Technical Report

Red Hat Enterprise Linux 6, KVM, and NetApp Storage: Best Practices Guide for Clustered Data ONTAP

Jon Benedict, NetApp
November 2012 | TR-4104

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1 Introduction

1.1 Overview

This technical report provides the current best practices for deploying Red Hat Enterprise Linux® (RHEL) 6 and clustered NetApp® Data ONTAP® 8.1. It is recommended that the reader browse through the document to get an understanding of each chapter and then reference this document as needed for specific requirements.

1.2 Intended Audience

This document addresses the needs of system architects, system administrators, and storage administrators who are investigating the use of KVM with RHEL 6 on clustered Data ONTAP. For Data ONTAP operating in 7-Mode, refer to TR-3848: RHEL 6, KVM, and NetApp Storage: Best Practices Guide.

1.3 Best Practices

Best practices provide the optimal balance of desirable features that have been shown to produce superior results; that are selected by a systematic process; and that are judged as exemplary, good, and/or successfully demonstrated. A balance struck is based on gaining optimal performance, reliability, and simplicity without sacrificing those characteristics that are targeted for improvement.

When incorporated within all areas of an organization, the use of best practices, particularly in the data center, can lead to world-class performance, create repeatability, and have positive effects on both operational and capital costs. Data center staff often use more than one best practice, but unless best practices are adopted consistently across all functions of the data center, these world-class levels of performance remain out of reach.

1.4 Scope of This Document

The objectives of this document are:

- Illustrate the best practices for deploying RHEL 6, KVM, and clustered Data ONTAP together.
- Illustrate the benefits of deploying the technologies together.

Considering the preceding objectives, the scope will be limited to where Red Hat and NetApp technologies intersect or directly affect each other in the context of Red Hat Enterprise Linux (RHEL) 6, KVM, and clustered Data ONTAP 8.1. Where appropriate, configuration examples will be provided.

Although the KVM kernel module is available for most modern Linux distributions, this document specifically applies to the Red Hat implementation. References made to “RHEL 6 KVM” are not meant to imply ownership or exclusivity; it is only meant to differentiate it from KVM as deployed with other Linux distributions.

1.5 Topics Out of Scope

Both Red Hat and NetApp have extensive product documentation describing all means of deploying their respective products. This document will not attempt to duplicate product documentation. It is written with the understanding that the reader of this document has an in-depth understanding of Red Hat Enterprise Linux, KVM, and NetApp storage.

Sections of this document cover best practices for networking technologies; therefore, it will not attempt to provide instructions specific to any particular brand or model of network gear.

Executive Summary

The virtualization of a data center results in many physical systems being virtualized as part of a cost-savings effort to reduce both capital expenditures (capex) and operating expenses (opex) through infrastructure consolidation and increased operational efficiencies. However, this trend of server virtualization and consolidation is occurring at the same time as a data explosion. By 2020, IDC expects business data to increase 44 times to 35 zettabytes (1 zettabyte = 1,000, 000,000,000,000,000,000 bytes).

There is no doubt that virtualization had to happen; there was no other way to meet the demands of consolidating servers and modernizing applications. But to be clear, virtualization has to happen everywhere in the data center to meet the demands of scalability in performance, capacity, and operations. Businesses demand this type of agility from their data centers to move the business forward. The scale, complexity, and pace of business put a premium on agility.

“Business today is putting enormous pressure on IT to respond almost instantly to very challenging demands like never before. The result is an urgency to think differently and better allocate resources—both financial and operational—so that IT can quickly adapt to and drive change—creating value. Organizations will use agile data infrastructure to fundamentally rethink how they architect and manage their data storage and ultimately to advance and accelerate their business objectives,” says Steve Duplessie, founder, Enterprise Strategy Group.

Server virtualization and network virtualization are still clear requirements, but they only cover two-thirds of the data center. The original core requirements that demand virtualization still stand true today: applications must be able to be run more quickly and inexpensively while still maintaining security, performance, and manageability. These constraints and requirements are not only limited to the applications and servers; any honest discussion about application modernization must include enterprise storage considerations to be complete and holistic in scope and meet the demands of the data explosion.

Enter Clustered Data ONTAP 8.1 and Red Hat Enterprise Linux 6 with KVM

Clustered Data ONTAP 8.1 is the next-generation storage platform to deploy modern business-critical applications, both virtual and bare metal. NetApp has constantly offered the ability to efficiently scale up storage in modular increments. The introduction of clustered Data ONTAP provides the ability to scale out performance for both SAN and NAS environments, all while maintaining NetApp’s trademark storage efficiencies, unified architecture, and built-in data protection technologies.

Red Hat Enterprise Linux 6 (RHEL 6) provides a highly performing, secure, and eminently manageable operating platform on which to deploy business-critical applications. The fact that it can be deployed as a bare-metal operating system, a guest (virtual) operating system, or a hypervisor speaks volumes about its flexibility. Because of the architecture of the KVM hypervisor, virtualized applications run at near native speeds, while gaining mobility, increased manageability, and security.

What does this mean specifically for deploying virtualized applications on RHEL 6 KVM and clustered Data ONTAP? It means that the data center can literally scale on demand without losing flexibility, security, performance, or manageability. Virtual machines can be deployed in a highly accelerated manner from the storage array. Applications can operate within secure tenants that span the entire hardware, network, and storage stack. Major components of the storage infrastructure can be replaced or upgraded nondisruptively.

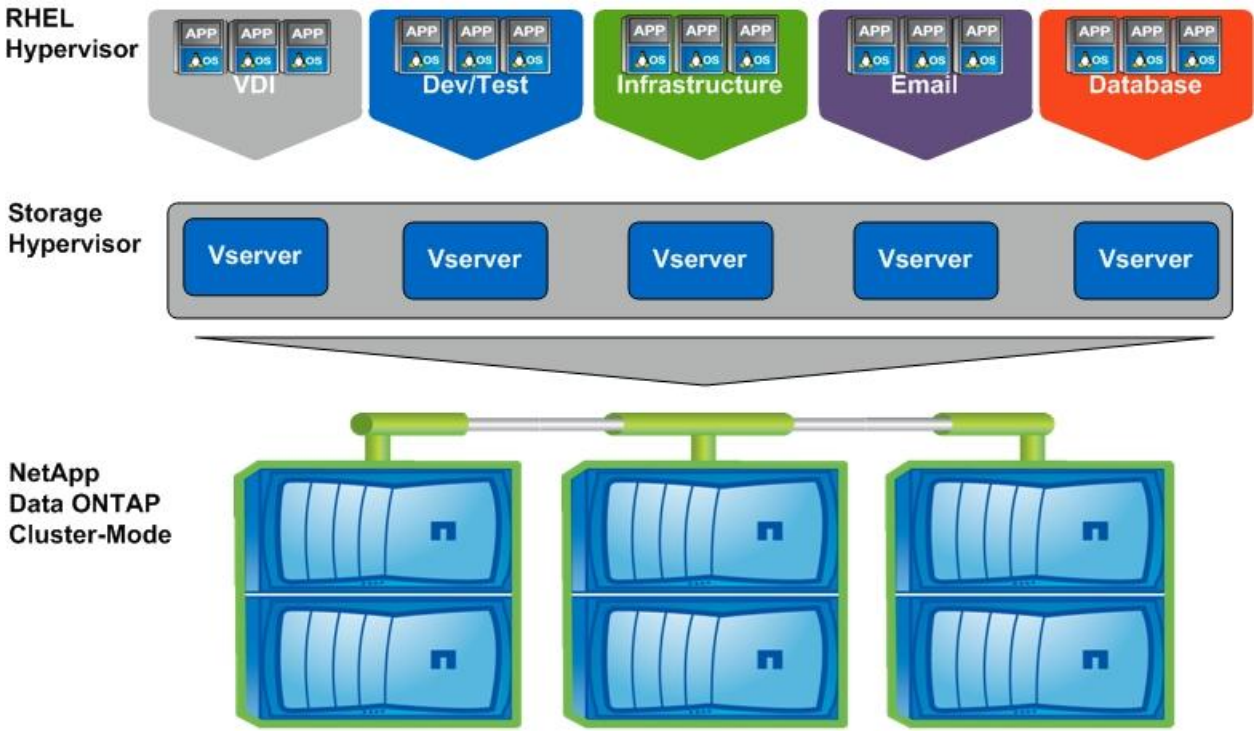
Ultimately, it provides the foundation for an agile infrastructure.

Guiding Principles for Clusters

All clustering technologies follow a common set of guiding principles. These principles include the following:

- **Nondisruptive operation.** The key to efficiency and the linchpin of clustering is the ability to make sure that the cluster does not fail ever.
- **Virtualized access is the managed entity.** Direct interaction with the nodes that make up the cluster is in and of itself a violation of the term cluster. During the initial configuration of the cluster, direct node access is a necessity; however, steady-state operations are abstracted from the nodes as the user interacts with the cluster as a single entity.
- **Data mobility and container transparency.** The end result of clustering—that is, the nondisruptive collection of independent nodes working together and presented as one holistic solution—is the ability of data to move freely within the boundaries of the cluster.
- **Delegated management and ubiquitous access.** In large complex clusters, the ability to delegate or segment features and functions into containers that can be acted upon independently of the cluster means the workload can be isolated. Equally important is that the cluster must not place conditions upon which the contents therein are accessed. This should not be confused with security concerns around the content being accessed.
- Clustered Data ONTAP embodies these guiding principles. Figure 1 shows how access is virtualized with NetApp clustered storage using Vservers, ubiquitous access through multiprotocol support, and the ability to move data within the cluster depending on workload needs and intracluster relationships.

Figure 1) NetApp clustered storage overview.



2 Clustered Data ONTAP Overview

Scaling performance while controlling costs is one of the most challenging efforts in the data center. High performance, technical computing, and digital media content applications place extreme demands on storage systems. Compute clusters running these applications can require multiple gigabytes per second of performance and many terabytes—or even petabytes—of capacity. To maintain peak application performance, users must be able to add storage and move data between systems and tiers of storage without disrupting ongoing operations. At the same time, to control costs, users must be able to effectively manage the storage environment.

Clustered Data ONTAP addresses these challenges and provides high-performance and high-capacity requirements. It enables organizations to address faster time to market by providing massive throughput and the scalability necessary to meet the demanding requirements of high-performance computing and virtualization infrastructures. These high performance levels address the growing demands of performance, manageability, and reliability for both virtualized and nonvirtualized workloads.

Clustered Data ONTAP is an operating system from NetApp that includes:

- Nondisruptive operations based on a clustered file system hosted on interconnected nodes
- Multinode scaling with global namespacing technologies
- NetApp FlexVol® volume for storage virtualization
- NetApp backup and recovery solutions based on local Snapshot™ copies, replication, and mirroring

2.1 Clustered Data ONTAP Benefits

NetApp's storage clustering feature within Data ONTAP provides a number of key benefits, including the ability to:

- **Accelerate performance.** Clustered Data ONTAP uses a clustered file system technology to provide maximum input/output (I/O) throughput and remove the bottlenecks that affect production. Information can be striped as volumes across any or all of the storage controllers and disks in the system, which enables balanced levels of throughput for even a single file or volume and allows technical teams to run multiple compute jobs concurrently. When many compute nodes simultaneously require data, you can use load-balancing mirrors within Data ONTAP with a clustering system or add NetApp FlexCache® storage accelerators in front of the system to deliver much higher read throughput.
- **Simplify storage and data management.** Clustered Data ONTAP supports fully integrated storage solutions that are easy to install, manage, and maintain. Enhancing this with its global namespace capability, administrators can simplify client-side management by mapping all data volumes in the cluster into a file system tree structure that automatically maps or remaps servers to their data, even if that data is moved. By offering a single system image across multiple storage nodes, the global namespace eliminates the need for complex automounter maps and symbolic link scripts.
- **Improve data access.** Storage is virtualized at the file system level to enable all compute nodes to mount a single file system, access all stored data, and automatically accommodate physical storage changes that are fully transparent to the compute cluster. Each client can access a huge pool of information residing anywhere in the storage cluster through a single mount point.
- **Keep resources in balance without disrupting operations.** As storage nodes are added to the cluster, physical resources, including CPU, cache memory, network I/O bandwidth, and disk I/O bandwidth, are kept in balance automatically. Clustered Data ONTAP enables you to add storage and move data between storage controllers and tiers of storage without disrupting users and applications. This ushers in a whole new paradigm in which capacity increases, workload balancing, eliminating storage I/O hot spots, and component deprecation become normal parts of the data center without the need to schedule downtime. More importantly, these tasks are accomplished without the need to remount shares, modify client settings, or stop active workloads as is typically the case with traditional or other high-performance computing storage systems.
- **Simplify installation and maintenance.** Using standard Network File System (NFS) and Common Internet File System (CIFS) protocol to access clustered Data ONTAP systems without the need to install special clients, network stack filters, or code on each server in the compute cluster is the value of a unified storage product. The clustered Data ONTAP architecture also reduces or eliminates routine capacity allocation and storage management tasks, resulting in more time to address organizational goals and objectives and less time spent managing storage.
- **Meet high-availability requirements.** Along with stringent performance requirements, high reliability is important for technical applications and cluster computing. Clustered Data ONTAP leverages core NetApp software such as WAFL® (Write Anywhere File Layout), RAID-DP®, and NetApp Snapshot. RAID-DP, a high-performance implementation of RAID 6, protects against double-disk failures, and transparent node failover automatically bypasses any failed components with no interruption in data

availability. In addition to having no single point of failure, clustered Data ONTAP supports the expansion or reconfiguration of the storage infrastructure while online, enabling applications to run uninterrupted as more storage capacity, processing power, and/or throughput are added.

- **Enable continuous operations.** Clustered Data ONTAP is configured for continuous operation with the use of high-performance and modular NetApp storage components. Each system consists of one or more fabric-attached storage (FAS) building blocks, and each FAS building block is a high-availability pair of controllers (storage nodes). Multiple controller pairs form a single, integrated cluster. Clustered Data ONTAP uses Ethernet technology—Gigabit (GB) and 10GB—for server connections and for interconnecting FAS controllers. Servers can also be connected through InfiniBand using a gateway device. Each controller can support any combination of high-performance SAS and cost-effective SATA disk drives. Data can move nondisruptively between nodes or between different tiers of disk as performance requirements change. This capability makes sure that data center and IT administrators can maximize performance where needed while simultaneously improving capacity utilization.

2.2 Clustered Data ONTAP Concepts

Regardless of size and complexity, data centers and IT organizations look for cost-effective approaches to solve challenges and address requirements. Infrastructure optimization has been part of the storage industry since it began, with vendors driving technologies such as thin provisioning, deduplication, storage tiering, and so on. Certainly the broad acceptance and rapid adoption of virtualization are prime examples of how quickly technology is put into place to promote a growing cliché: do more with less. Whether it is storage, virtualization, or something completely different, any new technology brings with it concepts and terms meant to bind concrete associations to support abstract ideas. With clustered Data ONTAP, it is no different. This section introduces new terms and concepts to establish a knowledge baseline for the remainder of this document.

2.3 NetApp Clustered Storage Classifications

NetApp clustered storage can be grouped into the following categories:

- **Monocluster.** A tightly integrated hardware stack (for example, four controllers and a series of disk shelves in one metal frame). The four controllers would be configured as a four-node cluster (and each node would also be paired with another as a highly available [HA] pair) upon which *N* number of Vservers would be created.
- **Protocluster.** A homogeneous collection of nodes connected by a qualified switch (for example, multiple FAS3XXX series controllers configured in HA pairs). The protocluster is ideal for shared storage and HA. It is the basis for creating consistent SLAs that are explicitly tied to hardware capabilities and consistent resource pooling requirements.
- **Heterogeneous cluster or qualified bunch of nodes (QBON).** A heterogeneous collection of nodes connected by a qualified switch (for example, a FAS2XXX series, FAS3XXX series, third-party array front ended by a V-Series, FAS6XXX series, and so on). Clusters must allow for heterogeneity. Multiple platforms, classes of storage, and the support of different service levels: this is clustered Data ONTAP.

2.4 What Makes Up a NetApp Storage Cluster?

Although normally reserved for a glossary, it is important to address some key terms early in the text to establish a common knowledge baseline for the remainder of this publication.

- **Cluster.** The information boundary and domain within which information moves. The cluster is where high availability is defined between physical nodes and where Vservers operate.
- **Node.** A physical entity running Data ONTAP. This physical entity can be a traditional NetApp FAS controller; a supported third-party array front ended by a V-Series controller; or NetApp's virtual storage appliance (VSA), Data ONTAP-v™.

- **Vserver.** A secure virtualized storage controller that behaves and appears to the end user to be a physical entity (similar to a VM). It is connected to one or more nodes through internal networking relationships (covered later in this document). It is the highest visible element to an external consumer, abstracting the layer of interaction from the physical nodes. Based on these two statements, it is the entity used to provision cluster resources and can be compartmentalized in a secured manner to prevent access to other parts of the cluster.

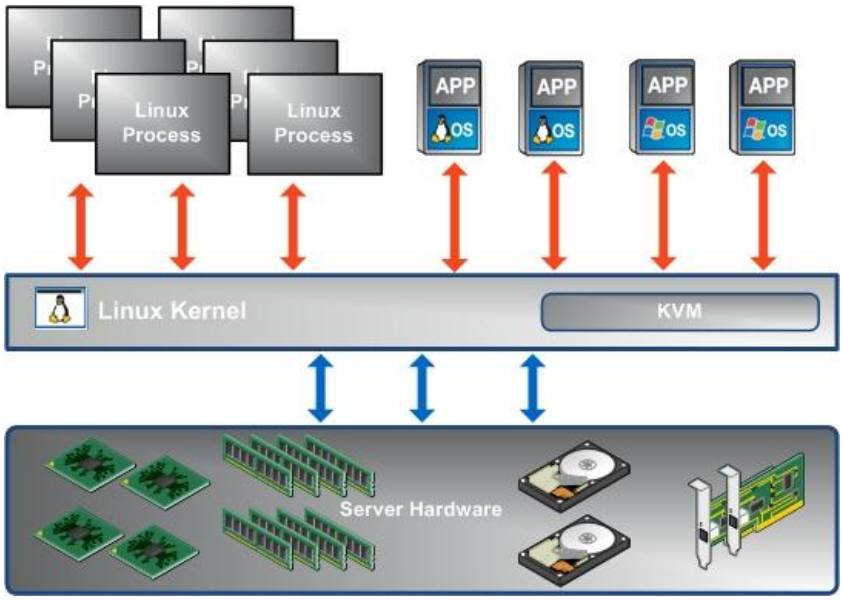
3 Red Hat Enterprise Linux 6 and KVM Overview

Red Hat Enterprise Linux 6 represents the latest major release of Red Hat's flagship operating system. It serves as the foundation for some of the IT world's most demanding applications, including databases, trading applications, high-performance grids, and others. It is optimized to be highly scalable and high performing and to do so while maintaining a high level of security.

Kernel-based Virtual Machine (KVM) is a loadable kernel module that turns the Linux kernel into a hypervisor. Because it is not a separate abstraction layer, it is in fact a "type 1" hypervisor; the hypervisor and each of the virtual machines run directly on the bare metal. The KVM-specific code was accepted by the upstream Linux kernel maintainers in January of 2007 and is now included in almost all modern Linux distributions.

Figure 2 illustrates Kernel-based Virtual Machine (KVM).

Figure 2) KVM.

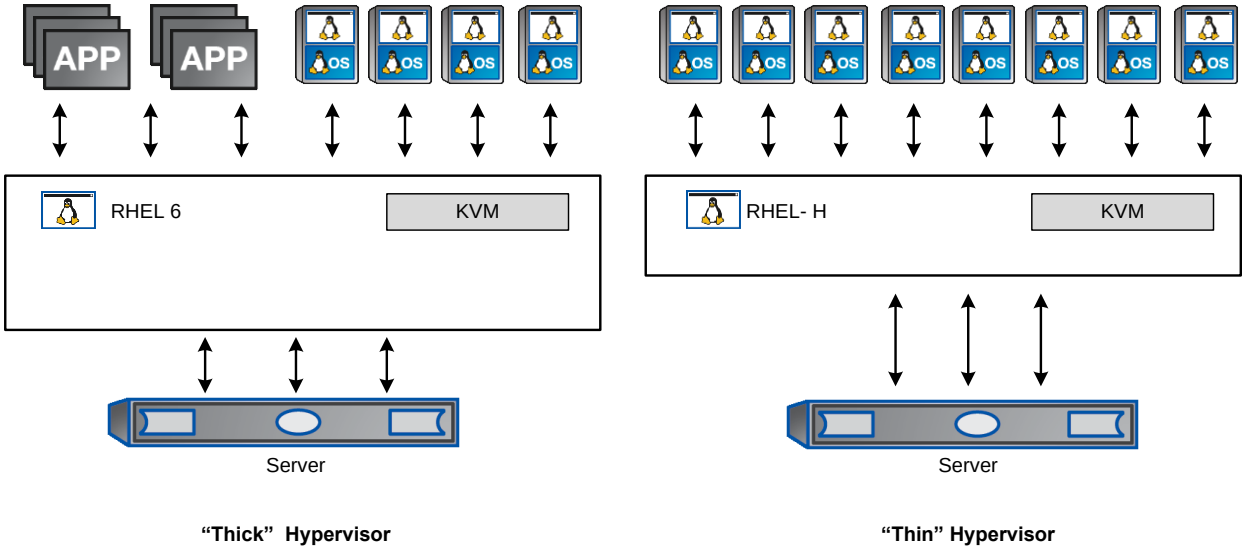


Thick and Thin Hypervisors

In the context of Red Hat, KVM can be deployed in one of two ways: a "thick" hypervisor, as deployed on Red Hat Enterprise Linux 6, or a "thin" hypervisor, as deployed in RHEV-H. Because KVM is part of the Linux kernel, both thick and thin deployments are still considered "type 1" hypervisors that run on bare metal.

Figure 3 illustrates the thick and thin hypervisors.

Figure 3) Thick and thin hypervisors.



Although both thick and thin hypervisors can be managed by RHEV-M, only RHEV-H is dependent on RHEV-M. The means of deployment, management, and integration are different when comparing thick and thin hypervisors, in addition to differences in support subscriptions. As mentioned earlier, the KVM hypervisor is available in non–Red Hat distributions. Considering these differences, it is incorrect to use the terms "KVM" and "RHEV" interchangeably. This document focuses solely on KVM as deployed with Red Hat Enterprise Linux 6 and does not discuss Red Hat enterprise virtualization.

4 RHEL 6 KVM Host Configuration

4.1 CPU and Memory Considerations

Outside of actual CPU and memory requirements published by Red Hat, some of the other considerations to follow are:

- For a production environment, multicore and multisocket CPUs should be used for providing a highest number of available virtual CPUs to virtual machines.
- Additionally, the more physical RAM available on the RHEL 6 KVM host, the more virtual memory available to the virtual machines. Preferably, use a high memory footprint (>24GB RAM).

4.2 Hardware Requirements for RHEL 6 KVM

RHEL 6 KVM hosts have the following hardware requirements:

- A 64-bit CPU with the hardware virtualization extensions; this means an AMD system with AMD-V or an Intel® system with Intel VT: maximum of 160 logical CPUs (4,096 theoretical limit).

To check that the CPUs have either the AMD extensions (svm) or the Intel extensions (vmx), run the following command:

```
egrep --color 'svm|vmx' /proc/cpuinfo
```

The output will be similar to the following, where a match of "svm" or "vmx" will be highlighted in red, confirming the extension is present:

```
flags           : fpu vme de pse tsc msr pae mce cx8 apic mtrr pge mca cmov pat pse36 clflush dts
acpi mmx fxsr sse sse2 ss ht tm pbe syscall nx pdpe1gb rdtscp lm constant_tsc arch_perfmon pebs
```

```
bts rep_good xtopology nonstop_tsc aperfmperf pni pclmulqdq dtes64 monitor ds_cpl vmx smx est tm2
ssse3 cx16 xtpr pdcm dca sse4_1 sse4_2 popcnt aes lahf_lm ida arat epb dts tpr_shadow vnmi
flexpriority ept vpid
```

- Additionally, the CPU feature Extended Page Tables (EPT) in Intel processors or Rapid Virtualization Indexing (RVI) in AMD processors is highly recommended for a KVM host, though it is not mandatory.
- At least one network controller with a minimum bandwidth of 1Gbps; NetApp and Red Hat best practices dictate at least two network controllers (10 Gigabit Ethernet [10GbE] for storage networks is preferred).
- At least 2GB of RAM for the RHEL 6 KVM host plus sufficient RAM for VMs, depending on guest OS requirements and workloads; 2GB is the minimum, and NetApp strongly recommends that additional RAM be available for virtualization: maximum of 2TB host RAM (64TB theoretical limit).
- 6GB of disk space plus the required space for each guest operating system.
- If overcommitting host memory, additional storage space will be required for swap. Refer to the RHEL 6 Virtualization Host documentation.
- Shared storage to support advanced virtualization capabilities (live migration, copy offload, clone offload, and so on).

  Note: For all hardware requirements, refer to the RHEL 6 Installation Guide and the RHEL 6 Virtualization Guide.

Network Cards and HBAs

An RHEL 6 KVM host should have at least one onboard Gigabit Ethernet (GbE) network card for management traffic as well as at least two 10GbE ports for storage traffic. Additionally, some form of out-of-band (OOB) ports should be available for power management and remote access. Multiple GbE network cards can be used if 10GbE is not available.

If attaching to Fibre Channel (FC) SAN, at least one FC host bus adapter (HBA) will be required. Although software-based initiators are supported, hardware-based iSCSI HBAs are recommended. The most recent list of Red Hat supported hardware components can be found at http://hardware.redhat.com/hcl/.

Boot from SAN

RHEL 6 KVM hosts should absolutely be installed to and boot from SAN (FC, FCoE, or iSCSI). This provides multiple benefits in the context of efficient use of storage, host mobility, disaster recovery, and backup. Boot LUNs on the NetApp controller can easily be recovered, deduplicated, and in some cases migrated. Refer to the instructions for your particular model of SAN HBA and server BIOS for enabling SAN boot.

Disk Layout for RHEL 6 KVM Host

The disk layout for RHEL 6 KVM hosts should be a balance between Red Hat best practices as well as the requirements of the data center hosting the virtual environment. Red Hat has recommendations for swap, depending on the size of the physical RAM, that should absolutely be followed. It is also a best practice to have the /boot and /var directories on separate partitions. Separate partitions for /home and other major directories are not required for the host nodes.

A Red Hat Enterprise Linux KVM host should have the following partitions created:

- A “/” partition of between 3GB and 6GB
- A “swap” partition (size is determined by amount of physical memory)
- A “/boot” partition of 250MB
- A “/var” partition of 1GB

To maintain optimal I/O performance, suitable file system alignment between a host and external storage or virtual machines and underlying storage must be considered. Although earlier versions of Red Hat

Enterprise Linux required steps to be taken to make sure of the proper file system alignment , RHEL 6 properly aligns by default. There is additional information later in the next few sections of this technical report. For a full explanation and additional details, see NetApp TR-3747: Best Practices for File System Alignment in Virtual Environments.

4.3 Package Selection

The packages installed on an RHEL 6 KVM host should be chosen on a very selective basis. Although graphical interface packages can be installed, the best practice to follow is to avoid a graphical environment on the RHEL 6 KVM host. If a graphical interface is required, then it is recommended to use a separate remote host that has the graphical packages installed. This makes sure that all available resources (CPU, RAM, and I/O) are available to support virtual machines.

Package groups to install in addition to the base server installation include:

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

Other packages to avoid on an RHEL 6 KVM host include development libraries, network sniffers, and services that are not needed. A good rule of thumb is if it does not support, service, or protect virtual machines, then do not install it. Not only does this leave more resources available to virtual machines, but also there are fewer packages to update and fewer security concerns.

4.4 KVM Host-Node Configuration

It is expected that the reader of this document has experience with Red Hat Enterprise Linux 6 and does not need detailed instructions on basic configuration of the RHEL 6 KVM host. With this in mind, basics such as basic networking, basic installation, and other basics are not covered in this report.

Register an RHEL 6 KVM Host to Red Hat Network

It is a best practice to subscribe all RHEL hosts and guests to Red Hat Network (RHN) to receive updates, patches, and fixes. This requires a valid subscription to RHN.

Configuring the libvirt Daemon

The primary purpose of the RHEL 6 KVM host is to service and protect virtual machines. To run virtual machines, the “libvirt” daemon must be running at all times and starting automatically at boot.

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

```
chkconfig libvirtd on
```

2. Check to see that “libvirt” is running with the following command:

```
service libvirtd status
```

3. If the libvirt daemon needs to be started, execute 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.

To configure Red Hat Enterprise Linux (RHEL) NTP, perform the following steps:

4. Log in to the host and perform an initial time sync.

```
service ntpd stop
ntpdate -q <IP_address_of_NTP_server>
```

Note: If the NTP service is not already running, the service ntpd stop command will fail. This is expected and acceptable.

5. Edit the `/etc/ntp.conf` file.

```
server <IP_address_of_NTP_server>
```

6. Enable the NTP service to start automatically on boot.

```
chkconfig ntpd on
```

7. Start the NTP service.

```
service ntpd start
```

8. To confirm that the NTP daemon (NTPD) is correctly receiving the data from the NTP servers, use the NTP query tool.

```
ntpq -p
```

9. This lists the NTP server(s) from which the NTP client is currently getting time updates.

Note: Repeat steps 1 through 5 on each host that must be configured.

10. Open port 123 (both UDP and TCP) on the IPtables firewall to allow NTP traffic.

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

VLAN Tagging

To maintain security and separation, it is a best practice to configure VLAN tagging at the network switch. It is a best practice to use VLANs to maximize network bandwidth on 10GbE interfaces as well as 1GB interfaces that are configured with channel bonding.

To configure VLAN tagging, perform the following steps:

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, VLAN tag 100.
4. Disable the Network Manager service by executing the following commands:

```
service NetworkManager stop
```

```
chkconfig NetworkManager off
```

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

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

```
DEVICE=eth0.100
BOOTPROTO=static
ONBOOT=yes
IPADDR=192.168.100.10
NETMASK=255.255.255.0
VLAN=yes
USERCTL=no
```

Note: Replace the “IPADDR” and “NETMASK” values with entries that match your own server.

6. Bring the interface up by executing the following command:

```
ifup eth0.100
```

Note: The steps to configure VLAN tagging are identical for 100MB, 1GB, and 10GbE Ethernet devices.

Note: If using VLAN tags in conjunction with channel bonding, create the channel bond first. For example, VLAN interfaces ifcfg-bond0.100, ifcfg-bond0.110, and ifcfg-bond0.120 might be configured on channel bond 0.

Network Configuration to Support Virtual Machines

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 default virtual bridge provides IP addresses (through internal DHCP) to the KVM guests. The default virtual bridge 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 at least one additional virtual public bridge. Additional virtual bridges can be configured for private networks as well as providing direct storage access from a virtual machine to storage.

Creating a virtual bridge involves taking over an interface (can be physical or a VLAN as created earlier) and editing the standard configuration file to reflect a bridge device of a standard Ethernet device. Create a virtual bridge by performing the following steps:

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 reflect the values for “BRIDGE” and “BOOTPROTO”:

```
DEVICE="eth2"
HWADDR="00:19:99:BB:FF:84”
NM_CONTROLLED="no"
ONBOOT="yes"
BRIDGE=br0
BOOTPROTO=none
```

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 key variables listed here are the “DEVICE” and “TYPE”:

```
DEVICE=br0
TYPE=Bridge
BOOTPROTO=static
IPADDR=192.168.29.50
NETMASK=255.255.255.0
```

```
ONBOOT=yes
DELAY=0
```

Note: The “Bridge” value for “TYPE” must start with a capital “B.”

4. Restart networking on the server:

```
service network restart
```

This creation of virtual bridges also requires a configuration to the IPtables firewall to forward all virtual machine traffic through the virtual bridge.

5. Allow traffic bound for virtual machines to pass through the KVM host firewall by executing 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.

Note: Virtual bridges can also be used in conjunction with channel bonding and VLAN tagging.

6. Disable netfilter on the virtual bridges by adding the following lines to /etc/sysctl.conf:

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

7. Make the changes in /etc/sysctl.conf by executing the following command:

```
sysctl -p
```

4.5 General KVM Host Security Guidelines

Host and guest security is paramount in protecting data. In this light, the following section provides some elementary, but critical guidelines. This is in no way a complete or comprehensive list of items to address in putting together a solid set of operating procedures around security. It is only meant to be a starting point when looking at RHEL 6 KVM security. For additional information on Red Hat and security, refer to the Security and Security-Enhanced Linux guides available for Red Hat Enterprise Linux 6.

1. Create separate logins for each required administrator and have all administrators be part of the same group, such as “wheel” or other.

   Note: Do not use a single “root” login and password.

2. Enforce the use of strong passwords based on the Red Hat Knowledge Base article at https://access.redhat.com/kb/docs/DOC-9128.
3. Determine which services are running on the RHEL 6 KVM host. Document the reason that each service is running or needed. If there is no requirement or reason to run the service, disable it and remove it if possible. To list out all services that are configured to start upon booting the server, execute 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 primary means of connecting to a virtual server and the virsh (virtual shell) console is by SSH and SSH tunnels, respectively. It is also possible to configure communication to the virsh console with the use of Transport Layer Security (TLS).

For secure communication between RHEL 6 KVM hosts, it is a best practice to set up and use SSH keys. If using a remote host to manage the KVM environment with graphical tools, then the SSH keys can be added to that remote host as well. This will allow for seamless automation tasks between hypervisors, such as with live migration of virtual machines.

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. Distribute the keys between the two hosts.
   a. On host1, type:

```
ssh-copy-id host2
```

   This should result in a prompt for password and a suggestion to try the login. If the login succeeds without being asked for a password, it was successful.

   b. On host2, type:

```
ssh-copy-id host1
```

   This should result in a prompt for password and a suggestion to try the login. If the login succeeds without being asked for a password, it was successful.

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

For instructions on how to configure hypervisor access using TLS, follow the instructions in this excellent Red Hat Knowledge Base article: https://access.redhat.com/knowledge/solutions/49120.

Firewall and Mandatory Access Control

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. The best practice is to leave IPtables running and open ports as necessary rather than disable it.

Determine the ports that need to be opened and allow them using the following procedures. See the appendix for a list of RHEL 6 KVM and NetApp specific 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 executing the following command:

```
iptables -L
```

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

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

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

```
# service iptables save
```

Refer to the appendix for a complete list of RHEL 6 KVM and NetApp related ports that need to be accounted for in all operating firewalls, including IPtables.

SELinux was developed largely by the National Security Agency (NSA) (and later incorporated into the 2.6 Linux kernel in 2003) to comply with the 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 such as files, devices, and processes.

In RHEL 6, KVM is configured to work with SELinux by default. The best practice is to leave SELinux enabled.

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

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

3. Any block-based storage 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. Execute the following two commands to change the security context:

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

Create an RHEL 6 KVM Host Template

After the first RHEL 6 KVM host has been created, take a few additional steps to create a template or golden image based on it. The rationale is that all subsequent RHEL 6 KVM hosts can be cloned by the NetApp storage much faster as compared to both creating them from scratch or even using traditional provisioning tools.

After the initial RHEL 6 KVM build is complete, the process to create a template is fairly straightforward. Essentially, it should be “made generic” by removing configuration artifacts such as host name, MAC addresses, and so on.

Not only does cloning RHEL 6 KVM hosts speed up deployment significantly, it also enables highly efficient use of storage. As new KVM hosts are created from the same template, the underlying storage can be deduplicated for significant reduction in space used.

Here are the high-level steps to creating and using an RHEL 6 KVM host template:

1. Install and create an RHEL 6 KVM host that includes all of the packages, security, and network configuration as described in the preceding sections.
2. Register the RHEL 6 KVM host to RHN and update all packages.
3. Remove “configuration artifacts,” thereby making the template generic.
4. Clone the boot LUN.

When a new RHEL 6 KVM host is needed, the template is cloned in a few seconds instead of going through all of the steps necessary to go through more traditional provisioning procedures. See the appendix for more complete steps that can be taken to make a template generic as well as the cloning procedures.

4.6 RHEL 6 KVM Datastores and File Types

This section describes the recommendations for datastores and file types and how to best plan for their deployment as well as their protection.

The 80/20 Rule and Determining a Storage Protocol to Use

In designing the storage architecture for a virtual data center, one can apply what is referred as the 80/20 rule in this document, which is that 80% of all systems virtualized are for consolidation efforts. The remaining 20% of the systems are classified as business-critical applications, and although they can be virtualized successfully, they tend to be deployed on shared storage pools. NetApp refers to these pools as isolated datasets.

Consolidated datasets have the following characteristics:

- Virtual machines that do not require application-specific backup and restore agents: a “crash-consistent” Snapshot copy of the underlying NetApp volume will suffice.
- Consolidated datasets typically contain the largest number of virtual machines.
- The virtual machines within do not typically have large network or storage I/O requirements, but collectively might represent a resolvable challenge.
- Consolidated datasets are ideally served by large, shared, policy-driven storage pools (or datastores).

Isolated datasets (for business-critical applications) have the following characteristics:

- Virtual machines that require application-specific backup and restore agents
- Each individual VM might address a large amount of storage and/or have high I/O requirements
- Storage design and planning applied in a similar manner as to the physical servers
- Datasets are ideally served by individual, high-performing, and nonshared datastores

Consolidated datasets work well with NFS datastores because this design provides greater flexibility in terms of capacity and flexibility than SAN datastores when managing hundreds or thousands of virtual machines (VMs). Isolated datasets run well on all storage protocols; however, some tools or applications might have restrictions around compatibility with NFS and/or Red Hat supported file systems on SAN datastores.

In most cases, the data center evolution from physical to virtual follows the 80/20 rule, and the native multiprotocol capabilities of NetApp and RHEL 6 KVM enable systems to become virtualized more quickly and easily than with a traditional storage array platform or multiple disparate storage arrays.

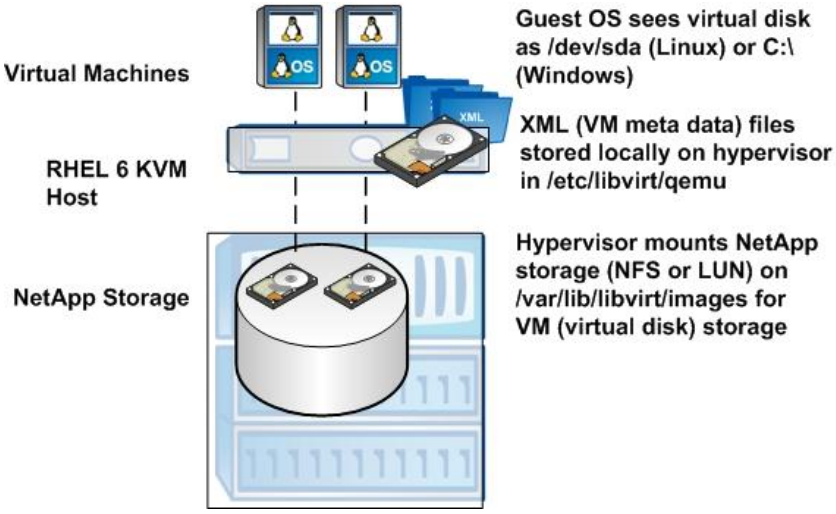
RHEL 6 KVM File Types

RHEL 6 KVM employs two primary types of files in the context of virtual machines: a disk image and an Extensible Markup Language (XML) descriptor file. The disk image is the “virtual block device” from which the virtual machine boots and on which it stores data. The XML descriptor file provides all of the metadata for a virtual machine, including the virtual machine UUID, network MAC address, disk image location, and other critical information.

The disk image files reside on the shared storage, but by default the XML descriptor files are saved in a directory that is local to the RHEL 6 KVM host.

Figure 4 illustrates disk and file locations.

Figure 4) Disk and file locations.



Because of this, the XML files need to be accounted for in any backup, disaster recovery, and site failover strategies. There are multiple ways of handling this, including creating a link between the default XML directory and the shared storage. Disk image files are discussed in the subsequent section.

Raw Disk Image Files and QCOW2 Disk Image Files

RHEL 6 KVM supports the use of two different disk image formats:

- Raw: A raw disk image is a faster format that supports both thick and sparse allocation.
- Qcow2: Qcom has some advanced features such as virtual machine–based Snapshot copies, but at the cost of performance.

The best practice when deploying RHEL 6 KVM on NetApp storage is to use raw disk files and allow the NetApp controller to thin provision, deduplicate data, and provide Snapshot copies on the underlying storage. It can do so much more quickly and efficiently without involving hypervisor CPU cycles.

File System Alignment

A more defined section on file system alignment exists later in this document. However, it is important to note that RHEL 6 properly aligns file systems and partitions to underlying storage by default. Therefore, no further considerations need to be made for LUN-based storage under RHEL 6.

4.7 LUN-Based Datastores

Overview

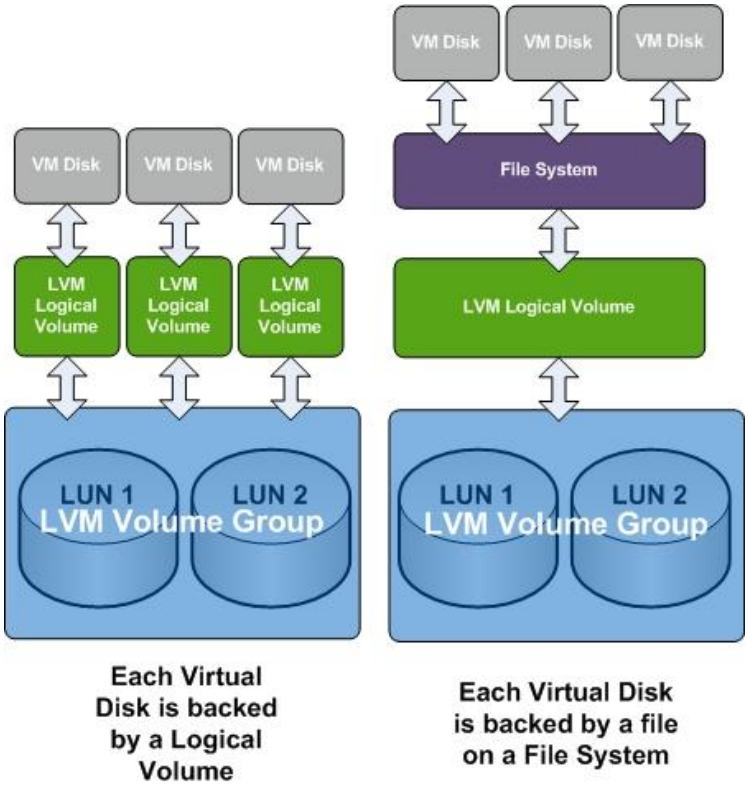
Red Hat provides and supports multiple means of utilizing LUN-based storage.

The primary choices in a production datacenter are:

- Red Hat Enterprise Linux standard file systems such as EXT3 or EXT4
- Red Hat Enterprise Linux Logical Volume Manager (LVM2)

To be clear, both choices actually make use of LVM; the difference is whether or not a file system is placed on top of LVM. It is a best practice to utilize LVM on local and external disks and LUNs. Figure 5 illustrates the difference between using LVM alone or in conjunction with a file system such as EXT3 or EXT4.

Figure 5) Comparing LVM only versus LVM plus file system in storing virtual disks.



The standard Red Hat Enterprise Linux file systems provide a stable and performing means of utilizing a LUN-based datastore. The consideration to be made with EXT3 or EXT4 is that there has to be a guarantee that any given virtual machine is only running on one hypervisor at any given moment. That is to say that if a virtual machine is running on one RHEL 6 KVM host, that it is not started on purpose or accidentally on another RHEL 6 KVM host. This will cause severe corruption to the virtual machine.

Allowing KVM to manage the datastore directly as opposed to allowing RHEL to manage it can mitigate this. Another option is to add a conditional test to any automation scripts that start or stop virtual machines. One final option is to use a clustered file system.

Logical Volume Manager (LVM2) is typically used as a layer of abstraction between a block-based storage device (local disk, direct attach, SAN, and so on) and the file system. However, LVM can also be used as a means of managing LUNs and individual virtual machines without using a file system between

the logical volume and the virtual machine. Essentially, a LUN is configured as an LVM volume group, and virtual machines are created as individual logical volumes within the volume group. This is actually how LUNs are managed under Red Hat enterprise virtualization.

Virtual machines that are stored on a LUN-backed file system (EXT3 or EXT4) are created as a simple file that acts as a block device. If using Logical Volume Manager without a file system, the logical volume is the block device for the virtual machine. An additional option for LVM is clustered LVM (CLVM). This allows multiple hosts to read and write from the same LUN-based datastore.

Spanning LUN Datastores

Regardless of whether LVM is used with or without a file system, it offers the benefit of growing the virtual machine dynamically if necessary. If additional storage is needed for the datastore, a new LUN is created on the NetApp controller, and it is added to the pool of storage within LVM on the RHEL 6 KVM host.

In the case of using LVM alone, this provides the means for additional logical volume–backed virtual machines to be created. In the case of using LVM with a file system, once the LVM volume group is extended, the file system is also extended. The combination of LVM on the Red Hat side and the flexible storage on the NetApp side allows traditional scaling obstacles to be quickly overcome.

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.

FC and FCoE LUNs

If the NetApp FC or FCoE LUNs are created and presented after the RHEL 6 KVM host is booted, it might be necessary to rescan the SCSI bus. To rescan the SCSI bus, either reboot the host or perform the following steps:

1. Issue a Loop Initialization Protocol (LIP) by executing the following command:

```
echo "1" > /sys/class/fc_host/hostX/issue_lip
```

   Where "hostX" matches the specific HBA number that is to be rescanned. This will need to be repeated on all HBAs that are connected and multipathed to the FC or FCoE LUN.

2. Add the storage device to the system by executing the following command:

```
echo "- - -" > /sys/class/scsi_host/hostX/scan
```

   Where "hostX" is the HBA to which the new LUN is attached. This will need to be repeated on all of the HBAs that are connected and multipathed to the FC or FCoE LUN.

iSCSI LUNs

To configure a software-based iSCSI initiator, perform the following steps:

1. If the iSCSI initiator tools have not already been installed, execute the following command:

```
yum install iscsi-initiator-utils
```

2. Start the iSCSI service and configure it to start automatically on boot by executing the following command:

```
service iscsi start; chkconfig iscsi on
```

3. Verify the iSCSI initiator name by executing the following command:

```
cat /etc/iscsi/initiatorname.iscsi
```

4. Add CHAP user names and passwords to the /etc/iscsi/iscsid.conf file.

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

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

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. Assuming that there are multiple paths to the iSCSI target, continue to the section “Configuring RHEL 6 DM-Multipath.”

View All Disks and Partitions

1. From the RHEL 6 host, execute 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 “Configuring RHEL 6 DM-Multipath” to complete the steps.

Configuring RHEL 6 DM-Multipath

This is relevant to all LUN-based storage protocols under RHEL 6. Note that the procedures for DM-Multipath under RHEL 6 is very different as compared to earlier versions.

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

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

2. Run the following command to automatically configure DM-Multipath:

```
mpathconf –enable --with_multipathd y --user_friendly_names n find_multipaths y
```

Note: This will automatically configure the multipath.conf file, start the DM-Multipath service, enable the service at boot, disable “friendly names”, and automatically blacklist local disks.

3. To view the and confirm the multipath configuration, run the multipath –ll command:

```
multipath -ll
360a98000572d4273685a687844474c6a dm-0 NETAPP,LUN
size=200G features='0' hwhandler='0' wp=rw
`-+- policy='round-robin 0' prio=4 status=active
  |- 6:0:0:1 sdb 8:16 active ready running
  `- 6:0:1:1 sdd 8:48 active ready running
360a98000572d4273685a6963472d326e dm-1 NETAPP,LUN
size=150G features='0' hwhandler='0' wp=rw
`-+- policy='round-robin 0' prio=4 status=active
  |- 6:0:0:0 sda 8:0  active ready running
  `- 6:0:1:0 sdc 8:32 active ready running
```

Note: This sample output highlights that there are 2 different multipath devices - dm-0 (200G) and dm-1 (150G) and that all devices and paths are operational.

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 executing 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 executing 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.

4.8 NFS Datastores

NetApp FAS controllers allow customers to leverage enterprise-class NFS arrays to provide datastores with concurrent access by all of the nodes utilizing the shared storage. The NetApp NFS provides high performance, the lowest per-port storage costs, and some advanced data management capabilities.

Configure NFS Client to Use Consistent 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 the NFS ports that are used, it is a best practice to configure the two primary NFS client services to use the same ports from connection to connection and host to host.

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.

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 NFS client port configuration 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 executing the following command:

```
showmount -e 192.168.29.100 | grep "/vol/kvm_export"
```

Note: Replace the IP address and NFS export volume with your own values.

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

```
echo "192.168.29.100:/vol/kvm_export /var/lib/libvirt/images nfs  _netdev,defaults 0 0" >>
/etc/fstab
```

Note: The "_netdev" option is required so that on boot, mounting of the NFS export is not attempted until all network devices are up and operational.

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

```
mount -a
```

NFS Datastores on NetApp

Deploying RHEL 6 KVM with the NetApp advanced NFS results in a high-performance, easy-to-manage implementation that provides VM-to-datastore ratios that cannot be accomplished with other storage protocols, such as FC. This architecture can result in a 10x increase in datastore density with a correlated reduction in the number of datastores. With NFS, the virtual infrastructure receives operational savings because there are fewer storage pools to provision, manage, back up, replicate, and so on.

Through NFS, customers receive an integration of RHEL 6 KVM virtualization technologies with WAFL® (Write Anywhere File Layout), the NetApp advanced data management and storage virtualization engine. This integration provides transparent access to VM-level storage virtualization offerings such as production-use data deduplication, immediate zero-cost VM and datastore clones, array-based thin provisioning, array-based virtual machine cloning, automated policy-based datastore resizing, and direct access to array-based Snapshot copies.

Virtual machines that are backed by NFS-based datastores are created as simple files that act as block devices.

4.9 Datastore Comparison Tables

Differentiating what is available with each type of datastore and storage protocol can require considering many points.

Table 1 compares the features available with each storage option.

Table 1) Datastore supported features.

Capability/Feature	FC/FCoE	iSCSI	NFS
Format	EXT3, EXT4, LVM	EXT3, EXT4, LVM	NetApp WAFL
Optimal queue depth per LUN/file system	64	64	N/A
Available link speeds	4 and 8GB FC and 10GbE	GbE and 10GbE	GbE and 10GbE

Table 2 compares storage-related functionality of RHEL 6 KVM features across different protocols.

Table 2) Red Hat–supported storage-related functionality.

Capacity/Feature	FC/FCoE	iSCSI	NFS
Live migration	Yes	Yes	Yes
NetApp cloned datastores	Yes	Yes	Yes
NetApp cloned virtual machines	No	No	Yes
Data deduplication	Yes	Yes	Yes
Resize datastore	Grow only	Grow only	Grow, autogrow, and shrink

Capacity/Feature	FC/FCoE	iSCSI	NFS
Thin provision datastores	Yes	Yes	Yes
NetApp Snapshot copies	Yes	Yes	Yes
Restore datastores and VMs from SnapMirror® and SnapRestore®	Yes	Yes	Yes
Boot from SAN	Yes	Yes with HBAs	No

5 RHEL 6 KVM Guest Configuration

This section highlights the best practices in configuring virtual guests.

Virtual Guest Limits

RHEL 6 KVM supports the configurations listed in Table 3.

Table 3) Red Hat supporting configurations.

Components	Units
Virtual CPUs	A maximum of 160 per guest
Virtual RAM	A maximum of 2TB per 64 bit guest
Virtual storage devices	A maximum of 16 per guest
Virtual network devices	A maximum of 16 per guest
Virtual PCI devices	A maximum of 32 per guest

RHEL 6 KVM Virtualized Guest Support

RHEL 6 KVM presently supports the following virtualized guest operating systems:

- Red Hat Enterprise Linux 3 (32 bit and 64 bit)
- Red Hat Enterprise Linux 4 (32 bit and 64 bit)
- Red Hat Enterprise Linux 5 (32 bit and 64 bit)
- Red Hat Enterprise Linux 6 (32 bit and 64 bit)
- Windows® XP Service Pack 3 and newer (32 bit only)
- Windows 7 (32 bit and 64 bit)
- Windows Server® 2003 Service Pack 2 and newer (32 bit and 64 bit)
- Windows Server 2008 (32 bit and 64 bit)
- Windows Server 2008 R2 (64 bit only)

Paravirtualized Drivers Support

The paravirtualized block and network drivers (the virtio drivers) support the following operating systems and versions. It is a best practice to use the virtio drivers as provided by Red Hat to optimize the performance for a guest's block and network devices. Virtio drivers are available for the following operating systems:

- Windows XP

- Windows 7 (32 bit and 64 bit)
- Windows Server 2008 (32 bit and 64 bit)
- Windows Server 2003 R2 (32 bit and 64 bit)
- Red Hat Enterprise Linux 4.8 and newer (32 bit and 64 bit)
- Red Hat Enterprise Linux 5.4 and newer (32 bit and 64 bit)
- Red Hat Enterprise Linux 6.0 and newer (32 bit and 64 bit)

5.1 File System Alignment Overview

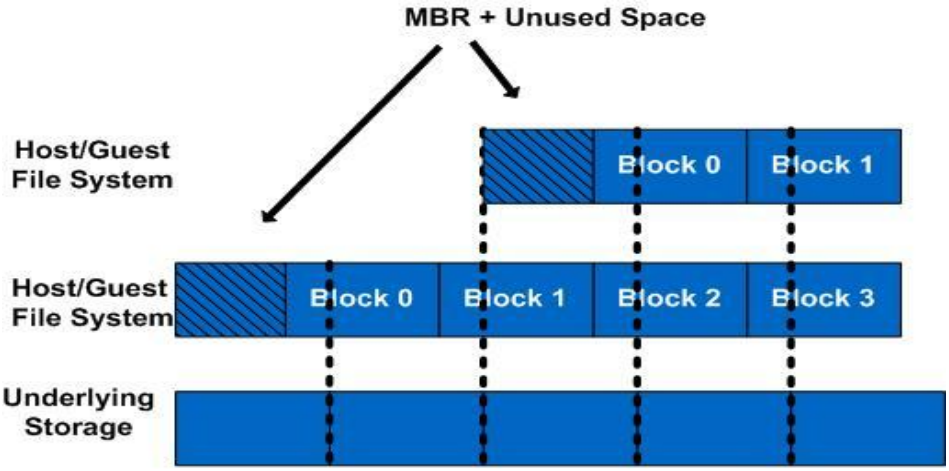
In any virtual environment, a number of layers of abstraction between physical disks and the VM's virtual disk exists. Each layer in turn is organized into blocks to make the most efficient use of storage. The focus is not the size of the block, but rather the starting offset.

To avoid latency caused by additional reads and writes, the starting offset of a file system on a virtual machine should line up with the start of the block at the next layer down and continue that alignment all the way down to data blocks at the aggregate layer on the NetApp controller.

This is in no way unique to NetApp; it applies to any storage vendor. It is a simple byproduct 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.

Figure 6 illustrates misaligned file system.

Figure 6) Misaligned file system.

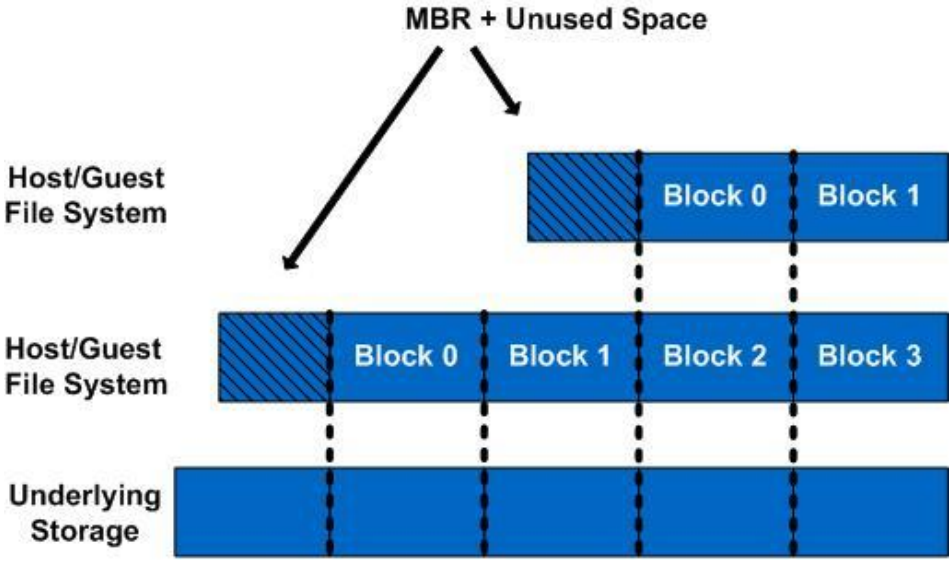


Without this appropriate alignment, significant latency occurs because the storage controller has to perform additional reads and writes for the misaligned blocks. For example, most modern operating systems such as RHEL and Windows 2000 and 2003 use a starting offset of sector 63. Pushing the offset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

Microsoft® Windows Server 2008, Windows 7, and RHEL 6 all align properly by default and require no additional consideration. However, earlier versions of Microsoft Windows (Windows Server 2003, Windows XP, and so on) and RHEL (3, 4, and 5) all require additional steps at deployment time to facilitate proper file system alignment.

Figure 7 illustrates a properly aligned file system.

Figure 7) Properly aligned file system.



5.2 Thick or Thin Provisioning of KVM Guests

RHEL 6 KVM allows for both thick and thin provisioning of guest virtual disks. The general recommendation from Red Hat is to thick provision production guests and thinly provision desktops and dev/test guests to balance performance and storage capacity.

However, when coupled with NetApp, the underlying storage can be thin provisioned and deduplicated. This allows all RHEL 6 KVM guests to be thick provisioned, but still maintain an efficient storage environment. The best practice is to thick provision the KVM guests, but thin provision and deduplicate the underlying storage.

5.3 Provisioning Virtual Machines in RHEL 6 KVM

The steps in this section provide the necessary guidance for provisioning virtual machines on the KVM hypervisor in RHEL 6 using the command line. The following steps assume that a Kickstart server is available on the network as well as the configuration of bridged networking on the RHEL 6 KVM host, as described earlier in this document.

Create a Virtual Machine Disk

The “qemu-img” tool is native to KVM and can be used to create thick (raw disk) and thin (qcow2) virtual disks for the virtual machines. As described earlier in this document, the best practice for deploying virtual machines on RHEL 6 KVM is to use thick virtual machines on thinly provisioned NetApp storage.

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

```
qemu-img create -f raw /var/lib/libvirt/images/new_vm.img 16g
```

Note: Tools such as “fdisk” or “parted” can be used to partition the newly created virtual disk prior to deploying a guest operating system. This will be necessary to facilitate proper file system alignment on guest operating systems prior to RHEL 6 or Windows Server 2008.

2. Provision an RHEL 6 virtual machine with 2 CPUs and 2GB of RAM; with the 16GB disk created earlier that uses the bridged network br0, execute 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/new_vm.img \
--file-size=16 --vnc --os-variant=rhel6 \
--location=http://yum_repo.infra.example.com/rhel6 -x \
ks=http://kickstart.infra.example.com/rhel6_guest.ks
```

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 completes 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 prior to Version 6 or a Microsoft operating system released prior to 2008, additional deployment steps need to be followed to facilitate proper file system alignment. See the next section.

Create a KVM Guest Template

Instead of creating new virtual machine guests from scratch each time, it is a best practice to create a template or series of templates. When new virtual machines are needed, the NetApp controller can clone NFS-based virtual machines very quickly.

The concept of creating a template for virtual machines is almost identical to that of creating a template for RHEL 6 KVM hosts. The base image is created using Kickstart or other provisioning tool, and then the image is "made generic." When new guests are needed, FlexClone® is used to clone NFS-based guests. In the case of LUN-based guests, the KVM native "`virt-clone`" command is used.

NetApp storage is much faster as compared with both creating NFS-based virtual machines from scratch or even using traditional provisioning tools.

After the initial virtual machine deployment is complete, the process to create a template is fairly straightforward. Essentially, it should be "made generic" by removing configuration artifacts such as host name, MAC addresses, and so on.

As in the case for RHEL 6 KVM hosts, cloning virtual machines using the storage instead of the hypervisor speeds up deployment significantly; it also enables highly efficient use of storage. As new virtual machines are created from the same template, the underlying storage can be deduplicated for significant reduction in space used.

Using the NetApp storage to perform the cloning means that virtual machines can be spun up on demand to meet business needs. Additionally, it means that in a recovery situation, VMs can be cloned much more quickly than restoring from backup.

Here are the high-level steps to creating and using an RHEL 6 KVM host template:

1. Install and configure the virtual machine that includes all of the packages, security, and network configurations needed to make the VM production ready.
2. If using a Red Hat–based VM:
3. Register the VM operating system to RHN and update all packages.
4. Remove "configuration artifacts," thereby making the template generic. See the appendix for detailed instructions.
5. If using a VM based on Microsoft:
   a. Apply all Microsoft updates.
   b. Use the Microsoft "sysprep" tool to make the VM generic.
6. Clone the VM. See the appendix for detailed instructions.

Kickstart

It is a best practice to use “Kickstart” to build an RHEL 6 KVM guest for the first time. Kickstart provides a semiautomated way to deploy Red Hat Enterprise Linux in a highly customizable way. After an RHEL 6 KVM guest is created and “made generic,” it is a best practice to repeat the deployment by using NetApp FlexClone.

FlexClone

FlexClone is highly efficient in many use cases and in the case of virtualization can be used to rapidly clone virtual machines (NFS-based), hypervisors (iSCSI, FC, and FCoE), and datastores. Offloading virtual machine cloning from the hypervisor to the NetApp controller provides an efficient means of scaling out the environment without taxing CPU cycles on the RHEL 6 KVM hypervisor. Additionally, FlexClone works with data deduplication and thin provisioning to make sure of a smaller storage footprint.

Note: There is a section dedicated to FlexClone.

Guest Timing Issues and Configuring Time for KVM Guests

All KVM (Windows and RHEL) guests need to be configured to use NTP to avoid issues that arise from time skew.

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 run on all RHEL KVM guests.

To configure Red Hat Enterprise Linux (RHEL) NTP, perform the following steps:

1. Log in to the guest and perform an initial time sync.

```
service ntpd stop
ntpdate -q <IP_address_of_NTP_server>
```

   Note: If the NTP service is not already running, the service ntpd stop command will fail. This is expected and acceptable.

2. Edit the /etc/ntp.conf file.

```
server <IP_address_of_NTP_server>
```

3. Enable the NTP service to start automatically on boot.

```
chkconfig ntpd on
```

4. Start the NTP service.

```
service ntpd start
```

5. To confirm that the NTP daemon (NTPD) is correctly receiving data from the NTP servers, use the NTP query tool.

```
ntpq -p
```

6. This lists the NTP server(s) from which the NTP client is currently getting time updates.

   Note: Repeat steps 1 through 5 on each host that must be configured.

7. Open port 123 (both UDP and TCP) on the IPtables firewall to allow NTP traffic.

Windows Server 2003 and XP Guests

1. Add the following line to the “boot.ini” file to enable the real-time clock.

```
/usepmtime
```

Windows Vista, Windows Server 2008, and Windows 7 Guests

2. 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
```

Review the chapter "KVM guest timing management" in the Virtualization Guide at http://docs.redhat.com for additional information regarding KVM guest time.

Security Considerations for Guest Operating Systems

See the earlier section on securing a Red Hat Enterprise Linux 6 KVM host for guidelines on basic security considerations for RHEL-based virtual machines.

The focus should be on firewall, mandatory access control, unnecessary services, and insecure services.

1. Leave IPTables enabled; configure necessary ports rather than disabling the firewall.
2. Leave SELinux enabled; configure additional MACs rather than disabling SELinux.
3. Limit the packages that are installed to only those that are needed.
4. Disable and/or remove unnecessary services.
5. Disable insecure services such as RSH, telnet, and FTP in favor of SSH, SCP, and SFTP.
6. Windows guests should be configured for firewall and antivirus software.

6 NetApp Storage Best Practices for RHEL 6 KVM

6.1 Aggregates 64-Bit Clustered Data ONTAP

Overview

NetApp Data ONTAP 8.x supports a new type of aggregate called a 64-bit aggregate, which supports much larger sizes than the 32-bit limit of 16TB. FlexVol volumes created in 64-bit aggregates also support much larger volumes ranging in size from 30TB to 100TB, based on the storage system. Refer to the following documents for more information:

- The system configuration guides on the NetApp Support (formerly NOW®) site (http://now.netapp.com/NOW/knowledge/docs/hardware/NetApp/syscfg) for maximum supported 8.x volume and aggregates sizes for specific hardware platforms
- TR-3786: A Thorough Introduction to 64-Bit Aggregates

Benefits

Using 64-bit aggregates results in the following benefits:

- **Better performance from more disks in the aggregate.** When the disks are the bottleneck in improving performance, using 64-bit aggregates with a higher spindle count can provide a performance boost.
- **Ability to add disks to an aggregate in fully populated RAID groups.** Using 64-bit aggregates provides maximum storage efficiency while also offering all the data protection benefits of RAID-DP®.
- **Ability to add more disk shelves into an aggregate.** Because of the larger size of 64-bit aggregates, storage administrators can manage a system with fewer aggregates, thus reducing the overhead of managing multiple aggregates.

- **All the advantages and features of Data ONTAP**. Space efficiency, thin provisioning, deduplication, FlexVol volumes, and many other features of Data ONTAP can be used on 64-bit aggregates with the same ease and simplicity as on 32-bit aggregates.
- **Bigger aggregate and volume sizes.** Use 64-bit aggregates to build a flexible storage solution with improved scalability and ease of management as your storage needs grow.
- **Easier data management.** Use 64-bit aggregates for applications that store and access huge amounts of data. The larger-size thresholds of FlexVol volumes in 64-bit aggregates provide the ability to store all the data in a single FlexVol volume, making it easier to manage the data.

A 64-bit aggregate containing the root volume is created during the Data ONTAP setup process. To create additional 64-bit aggregates, determine the aggregate name, the node on which to create it, and how many disks it will contain.

1. Execute the following command to create a new aggregate:

```
aggr create -aggregate new_aggr -nodes node_name -B 64 -diskcount num_of_disks
```

Note: Leave at least one disk (select the largest disk) in the configuration as a spare. A best practice is to have at least one spare for each disk type and size.

6.2 Vserver

Types of Vservers in Data ONTAP 8.1

Data ONTAP 8.1 Vservers include the following:

- Cluster admin Vserver
- Cluster Vserver
- Node Vserver

6.3 Cluster Admin Vserver

Some of the cluster admin Vserver features and limitations include:

- One per cluster
- Clusterwide scope and authority
- Does not own volumes or LUNs
- Cannot export or share file systems or provide access to LUNs

6.4 Cluster Vserver

Some of the cluster Vserver features and limitations include:

- One or more per cluster
- SAN or NAS data services require at least one
- Scope limited to the objects it owns
- Owns a set of clustered Data ONTAP volumes and LUNs and uses them to provide SAN or NAS data access
- Directly accessible by means of its own optional Vserver management LIF

Create a Vserver.

1. Run the Vserver setup wizard.

```
Welcome to the Vserver Setup Wizard, which will lead you through
the steps to create a virtual storage server that serves data to clients.

You can enter the following commands at any time:
```

```
"help" or "?" if you want to have a question clarified,
"back" if you want to change your answers to previous questions, and
"exit" if you want to quit the Vserver Setup Wizard. Any changes
you made before typing "exit" will be applied.

You can restart the Vserver Setup Wizard by typing "vserver setup". To accept a default
or omit a question, do not enter a value.

Step 1. Create a Vserver.
You can type "back", "exit", or "help" at any question.
```

2. Enter the Vserver name:

```
Enter the Vserver name:
```

3. Select the Vserver data protocols to configure. If you are uncertain, select all of them.

```
Choose the Vserver data protocols to be configured {nfs, cifs, fcp, iscsi}:
```

4. Select the Vserver client services to configure. If you are uncertain, select all of them.

```
Choose the Vserver client services to configure {ldap, nis, dns}:
```

5. Enter the Vserver's root volume aggregate:

```
Enter the Vserver's root volume aggregate {aggr01, aggr02} [aggr02]:
```

6. Enter the Vserver language setting. English is the default language [C].

```
Enter the Vserver language setting, or "help" to see all languages [C]:
```

7. Answer "No" to Do you want to create a data volume?

```
Vserver creation might take some time to finish....

Vserver test01 with language set to C created. The permitted protocols are
nfs, cifs, fcp, iscsi.

Step 2: Create a data volume
You can type "back", "exit", or "help" at any question.

Do you want to create a data volume? {yes, no} [yes]:
```

8. Enter "No" for all the remaining questions regarding service configuration. Individual services can be configured at a later time.

   Note: As an alternative to the individual configuration steps, create a Vserver using a one-line command, as specified in the following example:

```
vserver create -vserver name_of_vserver -rootvolume name_of_rootvol -aggregate name_of_aggr -ns-
switch file -language en_US -rootvolume-security-style unix -snapshot-policy none
```

6.5 Node Vserver

Some of the node Vserver features include:

- Managing the physical nodes in the cluster, if needed
- One per controller
- Identifying by <node_name> and D-blade UUID
- Owning one or more operating in 7-Mode volumes (including the root volume) and zero or more qtrees on its controller

6.6 FlexVol for Clustered Data ONTAP

FlexVol Limits and Maximums

Refer to NetApp product documentation associated with the specific version of Data ONTAP for a detailed explanation of product limits and scaling considerations that are consistent with various storage efficiency and other optional settings.

FlexVol Default Snap Reserve

Be sure to take into account the storage space that snap reserve can consume (20% of the volume by default when a FlexVol volume is created).

Thin Provisioning

Thin provisioning is covered in a dedicated section.

Create a FlexVol Volume in Clustered Data ONTAP

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 vol_name on aggregate aggr_name with a size of 10GB, run the following command:

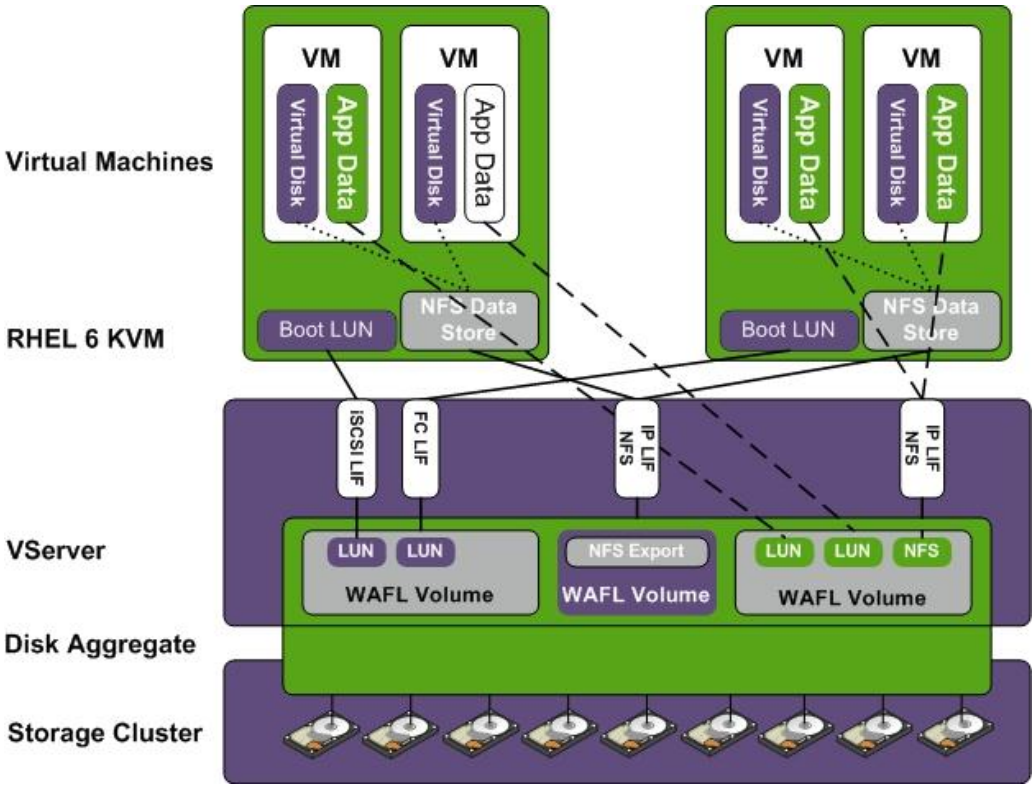
```
volume create -vserver vserver_name -volume vol_name -aggregate aggr_name -size 10g -state online
-type RW -policydefault -snapshot-policy none
```

6.7 LUN with Clustered Data ONTAP

Overview

Figure 8 shows how LUNs relate to volumes and Vservers and how they can be accessed through iSCSI or FC LIFs. This diagram also shows how a LUN serves as a VM or an application data container.

Figure 8) Diagram of LUNs mapping to hosts.



Clustered Data ONTAP LUNs and Root Volume

LUNs cannot be created on the root aggregate of the node. NetApp does not recommend creating them on the root volume of the Vserver.

Clustered Data ONTAP LUNs and Snapshot Copies

NetApp recommends grouping LUNs in a single volume with similar Snapshot requirements to facilitate the management of Snapshot copies.

Clustered Data ONTAP LUNs and Space Guarantees

NetApp recommends thin provisioning all LUNs, which means setting no space guarantees. If it is critical that writes to a LUN always complete, then space guarantees can be enabled.

Clustered Data ONTAP LUNs and Fractional Reserve

Fractional reserve controls the amount of space a volume reserves for overwrites to space-reserved LUNs when the volume has filled. NetApp recommends setting the fractional reserve to 0%.

6.8 Thin Provisioning NAS with Clustered Data ONTAP

Setting Up NAS Thin Provisioning

When thin provisioning is enabled, primary data and space for the associated Snapshot copy are allocated on demand. This variant achieves the best ratio of storage efficiency for provisioning applications from scratch. NetApp recommends that customers choose the thin-provisioning method to increase storage efficiency. Thin provisioning follows a 100% allocate-on-demand concept.

The thin-provisioning method has the following characteristics:

- Volumes are created without space guarantee.
- The size of the volume follows the formula X + Δ, where:
  - X is the size of the primary data (sum of all user files and directories within the volume)
  - Δ is the amount of space needed to hold Snapshot copy data
  - Sizing the volume defines a container with a virtual size for the consumers. NAS users are familiar with fixed-sized file shares and with the following considerations:
- Space used for Snapshot copies can grow unexpectedly. Administrators can use the autosize function to make space available when reaching a certain volume threshold or when the space reserved for user data becomes low.
- Space reserved for Snapshot copies is used to hide from the consumers (NAS clients) the capacity taken up by Snapshot copies.
- For volumes with deduplication enabled, volume autogrow is a mandatory option.
- Using the autodelete option is normally not recommended in NAS environments. Reserving a certain amount of space for Snapshot copies for file versioning or file restores is part of the SLAs defined for file services.

Table 4 lists thin provisioning volume options.

Table 4) Thin provisioning volume options.

Volume Option	Recommended Value	Notes
space-guarantee	none	This is the key setting for thin provisioning.
fractional-reserve	0	The default is 0%.
autosize	on	Set autosize to on. No artificial limited volume must be monitored. The autosize function allows the user data to grow beyond the guaranteed space limit.
space-mgmt-try-first	volume_grow	Increasing the size of the volume does not destroy any data or information. Therefore, the volume size can be increased or decreased as needed. For some configurations, automatic volume growth might not be desired.

Table 5 lists thin provisioning volume Snapshot options.

Table 5) Thin provisioning volume Snapshot options.

Volume Snapshot Option	Recommended Value	Notes
percent-snapshot-space	0	The value depends on the number of Snapshot copies and the change rate within the volume. Displaying only the committed usable space using SLA is the preferred way to provision NAS storage. However, in some situations, the Snapshot copy reserve area might be omitted.
autodelete	false	Deleting Snapshot copies is not recommended in most NAS environments so that data is not lost when SnapManager® products are being used. However, if running out of space is deemed more critical than losing Snapshot copies, then change the value to true to enable this option.

Thin Provisioning NFS with Clustered Data ONTAP

Run the following commands to modify the volume options and the volume Snapshot options. These commands set the default FlexVol volume to be thin provisioned.

```
vol modify -vserver vserver_name -volume vol_name -space-guarantee none -fractional-reserve  0 -
autosize true -max-autosize (2*SIZE)g -space-mgmt-try-first volume_grow -percent-snapshot-space 0
vol snap autodelete modify -vserver vserver_name -volume vol_name -enabled false
```

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.

Volume Size Considerations

Because physical allocation of data within a thin-provisioned volume is done on demand, theoretically, the volume size can be set to a very high value to easily store all application data and Snapshot copies. All other applications can benefit from the shared pool of unallocated storage because the unallocated space in the volume is not exclusively reserved for the volume itself.

NetApp recommends that customers size the volume to the expected size of its containing objects and use the autogrow option to let it grow on demand. The advantage of this method is that the commitment rate acts as a metric for data consolidation.

Note: The commitment rate reflects the amount of logical data consolidation. This metric is suitable for deciding when to leave space for organic growth.

In addition, the volume size limits should be taken into account when using deduplication because the maximum sizes depend on the storage controllers.

Requirement for Storage Monitoring

A high degree of data consolidation can be achieved by using thin provisioning. Because this effect depends on the usage characteristics of the corresponding applications, monitoring the aggregate is critical.

6.9 Deduplication with Clustered Data ONTAP

NetApp deduplication provides block-level deduplication within the entire flexible volume. Essentially, deduplication removes duplicate blocks, storing only unique blocks in the flexible volume, and creates a small amount of additional metadata in the process.

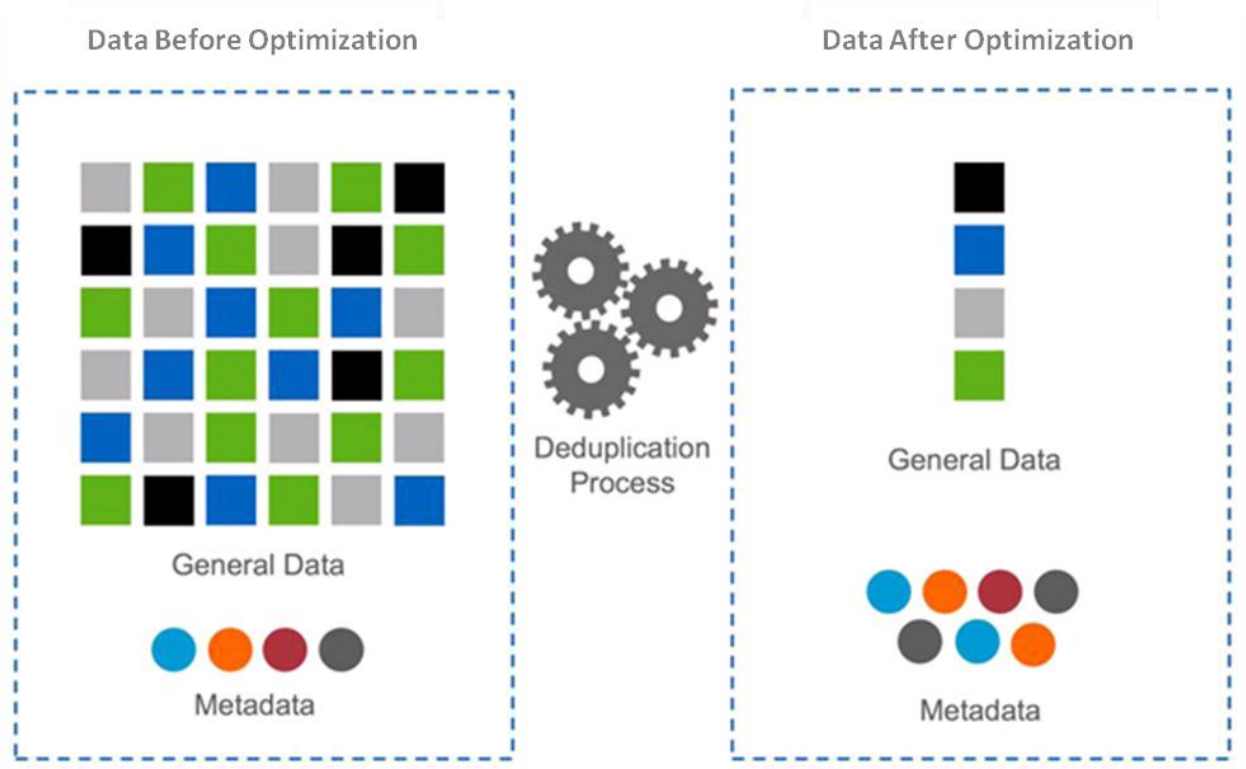
Overview

NetApp deduplication has the following characteristics:

- It works with a high degree of granularity at the 4kB block level.
- It operates on the active file system of the flexible volume. Any block referenced by a Snapshot copy is not made available until that Snapshot copy expires.
- It is a background process that can be configured to run automatically; be scheduled; or be run manually through the CLI, NetApp Systems Manager, or NetApp Provisioning Manager.
- It is application transparent and, therefore, can be used for the deduplication of data originating from any application that uses the NetApp system.

Figure 9 shows the NetApp deduplication process at highest level.

Figure 9) NetApp deduplication process at highest level.



Deduplication and Compression

Data ONTAP 8.1 and later provides deep integration of compression and deduplication. Deduplication must first be enabled on a volume before compression can be enabled; compression cannot be enabled without deduplication.

Deduplication Limitations

A maximum of eight deduplication processes can run concurrently on eight different volumes within the same NetApp storage system. If an attempt is made to run an additional deduplication process beyond the maximum number, the additional operations are placed in a pending queue, and they are automatically started when free processes become available.

Although deduplication can provide substantial storage savings in many environments, it is associated with a small amount of storage overhead. In Data ONTAP 8.1, the deduplication metadata for a volume continues to be located inside the aggregate; however, a copy of this metadata is stored in the volume. The guideline for the amount of additional space that should be left in the volume and aggregate for the deduplication metadata overhead is as follows:

- Volume deduplication overhead. For each volume with deduplication enabled, up to 4% of the logical amount of data written to that volume is required to store volume deduplication metadata.
- Aggregate deduplication overhead. For each aggregate that contains any volumes with deduplication enabled, up to 3% of the logical amount of data contained in all of those volumes with deduplication enabled is required to store the aggregate deduplication metadata.

Enable Deduplication in Clustered Data ONTAP

1. To enable deduplication on a volume, run the following volume efficiency command.

```
volume efficiency on –vserver vserver_name -volume vol_name
```

2. To start deduplication manually, run the following volume efficiency command.

```
volume efficiency start –vserver vserver_name -volume vol_name
```

6.10 NFSv3 with Clustered Data ONTAP

NFSv3 Overview

NFS environments manage security at a high level in the following ways:

- NFS exports data to a client system. NFS relies on the client to authenticate its own users. Given the complete local control a user might have over a client system, the NFS security model can be subverted relatively easily.
- NFS is derived from UNIX® and therefore implements a UNIX style set of file permissions that are limited to three categories of users (user, group, and other) and three types of file permissions (read, write, and execute).
- An NFS file server can apply various options (for example, restricting all access to read-only access) when exporting; an NFS client has similar options when mounting from an NFS file server.
- A NetApp system controller can export to specified NFS clients, netgroups (lists of clients), or even network subnets. Exporting by subnet is not always available on conventional NFS file servers from other vendors.

With UNIX NFS clients, file systems are mounted from the server to be logically indistinguishable from local file systems. This can happen without any user activity such as starting up the client, for example. The NFS server grants or denies the mount request based on the host name of the client as dictated by the restrictions specified. No password or other authentication is required by an NFS server.

User authentication is performed by the client, not the server. The client-authenticated user's UID must be passed across the network with all NFS requests sent to the server. Therefore, if a user can convince the NFS client of the user name identity associated with the UID `n`, then the NFS server will accept that any NFS requests issued by that client on that user's behalf are for UID `n`. One exception to this rule occurs when `n` corresponds to the superuser (`root`). In this case, the NFS server assigns unauthenticated status (the user `nobody`) to the NFS session that was established unless root access privileges have been assigned to that client.

Each time an export rule is loaded into memory, it undergoes a straightforward conversion from a string to an entry in a hash table. The basic string format is the following:

```
             Policy          Rule    Access   Client                RO
Vserver      Name            Index   Protocol Match                 Rule
------------ --------------- ------  -------- --------------------- ---------
vs2_nfs4     default         1       any      0.0.0.0/0             any
vs2_nfs4     nfs4_policy1    1       any      0.0.0.0/0             any
```

Enable NFSv3 in Clustered Data ONTAP

Run all commands to configure NFS on the Vserver.

1. Install the NFS license.

```
license add -license-code license_key
```

6.11 FC with Clustered Data ONTAP

FC Fabric Design Recommendations

NetApp requires the use of multipathing in an FC environment when using clustered Data ONTAP. NetApp also requires Asymmetric Logical Unit Access (ALUA) with clustered Data ONTAP so that multipath selection and cluster operations such as cluster failover and nondisruptive volume movement work correctly. ALUA is automatically enabled on the storage.

Two general topologies are available in an FC environment:

- **Single fabric.** All ports of the initiator and target connect to a single switch or a fabric of switches.
- **Multifabric**. Some ports of the initiator and/or target connect to separate fabrics for redundancy.

All these configurations include ALUA and are detailed in the Fibre Channel and iSCSI Configuration Guide. This guide includes diagrams and supported topologies for different NetApp platforms.

NetApp recommends using redundant components for any SAN solution to reduce or eliminate single points of failure. Therefore, use multiple HBAs, switches/fabrics, and storage clustering for FC.

Refer to the Fibre Channel and iSCSI Configuration Guide for the switch vendor maximum hop count.

Cascade, mesh, and core-edge fabrics are all industry-accepted and supported methods of connecting FC switches into a fabric.

Enable FC with Clustered Data ONTAP

1. License FCP.

```
system license add -license-code license_key
```

2. If needed, create the FC service on each Vserver. This command also starts the FC service and sets the FC alias to the name of the Vserver.

```
fcp create -vserver node_name
```

3. If needed, start the FC service on each Vserver.

```
fcp start -vserver node_name
```

4. Verify whether the FC ports are targets or initiators.

```
node run -node node_name fcadmin config
```

5. If needed, make an FC port into a target to allow connections into the node.

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

6. For example, to convert a port called, use the following syntax:

```
node run -node node_name fcadmin config -t target fc_target
```

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

NetApp Host Utilities

NetApp provides a SAN Host Utilities kit for every supported OS. This set of data collection applications and configuration scripts includes SCSI and path timeout values and path retry counts. Tools to improve the supportability of the host in a NetApp SAN environment are also included.

6.12 iSCSI with Clustered Data ONTAP

Overview

This solution enables the iSCSI protocol on a Vserver and sets the default policy to `deny`. NetApp requires the use of multipathing in an iSCSI environment when using clustered Data ONTAP. NetApp also requires ALUA with clustered Data ONTAP so that multipath selection and cluster operations such as cluster failover and nondisruptive volume movement work correctly. ALUA is automatically enabled on the storage.

All configurations include ALUA and are detailed in the "Data ONTAP® 8.1 SAN Configuration Guide For Cluster-Mode." This guide includes diagrams and supported topologies for different NetApp platforms.

Solution Notes

- This solution constitutes a basic enablement of iSCSI on a Vserver.
- The iSCSI alias defaults to the name of the Vserver. Having the same name for the iSCSI alias and for the Vserver makes it easier to identify the correct controller when selecting a target from among the iSCSI initiator interfaces.
- All deployment steps must be run on all Vservers that share data through iSCSI.
- NetApp recommends the use of password-protected CHAP entries for each iSCSI host that must authenticate with the storage system. Password-protected session authentication prevents iSCSI LUNs from being mounted through spoofing attacks over the network. Without password-protected session authentication, a storage environment with iSCSI enabled is not compliant with security regulations such as HIPAA, PCI DSS, or Sarbanes-Oxley.

Enable iSCSI with Clustered Data ONTAP

The following steps configure the iSCSI service on a Vserver. These steps do not include any host-specific configuration tasks.

1. From the cluster shell, license the iSCSI protocol.

```
system license add -license-code license_key
```

2. If needed, create the iSCSI service on each Vserver. This command also starts the iSCSI service and sets the iSCSI alias to the name of the Vserver.

```
iscsi create -vserver node_name
```

3. If needed, start the iSCSI service on each Vserver.

```
iscsi start -vserver node_name
```

6.13 LUN Creation with Clustered Data ONTAP

After the iSCSI and/or FCP licenses are installed, LUNs may be created within a volume on a Vserver.

1. To create a 2GB LUN for RHEL in volume vol_name called lun_name on Vserver vserver_name, execute the following command:

```
lun create –vserver vserver_name -volume vol_name -lun lun_name –size 2g –ostype linux
```

2. To provide access to LUN, an initiator group (igroup) must be created and then mapped to the LUN. Create the igroup by executing the following command:

```
igroup create -igroup igroup_name -protocol iscsi -ostype linux -initiator initiator01
```

The protocol option must be “fcp” or “iscsi,” depending on the requirement. The initiator names will either be a WWPN (FC or FCoE) or IQN (iSCSI initiator names).

3. Map the LUN to the igroup by executing the following command:

```
lun map -volume vol_cmode_nfs -lun testlun -igroup bob -lun-id 0
```

7 Storage Network Best Practices for RHEL 6 KVM

7.1 Storage Architecture Concepts

Production Ethernet Storage Networks

The goal of any storage network is to provide uninterrupted service to all nodes connecting to it. This section is focused primarily on how to establish a highly available Ethernet storage network.

Regardless of storage protocol, a storage network must address the following three goals:

- Be redundant across switches in a multiswitch environment
- Use as many available physical paths as possible
- Be scalable across multiple physical interfaces or ports

10GbE for Ethernet Storage

NetApp Data ONTAP, Red Hat Enterprise Linux, and RHEL 6 KVM all provide support for 10GbE. An advantage of 10GbE is the ability to reduce the number of network ports in the infrastructure, especially but not limited to blade servers. Additionally, 10GbE can handle several VLANs simultaneously. It is a NetApp best practice to use 10GbE, especially for storage.

7.2 IFGRP LACP with Clustered Data ONTAP

Overview

Dynamic multimode interface groups are based on Link Aggregation Control Protocol (LACP) as described by the IEEE 802.3ad (dynamic) standard. They are similar to static multimode interface groups that contain two or more active interfaces, share a single MAC address, and require configuration on both ends of the connection. In a dynamic multimode interface group, however, additional negotiation parameters are passed between the devices using an LACP PDU. This allows the two connected devices to dynamically add and remove links from the channel for reasons other than physical link failure. This is an important distinction because dynamic multimode interface groups can detect and compensate not only for a lost link, but also for a loss of data flow. This feature makes dynamic multimode interface groups compatible with HA environments.

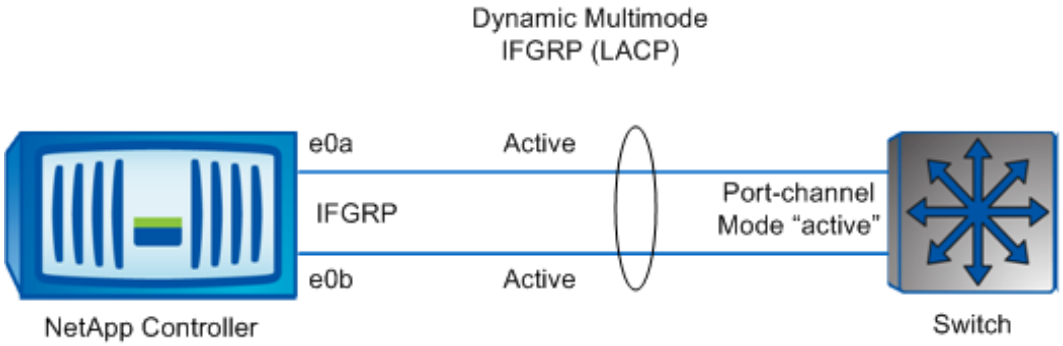
Dynamic multimode interface groups can continue operating even if all but one link has been lost. This allows for higher throughput than a single-mode interface group and still provides redundancy. Multiple methods of load balancing are available for outgoing packets from the NetApp FAS device, including:

- **IP address load balancing**. This method uses a fast-hashing algorithm on the source and destination IP addresses to equalize traffic on multimode interface groups.
- **MAC address load balancing**. This method uses a fast-hashing algorithm on the source and destination MAC addresses to equalize traffic on multimode interface groups.
- **Round robin**. This method is normally used for load balancing a single connection's traffic across multiple links to increase single-connection throughput.
- **Port (Data ONTAP 7.3.2 and later)**. This method uses a fast-hashing algorithm on the source and destination IP addresses along with the transport layer port number. Port-based load balancing adds a second hash for every unique source and destination port, which can result in more efficient load balancing of traffic over LACP links. NetApp recommends this distribution function in mixed workload environments.

Dynamic multimode interface groups must be connected to a switch that supports LACP.

Figure 10 shows the dynamic multimode interface group (LACP) configuration.

Figure 10) Dynamic multimode interface group (LACP).



Advantages

Dynamic multimode interface groups are compatible with an HA environment because they can detect not only the loss of link status but also a loss of data flow.

The advantages are:

- LACP allows for higher aggregate bandwidth based on the load-balancing algorithm chosen because all ports in the channel are active.
- The LACP PDUs that are used enable the dynamic multimode interface group to detect a loss of link on either side and to detect a loss of data flow. This avoids queue wedge or traffic black hole problems.
- Many newer switches support multichassis LACP, allowing for interface groups to be connected to multiple switches. This setup increases redundancy across the switch infrastructure.

Disadvantages

Dynamic multimode interface groups might not be compatible with some environments.

The disadvantages are:

- Older switches might not support the LACP standard.
- Older switches might not support multichassis LACP, which means that LACP interface groups would be limited to a single switch.

This type of interface group requires two or more Ethernet interfaces and a switch that supports LACP. Therefore, make sure that the switch is configured properly.

Run the following command on the command line. This example assumes that there are two network interfaces called `e0a` and `e0b`.

```
ifgrp create -node node_name -ifgrp name_of_vif -distr-func ip -mode multimode_lacp
network port ifgrp add-port -node name_of_node -ifgrp name_of_vif -port e0a
network port ifgrp add-port -node name_of_node -ifgrp name_of_vif -port e0b
```

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

Note: The interface group name must follow the standard naming convention of `x0x`.

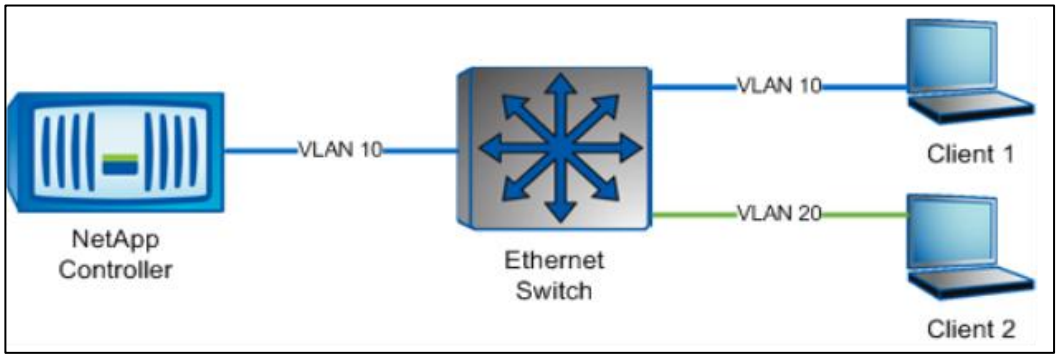
7.3 VLANs with Clustered Data ONTAP

Overview

VLANs operate at the data link layer (L2) of the OSI model; therefore, traffic is completely isolated between VLANs unless a bridge or a router (L3) is used to connect the networks. Although exceptions exist, a one-to-one relationship between an IP subnet and a VLAN helps simplify the network and facilitate management.

Figure 11 shows an example of VLAN connectivity.

Figure 11) Example of VLAN connectivity.



Implementing VLANs provides the following advantages within the network.

- **Higher utilization of switch infrastructure.** Allowing separate logical devices to live on the same physical switch eliminates the need for a completely separate hub or switch for each network. Higher density and more efficient switches can be used to aggregate devices.
- **Lower management costs.** Moving physical devices or cables is no longer a requirement. An administrator can logically assign devices from the management console of the switch to different networks if required.
- **Security and stability of the network.** An L3 device is required to communicate between VLANs; therefore, L3 access lists can be applied to prevent communication between certain networks. Broadcast storms and the effects of unicast flooding become isolated to a single VLAN. A malicious user with a packet capture tool will not be able to intercept traffic that is not destined for that user's host.

NetApp recommends that VLANs be deployed to separate data networks from storage networks.

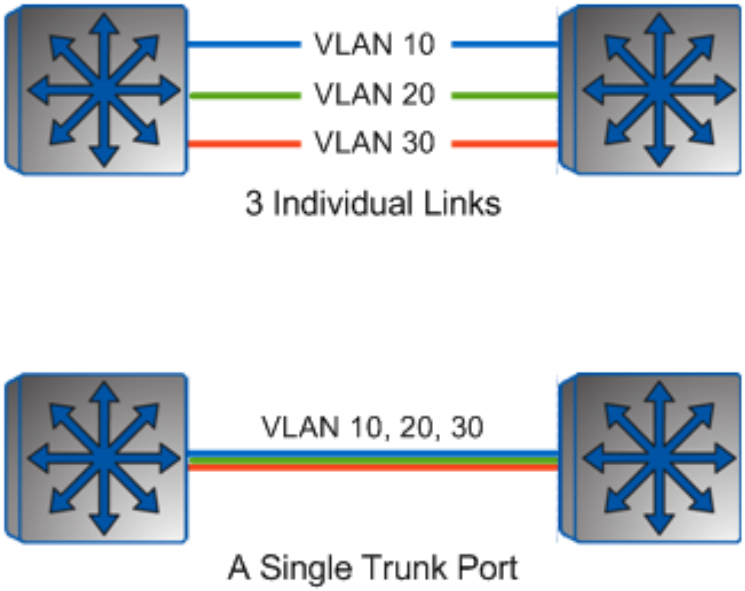
VLAN Trunking

Organizations typically require the use of multiple switches for redundancy and port density. Often, a logical VLAN might be required across multiple switches. A standard access port configuration allows only a single VLAN to be defined on a port. However, multiple ports, each assigned to a single VLAN, are required to pass VLANs across switches. This method does not scale well and is highly inefficient. The IEEE 802.1q standard provides a solution to this problem with a feature called VLAN trunking.

VLAN trunking allows a single link to carry multiple VLANs by tagging each packet with a 4-byte tag. This tag defines to which VLAN each packet is assigned as it travels throughout the network between VLAN-aware devices. Common VLAN-aware devices are network switches, routers, certain servers, and NetApp storage controllers. When the frame reaches an endpoint or access port, the VLAN tag is removed, and the original frame is then sent to the end device. The tag is simply a forwarding instruction to make sure that the frame is delivered to the proper broadcast domain or VLAN.

Figure 12 illustrates VLAN trunking.

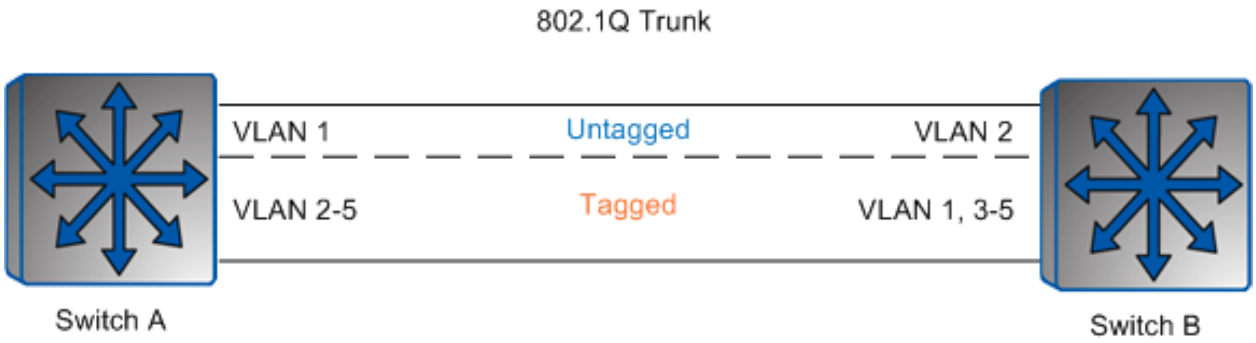
Figure 12) VLAN trunking.



Native VLAN is an important feature used within 802.1q trunking. Frames assigned to the VLAN, which are configured as the native VLAN, are sent untagged across the trunk link. All other VLANs that are configured in the trunk are tagged with their respective VLAN IDs. For this reason, make sure to configure the native VLAN the same on both ends of the connection. Improper configuration of the native VLAN can result in limited or no connectivity for the administratively defined native VLANs.

Figure 13 shows an example of a faulty configuration.

Figure 13) Example of a VLAN faulty configuration.

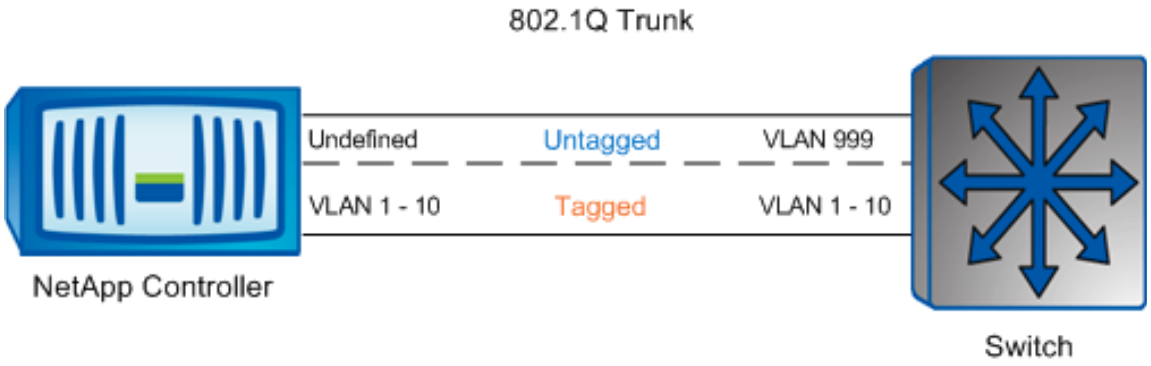


As mentioned earlier, NetApp storage controllers also provide the ability to configure multiple VLANs and VLAN trunking. This allows for greater flexibility and configuration options within the controller itself. For example, a NetApp controller with a 10GB interface might need to serve multiple functions such as iSCSI boot traffic and standard NAS traffic. The iSCSI boot network might require additional security and control, but because of the relatively low traffic requirement an administrator might not want to dedicate an entire 10GB link to this function. When VLAN trunking is implemented on the NetApp controller and switch, the single 10GB link can be shared between the two functions while still maintaining isolation between VLANs. When configuring VLANs on a NetApp controller, be aware that the Data ONTAP operating system does not currently use the native VLAN feature. For this reason, the native VLAN on the

switch port connected to a NetApp controller must be assigned to an unused VLAN to help forward data correctly.

Figure 14 shows the native VLAN NetApp configuration.

Figure 14) Native VLAN NetApp configuration.



7.4 Jumbo Frames with Clustered Data ONTAP

Overview

By default, a standard Ethernet frame has a 1,500-byte payload. The Ethernet header and the CRC checksum data add 18 bytes to the Ethernet frame for a total Ethernet frame size of 1,518 bytes or 1,522 bytes with a VLAN tag (IEEE 802.3ac). Because the header and the CRC checksum create a fixed amount of overhead per packet, efficiencies can be gained by sending a larger payload. Jumbo frames can be created by changing the MTU from the standard 1,500 bytes up to 9,000 bytes. Larger frames increase the network throughput by reducing the number of packets processed for the same amount of data.

Carefully implement jumbo frames in an Ethernet storage network. An incorrect design or configuration can result in poor performance and even limited or no connectivity at all. When configuring a NetApp FAS storage controller for jumbo frames, make sure that the following elements are properly configured:

- NetApp storage device's network ports and any associated IFGRPs or VLAN network ports
- Individual ports and port-channel interface on the external Ethernet switch, if applicable
- VLAN and ports on all switches and Layer 3 routers between the FAS device and the clients

Ports with a standard MTU size and ports with a jumbo MTU size should never be mixed on the same VLAN. For example, consider a host and a storage controller that are configured on the same VLAN, where the FAS controller is configured for jumbo frames and the host is not. The host can communicate with the FAS device using the standard 1,500-byte frames, but the reply from the FAS device will be in 9,000-byte frames. Because the two machines are located on the same VLAN, a device does not fragment the FAS device frames into the standard 1,500-byte size for consumption by the host.

To allow the NetApp storage controller to support both standard and jumbo frame requests from hosts, one option is to place a router in between the FAS device and the hosts because routers are able to fragment jumbo frames into smaller 1,500-byte increments. Devices that can use jumbo frames are placed onto a separate VLAN that is configured to directly pass jumbo frames to the NetApp storage controller. However, hosts that can only accept standard frames are placed onto a VLAN whose traffic is passed through a router for fragmentation. This configuration allows all hosts to properly communicate with the NetApp storage controller.

Another method is to directly connect VLANs for standard frame traffic and for jumbo frame traffic to separate ports on the NetApp storage controller. This method has the advantage of allowing traffic (with

the DF bit value set to true) to always reach its destination. The following scenarios make use of dedicated VLANs:

- **Local management VLAN (MTU 1,500)**. Supports SNMP, Operations Manager, SSH, RLM, and so on. Storage traffic never runs across this network.
- **Storage traffic (MTU 9,000)**. Isolated, nonrouted VLAN for NFS, CIFS, or iSCSI data.
- **Replication network (MTU 9,000)**. Isolated, nonrouted VLAN for high-speed storage replication such as SnapMirror and SnapVault® data. Separating this traffic allows more granular monitoring and the ability to support different WAN MTU sizes depending on the links used.
- **Intersite replication (MTU 1,500 or lower)** Useful for off-site backups where required WAN connections have different MTU values.

Jumbo Frame Recommendations

Using jumbo frames in an Ethernet storage network can significantly increase performance. Complete these steps to improve performance:

1. Configure jumbo frames throughout the network, from the Ethernet storage controller to the host.
2. Segment traffic with jumbo frames onto a different VLAN to achieve optimal network interface performance.
3. To configure a clustered Data ONTAP network port to use jumbo frames (which usually have an MTU of 9,000 bytes), run the following command from the clustershell:

```
network port modify –node node_name -port <network_port> -mtu 9000
WARNING: Changing the network port settings will cause a serveral second interruption in carrier.
Do you want to continue? {y|n}: y
```

The network port identified by `-port` can be a physical network port, a VLAN network port, or an IFGRP.

7.5 Firewall for Clustered Data ONTAP

Each LIF type has an attached default role and firewall policy.

Table 6 lists these default firewall policies.

Table 6) Default firewall policies.

Firewall Policy	DNS	HTTP	HTTPS	NDMP	NTP	SNMP	SSH	Telnet
Cluster	Allow	Allow	Allow	Allow	Allow	Allow	Allow	Allow
Mgmt	Allow	Allow	Allow	Allow	Allow	Allow	Allow	Block
Data	Allow	Block	Block	Allow	Block	Block	Block	Block
Intercluster	Block	Block	Block	Allow	Block	Block	Block	Block

7.6 Failover Groups for NAS with Clustered Data ONTAP

A failover group is a list of network ports available for use if a network port failure occurs. A network interface or a LIF can subscribe to a failover group and be automatically configured with a list of failover rules for each physical port in the failover group. As ports are added to or removed from the failover group, each LIF subscribed to that failover group is automatically configured to have a set of failover rules consistent with the ports present in the group.

Run the following commands in a failover pair to enable storage failover.

1. Enter the cluster license on both nodes.

```
node run first_node_name license add license_key
```

```
node run second_node_name license add license_key
```

2. Enable failover on one of the two nodes.

```
storage failover modify -node first_node_name -enabled true
```

Note: Enabling failover on one node enables it for both nodes.

3. Enable HA mode for two-node clusters only.

Note: Do not run this command for clusters with more than two nodes because it will cause problems with failover.

```
cluster ha modify -configured true
```

4. Select Yes if a prompt displays an option to enable SFO.

7.7 FCP LIF with Clustered Data ONTAP

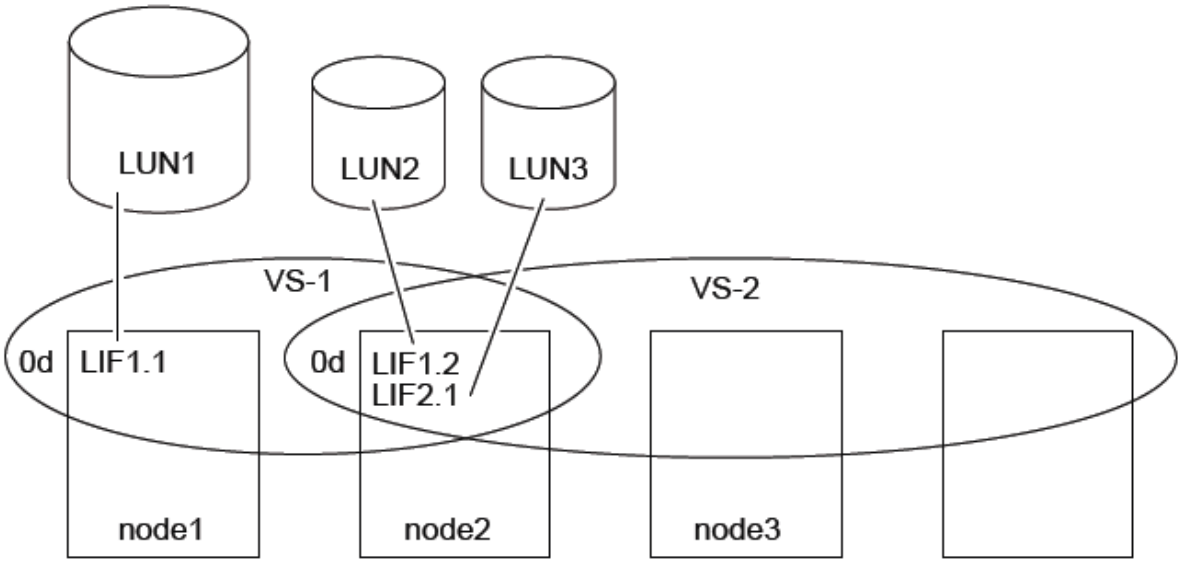
Overview

An FC LIF is a logical interface that belongs to a single Vserver and is assigned to a physical port. An FC LIF has its own unique WWPN and has the WWNN of the Vserver. The FC LIF permits access to LUNs belonging to the Vserver.

Multiple LIFs, whether from the same or different Vservers, can be assigned to a single physical port. In general, FC LIFs use a common ALUA LUN presentation model and rely on MPIO on hosts. For more information, refer to the Data ONTAP 8.1 Cluster-Mode Block Access Management Guide for iSCSI and FC.

Figure 15 shows an example of FC and iSCSI LIFs in a clustered Data ONTAP environment.

Figure 15) Diagram of FC and iSCSI LIFs in clustered Data ONTAP.



Limits

Table 7 lists the limits for the number of FC LIFs per port and for the FC ports per node.

Table 7) FC LIF limits.

Type	FC LIFs per Port	FC Ports per Node
FC LIF	8	32

Create an FCP LIF

Create FC LIFs named lif_name_1 and lif_name_2. In this example, the cluster consists of two nodes, node_name_1 and node_name_2, and two physical ports, port_name_1 and port_name_2.

```
network interface create -vserver vserver_name -lif lif_name_1 -role data -data-protocol fcp -
home-node node_name_1 -home-port port_name_1
network interface create -vserver vserver_name -lif lif_name_2 -role data -data-protocol fcp -
home-node node_name_2 -home-port port_name_2
```

7.8 iSCSI LIF with Clustered Data ONTAP

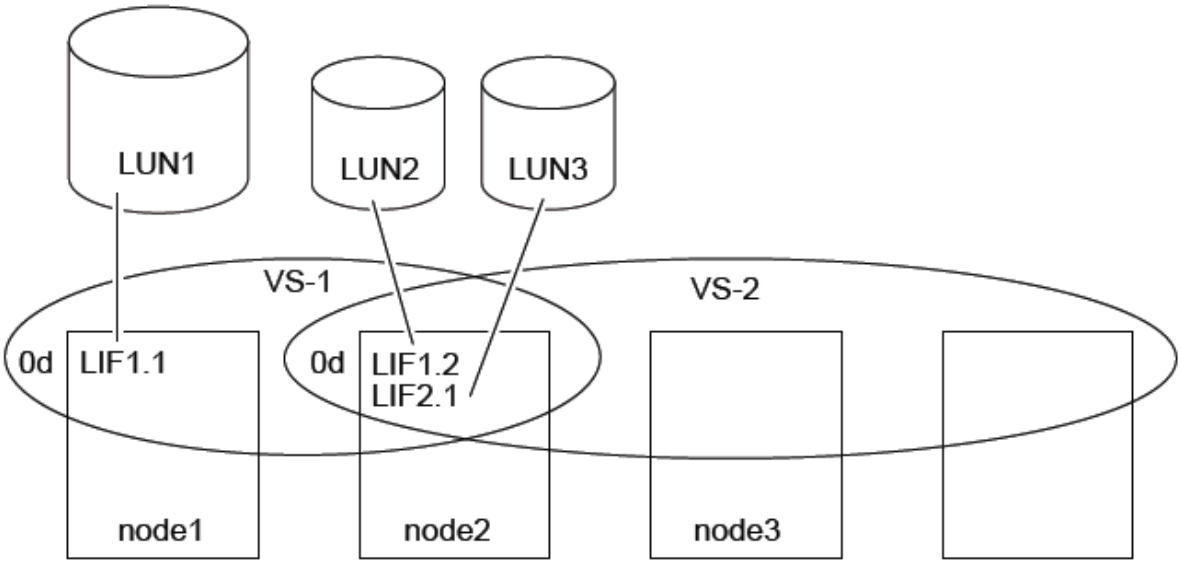
Overview

An iSCSI LIF is an IP-based logical interface that belongs to a single Vserver and is assigned to a physical port. The iSCSI LIF permits access to LUNs belonging to the Vserver.

Multiple LIFs, whether from the same or different Vservers, can be assigned to a single physical port. For more information, refer to the Data ONTAP 8.1 Cluster-Mode Block Access Management Guide for iSCSI and FC.

Figure 16 shows an example of FC and iSCSI LIFs in a clustered Data ONTAP environment.

Figure 16) Diagram of FC and iSCSI LIFs in clustered Data ONTAP.



Limits

Table 8 lists the limits for the number of IP LIFs per port and for the Ethernet ports per node.

Table 8) IP LIF limits.

Type	IP LIFs per Port	Ethernet Ports per Node
iSCSI LIF	8	24

1. Create an iSCSI LIF.
2. Create an iSCSI LIF named lif_name in the vserver_name Vserver, and assign the ip_address IP address.

   Note: The home node is node_name, and the physical port is port_name.

```
network interface create -vserver vserver_name -lif lif_name -role data -data-protocol iscsi -
home-node node_name -home-port port_name -address ip_address -netmask netmask_address
```

7.9 Intercluster LIF with Clustered Data ONTAP

Overview

An intercluster LIF is a logical interface that belongs to a single Vserver and is assigned to a physical port. The intercluster LIF permits access to transferring data between Vservers on different clusters. Intercluster LIFs can be configured on data ports or intercluster ports. An intercluster LIF must be created on each node in the cluster before a cluster-peering relationship can be established.

These LIFs can fail over to data or intercluster ports on the same node, but they cannot be migrated or failed over to another node in the cluster.

Limits

Table 9 lists the limits on the minimum number of intercluster LIFs per node if cluster peering is enabled and on the maximum total number of LIFs per node.

Table 9) Intercluster LIF limits.

Type	Minimum Intercluster LIFs per Node If Cluster Peering Is Enabled	Maximum Total LIFs per Node
Intercluster LIF	1	262

Create an Intercluster LIF

To create intercluster LIFs, perform the following steps:

1. Create intercluster LIFs either on data ports or on intercluster ports. Optionally, change one or more ports to the intercluster role to support the intercluster LIF.

```
network port modify -node node_name -port port_name -role intercluster
```

2. Create an intercluster LIF.

```
network interface create -vserver vserver_name -lif lif_name -role intercluster -home-node
node_name -home-port port_name -address ip_address -netmask netmask_address
```

3. Repeat step 2 for any additional LIFs.

8 Storage Network Services and Access

8.1 DNS with Clustered Data ONTAP

Domain Name Service

NetApp recommends that DNS be configured and enabled on all Vservers in the cluster.

DNS is a service used to convert host names to IP addresses. Host names contain alphabetic or numeric characters and are not case sensitive. Host names can also contain the hyphen, but other special characters are not permitted. A fully qualified domain name (FQDN) consists of the host name plus the domain name in the format host-name.domain.com.

The DNS service sends queries to the Domain Name System that stores the information for specific domains and the hosts within those domains. This process enables routing by using the host name instead of the IP address and is particularly useful when the IP address of various hosts must be changed. Applications that rely on communications to those hosts can continue to access the devices by using the host names without having to modify IP configurations at the application level.

Configure DNS

1. To configure DNS, run the following command for the Vserver:

```
vserver services dns create -vserver vserver_name -domains domain_name -name-servers dns_server -
state enabled
```

Note: To modify an existing entry, replace the word `create` with `modify` in the command.

8.2 NTP with Clustered Data ONTAP

NTP is a protocol used to synchronize the time of any network-attached device to a network time source. In this case, the time between the nodes and the chosen time server will be synchronized. Time synchronization allows logs and events to be correctly correlated with other network devices. Time synchronization is also required to connect to an Active Directory® server.

Configure NTP

1. Configure NTP for each node in the cluster.

```
system services ntp server create -node node_name -server ntp_server_ip_address
```

2. Enable NTP for the cluster.

```
system services ntp config modify -enabled true
```

8.3 SNMP with Clustered Data ONTAP

Overview

SNMP is used as a general management protocol for both polling and asynchronous notifications (traps). SNMP is a critical part of the interface to Operations Manager and must be configured for polling and traps if Operations Manager is used. Other upstream applications might also use SNMP for performance and fault management.

SNMP is a widely used network monitoring and control protocol. Data is passed from SNMP agents, which are hardware and/or software processes reporting activity in each network device (for example, hubs, routers, and bridges), to the workstation console used to oversee the network. The agents return information contained in a management information block (MIB), which is a data structure that defines

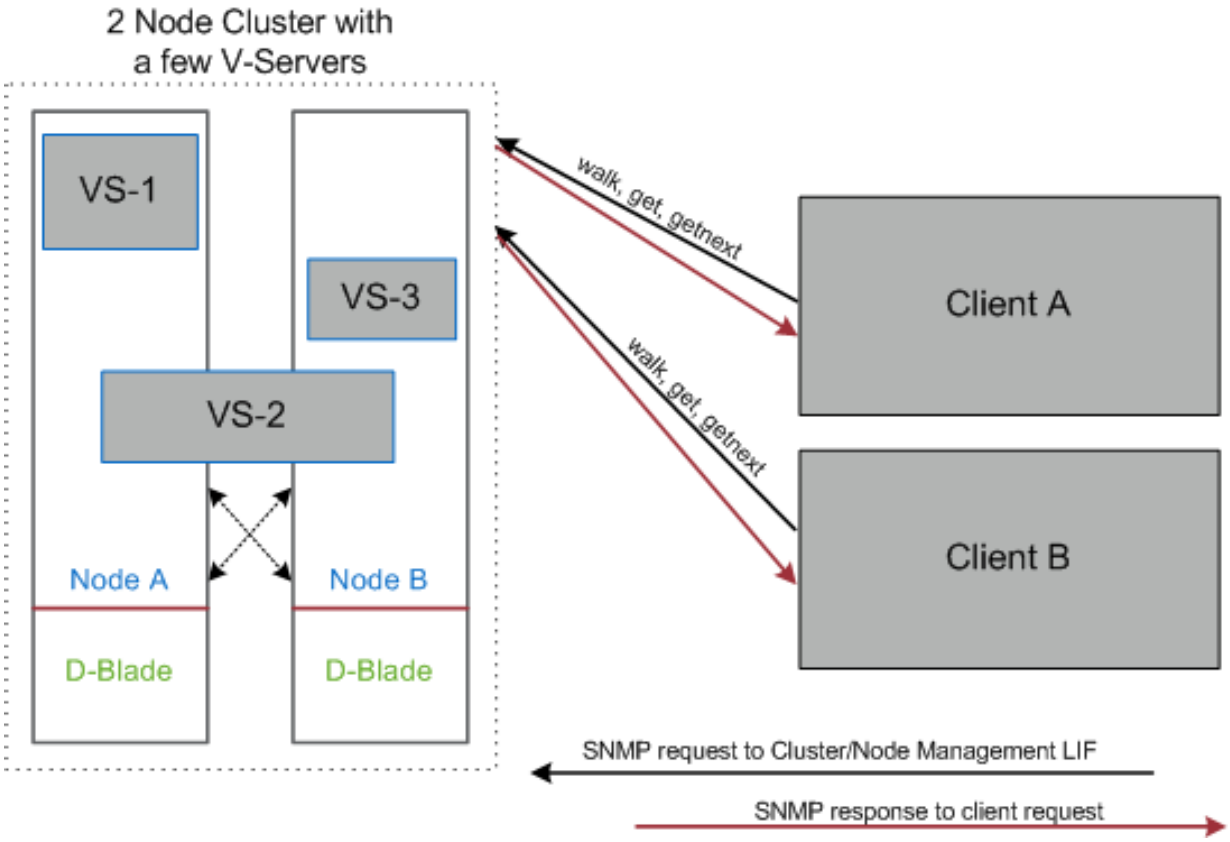
what is obtainable from the device and what can be controlled (for example, what can be turned off and on). SNMP originated in the UNIX community and has become widely used on all major platforms.

MIBs describe the structure of the management data of a device subsystem; they use a hierarchical namespace containing OIDs. Each OID identifies a variable that SNMP can read or set.

Note: NetApp does not support SNMP-set operations. Also, SNMP support is only available clusterwide and is not available on a per-Vserver basis. SNMP support will be available on a per-Vserver basis in releases after Version 8.1.

Figure 17 shows an SNMP diagram in clustered Data ONTAP.

Figure 17) SNMP diagram in clustered Data ONTAP.



SNMP Polling

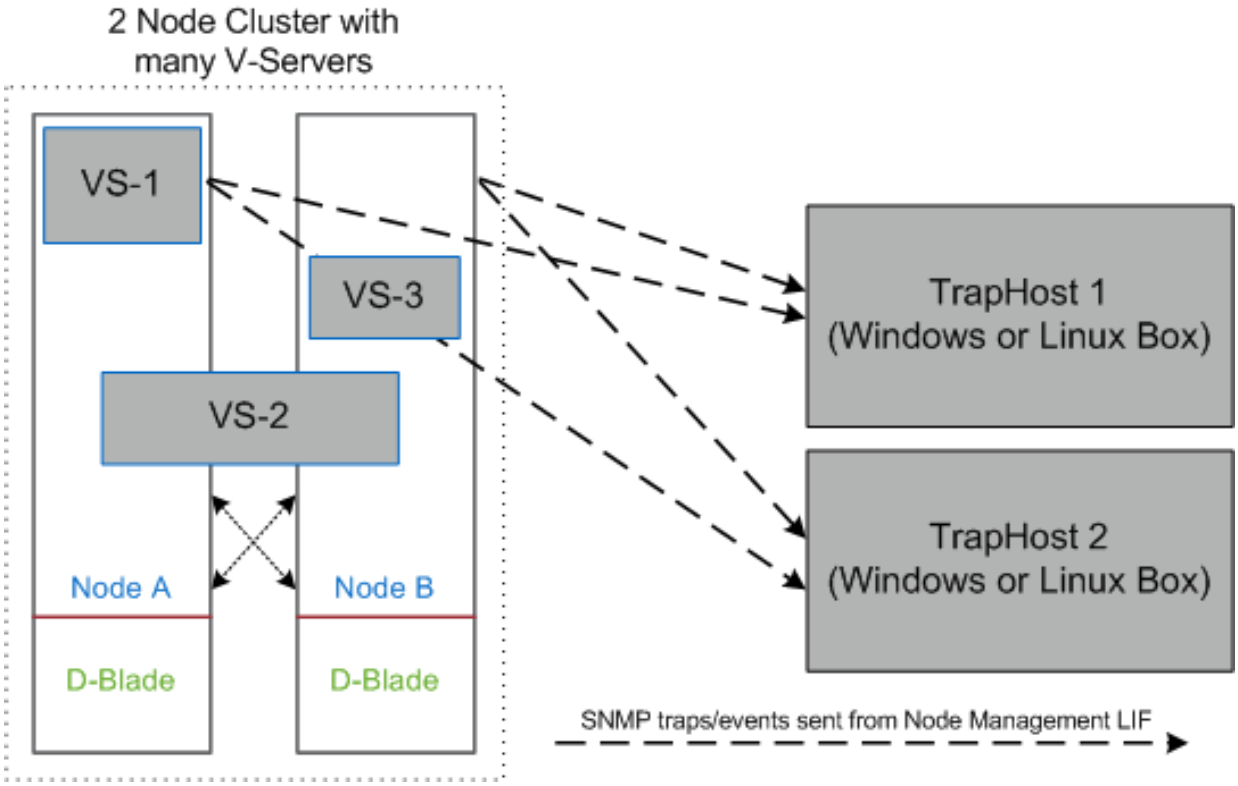
SNMP is used for polling. Data ONTAP supports two different versions of SNMP, Version 1 and Version 3. NetApp recommends using SNMPv3 because it has a stronger authentication mechanism. Earlier versions of SNMP are based on plain text authentication methods, which can be intercepted.

SNMP Traps

SNMP traps are configured to send to Operations Manager and other fault management stations. Data ONTAP sends SNMP traps as SNMPv1 traps.

Figure 18 shows SNMP traps in clustered Data ONTAP.

Figure 18) SNMP traps in clustered Data ONTAP.



8.4 AutoSupport HTTPS with Clustered Data ONTAP

AutoSupport™ is used to send configuration summary information and critical event information to NetApp Support. The transport mechanism is over HTTPS. Outgoing secure Web connections must be allowed from the storage controllers.

AutoSupport reports can be viewed in the My AutoSupport section of the NetApp Support (formerly NOW) site.

Configure AutoSupport HTTPS

AutoSupport sends support summary information to NetApp through HTTPS.

Execute the following commands to configure AutoSupport:

```
system node autosupport modify -node * -state enable -transport https -support enable -noteto
storage_admin_email
```

8.5 User Access for Clustered Data ONTAP

Overview

Data ONTAP provides several methods for specifying how a user can access the storage system. Use the `-application` parameter of the security login command to specify a method. The supported access methods include:

- System console (console)
- HTTP(S) (http)

- Data ONTAP API (ontapi)
- SNMP (snmp)
- SP or RLM (service-processor)
- SSH (ssh)
- Telnet (telnet)

  Note: By default, Telnet is hidden and disabled.

If a firewall is enabled, the access method must be added in the firewall policy to allow requests to go through the firewall. For more information, see the system services firewall policy main pages.

Note: Vserver user accounts cannot use console, snmp, service-processor, or telnet as access methods.

Configure User Access for Clustered Data ONTAP

There are two default administrative accounts: `admin` and `diag`. The `admin` account serves in the role of administrator and is allowed access using all applications. The best practice is to create a new account and then delete or lock this default admin account.

1. Create a login method for a new administrator from clustershell.

```
security login create –user name new_account -authmethod password -role admin -application ssh
security login create –user name new_account -authmethod password -role admin -application http
security login create –user name new_account -authmethod password -role admin -application
console
security login create –user name new_account -authmethod password -role admin -application ontapi
security login create –user name new_account -authmethod password -role admin -application
service-processor
```

2. Lock the default admin account.

```
security login lock –user name admin
```

Clustered Data ONTAP Authentication Methods for User Accounts

Data ONTAP provides several methods to specify how a user account is authenticated. Use the `-authmethod` parameter of the security login command to specify how a user account is authenticated. The supported authentication methods include:

- SNMP community strings (community)
- Windows Active Directory authentication (domain)
- LDAP or NIS authentication (nsswitch)
- User password (password)
- SSH public key authentication (publickey)
- SNMP user-based security model (usm)

For Windows Active Directory authentication, a CIFS server must be created for the Vserver. Windows domain users must be mapped to Vserver access control roles by using the security login create command with the `-authmethod` parameter set to `domain`. To use LDAP or NIS authentication, Vserver users must be mapped to Vserver access control roles using the security login create command and the `-authmethod` parameter set to `nsswitch`.

Cluster Administrator Compared to Vserver Administrator

Cluster administrators administer the entire cluster. Vserver administrators administer only their own data Vservers. Cluster administrators can administer the entire cluster and its resources. They can also set up data Vservers and delegate Vserver administration to Vserver administrators. The specific capabilities

that cluster administrators have depend on their access control roles. By default, a cluster administrator with the admin account name or role name has all of the capabilities for managing the cluster and Vservers.

Vserver administrators, in contrast, can administer only their own data Vservers' storage and network resources, such as volumes, protocols, LIFs, and services. The specific capabilities that Vserver administrators have to depend on the access control roles that are assigned by cluster administrators. For more information about Vserver administrator capabilities, refer to the Data ONTAP 8.1 Cluster-Mode Vserver Administrator Capabilities Overview Guide.

Default Cluster Context User Account Roles

The predefined roles for the cluster context are admin, readonly, and none, as shown in Table 10.

Table 10) Cluster context default roles and capabilities.

Role	Access Level to Command Directories	Capabilities
Admin	All	All
Readonly	Readonly	Readonly
None	None	None

Default Vserver Context User Account Roles

A Vserver can have its own user and administration authentication domain. The management of a Vserver to a Vserver administrator can be delegated.

Table 11 lists the four predefined roles for Vserver administrator.

Table 11) Vserver context predefined roles and capabilities.

Role	Default Capabilities
vsadmin	• Manage own user account, local password, and public key • Manage volumes, quotas, qtrees, Snapshot copies, FlexCache files, and files • Manage LUNs • Configure protocols • Configure services • Monitor jobs • Monitor network connections and network interface • Monitor the health of a Vserver
vsadmin-volume	• Manage volumes, quotas, qtrees, Snapshot copies, FlexCache files, and files • Manage LUNs • Configure protocols • Configure services • Monitor network interface • Monitor the health of a Vserver
vsadmin-protocol	• Configure protocols • Configure services • Manage LUNs • Monitor network interface • Monitor the health of a Vserver

Role	Default Capabilities
vsadmin-readonly	• Monitor the health of a Vserver • Monitor network interface • View volumes and LUNs • View services and protocols

Local Account Attributes

The modifiable account password attributes for clustered Data ONTAP are listed in Table 12.

Table 12) Account password attributes.

Attributes	Range	Default	Recommended
Length	3–64	8	Minimum 8
Alphanumeric enforcement	Enabled/disabled	Disabled	Enabled
Password history	6	6	
Minimum age	0	0	

Default Administrative Accounts

There are two default administrative accounts: admin and diag.

Administrative Account

The administrative account admin serves in the role of administrator and has the ability to access all applications. To enable this capability, first create another account (security login create) and configure the admin role for each application that the new admin account has permission to use.

Note: Do not use the off-box authentication method for the primary admin account, so that a networking problem does not lock the new admin account out of all administrative access.

After the new admin account has been tested, either delete (security login delete) or lock (security login lock) the account.

Note: The admin console entry cannot be deleted or locked if it is the only account with permissions to that application.

Diagnostic Account

The diagnostic account diag is provided with the storage system; it can be used to perform troubleshooting tasks in the systemshell. The diag account and the systemshell are intended only for low-level diagnostic purposes and should be used only with guidance from technical support. The diag account is the only account that can be used to access the systemshell, through the advanced command `system node systemshell`. Before accessing the systemshell, set the diag account password using the `security login password` command. Neither the diag account nor the systemshell is intended for general administrative purposes.

Network Security

Ports are either physical ports (NICs) or virtualized ports, such as interface groups or VLANs. A LIF communicates over the network through the port to which it is currently bound. A LIF is essentially an IP address with the following associated characteristics:

- Role
- Home node
- Home port
- Routing group
- Firewall policy
- Failover policy

Roles for Network Ports

Network ports can have roles that define their purpose and their default behavior. Port roles limit the types of LIFs that can be bound to a port. Network ports can have four roles:

- Node management
- Cluster
- Data
- Intercluster

Note: These four roles can be modified to obtain an optimal configuration.

Node Management Ports

Node management ports are used by administrators to connect to and manage a node. VLANs and interface groups can be created on node management ports. Some platforms have a dedicated management port (`e0M`). The role of these ports cannot be changed, and these ports cannot be used for data traffic. The management policy is applied by default; it allows DNS, HTTP, HTTPS, NDMP, NTP, SNMP, and SSH traffic. It blocks all telnet traffic.

Cluster Ports

Cluster ports are used for intracluster traffic only. By default, each node has two cluster ports. Cluster ports should reside on 10GbE ports and be enabled for jumbo frames. VLANs or IFGRPs cannot be created on cluster ports. The cluster policy is applied by default; it allows DNS, HTTP, HTTPS, NDMP, NTP, SNMP, SSH, and telnet traffic. It does not block any traffic.

Data Ports

Data ports are used for data traffic. These ports are accessed by NFS, CIFS, FC, and iSCSI clients for data requests. By default, each node has a minimum of one data port. VLANs and IFGRPs can be created on data ports. They possess the data role by default; however, their port role cannot be modified. The data policy is applied by default; it allows DNS and NDMP traffic. It blocks HTTP, HTTPS, NTP, SNMP, SSH, and telnet traffic.

Intercluster Ports

Intercluster ports are used for cross-cluster communication. An intercluster port should be routable to another intercluster port or data port of another cluster. The intercluster policy is applied by default; it allows NDMP traffic only. It blocks DNS, HTTP, HTTPS, NTP, SNMP, SSH, and telnet traffic.

SNMP

Enabling SNMP provides a mechanism to monitor a cluster to avoid issues before they occur and to respond to issues when they occur.

SNMP management includes performing the following tasks:

1. Enabling SNMP.

2. Configuring SNMP users.
3. Configuring SNMP trap hosts for specific events.

If SNMP is enabled in Data ONTAP, SNMP managers can query the storage system's SNMP agent for information. The SNMP agent then gathers information and forwards it to the SNMP managers. The SNMP agent also generates trap notifications whenever specific events occur. The SNMP agent on the storage system has read-only privileges that cannot be used for any set operations or for taking a corrective action in response to a trap. SNMP is enabled clusterwide in clustered Data ONTAP.

For diagnostic and other network management services, Data ONTAP provides an SNMP agent compatible with SNMP versions v1, v2c, and v3. SNMPv3 offers advanced security by using passphrases and encryption. When compared with SNMPv1 and SNMPv2c, SNMPv3 is a secure protocol. The SNMPv3 user must be configured to run the SNMP utilities from the SNMP manager.

Use the `security login create` command to create an SNMPv3 user.

Security level options include:

- Authentication, no privacy
- Authentication, privacy
- No authentication, no privacy

When prompted, provide the following information:

- **Engine ID** (default value is local EngineID)
- **Authentication protocol** (none, md5, sha)
- **Authentication password** (minimum 8 characters)
- **Privacy protocol** (none, des)
- **Privacy protocol password** (passphrase for encryption)

8.6 HTTPS Access with Clustered Data ONTAP

Access to the storage controller occurs by using a secure protocol. In this case, the protocol is HTTPS (HTTP over SSL). Other nonsecure access methods such as Telnet and HTTP must be disabled.

Configure HTTPS Access in Clustered Data ONTAP

Secure access to the storage controller must be configured.

1. Increase the privilege level to access the certificate commands.

```
set -privilege advanced
```

2. Generally, a self-signed certificate is already in place. Check it with the following command:

```
security certificate show
```

3. If a self-signed certificate does not exist, run the following command as a one-time command to generate and install a self-signed certificate:

```
security certificate create -vserver vserver_name -common-name lab.companyname.com -size 2048 -
country US -state CA -locality Sunnyvale -organization IT -unit Software -email-addr
user@example.com
```

4. Configure and enable SSL and HTTPS access and disable Telnet access.

```
system services web modify -external true -sslv3-enabled true
system services firewall policy delete -policy mgmt -service http -action allow
system services firewall policy create -policy mgmt -service http -action deny -ip-list 0.0.0.0/0
system services firewall policy delete -policy mgmt -service telnet -action allow
system services firewall policy create -policy mgmt -service telnet -action deny -ip-list
0.0.0.0/0
```

Note: It is normal for some of these commands to return an error message stating that the entry does not exist.

9 Management Best Practices

9.1 Managing Red Hat Enterprise Linux 6 and KVM

Any of the management servers in the subsequent section can 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).

Libvirt

libvirt is not a server, but a virtualization API and toolkit written in C. Although it is the primary API for KVM, it also supports Xen, OpenVZ, VMware® ESX®, GSX, Microsoft Hyper-V™ , and several others. Scripts and tools that are used to manage KVM should make API calls to libvirt.

Virtual Machine Manager

The Virtual Machine Manager (VMM) is a graphical interface to the "libvirt" virtualization API library. As such, it requires a graphical environment to operate. As mentioned earlier in this report, the best practice is to not install graphical packages on a production RHEL 6 KVM host. If graphical tools are required, it is better to set up a separate remote administration host to manage the RHEL 6 KVM hosts and guests.

Use of a Remote Administration Host

NetApp recommends using a remote host to administer the RHEL 6 KVM environment, if graphical tools are required. 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, as well as a GUI desktop such as Gnome or KDE.

It is important to note that all of the security best practices (IPtables, SELinux, SSH keys, and so on) for an RHEL 6 KVM host apply to a remote administration host as well.

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)
- Secure host to run Snap Creator™ Server (backup framework)

RHN and RHN Satellite

Red Hat Network (RHN) is a secure Web portal as well as the source for all Red Hat–related packages, updates, errata, and management tools for RHEL servers. It is a Red Hat and NetApp best practice to subscribe all Red Hat systems to RHN to keep up with security patches as well as compliance.

RHN Satellite is essentially an on-site instantiation of RHN. Instead of many systems subscribed to RHN, only the Satellite server is subscribed. All Red Hat systems are then subscribed to the Satellite server. This has many added benefits such as reduced external network traffic as well as the ability to highly customize software channels and user management. Additionally, RHN Satellite can be used to provision new RHEL servers. RHN Satellite can be deployed on an RHEL KVM guest.

Kickstart Server

Kickstart is a means of providing semiautomated RHEL installations and can be deployed using CD and answer file, HTTP, NFS, or FTP. It is a best practice to deploy Kickstart as a server on an RHEL KVM guest or as part of a RHN Satellite server.

9.2 NetApp OnCommand System Manager 2.0x RHEL

For all of the NetApp specific management tools described in the following section, refer to NetApp TR-3710: Operations Manager, Provisioning Manager, and Protection Manager Best Practices Guide for detailed explanations and deployment requirements.

Overview

NetApp System Manager is a simple tool based on Java® that enables administrators to easily manage NetApp storage systems. System Manager provides recommendations based on NetApp best practices for common storage management tasks and workflows to save time and prevent configuration errors.

System Manager is the primary element management GUI for Data ONTAP 8.1 and later. It replaces FilerView® for 7G management. It encompasses most of the element management features supported by the current Cluster Manager for clustered Data ONTAP. Data ONTAP CLI and autogenerated Web GUI are required only for advanced operations because most of the common goals can be achieved from System Manager. System Manager is an out-of-the-box management application that can be installed on laptop, desktop, and server systems.

Administrative Capabilities

System Manager enables administrators to easily control the powerful capabilities of NetApp storage systems:

- Manage disks, pooled storage, shares or exports, Snapshot copies, and network configuration
- Provision storage for SAN (iSCSI or FCP) and NAS (CIFS or NFS), for both physical and virtualized server environments
- Leverage space efficiency features such as thin provisioning and deduplication
- Get real-time views of system performance

Features

System Manager provides these features:

- **Storage management.** Volume, aggregate, SAN, NAS management, Vserver, and vFiler® unit setup
- **Diagnostics.** Cluster health dashboard
- **Configuration.** Network settings, protocol setup, user setup, security, syslog setup, and AutoSupport
- **Data protection.** SnapMirror setup

Guidelines

The following guidelines apply to System Manager:

- RHEL 5 is supported and should have Firefox 2.5 or 3.x. or higher.
- The installer detects JRE, and installation does not proceed unless JRE v1.6 or higher is on the system.
- The Linux platform requires Sun JRE v1.6 or higher. If the system contains non-Sun based JRE, the installer does not proceed.
- The number of instances of the application is restricted to one.

- A unique session ID is used every time the application is launched to prevent multiple users from accessing the same instance of System Manager.
- System Manager uses HTTPS as the default protocol for communication with the storage systems.

9.3 Operations Manager

Product Overview

Operations Manager provides NetApp customers with a comprehensive monitoring and management dashboard for IT storage infrastructure.

Operations Manager enables scalable storage management for NetApp NAS and SAN storage assets. From a central point of control, the solution provides alert, report, and configuration tools that help customers keep storage infrastructure in line with business requirements and policies to maximize availability and reduce the total cost of ownership. Operations Manager provides comprehensive reports of utilization and trend information to support capacity planning, space usage, and data backup space allocation.

9.4 NetApp Management Console 3.0

Overview

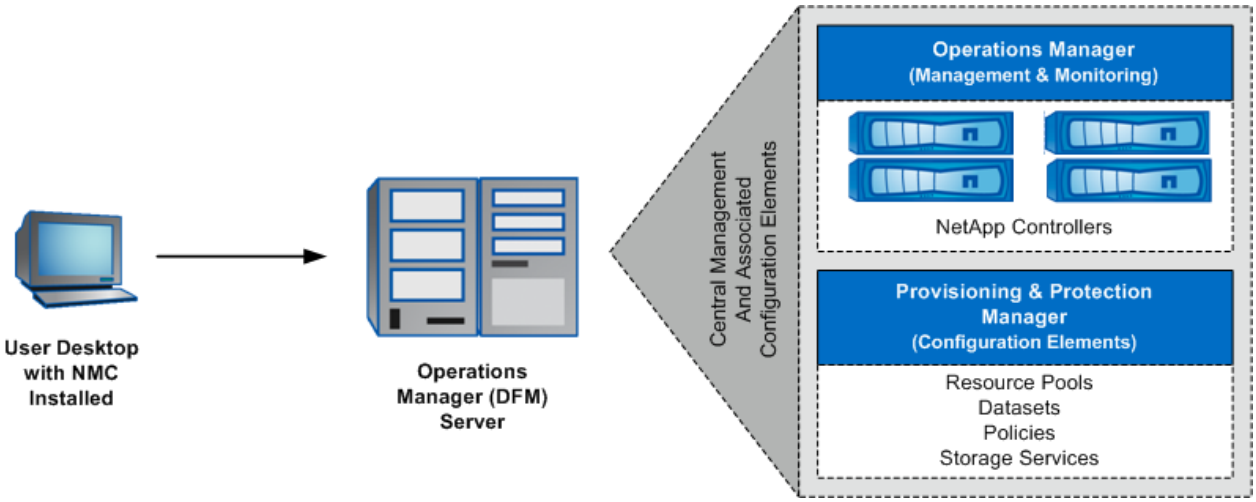
NetApp management console (NMC) is used by administrators to carry out management tasks. It is aided by DataFabric® Manager, but it runs on Windows or Linux system separate from the server on which DataFabric Manager is installed.

NMC allows storage, application, and server administrators to perform management tasks such as data backup protection, space management, resource provisioning, data migration, and performance tuning, without having to switch between separate user interfaces.

The DataFabric Manager server provides infrastructure services (such as discovery, monitoring, RBAC, auditing, and logging for products in the storage and data suites) for NetApp Manageability Software client applications. The DataFabric Manager software runs on a separate server and is itself managed through Operations Manager, the Web-based user interface of DataFabric Manager.

Figure 19Figure shows the NMC overview.

Figure 19) NMC overview.



Applications That Run in NetApp Management Console

The Performance Advisor and the licensed Protection Manager and Provisioning Manager applications run in NMC.

9.5 Performance Advisor

This application provides a single location from which to view comprehensive information about storage system and MultiStore® vFiler unit performance and perform short-trend analysis. The application also helps identify, in the data infrastructure, the causes and potential causes of reduced performance.

Performance Advisor is automatically enabled with the Operations Manager core license.

9.6 Protection Manager

This application provides a policy-based management tool to help unify and automate backup and mirroring operations. The application uses a holistic approach to data protection. It provides end-to-end workflow-based design and seamless integration of SnapVault, SnapMirror, and Open Systems SnapVault to make it possible to manage large-scale deployments easily.

The disaster recovery feature of the protection application enhances data protection services by making it possible to continue providing data access to users, even in the event of a mishap or disaster that disables or destroys the storage systems in the primary data node. If the disaster recovery license is installed, secondary storage systems can quickly be enabled to provide primary data storage access to users with little or no interruption, until the primary storage systems are reenabled or replaced.

9.7 Provisioning Manager

This application helps to simplify and automate the tasks of provisioning and managing storage. The application provides policy-based provisioning and conformance of storage in datasets. It also makes it possible to manually add volumes or qtrees to a dataset at any time, provides manual controls for space and capacity management of existing storage and newly provisioned storage, and allows the migration of datasets and vFiler units to a new storage destination.

The deduplication feature of the provisioning application enhances data provisioning services by making it possible to eliminate duplicate data blocks to reduce the amount of storage space used to store active data.

9.8 Storage Efficiency Dashboard

Overview

The NetApp Operations Manager Storage Efficiency Dashboard plug-in answers a number of questions related to storage utilization and storage efficiency savings. It also identifies ways to improve storage utilization. The storage efficiency dashboard is a script that can be installed on Operations Manager 3.8.1 and higher. It produces a graphical dashboard report of the utilization and efficiency of NetApp storage environments.

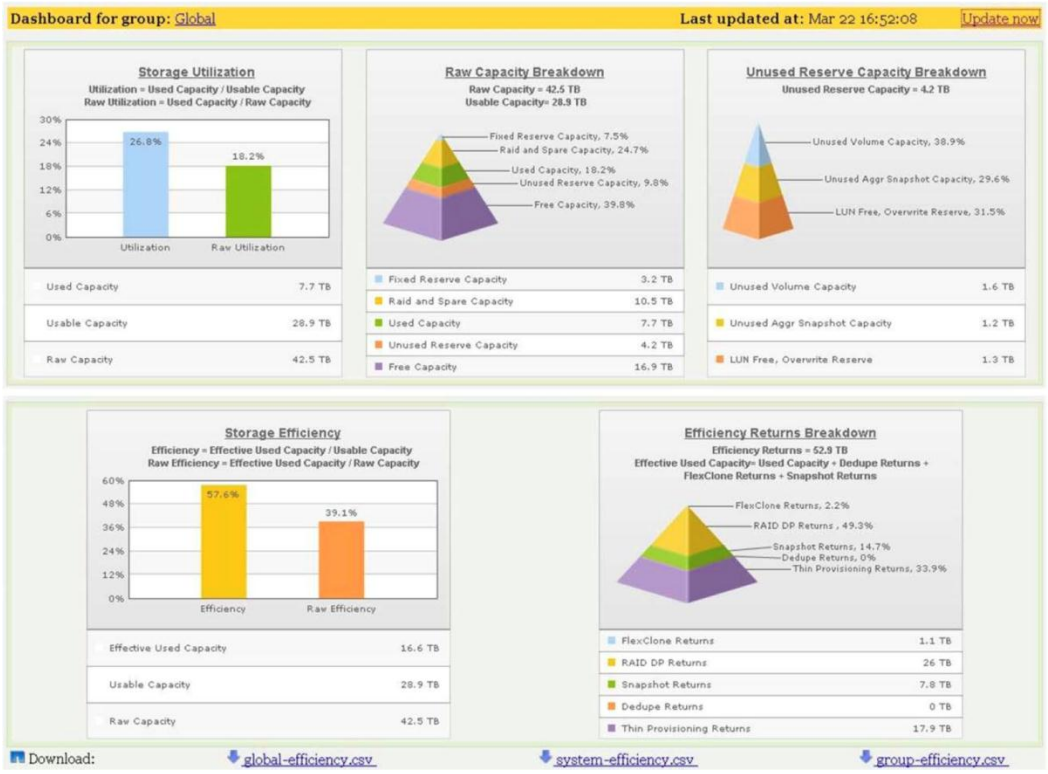
The script can be installed on an Operations Manager instance running on Windows or Linux servers and can be scheduled to generate a set of Web pages that provide an efficiency dashboard for the NetApp storage systems managed by Operations Manager. It produces two primary charts, with additional charts available to provide detailed breakdowns of how the storage space is consumed. These charts represent all storage systems monitored by Operations Manager, groups of storage systems as grouped in Operations Manager, or a single storage system.

The two primary chart types are:

- **Utilization charts.** These charts provide the utilization data for all systems in a resource group or for an individual storage system.
- **Efficiency charts**. These charts show the effect that NetApp storage efficiency technologies, such as deduplication, thin provisioning, and FlexClone, have across resource groups or individual storage systems.

Figure 20 provides a sample screenshot of the Storage Efficiency Dashboard in NetApp Operations Manager.

Figure 20) Storage Efficiency Dashboard.



9.9 RLM with Clustered Data ONTAP

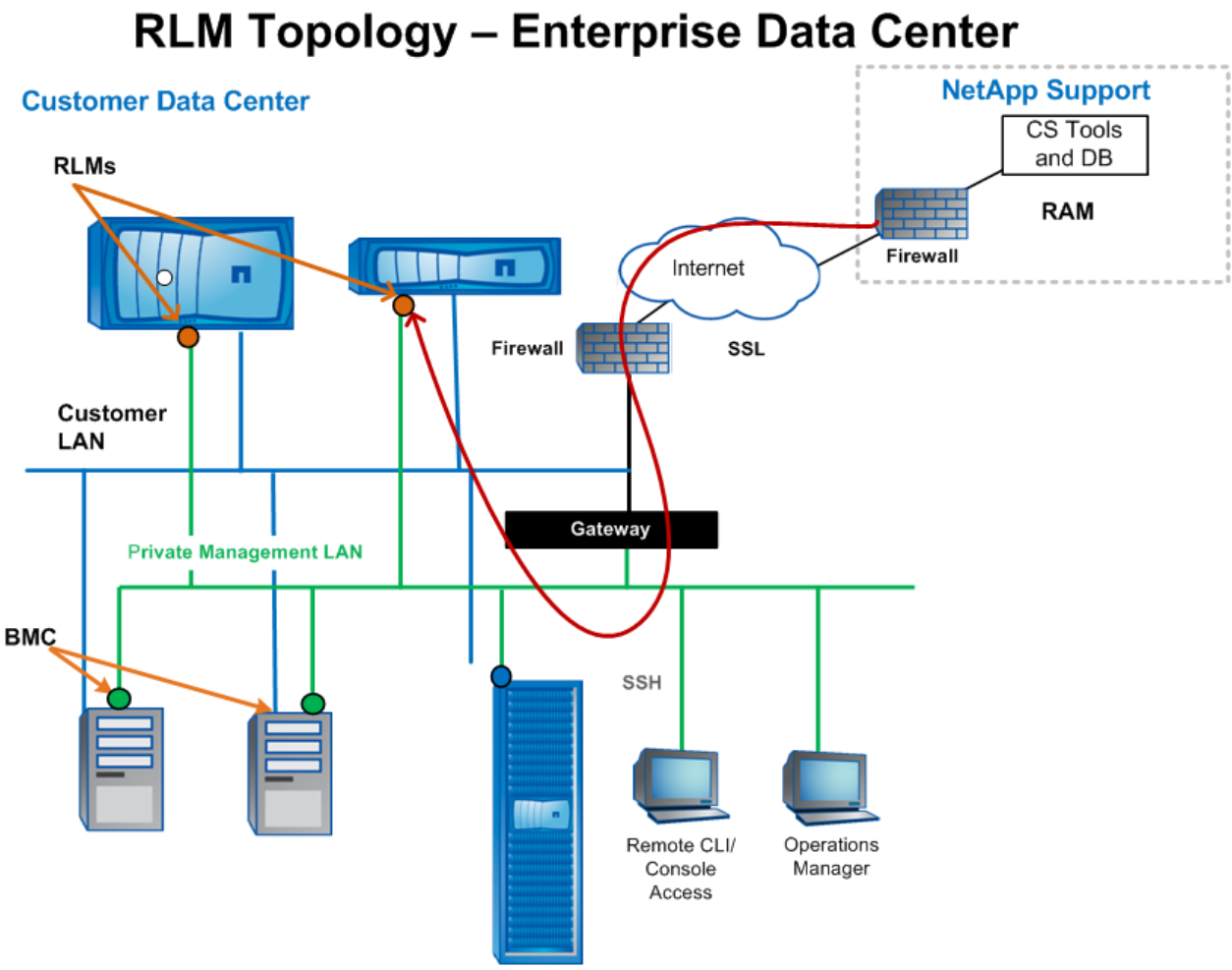
Overview

RLM is a physical card incorporated on NetApp FAS and V-Series storage systems that provides remote access and management capabilities for monitoring, troubleshooting, logging, and sending alert notifications. The V-Series and FAS3xxx and 6000 series storage systems have an expansion slot with a dedicated Ethernet interface for connecting the RLM card to the system controller. The RLM stays operational regardless of the operating state of the storage system, enabling storage administrators to remotely access their storage systems.

With the RLM enabled, customers can remotely manage their storage systems without the need for dedicated 24/7 on-site storage administrators, as shown in Figure 21. RLM technology utilizes existing data center IP infrastructure, eliminating the need for remote support infrastructure, such as terminal concentrators or power controllers.

Figure 21 illustrates the RLM topology.

Figure 21) RLM topology.



Easy-to-Use Remote Management Capabilities

The RLM provides the following easy-to-use remote management capabilities on NetApp storage systems:

- Remotely administers the storage system by using the Data ONTAP CLI through the RLM's system console redirection feature.
- Remotely accesses the storage system and diagnoses error conditions, even if the storage system has failed, by performing the following tasks:
- Views the storage system console messages captured in the RLM's console log
- Views the storage system events captured in the RLM's system event log
- Initiates a storage system core dump
- Power-cycles the storage system (or turns it on or off)
- Resets the storage system
- Reboots the storage system

Extends AutoSupport capabilities by sending alerts and down system or down filer notifications through an AutoSupport message when the storage system goes down, regardless of whether the storage system can send AutoSupport messages. Other than generating these messages on behalf of a system that is

down and attaching additional diagnostic information to AutoSupport messages, the RLM has no effect on the storage system's AutoSupport functionality.

Note: AutoSupport configuration settings and message content behavior of the RLM are inherited from Data ONTAP.

Note: The RLM does not rely on the system node autosupport modify command's -transport parameter setting to send notifications. The RLM uses SMTP.

In addition to AutoSupport messages, the RLM generates SNMP traps to configured trap hosts for all `down system` or `down filer` events, if SNMP is enabled for the RLM.

The RLM remains operational regardless of the operating state of the storage system. It is powered by a standby voltage, which is available as long as the storage system has input power to at least one of the storage system's power supplies.

The RLM has a single temperature sensor to detect ambient temperature around the RLM board. Data generated by this sensor is not used for any system or RLM environmental policies. It is only used as a reference point that might help troubleshoot storage system issues. For example, it might help a remote system administrator determine if a system was shut down due to an extreme temperature change in the system.

The RLM has a nonvolatile memory buffer that stores up to 4,000 system events in a SEL to help diagnose system issues. The event list from the SEL is automatically sent by the RLM to specified recipients in an AutoSupport message.

The records contain the following data:

- Hardware events detected by the RLM—for example, system sensor status about power supplies, voltage, or other components.
- Errors (generated by the storage system or the RLM) detected by the RLM—for example, a communication error, a fan failure, a memory or CPU error, or a `boot image not found` message.
- Critical software events sent to the RLM by the storage system—for example, a system panic, a communication failure, an unexpected boot environment prompt, a boot failure, or a user-triggered `down system` as a result of issuing the system reset or system power cycle command.

The RLM monitors the storage system console regardless of whether administrators are logged in or connected to the console. When storage system messages are sent to the console, the RLM stores them in the console log. The console log persists as long as the RLM has power from either of the storage system's power supplies. Since the RLM operates with standby power, it remains available even when the storage system is power-cycled or turned off.

Hardware-assisted takeover is available on systems that support the RLM and have the RLM modules set up. For more information about hardware-assisted takeover, refer to the Data ONTAP Cluster-Mode High-Availability Configuration Guide.

The RLM supports the SSH protocol for CLI access from UNIX clients and PuTTY for CLI access from PC clients.

9.10 Accounts That Can Access the RLM

User accounts that are created with the service-processor application type have access to the RLM CLI on any node of the cluster that supports the RLM. RLM user accounts are managed from Data ONTAP and authenticated by password.

Do not create user accounts directly from the RLM CLI. RLM user accounts are created and managed from Data ONTAP, using the security login commands with the `-application` parameter set to `service-processor`. RLM supports only password authentication; therefore, when the RLM user account is created, the authentication method (the `-authmethod` parameter) must be set to `password`.

Display current RLM user accounts using the `-application service-processor` parameter of the `security login show` command.

The cluster user account `admin` includes the service-processor application type and has access to the RLM CLI by default.

The system prevents user accounts with names that are reserved for the system, such as `root` and `naroot`, from being created.

Note: Do not use a system-reserved name to access the cluster or the RLM.

9.11 Service Processor

Overview

Service processor (SP) is a physical device that enables storage administrators to access, monitor, and troubleshoot the storage system remotely. The SP is available on all NetApp systems except for the 20xx, 30xx, 31xx, and 60xx systems. The SP remains operational regardless of the operating state of the storage system. It is powered by a standby voltage, which is available as long as the system has input power to at least one of the system's power supplies. It is also connected to the system through the serial console.

SP Remote Management Capabilities

The SP provides the following capabilities:

- Remotely administers the storage system by using the SP CLI.
- Remotely accesses the system console to run the Data ONTAP commands.

  Note: The SP can be accessed from the system console. If the system console becomes unresponsive, press Ctrl+G to access the SP CLI.

- Monitors environmental sensors and logs system events.
- Has a nonvolatile memory buffer that stores up to 4,000 system events in a single SEL.
- Monitors the system console regardless of whether administrators are logged in or connected to the console.
- The SP extends AutoSupport capabilities by sending alerts and `down system` or `down filer` notifications through an AutoSupport message when the storage system goes down, regardless of whether the storage system can send AutoSupport messages. Other than generating these messages on behalf of a system that is down and attaching additional diagnostic information to AutoSupport messages, the SP has no effect on the storage system's AutoSupport functionality. AutoSupport configuration settings and message content behavior of the SP are inherited from Data ONTAP.

  Note: The SP does not rely on the system node autosupport modify command's - transport parameter setting to send notifications. The SP uses SMTP.

  Note: In addition to AutoSupport messages, if SNMP is enabled for the SP, the SP generates SNMP traps to configured trap hosts for all down system or down filer events.

  Note: The SP supports the SSH protocol for SP CLI access and the ability to execute Data ONTAP commands remotely.

10 Data Protection Best Practices for RHEL 6 KVM

10.1 Snapshot Clustered Data ONTAP

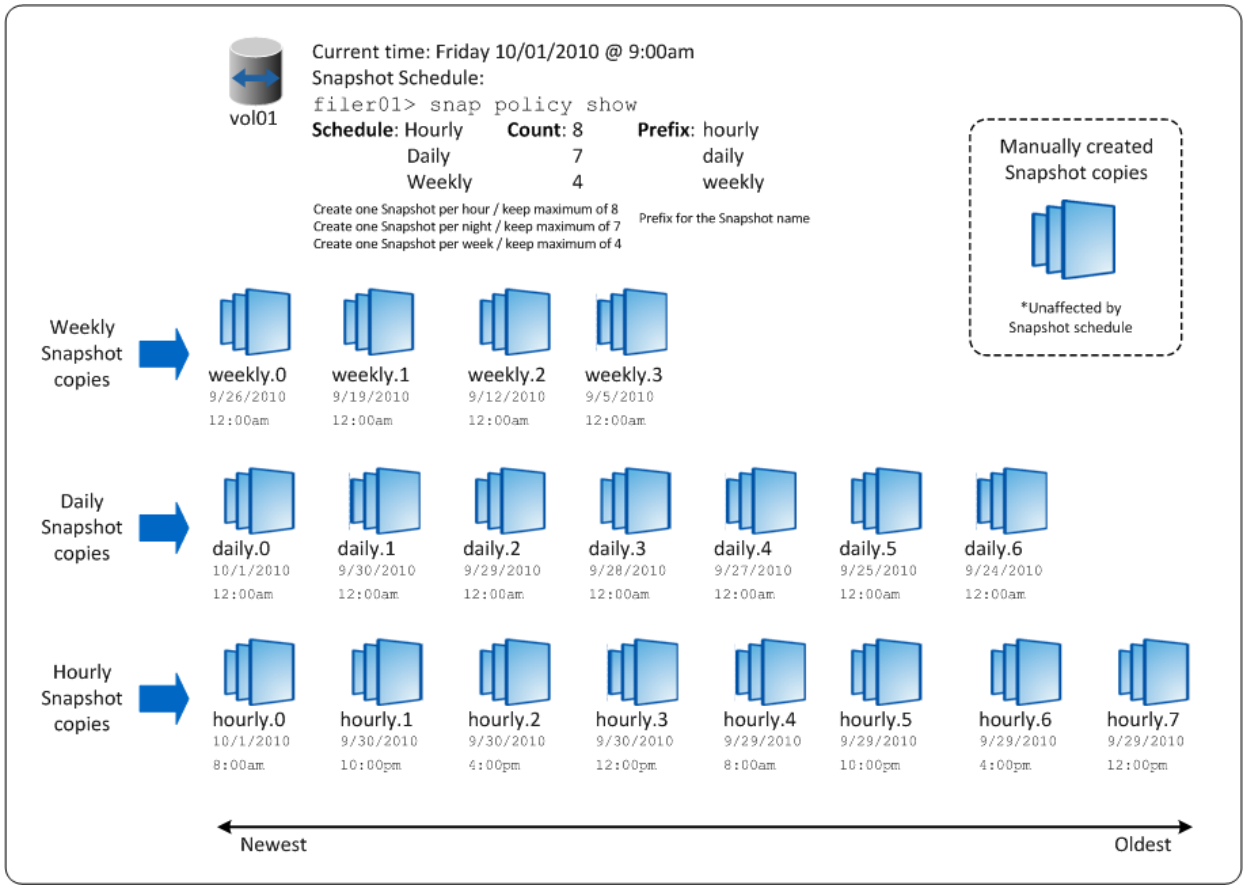
Overview

A Snapshot copy is a read-only image of a FlexVol volume or an aggregate that captures the state of the file system at a point in time.

For information about FlexVol volumes or aggregates, refer to the Data ONTAP 8.1 Cluster-Mode Physical Storage Management Guide. Data ONTAP maintains a configurable Snapshot copy schedule that creates and deletes Snapshot copies automatically for each volume. Snapshot copies can also be created and deleted manually by using the `snap create` and `snap delete` commands.

Figure 22 shows an example volume (`vol01`) and the corresponding Snapshot copies that are created by the schedule shown in the diagram.

Figure 22) Snapshot copy example that uses the `snap policy show` command.



Guidelines and Restrictions

Avoid scheduling Snapshot copy creation at the same time as SnapMirror updates or SnapVault activity. If these schedules conflict, Snapshot copy creation might not occur.

Stagger the Snapshot copy update schedules so that SnapMirror activity does not begin or end at the exact minute a Snapshot operation attempts to create a Snapshot copy.

Snapshot Copies in a SAN Environment

Administrators can use Snapshot technology to make copies in the SAN environment when the data within a Data ONTAP LUN is in a consistent state; however, Data ONTAP does not know whether an application is accessing the data inside the LUN (that is, whether or not it is in a consistent state). Therefore, before creating a Snapshot copy, administrators must quiesce the application or file system using the LUN. This action flushes the host file system buffers to disk and provides a consistent Snapshot copy.

One way to accomplish this is to use batch files and scripts on a host that has administrative access to the system. The SnapDrive®, SnapManager, and Snap Creator products also quiesce LUNs before creating Snapshot copies and should be used whenever possible.

Snapshot Restore Guidelines

Snapshot promote or Snapshot restore allows administrators to quickly revert a local volume or local file to the state it was in when a particular Snapshot copy was created. In most cases, reverting a file or volume is much faster than restoring files from tape or copying files from a Snapshot copy to the active file system. Snapshot restore is implemented in advanced mode.

Follow these guidelines for using Snapshot promote or Snapshot restore:

- If the amount of data to be recovered is large, Snapshot promote (restore) is the preferred recovery method, because it takes a long time to copy large amounts of data from a Snapshot copy or to restore from tape.
- If the file to be recovered needs more space than the amount of free space in the active file system, the file cannot be restored by copying it from the Snapshot copy to the active file system.
- If the volume that must be restored is a root volume, it is easier to copy the files from a Snapshot copy or restore the files from tape than to use Snapshot promote (restore) because rebooting is not necessary. However, if only a corrupted file on a root volume must be restored, a reboot is not necessary.
- If the entire root volume is reverted, the system reboots with the configuration files that were in effect when the Snapshot copy was created.

Create a Snapshot Copy in Clustered Data ONTAP

Manually create a Snapshot copy from the controller CLI. In this example, a Snapshot copy named snapshot_name on volume_name is created.

```
snap create -vserver vserver_name -volume volume_name -snapshot snapshot_name
```

Restore Entire Volume from Snapshot Copy

To perform a Snapshot restore, perform the following step:

1. Run the following CLI operation.

```
vol snapshot restore -vserver vserver_name -volume volume_name -snapshot snapshot_name
```

Restore File from Snapshot Copy

To restore a single file from a Snapshot copy, perform the following step:

1. Run the following CLI operation.

```
vol snapshot restore-file -vserver vserver_name -volume volume_name -snapshot snapshot_name -path
file_path
```

10.2 Snap Creator

Snap Creator is a software framework that integrates NetApp data protection technologies with multiple applications and platforms, such as RHEL 6 KVM. Snap Creator facilitates the management of backups, reduces the complexity of solutions, and provides a method to easily take advantage of NetApp technology with minimum programming effort on the user's part.

Overview

Snap Creator uses Snapshot, SnapRestore, and FlexClone technologies to simplify backups, restores, and cloning by communicating with NetApp storage through NetApp API commands. The framework allows for not only triggering NetApp Snapshot copies of volumes (datastores), but also any activities that need to occur before and after the Snapshot copy is created. Additionally, it can also trigger SnapMirror activities between NetApp controllers and/or data centers to meet disaster recovery and backup requirements.

Snap Creator Server

The following guidelines apply to configuring Snap Creator:

- The Snap Creator agent/server configuration is a client/server type of architecture.
- The Snap Creator agent is installed on the host with the targets to be backed up within an agent/server architecture.
- The default port used by the Snap Creator agent/server architecture for communication is 9090.
- The Snap Creator server is installed on the centralized host that communicates with other hosts that have a Snap Creator agent installed and running.
- The Snap Creator agent executes any prescripts or postscripts and commands on its host.
- If plug-ins and precommands or postcommands are not needed, then the agent installation might not be necessary.
- Snap Creator supports the use of both GUI and CLI. The GUI is configured when Snap Creator is installed.

Architecture Components

Figure 23 shows the components of the Snap Creator 3.x server architecture.

Figure 23)Snap Creator 3.x server architecture.

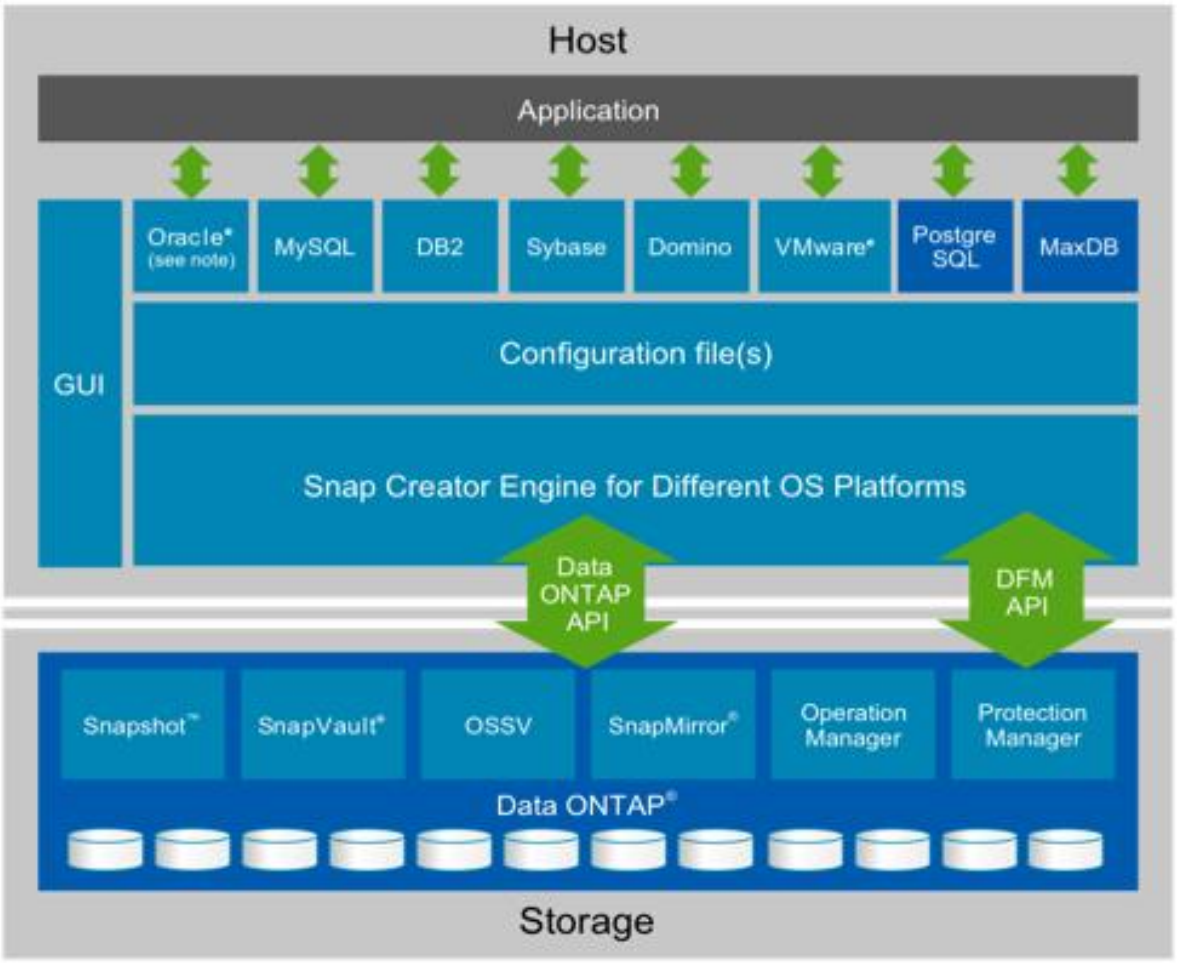


Figure 24 shows the components of the Snap Creator 3.x agent architecture.

Figure 24) Snap Creator 3.x agent architecture.

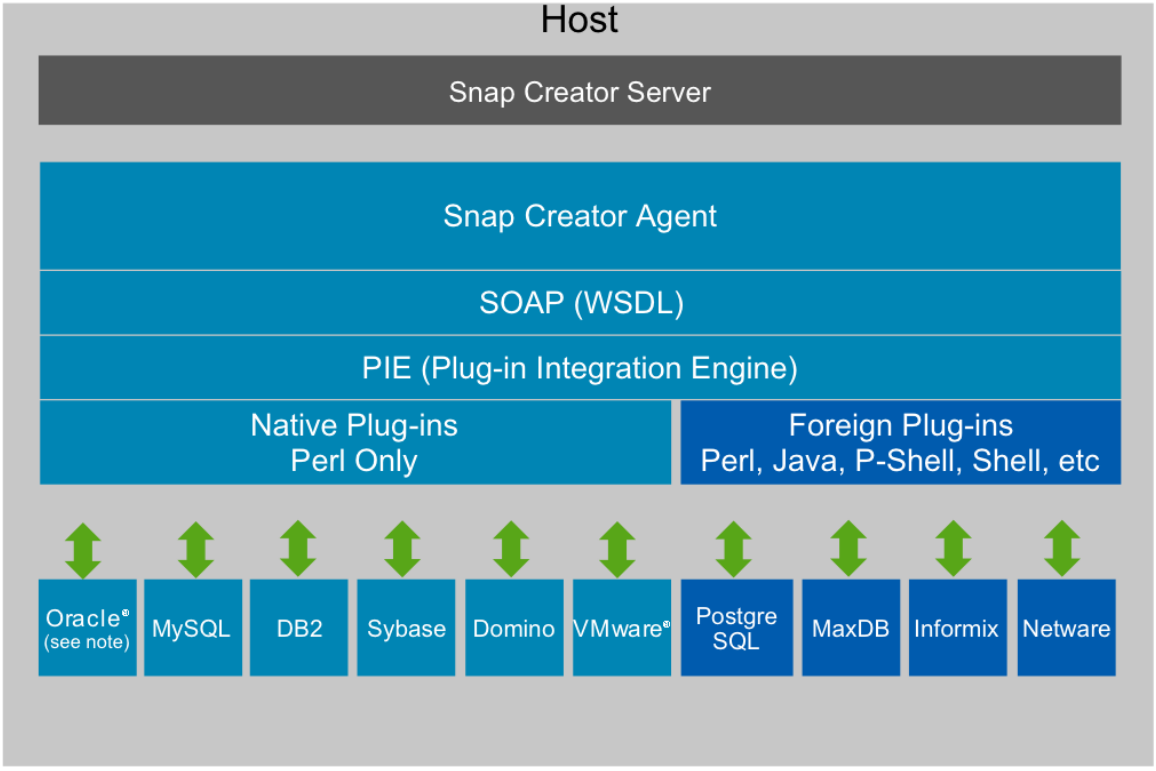
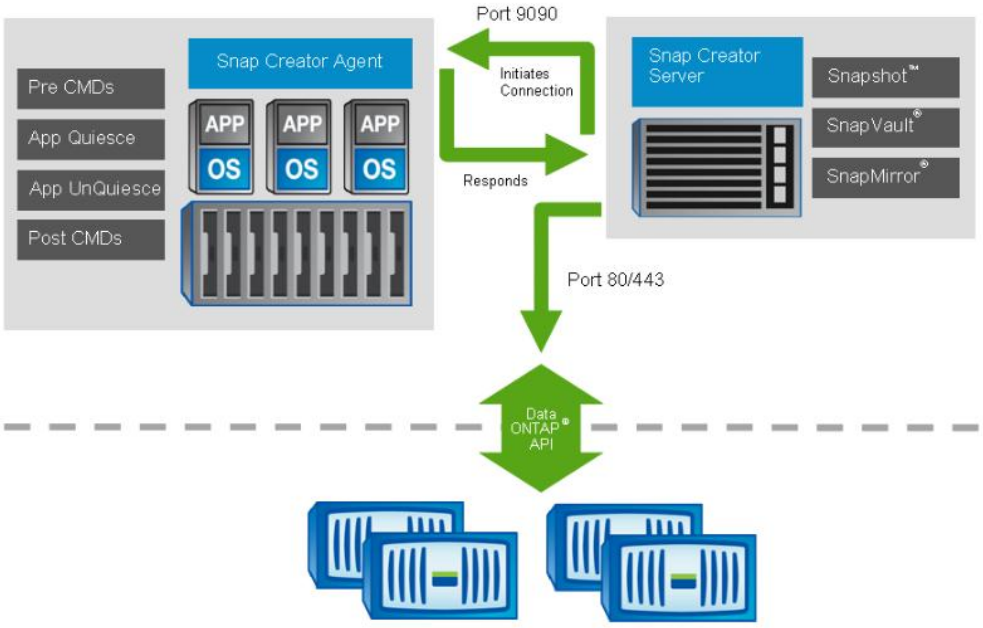


Figure 25 shows the combined agent/server architecture.

Figure 25) Snap Creator 3.x agent/server architecture.



The main components of the Snap Creator architecture are:

- **Snap Creator GUI.** The GUI allows the creation of customized configuration files and new profiles for the environment.
- **Snap Creator server.** The server is where the GUI, the configuration files, and the CLI reside. It is the main engine for the Snap Creator Framework. The Snap Creator server engine communicates with the NetApp storage controller through API commands. It also integrates with OSSV, Operations Manager, Protection Manager, and SnapDrive.
- **Snap Creator agent.** The agent is a lightweight daemon that runs remotely or locally and allows the Snap Creator server engine to send commands or operations to that agent based on the configuration of the workflow. The communication layer is HTTP or HTTPS, using SOAP.
- **Snap Creator configuration files.** The configuration files reside within a profile. Each profile is a directory within the installation directory. The configuration file controls the workflow of Snap Creator. A profile may contain many configuration files, but only one can be active during each execution of Snap Creator. The configuration file contains information about how to name Snapshot copies, how to manage Snapshot retention, whether cloning is required, how many FlexClone volumes to retain, how to manage the data protection integration, and so forth.
- **Snap Creator application plug-ins.** Built-in plug-ins and applicable support for them can be found in the documentation for the applications and in the IMT. The plug-ins are provided to assist with application consistency during the creation of backups or FlexClone volumes or for the management of specific components of the application (such as Oracle® archive logs).

Install Snap Creator Framework Server

1. Download Snap Creator Framework from the NetApp Support site.
2. Extract the Snap Creator Framework tar file.

```
cd /opt
tar -zxvf NetApp_Snap_Creator_Framework3.*
```

3. Run Snap Creator Framework as the root user for the initial setup.

Note: The serial number asked for during the prompts is the serial number of the storage controller, and it is optional.

```
cd /opt/scServer*
chmod 755 snapcreator
./snapcreator --profile setup
Welcome to the NetApp Snap Creator Framework!
End User License Agreement for NetApp, Inc. Software
.
.
.
Do you accept the End User License Agreement (y|n): y
Setup NetApp Snap Creator Framework Server (y|n): y
Enter serial number:
Enable GUI job monitor (Y|N): n
Please Enter GUI Administrator Username: snapcreator_admin
Please Enter password for snapcreator_admin:
Please Confirm password for snapcreator_admin:

INFO: Updated NetApp Snap Creator Framework GUI

INFO: To start GUI please do the following:

cd /opt/scServer*/gui
java -jar snapcreator.jar
or
java -jar snapcreator.jar  -db_password <db_password> -db_port <db_port> -db_user name <db_user
name> -gui_port <gui_port>

INFO: To access NetApp Snap Creator Framework GUI goto "http://unixhost:8080" or
"http://unixhost:<gui_port>"
```

4. Start the Snap Creator Framework GUI.

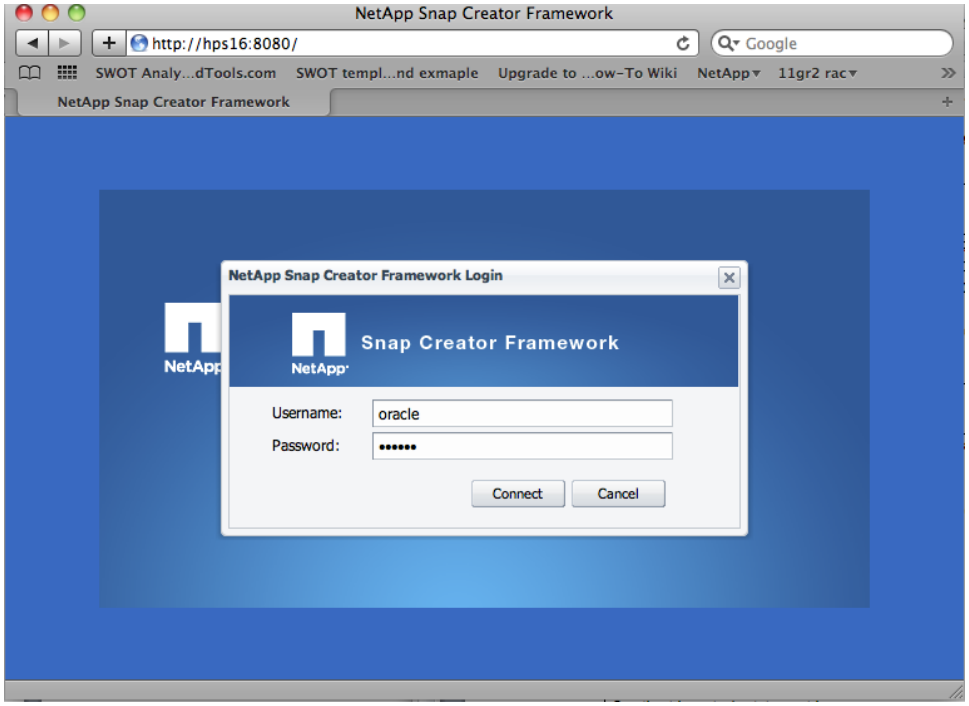
Note: Before starting the Snap Creator server, use netstat or a similar tool to verify that the network port that it uses (8080 by default) is not in use.

```
cd gui
java -jar snapcreator.jar

Starting Derby Server
Got an embedded connection.
Testing embedded connection by executing a sample query
number of rows in sys.systables = 22
Creating DB Schema
Creating table frequencies
Populating table frequencies
Creating table actions
Populating table actions
Creating table tasks
Creating table tasks_statuses
Creating trigger task_statuses_trigger
Creating Table qrtz_job_details
Creating Table qrtz_job_listeners
Creating Table qrtz_triggers
Creating Table qrtz_simple_triggers
Creating Table qrtz_cron_triggers
Creating Table qrtz_blob_triggers
Creating Table qrtz_trigger_listeners
Creating Table qrtz_calendars
Creating Table qrtz_paused_trigger_grps
Creating Table qrtz_fired_triggers
Creating Table qrtz_scheduler_state
Creating Table qrtz_locks
Populating qrtz_locks
log4j:WARN No appenders could be found for logger (org.mortbay.log).
log4j:WARN Please initialize the log4j system properly.
Starting Jetty Server
Jetty Server Started on port: 8080
```

5. Validate the Snap Creator Framework GUI startup by navigating to the local host on port 8080.

Note: If going through a firewall, open the network port (8080 by default).



6. If a Snap Creator user is created specifically on the NetApp cluster, then create an encrypted password. This step prevents a plain text password from being inserted into a configuration file on the host on which Snap Creator is installed.

```
./snapcreator  --cryptpasswd
Please Enter Password:
Your encrypted password is: 53616c7465645f5f614d4964d340f7f2d26eef38f443f5ea9c2f8020015a2dfa
```

7. It is also recommended that a standard Linux start/stop script be created to automatically start the Snap Creator server on boot. A sample start/stop script is located in the appendix.

Snap Creator Agent

Overview

The Snap Creator agent is a lightweight daemon that runs remotely or locally and allows the Snap Creator server to send quiesce or unquiesce operations to a given database.

The Snap Creator agent remotely handles operations on an application through the Snap Creator plug-ins. All Snap Creator configurations are centrally stored on the Snap Creator server, and all backup tasks can be scheduled from the same host. This architecture provides a single pane of glass (SPOG) for backup and restore operations.

Snap Creator uses the Snap Creator agent, which runs as a daemon, to quiesce applications. The default port is 9090, but any port can be used.

Single Object Access Protocol (SOAP) over HTTP is used for communication between the agent and the server. Any SOAP client can interact with the agent through the use of the Web Services Description Language (WSDL). Currently, Apache CXF (for Java) and Windows PowerShell™ (for Windows) can be used to create an agent.

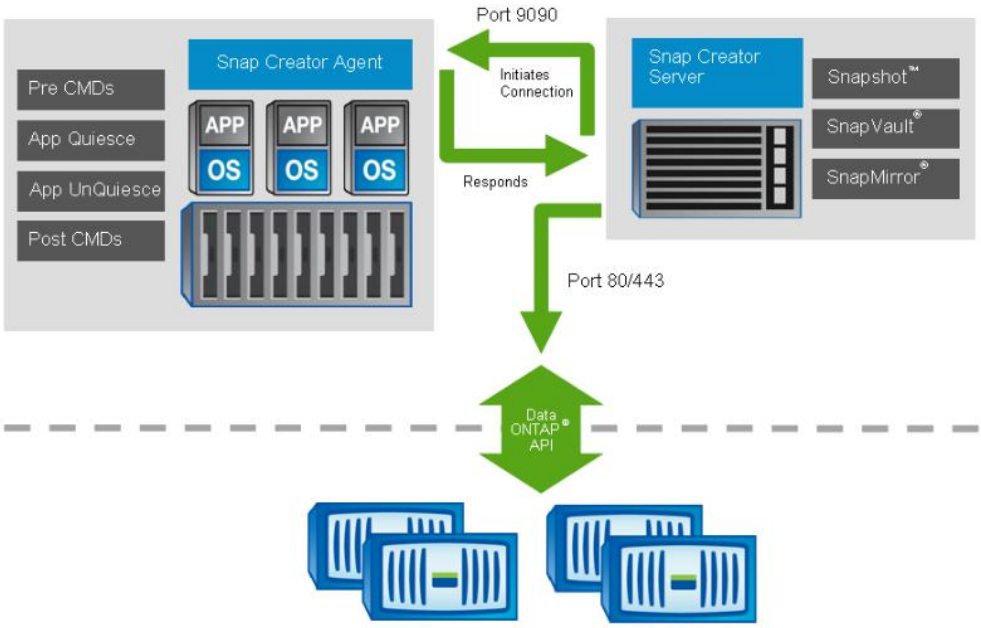
The supported application plug-ins are built into the agent. Other community plug-ins are available as source codes and can be added to the `/plug-ins` directory.

In addition to the application plug-ins, all PRE commands, POST commands, and APP commands can be executed remotely through the agent. This provides the ability to mount file systems or perform additional application processing remotely. The agent has a configuration file (`agent.conf`) in which certain commands can run. This file is located under `/path/so/ scAgent_v<#>/config/agent.conf`. By default, all commands are denied, which means that only the built-in or community plug-ins can execute commands through the agent. PRE or POST scripting commands or scripts must be added to the `agent.conf` file.

Note: Snap Creator 3.5.0 also supports Snap Creator 3.4.0 agents.

Figure 26 shows the Snap Creator agent communication.

Figure 26) Snap Creator agent communication.

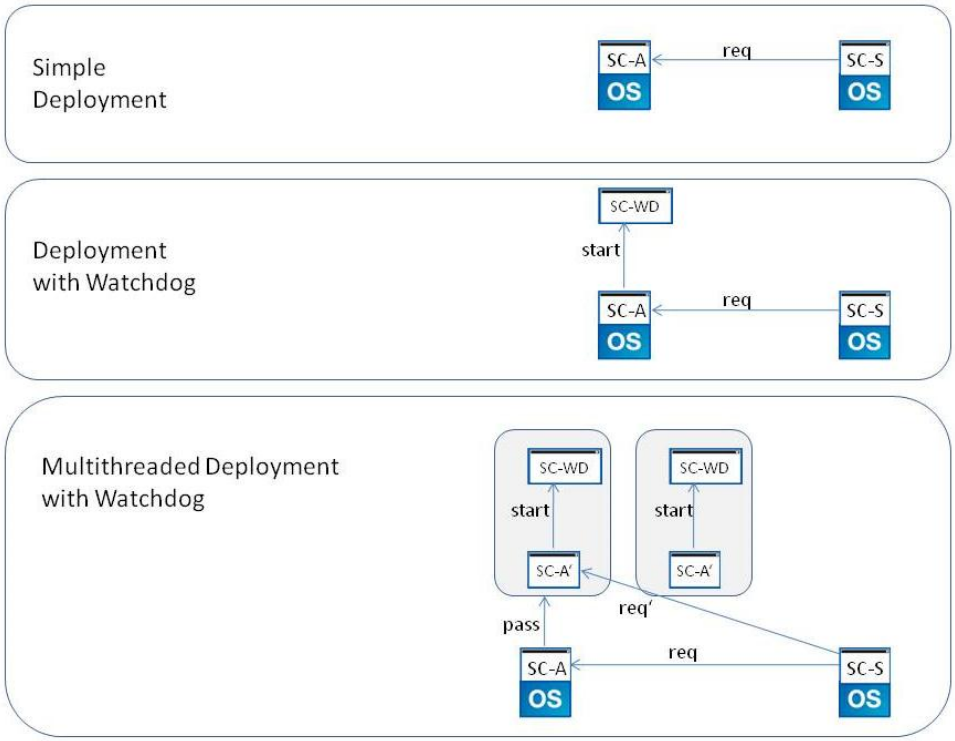


Snap Creator Agent Multithreading

The Snap Creator agent handles parallel requests by distributing the requests coming from the agent itself.

Figure 27 shows the different types of agent multithreading deployment.

Figure 27) Snap Creator agent multithreading.



Simple Deployment

In a simple deployment, the Snap Creator server opens a connection to the Snap Creator agent and sends the requests. Unless the Snap Creator server stops, the connection is open, and the agent is blocked from accepting further requests.

Deployment Using a Watchdog

After accepting the Snap Creator request to quiesce the database, the Snap Creator agent creates a separate process (watchdog). This watchdog inherits the environment and all the settings of the parent process and unquiesces the database after a configured timeout.

In this scenario, the agent is blocked until the Snap Creator server stops or explicitly closes the connection.

Multithreaded Deployment Using a Watchdog

In the intended multithreaded deployment, the Snap Creator agent immediately creates an additional process (`SC-A'`) and passes the communication responsibility (`req'`) to this process. This newly created process inherits the basic Snap Creator agent settings. The Snap Creator server communicates with this process, and if configured, the process creates a watchdog when the quiesce operation is called.

After an additional process is created, the Snap Creator agent is able to accept additional requests.

Note: A multithreaded agent is not supported for the Windows platform.

Snap Creator Agent Security

Snap Creator runs centrally and uses an agent to communicate with the database servers. The agent is a daemon that runs on any port; by default, it runs on port 9090. This port must be open between the Snap Creator server and the server running the agent. Communication between the agent and the Snap Creator server occurs through SOAP over HTTP. The host information and the commands should be specified in the agent configuration file.

The Snap Creator agent uses a file named `agent.conf` to secure its functionalities. The `agent.conf` file allows or restricts hosts and commands.

The `agent.conf` file is located in the same `config` subdirectory where the agent is installed: `/path/to/scAgent_v<#>/config/agent.conf`.

Hosts

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

```
host: scServer@*
```

The wildcard (`*`) tells Snap Creator to allow communications with any host. The wildcard can be replaced with a host name or IP address to restrict communications to a particular Snap Creator server.

Install the Snap Creator Agent

Perform these steps to install Snap Creator on Red Hat Enterprise Linux:

1. If Snap Creator has not already been downloaded, it can be found in the software download section of the NetApp Support site under Snap Creator Framework. Verify that the proper bit level for the OS is downloaded.
2. Copy the downloaded Snap Creator tar.gz file to /etc.

3. If a subdirectory must be made, use the mkdir command with the directory name (for example, SC).

```
mkdir SC
```

4. Use the cp command to copy the Snap Creator tar.gz file to the newly created directory. For example, from the directory where Snap Creator is downloaded, run the command to copy the Linux Snap Creator tar.gz file to the newly created SC_3.5 directory.

```
cp Snap_Creator_Community_Release_3.* /SC
```

5. If a new directory is created, the user running Snap Creator must be set as its owner. This is easiest to do before extracting the Snap Creator software, but it can be done at any point. File and folder ownership can be changed by using the chown command. When chown is used with the –R switch, ownership is also changed for files and folders under the directory. For example, to change ownership of all files and folders in the newly created directory for user snapcreator_user, run the following command:

```
chown –R snapcreator_user /SC
```

6. Use the cd command to change to the /SC directory, from which the tar.gz file will be extracted.

```
cd /SC
```

7. The tar.gz file must be unzipped before it can be extracted. To unzip the Linux Snap Creator tar.gz file, run the gunzip command.

```
gunzip Snap_Creator_Community_Release_3.*.gz
```

8. Use the tar command to extract the Linux Snap Creator .tar file. Tar is typically executed by using the –xvf switches, which tell tar to extract (-x) the file (-f) in verbose (-v) mode.

```
tar -xvf Snap_Creator_Community_Release_3.*.tar
```

9. Run the directory listing command (ls) on this directory; two directories should be displayed: scAgent<version#> and scServer<version#>, as well as the newly extracted .tar file.

Deploy and Start the Snap Creator Agent

Follow these steps to start the Snap Creator agent on open systems such as AIX, Linux, or Solaris:

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

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

Note: The Snap Creator executable should already be configured before extraction, with the proper permissions to be executed. If for some reason the profile setup command does not work, the permissions might have to be added.

```
chmod 755 snapcreator
```

2. Executing this command starts the Snap Creator setup wizard. The first page displays the EULA. At the end of the EULA, a prompt asks for EULA acceptance. At the prompt, type y and press Enter.

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

3. Confirm that the Snap Creator server should be set up. In this instance, it is the agent that must be configured, not the server. At the prompt, type n and press Enter.

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

4. Next, confirm that the Snap Creator agent should be set up. At the prompt, type y and press Enter.

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

5. This updates the environmental variables so that the Snap Creator agent scripts will work properly. The usage information for the agent appears on the page.

```
INFO: Updated NetApp Snap Creator Framework 3.5.0 Agent
INFO: To start the NetApp Snap Creator Framework 3.5.0 Agent run
"/SC_3.5/scAgent3.5.0/bin/scAgent start"
INFO: To stop the NetApp Snap Creator Framework 3.5.0 Agent run "/SC_3.5/scAgent3.5.0/bin/scAgent
stop"
```

6. Follow the onscreen information to start the Snap Creator Agent.

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

7. The default behavior of the Snap Creator Framework agent is to communicate with the Snap Creator server over port 9090. The port that Snap Creator uses for server-agent communication can be specified through the SC_AGENT_PORT environmental variable. This is a system environmental variable, so the commands to set this variable differ for each OS; check the OS documentation for information on setting environmental variables. For example, to set the SC_AGENT_PORT environment variable to port number 9091 when using Red Hat Linux, the command would include 9091.

```
export SC_AGENT_PORT=9091
```

   Note: It is recommended to use netstat or a similar tool to verify that the network port (9090 by default) is not already in use.

8. 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 will need to be opened on the virtual machine and the hypervisor.
9. It is also recommended to create a script to start and stop the Snap Creator agent automatically. An example is provided in the appendix.

Configure the Snap Creator Agent

The Snap Creator Agent can be installed in an RHEL 6 KVM host or a virtual machine. By default, the Snap Creator agent prevents any commands that are not part of the Snap Creator Framework or that are not part of a Snap Creator plug-in from being executed on remote agents. However, in some situations, the configuration file might be set up in such a way that additional commands must be executed on a remote agent. These commands include any entries that might be added to the PRE, POST, APP, or other commands.

Creator Framework. This is done by adding the command into the `agent.conf` file.

Because the Snap Creator agent blocks additional commands by default, following is the default command entry in the `agent.conf` file:

```
command:
```

Any commands or scripts that need to be given permission to run in the `agent.conf` file must be listed on a separate line. For example, to add the permission to run commands specific to affecting VM state or application state, add the following lines to the file:

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

Regular expressions can also be used to add commands to the `agent.conf` file in a more restrictive way.

Note: Although the wildcard (*) can be used to allow all commands to be executed, NetApp does not recommend this practice for security reasons.

There are numerous backup use cases; however, the following three use cases represent the vast majority that need to be addressed in a virtual environment such as RHEL 6 KVM.

Create a Backup Profile and Configuration for RHEL 6 KVM

1. Open a Web browser to the following URL:
2. http://myserver.mydomain.com:8080, replacing the host and domain with appropriate values.
3. Log in to the site with the user name and password created when the server profile was created.
4. Select “Management” > “Configurations” >, then click the “+” button under “Backup Profiles” and enter a name for the profile.
5. Under “Backup Profiles,” right-click the newly created profile and select “New Configuration” to launch the configuration wizard. Click “Next.”
6. Enter a “Config Name” and select both “Password Encryption” and “Advanced Options.” Click “Next.”
7. Select “None” for the “Plug-in type.” (The KVM plug-in does not yet support clustered Data ONTAP.) Click “Next.”
8. 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.”
9. For NetApp controller, enter the IP address of the Vserver, the login information, and select the transport protocol. If secure Web access is enabled on the NetApp controller, select HTTPS; otherwise, select HTTP. Click “Next.”
10. Select the volume(s) to be backed up by the profile. Click “Next.”
11. 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.”
12. Select “Operations Manager Alert.” Enter the IP and login information for the Operations Manager Server. Click “Next.”
13. Click “Finish.”

Configurations Before and After Snapshot Copy Creation for Red Hat KVM

1. Open a Web browser to the following URL:
2. http://myserver.mydomain.com:8080, replacing the host and domain with appropriate values.
3. Log in to the site with the user name and password created when the server profile was created.
4. Select “Management” -> “Configurations” and then select the profile and configuration to be edited.
5. Scroll to the bottom of the configuration to the “PRE/POST” section.
6. To quiesce an application and/or virtual machine prior to the Snapshot copy, click the “+” button following the “Application Quiesce Command” subsection. Enter the command(s) to be run. If there are multiple commands to be run, they will need to be listed in a 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. To resume an application and/or virtual machine after the Snapshot copy, click the “+” button following the “Application Un-Quiesce Command” subsection. Enter the command(s) to be run. If there are multiple commands to be run, they will need to be listed in a proper order.
9. 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.
10. Scroll to the top of the configuration and click the disk icon to save the updated configuration.

Crash-Consistent Backups with Snap Creator

This use case will create a Snapshot copy of a datastore without quiescing any virtual machines or applications, that is, while everything is “in flight.” The Snapshot copy can then be mirrored to another

controller for backup or archive. This is fine for capturing current state, but a restore would depend on file system logs on the guest operating system to replay properly.

Application-Consistent Backup with Snap Creator

This use case assumes that the application data is on a separate volume from the virtual machine datastore. Snap Creator first triggers the application to quiesce, then triggers the Snapshot copy on the application data volume, then triggers the application to resume. The Snapshot copy can then be mirrored to another controller for backup or archive.

Fully Consistent Snapshot Backup with Snap Creator

This use case is typically an “add-on” to the application-consistent backup. After the application is quiesced, Snap Creator tells the guest to sync its buffers (RHEL only), then tells RHEL 6 KVM to pause the virtual machine, triggers the Snapshot copies for the application volume and the virtual machine datastore, tells RHEL 6 KVM to resume the virtual machine, and finally resumes the application. The speed of the Snapshot copy as well as the speed of the virtual machine pause/resume means that this activity can occur very quickly (<5 seconds). The Snapshot copies can then be mirrored to another controller for backup or archive.

10.3 Volume SnapMirror Async with Clustered Data ONTAP

Overview

SnapMirror technology is used primarily for data protection; it enables customers to copy and back up their production data from a primary or source volume over to another volume. The primary goal of SnapMirror cross-cluster volume mirroring is to enable replication of individual volumes between independent clusters, independent of the specific network topology connecting the clusters. This intercluster volume replication provides appropriate performance and scalability when operating across clusters and over WANs, for both data transfer throughput and overall manageability of the relationships.

Clustered Data ONTAP Terminology

- **Cluster.** Consists of one or more nodes that are interconnected and managed as a single system.
- **Clustered Data ONTAP.** The Data ONTAP operating mode that supports interconnection of nodes into a cluster.
- **Cluster interconnect.** A dedicated high-speed, low-latency, private network used for communication and replication between nodes in the same cluster.
- **Data network.** The network used by clients to access data.
- **HA interconnect.** The dedicated interconnect between two nodes in one HA pair.
- **HA pair.** Two nodes configured in a pair for HA.
- **Ifgrp.** A collection of physical ports combined to create one logical port used for link aggregation.
- **Intercluster LIF.** A logical interface (LIF) used only for intercluster replication, assigned only to a node.
- **Intercluster network.** The network used for communication and replication between different clusters.
- **LIF.** A logical interface that is assigned an IP address that provides an Ethernet access point to a particular node in the cluster.
- **Management network.** The network used for administration of the cluster, the Vservers, and the nodes.
- **Node.** A single NetApp controller, one of an HA pair.

- **Port.** A physical port, such as `e0e` or `e0f`, or a logical port such as a VLAN or an interface group (`ifgrp`).
- **Vserver.** A logical storage server that provides data access to LUNs and/or a NAS namespace from one or more LIFs.

A clustered Data ONTAP system includes one or more storage nodes, as shown in Figure 28. Each storage node is a FAS storage controller. In this example, there are four FAS storage controllers in a clustered Data ONTAP cluster.

Figure 28) Cluster setup.

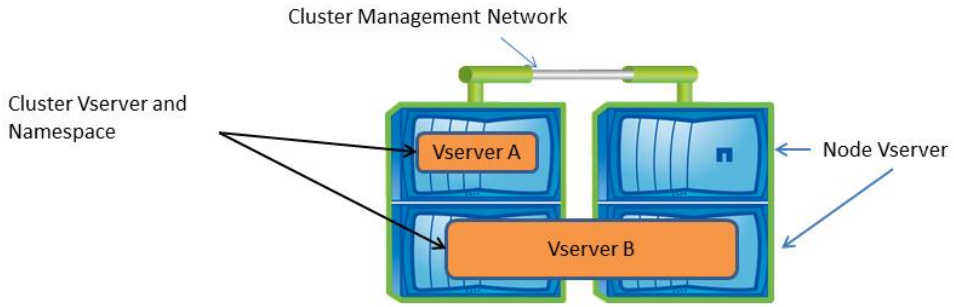


Figure 29 shows that an intracluster replication is a replication inside the cluster for data protection and data transfers within a cluster network.

Figure 29) Intracluster data protection mirrors.

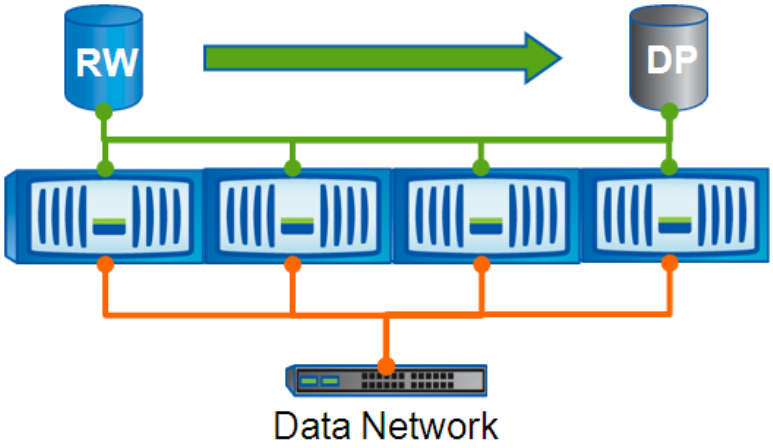


Figure 30 shows asynchronous replication for (read-only) load sharing and data transfers within a cluster.

Figure 30) Load sharing in intracluster mirrors (LS mirrors).

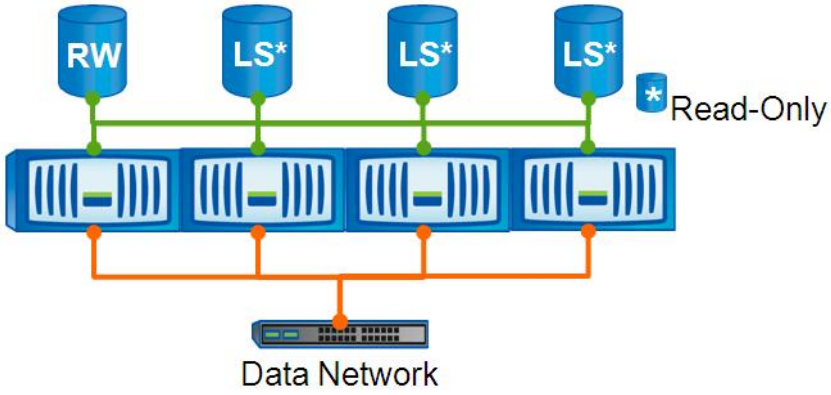
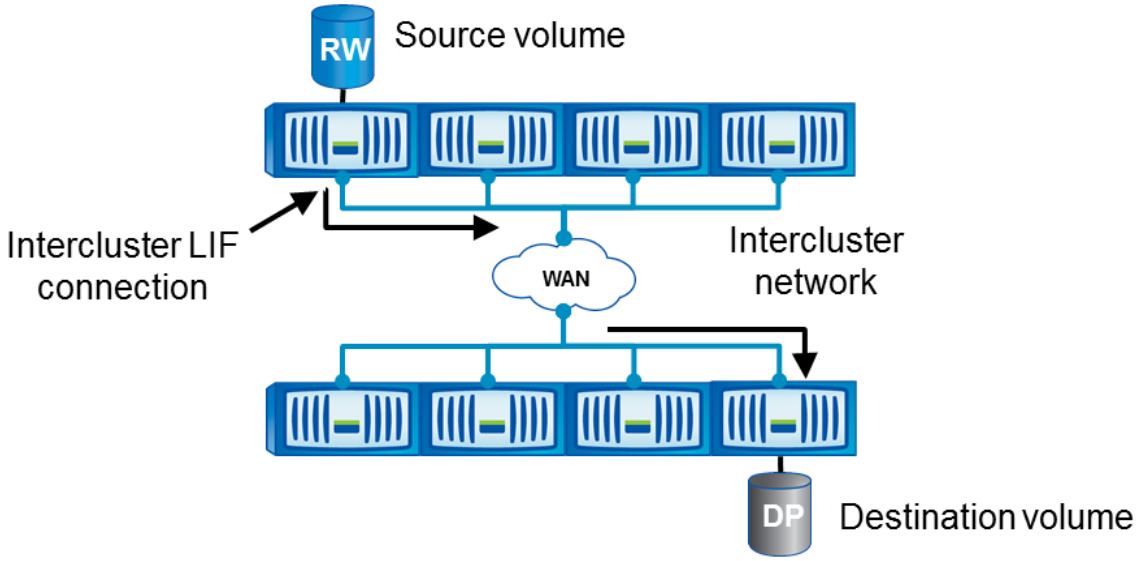


Figure 31 shows a replication between clusters for DR and data transfers between cluster networks.

Figure 31) Intercluster clustered Data ONTAP SnapMirror.



Firewall Configuration

The following ports are required for operating clustered Data ONTAP SnapMirror through a firewall:

- netapp-icmgmt: 11104
- netapp-icdata: 11105

Configure SnapMirror Async in Clustered Data ONTAP

Perform the following steps to configure and synchronize a SnapMirror relationship. Here, cluster_name_1 is the source cluster, vserver_name_1 is the source server, and volume_name_1 is the source volume. Similarly, on the destination, cluster_name_2, vserver_name_2, and volume_name_2 are the destination cluster, the destination server, and the destination volume, respectively.

1. Create the destination volume first to make sure it has the correct attributes.

```
vol create -vserver vserver_name_2 -volume volume_name_2 -aggregate aggr_name -size size_of_vol -
type DP
```

2. From the cluster containing the destination volume, run snapmirror create.
3. The destination volume must be of the type DP. The size of the destination volume must be the same as or larger than the size of the source volume.

```
snapmirror create -source-path cluster_name_1://vserver_name_1/volume_name_1 -destination-path
cluster_name_2://vserver_name_2/volume_name_2 -type DP
```

4. From the cluster containing the destination volume, run snapmirror initialize.

```
snapmirror initialize -source-path cluster_name_1://vserver_name_1/volume_name_1 -destination-
path cluster_name_2://vserver_name_2/volume_name_2
```

11 FlexClone with Clustered Data ONTAP

Overview

FlexClone volumes, LUNs, and files are writable, point-in-time copies of a parent volume, LUN, or file, respectively.

NetApp FlexClone technology gives administrators the ability to create space-efficient clones of volumes, LUNs, and files. FlexClone clones provide a fast and efficient way to perform testing and development on production data without risk to the production environment and without having to duplicate the hardware resources to support the data. FlexClone clones are also excellent for provisioning servers, for server patch, and for DR testing with products such as RHEL 6 KVM.

The main context of the following sections will highlight using FlexClone for:

- Cloning NetApp FlexVol volumes for rapid dev/test deployment
- Cloning NFS-based files for rapid deployment of virtual machines

FlexClone Clones of Volumes

The FlexClone feature for volumes provides a mechanism for creating a read/write Snapshot copy of a FlexVol volume. A Snapshot copy shares blocks with the active file system data, but the clone does not require any additional disk space. The clone occupies the same blocks as the active file system data. When new data is written, new blocks are allocated and used to store the new data. The clone and the parent volume share common data blocks, but function independently.

FlexClone clones are great for short-term use, but if clones are needed for an extended period of time, they should be split from their parent volumes. This split might become necessary because the delta between the parent and the clone can grow over time to the point where it is no longer efficient to keep the clone. Additionally, the Snapshot copy must be kept until the clone is deleted.

Make sure that enough disk space is available before splitting a clone from its parent. All data blocks that were shared by the clone and its parent will be duplicated so that they can be separately owned by and dedicated to the clone. The `vol clone split estimate` command can be used to determine how much space is required to successfully complete the split.

FlexClone volumes are created by using the `vol clone create` command. With the FlexClone license, it is also possible to clone files and LUNs by using the `clone start` command.

11.1 How FlexClone Volumes Work

FlexClone volumes can be managed similarly to regular FlexVol volumes, with a few key differences. The following list outlines some key facts about FlexClone volumes:

- A FlexClone volume is a point-in-time, writable copy of the parent volume. Changes made to the parent volume after the FlexClone volume is created are not reflected in the FlexClone volume.

- FlexClone volumes are fully functional volumes, such as the parent volume, which are managed by using the vol command.
- A FlexClone volume is always created in the same aggregate as its parent.
- A FlexClone volume is always created in the same Vserver as its parent.
- A FlexClone volume can itself be cloned to create another FlexClone volume.
- A FlexClone volume and its parent share the same disk space for any common data. This means that creating a FlexClone volume is instantaneous and requires no additional disk space (until changes are made to the FlexClone volume).
- A FlexClone volume is created with the same space guarantee as its parent. The space guarantee setting is enforced for the new FlexClone volume only if enough space is available in the containing aggregate.
- A FlexClone volume is created with the same space reservation and fractional reserve settings as its parent.
- A FlexClone volume is created with the same Snapshot schedule as its parent.
- Although a FlexClone volume exists, some operations on its parent are not allowed, such as deleting the parent volume or deleting the parent Snapshot copy.
- FlexClone volumes can be split from the parent volume. Splitting removes all restrictions on the parent volume and causes the FlexClone volume to use its own additional disk space rather than sharing space with its parent.

FlexClone Clones of LUNs and Files

FlexClone files and LUNs are writable, space-efficient clones of parent files and parent LUNs. FlexClone files and LUNs aid in space storage utilization of the physical aggregate space.

FlexClone files and LUNs use 0.4% of their size to store metadata. Clones share the data blocks of their parent files and parent LUNs and occupy negligible storage space until clients write new data either to the parent file or LUN or to the clone.

Clients can perform all file and LUN operations on both the parent and the clone entities.

Clone a Volume in Clustered Data ONTAP

A cloned volume that contains virtual machines can be used to create a duplicate of a production environment for dev/test in seconds.

To provision a FlexClone volume, complete the following steps:

1. Clone the volume.

```
volume clone create  flexclone_name  -parent-volume volume_name
```

2. To split the cloned volume from the original, run the following command:

```
volume clone split start flexclone_name
```

Clone an RHEL 6 KVM Guest in Clustered Data ONTAP

Cloning an RHEL 6 KVM guest is much faster than creating a new virtual machine from scratch. After a template has been created, follow these steps to clone that template to spin up new virtual machines.

1. Create a base image.
2. Follow the procedures on applying updates and making the image generic (see the appendix).
3. Shut down the virtual machine.
4. From the vserver prompt, issue the FlexClone command to clone the base image:

```
vol file clone create -volume volume_name -source-path /template_name.img -destination-path
/new_vm_name.img
```

Note: The source path for the file is relative to the NetApp volume that contains it. For example, if the NFS export is mounted at /images on the RHEL 6 KVM host and the file "template_name.img" is found in /images/templates, then the source path in the cloning command would be /templates/template_name.img.

5. From the RHEL 6 KVM host, use the native "virt-clone" tool to clone the XML descriptor file.

```
virt-clone -o template_name.img --original-xml=/etc/libvirt/qemu/template_name.xml -n new_vm_name
--preserve-data --file=/cmode/new_vm_name.img
```

Note: The default action of the virt-clone command will result in the cloning of the XML file and the disk file. However, by using the "--preserve-data" command along with "--file," the newly cloned virtual machine disk file is used instead. Also, the virt-clone automatically creates new MAC addresses for any network devices as well as the virtual machine UUID.

6. Make RHEL 6 KVM "aware" of the new virtual machine by executing the following command:

```
virsh define /etc/libvirt/qemu/new_vm_name.xml
```

11.2 Benefits of FlexClone Files and LUNs

Creating FlexClone files or LUNs is highly space efficient and time efficient because the cloning operation does not involve physically copying any data.

Space-efficient copies of data can be instantaneously created by using FlexClone files and LUNs in the following situations:

- When deploying, upgrading, or redeploying thousands of standardized virtual desktops or servers
- When testing video, sound, or image processing applications
- When booting servers in a server farm

FlexClone LUNs of the parent boot LUN can be created and then used to boot a server in a server farm.

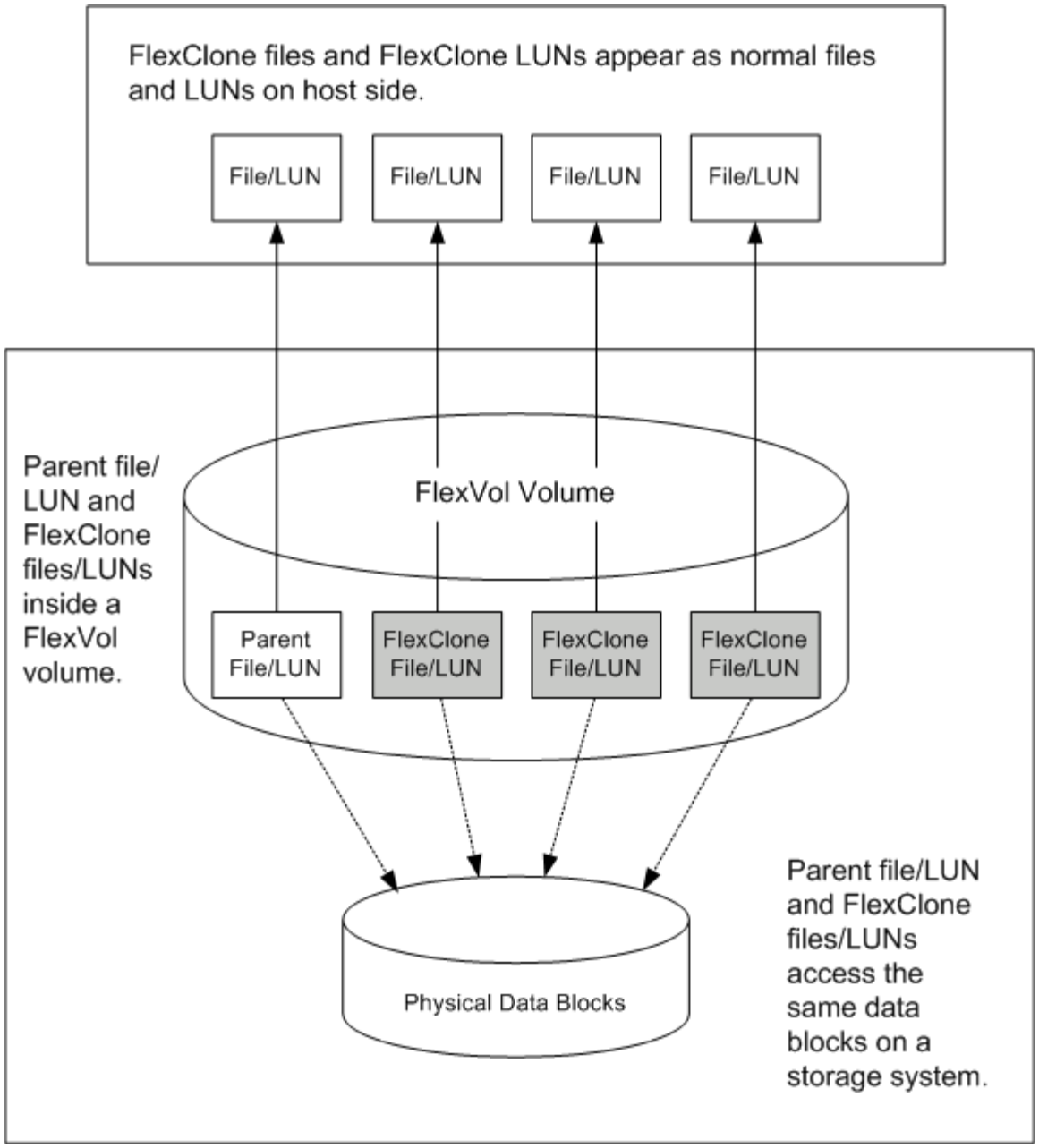
11.3 How FlexClone Files and LUNs Work

FlexClone files and LUNs share the same physical data blocks with their parents and occupy negligible space in the form of metadata. FlexClone technology can be used to create a clone of a file that is present in a FlexVol volume in a NAS environment or to clone a complete LUN without requiring an additional Snapshot copy in a SAN environment.

FlexClone technology is space efficient and time efficient because the cloning operation does not copy physical blocks of data. When new data is written to a parent or to a clone, then the entity on which new data is written starts occupying additional storage space.

Figure 32 shows the parent files or LUNs and the FlexClone files or LUNs accessing the same data blocks on the storage system. On the host side, the parent files or LUNs and the FlexClone files or LUNs appear as normal files and LUNs.

Figure 32) FlexClone overview.



The cloning operation is instantaneous and has no impact on client access to the parent file or LUN. Clients that are accessing the parent file or LUN do not experience any disruption or outage. Clients can perform the same operations on FlexClone files and LUNs as they can on standard files and LUNs.

The maximum number of FlexClone files or LUNs that can be created from a parent file or LUN without creating a physical copy of the parent entity is 32,767. If you try to create more than 32,767 clones, Data ONTAP automatically creates a new physical copy of the parent file or LUN.

The maximum logical size of all FlexClone files and LUNs in a FlexVol volume is 16TB. If you try to create a FlexClone file or LUN after the maximum size is reached, Data ONTAP automatically creates a new physical copy of the parent file or LUN.

12 Conclusion

Red Hat's implementation of KVM offers a highly configurable and high-performance virtual environment that is easy to deploy. 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.

NetApp FAS controllers offer a unified, flexible approach to data storage and data management. The flexibility allows Red Hat customers to deploy KVM-based virtualization solutions that match the needs of their business requirements. Additional NetApp tools and technologies provide the protection and storage efficiency required in any enterprise infrastructure.

Using the best practices in this guide provides a means to play, deploy, and manage data in a Red Hat Enterprise Linux 6 KVM environment. However, it is also recommended to involve local NetApp and Red Hat experts when planning such an environment in order to tailor the solution to specific application and business requirements.

Appendixes

Appendix A: HBA (FC, FCoE, and iSCSI) Configuration for SAN Boot

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)

Note: 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.

Appendix B: Ports to Allow Through Firewall

The following ports need to be allowed through any relevant firewalls (host and network) when working with RHEL 6 KVM.

Table 13 lists the allowed ports.

Table 13) Allowed ports.

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, 5353	TCP, UDP	DNS, mDNS
54321	TCP	KVM interhost communication
5900-5910	TCP	VNC consoles (optional)
32803, 662	TCP	NFS client
49152-49216	TCP	KVM migration
67, 68	TCP, UDP	DHCP
8080	TCP	Snap Creator portal
9090	TCP	Snap Creator agent
N/A	N/A	ICMP

Appendix C: Making a Template Generic

This section covers the major steps that need to be taken to take an existing boot LUN or existing virtual machine and make it usable as a template or “golden image.” These steps should be considered basic, in that there might be additional steps that need to be taken depending on the application and other configurations that need to occur per business requirements.

Removing Configuration Artifacts

The two major following areas are broken into “static configuration artifacts” and “dynamic configuration artifacts.” The explanation is this:

- Static configuration artifacts are items that require explicit reconfiguration once a clone of a template is created. A server’s host name is an example of something that needs to be configured when a cloned RHEL 6 KVM host or guest is booted up for the first time.

- Dynamic configuration artifacts are items that will automatically configure themselves. For example, if the SSH host keys are removed, they will be automatically regenerated when a cloned RHEL 6 KVM host or guest is booted up for the first time.

When looking at any other configurations that need to occur to support additional applications or business requirements, make the determination as to whether those configurations are “static” or “dynamic.” Also consider that many, if not all, of the following “static” configuration artifacts could be converted to “dynamic” by way of automation tools.

Perform the following steps:

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 business and/or application requirements, there might be additional items that need to be stripped out and/or accounted for.

2. Strip host name, gateway, and IP information (/etc/hosts, /etc/sysconfig/network, /etc/sysconfig/network-scripts/ifcfg-{eth*,br*,bond*}).
3. Strip MAC addresses from Ethernet configuration files (/etc/sysconfig/network-scripts/ifcfg-{eth*,br*,bond*}).
4. If registered to RHN (including RHN satellite and RHN proxy), strip the system ID (/etc/sysconfig/rhn/systemid).
5. Strip the iSCSI initiator name in /etc/iscsi/initiatorname.conf (can be regenerated with iscsi-name command).
6. 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* (RHEL 6 KVM boot LUN only).
7. Clear out multipath bindings (RHEL 6) (/etc/multipath/bindings).
8. Rebuild initramfs using dracut (must happen after editing the LUN WWID and multipath bindings).
9. Use the e2label tool to label the boot device, then edit /etc/fstab such that the boot device is identified by the label and not a UUID (RHEL 6) or path (RHEL 5).
10. Strip out all dynamic configuration artifacts.

Note: These items will be recreated automatically when the RHEL 6 KVM host reboots. Depending on business and/or application requirements, there might be additional items that need to be stripped out and/or accounted for.

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

Cloning A Boot LUN

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 earlier section.
2. Log in to the NetApp Vserver and execute the following command to clone the boot LUN:

```
vol file clone create -volume vol_name -source-path /lun_name -destination-path /new_lun
```

Note: The source path for the file, a LUN in this case, is relative to the volume that contains it. Because it is a LUN, it is directly under the volume, therefore represented by “/lun_name” in the preceding example.

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.

For RHEL Guest Virtual Machines

In the case of RHEL-based VMs (Versions 3, 4, and 5), the "%pre" section of the Kickstart file is used to create partitions that will enable proper file system alignment with the underlying storage. Then in the main section of the Kickstart file, the disk layout matches the aligned partitions.

The following example creates two partitions; the first for "/boot," the second for everything else. Both partitions start on sectors that align properly with the underlying storage.

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

```
%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 more information on file system alignment, see NetApp TR-3747: Best Practices for File System Alignment in Virtual Environments.

Appendix D: 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 (for Windows operating systems prior to Windows Server 2008).

Note: Kickstart cannot actually perform the Windows install, only the alignment. Although it introduces an extra step in the creation of a Windows VM template, it is still much faster than manually adjusting partitions after the VM is created.

The following example creates a single NTFS partition that takes up the entire disk, but starts on a sector that enables proper file system 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 it must be a valid 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 to the very end:

```
%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.

Appendix E: Sample Start/Stop Script for Snap Creator Portal

1. Create the file sc_server in /etc/init.d with the permissions “755.”
2. Copy the contents of the block following these steps into the file.
3. Run the following commands:

```
chkconfig –add sc_portal; chkconfig sc_portal on
#!/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/scServer3.6.0/gui
JAR_FILE=snapcreator.jar
PID_FILE=/var/lock/subsys/sc_portal

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

       touch $PID_FILE
        echo
}

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

       rm -f  $PID_FILE
        echo
}

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

exit 0
```

Appendix F: Sample Start/Stop Script for Snap Creator Agent

1. Create the file sc_agent in /etc/init.d with the permissions “755.”
2. Copy the contents of the block following these steps into the file.
3. Run the following commands:

```
chkconfig –add sc_agent; chkconfig sc_agent on
#!/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/scAgent3.6.0/bin

start()
{
        echo -n $"Starting scagent: "
        $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
        ;;
  *)
        echo $"Usage: $0 {start|stop|restart}"
        exit 1
esac

exit 0
```

References

The following references were used in this technical report:

- 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
  http://www.redhat.com/f/pdf/rhev/DOC-KVM.pdf
- Red Hat Enterprise Linux 6 Virtualization Guide
  http://docs.redhat.com/docs/en-US/Red_Hat_Enterprise_Linux/6/html/Virtualization_Administration_Guide/index.html
- Red Hat Enterprise Linux 6 Deployment Guide
  http://docs.redhat.com/docs/en-US/Red_Hat_Enterprise_Linux/6/html/Deployment_Guide/index.html
- Red Hat Enterprise Linux 6 Installation Guide
  http://docs.redhat.com/docs/en-US/Red_Hat_Enterprise_Linux/6/html/Installation_Guide/index.html
- Best Practices for File System Alignment in Virtual Environments
  http://www.netapp.com/us/library/technical-reports/TR-3747.html
- Data ONTAP Cluster-Mode Security Guidance
  http://www.netapp.com/templates/mediaView?m=tr-3964.pdf&cc=us&wid=130617723&mid=56872293
- Best Practices for Scalable SAN in Data ONTAP 8.1
  http://www.netapp.com/templates/mediaView?m=tr-4080.pdf&cc=us&wid=163988916&mid=82428326
- Data ONTAP 8.1 and 8.1.1 Operating in Cluster-Mode: An Introduction
  http://www.netapp.com/templates/mediaView?m=tr-3982.pdf&cc=us&wid=131201824&mid=57332234

Refer to the Interoperability Matrix Tool (IMT) on the NetApp Support site to validate that the exact product and feature versions described in this document are supported for your specific environment. The NetApp IMT defines the product components and versions that can be used to construct configurations that are supported by NetApp. Specific results depend on each customer's installation in accordance with published specifications.

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