Veritas Cluster Server Application Note: Support for Kernel-Based Virtual Machines

RHEL

5.1 Service Pack 1

Veritas™ Cluster Server Application Note: Support for Kernel-Based Virtual Machines

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: 5.1 SP1

Document version: 5.1SP1.0

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 software disc 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 website.

http://www.symantec.com/business/support/overview.jsp?pid=15107

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:

docs@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

Additional enterprise services

Symantec offers a comprehensive set of services that allow you to maximize your investment in Symantec products and to develop your knowledge, expertise, and global insight, which enable you to manage your business risks proactively.

Enterprise services that are available include the following:

Managed Services	These services remove the burden of managing and monitoring security devices and events, ensuring rapid response to real threats.

Consulting Services

Symantec Consulting Services provide on-site technical expertise from Symantec and its trusted partners. Symantec Consulting Services offer a variety of prepackaged and customizable options that include assessment, design, implementation, monitoring, and management capabilities. Each is focused on establishing and maintaining the integrity and availability of your IT resources.

Education Services

Education Services provide a full array of technical training, security education, security certification, and awareness communication programs.

To access more information about enterprise services, please visit our Web site at the following URL:

www.symantec.com/business/services/

Select your country or language from the site index.

Contents

Chapter 1

Veritas Cluster Server Support for Kernel-Based Virtual Machines

This chapter includes the following topics:

- Overview
- KVM and Veritas Cluster Server clustering configurations
- Limitations and unsupported KVM features

Overview

The Linux Kernel Virtual Machine (KVM) is the latest offer from Red Hat Enterprise Linux (RHEL), starting in RHEL 5.4 for a complete virtualization solution. This document explains how you can use Veritas Cluster Server software in RHEL KVM-based virtualization environments to provide mission-critical clustering and failover capabilities. This document also explains a set of supported clustering architectures that you can implement.

KVM architecture

The KVM virtualization architecture represents the latest generation of virtualization hypervisors. It leverages the hardware-assisted virtualization features of Intel and AMD developed within their CPU architectures. Even though Intel and AMD have different architectures, both significantly reduce the CPU and hypervisor overhead required for virtualization.

KVM is available in the Linux kernel from 2.6.20 and later. It consists of a loadable kernel module kvm.ko that provides the core virtualization infrastructure and converts the standard Linux kernel to a bare-metal hypervisor. Its processor-specific modules are kvm-intel.ko or kvm-amd.ko. Therefore, KVM requires Intel Vt-x and AMD-V enabled processors. It leverages these features to virtualize the CPU. KVM uses QEMU as an adjunct tool to handle device emulation, making it a complete virtualization suite. The KVM architecture benefits from using the same memory manager, process scheduler, and I/O network stack as provided within the Linux kernel.

Each guest VM is implemented as a regular Linux process. The KVM module is used to start and run new guest operating systems, and to provide them with virtualized environments.

Since KVM leverages hardware-assisted virtualization, the guest VM kernel is a normal, unmodified kernel. Hence, the KVM kernel is a hypervisor that can also run any other applications exactly like a regular Linux distribution without requiring a specific Console OS (VMware ESX) or domain0 (Xen). CPU virtualization (virtual processor within the guest) is simply provided as a separate Linux process. Memory virtualization is provided through the kernel memory manager, by a special device (of KVM: /dev/kvm) which maps the guest operating systems physical addresses to the virtual addresses on the hypervisor. I/O virtualization for the guest in KVM is provided by QEMU. A separate QEMU process runs for each guest OS and virtualizes (or emulates) the entire set of devices on the host and makes them available to the guest. Any I/O done to these devices by the guest is intercepted and re-routed to the device in user-mode by the QEMU process. The flexibility of utilizing large set of devices is offset by the relative small performance toll of rerouting I/O. RHEL based KVM also provides para-virtualized (virtio) drivers for all supported operating systems.

RHEL-based KVM installation and usage

KVM is available as a part of RHEL 5.4 and later. You can manage KVM either through the Red Hat Enterprise Virtualization Manager (RHEV-M) or through separate RPMs that can be downloaded into the standard RHEL 5.4 installation. The installation and usage information given in this document is focused on using KVM-based virtualization as provided through the RHEL 5.4 distribution.

The standard installation does not yet install the virtualization tools. The following additional RPMs are required to be installed for enabling the virtualization capabilities:

- `kvm-83-105.el5.x86_64.rpm`
  `virt-viewer-0.0.2-3.el5.x86_64.rpm`
  `virt-manager-0.6.1-8.el5.x86_64.rpm`

```
python-virtinst-0.400.3-5.el5.noarch.rpm
libvirt-python-0.6.3-20.el5.x86_64.rpm
libvirt-0.6.3-20.el5.x86_64.rpm
kvm-qemu-img-83-105.el5.x86_64.rpm
etherboot-zroms-kvm-5.4.4-10.el5.x86_64.rpm
kmod-kvm-83-105.el5.x86_64.rpm
celt051-0.5.1.3-0.el5.x86_64.rpm
celt051-devel-0.5.1.3-0.el5.x86_64.rpm
log4cpp-1.0-4.el5.x86_64.rpm
log4cpp-devel-1.0-4.el5.x86_64.rpm
qcairo-1.8.7.1-3.el5.x86_64.rpm
qspice-0.3.0-39.el5.x86_64.rpm
qspice-libs-0.3.0-39.el5.x86_64.rpm
qspice-libs-devel-0.3.0-39.el5.x86_64.rpm
qcairo-devel-1.8.7.1-3.el5.x86_64.rpm
qffmpeg-devel-0.4.9-0.15.20080908.el5.x86_64.rpm
qffmpeg-libs-0.4.9-0.15.20080908.el5.x86_64.rpm
qpixman-0.13.3-4.el5.x86_64.rpm
qpixman-devel-0.13.3-4.el5.x86_64.rpm
```

The above-stated RPMs also have the following essential dependencies:

```
/Server/xen-libs-3.0.3-94.el5.x86_64.rpm
/Server/gnome-python2-gnomekeyring-2.16.0-3.el5.x86_64.rpm
/Server/gtk-vnc-python-0.3.8-3.el5.x86_64.rpm
/Server/cyrus-sasl-md5-2.1.22-5.el5.x86_64.rpm
/Server/gtk-vnc-0.3.8-3.el5.x86_64.rpm
```

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

```
# yum grouplist|grep KVM
```

Subsequently, you can install the KVM group with the following command:

```
# yum groupinstall "KVM"
```

KVM Terminology used in this document

Table 1-1 KVM terminology used in this document

Term	Definition
KVM	Kernel-based Virtual Machine
KVMGuest	KVM virtualized guest.

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

Term	Definition
Host	The physical host on which KVM is installed.
PM	The physical machine running VCS.
KVM-KVM	VCS-supported configuration in which a cluster is formed between KVMGuests running on top of the same or different hosts.
KVM-PM	VCS-supported configuration in which a cluster is formed between KVMGuests and physical machines.
PM-PM	VCS-supported configuration in which a cluster is formed between hosts, and which is mainly used to manage KVMGuests 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 KVMGuests.

VCS setup checklist

Table 1-2 System requirements for the KVM-supported configurations

VCS version	5.1 Service Pack 1
Supported OS version in host	RHEL 5.4 and 5.5
Supported OS in KVMGuest	RHEL 5.4 and 5.5
Hardware requirement	Full virtualization-enabled CPU

KVM and Veritas Cluster Server clustering configurations

The following Veritas Cluster Server configurations are supported on KVM:

- VCS cluster across KVMGuests (KVM-KVM) on the same or different physical hosts - for application availability
- VCS cluster across physical machines (PM-PM) without resource monitoring inside KVMGuests - for virtual machine availability
- VCS cluster across physical machines (PM-PM) with resource monitoring inside KVMGuests - for both virtual machine and application availability

VCS cluster across KVMGuests (KVM-KVM) on the same or different physical hosts

You can run VCS within each guest machine to provide high availability to applications running within the guest. Note that in the KVM environment, full SCSI-3 PGR operations are not available to guest virtual machines and hence SCSI-3 based I/O fencing is not supported.

A VCS cluster is formed among the KVMGuests in this configuration. The KVMGuests in the cluster can be either on the same physical host or on different physical hosts. VCS is installed in the KVMGuests 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 KVMGuests. Any faulted application or service is failed over to other KVMGuest in the cluster. This configuration does not take care of the KVMGuest fail-overs since VCS runs inside the KVMGuest.

Figure 1-1 VCS cluster across KVMGuests on separate physical machines

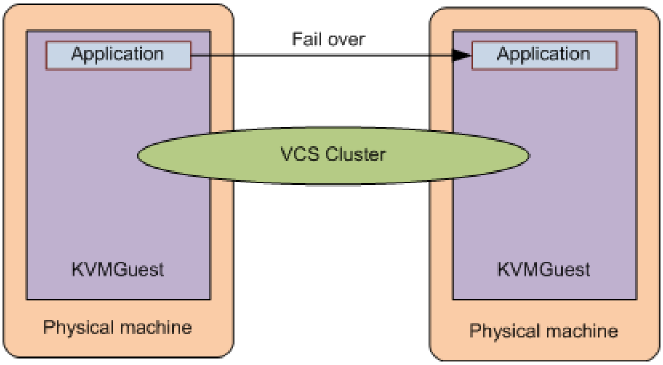
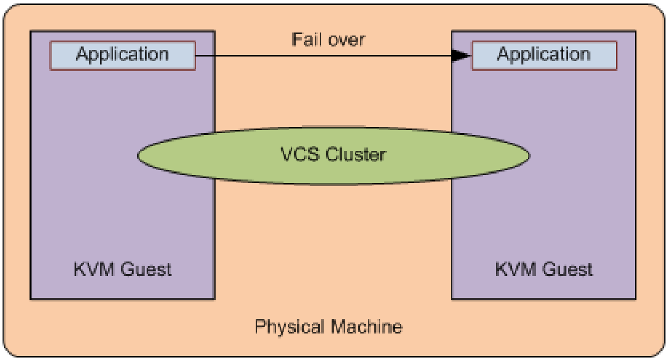


Figure 1-2 VCS cluster across KVMGuests on the same physical machine



Network configuration for KVM-KVM cluster

To manage the VCS cluster between the virtual machines, you must configure the network and the storage domains for the cluster. The setup details for network and storage configurations are explained in the subsequent sections. Figure 1-1 shows a cluster setup between two KVMGuests running on two different hosts.

See "Bridge network configuration" on page 29.

Setting up KVMGuest

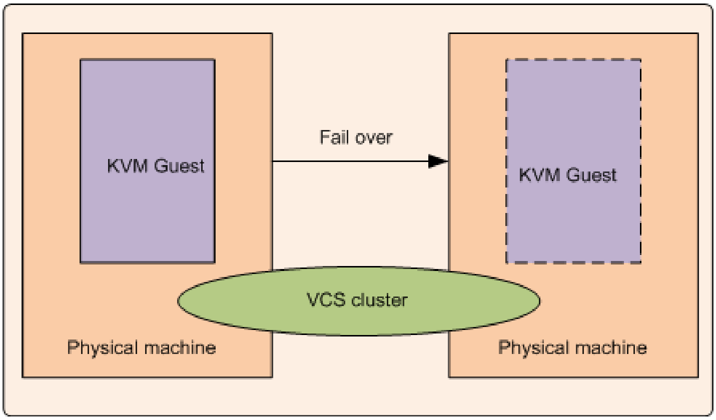
Following is a high-level overview of the steps required for setting up KVMs. For detailed instructions, refer to *Red Hat Enterprise Linux Virtualization Guide*.

1. Before creating KVMs, ensure that CPU and memory resources are available to create KVMGuests 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 KVMGuests are to be created.
4. Create KVMGuests. For network configuration, refer to the *Network configuration for KVM-KVM cluster*.
5. Install the operating system in the KVMGuests.
6. Repeat the above steps for all KVMGuests that you want to be a part of the cluster.
7. Install VCS on all the KVMGuests. 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.

VCS cluster across physical machines (PM-PM) without resource monitoring inside KVMGuests

In this configuration, the physical machines (PMs) hosting KVMGuests form a cluster. Therefore, there is a VCS cluster consisting of hosts only. A KVMGuest cannot be a cluster node in this configuration. VCS is installed on the hosts in the cluster. VCS running on hosts does not monitor resources inside KVMGuests. VCS controls and manages the virtual machines with the help of the application agent. If a KVMGuest faults, it fails over to the other host. The KVMGuests configured as VCS service groups in VCS must have same configuration across all hosts. The storage for the KVMGuests must be accessible to all the hosts in the cluster.

Figure 1-3 VCS cluster across physical machines with KVMGuests



Network and storage configuration

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 KVMGuests are configured.

How VCS manages KVMGuests

Following is a high-level overview of how VCS manages KVMGuests. For detailed instructions on creating and configuring a KVMGuest, refer to the *Installation* section in the *Red Hat Enterprise Linux Virtualization Guide*.

1. Physical machines form a cluster with VCS installed on them. For information about installing VCS, refer to the *Veritas Cluster Server Installation Guide*.

2. CPU and memory resources are made available to create KVMGuests on all nodes in the cluster.
3. VCS is installed on all the hosts to manage the KVMGuest.
4. The operating system is installed on the KVMGuest on any one host.

   Note: The KVMGuest can be created on an image file or on a shared raw disk.

5. Dummy KVMGuest is created on all other cluster nodes. See "Creating a dummy KVMGuest" on page 18.
6. KVMGuest is configured as a resource in VCS.

About configuring KVMGuest for PM-PM configuration

You must configure a KVMGuest on one node with operating system installed on a shared storage accessible to all the VCS cluster nodes.

To configure the KVMGuest as a resource-controlled through VCS, a dummy guest configuration with the same name must be created on all the subsequent nodes in the cluster. This step guarantees that the virtual guest configuration is present and is registered with the libvirtd service across all the nodes. This dummy KVMGuest does not need an OS to be present as it starts from an already-created disk image. The mount point of the disk image must be mentioned even if it is not mounted on any of the subsequent nodes, just to record the configuration. 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 KVMGuest using the standard installation procedure described in the *Red Hat Enterprise Linux Virtualization Guide*.

See "Creating a dummy KVMGuest" on page 18.

Creating a dummy KVMGuest

The following steps describe how to create a dummy KVMGuest:

1. Run the `service libvirtd start` command to start the libvirtd service.
2. From the command terminal, run the `virt-manager` command. It opens the **Virtual Machine Manager** installer window.
3. On the **Virtual Machine Manager** window, select the host on which you want to configure the KVMGuest. The corresponding host ID must be **QEMU** and the status must be **Active**.
4. Click the host name and then click **New**. This initiates Virtual Machine Creation and opens the **Create a new virtual machine** window.

5. Click **Forward** on the **Create a new virtual machine** window. The **Virtual Machine Name** page is displayed.
6. Type the KVMGuest name in the **Name** field and click **Forward**. This opens the **Virtualization Method** page.
7. On the **Virtualization Method** page:
   - Select the **Fully virtualized** option.
   - Set the appropriate **CPU architecture**.
   - Set **Hypervisor** value as **kvm**.
   - Click **Forward**. This opens the **Installation Method** page.
8. Select the appropriate installation media location, **OS Type**, **OS Variant**, and click **Forward**. This opens the **Installation Media** page corresponding to the media that you have selected.
9. Enter the installation media path and click **Forward**. This opens the **Storage** page.
10. Select the appropriate option on the **Storage** page based on the following descriptions and then click to go to the **Network** page.
    - **Block device**: This can be a local disk, a storage array LUN, or an iSCSI LUN.
    - **File (disk image)**: This is an image file residing on a file system that can be mounted over a disk from a storage array or can be shared from an NFS shared mountpoint. You must provide the name of the image file and the mount point if it is mounted on a disk other than the default location, which is `/var/lib/libvirt/images/`.
11. On the **Network** page, select the networking method you want to use to connect your KVMGuest to the host and then click **Forward**. There are two options:
    - **Virtual network**: This is a natted network which generates private IPs in the network 192.168.122.0 through the virbr0 interface. The libvirtd service creates this interface by default.
    - **Shared physical device**: This requires a bridge to be created on the host. The bridge must be bound to an Ethernet device on the host. For detailed steps on setting up a bridge, refer to the *Red Hat Enterprise Linux Virtualization Guide*.
12. On **Memory and CPU Allocation** page, specify the RAM and CPU information for your KVMGuest and then click **Forward**.

13. The **Finish Virtual Machine Creation** page provides the summary of configuration for the KVMGuest you are about to create. If the configuration is appropriate, click **Finish**. The installer creates the KVMGuest using your specified configuration.

Bundled agents for managing the KVMGuest

The Application agent can be used to manage and provide high availability for KVMGuests.

Application Agent functions

The Application agent performs the following functions:

Online:	Starts the KVMGuest.
Offline:	Shuts down the KVMGuest.
Monitor:	Monitors the KVMGuest using PidFile attribute of the Application agent to confirm whether KVMGuest is running.

Configuring VCS service groups to manage the KVMGuest

To configure KVM service groups:

1. Create a KVM service group,
2. Configure storage resources for the KVMGuest resource.
3. Configure an Application resource for KVMGuest.

Creating KVM service group

To create a KVM service group:

1. Configure a failover service group called KVM.
2. Configure all the cluster nodes (hosts) in the SystemList attribute where the KVMGuest can fail over.

Configure storage resources for the KVM Guest resource

1. If you intend to use a disk group to store the KVMGuest image file, then configure a DiskGroup resource "dg" with the diskgroup name in the DiskGroup attribute.
2. In case, the KVMGuest is installed on a VxVM volume, create a Volume resource "vol". This is the volume in which the KVM guest image file is stored.
3. Create a Mount resource “mnt” to mount the VxVM volume on the local file system.

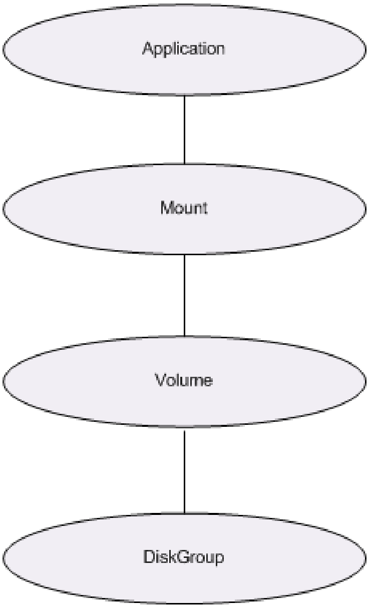
4. Create a dependency between mnt (parent) and "vol" (child).
5. Create a dependency between vol (parent) and "dg" (child).
6. Ensure that all resources are enabled before bringing them online.

Configuring Application resource for the KVMGuest

1. Create an Application resource inside this service group KVM.
2. Configure the StartProgram attribute of the Application agent as `/usr/bin/virsh start .`
3. Configure the StopProgram attribute of the Application agent as `/usr/bin/virsh shutdown <kvm_guest_name>`.
4. Configure the PidFiles attribute of the Application agent as `/var/run/libvirt/qemu/<kvm_guest_name>.pid`.

Sample dependency diagram

Figure 1-4 Sample service group for application resource managing KVMGuest



Sample configuration

```
include "types.cf"
cluster kvmtest (
        )
```

```
system sysa (
        )

system sysb (
        )

group virtgrp (
        SystemList = { sysa = 0, sysb = 1 }
        )

        Application virtappres (
                StartProgram = "/usr/bin/virsh start PM1vxfsVM1"
                StopProgram = "/usr/bin/virsh shutdown PM1vxfsVM1"
                PidFiles = { "/var/run/libvirt/qemu/PM1vxfsVM1.pid" }
                )

        DiskGroup kvmdgres (
                DiskGroup = sub_kvmdg
                PanicSystemOnDGLoss = 1
                )

        Mount kvmmountres (
                MountPoint = "/kvmmount"
                BlockDevice = "/dev/vx/dsk/sub_kvmdg/kvmdisk1"
                FSType = vxfs
                FsckOpt = "-y"
                )

        Volume kvmvolres (
                DiskGroup = sub_kvmdg
                Volume = kvmdisk1
                )

        kvmmountres requires kvmvolres
        kvmvolres requires kvmdgres
        virtappres requires kvmmountres
```

VCS cluster across physical machines (PM-PM) with resource monitoring inside KVMGuests

The physical machines (PMs) hosting VMGuests form a cluster in this configuration. Thus, the VCS cluster consists only of VMHosts. A KVMGuest cannot be a cluster node in this configuration. VCS is installed on the VMHosts in the cluster. VCS running on VMHosts monitors resources inside VMGuests with the help of the RemoteGroup agent. Thus, VCS controls and manages the virtual machines with the help of an application agent.

Network and storage configuration

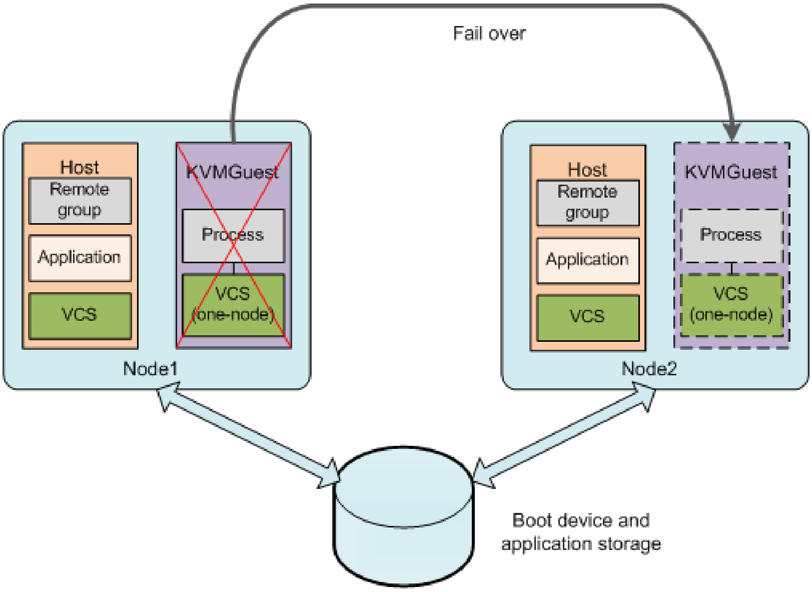
The network configuration and storage of the hosts is similar to the VCS cluster configurations. For configuration-related information, refer to the *Veritas Cluster Server Installation Guide*. However, you must set up a private link and a shared storage between the physical hosts on which the KVMGuests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

For information on RemoteGroup agent, refer to *Veritas Cluster Server Bundled Agents Reference Guide.*

VCS in the host to monitor applications in KVMGuest

The following figure illustrates the typical setup in which VCS installed in the host provides high availability to applications that run in the KVMGuests.

Figure 1-5 Typical setup for application high availability with VCS in KVMGuests



A typical two-node VCS configuration that fails over the KVMGuest so as to keep the applications that run in KVMGuest highly available has the following infrastructure:

- KVM packages are installed on each system - Node1 and Node2.
- Shared storage is attached to each system.
- A KVMGuest with same configuration details is configured on both the nodes with a shared storage.
- The operating system is installed on the KVMGuest on any one host.

Note: The KVMGuest can be created on an image file or on a shared raw disk.

- VCS is installed in the host of each node.
- Each KVMGuest has single-node VCS installed in it. VCS kernel components need not be started.

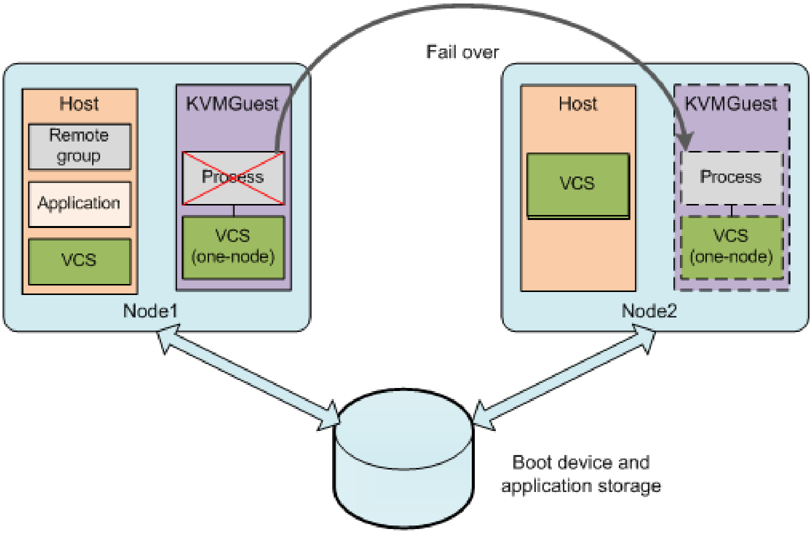
  Here the single-node VCS means the service group configured inside KVMGuest cannot fail over to any other KVMGuest or host.
- VCS service group is present for the applications that VCS must manage inside the KVMGuest.
- VCS RemoteGroup service group with an online local hard dependency to the KVMGuest service group is created to monitor the application service group that VCS is managing inside the KVMGuest.

See “Sample configuration” on page 25.

VCS setup to fail over an application on a failure

The following figure illustrates the typical VCS setup to provide high availability for applications that run inside KVMGuests.

Figure 1-6 Typical setup for application high availability with VCS in host



The typical two-node configuration where VCS keeps the applications that run in KVMGuests highly available has the following software and hardware infrastructure:

- KVM package is installed on each system Node1 and Node2.
- Shared storage is attached to each system.
- KVMGuests are created on both the nodes that may have local boot devices.
- Operating system of the KVMGuests is installed in a shared storage accessible to all the VCS cluster nodes.
- A single-node VCS is installed in each KVMGuest. VCS kernel components need not be started.

Sample configuration

Host configuration:

```
include "types.cf"

cluster kvmclus (
)
system sysA (
)
system sysB (
)
group rsg (
SystemList = { sysA = 0, sysB = 1 }
)

RemoteGroup rvg1 (
IpAddress = "192.203.47.61"
Username = vcsuser
Password = CQIoFQf
GroupName = appsg
VCSSysName = north
ControlMode = OnOff
)
requires group vmgrp online local hard

group vmgrp (
SystemList = { sysA = 0, sysB = 1 }
)
Application app1(
                StartProgram = "/usr/bin/virsh start PM1vxfsVM1"
                StopProgram = "/usr/bin/virsh shutdown PM1vxfsVM1"
                PidFiles = { "/var/run/libvirt/qemu/PM1vxfsVM1.pid" }
 )
```

KVMGuest configuration:

```
include "types.cf"

cluster appclus (
)

system north (
)
group appsg (
SystemList = { north = 0 }
)

Process proc (
PathName = /test
)
```

Limitations and unsupported KVM features

- If you have VCS running on the physical hosts, the KVMGuests get restarted as the application agent calls the shutdown and start programs. Therefore, live migration is not supported without downtime for the KVM running under VCS control.
- You must create a dummy KVMGuest on the secondary node where the service group is supposed to fail over.
- VCS does not support the virtio driver. Veritas Volume Manager does not recognize disks exported to guests over a virtio bus and VolumeSet agents cannot be used. Moreover, DiskReservation agent cannot work with disks exported over a virtio bus.

Chapter 2

Network configuration for VCS cluster across KVMGuests (KVM-KVM)

This chapter includes the following topics:

Bridge network configuration

The bridge network configuration can be performed in two parts:

- Configuring Host network
- Configuring KVMGuest network

Host network configuration

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:

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.

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:

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
   ```