



# Red Hat<sup>®</sup> Linux Clusters Using a PS Series Group as Shared Storage

---

## **Abstract**

This Technical Report describes using a PS Series group as scalable and reliable shared storage in a Red Hat Linux cluster. A PS Series group is a fault-tolerant iSCSI SAN that is easy to set up, manage, and scale and increases data and application availability.

Copyright © 2004, 2005 EqualLogic, Inc.

January 2005

EqualLogic is a registered trademark of EqualLogic, Inc.

All trademarks and registered trademarks mentioned herein are the property of their respective owners.

Possession, use, or copying of the documentation or the software described in this publication is authorized only under the license agreement specified herein.

EqualLogic, Inc. will not be held liable for technical or editorial errors or omissions contained herein. The information in this document is subject to change.

PS Series Firmware Version 2.0 or later.

## Table of Contents

---

Introduction to Clustering .....	1
Using a PS Series Group as Shared Storage .....	2
Basic Steps .....	3
Setting Up PS Series Group Volumes.....	5
Restricting Access to PS Series Volumes .....	5
Installing an Initiator and Connecting to Targets.....	6
Installing the Cisco Driver for Linux .....	6
Connecting to Targets from the Cisco Driver for Linux.....	7
Configuring Persistent Bindings .....	8
Preventing Timing Issues in the rawdevices File.....	9
Testing Access to the Shared Storage .....	9
More Information and Customer Support .....	10



# Introduction to Clustering

Your business demands continuity – without putting undue pressure on staff or budget. Because customers suffer when a critical service goes offline, you must plan for unpredictable events. Now, you can meet this demand for 24x7 operation with a cost-effective solution: a Red Hat Linux cluster (running on Red Hat Enterprise Linux 3) combined with a highly available and scalable PS Series group as the shared storage.

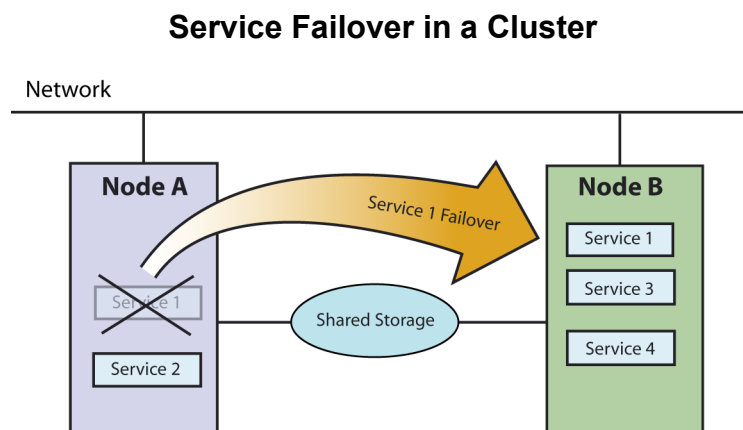
The Red Hat Cluster Manager is a collection of technologies working together to provide data integrity and the ability to maintain application availability in the event of a failure. Using redundant hardware, shared disk storage, power management, and robust cluster communication and application failover mechanisms, a cluster can meet the availability needs of a broad range of businesses.

The following list summarizes Red Hat Cluster Manager features:

- Application and data availability
- Cluster administration user interface
- Multiple cluster communication methods
- Automatic service failover capability
- Manual service relocation capability

In a Red Hat Cluster Manager configuration if the hardware or software fails and causes a service failure, the cluster will automatically restart the failed service on the functional cluster node. This service failover capability ensures that no data is lost, and there is little disruption to users. When the problem is corrected, the cluster can re-balance the services across the two nodes.

The following figure illustrates a service failover (Service 1) on a typical active-active cluster configuration in which both nodes are running different services.



This Technical Report describes using a PS Series group as scalable and reliable shared storage in a Red Hat Linux cluster. A PS Series group is a fault-tolerant iSCSI storage area network (SAN) that is easy to set up, manage, and scale and increases data and application availability.

## Using a PS Series Group as Shared Storage

---

Traditionally, many cluster solutions use direct attached storage (DAS), which requires that the cluster nodes be physically close to one another. DAS is usually SCSI-based and offers good performance; however, it has proved challenging to overcome cable length limitations and to share the storage.

In DAS configurations, the data resides on storage that is directly connected to a server and, usually, there is only one path to the data. As more servers are added and connected to the storage, management costs and complexities increase. In addition, unless the storage is highly available, a failure would cause the cluster and applications to go offline.

A PS Series group overcomes the challenges of DAS and provides a dedicated iSCSI SAN to which you can connect your cluster nodes. The basis of a group is a PS Series storage array, a no-single-point-of-failure storage device that combines reliability and scalability with an easy-to-use management interface for a single system view of the storage.

Using a PS Series group, cluster nodes can be connected to a pool of shared storage that is:

- **Highly available.** PS Series storage array hardware delivers redundant, hot-swappable components—disks, control modules, fans, and power supplies—for a no-single-point-of-failure configuration. Components fail over automatically without user intervention or disrupting data availability.
- **Scalable.** With a PS Series storage array, increasing array capacity is as easy as installing additional drives or adding network connections. You can expand group capacity—from hundreds of gigabytes to hundreds of terabytes of storage—by adding another array to a group. Automatically, the new disks and arrays are configured and the storage pool expanded.

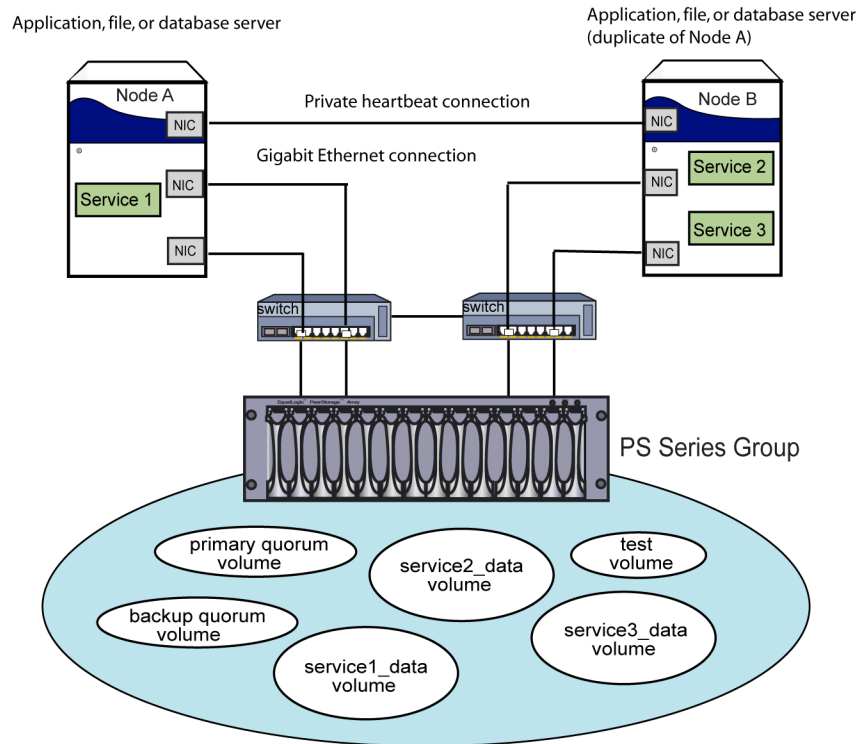
During this process, volumes remain available with no impact on hosts and applications. There is no need to open a server cabinet or reconfigure an operating system. The additional storage space is immediately available for use by any application on any server because, in a cluster, all the servers have access to all the shared storage.

- **Easy and inexpensive to manage.** Centralized storage means you can manage more storage, more efficiently. A simple setup utility lets you quickly configure an array on the network and create a group. In minutes, you have a functioning iSCSI SAN. Automation of complex operations like RAID configuration, disk sparing, data provisioning, and load balancing means that even novices can effectively manage the SAN.

Each array in a group (member) contributes to the pool of storage space. As needed, you allocate portions of the pool to volumes, specifying a size, access controls, and other attributes. Each volume can be spread across multiple disks and group members, but it is seen on the network as a single iSCSI target.

The following figure illustrates a typical active-active cluster configuration and shows a PS Series group as the shared storage.

## PS Series Group as the Shared Storage in a Red Hat Linux Cluster



---

## Basic Steps

The basic steps for using a PS Series group as shared storage in a Red Hat cluster are as follows:

1. Understand the network configuration requirements for the cluster nodes and the arrays in the group. Each node *must* have network connectivity to the group IP address. See *Testing Access to the Shared Storage* for more information.
2. Install the cluster hardware and all non-cluster software on the cluster nodes. The nodes *must* be running Red Hat Enterprise Linux 3. For details on setting up a cluster configuration, see the Red Hat documentation:

<http://www.redhat.com/docs/manuals/enterprise/RHEL-3-Manual/cluster-suite/>

In addition to the usual cluster requirements, you will need the following:

- An Ethernet connection to the group. Optimally, use a dedicated Gigabit Ethernet connection.
- An iSCSI initiator installed on each node. A group volume is seen on the network as an iSCSI target, which can only be accessed with a standards-compliant iSCSI initiator. Both hardware and software initiators are available from a variety of vendors.

For information about initiators, see *Installing an Initiator and Connecting to Targets*.

3. Configure the cluster's shared disk storage. Create the following volumes:

- Two volumes for quorum devices
- Volumes for service data, as needed
- Test volume with unrestricted access for testing purposes

For information about creating volumes and access control records, see *Setting Up PS Series Group Volumes* and *Restricting Access to PS Series Volumes*.

4. Establish a connection to the shared storage, one node at a time:

a. Use the `ping` command to ensure that each node can access the group IP address.

If a node cannot access the group IP address, see *Testing Access to the Shared Storage*.

b. Establish an iSCSI connection to the target volumes. See *Installing an Initiator and Connecting to Targets*.

If a node cannot access a target, see *Testing Access to the Shared Storage*.

c. Establish persistent bindings to the volumes as described in this document, *Configuring Persistent Bindings*.

d. Format the disks. Use normal commands, including `fdisk`.

**Note:** Do not create a file system on the quorum devices.

5. Install the cluster software on each node. Installation includes editing the `rawdevices` file to specify the quorum partitions and running `cluconfig`. In addition, when editing the `rawdevices` file, you must add timing logic as described in *Preventing Timing Issues in the rawdevices File*.

For information about installing the cluster software, see the Red Hat documentation:

<http://www.redhat.com/docs/manuals/enterprise/RHEL-3-Manual/cluster-suite/>

**Notes:** The quorum partitions must be available for the cluster to run.

6. Optionally, enable Ethernet channel bonding (recommended). This type of channel bonding (called an active-backup policy) combines two Ethernet devices into one virtual device and ensures that if one device fails, the other will become active. For details, see the Red Hat documentation:

<http://www.redhat.com/docs/manuals/enterprise/RHEL-3-Manual/cluster-suite/s1-hardware-connect.html#S2-HARDWARE-ETHBOND>

7. Test the cluster as described in the Red Hat documentation:

<http://www.redhat.com/docs/manuals/enterprise/RHEL-3-Manual/cluster-suite/s1-admin-problems.html>

## Setting Up PS Series Group Volumes

---

For detailed information about setting up and configuring a PS Series group and volumes, see the *QuickStart*.

To complete your initial cluster configuration, you must create the following:

- Two volumes, one for the primary quorum partition and one for the backup quorum partition. Each quorum partition must have a minimum size of 10 MB.

Quorum partitions are used to hold cluster state information. Periodically, each cluster system writes its status, a timestamp, and the state of its services. In addition, the quorum partitions contain a version of the cluster database. This ensures that each cluster system has a common view of the cluster configuration.

- One or more volumes for the service (application) data. You can create and modify volumes for services at any time. The size of these volumes will vary according to your application requirements.
- Test volume (only for testing purposes). You can delete the test volume after it is no longer needed (for example, after the cluster configuration is complete).

It is recommended that you select volume names that are meaningful, such as `quorum1` and `quorum2` for the quorum partitions, `data1` for a service volume, and `test` for the testing volume.

You can use the Group Manager GUI or CLI to create and modify volumes at any time.

In addition, to ensure that all cluster nodes can access the quorum and service volumes, create one or more access control records for each volume. See *Restricting Access to PS Series Volumes* for more information.

## Restricting Access to PS Series Volumes

---

Access control records are used to restrict access to data in a PS Series group. A volume and its snapshots share a list of access control records (sometimes called the access control list). You can configure a record to apply to the volume, its snapshots, or both, as needed.

The cluster software ensures that only one cluster node can access a given volume at a time. However, you *must* create access control records for the quorum partition volumes and the service volumes to allow access from the cluster nodes and to deny access to all other nodes. For the test volume, create an access control record that provides unrestricted access. Unrestricted access should *only* be used for testing purposes.

In each access control record, you can specify an IP address, iSCSI initiator name, or CHAP (Challenge Handshake Authentication Protocol) user name (or any combination). A cluster node must match *all* the requirements in *one* record in order to access the volume or snapshot. For example, if a record includes both an IP address and a CHAP user name, a cluster node must present the IP address *and* supply the CHAP user name and its associated password (using the iSCSI initiator) in order to match the record.

**Note:** If you use IP addresses or iSCSI initiator names to restrict access, create an access control record for each IP address or initiator name presented by the cluster nodes. As an example, for each NIC that is supposed to handle iSCSI traffic on a node, you could create a record and specify the IP address assigned to the NIC. This ensures that the node can access the volume or snapshot, regardless of which NIC is used for the connection.

If you use a CHAP user name to restrict access, it is recommended that you also specify an IP address in the access control record. If you only use CHAP, initiators that support discovery will attempt to log in to the target, even if they do not have the right access credentials, resulting in a large number of events logged in the group and an inefficient use of resources.

When you create a volume, you are given the opportunity to create an access control record. You can also create and modify access control records at any time.

**Note:** When using the GUI to create or modify an access control record, you can allow unrestricted access to a volume (*only* recommended for a test volume) either by selecting the `Unrestricted Access` button or by not selecting any method of restricting access. With the CLI, specify `0.0.0.0` for an IP address to allow unrestricted access.

For complete and detailed information about creating access control records, see the *Group Administration* or the *CLI Reference* manual.

---

## Installing an Initiator and Connecting to Targets

You must install and configure an iSCSI initiator on each node and connect to the PS Series volumes. A volume is seen on the network as an iSCSI target, which can only be accessed with a standards-compliant iSCSI initiator. Both hardware and software initiators are available from a variety of vendors.

The following sections describe how to install the Open Source Cisco iSCSI Driver for Linux and connect to targets. Alternately, you may be able to use iSCSI HBAs as initiators. For the latest initiator information, contact EqualLogic Customer Support.

---

### Installing the Cisco Driver for Linux

The Open Source Cisco iSCSI Driver V3.4.3 for Linux requires a host running the Linux operating system with a kernel version of 2.4.20 or later.

The Linux iSCSI Project webpage provides access to the Linux iSCSI driver, documentation, and other relevant information:

<http://linux-iscsi.sourceforge.net>

For requirements and instructions on installing the driver, see the following README:

[Installation Notes for iSCSI Driver for Linux](#)

Be sure to remove any existing iSCSI drivers from the host before installing the new driver.

**Note:** After installing and starting the initiator, examine the host's startup scripts to ensure that all dependencies are met. For example, be sure the network interfaces are started before the

iSCSI driver. Otherwise, target connection will fail. Also, examine shutdown scripts to ensure that components stop in the right order.

## Connecting to Targets from the Cisco Driver for Linux

---

After installing the Cisco iSCSI Driver for Linux, you can connect to the iSCSI targets associated with the PS Series volumes.

First, edit the `/etc/iscsi.conf` file and use `DiscoveryAddress` setting to specify one or more PS Series group IP addresses as discovery addresses. The driver will examine the file and attempt to connect and log in to the iSCSI targets presented by each group.

If desired, specify CHAP credentials (user name and password) for initiator and target authentication in the Authentication Settings section of the `/etc/iscsi.conf` file. The location of these settings in the file, in relation to the `DiscoveryAddress` settings, is important:

- Specify the CHAP credentials *before* the discovery addresses to apply them to all iSCSI targets.
- Specify the CHAP credentials *after* a discovery address and *indent* the settings to make them apply to that address.

See the [Installation Notes for iSCSI Driver for Linux](#) for details about editing the `/etc/iscsi.conf` file.

To start the iSCSI driver and discover the iSCSI targets associated with the group IP addresses specified in the `/etc/iscsi.conf` file, enter one of the following commands:

```
# /etc/init.d/iscsi start
# service iscis start
```

If the iSCSI driver is already started, enter one of the following commands:

```
# /etc/init.d/iscsi reload
# service iscsi reload
```

Once connected, targets appear as normal disks and you can partition them with the `fdisk` command. To create a file system, use the `mkfs` command. Although you can partition the quorum devices, *do not* create file systems on the partitions.

There are two device naming schemes for iSCSI targets:

- Standard Linux SCSI device names in the `/dev` directory (for example, `/dev/sda1`). It is not recommended that you use these names when referring to the devices, because Linux can assign different names to the iSCSI devices each time the iSCSI driver starts.

- Persistent device names shown as symbolic links in the `/dev/iscsi` directory (for example, `/dev/iscsi/bus0/target0/lun0/disk`). Because these names are persistent, it is recommended that you use them when referring to the devices. The symbolic links in the `/dev/iscsi` device tree are created by iSCSI startup script during each boot process.

To determine which iSCSI target is associated with a LUN, use the following command:

```
# /sbin/iscsi-ls
```

If you cannot connect to a target, see *Testing Access to the Shared Storage*.

## Configuring Persistent Bindings

Each shared storage device (group volume) in a cluster, including quorum and service volumes, must have the same path name on each cluster node. You can meet this requirement by editing the `/var/lib/iscsi/bindings` file on each cluster node.

The reference to shared storage in a cluster is by BusID, TargetID and iSCSI target name. *All three must be the same across all cluster nodes*. You can change the BusID and TargetID to make them easy to manage as long as the combination of the three is unique in that binding file. (Note that there may be non-cluster storage volumes in the file, too.) It may be easier to make changes on one node, and simply copy the information to the binding file on the other cluster node.

First, obtain the iSCSI target names for the volumes. In the Group Manager GUI, click:

```
Volumes → volume_name → Status tab
```

The following is an example of an iSCSI target name:

```
iqn.2001-05.com.equallogic:6-8a0900-236a20001-b96005ceh9c40b75-qvol1
```

For example, assume there are three shared volumes with iSCSI target names ending in `qvol1`, `qvol2`, and `dbvol`, respectively. On the first cluster node, (Node A, for example) entries in the `/var/lib/iscsi/bindings` file appear as shown next.

```
# Format:
# bus target iSCSI

# id id TargetName
#
0 0 iqn.2001-05.com.equallogic:4-6v3700-8713c0346-8d855a2e98840042-qvol1
0 1 iqn.2001-05.com.equallogic:6-8a0900-6015d0001-67855a2i42765906-qvol2
0 2 iqn.2001-05.com.equallogic:5-6d0984-4536g3444-4b434v5g23564389-dbvol
```

The three entries should appear in the binding file of the second cluster node, (Node B, for example) exactly as they appear above.

A reboot is required for the change to take effect.

In addition, the path names to shared volumes in a cluster should use the `/dev/iscsi` naming scheme instead of the standard Linux SCSI device names in the `/dev` directory (for example, `/dev/sda1`).

For example, the path to shared volume `dbvol` should be the same on both cluster nodes (that is, `/dev/iscsi/bus0/target0/lun0/disk`).

The symbolic links in the `/dev/iscsi` device tree are created by iSCSI driver during each boot process.

**Note:** The volume names must remain the same after editing the binding files on both cluster nodes. If you change the volume name, you must re-edit the binding files.

---

## Preventing Timing Issues in the `rawdevices` File

Because the `rawdevices` script can start before the iSCSI initiator establishes access to all volumes, you can encounter timing issues. To ensure that you do not encounter timing issues, you must add retry and timeout logic to the `/etc/init.d/rawdevices` file.

Locate this section of the `rawdevices` file:

```
echo "          $RAW -->  $BLOCK";
raw $RAW $BLOCK
```

Replace the previous code with the following code:

```
retries=20
while [ $retries -gt 0 ];
do
    echo "          $RAW -->  $BLOCK";
    raw $RAW $BLOCK
    if [ $? -ne 0 ]; then
        sleep 3
        retries=$((retries - 1))
        echo "*** iSCSI initiator not ready yet ..., $retries retries remain."
    else
        break
    fi
done
```

---

## Testing Access to the Shared Storage

To ensure that each initiator on each cluster node can access volumes:

- The PS Series group must be correctly connected to the network. For more information, see the Technical Report *Network Connection and Performance Guidelines* on the EqualLogic Customer Support website:

[http://support.equallogic.com/cgi-bin/pdesk.cgi?do=tech\\_report](http://support.equallogic.com/cgi-bin/pdesk.cgi?do=tech_report)

Log in to a support account and click `Tech Reports`. If you do not have an account, create one by clicking the link under the login prompt.

- The node must be able to access the group IP address. Use the `ping` command to ensure that you have network connectivity between each node and the group. If you do not have connectivity to the group IP address, examine your network configuration.
- If a node can access the unrestricted test volume but not a restricted service volume, the service volume's access control records must not be allowing the node access. See *Restricting Access to PS Series Volumes* for information about creating access control records.

You can delete the test volume after it is no longer needed for testing purposes (for example, after the cluster configuration is complete).

See also the EqualLogic Knowledge Base articles, “Recommendations for new installations” and “Initiator will not find iSCSI targets unless the network configuration is correct and the host is allowed access.” Knowledge Base articles require an EqualLogic Customer Support account:

<http://support.equallogic.com/cgi-bin/kb.cgi>

## **More Information and Customer Support**

---

Visit the EqualLogic Customer Support website, where you can download the latest documentation and firmware for PS Series storage arrays. You can also view FAQs, the Knowledge Base, and Tech Reports and submit a service request.

EqualLogic PS Series storage array documentation includes the following:

- *Release Notes*. Provides the latest information about PS Series storage arrays.
- *QuickStart*. Describes how to set up the hardware and start using PS Series storage arrays.
- *Group Administration*. Describes how to use the Group Manager GUI to manage a PS Series group. This manual provides comprehensive information about product concepts and procedures.
- *CLI Reference*. Describes how to use the Group Manager command line interface to manage a group and individual arrays.
- *Hardware Maintenance*. Provides information on maintaining the PS Series storage array hardware.

To access the Customer Support website, from the EqualLogic website ([www.equallogic.com](http://www.equallogic.com)), click `Support` and log in to a support account. If you do not have an account, create one by clicking the link under the login prompt.

To contact customer support, send e-mail to [supportnp@equallogic.com](mailto:supportnp@equallogic.com). If the issue is urgent, call 1-877-887-7337 to speak with a member of the customer support team.