



Parallels Server Bare Metal 5.0

Deploying Clusters in Parallels-Based Systems

Parallels Holdings, Ltd.
c/o Parallels International GmbH.
Parallels International GmbH
Vordergasse 49
CH8200 Schaffhausen
Switzerland
Tel: + 41 526320 411
Fax: + 41 52672 2010
www.parallels.com

Copyright © 1999-2011 Parallels Holdings, Ltd. and its affiliates. All rights reserved.

This product is protected by United States and international copyright laws. The product's underlying technology, patents, and trademarks are listed at <http://www.parallels.com/trademarks>.

Microsoft, Windows, Windows Server, Windows NT, Windows Vista, and MS-DOS are registered trademarks of Microsoft Corporation.

Apple, Mac, the Mac logo, Mac OS, iPad, iPhone, iPod touch, FaceTime HD camera and iSight are trademarks of Apple Inc., registered in the US and other countries.

Linux is a registered trademark of Linus Torvalds.

All other marks and names mentioned herein may be trademarks of their respective owners.

Contents

Introduction	4
Preparing to Create a Cluster	5
Understanding the Cluster Configuration.....	6
Setting Up Cluster Hardware	7
Installing Parallels Server Bare Metal 5.0	8
Deploying a Failover Cluster	9
Cluster Configuration Options	9
Configuring Shared Storage for Failover Clusters.....	10
Configuring iSCSI Data Storages	12
Creating and Configuring a Failover Cluster.....	14
Deploying a Data Sharing Cluster.....	18
Cluster Configuration Options	19
Configuring Shared Storages for Data Sharing Clusters.....	19
Configuring Shared Storages on NFS.....	20
Configuring Shared Storages on GFS.....	22
Creating and Configuring a Data Sharing Cluster.....	24
Managing Clusters	27
Index	28

CHAPTER 1

Introduction

In Parallels Server Bare Metal 5.0, you can use Red Hat Cluster Suite (RHCS) to deploy the following types of clusters from Parallels servers:

- *Failover cluster.* A failover cluster is used to ensure high-availability for your Parallels Server Bare Metal installations. In a failover cluster, all Parallels Server Bare Metal critical services, virtual machines, and Containers are treated by the RHCS clustering software as clustered services and, if necessary, failed over from the active cluster node to a standby one.
- *Data sharing cluster.* A data sharing cluster provides a consistent file system image across all Parallels servers in the cluster and allows the servers to simultaneously read and write to a single shared file system on a SAN data storage. A shared file system can be one of the following:
 - Network Files System (NFS)
 - Global File System version 2 (GFS v2)

In the current version of Parallels Server Bare Metal, using shared storages formatted with GFS v2 are supported for virtual machines only.

The procedure of deploying both types of RHCS clusters is described in the following sections in detail.

Note: The document provides the description specific for cluster configurations in Parallels systems only. For general information regarding Red Hat Cluster Suite, see the clustering documentation on the Red Hat web site (available at http://docs.redhat.com/docs/en-US/Red_Hat_Enterprise_Linux/).

CHAPTER 2

Preparing to Create a Cluster

This section provides general information on clusters you can deploy in Parallels systems and describes the operations you need to perform to prepare for creating a cluster.

In This Chapter

Understanding the Cluster Configuration	6
Setting Up Cluster Hardware	7
Installing Parallels Server Bare Metal 5.0	8

Understanding the Cluster Configuration

The implementation of failover and data sharing clusters in Parallels systems is based on the following statements:

- Red Hat Cluster Suite (RHCS) from the RHEL 5 Linux distribution is used as the clustering software.
- In the used clustering scenario, the *vz* and *parallels-server* services act as clustered services. These services are responsible for managing all virtual machines and Containers on your Parallels server. If the Parallels server becomes inaccessible in the case of a hardware or software failure or the *vz* and *parallels-server* services are stopped accidentally, the clustering software will try to restart these services and all virtual machines and Containers on the problem server, and if the latter is not possible, on another (passive) Parallels server in the cluster.
- A typical cluster configuration consists of 'X' + 'Y' servers where 'X' and 'Y' denote the number of active and passive Parallels servers in the cluster, respectively. It means that you can run 'X' Parallels Server Bare Metal installations and recover from 'Y' server failures. For example, if your cluster contains 3 active nodes and 1 passive node, only one active node can fail over its resources (the *vz* and *parallels-server* services and all virtual machines and Containers on that node) to the passive node.
- The minimal hardware requirements include:
 - 'X' + 'Y' physical servers
 - 'X' data storages available on all nodes in the cluster
 - 'X' + 'Y' ports on fencing devices

Cluster configuration examples: 4 active nodes and 1 passive node or 10 active nodes and 2 passive nodes.

- A single data storage is created per clustered service. In our case, each Parallels Server Bare Metal installation is regarded as a clustered service.

Setting Up Cluster Hardware

In the first step, you need to set up hardware for your cluster. This procedure consists of the following main steps:

- 1 Deciding on the hardware to be included in the cluster. The amount and type of hardware may vary depending on the purpose and availability requirements of your cluster. However, the following hardware is always present in any cluster:
 - *Cluster nodes* represented by two or more physical servers capable of running Parallels Server Bare Metal.
 - *Storage* usually represented by a common disk array on a SAN and used to store all Parallels Server Bare Metal-related data (virtual machines and Containers data, templates, backups, and so on).
 - *Switches* (Fibre Channel, iSCSI, or NFS) providing client access to the cluster and enabling the communication between each cluster node and the shared cluster storage.

Note: If you are planning to use an iSCSI or NFS storage device in your cluster, it is highly recommended to use Gigabit network adapters on your servers. This can prevent possible bottlenecks which are likely to arise in a 100-Mb network.

- 2 Connecting all hardware components into one subnet and ensuring that each hardware component can access all the other components in the subnet.

For more information about installing and configuring cluster hardware, see the clustering documentation on the Red Hat web site (available at http://docs.redhat.com/docs/en-US/Red_Hat_Enterprise_Linux/).

Installing Parallels Server Bare Metal 5.0

After you set up cluster hardware and install clustering software, you can proceed with installing Parallels Containers 4.6. The configuration and partitioning requirements for cluster nodes slightly differ from those for standalone servers. Use the guidelines below when installing Parallels Server Bare Metal on each server you plan to join to the cluster.

Follow the standard installation instructions until you are asked to partition your server hard disk drives. Use these guidelines:

- Choose to manually partition the disk drives.
- Create the root (/), `swap`, and `/boot` partitions.
- Do not create the `/vz` partition. All nodes in the cluster will use the `/vz` partition located on a shared data storage (Fibre Channel, iSCSI, or NFS) and be mounted to them.

After installing Parallels Server Bare Metal, we highly recommend that you update your software to its latest version (for example, using the `vzupdate` utility).

Deploying a Failover Cluster

This section familiarizes you with the way to use the Red Hat Cluster Suite (RHCS) software to create failover clusters from Parallels servers.

In This Chapter

Cluster Configuration Options	9
Configuring Shared Storage for Failover Clusters.....	10
Creating and Configuring a Failover Cluster.....	14

Cluster Configuration Options

Parallels Server Bare Metal allows you to create a cluster of two or more Parallels servers to provide a high degree of availability for your servers. The failover cluster implementation is based on the following statements:

- The cluster configuration is as follows:
 - A shared SCSI storage device (iSCSI or Fibre Channel) is used as a cluster storage.
 - The `/vz` partition is located on a shared SCSI storage device and formatted as ext3.
 - Clustered file systems (GFS v1, GFS v2, and NFS) are NOT used.
- One and the same version of the Parallels Server Bare Metal software is installed on all servers in the cluster.
- Only one `/vz` partition is created per clustered service.
- The following cluster resources are available in the cluster: the `/vz` filesystem, the `/etc/init.d/vz-cluster` script, and the `IP address` resource.

Configuring Shared Storage for Failover Clusters

Once you install Parallels Server Bare Metal 5.0, you need to configure the shared storage to use it as the `/vz` partition. The space requirements for this partition are calculated on the basis of space demands for virtual machines and Containers and templates. The number of partitions must correspond to the number of clustered services and be less than the number of physical servers in the cluster.

Make sure that all servers can access all shared Logical Unit Numbers (LUNs). The cluster will decide itself where to start the clustered service; so it must be able to mount the partition with all required virtual machines and Containers and templates.

Using volume labels for your partitions will greatly simplify the further management of your cluster. We recommend that you have a match between a clustered service and a partition name. For example, the `PSBM-1` service can use a partition with the `psbm-1` label. Also note that these volume labels do not correspond to mount points which are always `/vz` on all servers in the cluster.

Note: If you are planning to use iSCSI storages, see first **Configuring iSCSI Data Storages** (p. 12) for information on setting up this kind of storages. Once you are done, you can proceed with the instructions below.

Once you create all necessary LUNs, you need to create and configure partitions on them.

Creating Partitions on LUNs

First, create partitions on LUNs:

- 1 Make partitions on LUNs using the `fdisk` utility or other disk partitioning software.
- 2 Create the file system on the new partitions. For example, to create the `ext3` filesystem on the `/dev/sdc1` partition and set the `psbm-1` label for the clustered service, you can run this command:

```
# mkfs.ext3 -L psbm-1 /dev/sdc1
```

- 3 Check that all shared partitions can be successfully mounted to each cluster node. To do this, create a temporary partition on each node and mount the shared partitions (one after another) to this temporary partition. For example, you can log in to each cluster node and use the following commands to check that the partition with the `psbm-1` label can be mounted to the `/vz_temp` partition:

```
# mkdir /vz_temp  
# mount LABEL=psbm-1 /vz_temp
```

Once you check that the shared partition can be successfully mounted, unmount it:

```
# umount /vz_temp
```

Configuring Partitions on LUNs

Now that you have created all the necessary partitions, perform the following operations on each active cluster node:

- 1 Stop the `vz`, `parallels-server`, `vz1mond`, and PVA Agent services:

```
# prlsrvctl set --cluster-mode yes
# service vz stop
# service parallels-server stop
# service pvaagentd stop
# service pvapp stop
# service vz1mond stop
```

- 2 Move the `/vz` directory to a temporary directory `/vz1` and create a new `/vz` directory:

```
# mv /vz /vz1; mkdir /vz
```

- 3 Mount one of the shared LUN partitions to `/vz`, for example:

```
# mount LABEL=psbm-1 /vz
```

You need to choose a different LUN partition on each cluster node. Let us assume that you have a cluster with two active nodes and two shared partitions—`psbm-1` and `psbm-2`. In this case, if you mount the `psbm-1` partition to the `/vz` directory on the first node, you have to mount the `psbm-2` partition to the `/vz` directory on the second node.

- 4 Move all data from the temporary `/vz1` directory to `/vz`, and remove `/vz1`:

```
# mv /vz1/* /vz/
# rm -rf /vz1
```

- 5 If you plan to use Parallels Virtual Automation to manage your cluster node, also run the following command:

```
# /opt/pva/agent/bin/pvacluster.sh private
```

- 6 Unmount the shared data storage:

```
# umount /vz
```

- 7 Disable the automatic mounting of the `/vz` partition on the node start. To do this, remove the information about the `/vz` partition from the `/etc/fstab` file.

Configuring iSCSI Data Storages

If you plan to use iSCSI as a data storage, do the following:

Note: Consider using Gigabit Ethernet adapters and the corresponding network infrastructure. In most cases, a 100 Mb network is very likely to become a bottleneck.

- 1 Configure the `iscsi` service to start automatically on boot, and start it:

```
# chkconfig iscsi on
# service iscsi start
```

- 2 Search for iSCSI disks on the SAN:

```
# iscsiadm --mode discovery --type sendtargets --portal 10.0.0.10
```

where `10.0.0.10` is the IP address of the iSCSI storage device. Once the command is complete, the `iscsi` service should be able to connect to the discovered iSCSI disks on each system restart.

- 3 Restart the `iscsi` service and check if iSCSI disks are accessible (they must be listed in `/proc/partitions`):

```
# service iscsi restart
```

Now you can create partitions, filesystems on them, put LABELS on found iSCSI disks, and check if they can be mounted successfully. See [Configuring the Shared Storage for Failover Clusters](#) (p. 10) for information on how you can do it.

Setting up authentication

When configuring an iSCSI data storage, you may also need to set up authentication to allow your Parallels server to connect to the storage. To do this:

- 1 Open the `/etc/iscsi/iscsid.conf` file on the server for editing, and locate the CHAP settings section:

```
# *****
# CHAP Settings
# *****

# To enable CHAP authentication set node.session.auth.authmethod
# to CHAP. The default is None.
#node.session.auth.authmethod = CHAP
...
```

- 2 Set up authentication:

- a Uncomment the following lines:

```
#node.session.auth.authmethod = CHAP
#discovery.sendtargets.auth.authmethod = CHAP
```

- b Uncomment and set the correct parameters for the following lines:

```
#node.session.auth.username = username
#node.session.auth.password = password
#discovery.sendtargets.auth.username = username
#discovery.sendtargets.auth.password = password
```

where

- `username` and `password` in the `node.session.auth` lines denote the CHAP username and password for the initiator authentication by the target.
- `username` and `password` in the `discovery.sendtargets.auth` lines denote the initiator name discovery session CHAP username and password for the initiator.

3 Save the file.

Note: For more information on authentication parameters and their configuration, see the `iscsid.conf` file and the `iscsiadm` man pages.

Creating and Configuring a Failover Cluster

Before you start creating and configuring the failover cluster, perform the operations below.

On each active node:

- 1 Make sure the `vz` and `parallels-server` services are stopped and disable them:

```
# service vz stop
# service parallels-server stop
# chkconfig vz off
# chkconfig parallels-server off
```

- 2 Edit the `/usr/share/cluster/fs.sh` script by locating the following string in this script

```
<parameter name="mountpoint" unique="1" required="1">
```

and setting the value of the `unique` parameter to 0:

```
<parameter name="mountpoint" unique="0" required="1">
```

On each passive node:

- 1 Enable the cluster mode, stop the `vz` and `parallels-server` services if they are running, and disable them:

```
# prlsrvctl set --cluster-mode on
# service vz stop
# service parallels-server stop
# chkconfig vz off
# chkconfig parallels-server off
```

- 2 Edit the `/usr/share/cluster/fs.sh` script by locating the following string in this script

```
<parameter name="mountpoint" unique="1" required="1">
```

and setting the value of the `unique` parameter to 0:

```
<parameter name="mountpoint" unique="0" required="1">
```

- 3 If you have created the `/vz` partition when installing Parallels Server Bare Metal, also ensure that the `/vz` partition is unmounted. If it is not, unmount the partition:

```
# umount /vz
```

- 4 Disable the automatic mounting of the `/vz` partition on the node start. To do this, remove the information about the `/vz` partition from the `/etc/fstab` file.

Creating the Cluster

Now you can start creating the failover cluster. The example below demonstrates how to set up a new cluster using the `ccs_tool` command-line tool:

- 1 Log in to any of your cluster nodes, and create a configuration file for the cluster, for example:

```
# ccs_tool create psbmCluster
```

This command creates a new configuration file for the `psbmCluster` cluster with the default path `/etc/cluster/cluster.conf`.

- 2 Set fence devices for the cluster. The example below uses the `apc` network power switch as the fencing device:

```
# ccs_tool addfence apc fence_apc ipaddr=apc.server.com login="user1"
passwd="XXXXXXXX"
```

For detailed information on fence device parameters, see the *Cluster Administration* guide at http://docs.redhat.com/docs/en-US/Red_Hat_Enterprise_Linux.

Note: Manual fencing is supported for testing purposes only and is not recommended for use in production.

- 3 Add all Parallels servers to the cluster. For example, to add the `psbmNode` server to the cluster, you can execute this command:

```
# ccs_tool addnode psbmNode1 -n 1 -v 1 -f apc port=1
```

where

- `-n` specifies the ID that will uniquely identify the node among other nodes in the cluster. Each node in the cluster must have its own unique port ID.
- `-v` denotes the number of votes for the `psbmNode` server. You can use 1 as the default vote number and change it later on, if necessary.
- `apc port=1` is the name of the fencing device (APC) and the unique port ID of the APC switch power for the `psbmNode` server. Each node in the cluster must have its own unique port ID.

Run the `ccs_tool addnode` command for each node you want to add to the cluster.

- 4 Create cluster resources. For a failover cluster, you need to create the following resources: filesystem, script, and IP address.

- a** *Filesystem*. One filesystem resource per each shared SCSI partition is required. When creating the resource, make sure that you refer to the partition by the volume label rather than by the device name and specify the `noatime` option. For example:

```
# ccs_tool addfs -t ext3 -p noatime PSBM_partition_1 "LABEL=psbm-1" /vz
```

where

- `PSBM_partition_1` is the name to assign to the filesystem resource.
- `ext3` denotes the type of the filesystem.
- `LABEL=psbm-1` specifies the device associated with the filesystem resource.
- `/vz` is the path where to mount the filesystem.

- a** *Script*. One script per cluster is required. The following command creates the script resource `vzservice`:

```
# ccs_tool addscript vzservice /etc/init.d/vz-cluster
```

When creating scripts, just replace `vzservice` with your own name.

If you plan to use the Parallels Virtual Automation (PVA) application for managing your cluster nodes, you also need to create two additional script resources: `pvascr` and `ppscr`. Use the following commands to create the PVA scripts:

```
# ccs_tool pvascr /etc/init.d/pvaagentd
```

```
# ccs_tool ppscr /etc/init.d/pvapp
```

- b** *IP address.* An IP address is required for each pair of *vz* and *parallels-server* services. The IP address is used for a direct SSH connection to the server. Note that the IP address will be managed by the cluster and, therefore, must not be already in use or assigned to any node. For example, to create an IP resource, you can run this command:

```
# ccs_tool addip 10.10.10.111
```

- 5** Create and configure failover domains. You need to create one failover domain per cluster service managed by the cluster and configure the list of cluster nodes that will be able to run cluster services from these domains. For example, you can run the following command to create the failover domain `domain1` and add the nodes `psbmNode1` and `psbmNode2` to this domain:

```
# ccs_tool addfdomain domain1 psbmNode1 psbmNode2
```

- 6** Create clustered services. The number of services must correspond to the number of active servers and shared partitions. When creating a clustered service, do the following:
- Enable the service autostart.
 - Configure the service to run exclusively. This will prevent the cluster from trying to run more than one pair of *vz* and *parallels-server* services on the same physical server.
 - Set the service recovery policy to `relocate` or `restart`. In the latter case, if the *vz* and *parallels-server* services are stopped for some reason, the cluster will try to restart this service on the same server before relocating it to another one.
 - Specify the IP address resource for the service.
 - Specify the proper failover domain.
 - Specify the script and filesystem resources for the service. This will ensure the proper start order of the services.

If you plan to use the Parallels Virtual Automation application for managing your cluster nodes, you also need to indicate the PVA scripts.

- Specify the name to assign to the service.

For example:

```
# ccs_tool addservice -a 1 -x 1 -r restart -i 10.10.10.200 -d domain1 -s vzservice -s pvascr -s ppscr -f PSBM_partition_1 pservice1
```

This command creates the clustered service `pservice1`; enables its autostart (`-a 1`); configures the service to run exclusively (`-x 1`); sets its recovery policy to `restart` (`-r restart`); assigns IP address `10.10.10.200` to the service (`-i 10.10.10.200`); and associates the service with the failover domain `domain1` (`-d domain1`), the scripts `vzservice`, `pvascr`, `ppschr` (`-s vzservice`, `-s pvascr`, `-s ppschr`), and the filesystem resource `PSBM_partition_1` (`-f PSBM_partition_1`).

Configuring the Cluster

Once you create the cluster and set up Parallels Virtual Automation, configure the cluster as follows:

- 1 Distribute the configuration file (`/etc/cluster/cluster.conf`) to all servers, and start the clustering services on two cluster nodes:

```
# service cman start
# service rgmanager start
```

Try to start the clustering services simultaneously (or as fast as possible, one after another) on both nodes.

- 2 Wait for the clustering services you started on two cluster nodes to restart all the other nodes in the cluster.
- 3 Once the nodes are restarted, start the clustering services on these nodes:

```
# service cman start
# service rgmanager start
```

- 4 Configure the clustering service on each node in the cluster to start in the default runlevel. For example, if your system default runlevel is set to 3, you can enable the service by executing the following commands on each of the cluster nodes:

```
# chkconfig --level 3 cman on
# chkconfig --level 3 rgmanager on
```

- 5 Use the `clustat` utility (you can run it on any cluster node) to make sure that all the services have been successfully started. If they have not, investigate the cluster logs stored in `/var/log/messages` by default. Keep in mind that the information you are looking for may be placed on different servers in the cluster.
- 6 If you plan to use Parallels Virtual Automation for managing your cluster, log in to the PVA Management Node, and register all clustered services. To register a clustered service, use the IP address you assigned to the service when creating the cluster.

Configuring Containers in Failover Clusters

When creating Containers on nodes participating in a failover cluster, make sure that the `ONBOOT` parameter in their configuration files (`/etc/vz/conf/CT_ID.conf`) is set to `yes`. If this parameter is set to `yes` for a Container, the Container is automatically started once it is failed over from one cluster node to another; otherwise, the Container remains in the stopped state.

If the `ONBOOT` parameter is set to `no`, change it to `yes` using this command:

```
# pct1 set CT_ID --onboot yes --save
```

Deploying a Data Sharing Cluster

Data sharing clusters are server farms that share storage devices on a storage area network (SAN) and share data on those storage devices. In a data sharing cluster, data can be written or read by any server to or from any file on a common shared storage device. Parallels Server Bare Metal allows you to deploy data sharing clusters in your working environments by combining the functionality provided by a shared file system with that of Red Hat Cluster Suite. By implementing a data sharing cluster made up of Parallels servers, you can achieve the following goals:

- Simplify the process of managing your virtual machines and Containers and templates because all virtual machines and Containers and templates are residing on a single SAN storage shared by all servers in the cluster.
- Greatly speed up the process of migrating running virtual machines and Containers between the cluster nodes. In fact, the migration is almost imperceptible to users since all virtual machine and Container data in the cluster is stored on a shared SAN storage and there is no need to move this data between the nodes during the migration.
- Provide failover support for Parallels Server Bare Metal vital services, virtual machines, and Containers. Each server in the cluster is running the clustering software responsible for monitoring the health of Parallels Server Bare Metal installations and failing over the services and virtual machines and Containers from the failed node to a passive node.

The following subsections provide detailed information on how to deploy a data sharing cluster using Red Hat Cluster Suite.

In This Chapter

Cluster Configuration Options	19
Configuring Shared Storages for Data Sharing Clusters	19
Creating and Configuring a Data Sharing Cluster	24

Cluster Configuration Options

The implementation of a data sharing cluster is based on the following statements:

- The cluster configuration is as follows:
 - a** A shared storage is used as a cluster storage.
 - b** The shared storage is mounted simultaneously to `/vz` on all cluster nodes.
 - c** A clustered file system (GFS v2 or NFS) is used.

Note: In the current version of Parallels Server Bare Metal, using shared storages formatted with GFS v2 is supported for virtual machines only.

- One and the same version of the Parallels Server Bare Metal software is installed on all servers in the cluster.
- Only one `/vz` partition is created.
- The following cluster resources are present in the cluster: the `/etc/init.d/vz-cluster` script and the `IP address` resource (not mandatory).

Configuring Shared Storages for Data Sharing Clusters

Once you have set up Parallels Server Bare Metal on each cluster node, you need to configure a data storage (iSCSI, Fibre Channel, or NFS) for the cluster. This data storage will be then used for hosting the `/vz` partition with all virtual machine and Container data and templates from all Parallels servers in the cluster and will be shared by all cluster nodes. The shared data storage can reside on an NFS or GFS v2 partition.

Note: In the current version of Parallels Server Bare Metal, using shared storages formatted with GFS v2 is supported for storing virtual machines only.

Configuring Shared Storages on NFS

The process of configuring a data shared storage located on an NFS volume includes the following steps:

- 1 Configuring the data shared storage for the first node in the cluster.
- 2 Configuring the data shared storage for all the other nodes in the cluster.

Configuring the Shared Storage for the First Cluster Node

To configure the data shared storage for the first node in the cluster, do the following:

- 1 Log in to any of your cluster nodes.
- 2 Make sure that all NFS-related services are enabled on the node:

```
# chkconfig rpcbind on
# chkconfig nfslock on
# chkconfig nfs on
# chkconfig netfs on
```

- 3 Stop the *vz*, *parallels-server*, and PVA Agent services:

```
# prlsrvctl set --cluster-mode yes
# service vz stop
# service parallels-server stop
# service pvaagentd stop
# service pvapp stop
```

- 4 Move the */vz* directory to a temporary directory */vz1* and create a new */vz* directory:

```
# mv /vz /vz1; mkdir /vz
```

- 5 Mount the shared data storage located on the NFS volume to */vz*:

```
# service rpcbind start
# service nfslock start
# mount SERVER:/PATH /vz
```

where *SERVER* is the IP or hostname of your NFS server and *PATH* denotes the path to the shared directory on the NFS server.

- 6 Move all data from the temporary */vz1* directory to */vz*, and remove */vz1*:

```
# mv /vz1/* /vz/
# rm -rf /vz1
```

- 7 Configure the shared data storage to be automatically mounted to */vz* on the system boot. To do this, add the */vz* entry to the */etc/fstab* file:

```
SERVER:PATH /vz nfs defaults,noatime 0 0
```

Configuring the Shared Storage for Other Cluster Nodes

To configure the data shared storage for the second and all subsequent nodes in the cluster, do the following:

- 1 Log in to the respective node.
- 2 Make sure that all NFS-related services are enabled on the node:

```
# chkconfig rpcbind on
# chkconfig nfslock on
# chkconfig nfs on
# chkconfig netfs on
```

3 Stop the `vz` and `parallels-server` services:

```
# prlsrvctl set --cluster-mode on
# service vz stop
# service parallels-server stop
```

4 Remove the contents of the `/vz` directory:

```
# rm -rf /vz/*
```

5 Configure the shared data storage to be automatically mounted to `/vz` on the system boot. To do this, add the `/vz` entry to the `/etc/fstab` file:

```
SERVER:PATH /vz nfs defaults,noatime 0 0
```

Once you have configured all your cluster nodes, reboot them and make sure each node has the shared data storage mounted to `/vz`.

Configuring Shared Storages on GFS

Setting up a shared data storage located on a partition formatted with GFS v2 is supported for virtual machines only and includes the following steps:

- 1 Configuring the data storage for the first node in the cluster.
- 2 Configuring the data storage for all the other nodes in the cluster.

Note: Parallels Server Bare Metal 5.0 does not support using shared storages formatted with GFS v1.

Configuring the Data Storage for the First Node in the Cluster

To configure the shared data storage for the first node in the cluster, do the following:

- 1 Log in to any of your cluster nodes.
- 2 Use standard Linux tools, such as Logical Volume Manager, to set up a logical volume (for example, `/dev/vg01/lv01`) on your data storage. This logical volume will host the `/vz` partition. Please notice that one logical volume is required for each Red Hat GFS file system.

Note: If you are going to use Logical Volume Manager (LVM) for creating logical volumes, make sure that it is configured with the clustered locking support. Otherwise, the LVM metadata may become corrupted. For detailed information on LVM and its configuration settings, turn to the LVM documentation and `lvman.conf` man pages.

For example:

```
# pvcreate /dev/sdb1
# vgcreate vg01 /dev/sdb1
# lvcreate -l 100%VG -n lv01 vg01
```

- 3 Create a GFS file system on the logical volume using the `gfs2_mkfs` utility. For example, you can run the following command to do this:

```
# gfs2_mkfs -p lock_dlm -t psbmCluster:gfs2 -j 4 /dev/vg01/lv01
```

In this example:

- `-p lock_dlm` denotes the name of the locking protocol that will be used by the GFS file system. The currently recognized cluster-locking protocols include `lock_dlm` and `lock_nolock`.
- `-t psbmCluster:gfs2` denotes the name of the cluster (`psbmCluster`) for which the GFS file system is created and the name that will be assigned to the GFS file system (`gfs2`).

Note: Keep in mind that you will need to specify this name when creating a cluster configuration.

- `-j 4` is the number of journals that will be created by the `gfs2_mkfs` utility. When deciding on the number of journals, keep in mind that one journal is required for each cluster node which is to mount the GFS file system. You can also make additional journals at the time of the GFS file system creation to reserve them for future use.
- `/vg01/lv01` denotes the logical volume where the GFS file system is to be located.

As a result of the aforementioned command, a new GFS file system with the `gfs2` name for the `psbmCluster` cluster will be created. The file system will use the `lock_dlm` protocol, contain 4 journals, and reside on the `/vg01/lv01` volume.

- 4 Make sure that the created logical volumes can be accessed by all servers in the cluster. This ensures that the clustering software can mount the `/vz` partition that you will create on the logical volume in the next step to any of your cluster nodes.
- 5 Tell the node to automatically mount the `/vz` partition on the node boot. To do this, add the `/vz` entry to the `/etc/fstab` file on the node. Assuming that your GFS file system resides on the `/vg01/lv01` logical volume, you can add the following entry to the `fstab` file:

```
/dev/vg01/lv01 /vz gfs2 defaults,noatime 0 0
```

If you use LVM on a GFS filesystem over a partition provided via the iSCSI protocol, you need to define the extra option `_netdev` in `/etc/fstab` in the order LVM tools search for the volumes after network filesystems are initialized.

```
/dev/vg01/lv01 /vz gfs2 defaults,noatime,_netdev 0 0
```

Also make sure that the `netfs` service is enabled by default.

```
# chkconfig netfs on
```

- 6 Configure the `gfs2` service on the node to start in the default runlevel. You can enable the `gfs2` service by executing the following command on each of the cluster nodes:

```
# chkconfig --level 3 gfs2 on
```

- 7 Stop the `vz`, `parallels-server`, and PVA Agent services:

```
# prlsrvctl set --cluster-mode yes
# service vz stop
# service parallels-server stop
# service pvaagentd stop
# service pvapp stop
```

- 8 Move `/vz` to a temporary directory `/vz1`, and create a new `/vz` directory:

```
# mv /vz /vz1; mkdir /vz
```

Later on, you will mount a shared data storage located on a GFS volume to the created `/vz` directory and move there all data from the `/vz1` directory.

Configuring the Data Storage for Other Nodes in the Cluster

To configure the shared data storage for the second and all subsequent nodes in the cluster, do the following:

- 1 Tell each node in the cluster to automatically mount the `/vz` partition on the node boot. To do this, add the `/vz` entry to the `/etc/fstab` file on each node in the cluster. Assuming that your GFS file system resides on the `/vg01/lv01` logical volume, you can add the following entry to the `fstab` file:

```
/dev/vg01/lv01 /vz gfs2 defaults,noatime 0 0
```

- 2 Configure the `gfs` service on each node in the cluster to start in the default runlevel. For example, if your system default runlevel is set to 3, you can enable the `gfs` service by executing the following command on each of the cluster nodes:

```
# chkconfig --level 3 gfs2 on
```

Creating and Configuring a Data Sharing Cluster

Before creating and configuring a data sharing cluster, enable the cluster mode on all cluster nodes and make sure that the `vz` and `parallels-server` services are stopped:

```
# prlsrvctl set --cluster-mode on
# service vz stop
# service parallels-server stop
```

Creating the Cluster

Now you can start creating the data sharing cluster. The example below demonstrates how to set up a new cluster using the `ccs_tool` command-line tool. Using this tool, you can create two types of data sharing clusters:

- *Active/passive clusters.* An active/passive cluster includes both active and passive nodes. In this type of cluster, a passive node is used only if one of the active nodes fails.
- *Active/active clusters.* An active/active cluster consists of active nodes only, each running the Parallels Server Bare Metal software and hosting a number of virtual machines and Containers. In the event of a failover, all virtual machines and Containers running on the problem node are failed over to one of the healthy active nodes.

The process of creating both types of clusters is almost identical and is described below:

- 1 Log in to any of your cluster nodes, and create a configuration file for the cluster, for example:

```
# ccs_tool create psbmCluster
```

This command creates a new configuration file for the `psbmCluster` cluster with the default path `/etc/cluster/cluster.conf`.

- 2 Set fence devices for the cluster. The example below uses the `apc` network power switch as the fencing device:

```
# ccs_tool addfence apc fence_apc ipaddr=apc.server.com login="user1"
passwd="XXXXXXXX"
```

For detailed information on fence device parameters, see the *Cluster Administration* guide at http://docs.redhat.com/docs/en-US/Red_Hat_Enterprise_Linux.

Note: Manual fencing is supported for testing purposes only and is not recommended for use in production

- 3 Add all Parallels servers to the cluster. For example, to add the `psbmNode1` server to the cluster, execute this command:

```
# ccs_tool addnode psbmNode1 -n 1 -v 1 -f apc port=1
```

where

- `-n` specifies the ID that will uniquely identify the node among other nodes in the cluster. Each node in the cluster must have its own unique port ID.
- `-v` denotes the number of votes for the `psbmNode` server. You can use 1 as the default vote number and change it later on, if necessary.

- `apc port=1` is the name of the fencing device (APC) and the unique port ID of the APC switch power for the `psbmNode` server. Each node in the cluster must have its own unique port ID.

Run the `ccs_tool addnode` command for each node you want to add to the cluster.

- 4 Create cluster resources. For a data sharing cluster, you need to create two resources—script and IP address.

- a *Script.* One script per cluster is required. The following command creates the script resource `vzservice`:

```
# ccs_tool addscript vzservice /etc/init.d/vz-cluster
```

When creating scripts, just replace `vzservice` with your own name.

- b *IP address.* An IP address is required for each pair of `vz` and `parallels-server` services. This IP address is used for a direct SSH connection to the server. Note that the IP address will be managed by the cluster and, therefore, must not be already in use or assigned to any node. For example:

```
# ccs_tool addip 10.10.10.111
```

- 5 Create and configure failover domains. You need to create one failover domain per cluster service managed by the cluster and configure the list of cluster nodes that will be able to run cluster services from these domains. For example, you can run the following command to create the failover domain `domain1` and add the nodes `psbmNode1` and `psbmNode2` to this domain:

```
# ccs_tool addfdomain domain1 psbmNode1 psbmNode2
```

- 6 Create clustered services. The number of services must correspond to the number of active servers and shared partitions. When creating a clustered service, do the following:

- a Enable the service autostart.
- b Configure the service operation mode:
 - For an active/passive cluster, configure the service to run exclusively. This will prevent the cluster from trying to run more than one pair of `vz` and `parallels-server` services on the same physical server.
 - For an active/active cluster, make sure that the service is configured not to run exclusively so that more than one `vz` service will be able to run on the same node.
- a Set the service recovery policy to `relocate` or `restart`. In the latter case, if the `vz` and `parallels-server` services are stopped for some reason, the cluster will try to restart these services on the same server before relocating it to another one.
- b Specify the IP address resource for the service.
- c Specify the proper failover domain.
- d Specify the script resource for the service.
- e Specify the name to assign to the service.

For example:

```
# ccs_tool addservice -a 1 -x 1 -r restart -i 10.10.10.200 -d domain1 -s vzservice  
pservice1
```

This command creates the clustered service `p_service1`; enables its autostart (`-a 1`); configures the service to run exclusively (`-x 1`); sets its recovery policy to `restart` (`-r restart`); assigns IP address `10.10.10.200` to the service (`-i 10.10.10.200`); and associates the service with the failover domain `domain1` (`-d domain1`) and the script `vzservice` (`-s vzservice`).

Configuring the Cluster

Once you create the cluster configuration, distribute the configuration file (`/etc/cluster/cluster.conf`) to all cluster nodes, and start the `cman` service on each cluster node one after another.

```
# service cman start
```

Once the `cman` service successfully starts on all cluster nodes, complete the following tasks on each node:

- 1 Start the `gfs2` service:

```
# service gfs2 start
```

- 2 Move all the data from the temporary `/vz1` directory to `/vz`, and then remove `/vz1`:

```
# mv /vz1/* /vz/  
# rm -rf /vz1
```

- 3 Start the `rgmanager` service:

```
# service rgmanager start
```

- 4 Configure the clustering service to start in the default runlevel. For example, if your system default runlevel is set to 3, you can enable the service by executing the following commands on each of the cluster nodes:

```
# chkconfig --level 3 cman on  
# chkconfig --level 3 rgmanager on
```

Once you perform the operations above, use the `clustat` utility (you can run it on any cluster node) to make sure that all the services have been successfully started. If they have not, investigate the cluster logs stored in `/var/log/messages` by default. Keep in mind that the information you are looking for may be placed on different servers in the cluster.

Configuring Parallels Virtual Automation

If you plan to use the Parallels Virtual Automation application for managing your cluster nodes:

- 1 On each cluster node, start the PVA Agent services, and configure them to automatically start when you restart the nodes:

```
# service pvaagentd start  
# service pvapp start  
# chkconfig pvaagentd on  
# chkconfig pvapp on
```

- 2 Log in to the Parallels Virtual Automation Management Node.

- 3 Register all cluster nodes one after another. To register a node, use its IP address.

CHAPTER 5

Managing Clusters

Once you create the cluster, you can start managing it using the `clusvcadm` and `clustat` utilities. Below are some examples of using these utilities:

- Displaying the status of a clustered service:

```
# clustat -s <service_name>
```

- Restarting a clustered service:

```
# clusvcadm -R <service_name>
```

- Stopping a clustered service:

```
# clusvcadm -s <service_name>
```

- Relocating a clustered service from one cluster node to another:

```
# clusvcadm -r <service_name> - m <member>
```

For more information on both utilities, see their man pages.

Testing Clusters

To test the created cluster, you can do the following:

- Use the `clusvcadm` utility to relocate a clustered service from one node to another:

```
# clusvcadm -r <service_name> - m <member>
```

- Shut down or unplug a power cord on a cluster node.

If the cluster is operating correctly, the `vz` and `parallels-server` services are relocated from the source server to the destination one (if you are using manual fencing, you may need to use the `fence_ack_manual` tool to inform the cluster of the fenced node). If some service has failed to start, investigate the cluster logs stored in `/var/log/messages` by default. Please keep in mind that the information you are looking for may be placed on different cluster nodes.

Index

C

- Cluster Configuration Options - 9, 19
- Configuring iSCSI Data Storages - 12
- Configuring Shared Storage for Failover Clusters - 10
- Configuring Shared Storages for Data Sharing Clusters - 19
- Configuring Shared Storages on GFS - 22
- Configuring Shared Storages on NFS - 20
- Creating and Configuring a Data Sharing Cluster - 24
- Creating and Configuring a Failover Cluster - 14

D

- Deploying a Data Sharing Cluster - 18
- Deploying a Failover Cluster - 9

I

- Installing Parallels Server Bare Metal 5.0 - 8
- Introduction - 4

M

- Managing Clusters - 27

P

- Preparing to Create a Cluster - 5

S

- Setting Up Cluster Hardware - 7

U

- Understanding the Cluster Configuration - 6