



Confidence in a connected world.

# Using Stateless Linux with Veritas™ Cluster Server

## Linux

*Pranav Sarwate, Assoc SQA Engineer  
Server Availability and Management Group*

# Content

- Introduction..... **3**
- About a Stateless Linux environment..... **3**
- Communication between the Stateless Linux server and the client systems..... **3**
- Advantages of using a Stateless Linux server ..... **3**
- About setting up the Stateless Linux environment..... **4**
- About Stateless Linux with VCS..... **4**
- Setting up the Stateless Linux environment for VCS..... **5**
- Prerequisites to set up a Stateless Linux environment ..... **5**
- Setting up the Stateless Linux environment ..... **5**
- Prerequisites to configure VCS with Stateless Linux..... **8**
- Configuring VCS with Stateless Linux ..... **8**
- Troubleshooting ..... **11**
- Appendix: Implications of Stateless Linux on packages and capabilities of the operating system **13**
- Technical support..... **13**

## Introduction

This white paper describes the procedure to configure Veritas™ Cluster Server (VCS) in a Stateless Linux environment. The configuration described in this paper uses one server and two client systems.

## About a Stateless Linux environment

A Stateless Linux environment comprises of a server having the Linux operating system installed on it and client systems that are connected to the server. The client systems do not require Linux to be installed on them since they access the operating system from the server. However, the client systems must be connected to the server over a network.

The server also functions as:

- A Network File System (NFS) server that hosts the root file system of the operating system in read-only mode
- An NFS server that hosts separate disk space for each client system to store and update its read-write files
- A DHCP server and a PXE boot server

## Communication between the Stateless Linux server and the client systems

Client systems use their network card to establish communication with the Stateless Linux server. The client systems access the operating system from the Stateless Linux server by mounting it over the network, typically using NFS. As the operating system is shared among all client systems, any modification to the Stateless Linux server is reflected on all the client systems.

## Advantages of using a Stateless Linux server

The advantages of using a Stateless Linux server are as follows:

- Applications run on client systems
  - Avoids the need for huge servers with complex load balancing.
- Software and data are cached on client systems
  - Reduces bandwidth and increases speed.
  - Ensures cache data can be read-only.
- Centralized management of all deployed systems
- Storage optimization

- Easy maintenance of thin clients
- An in-time snapshot of each client system is captured

## About setting up the Stateless Linux environment

Symantec recommends that you set up the Stateless Linux environment on a prototype system and replicate the environment onto the server.

The high-level procedure to set up a Stateless Linux environment involves using a client system on the network as a prototype system to install the operating system and VCS. After setting up the environment on the prototype system, it is replicated to the server. Note that VCS needs to be configured to work as desired in a Stateless Linux environment.

## About Stateless Linux with VCS

VCS provides high availability to client systems connected to the Stateless Linux server. For example, refer to figure 1 which is a topological representation of the Stateless Linux environment. When client A goes down, the failover happens to client B.

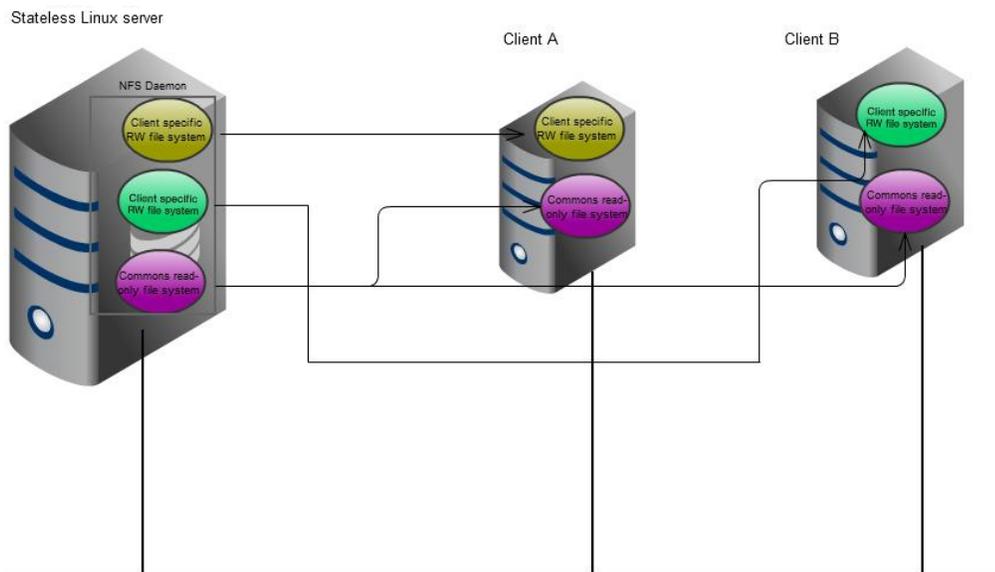


Figure 1: Stateless Linux with VCS

Setting up the Stateless Linux environment involves using one of the client systems, which acts as a prototype system (master node) with VCS installed, and which is cloned to the Stateless server. VCS is then started and made functional on the remaining client systems.

In a typical VCS environment, Symantec recommends private links between cluster nodes. However, for VCS with Stateless Linux, Symantec recommends the use of public links for the Low Latency Transport (LLT) heartbeat. This ensures that LLT is aware when the link between the cluster nodes and the server goes down. NFS also uses the public link to communicate with the cluster nodes.

By design, Global Atomic Broadcast (GAB) reports the availability of a single heartbeat between cluster nodes. This functionality of the GAB needs to be disabled for LLT to use a single public link to report heartbeat in a Stateless Linux environment. While configuring VCS for the environment, you need to disable the single heartbeat functionality of GAB.

### Setting up the Stateless Linux environment for VCS

The high-level steps to set up a Stateless Linux environment for VCS are:

- 1) Ensuring that the prerequisites to set up a Stateless Linux environment are met
- 2) Setting up the Stateless Linux server
- 3) Ensuring that the prerequisites to configure VCS are met
- 4) Configuring VCS

### Prerequisites to set up a Stateless Linux environment

Before you set up the Stateless Linux environment, ensure that network connectivity exists between the Stateless server and the client systems. This connectivity is essential for the client system to communicate its MAC address to the Stateless server over the network.

### Setting up the Stateless Linux environment

The procedure describes the steps to set up the Stateless Linux environment over a prototype system and replicating that environment to the server.

**Important:** Before proceeding make sure that you have configured the server to be a PXE/DHCP server.

To set up a Stateless Linux environment:

- 1) Ensure that the `pxe linux.0` file exists on the server with lines such as:

```
default Linux
label Linux
    kernel vmlinuz-2.6.18-53.el5
    append initrd=initrd-2.6.18-53.el5.img selinux=0
    root=Statelessserver.fqdn.ml.com:/osimages/RHCP-5.1-1.0
```

- 2) Create an `initrd` file for client systems depending on one of the following conditions:
  - If the kernel versions and hardware used on the client systems are the same, you need to create a single `initrd` file.
  - If there is a mismatch in either the kernel version or the hardware used on the client systems, you need to create separate `initrd` files for client systems.

**Note:** Ensure that you install the Network Interface Card (NIC) drivers required by the network card on the client system.

```
# mkinitrd with=e1000 --with=bnx2 \  
rootdev statelessserver.fqdn:/export-area/RHCP-5.1-1.0 \  
rootfs nfs \  
net-dev=eth0 \  
rootopts=ro,nolock \  
/tmp/initrd-2.6.18-53.1.13.el5.nfs.img \  
2.6.18-53.1.13.el5
```

(The kernel version of the operating system on the prototype server.)

**Important:** If you are unable to create the `initrd` file on a client system, use the `noncompress` option in the `mkinitrd` command to create the file on the server. Then, extract the image from the file and manually replace the MAC address of the Ethernet port with that of the client system.

- 3) Create a read-write and read-only directories under the `/stateless` directory on the server.

```
# /stateless/rw  
# /stateless/ro
```

- 4) Create client-specific directories under the `/stateless/rw` directory on the server.

```
# /stateless/rw/nodeA  
# /stateless/rw/nodeB
```

The following table lists the file-systems and the mode in which they need to be exported to the client systems.

**Table 1: File systems**

File-system	Description	Mode
/etc	Contains all the VCS configuration and licensing information	Read-only
/opt	Contains all the VCS related binaries	Read-only
/var	Log and debug information for VCS	Read-write

**Note:** Some files under the /etc and the /var directories require persistent read-write permissions.

- Clone the operating system image from a freshly built server (prototype system) with VCS installed on the Stateless server (NFS).

```
# rsync -a --exclude '/proc/*' --exclude '/sys/*' prototype
system:/ \
stateless server:/stateless/ro
```

The operating system is stored on the Stateless server (NFS). The above commands export the area to allow root access to a client. The exported area is read-only.

The Stateless server uses the DHCP protocol for network addressing.

**Note:** As the operating system is intended only for servers it contains very few graphical packages. Otherwise, it is a standard Red Hat install.

**Important:** The *rsync* command synchronizes the operating system on node A with the server. The client snapshot is stored in the */stateless* directory. You must run the *rsync* command only from the server or from a third node. Running the *rsync* command from a client can cause issues with files that are being used.

- Modify the following files in the cloned operating system on the server.

```
# echo "DEVICE=eth0" > /stateless/ro/etc/sysconfig/network-
scripts/ifcfg-eth0
# echo "ONBOOT=on" >> /stateless/ro/etc/sysconfig/network-
scripts/ifcfg-eth0
# echo "127.0.0.1 localhost" > /stateless/ro/etc/hosts
# echo "NETWORKING=yes" > /stateless/ro/etc/sysconfig/network
# echo 'none /dev/pts devpts gid=5,mode=620 0 0' >
/stateless/ro/etc/fstab
```

- Make the */etc/mtab* file a link in the */proc/mounts* directory.

```
# rm /stateless/ro/etc/mtab ; ln -s /proc/mounts /etc/mtab
```

- 8) Ensure that the `/etc/sysconfig/readonly-root` file has the following settings.

```
READONLY=yes
TEMPORARY_STATE=yes
RW_MOUNT=/var/lib/stateless/writable
RW_LABEL=stateless-rw
```

- 9) Add the following entries to get tmpfs areas mounted at boot time.

```
# echo "empty /var/log/sa" > /etc/rwtab.d/varlogsa
# echo "empty /var/log/audit" > /etc/rwtab.d/varlogaudit
# echo "empty /var/lib/statefiles" > /etc/rwtab.d/varlibstatefiles
# echo "empty /var/lib/nfs/statd" > /etc/rwtab.d/varlibstatd
# echo "dirs /var/spool" > /etc/rwtab.d/varspool
# echo "empty /var/lib/mlocate" > /etc/rwtab.d/varlibmlocate
# echo "empty /var/lib/misc" > /etc/rwtab.d/varlibmisc
```

### Prerequisites to configure VCS with Stateless Linux

Before you configure VCS with Stateless Linux, ensure that the following components are ready:

- A Stateless Linux server that has RHEL5 installed (with the latest updates)
- Two nodes that form a VCS cluster
- VCS installed on one of the client nodes

### Configuring VCS with Stateless Linux

The high-level steps to configure VCS with Stateless Linux are:

- 1) Disabling the single heartbeat availability in GAB.
- 2) Dividing the file-systems on the server based on VCS convention.
- 3) Making client-specific changes, if any.

#### ***Disabling the single heartbeat awareness in GAB***

In a typical VCS environment, the GAB file reports the availability of only one heartbeat link as a jeopardy state. However, in a Stateless Linux environment you need to disable this functionality by editing the GAB file.

To disable single heartbeat availability in GAB:

- 1) Edit the `/etc/gabtab` file.

```
# /sbin/gabconfig -s -c -n<No. Of nodes> GAB file
```

- 2) Add the following attributes to the `/opt/VRTSvcs/bin/vcsenv` file.

```
# export VCS_GAB_RMACTION=PANIC
# export VCS_GAB_TIMEOUT=60000
```

These attributes are used to decide what GAB must do if the High Availability Daemon (`_had`) fails to heartbeat with GAB. These attributes are specific to the Veritas Cluster Server 5.1 Service Pack 1 release.

For more information about attributes, see the *Veritas Cluster Server Administrator's Guide*.

### ***Dividing the file systems on the server based on the VCS convention***

The file systems categories are as follows:

- A read-only category that contains all the read-only file systems. These files are exported to client systems over NFS.
- A read-write category which contains files that VCS modifies at run-time or those files that are client-specific. This category contains files and directories that require persistent read-write permissions during reboot. Any other read-write files or directories that are required only while the client is running are mounted on `tmpfs`.

To divide file systems on the server based on the VCS convention:

- 1) Copy the `/var/VRTSvcs` directory from `/stateless/ro` to `/stateless/rw/nodeA` and to `/stateless/rw/nodeB`. Remove the contents of the `/var/VRTSvcs` directory.

```
# cp -rp /stateless/ro/var/VRTSvcs /stateless/rw/nodeA/var
# rm -fr /stateless/ro/var/VRTSvcs/
```

- 2) Copy the `vx` directory from `/stateless/ro/etc` to the client-specific directories.

```
# mkdir /stateless/rw/nodeA/etc
# mkdir /stateless/rw/nodeB/etc
# cp -rp /stateless/ro/etc/vx /stateless/rw/nodeA/etc
# cp -rp /stateless/ro/etc/vx /stateless/rw/nodeB/etc
# rm -fr /stateless/etc/vx/
```

- 3) Copy the `VRTSvcs/conf` directory from `/stateless/ro/etc` to the client-specific directories.

```
# mkdir -p /stateless/rw/nodeA/etc/VRTSvcs/conf
# mkdir -p /stateless/rw/nodeB/etc/VRTSvcs/conf
# cp -rp /stateless/ro/etc/VRTSvcs/conf/
/stateless/rw/nodeA/etc/VRTSvcs/conf
```

```
# cp -rp /stateless/ro/etc/VRTSvcs/conf/  
/stateless/rw/nodeA/etc/VRTSvcs/conf  
# rm -fr /stateless/ro/etc/VRTSvcs/conf
```

- 4) Move the llttab, gabtab, llthosts files to the /stateless/rw/etc/vx directory.

```
# mv /stateless/ro/etc/llttab /stateless/rw/etc/vx  
# mv /stateless/ro/etc/gabtab /stateless/rw/etc/vx  
# mv /stateless/ro/etc/llthosts /stateless/rw/etc/vx
```

- 5) Create link files under the /stateless/ro/etc directory.

```
# cd /stateless/ro/etc  
# ln -s vx/llttab llttab  
# ln -s vx/gabtab gabtab  
# ln -s vx/llthosts llthosts
```

- 6) Mount /lib/modules/kernel version/veritas/ on tmpfs.

```
# echo "dirs /lib/modules/kernel version/veritas/" >>  
/stateless/ro/etc/rwtab  
# echo "dirs /var/VRTSvcs/lock" >> /stateless/ro/etc/rwtab  
# echo "dirs /lib/modules/2.6.18-194.el5" >>  
/stateless/ro/etc/rwtab  
# echo "dirs /var/VRTSvcs/lock" >> /stateless/ro/etc/rwtab
```

The module files created by LLT and GAB are stored in /lib/modules/kernel version/veritas/vcs. These files require read-write permissions.

These links appear as broken links. However, they will appear on the client system as soft links after /etc/vx is mounted.

### ***Making client-specific changes***

After you clone the OS images, you need to make some client-specific changes in the respective read-write directories.

To make the client-specific changes:

- 1) Change the links that are used for LLT communication located in the /stateless/rw/node B/etc/vx/llttab directory. Change the values for all the client directories.
- 2) Add other clients with their node IDs in llttab in the /stateless/rw/<node name>/llthosts directory. This file needs to be the same for all clients.

- 3) Change the number of nodes needed for seeding in the `/stateless/rw/<nodename>/gabtab` directory.
- 4) Edit the `sysname` file and change the value to the client name (without specifying the domain) in the `/stateless/rw/<nodename>/etc/VRTSvcs/conf/sysname` directory.
- 5) Before LLT and GAB start the read-write process, create an init script to mount `/etc/vx`, `/etc/VRTSvcs/conf` and `/var/VRTSvcs` to mount the file systems before the LLT start script.

### Troubleshooting

The Stateless Linux setup requires the root file system to be mounted on NFS. When the client is up, multiple processes are running which require read-write permissions on the root file system. Any disruption in this link causes the node to stop responding. After the link is restored, the client needs a hard reboot. The client then comes up and forms GAB membership successfully. However, it may face issues in starting VCS.

- Client A crashes and comes up. If you start VCS on a client, the system displays the following error message:

```
VCS ERROR V-16-1-11009 Exiting: Another copy of VCS may be running
```

**Reason:** When client A is running, it acquires locks on certain files on the server. The lock files reside in the `/var/VRTSvcs/lock/` directory. When the client stops responding, the server does not clear these locks. Now, when the client reconnects, the server refuses the lock permission on the file.

**Workaround:** Clear all the lock files under the `/var/VRTSvcs/lock` directory for the `_had` and `_hashadow` processes and the configured agents. Restart `_had`.

- Failover fails from client A to client B and reconfiguration starts. During failover, service group 1 must come online on Client A.

**Reason:** Failover of service group 1 fails on client B that is an IP resource monitoring a virtual IP. VOIP resource is online on client A. Before attempting to failover on client B, client A comes back online. VCS restarts on client A and reconfiguration occurs, but the IP resource that monitors a virtual IP fails to come online on client A. The system displays the message "Group SG1 will not start automatically as at least one system in the System List attribute of the group is in restart mode." Service group 1 faults on client B.

**Workaround:** There is no workaround. This is the expected behavior of VCS.

- During failover, client A hangs failover of service group 1 to client B. VCS fails to come up on client A after reboot.

**Reason:** The network is up, however the NFS server does not respond. When the connection is restored and if \_hadshadow is up, it tries to start \_had, but \_had fails because the NFS server refuses lock rights to specific files.

**Workaround:** Manually remove `/var/VRTSvcs/lock/.hadlock` and restart VCS. This brings up service group 1 on other clients.

## **Appendix: Implications of Stateless Linux on packages and capabilities of the operating system**

Stateless Linux has the following implications on how specific packages in the operating system work and on the capabilities of the operating system.

- The read-only root file system of Stateless Linux enforces statelessness. However, some files do not need to be persistent. For example, the files that get written to the `/tmpfs` directory. Also, dynamically-generated files that are part of the memory do not need to be persistent.
- The Stateless Linux server needs DHCP and hardware probing capability which allows computation of configuration each time rather than having a configuration that needs to be set in advance prior to the use of the operating system by the client systems. The `initscript` dynamically determines how to set up a specific client.
- Any update to the file system on the Stateless Linux server is automatically reflected on the client servers. This is possible because of the public connection between client servers and the Stateless server. The public connection helps the server to automatically detect a client during its boot process. A client system does not require root access even though an instance of the root file system is running on its system.
- In the event a client system fails, you can replace it with a new system. The brief procedure is to plug the new client to the Stateless Linux server and select the client type. The server instantiates a copy of the operating system on the new client system. The user data of the previous client system is copied to the user directory of the new client system.

### **Technical support**

For technical assistance, visit:

[http://www.symantec.com/enterprise/support/assistance\\_care.jsp](http://www.symantec.com/enterprise/support/assistance_care.jsp)

Select phone or email support. Use the Knowledge Base search feature to access resources such as TechNotes, product alerts, software downloads, hardware compatibility lists, and our customer email notification service.

## About Symantec

Symantec is a global leader in infrastructure software, enabling businesses and consumers to have confidence in a connected world. The company helps customers protect their infrastructure, information, and interactions by delivering software and services that address risks to security, availability, compliance, and performance. Headquartered in Cupertino, Calif., Symantec has operations in 40 countries. More information is available at [www.symantec.com](http://www.symantec.com).

For specific country offices and contact numbers, please visit our Web site. For product information in the U.S., call toll-free 1 (800) 745 6054.

Symantec Corporation  
World Headquarters  
20330 Stevens Creek Boulevard  
Cupertino, CA 95014 USA  
+1 (408) 517 8000  
1 (800) 721 3934  
[www.symantec.com](http://www.symantec.com)

Copyright © 2011 Symantec Corporation. All rights reserved. Symantec and the Symantec logo are trademarks or registered trademarks of Symantec Corporation or its affiliates in the U.S. and other countries. Other names may be trademarks of their respective owners.

04/11