Veritas Storage Foundation™ and High Availability Solutions Virtualization Guide

Linux

6.0

February 2012

Veritas Storage Foundation and High Availability Solutions Virtualization Guide

The software described in this book is furnished under a license agreement and may be used only in accordance with the terms of the agreement.

Product version: 6.0

Document version: 6.0.2

Legal Notice

THE DOCUMENTATION IS PROVIDED "AS IS" AND ALL EXPRESS OR IMPLIED CONDITIONS, REPRESENTATIONS AND WARRANTIES, INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT, ARE DISCLAIMED, EXCEPT TO THE EXTENT THAT SUCH DISCLAIMERS ARE HELD TO BE LEGALLY INVALID. SYMANTEC CORPORATION SHALL NOT BE LIABLE FOR INCIDENTAL OR CONSEQUENTIAL DAMAGES IN CONNECTION WITH THE FURNISHING, PERFORMANCE, OR USE OF THIS DOCUMENTATION. THE INFORMATION CONTAINED IN THIS DOCUMENTATION IS SUBJECT TO CHANGE WITHOUT NOTICE.

Symantec Corporation
350 Ellis Street
Mountain View, CA 94043

http://www.symantec.com

Technical Support

Symantec Technical Support maintains support centers globally. Technical Support's primary role is to respond to specific queries about product features and functionality. The Technical Support group also creates content for our online Knowledge Base. The Technical Support group works collaboratively with the other functional areas within Symantec to answer your questions in a timely fashion. For example, the Technical Support group works with Product Engineering and Symantec Security Response to provide alerting services and virus definition updates.

Symantec's support offerings include the following:

- A range of support options that give you the flexibility to select the right amount of service for any size organization
- Telephone and/or Web-based support that provides rapid response and up-to-the-minute information
- Upgrade assurance that delivers software upgrades
- Global support purchased on a regional business hours or 24 hours a day, 7 days a week basis
- Premium service offerings that include Account Management Services

For information about Symantec's support offerings, you can visit our Web site at the following URL:

www.symantec.com/business/support/index.jsp

All support services will be delivered in accordance with your support agreement and the then-current enterprise technical support policy.

Contacting Technical Support

Customers with a current support agreement may access Technical Support information at the following URL:

www.symantec.com/business/support/contact_techsupp_static.jsp

Before contacting Technical Support, make sure you have satisfied the system requirements that are listed in your product documentation. Also, you should be at the computer on which the problem occurred, in case it is necessary to replicate the problem.

When you contact Technical Support, please have the following information available:

- Product release level

- Hardware information
- Available memory, disk space, and NIC information
- Operating system
- Version and patch level
- Network topology
- Router, gateway, and IP address information
- Problem description:
  - Error messages and log files
  - Troubleshooting that was performed before contacting Symantec
  - Recent software configuration changes and network changes

Licensing and registration

If your Symantec product requires registration or a license key, access our technical support Web page at the following URL:

www.symantec.com/business/support/

Customer service

Customer service information is available at the following URL:

www.symantec.com/business/support/

Customer Service is available to assist with non-technical questions, such as the following types of issues:

- Questions regarding product licensing or serialization
- Product registration updates, such as address or name changes
- General product information (features, language availability, local dealers)
- Latest information about product updates and upgrades
- Information about upgrade assurance and support contracts
- Information about the Symantec Buying Programs
- Advice about Symantec's technical support options
- Nontechnical presales questions
- Issues that are related to CD-ROMs or manuals

Support agreement resources

If you want to contact Symantec regarding an existing support agreement, please contact the support agreement administration team for your region as follows:

Asia-Pacific and Japan	customercare_apac@symantec.com
Europe, Middle-East, and Africa	semea@symantec.com
North America and Latin America	supportsolutions@symantec.com

Documentation

Product guides are available on the media in PDF format. Make sure that you are using the current version of the documentation. The document version appears on page 2 of each guide. The latest product documentation is available on the Symantec Web site.

https://sort.symantec.com/documents

Your feedback on product documentation is important to us. Send suggestions for improvements and reports on errors or omissions. Include the title and document version (located on the second page), and chapter and section titles of the text on which you are reporting. Send feedback to:

doc_feedback@symantec.com

About Symantec Connect

Symantec Connect is the peer-to-peer technical community site for Symantec's enterprise customers. Participants can connect and share information with other product users, including creating forum posts, articles, videos, downloads, blogs and suggesting ideas, as well as interact with Symantec product teams and Technical Support. Content is rated by the community, and members receive reward points for their contributions.

http://www.symantec.com/connect/storage-management

Contents

Chapter 1

Overview

This chapter includes the following topics:

Introduction to Kernel-based Virtual Machine (KVM) technology

The Veritas Storage Foundation High Availability (SFHA) solutions can be used in Kernel-based Virtual Machine-based virtualization environments to provide advanced storage management, mission-critical clustering, and fail-over capabilities.

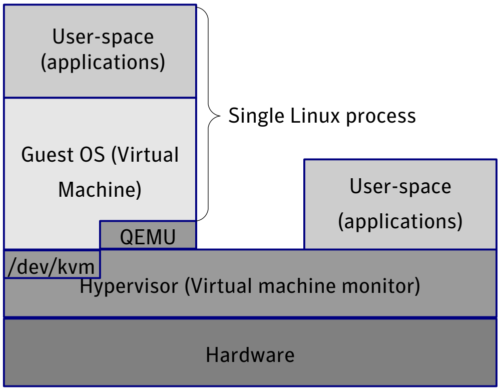
Linux Kernel-based Virtual Machine (KVM) is released by Red Hat with Red Hat Enterprise Linux (RHEL) 6, Update 1, 2 as a full virtualization solution. KVM differs from other popular alternatives like Xen and VMware in terms of operation, performance and flexibility. KVM comes as a kernel module, with a set of user space utilities to create and manage virtual machines (VM).

Kernel-based Virtual Machine technology includes the following:

- A full virtualization solution for Linux on AMD64 & Intel 64 hardware.
- Each virtualized machine or "guest" is run as a single Linux process.

- A hypervisor-independent virtualization API, “libvirt,”which provides a common generic and stable layer to securely manage virtualized guest on a host.
- A command line tool “virsh” used to manage the virtualized guests.
- A GUI for managing the virtualized guests, “virt-manager.”
- Configuration of each guest is stored in an xml file.

Figure 1-1 KVM process


User-space (applications)
Single Linux process
Guest OS (Virtual Machine)
User-space (applications)
QEMU
/dev/kvm
Hypervisor (Virtual machine monitor)
Hardware

This guide illustrates some reference configurations which can be customized to fit most implementations. An assumption is made that the reader understands the RHEL operating system, including its architecture, as well as how to configure and manage KVM virtual machines using the management software already provided by Red Hat. There is also an expectation that the user is familiar with the basic Veritas Storage Foundation and High Availability Solutions software and is well versed with its administration and management utilities. Additional details regarding RHEL, KVM, and Veritas Storage Foundation and High Availability Solutions software are available in the Additional documentation section.

See “Additional documentation” on page 43.

Kernel-based Virtual Machine Terminology

Table 1-1 KVM terminology used in this document

Term	Definition
KVM	Kernel-based Virtual Machine

Table 1-1 KVM terminology used in this document *(continued)*

Term	Definition
KVMGuest	VCS agent for managing KVM virtualized guest.
VM guest	KVM virtualized guest.
Host	The physical host on which KVM is installed.
PM	The physical machine running VCS.
VM-VM	VCS-supported configuration in which a cluster is formed between VM guests running on top of the same or different hosts.
VM-PM	VCS-supported configuration in which a cluster is formed between VM guests and physical machines.
PM-PM	VCS-supported configuration in which a cluster is formed between hosts, and which is mainly used to manage VM guests running inside them.
Bridge	A device bound to a physical network interface on the host which enables any number of guests to connect to the local network on the host. It is mapped to a physical NIC which acts as a switch to VM guests.
ApplicationHA	Symantec ApplicationHA provides monitoring capabilities for applications running inside virtual machines.

VirtIO disk drives

VirtIO is an abstraction layer for paravirtualized hypervisors in Kernel-based Virtual Machine technology. Unlike full virtualization, VirtIO requires special paravirtualized drivers running in each guest. VirtIO provides support for many devices including network devices and block (disk) devices. Using the VirtIO to export block devices to a host allows files, VxVM volumes, DMP meta-nodes, SCSI devices or any other type of block device residing on host to be presented to the guest. When SCSI devices are presented to a guest using VirtIO, in addition to simple reads and writes, SCSI commands such as SCSI inquiry commands can be performed allowing VxVM to perform deep device discovery. Running VxVM and DMP in the host and the guest provides for consistent naming of SCSI devices from the array, to the host through to the guest.

Veritas Storage Foundation and High Availability Solutions 6.0 supports VirtIO block devices with Red Hat Enterprise Linux 6.1 and greater.

VirtIO features:

- Dynamically adding devices:
  VirtIO disk devices can be both added and removed from a running guest dynamically, without the need of a reboot.

VirtIO limitations:

- Disk caching:
  When disks are exported to the guest with the cache enabled, the VxVM configuration changes may get cached on the KVM host and not be applied to the the disks. When disks are shared between more than one guest, such a configuration change is not visble from other guest systems than the one which made the change. To avoid potential configuration conplict, caching the host must be disabled (cache=no) while exporting the disks.
- SCSI Commands:
  SCSI devices which are presented as VirtIO devices to a guest support a limited subset of the SCSI command set. The KVM hypervisor blocks the restricted commands.
- PGR SCSI-3 Reservations:
  PGR SCSI-3 reservations are not supported on VirtIO devices. This limitation may be removed in future releases of Red Hat Enterprise Virtualization.
- DMP Fast Recovery with SCSI devices:
  DMP Fast Recovery bypasses the normal VirtIO read/write mechanism, performing SCSI commands directly against the device. If DMP Fast Recovery is used within the guest, caching in the host must be disabled (cache=none), to avoid data integrity issues.
- Thin Reclamation:
  Thin reclamation is not supported on VirtIO devices. The ‘WRITE-SAME’ command is blocked by the hypervisor. This limitation may be removed in future releases of Red Hat Enterprise Virtualization.
- Resizing devices:
  Red Hat Linux Enterprise 6.1 does not support online disk re-sizing of VirtIO devices. To re-size a VirtIO device the guest must be fully shut down and re-started. Support for online re-sizing of block devices is under evaluation for Red Hat Enterprise Linux 6.2.
- Maximum number of devices:
  VirtIO currently has a per-guest limitation of 32 devices. This device limitation includes all VirtIO devices, such as network interfaces and block devices. The device limitation is a result of the current VirtIO implementation where each device acts as a separate PCI device.

About Veritas Storage Foundation and High Availability Solutions products

The following sections describe the products and component software available in this Veritas Storage Foundation and High Availability Solutions release.

About Veritas Storage Foundation

Veritas Storage Foundation by Symantec includes Veritas File System (VxFS) and Veritas Volume Manager (VxVM.)

Veritas File System is a high performance journaling file system that provides easy management and quick-recovery for applications. Veritas File System delivers scalable performance, continuous availability, increased I/O throughput, and structural integrity.

Veritas Volume Manager removes the physical limitations of disk storage. You can configure, share, manage, and optimize storage I/O performance online without interrupting data availability. Veritas Volume Manager also provides easy-to-use, online storage management tools to reduce downtime.

VxFS and VxVM are included in all Veritas Storage Foundation products. If you have purchased a Veritas Storage Foundation product, VxFS and VxVM are installed and updated as part of that product. Do not install or update them as individual components.

Veritas Storage Foundation includes the dynamic multi-pathing functionality.

The Veritas Replicator option, which replicates data to remote locations over an IP network, can also be licensed with this product.

Before you install the product, read the *Veritas Storage Foundation Release Notes*.

To install the product, follow the instructions in the *Veritas Storage Foundation Installation Guide*.

About Veritas Storage Foundation High Availability

Storage Foundation High Availability includes Veritas Storage Foundation and Veritas Cluster Server. Veritas Cluster Server adds high availability functionality to Storage Foundation products.

Before you install the product, read the *Veritas Storage Foundation and High Availability Release Notes*.

To install the product, follow the instructions in the *Veritas Storage Foundation and High Availability Installation Guide*.

For HA installations, also read the *Veritas Cluster Server Release Notes*.

About Veritas Storage Foundation Cluster File System High Availability

Veritas Storage Foundation Cluster File System High Availability by Symantec extends Veritas Storage Foundation to support shared data in a storage area network (SAN) environment. Using Storage Foundation Cluster File System High Availability, multiple servers can concurrently access shared storage and files transparently to applications.

Veritas Storage Foundation Cluster File System High Availability also provides increased automation and intelligent management of availability and performance.

Storage Foundation Cluster File System High Availability includes Veritas Cluster Server, which adds high availability functionality to the product.

Veritas Replicator Option can also be licensed with this product.

Before you install the product, read the *Veritas Storage Foundation Cluster File System High Availability Release Notes*.

To install the product, follow the instructions in the *Veritas Storage Foundation Cluster File System High Availability Installation Guide*.

For information on high availability environments, read the Veritas Cluster Server documentation.

About Veritas Replicator Option

Veritas Replicator Option is an optional, separately-licensable feature.

Veritas File Replicator enables replication at the file level over IP networks. File Replicator leverages data duplication, provided by Veritas File System, to reduce the impact of replication on network resources.

Veritas Volume Replicator replicates data to remote locations over any standard IP network to provide continuous data availability.

This option is available with Storage Foundation for Oracle RAC, Storage Foundation Cluster File System, and Storage Foundation Standard and Enterprise products.

Before installing this option, read the Release Notes for the product.

To install the option, follow the instructions in the Installation Guide for the product.

About Veritas Cluster Server

Veritas Cluster Server (VCS) by Symantec is a clustering solution that provides the following benefits:

- Minimizes downtime.
- Facilitates the consolidation and the failover of servers.
- Effectively manages a wide range of applications in heterogeneous environments.

Before you install the product, read the *Veritas Cluster Server Release Notes*.

To install the product, follow the instructions in the *Veritas Cluster Server Installation Guide*.

About Veritas Cluster Server agents

Veritas agents provide high availability for specific resources and applications. Each agent manages resources of a particular type. For example, the Oracle agent manages Oracle databases. Typically, agents start, stop, and monitor resources and report state changes.

Before you install VCS agents, review the configuration guide for the agent.

In addition to the agents that are provided in this release, other agents are available through an independent Symantec offering called the Veritas Cluster Server Agent Pack. The agent pack includes the currently shipping agents and is re-released quarterly to add the new agents that are now under development.

Contact your Symantec sales representative for the following details:

- Agents that are included in the agent pack
- Agents under development
- Agents available through Symantec Consulting Services

You can download the latest agents from the Symantec Operations Readiness Tools website:

sort.symantec.com/agents

About Veritas Dynamic Multi-Pathing

Veritas Dynamic Multi-Pathing (DMP) provides multi-pathing functionality for the operating system native devices configured on the system. The product creates DMP metadevices (also known as DMP nodes) to represent all the device paths to the same physical LUN.

In earlier releases, DMP was only available as a feature of Veritas Volume Manager (VxVM). DMP supported VxVM volumes on DMP metadevices, and Veritas File System (VxFS) file systems on those volumes.

Symantec now extends DMP metadevices to support OS native logical volume managers (LVM). You can create LVM volumes and volume groups on DMP metadevices.

Note: Veritas Dynamic Multi-Pathing is a standalone product. Support for dynamic multi-pathing is also included in Veritas Storage Foundation products.

Before you install this product, review the *Veritas Dynamic Multi-Pathing Release Notes*.

To install the product, follow the instructions in the *Veritas Dynamic Multi-Pathing Installation Guide*.

About Veritas Operations Manager

Symantec recommends use of Veritas Operations Manager to manage Storage Foundation and Cluster Server environments.

Veritas Operations Manager provides a centralized management console for Veritas Storage Foundation and High Availability products. You can use Veritas Operations Manager to monitor, visualize, and manage storage resources and generate reports.

You can download Veritas Operations Manager at no charge at http://go.symantec.com/vom.

Refer to the Veritas Operations Manager documentation for installation, upgrade, and configuration instructions.

The Veritas Enterprise Administrator (VEA) console is no longer packaged with Storage Foundation products. If you want to continue using VEA, a software version is available for download from http://go.symantec.com/vcsm_download. Veritas Storage Foundation Management Server is deprecated.

If you want to manage a single cluster using Cluster Manager (Java Console), a version is available for download from http://go.symantec.com/vcsm_download. You cannot manage the new features of this release using the Java Console. Veritas Cluster Server Management Console is deprecated.

About Symantec Product Authentication Service

Symantec Product Authentication Service is a common Symantec feature. This feature validates the identities that are based on existing network operating system domains (such as NIS and NT) or private domains. The authentication service protects communication channels among Symantec application clients and services through message integrity and confidentiality services.

About Symantec ApplicationHA

Symantec ApplicationHA provides monitoring capabilities for applications running inside guest virtual machines in the KVM virtualization environment. Symantec ApplicationHA adds a layer of application awareness to the core high availability (HA) functionality offered by Veritas™ Cluster Server (VCS) in the physical host. Symantec ApplicationHA is based on VCS, and uses similar concepts such as agents, resources, and service groups. However, Symantec ApplicationHA has a lightweight server footprint that enables faster installation and configuration in virtualization environments.

Before you install the product, read the *Symantec ApplicationHA Release Notes*.

To install the product, follow the instructions in the *Symantec ApplicationHA Installation Guide*.

Storage Foundation and High Availability Solutions for the Kernel-based Virtual Machine (KVM) environment

Veritas Storage Foundation High and Availability Solutions (SFHA Solutions) products support various configurations in the Kernel-based Virtual Machine (KVM) environment. Veritas Storage Foundation High Availability Solutions 6.0 is certified on the Red Hat Enterprise Linux (RHEL) 6.1 and 6.2 distributions.

Storage Foundation and High Availability Solutions provide the following functionality for KVM guest virtual machines:

- Storage visibility
- Storage management
- Replication support
- High availability
- Cluster failover

The configurations profiled in the table below are the minimum required to achieve the storage and availability objectives listed. You can mix and match the use of SFHA Solutions products as needed to achieve the desired level of storage visibility, management, replication support, availability, and cluster failover for your KVM hosts and guest virtual machines.

Table 1-2 Storage Foundation and High Availability Solutions features in guest and host

Objective	Recommended SFHA Solutions product configuration
Storage visibility for KVM guest virtual machines	Dynamic Multi-Pathing (DMP) in the KVM guest virtual machines
Storage visibility for KVM hosts	DMP in the KVM hosts
Storage management features and replication support for KVM guest virtual machines	Storage Foundation (SF) in the KVM guest virtual machines
Advanced storage management features and replication support for KVM hosts	Storage Foundation Cluster File System (SFCFSHA) in the KVM hosts
End-to-end storage visibility in KVM hosts and guest virtual machines	DMP in the KVM host and guest virtual machines
Storage management features and replication support in the KVM guest virtual machines and storage visibility in in the KVM host	DMP in the KVM host and SF in the KVM guest virtual machines
Application monitoring and availability for KVM guest virtual machines	Symantec ApplicationHA in the KVM guest virtual machines
Virtual machine monitoring and failover for KVM hosts	Veritas Cluster Server (VCS) in the KVM hosts
Application failover for KVM guest virtual machines	VCS in the KVM guest virtual machines
Application availability and virtual machine availability	Veritas Application HA in the KVM guest virtual machine and VCS in the KVM host
Application failover across KVM guest virtual machines and physical hosts	VCS in a cluster across KVM guest virtual machines and KVM physical host machines

Each configuration has specific advantages and limitations.

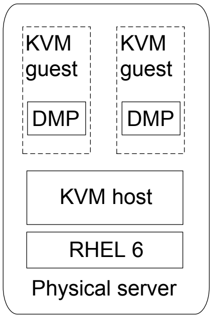
Veritas Dynamic Multi-Pathing in the KVM guest virtualized machine

Veritas Dynamic Multi-Pathing (DMP) by Symantec can provide storage visibility in KVM guest virtualized machines.

DMP in the KVM guest virtualized machine provides:

- Multi-pathing functionality for the operating system devices configured in the guest.
- DMP metadevices (also known as DMP nodes) to represent all the device paths to the same physical LUN.
- Support for enclosure-based naming.
- Support for standard array types.

Figure 1-2 Veritas Dynamic Multi-Pathing in the guest



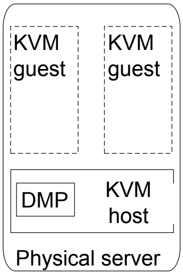
For more information on DMP features, see the *Veritas Dynamic Multi-Pathing Administrator's Guide*.

Veritas Dynamic Multi-Pathing in the KVM host

Veritas Dynamic Multi-Pathing (DMP) by Symantec can provide storage visibility in the KVM hosts. Using DMP in the KVM host enables:

- Centralized multi-pathing functionality.
- Enables active/passive array high performance failover.
- Centralized storage path management.
- Fast proactive failover.
- Event notification.

Figure 1-3 Veritas Dynamic Multi-Pathing in the KVM host



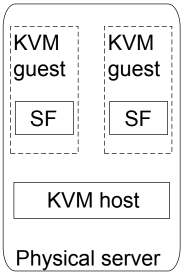
For more information on DMP features, see the *Veritas Dynamic Multi-Pathing Administrator's Guide*.

Veritas Storage Foundation in the virtualized guest machine

Veritas Storage Foundation (SF) by Symantec in the guest provides storage managment functionality for KVM guest virtual machine resources. Veritas Storage Foundation enables you to manage your KVM guest storage resources more easily by providing:

- Enhanced database performance.
- Point-in-time copy features for data back-up, recovery, and processing.
- Options for setting policies to optimize storage.
- Methods for migrating data easily and reliably.
- Replication support.

Figure 1-4 Veritas Storage Foundation in the virtualized guest machine



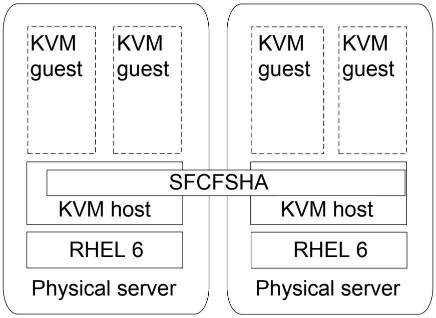
For more information on Veritas Storage Foundation features, see the *Veritas Storage™ Foundation Administrator's Guide*.

Veritas Storage Foundation Cluster File System High Availability in the KVM host

Veritas Storage Foundation Cluster File System High Availability (SFCFSHA) by Symantec provides advanced storage managment funtionality for the KVM host. SFCFSHA enables you to manage your KVM host storage resources more easily by providing:

- Enhanced database performance.
- Point-in-time copy features for data back-up, recovery, and processing.
- Options for setting policies to optimize storage.
- Methods for migrating data easily and reliably.
- Replication support.
- Highly available storage for virtual machines.
- Simplified management of virtual machines.

Figure 1-5 Veritas Storage Foundation Cluster File System High Availability in the KVM host



For more information on Storage Foundation features, see the *Veritas Storage Foundation™ Cluster File System High Availability Administrator's Guide.*

Veritas Dynamic Multi-Pathing in the KVM host and guest virtual machine

Veritas Dynamic Multi-Pathing (DMP) by Symantec can provide end-to-end storage visibility across both the KVM host and guest virtual machine.

Using DMP in the KVM guest virtualized machine provides:

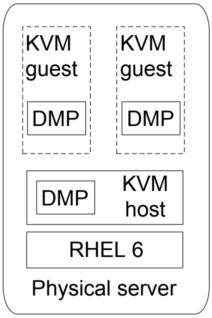
- Multi-pathing functionality for the operating system devices configured in the guest.

- DMP metadevices (also known as DMP nodes) to represent all the device paths to the same physical LUN.
- Support for enclosure-based naming.
- Support for standard array types.

Using DMP in the KVM host enables:

- Centralized multi-pathing functionality.
- Enables active/passive array high performance failover.
- Centralized storage path management.
- Fast proactive failover.
- Event notification.

Figure 1-6 Veritas Dynamic Multi-Pathing in the KVM virtualized guest and the KVM host



For more information on DMP features, see the *Veritas Dynamic Multi-Pathing Administrator's Guide*.

Veritas Storage Foundation HA in the KVM guest virtual machine and Veritas Dynamic Multi-Pathing in the KVM host

Using Veritas Storage Foundation and High Availability (SFHA) by Symantec in the guest in combination with Dynamic Multi-Pathing (DMP) in the KVM host provides provides storage managment functionality for KVM guest virtual machine resources and storage visibility in the KVM host.

Using SFHA in the KVM guest provides:

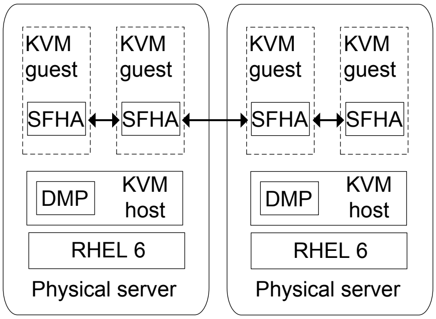
- Enhanced database performance.
- Point-in-time copy features for data back-up, recovery, and processing.

- Options for setting policies to optimize storage.
- Methods for migrating data easily and reliably.
- Replication support.
- Highly available storage for virtual machines.

Using DMP in the host provides:

- Centralized multi-pathing functionality.
- Active/passive array high performance failover.
- Centralized storage path management.
- Fast proactive failover.
- Event notification.

Figure 1-7 Veritas Storage Foundation HA in the KVM guest virtual machine and DMP in the KVM host



For more information on SFHA features, see the *Veritas Storage Foundation™ Cluster File System High Availability Administrator's Guide*.

For more information on DMP features, see the *Veritas Dynamic Multi-Pathing Administrator's Guide*.

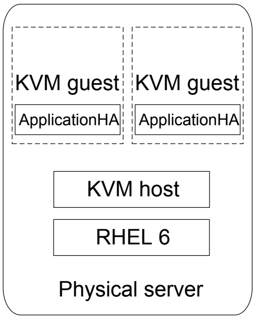
Symantec ApplicationHA in the KVM virtualized guest machine

Symantec ApplicationHA enables configuration of KVM virtualized guest resources for application failover. ApplicationHA provides the following for KVM virtualized guest machines:

- Full visibility and control over applications with the ability to start, stop, and monitor applications running inside virtual machines.
- High availability of the application as well as the virtual machine on which the application runs.

- Graded application fault-management responses such as:
  - Application restart
  - ApplicationHA-initiated, internal or soft reboot of a virtual machine
- VCS-initiated or hard reboot of virtual machine
- Standardized way to manage applications using a single interface that is integrated with the Veritas Operations Manager (VOM) dashboard.
- Specialized Application Maintenance mode, in which ApplicationHA enables you to intentionally take an application out of its purview for maintenance or troubleshooting.

Figure 1-8 Symantec ApplicationHA in the virtualized guest machine



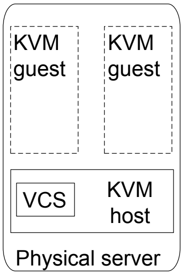
For more information on Symantec ApplicationHA features, see the *Symantec™ ApplicationHA User's Guide*.

Veritas Cluster Server in the KVM host

Veritas Cluster Server (VCS) by Symantec provides virtual machine monitoring and failover for the KVM host. VCS enables the following for KVM hosts:

- Connects multiple, independent systems into a management framework for increased availability.
- Enables nodes to cooperate at the software level to form a cluster.
- Links commodity hardware with intelligent software to provide application failover and control.
- Enables other nodes to take predefined actions when a monitored application fails, for instance to take over and bring up applications elsewhere in the cluster.

Figure 1-9 Veritas Cluster Server in the KVM host



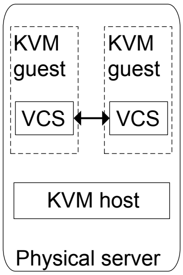
For more information on Veritas Cluster Server features, see the *Veritas Cluster Server Administrator's Guide*.

Veritas Cluster Server in the guest

Veritas Cluster Server (VCS) by Symantec enables configuration of KVM virtualized guest resources for high availability. VCS provides the following functionality for KVM virtualized guest machines:

- Connects multiple, independent systems into a management framework for increased availability.
- Enables nodes to cooperate at the software level to form a cluster.
- Links commodity hardware with intelligent software to provide application failover and control.
- Enables other nodes to take predefined actions when a monitored application fails, for instance to take over and bring up applications elsewhere in the cluster.

Figure 1-10 Veritas Cluster Server in the guest



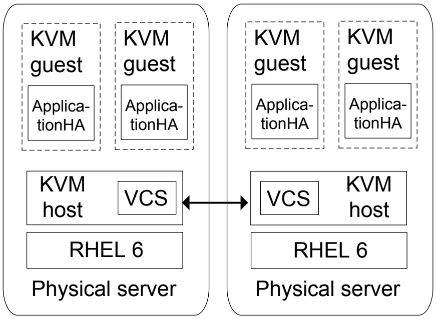
For more information on Veritas Cluster Server features, see the *Veritas Cluster Server Administrator's Guide*.

Symantec ApplicationHA in the guest and Veritas Cluster Server in the host

Using Symantec Application HA in the KVM virtualized guest in combination with Veritas Cluster Server (VCS) by Symantec in the KVM host provides an end-to-end availability solution for virtual machines and their resources.

- VCS in the host enables virtual machine availability.
- ApplicationHA in the guest enables application availability.
- ApplicationHA configured in the guest monitors the applications running inside the guest. ApplicationHA restarts the application in case of application fault.
  ApplicationHA running in the guest can notify VCS running in the host to trigger a virtual machine failover.

Figure 1-11 Veritas Cluster Server in the guest and host



For more information on Symantec ApplicationHA features, see the *Symantec ApplicationHA User's Guide*.For more information on Veritas Cluster Server features, see the *Veritas Cluster Server Administrator's Guide*.

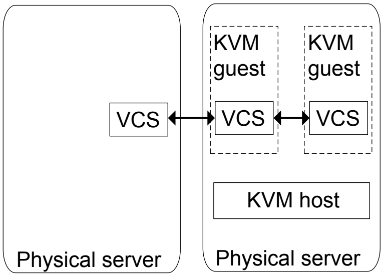
Veritas Cluster Server in a cluster across VM guests and physical machines

Using Veritas Cluster Server (VCS) by Symantec in both guests and hosts enables an integrated solution for resource management across virtual machines and physical hosts. You can create a physical to virtual cluster combining VCS in the guest together with VCS running in the host on another physical host, enabling VCS to:

- Monitor applications running within the guest
- Fail the applications over to another physical host

- Failover an application running on a physical host into a VM virtualized guest machine

Figure 1-12 Veritas Cluster Server in a cluster across guests and physical machines



For more information on Storage Foundation features, see the *Veritas Cluster Server Administrator's Guide*.

KVM use cases that are addressed by Storage Foundation and High Availability (SFHA) Solutions

Use cases where Storage Foundation and High Availability Solutions products can improve the KVM environment:

Table 1-3 SFHA Solutions for KVM use cases

KVM use case	Symantec solution	Implementation details
Server consolidation	SFHA or SCFSHA in the guest	How to run virtual machines as physical servers See "Server consolidation with KVM and Veritas Storage Foundation High Availability (SFHA) Solutions" on page 57.
Physical to virtual migration	SFHA or SFCFSHA in the guest and SF in the host	How to migrate data from physical to virtual environments safely and easily See "Physical to virtual migration with KVM and Veritas Storage Foundation High Availability (SFHA) Solutions" on page 61.

Table 1-3 SFHA Solutions for KVM use cases *(continued)*

KVM use case	Symantec solution	Implementation details
Simplified management	SFHA or SFCFSHA in the host	How to manage virtual machines using the same command set, storage namespace, and environment as in a non-virtual environment See "Simplified management with KVM and Veritas Storage Foundation High Availability (SFHA) Solutions" on page 67.
Application failover	ApplicationHA, VCS, or SFHA in the guest	How to manage application failover on virtual machines See "Veritas Cluster Server in a KVM environment: architecture summary" on page 76.
Virtual machine availability	VCS in the host	How to manage virtual machine failover See "VCS in host monitoring the Virtual Machine as a resource" on page 82.
Live Migration	SFCFSHA in the host, guest is not needed	How to use features such as instant snapshots to contain boot images and manage them from a central location in the host See "About Live Migration with KVM and Veritas Storage Foundation High Availability (SFHA) Solutions" on page 83.

Storage Foundation and High Availability Solutions products can be used in other combinations than those listed above. The configurations listed above are the minimum required to accomplish the objectives of the respective use cases.

Getting started

This chapter includes the following topics:

About setting up KVM with Veritas Storage Foundations and High Availability Solutions products

Before setting up KVM, verify your planned configuration will meet the system requirements, licensing and other considerations for installation with Veritas Storage Foundations and High Availability (SFHA) Solutions products.

- Licensing: customers running Veritas Storage Foundation or Veritas Storage Foundation Cluster File System in a KVM environment are entitled to use an unlimited number of guests on each licensed server or CPU.
- Red Hat system requirements: see Table 2-1

- Symantec product requirements: see Table 2-2
- *Release Notes*: each Veritas product contains last minute news and important details for each product, including updates to system requirements and supported software. Review the Release Notes for the latest information before you start installing the product.
  The product documentation is available on the Web at the following location:
  https://sort.symantec.com/documents

Table 2-1 Red Hat system requirements

Supported architecture	■ Intel 64 ■ AMD64
Minimum system requirements	■ 6GB free disk space ■ 2GB of RAM
Recommended system requirements	■ 6GB plus the required disk space recommended by the guest operating system per guest. For most operating systems more than 6GB of disk space is recommended ■ One processor core or hyper-thread for each virtualized CPU and one for the host ■ 2GB of RAM plus additional RAM for virtualized guests
Red Hat documentation for more information	http://www.redhat.com/virtualization/rhev/server/library/

Table 2-2 Symantec product requirements

Hardware	The hardware compatibility list contains information about supported hardware and is updated regularly. Before installing or upgrading Storage Foundation and High Availability Solutions products, review the current compatibility list to confirm the compatibility of your hardware and software. For the latest information on supported hardware, visit the following URL: http://www.symantec.com/docs/TECH170013

Table 2-2 Symantec product requirements *(continued)*

Software	■ Veritas Dynamic Multi-pathing 6.0 Used for storage visibility on KVM hosts and guest virtual machines ■ Veritas Storage Foundation 6.0 Used for storage management on KVM hosts and guest virtual machines ■ Veritas Storage Foundation HA 6.0 Used for storage management and clustering on KVM hosts and guest virtual machines ■ Storage Foundation Cluster File System High Availability 6.0 Used for storage meangement and clustering multiple KVM hosts to enable live migration of guest virtual machines ■ Veritas Cluster Server 6.0 Used for virtual machine monitoring and failover ■ Symantec ApplicationHA 6.0 Used for application monitoring and availability
Storage	■ Shared storage for holding the guest image. (VM failover) ■ Shared storage for holding the application data. (Application failover)
Networking	■ Configure the guest for communication over the public network ■ Setup virtual interfaces for private communication. ■ For detailed configuration: See put VCS topic link here ■
Documentation: see the product release notes to for the most current system requirements, limitations, and known issues:	■ *Veritas Dynamic Multi-Pathing Release Notes* ■ *Veritas Storage Foundation Release Notes* ■ *Veritas Storage Foundation HA Release Notes* ■ *Veritas Storage Foundation for Cluster Server HA Release Notes* ■ *Veritas Cluster Server HA Release Notes* ■ *Symantec ApplicationHA Release Notes* ■ Symantec Operations Readiness Tools: https://sort.symantec.com/documents ■ Storage Foundation DocCentral Site: http://sfdoccentral.symantec.com/

Table 2-3 VCS system requirements for KVM-supported configurations

VCS version	6.0
Supported OS version in host	RHEL 6, Update 1, 2

Table 2-3 VCS system requirements for KVM-supported configurations *(continued)*

Supported OS in VM guest	RHEL 5 Update 4, Update5, Update 6, Update 7 RHEL 6, Update 1, 2
Hardware requirement	Full virtualization-enabled CPU

See “Additional documentation” on page 43.

Limitations and unsupported KVM features

DiskReservation agent cannot work with disks exported over a virtio bus.

For more information on limitations and known issues, refer to VCS 6.0 Release Notes for Linux.

For KVM related limitations, refer to the Red Hat release notes.

Creating and Launching a KVM

KVM is available as part of Red Hat Enterprise Linux (RHEL 6, Update 1, 2) and also as a separate bare-metal stand-alone hypervisor, Red Hat Enterprise Virtualization Hypervisor (RHEV-H). Management for KVM is either provided through the Red Hat Enterprise Virtualization Manager (RHEV-M) or through separate RPMs that can be downloaded into the standard RHEL 6, Update 1, 2 installations. This installation and usage guide is focused on using KVM based virtualization as provided through the RHEL 6, Update 1, 2 distributions.

The virt-manager tool provides a very simple, easy-to-use and intuitive GUI interface for all virtual machine operations, along with virt-viewer. A command line alternative, `virsh`, also provides a shell that can be used to create and manage virtual machines using a rich set of commands. The features provided by these tools include taking snapshots of virtual machines, creating virtual networks and live migration of virtual machines to another KVM host.

Once you have configured the required hardware setup:

- Install KVM on the target systems.
- Create and launch the required KVM virtual machines.
- Proceed to install the required SFHA product on the guest or host:
  See “Installing and configuring storage solutions in the KVM guest” on page 35.
  See “Installing and configuring storage solutions in the host” on page 37.

See "Installing and configuring Veritas Cluster Server for virtual machine and application availability " on page 39.

See "Installing and configuring ApplicationHA for application availability" on page 42.

For RHEL 6, Update 1, 2 installation information:

http://www.redhat.com/virtualization/rhev/server/library/

For a full set of features and capabilities, please refer to the Red Hat documentation.

See "Additional documentation" on page 43.

Setting up the VM guest

Following is a high-level overview of the steps required for setting up KVMs. For detailed instructions, refer to *Red Hat Enterprise Linux Virtualization Guide*.

1. Before creating VM guests, ensure that CPU and memory resources are available to create VM guests on all nodes in the cluster.
2. Make sure that the required KVM packages are installed on the hosts.
3. Make sure that the service libvirtd is running on the hosts where VM guests are to be created.
4. Create VM guests. For network configuration, refer to the *Network configuration for VM-VM cluster* in Appendix A..
5. Install the operating system in the VM guests.
6. Repeat the above steps for all VM guests that you want to be a part of the cluster.
7. Install VCS on all the VM guests. For information about installing VCS, refer to the *Veritas Cluster Server Installation Guide*.
8. Configure the VCS resources that you want VCS to manage. For more information, refer to the VCS documentation.

See "Network configuration for VM-VM cluster" on page 55.

Installing and configuring storage solutions in the KVM guest

To set up a virtual guest environment with SFHA solutions after installing KVM:

- Install the Storage Foundation and High Availability (SFHA) Solutions product on the required KVM guest virtual machines.
- Configure the SFHA Solutions product on the required KVM guest virtual machines.
- For SFHA Solutions product installation information:
  - *Veritas Dynamic Multi-Pathing Installation Guide*
  - *Veritas Storage Foundation Installation Guide*
  - *Veritas Storage Foundation High Availability Installation Guide*
  - *Veritas Storage Foundation for Cluster Server High Availability Installation Guide*
  - See “Additional documentation” on page 43.
- - The Installation and configuration of VCS inside KVM guest is similar to that of the physical system.
  - No additional VCS configuration is required to make it work inside the guest.

The steps above apply for the following guest configurations:

- Dynamic Multi-pathing in the guest
- Storage Foundation in the guest
- Storage Foundation High Availability in the guest
- Storage Foundation for Cluster File System in the guest

Figure 2-1 Dynamic Multi-pathing in the guest

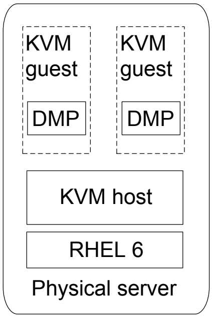


Figure 2-2 Storage Foundation in the guest

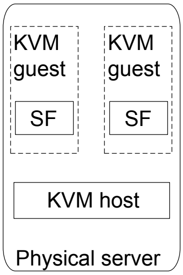


Figure 2-3 Storage Foundation High Availability in the guest

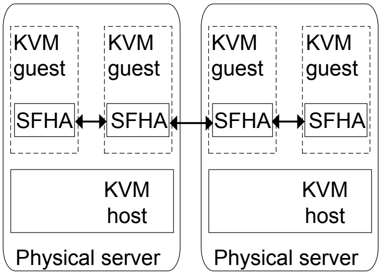
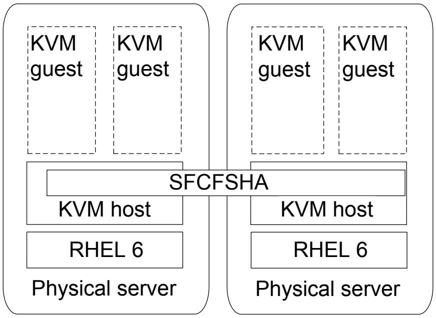


Figure 2-4 Storage Foundation for Cluster File System in the guest



Installing and configuring storage solutions in the host

To set up a virtual host environment with Storage Foundation and High Availability (SFHA) Solutions after installing KVM:

- Install the SFHA Solutions product on the required physical machines.
- Configure the SFHA Solutions product on the required physical machines.

- For SFHA Solutions product installation information:
  - *Veritas Dynamic Multi-Pathing Installation Guide*
  - *Veritas Storage Foundation Installation Guide*
  - *Veritas Storage Foundation High Availability Installation Guide*
  - *Veritas Storage Foundation for Cluster Server High Availability Installation Guide*
  - See "Additional documentation" on page 43.

The steps above apply for the following host configurations:

- Dynamic Multi-pathing in the host
- Storage Foundation for Cluster file System in the host

Figure 2-5 Dynamic Multi-pathing in the host

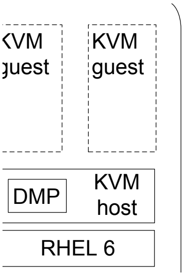
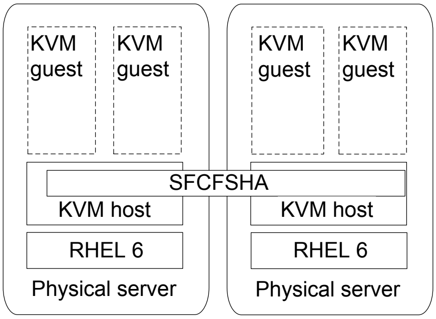


Figure 2-6 Storage Foundation for Cluster File System High Availability in the host



Installing and configuring Veritas Cluster Server for virtual machine and application availability

To set up a virtual guest environment with Veritas Cluster Server (VCS) after installing KVM:

- Install VCS.
- Configure VCS.
- The installation and configuration of VCS inside a KVM guest is similar to that of the physical system.
- No additional VCS configuration is required to make it work inside the guest.
- For installation information:
  Veritas Cluster Server Installation Guide
  See "Additional documentation" on page 43.

The steps above apply for the following guest configurations:

- VCS in the KVM host
- VCS in the KVM guest virtual machine
- VCS in the KVM host and ApplicationHA in the KVM guest virtual machine
- VCS in a cluster across guests and physical machines

Figure 2-7 VCS in the KVM host

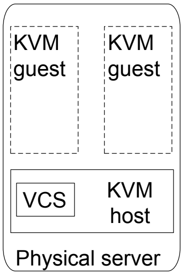


Figure 2-8 VCS in the KVM guest virtual machine

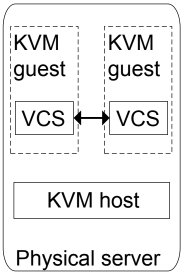


Figure 2-9 VCS in the KVM host and ApplicationHA in the KVM guest virtual machine

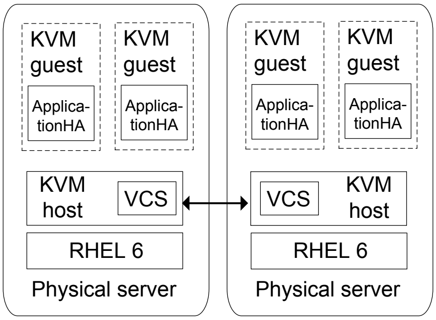


Figure 2-10 VCS in a cluster across guests and physical machines

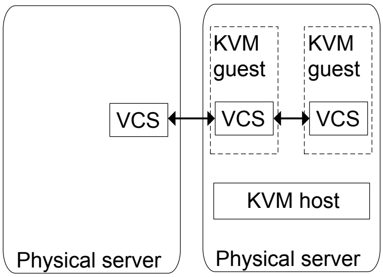


Table 2-4 VCS system requirements for the KVM-supported configurations

VCS version	6.0
Supported OS version in host	RHEL 6, Update 1, 2
Supported OS in VM guest	RHEL 5 Update 4, 5, 6, 7 RHEL 6, Update 1, 2

Table 2-4 VCS system requirements for the KVM-supported configurations *(continued)*

Hardware requirement	Full virtualization-enabled CPU

How Veritas Cluster Server (VCS) manages KVM guests

High-level overview of how VCS manages virtual machine (VM) guests.

- Physical machines form a cluster with VCS installed on them.
  For information about installing VCS, see the *Veritas Cluster Server Installation Guide*.
- CPU and memory resources are made available to create VM guests on all nodes in the cluster.
- VCS is installed on all the hosts to manage the VM guest.
- The operating system is installed on the VM guest on any one host.

  Note: The VM guest can be created on an image file or on a shared raw disk, provided the disk names are persistent across all the physical hosts.

- The VM guest is configured as a KVMGuest resource in VCS.

For detailed instructions on creating and configuring a VM guest, see the *Installation* section in the *Red Hat Enterprise Linux Virtualization Guide*.

To configure a VM guest for a physical machine to physical machine (PM-PM) configuration, the following conditions apply:

- You must configure a VM guest on one node with operating system installed on a shared storage accessible to all the VCS cluster nodes.
- Ensure that the image file resides on the shared storage so that the virtual machines can fail over across cluster nodes.
- You can configure the first VM guest using the standard installation procedure.
  See "Installing and configuring storage solutions in the KVM guest" on page 35.

Bundled agents are included with VCS for managing many applications. The KVMGuest agent is included and can be used to manage and provide high availability for KVM guests. For information on KVMGuest agent attributes, resource dependency and agent function, refer to the *Veritas Cluster Server Bundled Agents Reference Guide*.

Installing and configuring ApplicationHA for application availability

To set up a virtual guest environment with Symantec ApplicationHA after installing KVM:

- Install ApplicationHA.
- Configure ApplicationHA.
- For installation information:
  Symantec ApplicationHA Installation Guide
  See “Additional documentation” on page 43.

The steps above apply for the following guest configurations:

- ApplicationHA in the KVM guest virtual machine
- VCS in the KVM host and ApplicationHA in the KVM guest virtual machine

Figure 2-11 ApplicationHA in the KVM guest virtual machine

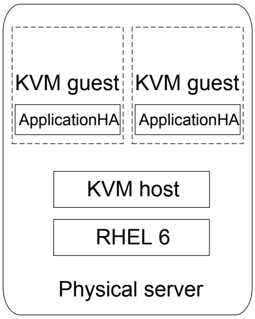
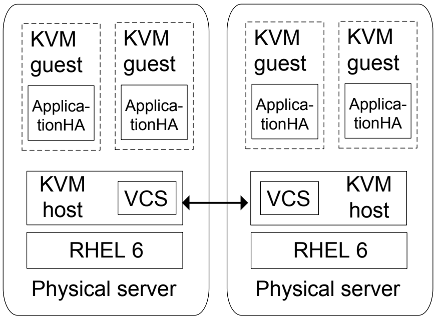


Figure 2-12 VCS in the KVM host and ApplicationHA in the KVM guest virtual machine



Additional documentation

For Red Hat documentation:

- RHEL:
  http://www.redhat.com/virtualization/rhev/server/library/
- KVM Whitepaper:
  http://www.redhat.com/f/pdf/rhev/DOC-KVM.pdf
- KVM Open source Project Site:
  http://www.linux-kvm.org/page/Main_Page

For Symantec product installation and configuration information:

- *Veritas Dynamic Multi-Pathing Installation Guide*
- *Veritas Storage Foundation Installation Guide*
- *Veritas Storage Foundation High Availability Installation Guide*
- *Veritas Storage Foundation for Cluster Server High Availability Installation Guide*
- *Veritas Cluster Server High Availability Installation Guide*
- *Veritas Cluster Server Bundled Agents Reference Guide*
- *Symantec ApplicationHA Installation Guide*

To locate Symantec product guides:

- Symantec Operations Readiness Tools:
  https://sort.symantec.com/documents
- Storage Foundation DocCentral Site:
  http://sfdoccentral.symantec.com/

Chapter 3

Configuring KVM resources

This chapter includes the following topics:

Configuring KVM resources with Veritas Storage Foundation High Availability (SFHA) Solutions

After installing KVM, SFHA Solutions products, and creating the virtual machines, you can configure your KVM resources to optimize your environment. Configuration options depend on the SFHA Solutions products you are using:

- You can optimize your storage for visibility and management convenience if you are using one of the following products in your KVM guests or hosts:
  - Veritas Dynamic Multi-Pathing (DMP)
  - Veritas Storage Foundation (SF)
  - Veritas Storage Foundation HA (SFHA)
  - Veritas Storage Foundation Cluster File System HA (SFCFSHA)

  See "Consistent storage mapping in the KVM environment" on page 46.

- You can optimize your network to make your KVM resources highly available if you are using one of the following products in your KVM guests or hosts:
  - Veritas Cluster Server (VCS)
  - Veritas Storage Foundation HA (SFHA)
  - Veritas Storage Foundation Cluster File System HA (SFCFSHA)

Veritas Storage Foundation and High Availability Solutions products enable you to configure networking for:

- Application failover
- Virtual machine availability

Consistent storage mapping in the KVM environment

Veritas Storage Foundation and High Availability Solutions enable you to map and manage your storage more efficiently whether you have a guest or host solution. Managing storage in the KVM environment requires consistent mapping. Storage which is presented to the guest either using the para-virtualized VirtIO drivers, or the fully virtualized IDE emulation, needs to be mapped from the host to the guest. Due to the volatile nature of the device naming used in Linux, care must be taken when mapping storage from the host to the guest. In Linux, the device names are based on enumeration order which can change when systems are rebooted.

Consistent mapping can be achieved by using:

- DMP meta-device
  See "Mapping DMP meta-devices" on page 47.
- Mapping devices using device ID
  See "Consistent naming across KVM hosts" on page 48.
  See "Mapping devices using device identification" on page 49.
- Mapping devices using paths
  See "Mapping devices using paths" on page 49.
- Mapping devices using volumes
  Mapping can be achieved by using Veritas Volume Manager volumes (VXVM volumes).
  For more about mapping a VxVM volume to a guest:
  See "Simplified management with KVM and Veritas Storage Foundation High Availability (SFHA) Solutions" on page 67.
- Linux `udev` device sym-links.

Avoid using disk labels when mapping storage to a guest. Disk labels can be modified by a guest and are not guaranteed.

In clustered environments, Active-Passive DMP devices cannot be mapped directly to a guest.

Non-persistent mappings can be made using 'virsh attach-device'. The non-persistent mappings can be made persistent by redefining the KVM guests using 'virsh dumpxml *domain*' followed by 'virsh define *domain*'. Alternatively, persistent mappings can be created when a virtual machine is rebooted, these non-persistent mappings are lost. Persistent mappings can be created on the host using either 'virt-manager' or by modifying the guests XML configuration using 'virsh edit <domain>'.

In the following examples,using 'virsh attach-disk'.

Mapping DMP meta-devices

Consistent mapping can be achieved from the host to the guest by using the Persistent Naming feature of DMP.

Running DMP in the host has other practical benefits:

- Multi-path device can be exported as a single device. This makes managing mapping easier, and helps alleviate the 32 device limit, imposed by the VirtIO driver.
- Path failover can be managed efficiently in the host, taking full advantage of the Event Source daemon to proactively monitor paths.
- When Veritas Storage Foundation and High Availability Solutions products are installed in the guest, the 'Persistent Naming' feature provides consistent naming of supported devices from the guest through the host to the array. The User Defined Names feature, or UDN, allows DMP virtual devices to have custom assigned names.

To map a DMP meta-device to a guest

1. Map the device to the guest. In this example the dmp device *xiv0_8614* is mapped to *guest_1*.

   ```
   # virsh attach-disk guest_1 /dev/vx/dmp/xiv0_8614 vdb
   ```

2. The mapping can be made persistent by redefining the guest.

   ```
   # virsh dumpxml guest_1 > /tmp/guest_1.xml
   # virsh define /tmp/guest_1.xml
   ```

Consistent naming across KVM hosts

While enclosure based naming (EBN) provides persistent naming for a single node, it does not guarantee consistent naming across nodes in a cluster. The User Defined Names (UDN) feature of DMP allows DMP devices to be given both persistent and consistent names across multiple hosts. When using User Defined Names, a template file is created on a host, which maps the serial number of the enclosure and device to unique device name. User Defined Names can be manually selected, which can help make mappings easier to manage.

To create consistent naming across hosts

1 Create the User Defined Names template file.

```
# /etc/vx/bin/vxgetdmpnames enclosure=3pardata0 > /tmp/user_defined_names
# cat /tmp/user_defined_names
enclosure vendor=3PARdat product=VV serial=1628 name=3pardata0
 dmpnode serial=2AC00008065C name=3pardata0_1
 dmpnode serial=2AC00002065C name=3pardata0_2
 dmpnode serial=2AC00003065C name=3pardata0_3
 dmpnode serial=2AC00004065C name=3pardata0_4
```

2 If necessary, rename the devices. In this example, the DMP devices are named using the name of the guest they are to be mapped to.

```
# cat /dmp/user_defined_names
enclosure vendor=3PARdat product=VV serial=1628 name=3pardata0
         dmpnode serial=2AC00008065C name=guest1_1
         dmpnode serial=2AC00002065C name=guest1_2
         dmpnode serial=2AC00003065C name=guest2_1
         dmpnode serial=2AC00004065C name=guest2_2
```

3 Apply the user-defined-names to this node, and all other hosts.

```
# vxddladm assign names file=/tmp/user_defined_names
```

4 Verify the user defined names have been applied.

```
# vxdmpadm getdmpnode enclosure=3pardata0
NAME        STATE     ENCLR-TYPE  PATHS  ENBL  DSBL  ENCLR-NAME
====================================================================
guest_1_1   ENABLED   3PARDATA    2      2     0     3pardata0
guest_1_2   ENABLED   3PARDATA    2      2     0     3pardata0
guest_2_1   ENABLED   3PARDATA    2      2     0     3pardata0
guest_2_2   ENABLED   3PARDATA    2      2     0     3pardata0
```

Mapping devices using device identification

Mapping can be achieved using device ID: /dev/disk/by-id/

These sym-links use the persistent properties of a device such as the device serial number. If a device has two or more paths, the sym-link will point to a single path or meta-device, which has the greatest weight. If the Linux Device Mapper multi-path driver is configured, the device mapper meta-device will have a greater 'weight' and the sym-link will point to the device mapper device.

The persistent device name can be obtained by mapping a device to a guest. In this example the device *sdb* is mapped to *guest_2*.

To map a device to a Guest

1 Identify the device(s) to map to the guest.

```
# udevadm info -q symlink --name sdb | cut -d\  -f 2
disk/by-id/scsi-200173800013420cd
```

2 Map the device to the guest using the path using the device ID.

```
# virsh attach-disk guest_2 /dev/disk/by-id/scsi-200173800013420cd
```

3 The mapping can be made persistent by re-defining the guest.

```
# virsh dumpxml guest_2 > /tmp/guest_2.xml
# virsh define /tmp/guest_2.xml
```

4

Mapping devices using paths

Mapping can be achieved using device ID: /dev/disk/by-path/

These links use the persistent properties of a path. For fibre channel devices, the sym-link name is composed of the bus identifier, the WWN of the target, followed by the LUN identifier. A device will have an entry for each path to the device. In environments where multi-pathing is to be performed in the guest, make a mapping for each path for the device.

In the following example both paths to device *sdd* are mapped to *guest_3*.

To map a path to a guest

1. Identify the devices to map to the guest. Obtain the device IDs.

   ```
   # udevadm info -q symlink --name sdd | cut -d\  -f 3
   disk/by-id/scsi-200173800013420cd
   ```

   In multi-path environments the device ID can be used to find all paths to the device.

   ```
   # udevadm info --export-db |grep disk/by-id/scsi-200173800013420cd\  \
   | cut -d\  -f 4
   /dev/disk/by-path/pci-0000:0b:00.0-fc-0x5001738001340160:0x000000
   /dev/disk/by-path/pci-0000:0c:00.0-fc-0x5001738001340161:0x000000
   ```

2. Map the device to the guest using the path using the device path.

   ```
   # virsh attach-disk guest_3  \
   /dev/disk/by-path/pci-0000:0b:00.0-fc-0x5001738001340160:0x000000 vdb
   Disk attached successfully
   # virsh attach-disk guest_3  \
   /dev/disk/by-path/pci-0000:0c:00.0-fc-0x5001738001340161:0x000000 vdc
   Disk attached successfully
   ```

3. Make the mapping persistent by re-defining the guest.

   ```
   # virsh dumpxml guest_3 > /tmp/guest_3.xml
   # virsh define /tmp/guest_3.xml
   ```

Resizing VirtIO devices

Red Hat Linux Enterprise (RHEL) 6, Update 1 does not support online disk resizing of VirtIO devices. To resize a VirtIO device the guest must be fully shut down and restarted. Support for online resizing of block dvices is under evaluation for RHEL 6, Update 2.

You can use the following methods to resize the devices.

To grow VirtIO devices

1. Grow the storage.

   - If the storage device is a VxVM Volume, re-size the volume.

- If the storage device is a LUN from a storage array, re-size the device on the array.

2 Update the size of the disk device in the host.

- Stop all virtual machines using the storage device.
- If the device is a LUN from a storage array, issue 'blockdev --rereadpt <device>' to update the size of the device.
- Restart the virtual machines.

3 Update the size of the storage device in the guest .

- If VxVM is managing the storage in the guest, use `vxdisk resize`.
- If VxVM is not managing the storage in the guest, see the appropriate documentation.

To shrink guest disk devices

1 Update the size of the disk device in the guest.

- If VxVM is managing the device in the guest, if necessary, first use the `vxresize` utility to shrink any file systems and volumes which are using the device. Use `vxdisk resize` *`access_name`* `length=`*`new_size`* to update the size of the public region of the device.
- If VxVM is not managing the storage in the guest, see the appropriate documentation.

2 Shrink the storage in the guest.

- If the device is a VxVM volume, shrink the volume with the vxassist utility.
- If the device is a LUN from a storage array, shrink the device on storage array.

3 Update the size of the disk device in the host.

- Stop the guests which are using the devices.
- If the device is a LUN from a storage array, use the `blockdev --rereadpt device` command.

4 Start the guests.

Bridge network configuration

The bridge network configuration can be performed in two parts:

- Configuring the host network

The libvirtd service creates a default bridge virbr0 which is a natted private network. It allocates private IPs from the network 192.168.122.0, to the guests using virbr0 for networking. If the KVMGuests are required to communicate on the public network of the host machines, then a bridge must be configured. This bridge can be created using the following steps:

- Configuring KVMGuest network
  Guest network configuration differs from the standard guest configuration by a single step. Use the following steps to configure the KVMGuest network:

To configure the host network

1. Create a new interface file with the name `ifcfg-br0` in `/etc/sysconfig/network-scripts/` location where all the other interface configuration files are present. Its contents are as follows:

   ```
   DEVICE=br0
   Type=Bridge
   BOOTPROTO=dhcp
   ONBOOT=yes
   ```

2. Add the physical interface to the bridge using the following command.

   ```
   # brctl addif eth0 br0
   ```

   This adds the physical interface that the KVMGuests shares with the br0 bridge created in the previous step.

3. Verify that your eth0 was added to the br0 bridge using the `brctl show` command.

   ```
   # brctl show
   ```

   The output must look similar to the following:

   ```
   bridge name     bridge id               STP enabled     interfaces
   virbr0          8000.000000000000       yes
   br0             8000.0019b97ec863       yes             eth0
   ```

4. The eth0 network configuration must be changed. The ifcfg-eth0 script is already present.

5 Edit the file and add a line **BRIDGE=br0**, so that the contents of the configuration file look like the following example:

```
DEVICE=eth0
BRIDGE=br0
BOOTPROTO=none
HWADDR=00:19:b9:7e:c8:63
ONBOOT=yes
TYPE=Ethernet
USERCTL=no
IPV6INIT=no
PEERDNS=yes
NM_CONTROLLED=no
```

6 Restart the network services to bring all the network configuration changes into effect.

To configure the KVM guest network

1 Begin the standard guest configuration.

2 On the **Network** page, specify the networking method, select **Shared physical device**, and from the **Device** list, select the respective physical interface

3 Start the KVMGuest and make sure it connects to the local network of the host.

4 Run the `brctl show` command to verify that the bridge br0 is bounded to eth0 and vnet1 on the guest network.

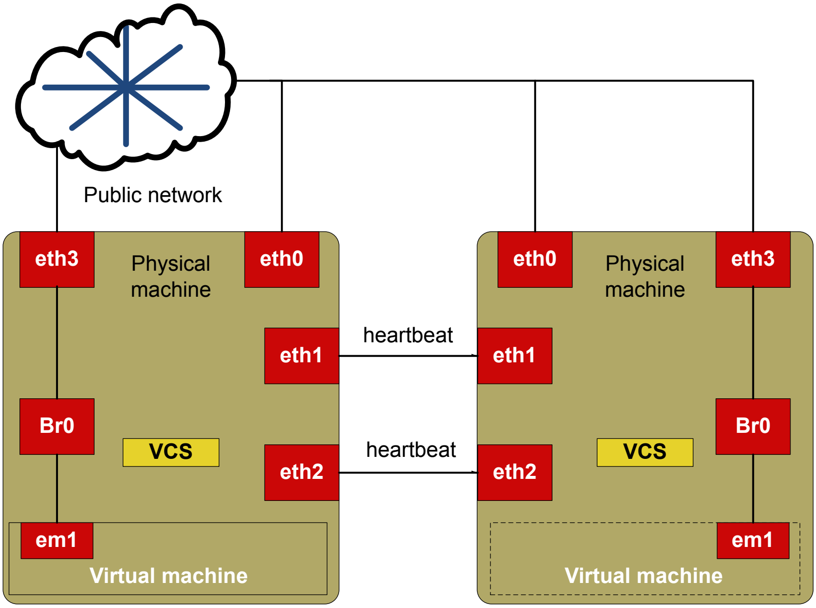
For example, the command must display an output similar to the following:

```
bridge name     bridge id               STP enabled     interfaces
virbr0          8000.000000000000       yes
br0             8000.0019b97ec863       yes             eth0
                                                        vnet1
```

Network configuration for VCS cluster across physical machines (PM-PM)

The network configuration and storage of the hosts is similar to the VCS cluster configurations. For configuration-related information, refer to the *Veritas Cluster Server Installation Guide*. However, you must set up a private link and a shared storage between the physical hosts on which the VM guests are configured.

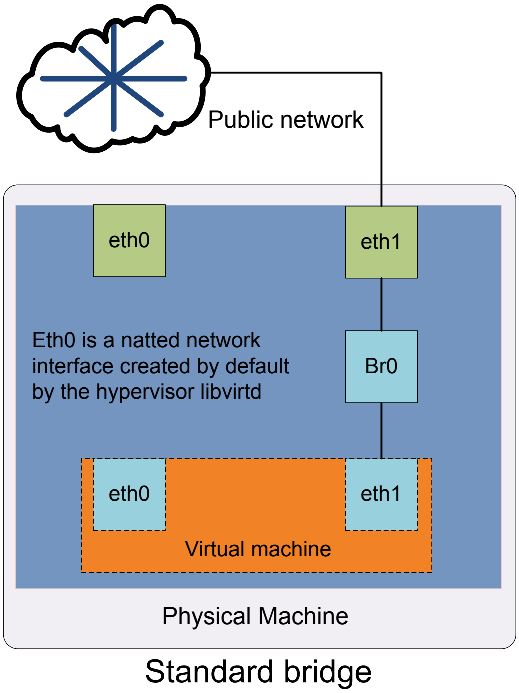
Figure 3-1



Standard bridge configuration

The standard bridge configuration is a generic network configuration for bridge networking.

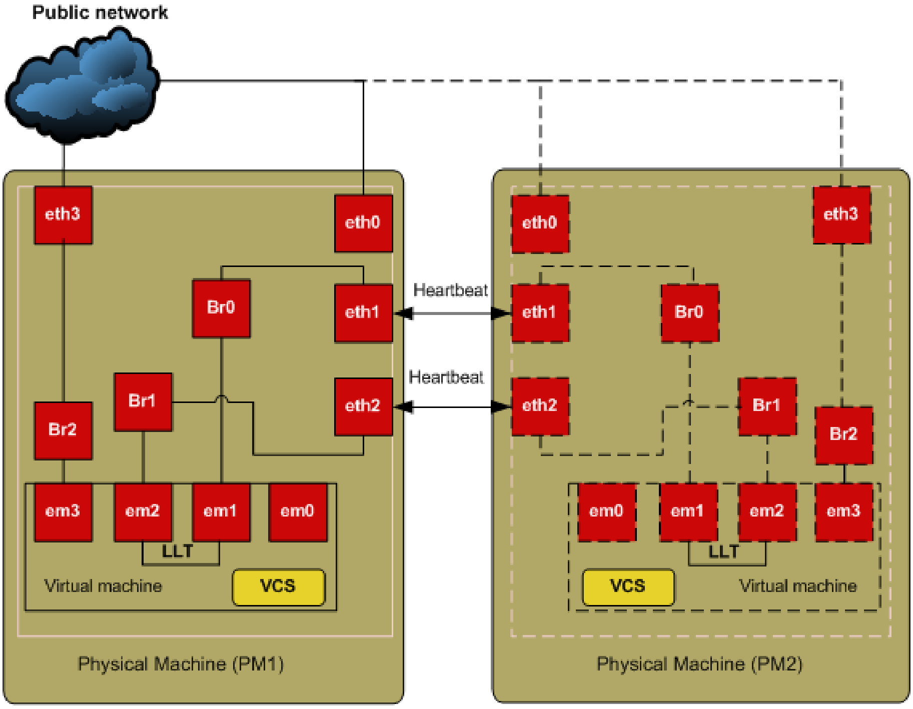
Figure 3-2 Standard bridge configuration


Public network
eth0
eth1
Eth0 is a natted network interface created by default by the hypervisor libvirtd
Br0
eth0
eth1
Virtual machine
Physical Machine
Standard bridge

Network configuration for VM-VM cluster

To manage the VCS cluster between the virtual machines, you must configure the network and storage for the cluster. The setup details for network and storage configurations are explained in the subsequent sections. Figure 3-3 shows a cluster setup between two VM guests running on two different hosts.

Figure 3-3 Network configuration for VM- VM cluster



em0 is a default NATed network
interface created by the KVM hypervisor.

Bridge with Heartbeat

Chapter 4

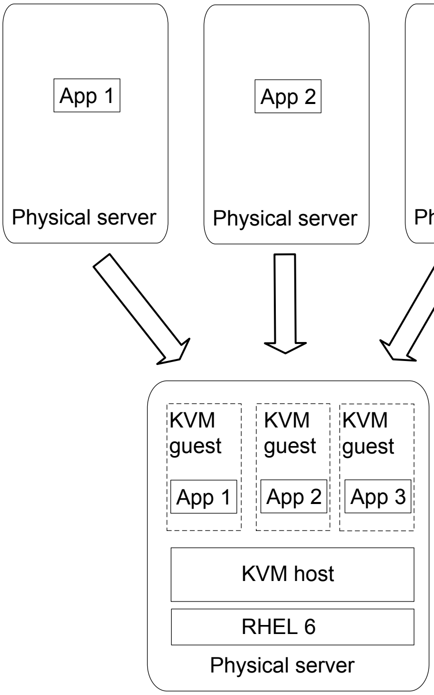
Server consolidation

This chapter includes the following topics:

Server consolidation with KVM and Veritas Storage Foundation High Availability (SFHA) Solutions

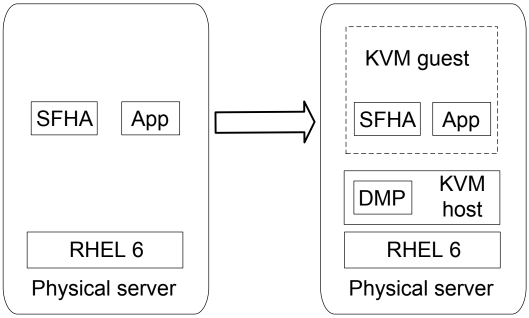
Server consolidation enables you to run virtual machines as physical servers, combining the multiple appllications and their workloads onto a single server for better server utilization.

Figure 4-1 Server consolidation



This solution for a single server with Veritas Storage Foundation High Availability (SFHA) illustrates the migration of a single workload into a KVM Guest.

Figure 4-2 Server consolidation for a simple workload



To implement server consolidation for a simple workload

1. Install SFHA in the virtual machine.

   See "Installing and configuring storage solutions in the KVM guest" on page 35.

2. Map the storage from the array to the host.

   See "Consistent storage mapping in the KVM environment" on page 46.

3. Map the storage from the array to the guest.

4. Go into the guest and make sure you can import disk groups.

Chapter 5

Physical to virtual migration

This chapter includes the following topics:

Physical to virtual migration with KVM and Veritas Storage Foundation High Availability (SFHA) Solutions

Migrating data from physical servers to virtual machines can be painful. Veritas Storage Foundation and High Availability Solutions products can make painful migrations of data from physical to virtual environments easier and safer to execute.

With Veritas Storage Foundation and High Availability Solutions, there is no need to copy any data from source to destination, but rather the administrator reassigns the same storage or a copy of the storage for a test migration, to the virtual environment. Data migration with Storage Foundation (SF), Storage Foundation HA (SFHA), or Storage Foundation Cluster File System HA (SFCFSHA)can be executed in a central location, migrating all storage from an array utilized by Storage Foundation managed hosts.

Physical to virtual migration (P2V) requires migrating data from a physical server to a virtualized guest. The LUNs are first physically connected to the host, and then the LUNs are mapped in KVM from the host to the guest.

Without SF, SFHA, or SFCFS in the host, you must identify which storage devices with mapping to the guest. Putting SF, SFHA, or SFCFS in the host enables quick and reliable identification of storage devices to be mapped. If you are running DMP in the host, you can map the DMP devices directly. Veritas Storage Foundation and High Availability Solutions products add manageability and ease of use to an otherwise tedious and time-consuming process.

How to implement physical to virtual migration (P2V)

Migrating data from a physical server to a virtualized guest, the LUNs are first physically connected to the host, and then the LUNs are mapped in KVM from the host to the guest.

This use case procedure is very similar to the server consolidation use case and the procedures are quite similar. Physical to virtual migration is the process used to achieve server consolidation.

This use case requires Veritas Storage Foundation HA or Veritas Storage Foundation Cluster File System HA in the KVM host and Veritas Storage Foundation in the KVM guest. For setup information:

See “Installing and configuring storage solutions in the host” on page 37.

See “Installing and configuring storage solutions in the KVM guest” on page 35.

There are two options:

- If SFHA Solutions products are installed on both the physical server and the virtual host, identifying the LUNs which need mapping is made easy. Once the LUNs are connected to the virtual host, ‘vxdisk –o alldgs list’ can be used to identify the devices in the disk group which require mapping.
- If Veritas Storage Foundation and High Availability Solutions (SFHA Solutions) products are not installed on the virtual host and the physical server is a Linux system, the devices which need mapping can be identified by using the device IDs on the physical server.

To implement physical to virtual migration with Storage Foundation in the host and guest

1 Find the Linux device IDs of the devices which need mapping.

```
# vxdg list diskgroup
```

2 For each disk in the disk group:

```
# vxdmpadm getsubpaths dmpnodename=device
# ls -al /dev/disk/by-id/* | grep subpath
```

If Storage Foundation is not installed on the host, before decommissioning the physical server, identify the LUNs which require mapping by using the devices serial numbers. The LUNs can be mapped to the guest using the persistent "by-path" device links.

To implement physical to virtual migration if Storage Foundation is not installed in the host

1. On the physical server, identify the LUNs which must be mapped on the KVM host.

   - Collect a list of disks and associated disk groups.

     ```
     # vxdisk -o alldgs list
     DEVICE        TYPE            DISK       GROUP      STATUS
     disk_1        auto:none       -          -          online invalid
     sda           auto:none       -          -          online invalid
     3pardata0_2   auto:cdsdisk    disk01     data_dg    online
     3pardata0_3   auto:cdsdisk    disk02     data_dg    online
     ```

   - Collect a list of the disks and the disks serial numbers.

     ```
     # vxdisk -p -x LUN_SERIAL_NO list
      DEVICE       LUN_SERIAL_NO
      disk_1               3JA9PB27
             sda                     0010B9FF111B5205
             3pardata0_2             2AC00002065C
             3pardata0_3             2AC00003065C
     ```

2. Deport the disk group on the physical machine.

3. Map the LUNs to the virtualization host.

   On the virtualization host, identify the LUNs which were part of the disk group using the serial number. The udev database can be used to identify the devices on the host which need to be mapped.

   ```
   # udevadm info --export-db  | grep -v part |
                   grep -i DEVLINKS=.*200173800013420d0.* | \
                   cut -d\  -f 4
   /dev/disk/by-path/pci-0000:0a:03.0-fc-0x20210002ac00065c:0x0020000
   /dev/disk/by-path/pci-0000:0a:03.1-fc-0x21210002ac00065c:0x0020000

   # udevadm info --export-db  | grep -v part |
                   grep -i DEVLINKS=.*200173800013420d0.* | \
                   cut -d\  -f 4
   /dev/disk/by-path/pci-0000:0a:03.0-fc-0x20210002ac00065c:0x0040000
   /dev/disk/by-path/pci-0000:0a:03.1-fc-0x21210002ac00065c:0x0040000
   ```

   Map the LUNs to the guest. As there are multiple paths in this example, the paths syn-link can be used to ensure consistent device mapping for all four paths.

   ```
   # virsh attach-disk guest1 \
   /dev/disk/by-path/pci-0000:0a:03.0-fc-0x20210002ac00065c:0x0020000 \
         vdb
   # virsh attach-disk guest1 \
   /dev/disk/by-path/pci-0000:0a:03.1-fc-0x21210002ac00065c:0x0020000 \
         vdc
   # virsh attach-disk guest1 \
   /dev/disk/by-path/pci-0000:0a:03.0-fc-0x20210002ac00065c:0x00040000 \
         vdd
   # virsh attach-disk guest1 \
   /dev/disk/by-path/pci-0000:0a:03.1-fc-0x21210002ac00065c:0x00040000 \
         vde
   ```

4. Verify that the devices are correctly mapped to the guest. The configuration changes can be made persistent by redefining the guest.

   ```
   # virsh dumpxml guest1 > /tmp/guest1.xml
   # virsh define /tmp/guest1.xm
   ```

In the procedure example, the disk group *data_dg* is mapped to *guest1* using the DMP devices to map the storage.

To implement physical to virtual migration with Storage Foundation in the guest and host

1. Map the LUNs to the virtualization host.

2. On the virtualization host, identify the devices which require mapping. For example, the devices with the disk group *data_dg* are mapped to *guest1*.

   ```
   # vxdisk -o alldgs list |grep data_dg
   3pardata0_1  auto:cdsdisk    -            (data_dg)    online
   3pardata0_2  auto:cdsdisk    -            (data_dg)    online
   ```

3. Map the devices to the guest.

   ```
   # virsh attach-disk guest1 /dev/vx/dmp/3pardata0_1 vdb
   Disk attached successfully

   # virsh attach-disk guest1 /dev/vx/dmp/3pardata0_2 vdc
   Disk attached successfully
   ```

4. In the guest, verify that all devices are correctly mapped and that the disk group is available.

   ```
   # vxdisk scandisks
   # vxdisk -o alldgs list |grep data_dg
   3pardata0_1  auto:cdsdisk    -            (data_dg)    online
   3pardata0_2  auto:cdsdisk    -            (data_dg)    online
   ```

5. In the virtualization host make the mapping persistent by redefining the guest:

   ```
   # virsh dumpxml guest1 > /tmp/guest1.xml
   # virsh define /tmp/guest1.xml
   ```

To use a Veritas Volume Manager volume as a boot device when configuring a new virtual machine

1. Follow Red Hat's recommended steps to install and boot a VM guest.

   When requested to select managed or existing storage for the boot device, use the full path to the VxVM storage volume block device, for example */dev/vx/dsk/boot_dg/bootdisk-vol*.

2. If using the `virsh-install` utility, enter the full path to the VxVM volume block device with the --disk parameter, for example, *--disk path=/dev/vx/dsk/boot_dg/bootdisk-vol*.

Simplified management

This chapter includes the following topics:

Simplified management with KVM and Veritas Storage Foundation High Availability (SFHA) Solutions

Independent of how an operating system is hosted, consistent storage management tools save an administrator time and reduce the complexity of the environment. Veritas Storage Foundation and High Availability Solutions provide in the guest provide the same command set, storage namespace, and environment as in a non-virtual environment.

This use case requires Veritas Storage Foundation HA or Veritas Storage Foundation Cluster File System HA in the KVM host. For setup information:

See "Installing and configuring storage solutions in the host" on page 37.

Provisioning Storage Foundation High Availability (SFHA) storage for a KVM guest virtual machine

A volume can be provisioned within a VM guest as a data disk or a boot disk.

- Data disk: provides the advantage of mirroring data across arrays.

- Boot disk: provides the ability to migrate across arrays.

Adding a VxVM storage volume as a data disk to a running guest virtual machine can be done in the following ways:

- Using the `libvirt` GUI
- Using the `virsh` command line.

Provisioning Veritas Volume Manager volumes as data disks for KVM virtual machine guests

The following procedure uses Veritas Volume Manager (VxVM) volumes as data disks (virtual disks) for VM guests. The example host is *host1* and the VM guest is *guest1*. The prompts in each step show in which domain to run the command.

To provision Veritas Volume Manager volumes as data disks

1 Create a VxVM disk group (*mydatadg* in this example) with some disks allocated to it:

```
host1# vxdg init mydatadg TagmaStore-USP0_29 TagmaStore-USP0_30
```

2 Create a VxVM volume of the desired layout (in this example, creating a simple volume):

```
host1# vxassist -g mydatadg make datavol1 500m
```

3 Map the volume *datavol1* to the VM guest:

```
host1# virsh attach-disk guest1/dev/vx/dsk/mydatadg/datavol1 vdb
```

4 To make the mapping persistent, redefine the VM guest.

```
host1# virsh dumpxml guest1 > /tmp/guest1.xml
```

```
host1# virsh define /tmp/guest1.xml
```

5 On the guest, create a VxVM volume of a size that is recommended for OS installation. In this example, a 16GB volume is created:

```
host1# vxassist -g boot_dg make bootdisk-vol 16g
```

6 Follow Red Hat's recommended steps to install and boot a VM guest.

When requested to select managed or existing storage for the boot device, use the full path to the VxVM storage volume block device, for example */dev/vx/dsk/boot_dg/bootdisk-vol*.

Provisioning Veritas Volume Manager volumes as boot disks for guest virtual machines

The following procedure provisions boot disks for a VM guest.

The following process gives the outline of how a Veritas Volume Manager (VxVM) volume can be used as a boot disk.

The example host is *host1* the VM guest is *guest1*. The prompts in each step show in which domain to run the command.

To provision Veritas Volume Manager volumes as boot disks for guest virtual machines

1 On the host, create a VxVM volume of a size that is recommended for Red Hat Enterprise Linux (RHEL) 6.1 and 6.2 installations. In this example, a 16GB volume is created:

```
host1# vxassist -g boot_dg make bootdisk-vol 16g
```

2 Follow Red Hat’s recommended steps to install and boot a VM guest, and use the virtual disk as the boot disk.

Boot image management with KVM and Veritas Storage Foundation High Availability (SFHA) Solutions

With the ever-growing application workload needs of data centers comes the requirement to dynamically create virtual environments. This creates a need for the ability to provision and customize virtual machines on-the-fly. Every virtual machine created needs to be provisioned with a CPU, memory, network and I/O resources.

As the number of guest virtual machines increase on the physical host, it becomes increasingly important to have an automatic, space-optimizing provisioning mechanism. Space-savings can be achieved as all the guest virtual machines can be installed with the same operating system, i.e., boot volume. Hence, rather than allocate a full boot volume for each guest, it is sufficient to create single boot volume and use space-optimized snapshots of that "Golden Boot Volume" as boot images for other virtual machines.

The primary I/O resource needed is a boot image, which is an operating system environment that consists of: the following

- A bootable virtual disk with the guest operating system installed
- A bootable, a guest file system
- A custom or generic software stack

For boot image management, Storage Foundation and High Availability (SFHA) Solutions products enable you to manage and instantly deploy virtual machines based on templates and snapshot-based boot images (snapshots may be full or space optimized). For effective boot image management in KVM based virtual environments, deploy the SFHA Solutions products in the combined host and guest configuration.

Benefits of boot image management:

- Eliminates the installation, configuration and maintenance costs associated with installing the operating system and complex stacks of software
- Infrastructure cost savings due to increased efficiency and reduced operational costs.
- Reduced storage space costs due to shared master or gold image as well as space-optimized boot images for the various virtual machines
- Enables high availability of individual guest machines with Veritas Cluster Server (running on the host) monitoring the VM guests and their boot images
- Ability to create and deploy virtual machines across any remote node in the cluster

Creating the boot disk group

Once Storage Foundation High Availability (SFHA) is installed on the Red Hat Enterprise Linux (RHEL) server using the combined host and virtual machine (VM guest configuration, the next step is to create a disk-group in which the Golden Boot Volume and all the various space-optimized snapshots (VM) boot images) will reside. For a single-node environment, the disk-group is local or private to the host. For a clustered environment (recommended for live migration of VMs), Symantec recommends creating a shared disk-group so that the Golden Boot Volume can be shared across multiple physical nodes.

It is possible to monitor the disk-group containing the Guest VM boot image(s) and the guest VMs themselves under VCS so that they can be monitored for any faults. However it must be kept in mind that since the boot images are in the same disk-group, a fault in any one of the disks backing the snapshot volumes containing the boot disks can cause all the guest VMs housed on this node to failover to another physical server in the SFCFS cluster. To increase the fault tolerance for this disk-group, mirror all volumes across multiple enclosures making the volumes redundant and less susceptible to disk errors.

To create a shared boot disk group

1. Create a disk group, for example *boot_dg*.

   ```
   $ vxdg -s init boot_dg  device_name_1
   ```

2. Repeat to add multiple devices.

   ```
   $ vxdg -g boot_dg adddisk device_name_2
   ```

Creating and configuring the golden image

The basic idea is to create a point-in-time image based on a master or gold image. The image will serve as the basis for all boot images once it is set up. Hence, first set up a complete virtual machine boot volume as a golden boot volume.

To create the golden image

1. In the selected disk group, create a VxVM volume of a size that is recommended for Red Hat Enterprise Linux (RHEL) 6.1 and 6.2 installations. For example, the disk group is *boot_dg*, the golden boot volume is *gold-boot-disk-vol*, the volume size is 16GB.

   ```
   host1# vxassist -g boot_dg make gold-boot-disk-vol 16g
   ```

2. Follow Red Hat's recommended steps to install and boot a VM guest.

   When requested to select managed or existing storage for the boot device, use the full path to the VxVM storage volume block device, for example */dev/vx/dsk/boot_dg/bootdisk-vol.*

3. If using the `virsh-install` utility, enter the full path to the VxVM volume block device with the --disk parameter, for example, *--disk path=/dev/vx/dsk/boot_dg/bootdisk-vol.*

4. After the virtual machine is created, install any guest operating system with the boot volume and the virtual machine configured exactly as required.

5. After the virtual machine is created and configured, shut it down.

You can now use the boot image as a image (hence called a golden image) for provisioning additional virtual machines that are based on snapshots of the Golden Boot Volume. These snapshots can be full copies (mirror images) or they can be space-optimized snapshots. Using space-optimized snapshots greatly reduces the storage required to host the boot disks of identical multiple virtual machines. Note that since both, the full and space-optimized snapshots, are instantly available (no need to wait for the disk copy operation), provisioning of new virtual machines can now be instantaneous as well.

Rapid provisioning of virtual machines using the golden image

As mentioned above, for rapid provisioning of new virtual machines based on the golden image, we need to have full or space-optimized snapshots of the Golden Boot Volume. These snapshots can then be used as boot images for the new virtual machines. The process to create these snapshots is outlined below in the procedures below.

Creating instant, full snapshots of golden boot volume for rapid virtual machine provisioning

To create instant, full snapshots of the golden boot volume for rapid virtual machine provisioning

1. Prepare the volume for an instant full snapshot. In the example, the disk group is *boot_dg* and the golden boot volume is “gold-boot-disk-vol”)

   ```
   $ vxsnap -g boot_dg prepare gold-boot-disk-vol
   ```

2. Create a new volume which will be used as the boot volume for the new provisioned guest. The size of the guests boot volume must match the size of the golden boot volume.

   ```
   $ vxassist -g boot_dg make guest1-boot-disk-vol 16g layout=mirror
   ```

3. Prepare the new boot volume so it can be used as a snapshot volume.

   ```
   $ vxsnap -g boot_dg prepare guest1-boot-disk-vol
   ```

4. Create the full instant snapshot of the golden boot volume.

   ```
   $ vxsnap -g boot_dg make source=gold-boot-disk-vol/snapvol=\
     guest1-boot-disk-vol/syncing=off
   ```

5. 5. Create a new virtual machine, using the snapshot *guest1-boot-disk-vol* as an "existing disk image."

To create instant, space-optimized snapshots of the golden boot volume for rapid virtual machine provisioning

1 Prepare the volume for an instant snapshot. In the example, the disk group is*boot_dg* and the golden boot volume is “gold-boot-disk-vol”)

```
$ vxsnap -g boot_dg prepare gold-boot-disk-vol
```

2 Use the `vxassist` command to create the volume that is to be used for the cache volume. The cache volume will be used to store writes made to the space-optimized instant snapshots.

```
$ vxassist -g boot_dg make cache_vol 5g  layout=mirror init=active
```

3 Use the `vxmake cache` command to create a cache object on top of the cache volume which you created in the previous step.

```
$ vxmake -g boot_dg cache cache_obj cachevolname=cache_vol autogrow=on
```

4 Start the cache object:

```
$ vxcache -g boot_dg start cache_obj
```

5 Create a space-optimized instant snapshot of the golden boot image:

```
$ vxsnap -g boot_dg make source=\
gold-boot-disk-vol/newvol=guest1-boot-disk-vol/cache=cache_obj
```

6 6. Create a new virtual machine, using the snapshot of the golden image as an existing disk image.

Storage savings from space-optimized snapshots

With the large number of virtual machines housed per physical server, the number of boot images used on a single server is also significant. A single bare-metal RHEL (v 6.0) boot image needs around 3 GB of space at a minimum. Installing software stacks and application binaries on top of that requires additional space typically resulting in using around 6 GB of space for each virtual machine that houses a database application.

When a user provisions a new virtual machine, the boot image can be a full copy or a space-optimized snapshot. Using a full copy results in highly inefficient use of storage. Not only is storage consumed to house identical boot images, storage is also consumed in making the boot images highly available (mirror across enclosures) as well in their backup.This large amount of highly available, high performance storage is very expensive, and likely to eliminate the cost advantages that server virtualization would otherwise provide. To add to it, backup and recovery of such capacity is also an expensive task.

In order to address the above issue, Symantec recommends the use of space-optimized napshots of the gold image as boot images of the various VM guests. Space-optimized snapshots do not make a full copy of the data in the gold image, rather they work on the copy-on-write principle where only the changed blocks are stored locally. This set of changed blocks is called a Cache Object and it is stored in a repository for all such space-optimized snapshots, called the Cache Object Store, which is backed by physical storage. The Cache Object offers a significant storage space reduction, typically occupying a 5-20% storage footprint, relative to the parent volume (the gold image volume in this case). The same Cache Object Store can be used to store changed blocks for multiple snapshot volumes.

Each Snapshot held in the Cache Object Store contains only changes made to the gold image to support that installation’s boot environment. Hence, to achieve the best possible storage reduction, install software on data disks rather than root file systems and limit as many changes as possible to the gold image operating files (i.e., system, hosts, passwd, etc.).

Chapter 7

Application availability

This chapter includes the following topics:

Application availability options with KVM and Veritas Storage Foundation High Availability (SFHA) Solutions

Symantec products can provide the ultimate levels of availability in your KVM environment. In an KVM environment, you can choose a different combination of Symantec High Availability solutions: ApplicationHA and Veritas Cluster Server (VCS).

ApplicationHA by itself provides application monitoring and restart capabilities while providing ultimate visibility and manageability through Veritas Operations Manager. When ApplicationHA is adopted together with Veritas Cluster Server in the host, the two solutions work together to ensure that the applications are monitored and restarted if needed, and virtual machines are restarted if application restarts are not effective. These two solutions work together to provide the ultimate level of availability in your KVM environment.

If your KVM environment requires the same level of application availability provided by a VCS cluster in a physical environment, you can choose to adopt

Veritas Cluster Server in the virtual machines. In this configuration, your application enjoys fast failover capability in a VCS cluster in the virtual machines.

Table 7-1 Comparison of availability options

Required availability level	Recommended solution
Application monitoring and restart	ApplicationHA in the virtual machines
Virtual machine monitoring and restart	VCS cluster in the host monitoring the virtual machines as a resource
Combined applcation and virtual machine availability	ApplicationHA in the KVM guest and VCS cluster in the KVM host
Application failover to standby node in cluster	VCS cluster in the virtual machines

For setup information for ApplicationHA or VCS:

See "Installing and configuring Veritas Cluster Server for virtual machine and application availability " on page 39.

Note: You can also use the cluster functionality of Veritas Storage Foundation HA or Veritas Storage Foundation Cluster File System HA if you need storage management capabilities in addition to application availability for your KVM environment.

Veritas Cluster Server in a KVM environment: architecture summary

VCS in host architecture	VCS in guest architecture
■ Manages multiple guest virtual machines as a single unit of control ■ Provides automatic restart or fail-over of individual guest virtual machines in response to failures ■ Provides Start / Stop / Monitor of individual guest virtual machines from a common console across the entire server pool using Veritas Operations Manager (VOM)	■ Manages applications running in the guest virtual machine as a single unit of control ■ Provides automatic restart or fail-over of individual applications to other guest virtual machine or physical machine. ■ Provides Start / Stop / Monitor of individual applications from a common console across appropriate guest virtual machines in the farm using Veritas Operations Manager (VOM)

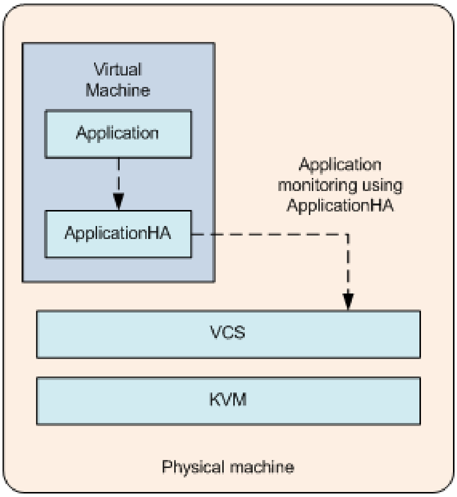
VCS in host to provide the Virtual Machine high availability and ApplicationHA in guest to provide application high availability

VCS running in the host monitors the virtual machine to provide the VM high availability. ApplicationHA running in the VM guest ensures the application high availability by monitoring the configured application . VCS and ApplicationHA can be combined together to provide the enhanced solution for achieving application and VM high availability.

VCS in host provides the primary VCS monitoring. It can start/stop the virtual machine and fail-over it to another node in case of any fault. We then run ApplicationHA within the guest that monitors the application running inside the guest virtual machine. ApplicationHA in guest will not trigger an application fail-over in case of application fault, but it'll try to restart the application on same VM guest. If ApplicationHA fails to start the application, it can notifie the VCS running in the host to take corrective action which includes virtual machine restart or virtual machine fail-over to another host. For detailed information about ApplicationHA and integration of ApplicationHA with VCS, please refer ApplicationHA documentation.

For detailed information about ApplicationHA and integration of Application HA with VCS, please refer ApplicationHA documentation.

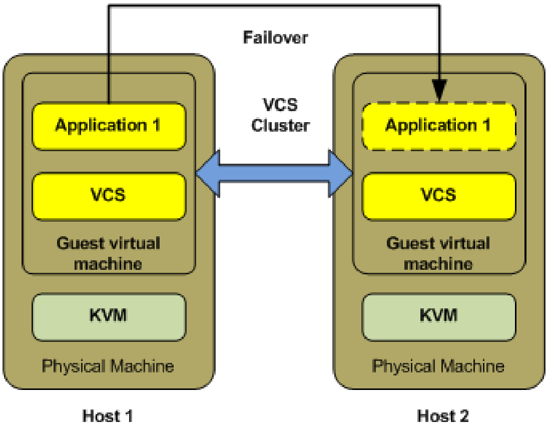
Figure 7-1 VCS In host for VM HA and ApplicationHA in guest for application HA



Virtual to Virtual clustering and failover

Running VCS in multiple guest virtual machines enables guest-to-guest clustering. VCS can then monitor individual applications running within the guest and then fail over the application to another guest in the virtual – virtual cluster.

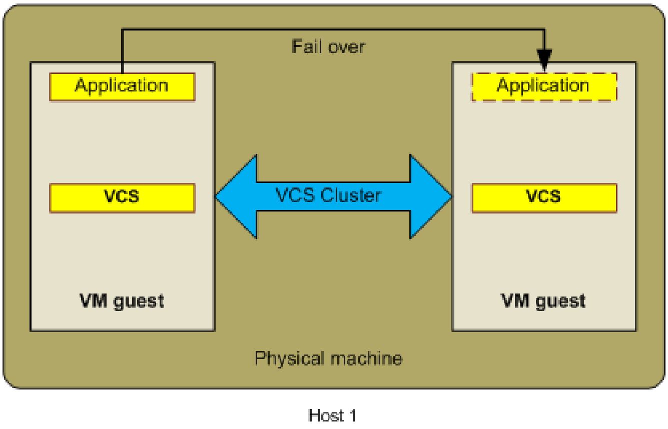
Figure 7-2 Clustering between guests for application high availability



You can run VCS within each guest machine to provide high availability to applications running within the guest.

A VCS cluster is formed among the VM guests in this configuration. The VM guests in the cluster can be either on the same physical host or on different physical hosts. VCS is installed in the VM guests in the cluster. This VCS is similar to the VCS installed in the physical machine clusters. This VCS cluster manages and controls the applications and services that run inside the VM guests. Any faulted application or service is failed over to other VM guest in the cluster. This configuration does not take care of the VM guest fail-overs since VCS runs inside the VM guest.

Figure 7-3 VCS cluster across VM guests on the same physical machine



Virtual to Physical clustering and failover

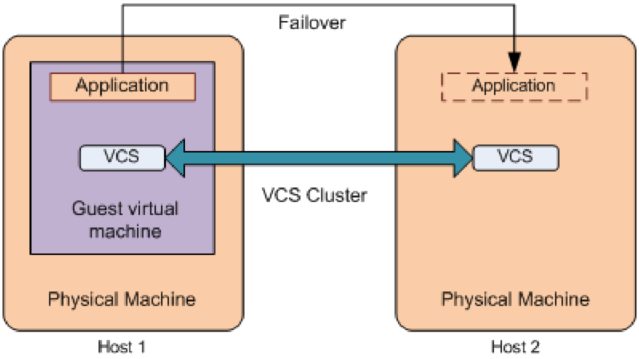
One can also create a physical to virtual cluster by combining VCS In-guest together with VCS running on any other physical host. This virtual-physical cluster enables VCS to monitor applications running within the guest and then fail over the application to another host. The reverse flow is also true, thus enabling the fail-over of an application running on a physical host into a VM guest machine.

A VCS cluster is formed among the VM guests and physical machines. VCS is installed on the VM guests and on different physical machines in the cluster. VM guests are connected to physical machines through the network of their VM hosts. In this case, the VM host is a physical machine on which one or more VM guests forming the cluster are hosted.

This VCS cluster manages and monitors the services and applications running on cluster nodes that can either be VM guests or physical machines. Any faulted application on one node fails over to other node that can either be a virtual machine or a physical machine.

See “Standard bridge configuration” on page 54.

Figure 7-4 VCS cluster across VM guest and physical machine



Virtual machine availability

This chapter includes the following topics:

About virtual machine availability options with KVM and Veritas Storage Foundation High Availability (SFHA) Solutions

While application availability is very important for KVM users, virtual machine availability is equally important in KVM environments. Virtual machine availability can be provided by adopting Veritas Cluster Server (VCS) in the host. VCS in this case monitors the virtual machines as a resource.

See Table 7-1 on page 76.

For setup information for VCS:

See "Installing and configuring Veritas Cluster Server for virtual machine and application availability " on page 39.

Note: You can also use the cluster functionality of Veritas Storage Foundation HA or Veritas Storage Foundation Cluster File System HA if you need storage management capabilities in addition to virtual machine availability for your KVM host.

VCS in host monitoring the Virtual Machine as a resource

In this scenario, VCS runs in the host, enabling host-level clustering. Running VCS in the host also enables the monitoring and fail-over of individual guest virtual machines. Each guest virtual machine is simply a process in the KVM architecture and hence can be monitored by VCS running on the host. This capability allows us to monitor the individual virtual machine as an individual resource and restart/fail-over the VM on the same (or another physical) host. To enable support for guest live migration, it is recommended to run CVM in the host.

In this configuration, the physical machines (PMs) hosting VM guests form a cluster. Therefore, VCS does not monitor applications running inside the guest virtual machines. VCS controls and manages the virtual machines with the help of the KVM agent for VCS. If a VM guest faults, it fails over to the other host. The VM guests configured as failover service groups in VCS must have same configuration across all hosts. The storage for the VM guests must be accessible to all the hosts in the cluster.

See "Network configuration for VCS cluster across physical machines (PM-PM)" on page 53.

See "Sample configuration" on page 87.

Chapter 9

Virtual machine availability using Live Migration

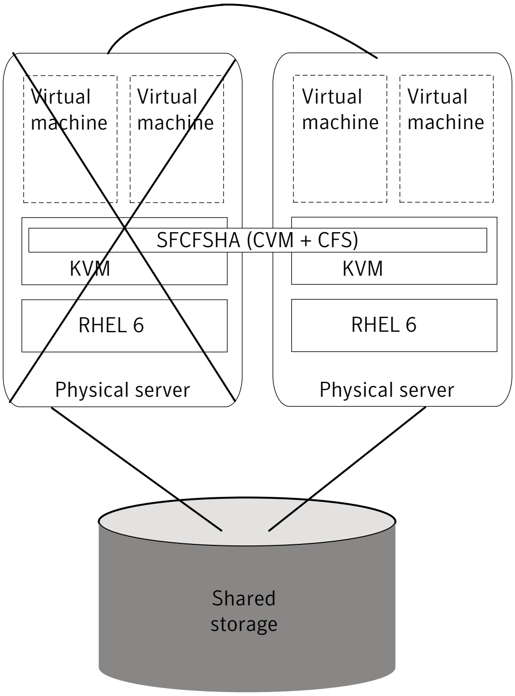
This chapter includes the following topics:

About Live Migration with KVM and Veritas Storage Foundation High Availability (SFHA) Solutions

You can enable Live Migration of guest virtual machines using shared storage through Veritas Cluster Volume Manger (CVM), a component of Veritas Cluster File System HA. Using CVM significantly reduces planned downtime for individual virtual machines. Individual virtual machines can now be statefully migrated from host to host, enabling better load-balancing, lower machine downtime and path-management of individual physical servers. Physical servers (hosts) can now join and exit the server pool (physical server cluster) at will while the individual guest virtual machines and their corresponding applications continue to run.

For Live Migration, by using Fast Failover using CVM in the guest and host, rather than running a single-node Veritas Volume Manager (VxVM) in the host, you can run the CVM in the host and cluster multiple physical servers within the same server cluster or server pool. This configuration includes Veritas Cluster Server (VCS) also within the host. The significant advantage of creating a cluster of physical servers is that Live Migration of KVM guest virtual machines from one physical server to another is fully operational and supported.

Figure 9-1 Live Migration setup



Live Migration requirements

The following conditions are required for migrating a VM guest from source host to destination host:

- The required guest image must be available on the destination host at the same location.
- The storage and network devices configured in the migrating guest must be identical on source and destination hosts. Any difference these may cause the migration process to terminate.
- The KVM hypervisor version on both the hosts should be same along with the operating system level.

For detailed information about the required and limitation of guest migration, see the Red Hat Virtualization Guide.

http://docs.redhat.com/docs/en-US/Red_Hat_Enterprise_Linux/6/pdf/Virtualization/Red_Hat_Enterprise_Linux-6-Virtualization-en-US.pdf

Implementing Live Migration for virtual machine availability

A VM guest can be migrated from one host to another host. This migration can be a live migration or pause migration. You can initiate the migration using either the `virsh migrate` command or using `virt-manager` console. Veritas Cluster Server (VCS) monitors the migrated guest and can detect the migration process. VCS changes the resource state according to the state, i.e. if the guest is live-migrated from one host to another host, the associated KVMGuest resource is brought online on the host where the guest is migrated. VCS does not initiate the VM guest migration. Sample Configuration Symantec recommends the use of CVM-CFS in case of VM guest migration for storing the guest image.

See “Sample configuration” on page 87.

SFCFSHA in the host, guest is not needed

Reference information

This appendix includes the following topics:

- RHEL-based KVM installation and usage
- Sample configuration

RHEL-based KVM installation and usage

You can install all the required RPMs through the following `yum` command:

```
# yum grouplist|grep -i virtualization
```

Subsequently, you can install the virtualization package with the following command:

```
# yum groupinstall "Virtualization"
```

Sample configuration

You can use any of the followoing sample confirgurations:

- Sample configuration 1: Native LVM volumes are used to store the guest image
- Sample configuration 2: Native VxVM volumes are used to store the guest image
- Sample configuration 3: Native CVM-CFS is used to store the guest image

Sample configuration 1: Native LVM volumes are used to store the guest image

```
group kvmtest1 (
SystemList = { north = 0, south = 1 }
)
KVMGuest res1 (
GuestName = kvmguest1
GuestConfigFilePath = "/kvmguest/kvmguest1.xml"
DelayAfterGuestOnline = 10
DelayAfterGuestOffline = 35
)
Mount mnt1 (
BlockDevice = "/dev/mapper/kvmvg-kvmvol"
MountPoint = "/kvmguest"y of the
FSType = ext3
FsckOpt = "-y"
MountOpt = "rw"
)
LVMLogicalVolume lv1 (
VolumeGroup = kvmvg
LogicalVolume = kvmvol
)
LVMVolumeGroup vg1 (
VolumeGroup = kvmvg
)
res1 requires mnt1
mnt1 requires lv1
lv1 requires vg1
```

Sample configuration 2: Native VxVM volumes are used to store the guest image

```
group kvmtest2 (
SystemList = { north = 0, south = 1 }
)
KVMGuest res1 (
GuestName = kvmguest1
GuestConfigFilePath = "/kvmguest/kvmguest1.xml"
DelayAfterGuestOnline = 10
DelayAfterGuestOffline = 35
)
Mount mnt1 (
BlockDevice = "/dev/vx/dsk/kvmvg/kvmvol"
MountPoint = "/kvmguest"
FSType = vxfs
FsckOpt = "-y"
MountOpt = "rw"
)
Volume vol1 (
Volume = kvm_vol
DiskGroup = kvm_dg
)
DiskGroup dg1 (
DiskGroup = kvm_dg
)
res1 requires mnt1
mnt1 requires vol1
vol1 requires dg1
```

Sample configuration 3: Native CVM-CFS is used to store the guest image

```
group kvmgrp (
SystemList = { kvmpm1 = 0, kvmpm2 = 1 }
)
KVMGuest kvmres (
GuestName = kvmguest1
GuestConfigFilePath = "/cfsmount/kvmguest1.xml"
DelayAfterGuestOnline = 10
DelayAfterGuestOffline = 35
)

kvmgrp requires group cvm online local firm

group cvm (
SystemList = { kvmpm1 = 0, kvmpm2 = 1 }
AutoFailOver = 0
Parallel = 1
AutoStartList = { kvmpm1, kvmpm2 }
)
CFSMount cfsmount (
MountPoint = "/cfsmount"
BlockDevice = "/dev/vx/dsk/cfsdg/cfsvol"
)
CFSfsckd vxfsckd (
)
CVMCluster cvm_clus (
CVMClustName = kvmcfs
CVMNodeId = { kvmpm1 = 0, kvmpm2 = 1 }
CVMTransport = gab
CVMTimeout = 200
)
CVMVolDg cfsdg (
CVMDiskGroup = cfsdg
CVMVolume = { cfsvol }
CVMActivation = sw
)
CVMVxconfigd cvm_vxconfigd (
Critical = 0
CVMVxconfigdArgs = { syslog }
)

cfsmount requires cfsdg
cfsmount requires cvm_clus
cvm_clus requires cvm_vxconfigd
vxfsckd requires cvm_clus
```