML file is created and must be copied to each of the host nodes. For more information on storage pools, refer to the "Red Hat Enterprise Linux 5 Virtualization Guide," referenced in Appendix C.

7.3 KVM HOST NODE NETWORK CONFIGURATION

By default, KVM creates a single virtual bridge on the host nodes that allows the virtual guests to communicate with each other and the outside world. The virtual bridge is part of the `bridge-utils` package and works in conjunction with `libvirtd` and `dnsmasq` to provide IP addresses to the KVM guests. It is functional but very basic because it does not allow hosts outside of the host node to reach back to the KVM guests.

The best practice is to extend the virtual bridge configuration to create a virtual public bridge. This involves taking over a physical interface and editing the standard NIC configuration file to reflect a bridge instead of an Ethernet device.

In addition, this requires iptables to be configured in one of two ways. The first way is to create a rule in iptables that allows the bridged traffic. This is the quickest configuration.

The second way is to have bridging removed from the control of iptables by way of a few kernel tunable parameters. This has no effect on the ability of iptables to protect the host nodes or virtual guests. For more information, see Appendix B: Kernel Tunable Parameters for Proper Bridging."

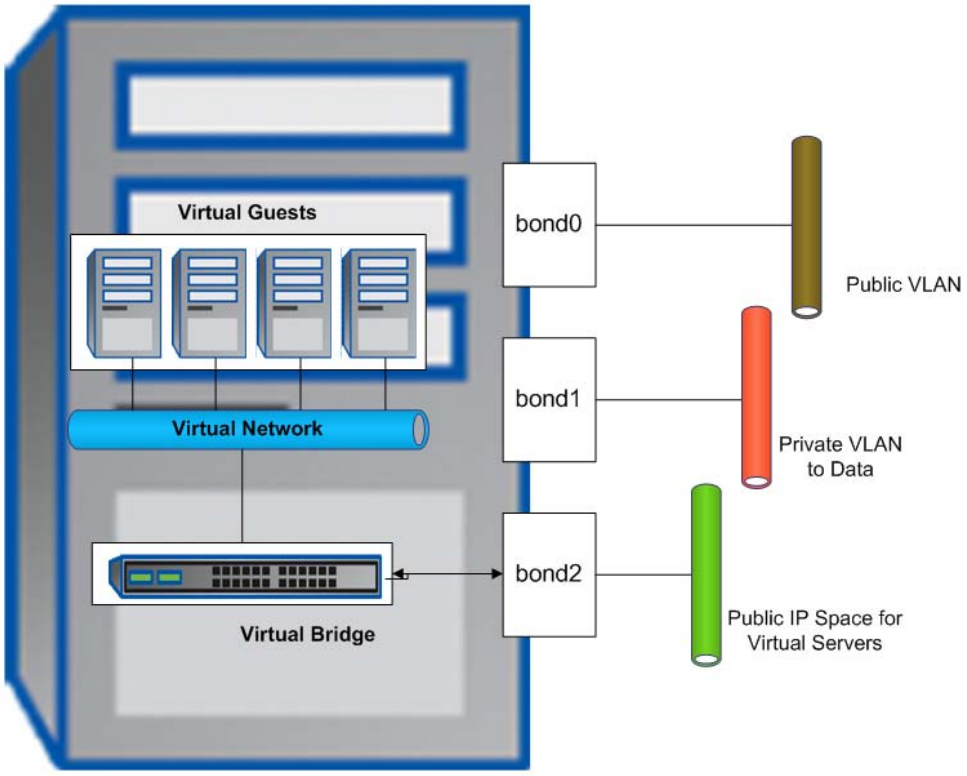

Figure 7) Virtual guests using a virtual bridge to access the network.

Figure 7 illustrates the virtual guests accessing the network by way of a public (virtual) bridge. Because the virtual guests have their own twp-way access to the network, the only access that the host node needs is for management purposes.

8 HOST NODE CONFIGURATION OF NFS SHARED STORAGE

The configuration of the NFS-based shared storage is typically categorized in terms of the network configuration, defining specific ports for NFS, and the mount options. Also, there are some things to consider regarding SELinux.

8.1 NFS NETWORK

Following the best practices for setting up the NFS export on the NetApp FAS controller results in the host having access to a private NFS network. NFS traffic should reside on its own network and VLAN of at least 1GB in speed. Also, NetApp strongly encourages the use of jumbo frames to prevent fragmentation of datagrams.

The separate NFS network used in the environment built for this document uses a nonroutable IP space. This simplifies the network configuration because the switch between the host node and the storage needs to be configured only for the NFS VLAN.

NetApp strongly encourages the use of channel bonding for the private NFS network. Channel bonding is the term that Red Hat uses to describe two or more NICs that are bound together for failover purposes. This also requires that a separate switch be used (the primary link to one switch and the secondary link to the second switch). Channel bonding can be configured for active-passive, round-robin, or even an LACP aggregate.

8.2 FORCING THE NFS CLIENT TO USE PREDICTABLE PORTS

By default, the `portmap` service dynamically assigns ports for RPC services that can be troublesome when interacting with a firewall. To gain consistent control over which NFS ports are used, it is necessary to force the NFS client to use the same ports from connection to connection and host to host.

To configure NFS ports in a predictable manner, the `/etc/sysconfig/nfs` file needs to be edited so that the TCP ports for `lockd` and `statd` are defined.

8.3 MOUNT OPTIONS

Although most of the mount options for an NFS-based storage pool are typical, the following explanations may prove helpful.

- **rw.** The shared storage requires read and write access to create and alter disk images.
- **bg.** This option puts the mount option in the background if the NetApp FAS controller cannot be reached initially.
- **vers=tcp.** This option forces the use of NFS over TCP as opposed to UDP. Do not use UDP for the shared storage.
- **timeo=600.** This option causes a time-out after 60 seconds if the client does not receive a reply for an RPC. The value is in tenths of seconds.
- **rsize=65536,wsize=65536.** These options state that the NFS client caches 64KB of reads or writes before it actually performs the read or write action.
- **hard.** This option is used to ensure data integrity as compared to a soft mount.
- **intr.** This option allows the NFS client to interrupt the NFS mount if the server or network is not responding.
- **_netdev.** This option states that the device is on the network and no attempt should be made to mount it until after network services have started. Note the leading underscore.

Most of these options are defaults that get configured even if no options are specified on the command line. However, it is important to understand the basics of each option in order to understand the best practice. The NFS mount options should be listed as `defaults,bg,_netdev`.

8.4 SELINUX CONSIDERATIONS FOR NFS-BASED SHARED STORAGE

The default location of the disk images is in /var/lib/libvirt/images. It is assumed that subdirectories will be created under the images directory to keep things organized. Common subdirectories include one for each operating system, as well as a place for storing golden images and a place for storing ISO images. Some virtualization administrators prefer to move the images directory directly under the root directory (/).

Many decisions depend on the business needs of the virtual environment as well as the preferences of the administrators and engineers responsible for the environment. Regardless of whether the images directory is moved or subdirectories are created, SELinux must be configured to allow access for KVM activity.

It is a minor matter to update the security context of the KVM image subdirectories so that any new files created under them also inherit the proper context.

8.5 DISK ALIGNMENT

To align guest disks on NFS shared storage, see section 13, “Creation and Configuration of Golden Images.”

9 HOST NODE CONFIGURATION OF ISCSI SHARED STORAGE

The configuration and tuning of iSCSI shared storage on the host side depends on how the network is set up. Much like the NFS network, the iSCSI network needs to be at least a 1GB minimum on its own VLAN. Also, multiple NICs or iSCSI HBAs should be used for redundancy. In the case of a 10GB network, just having a separate VLAN on the wire along with at least one redundant 10GB NIC is the best practice.

In the case of a software-based iSCSI initiator, the traffic needs to go over at least two NICs in a channel bond for redundancy, each going to its own switch. For a hardware-based iSCSI initiator, the same applies. In both cases, a single LUN appears as two devices, so multipathing software (RHEL ships with DM-MPIO) is also required.

After the LUNs are created on the NetApp FAS controller, the host node needs to rescan the iSCSI bus to see the LUN or LUNs. The multipathing configuration comes next, followed by the LVM and file system creation. Configuring the mount point in fstab or a storage pool and mounting the LUN completes the storage process.

Using multiple host nodes to mount the same LUN-based file system in read/write mode requires the use of a clustered file system. Red Hat Enterprise Linux 5.4 AP ships with GFS and GFS2 as well as Red Hat Cluster Suite. Considerations for GFS and GFS2 are discussed in section 11.

Note: When using Red Hat Cluster Suite, SELinux must be disabled.

9.1 DISK ALIGNMENT

To align guest disks on iSCSI shared storage, see sections 11 and 13.

10 HOST NODE CONFIGURATION OF FCP SHARED STORAGE

The best way to start is to install the fibre HBAs before installing the operating system. This makes sure that the proper drivers and automatic configuration occur. The best practice also specifies the use of at least two HBAs, or at least dual ports, for redundancy. When using multiple HBAs, a single LUN appears as two devices, so multipathing software is also required. (RHEL ships with DM-MPIO.)

After the LUN is created on the NetApp FAS controller, the host node needs to rescan the SCSI bus to see the LUN or LUNs. The multipathing configuration comes next, followed by the LVM and file system creation. Configuring the mount point in fstab or a storage pool and mounting the LUN completes the storage process.

Using multiple host nodes to mount the same LUN-based file system in read/write mode requires the use of a clustered file system. Red Hat Enterprise Linux 5.4 AP ships with GFS and GFS2 as well as Red Hat Cluster Suite. Considerations for GFS and GFS2 are discussed in section 11.

Note: When using Red Hat Cluster Suite, SELinux must be disabled.

10.1 DISK ALIGNMENT

For information about aligning guest disks on FCP shared storage, see section 11, “Host Node Configuration and Tuning of GFS2 Shared Storage” and section 13, "Creation and Configuration of Golden Images."

11 HOST NODE CONFIGURATION OF GFS2 SHARED STORAGE

For multiple host nodes to safely read and write to the same LUN-based file system at the same time, a clustered file system is required. The use of GFS2 (or GFS) as well as Red Hat Cluster Suite satisfies this requirement. This technical report focuses on GFS2.

11.1 GFS2 AND RED HAT CLUSTER SUITE

The Red Hat Cluster Suite best practice includes having a host separate from the cluster nodes (in this case, the host nodes) to manage the cluster. The use of a remote administration host is discussed in section 12.

The use of GFS2 satisfies the clustered file system, and the Red Hat Cluster Suite supports part of the data integrity requirements for GFS2. Figure 8 illustrates the Red Hat Cluster Suite.

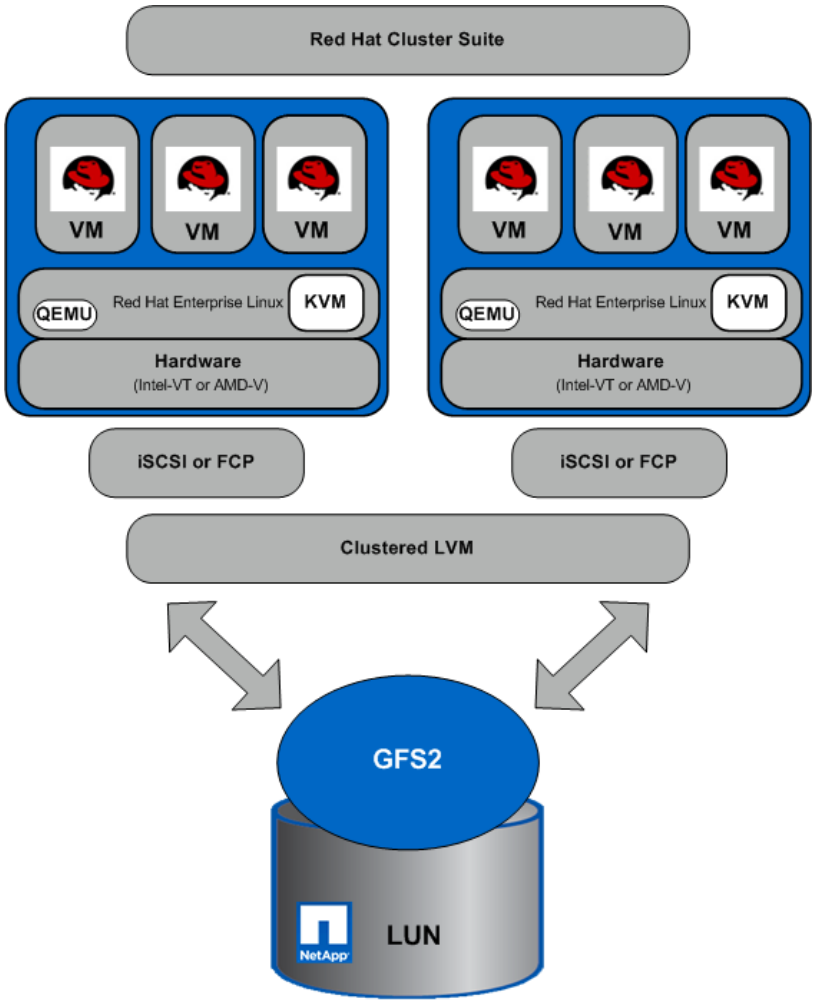

Figure 8) Red Hat Cluster Suite.

To configure the Red Hat Cluster Suite with GFS2, follow these steps.

1. Install the host nodes as described in section 7, with the addition of the GFS2-specific packages.
2. Create the LUN to be used for the shared storage (iSCSI or FCP).
3. Configure the multipathing.
4. Install the cluster management piece on the remote administration host (described in section 12).
5. Configure the cluster.
6. Using clustered LVM (part of the GFS2-related packages), create a volume using the entire device (`/dev/sdb, not /dev/sdb1`) for proper disk alignment.
7. Create the GFS2 file system with the `noatime` option.
8. Configure the mount in `fstab` or storage pool.

Note: Although the use of LVM is usually optional for the host nodes, it is required for GFS and GFS2.

11.2 DISK ALIGNMENT

GFS2 is aligned properly by default because the default block size is 4096 bytes. Create an LVM volume that encompasses the entire device so that the clustered LVM is aligned. The only remaining step is to align the disk images to be used, as described in the deployment guide and NetApp TR-3747.

Note: When using Red Hat Cluster Suite, SELinux must be disabled.

12 USE OF A REMOTE ADMINISTRATION HOST

NetApp recommends using a remote host to administer the remote environment. The remote host is used to run the entire virtual environment from a central server or workstation instead of on one or more of the host nodes. The only requirement is to install the basic KVM packages needed to run the various administrative commands on the remote host.

A remote administration host has the following uses in a KVM environment:

- Secure host to manage the KVM environment
- Secure host to manage the NetApp FAS controller
- Secure host to manage Red Hat Cluster Suite (if using GFS or GFS2)

Using a remote host to administer the remote environment avoids the following scenario.

Host node 1 is the designated server from which to run all commands. It contains all of the SSH keys for the other nodes, as well as all of the scripts written to ease the administration of the KVM environment. The server goes down because of maintenance, hardware failure, or worse, an intrusion. Although it is possible to perform the duties on the other host nodes until host node 1 is brought back up, it is much more complicated.

The preferred solution is to designate as a remote host a server or workstation that is otherwise not related to the host nodes. A backup remote host is also advisable to avoid the scenario just described. The remote host needs to have only a small subset of the KVM packages in order to be effective. Virtual servers can be migrated, created, destroyed, and maintained from the remote host. If the NetApp FAS controller is dedicated to the KVM environment, then the remote host for KVM could also be the admin host for the NetApp controller. Finally, when using GFS2 and Red Hat Cluster Suite, the remote host serves as the remote host for that as well. For more information about a remote host in the context of GFS2/Red Hat Cluster Suite, see section 11.1.

It is important for the remote host to be secured properly. This means using iptables, SELinux, and SSH keys, not running unnecessary services, and removing unnecessary packages. In addition, the remote host requires synchronization with an NTP server to keep the time consistent with the rest of the virtual environment.

Note: The Red Hat Cluster Suite requires the use of a fencing device to be supported by Red Hat. When a cluster node fails to communicate properly (because of network failure, host failure, and so on), another node in the cluster “fences” it from running the application or maintaining access to a shared file system. When a failed node is fenced, it is forced to stop I/O, rejoin the cluster, and thereby preclude any data

corruption. The Red Hat-recommended fencing device is a networked power switch that, when triggered, forces the host node to reboot. A complete list of supported fencing devices is available on the Red Hat site.

13 CREATION AND CONFIGURATION OF GOLDEN IMAGES

Although many data centers and IT departments have their own methods for creating a golden image or template, the methods described in this section are a reliable starting point. Because there are many ways to perform this task, it is difficult to prescribe a single best practice. In general, the best practice is to use golden images rather than to create new images each time.

A golden image or template is simply a means of creating a particular type of server once. When that type of server is needed, or when additional instances of that type are needed, simply clone the golden image. The cloning process takes a fraction of the time that manual creation or traditional network installation takes.

If they are not already completed, create the subdirectories that will be used under the `images` directory. These subdirectories include names like gold, RHEL, Win, ISO, and so on. The purpose is to organize golden images, VMs based on different operating systems, and ISO images. When using NFS or nonshared storage, SELinux must be configured to provide the new directories with the proper security context.

Next, create a raw disk image in the gold directory and then align it. These processes are described in the "Deployment Guide for KVM and Red Hat Enterprise Linux on NetApp Storage." Then install the operating system on the newly aligned partitions.

After the host reboots, configure the operating system as needed. This includes third-party applications, configuration files, and firewall, as well as stripping out specific host name and IP information. This allows new information to be specified at cloning time.

Items like UUID and MAC address can be forced at clone time, allowing many possibilities. For instance, if a pool of MAC addresses is designated, they can be matched with a pool of IP addresses and hostnames. When a new server needs to be cloned, it can be called from a script that also pulls a MAC address from the pool that has an IP address already associated with it.

14 USE OF NETAPP STORAGE PROTECTION AND EFFICIENCY

14.1 COPYING VOLUMES WITH NETAPP SNAPSHOT COPY

In the context of a NetApp volume, a Snapshot copy is a point-in-time copy of what that volume looks like. Creating a Snapshot copy is near instantaneous, with most copies complete in less than a second. After a Snapshot copy is created, it can be used to recover data lost by human error or application error. By default, when a flexible volume is created, a reserve of 20% is maintained. This is the reason for the option of total space versus total usable space at volume creation time.

In general, the best practice for Snapshot reserve in a virtual environment is to set it to 0% and to disable the Snapshot copy schedule. To guarantee file consistency, it is best to quiesce (pause) the virtual guests, perform the Snapshot copy, and then resume the virtual guests. Following are some key points about Snapshot copies in the context of a KVM environment:

- Snapshot copies provide a natively integrated and easy-to-use data protection utility that helps storage administrators recover data.
- Snapshot copies protect against inadvertent file modification or deletion by making point-in-time backups available.

14.2 BACKING UP A NETAPP FAS CONTROLLER TO SNAPMIRROR AND SNAPVAULT

NetApp SnapMirror and SnapVault are data replication products that improve data protection by automatically maintaining duplicate copies of data either locally or remotely. SnapMirror is geared more toward a disaster recover solution, and SnapVault is more suited for local backup. After the initial baseline data transfer, SnapMirror and SnapVault replicate only changed blocks from the primary storage controller to minimize performance impact on storage and bandwidth impact on the network.

Because only changed blocks are replicated and bandwidth impact is limited to the rate of data change on the primary storage controller, both SnapMirror and SnapVault are excellent choices to replicate data over generally slow WANs to increase data protection options. Each replication option is highly configurable to meet business requirements.

SnapMirror can be configured to replicate data in asynchronous mode, semisynchronous mode, and full synchronous mode. SnapVault replicates NetApp Snapshot copies, and the frequency of the Snapshot replication process can be configured during initial SnapVault configuration or changed as needed afterward.

As a critical piece of disaster recovery planning and implementation, the best practice is to choose one of the products for data replication. In addition, replicating the data to a second tier of storage allows faster backup and recovery in comparison to traditional tape backups. Although having a local second tier of storage is good, NetApp highly recommends having a second tier of storage at a remote site.

After a product is chosen and implemented, it is important to stagger the transfers for non-peak-load times. For SnapMirror, NetApp also recommends throttling the bandwidth on transfers.

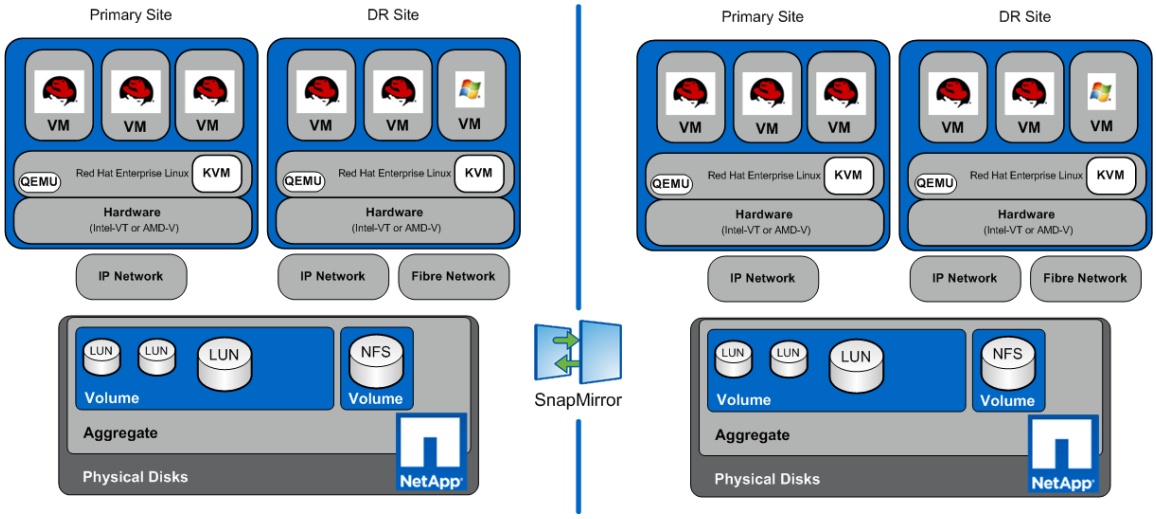

Figure 9) SnapMirror DR solution.

14.3 MAXIMIZING STORAGE EFFICIENCY WITH NETAPP DEDUPLICATION AND THIN PROVISIONING

For maximum storage efficiency, FAS deduplication should be used. Deduplication is a means of reclaiming redundant disk space by breaking data objects into small blocks. Each block has a digital signature that is compared to signatures in the same flexible volume. Space is reclaimed when duplicate block signatures are found and the duplicates are discarded. Deduplication can be used regardless of the storage protocol used in the KVM virtual environment, and the results are significant.

When using deduplication with a LUN, several factors must be considered to get the maximum benefit from the process. When creating a LUN of 100GB out of a 500GB volume, the default behavior of Data ONTAP is to reserve 100GB of space. The result is a volume having only 400GB left (if Snap Reserve is 0%). Although deduplication may be enabled on the volume and the duplicate blocks are discarded, 100GB of space is still taken up.

To maximize the benefits of deduplication on a LUN, thin provisioning must be enabled at creation time. When creating a thin provisioned LUN of 100GB, only a small increment is reserved and the LUN grows as needed up to the 100GB limit. The KVM host nodes still see a 100GB LUN, but the NetApp storage administrator might see only 20GB used.

The best practice for thin provisioning a LUN also includes some additional configuration. Essentially, it means defining a policy on the LUN that allows automatic growth, specifying when Snapshot copies can be deleted, and using fractional reserve. The Volume Auto Size setting defines the increments of growth when

the LUN needs to expand. Snapshot Auto Delete deletes the oldest Snapshot copies when the volume reaches a soft limit and is nearing capacity. The recommended soft limit is 5% remaining space. Finally, Fractional Reserve is a policy to define any additional space to reserve for LUN writes if the volume becomes full. When Auto Size and Auto Delete are in use, Fractional Reserve should be set to 0%

When using deduplication with NFS, the storage efficiency is seen immediately. Beyond the initial enabling of deduplication, there are no other considerations and no other configurations to be made.

More information is available in the "NetApp Deduplication for FAS and V-Series Deployment and Implementation Guide" referenced in Appendix C.

14.4 TRADITIONAL BACKUP METHODS

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` commands are available to replicate data to tape drives or to other storage, respectively.

Both commands are easy to use and include incremental functionality that backs up only files that have changed, reducing impact on both storage and tape backup libraries. Tape backups generated by `dump` can be stored off site, while `ndmpcopy` can replicate to NetApp storage across LANs or WANs without the need for tape library overhead. Both backup utilities increase data protection.

NetApp recommends using data backup as a key foundation piece of enterprise data protection.

15 CONCLUSION

Although KVM does not have the robust management tools included in RHEV, it offers a highly configurable and high-performance virtual environment that is easy to learn. This makes it a primary candidate for IT infrastructures that already have their own tools, a foundation of Linux or Linux skills, and the need for a solid virtualization platform that plugs in to an existing environment.

In a matter of minutes, a simple KVM environment can be set up and tested. A more complex production KVM infrastructure can be planned and deployed in a few short weeks. The graphical tools enable newcomers to quickly grasp the concepts; and the command line tools are very easily integrated into automation, management, and monitoring applications and tools.

From a storage and data efficiency standpoint, NetApp FAS controllers offer a unified, flexible approach to storage. The ability to deliver NFS, iSCSI, and FCP to multiple KVM environments simultaneously means that the storage scales nondisruptively with the KVM environment. Multiple KVM environments with different storage needs can be supported from the same NetApp FAS controller.

Additional NetApp products like Snapshot, SnapMirror, and deduplication offer the protection and storage efficiency required in any infrastructure.

Using the best practices in this guide provides a solid virtual infrastructure based on KVM and NetApp that serves as a solid foundation for many applications.

16 APPENDIXES

APPENDIX A: PORTS TO ALLOW IN IPTABLES FIREWALL

Table 1) Service and KVM-related ports.

Service and KVM-Related Ports		
Port	Protocol	Description
22	TCP	SSH
53	TCP, UDP	DNS
111	TCP, UDP	Portmap
123	TCP	NTP
3260	TCP,UDP	iSCSI (optional)
5353	TCP, UDP	mDNS
54321	TCP	KVM Inter-host Communication
32803, 662	TCP	NFS (optional)
49152-49216	TCP	KVM Migration
5900-5910	TCP	Virtual Consoles (extend out for additional consoles)
67, 68	TCP, UDP	DHCP
N/A	N/A	ICMP

Table 2) Cluster-related ports.

Cluster-Related Ports		
Port	Protocol	Description
5404, 5405	UDP	cman
8084	TCP	luci
11111	TCP	ricci
16851	TCP	modclusterd
21064	TCP	dlm
50007	UDP	ccsd
50006, 50008, 50009	TCP	ccsd

Table 3) Remote host ports.

Remote Host Ports		
Port	Protocol	Description
22	TCP	SSH
53	TCP, UDP	DNS
123	TCP	NTP
N/A	N/A	ICMP
8084	TCP	luci (if using Cluster Suite and/or GFS)
11111	TCP	ricci (if using Cluster Suite and/or GFS)

APPENDIX B: KERNEL TUNABLE PARAMETERS FOR PROPER BRIDGING

The following parameters must be changed from their default value of `1` (enabled) to `0` (disabled).

```
net.bridge.bridge-nf-call-ip6tables = 0
net.bridge.bridge-nf-call-iptables = 0
net.bridge.bridge-nf-call-arptables = 0
```

This allows traffic to move freely from the virtual bridge on the host node to the virtual server. As stated in section 7.3, this has no effect on iptables. If iptables is configured properly on the host nodes and virtual servers, this has no effect on the ability of iptables to protect at the packet layer.

Note: If the bridge module is removed and then reinserted, it is necessary to reset these parameters.

APPENDIX C: REFERENCES

Home page for KVM
www.linux-kvm.org

Red Hat and Microsoft Virtualization Interoperability
http://www.redhat.com/promo/svvp/

KVM – Kernel-Based Virtual Machine
www.redhat.com/f/pdf/rhev/DOC-KVM.pdf

Red Hat Enterprise Linux 5 Virtualization Guide
http://www.redhat.com/docs/en-US/Red_Hat_Enterprise_Linux/5.4/html/Virtualization_Guide/index.html

Red Hat Enterprise Linux 5 Deployment Guide
http://www.redhat.com/docs/en-US/Red_Hat_Enterprise_Linux/5.4/html/Deployment_Guide/index.html

Red Hat Enterprise Linux 5 Installation Guide
http://www.redhat.com/docs/en-US/Red_Hat_Enterprise_Linux/5.4/html/Installation_Guide/index.html

Red Hat Enterprise Linux 5.5 Online Storage Guide
http://www.redhat.com/docs/en-US/Red_Hat_Enterprise_Linux/html/Online_Storage_Reconfiguration_Guide/index.html

Best Practices for File System Alignment in Virtual Environments
http://www.netapp.com/us/library/technical-reports/tr-3747.html

Using the Linux NFS Client with NetApp Storage
www.netapp.com/us/library/technical-reports/tr-3183.html

Storage Best Practices and Resiliency Guide
http://media.netapp.com/documents/tr-3437.pdf

KVM Known Issues
http://www.redhat.com/docs/en-US/Red_Hat_Enterprise_Linux/5.4/html/Technical_Notes/Known_Issues-kvm.html

NetApp Deduplication for FAS and V-Series Deployment and Implementation Guide
http://www.netapp.com/us/library/technical-reports/tr-3505.html

SnapMirror Async Overview and Best Practices Guide
http://www.netapp.com/us/library/technical-reports/tr-3446.html

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.

www.netapp.com

© 2010 NetApp. All rights reserved. Specifications are subject to change without notice. NetApp, the NetApp logo, Go further, faster, Data ONTAP, FlexClone, MultiStore, NOW, RAID-DP, Snapshot, SnapMirror, and SnapVault are trademarks or registered trademarks of NetApp, Inc. in the United States and/or other countries. Intel is a registered trademark of Intel Corporation. Java is a trademark of Oracle Corporation. Linux is a registered trademark of Linus Torvalds. Windows and Windows Server are registered trademarks of Microsoft Corporation. UNIX is a registered trademark of the Open Group. All other brands or products are trademarks or registered trademarks of their respective holders and should be treated as such. TR-3858