Parallels

Parallels Server 4 Bare Metal

Deploying Clusters in Parallels-Based Systems

ISBN: N/A
Parallels Holdings, Ltd.
c/o Parallels Software, Inc.
13755 Sunrise Valley Drive
Suite 600
Herndon, VA 20171
USA
Tel: +1 (703) 815 5670
Fax: +1 (703) 815 5675

Contents

Introduction

Parallels Server 4 Bare Metal allows you to use Red Hat Cluster Suite (RHCS) on Parallels servers for deploying the following types of clusters:

- A failover cluster. This type of cluster is used to ensure high-availability for your Parallels Server Bare Metal installations. In a failover cluster, each Parallels Server Bare Metal critical service and Container is treated by the RHCS clustering software as a clustered service and, if necessary, failed over from the active cluster node to a standby one.
- A data sharing cluster. This type of 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
  - NFS (Network Files System)
  - GFS (Global File System)

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, refer to the *Red Hat Cluster Suite: Configuring and Managing a Cluster* guide (available at http://www.redhat.com/docs/manuals/csgfs).

Preparing for Creating 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.

General Information on Clusters

The implementation of a failover or data sharing cluster 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` service acts as a clustered service. This service is responsible for managing all Containers on your Parallels server. If the Parallels server becomes inaccessible in the case of a hardware or software failure or the `vz` service is stopped accidentally, the clustering software will try to restart the service and all the Containers on the problem server, and if the latter is not possible, on another (passive) Parallels server in the cluster.

  Note: This version of Parallels Server Bare Metal does not provide the automatic failover of virtual machines.

- 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 will be able to fail its resources (i.e. the `vz` service and all 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 should set up hardware for your cluster. This procedure consists of the following main steps:

- 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 (Container data, templates, backups, etc.).
  - *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, you are highly recommended using Gigabit network adapters on your servers. This will allow you to prevent possible bottlenecks which are likely to arise in a 100-Mb network.
- Uniting all hardware components into one subnet and ensuring that each hardware component can access all the other components in the subnet.

Note: You will also need a server running RHEL 5 or CentOS 5 operating system and having the `X-Window System` component installed. You will run the Cluster Configuration Tool on this server to create and configure your cluster.

For more information about installing and configuring cluster hardware, refer to the *Configuring and Managing a Red Hat Cluster* document (available at https://www.redhat.com/docs/manuals/enterprise/).

Installing Parallels Server Bare Metal

In the case of a failover cluster, the server configuration and partitioning requirements slightly differ from those used for a standalone installation. Use the guidelines below when installing Parallels Server Bare Metal on each of the servers to be included in 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 `swap` and `/boot` partitions.
- Make sure that at least 10 GB of disk space is allocated to the root (`/`) partition.
- 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 mounted to them.

After installing Parallels Server Bare Metal, we highly recommend that you update your server software to its latest version (e.g. using the `vzup2date` 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 your Parallels servers.

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.
  - The clustered file system (GFS and NFS) is 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 present in the cluster: the `/vz` file system, the `/etc/init.d/vz-cluster` script, and (optionally) the `IP address` resource.

Configuring the Shared Storage

After installing Parallels Server Bare Metal, you need to configure the shared storage you plan to use as the `/vz` partition. The shared storage can be configured from any node in the cluster. The space requirements for this partition should be calculated on the basis of the space demands for your Containers and Parallels templates. The number of partitions must correspond to the number of clustered services and be less than the number of physical servers in your cluster.

Keep in mind that all your servers must be able to access your shared LUNs. The cluster will decide where to start the clustered service by itself, so it must be able to mount the partition with all the required 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 the servers in the cluster.

After the corresponding LUNs have been successfully created, create partitions on them using the `fdisk` utility or other disk partitioning software, and then create the file system. For example:

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

Do not forget to replace `/dev/sdc1` in the example above with your real disk device and `psbm-1` with a label set for a particular clustered service.

Now check that the device can be successfully mounted on all cluster nodes:

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

After that, unmount the device:

```
# umount /vz
```

Configuring iSCSI Data Storages

If you plan to use iSCSI as a data storage, perform the following operations before creating the cluster:

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.

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

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

- Search for iSCSI disks on the SAN:

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

  where `10.0.0.10` is an IP address of the SAN provided data via the iSCSI protocol.

  Now the `iscsi` service must be able to connect to the discovered iSCSI disks on each system restart.

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

```
# service iscsi restart
```

The above configuration steps must be performed on each cluster node. After that, 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 (p. 9)).

Creating and Configuring the Failover Cluster

Before creating and configuring the failover cluster, perform the following operations on each cluster node:

- Disable the automatic start of the `vz` service (it will be managed by the cluster script):

```
# service vz off
```

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

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

and changing it as follows:

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

Now you can start creating the failover cluster. The example below demonstrates how to set up a new cluster using the `system-config-cluster` cluster configuration GUI tool:

1. Log in to any server that runs the RHEL 5 or CentOS 5 operating system and have the `X-Window System` component installed, and launch the Cluster Configuration Tool as follows:

```
# system-config-cluster
```

2. Your further steps will be different depending on whether you are starting the tool for the first time or have already launched it before. Click Create New Configuration in the displayed dialog window if you are running the Cluster Configuration Tool for the first time. Otherwise, click File > New.

3. In the New Configuration window, enter a descriptive name for the cluster in the field provided, and click OK.

Note: The dlm lock manager has been tested and recommended for the production usage; other locking schemes (quorum disk, etc.) are not supported.

4. Set one or more fence devices for the cluster:

   a. Select the Fence Devices item, and click Add a Fence Device at the bottom of the Cluster Configuration window:

b In the Fence Device Configuration window, select the type of fence device to configure, and provide the necessary information in the appropriate fields.

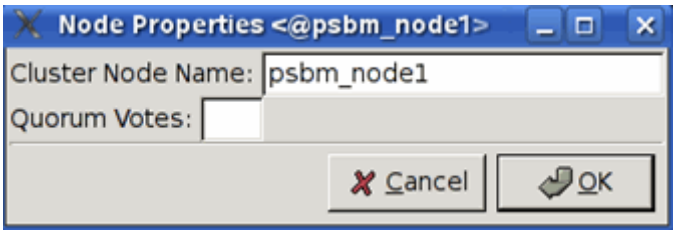
Note: Refer to the *Configuring and Managing a Red Hat Cluster* document for detailed information on available fence devices and their parameters. Manual fencing is supported for testing purposes only and should not be used in production.

c Click OK.

5 Add all your Parallels servers to the cluster:

a Select the Cluster Nodes item, and click Add a Cluster Node at the bottom of the Cluster Configuration window.

b In the Node Properties window, type the name of the server you want to add to the cluster (or its IP address on the cluster subnet), and click OK. For example:

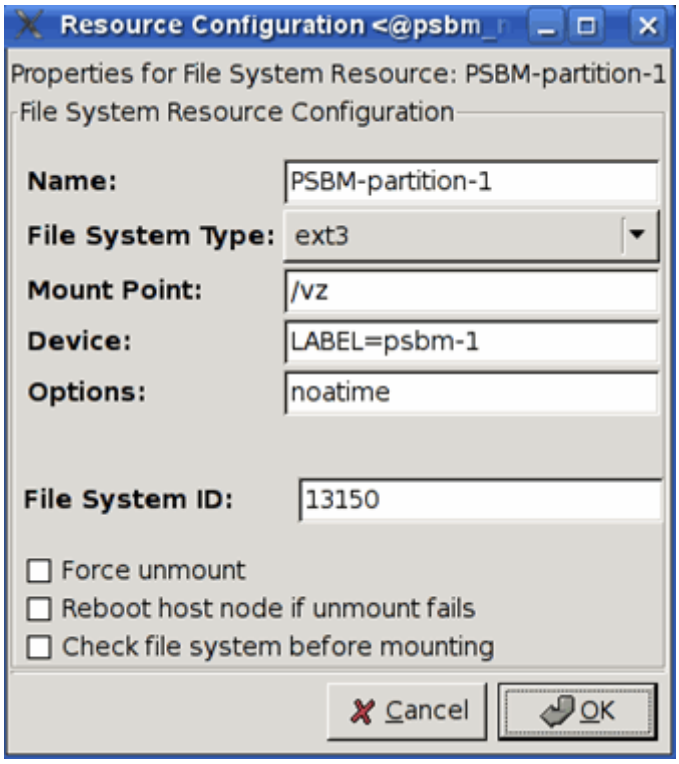


c Configure fencing for the added node using the Manage Fencing For This Node button at the bottom of the Cluster Configuration right frame.

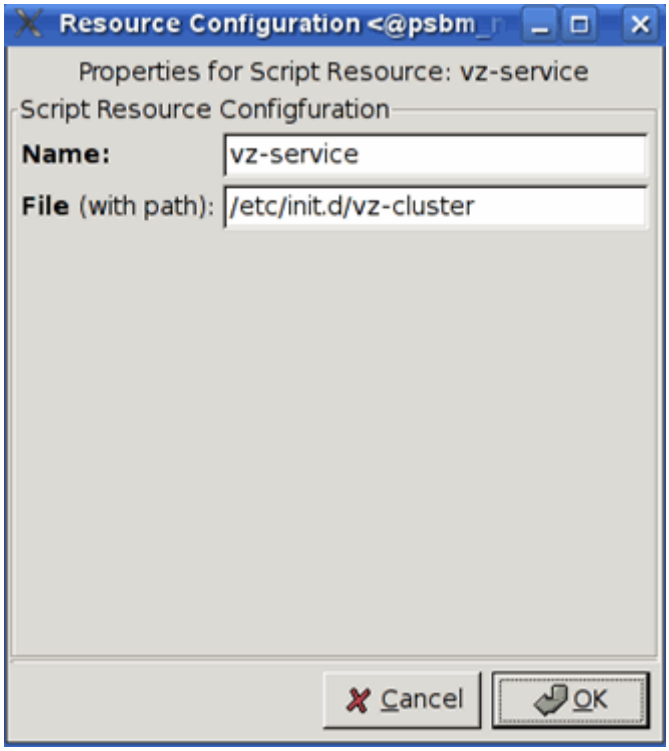
d Repeat the aforementioned operations for each server to be included in the cluster.

6 Create resources:

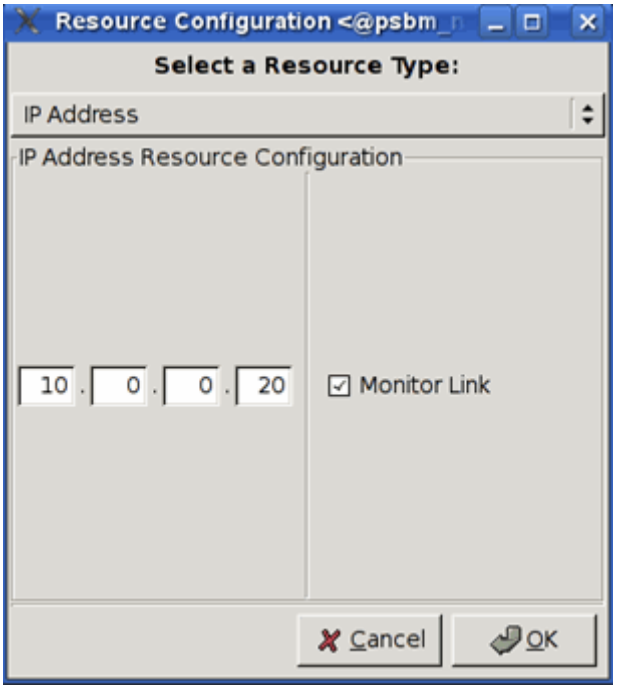
- *File system*: one resource per each shared SCSI partition. Make sure that the partition is referred to by the volume label rather than by the device name. Also, do not forget to use the `noatime` option. See the picture below:



- *Script*: one script per cluster:

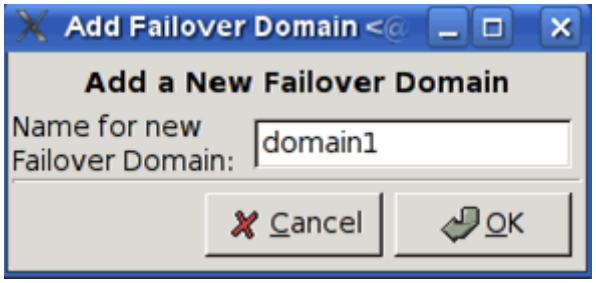


- (Optional) *IP address*: an IP address is needed for each `vz` service (it will be used for a direct SSH connection to the host). Note that the IP address will be managed by the cluster, so it must not be already in use or assigned to the nodes directly:

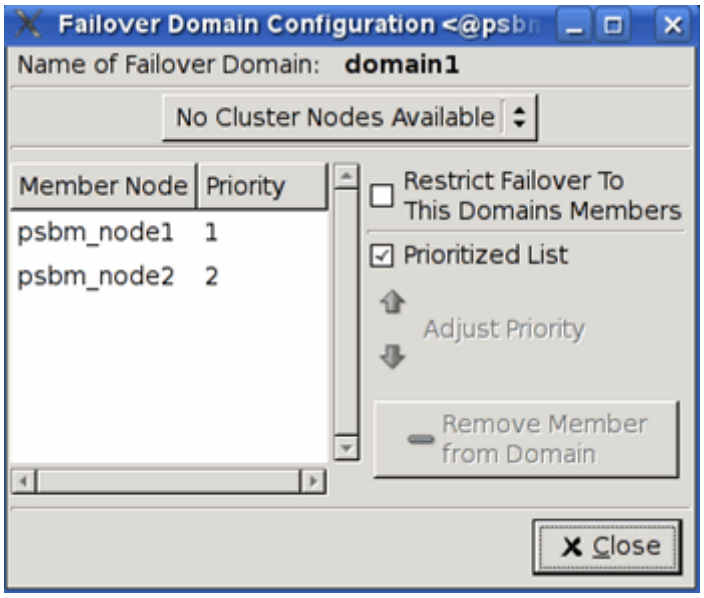


7 Configure failover domains:

- Create one failover domain per cluster service managed by the cluster:

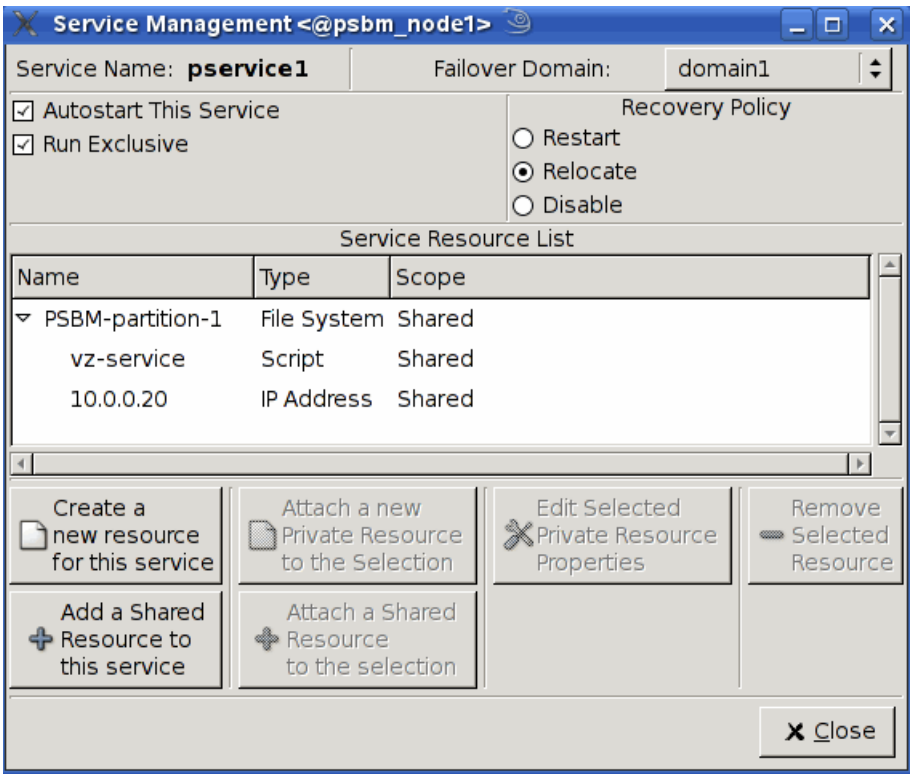


- Configure the list of cluster nodes that will be able to run cluster services from these domains:



8 Create the corresponding clustered services. The number of services must correspond to the number of active servers and shared partitions. Also make sure that:

- The service autostart is enabled.
- The service is configured to run exclusively; thus, the cluster will not attempt to run more than one `vz` service on the same physical server.
- The service recovery policy is `Relocate` or `Restart`. In the latter case, if the `vz` service is stopped for some reason, the cluster will attempt to restart this service on the same server before relocating it to another one.
- Make sure that the proper domain is specified.
- The `Script` resource (`pservice1` in our example) is attached to the `File System` resource (`PSBM-partition-1`). This ensures the proper start-up order of the services:



9 Distribute the configuration file (`/etc/cluster/cluster.conf`) to all the servers, and start the clustering service:

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

10 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
```

11 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.

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 Containers and Parallels templates since all 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 Containers between the cluster nodes. In fact, the migration is almost imperceptible to users since all 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 Container migration.
- Provide failover support for Parallels Server Bare Metal vital services 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 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.

Cluster Configuration Options

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

- The cluster configuration is as follows:
  - A shared storage is used as a cluster storage.
  - The shared storage is mounted simultaneously to `/vz` on all cluster nodes.
  - A clustered file system (GFS or NFS) is 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.
- The following cluster resources are present in the cluster: the `/etc/init.d/vz-cluster` script and (optionally) the `IP address` resource.

Configuring a Shared Storage

After you have set up Parallels Server Bare Metal on each cluster node, you should configure a data storage (iSCSI, Fibre Channel, or NFS) for the cluster. This data storage will be used for hosting the /vz partition with all 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 volume
- a GFS partition

Configuring a Shared Storage on NFS

Configuring a data shared storage located on an NFS volume includes the following steps:

- Configuring the data shared storage for the first node in the cluster.
- 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 fist node in the cluster, do the following:

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

```
# chkconfig portmap on
# chkconfig nfslock on
# chkconfig nfs on
# chkconfig netfs on
```

3. Stop the vz service:

```
# service vz 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 an NFS volume to /vz:

```
# service portmap 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 the 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 the subsequent nodes in the cluster, do the following:

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

```
# chkconfig portmap on
# chkconfig nfslock on
# chkconfig nfs on
# chkconfig netfs on
```

3. Stop the `vz` service:

```
# service vz stop
```

4. (Optional) 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 nodes, reboot all the cluster nodes and make sure all of them have the shared data storage mounted to `/vz`.

Configuring a Shared Storage on GFS

Setting up a shared data storage located on a GFS partition consists of the following steps:

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

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 (e.g. Logical Volume Manager) to set up a logical volume on your data storage (e.g. `/dev/vg01/lv01`). 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 `lvm.conf` man pages.

3. Make sure that the created logical volumes can be accessed by all servers in the cluster. This will ensure that the clustering software will be able to mount the `/vz` partition that you will create on the logical volume in the next step to any of your cluster nodes.
4. Create the `/vz` partition on the logical volume (e.g. using the `fdisk` utility or any other disk partitioning software).

   Example:

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

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

```
# gfs_mkfs -p lock_dlm -t pcluster:gfs1 -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 pcluster:gfs1` denotes the name of the cluster (`pcluster`) for which the GFS file system is created and the name that will be assigned to the GFS file system (`gfs1`).

     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 `gfs_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 `gfs1` name for the `pcluster` cluster will be created. The file system will use the `lock_dlm` protocol, contain 4 journals, and reside on the `/vg01/lv01` volume.

6 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 gfs 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 gfs defaults,noatime,_netdev 0 0
```

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

```
# chkconfig netfs on
```

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

```
# chkconfig --level 3 gfs on
```

8 Stop the `vz` service:

```
# service vz stop
```

9 Move `/vz` to a temporary directory `/vz1` and create a new `/vz` directory:

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

10 Mount a shared data storage located on an GFS volume to `/vz`:

```
# mount /dev/vg01/lv01 /vz
```

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

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

Configuring the Data Storage for Other Nodes in the Cluster

To configure the shared data storage for the second and all the 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 gfs 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 gfs on
```

Creating and Configuring the Data Sharing Cluster

Before configuring the data sharing cluster, make sure that the automatic start of the `vz` service is disabled on each cluster node. If it is not, run this command on the corresponding nodes:

```
# service vz off
```

Once the `vz` service is disabled, you can proceed with configuring the data sharing cluster. The example below demonstrates how to set up a new data sharing cluster using the `system-config-cluster` cluster configuration GUI tool:

1. Log in to any sever that runs the RHEL 5 or CentOS 5 operating system and have the `X-Window System` component installed, and launch the Cluster Configuration Tool:

```
# system-config-cluster
```

2. Your further steps will be different depending on whether you are starting the tool for the first time or have already launched it before. Click Create New Configuration in the displayed dialog window if you are running the Cluster Configuration Tool for the first time. Otherwise, click File > New.

3. In the New Configuration window, enter a descriptive name for the cluster in the field provided, and click OK.

   Note: The dlm lock manager has been tested and recommended for the production usage; other locking schemes (e.g. quorum disk) are not supported.

4. Set one or more fence devices for the cluster:

   a. Select the Fence Devices item, and click Add a Fence Device at the bottom of the Cluster Configuration window:

b In the Fence Device Configuration window, select the type of fence device to configure, and provide the necessary information in the appropriate fields.

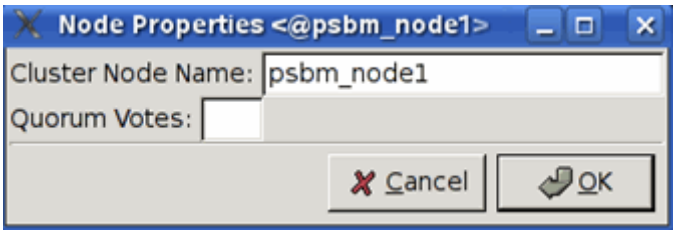
Note: Refer to the *Configuring and Managing a Red Hat Cluster* document for detailed information on available fence devices and their parameters. Manual fencing is supported for testing purposes only and should not be used in production.

c Click OK.

5 Add all your Parallels servers to the cluster:

a Select the Cluster Nodes item, and click Add a Cluster Node at the bottom of the Cluster Configuration window.

b In the Node Properties window, type the name of the server you wish to add to the cluster (or its IP address on the cluster subnet), and click OK. For example:

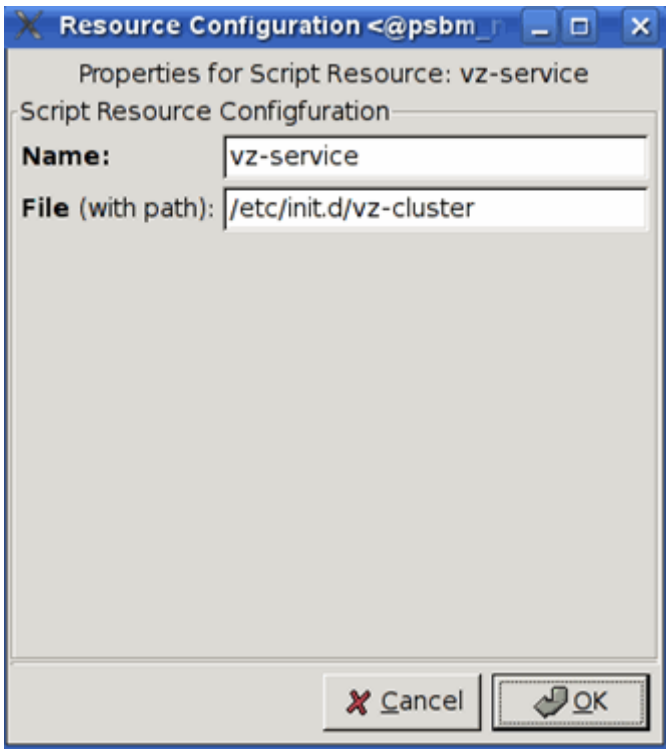


c Configure fencing for the added node using the Manage Fencing For This Node button at the bottom of the Cluster Configuration right frame.

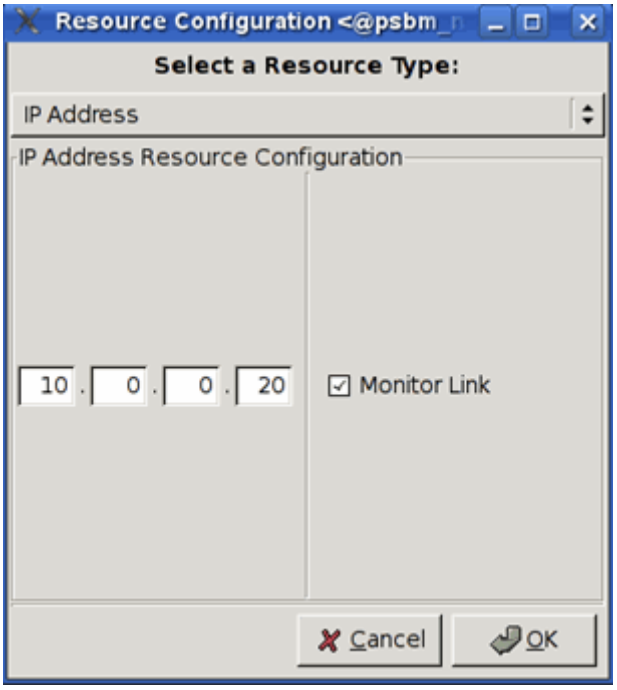
d Repeat the operations above for each server to be included in the cluster.

6 Create resources:

- *File system*: you do not need to configure the file system resource for the data sharing cluster.
- *Script*: one script per cluster:



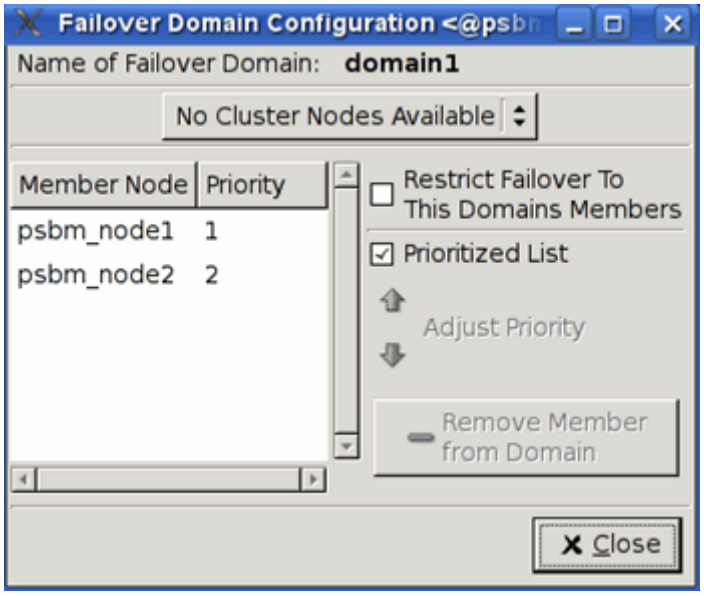
- (Optional) *IP address*: an IP address is needed for each `vz` service (it will be used for a direct SSH connection to the server). Note that the IP address will be managed by the cluster; so, it must not be already in use and assigned to the nodes directly:



7 Configure failover domains:

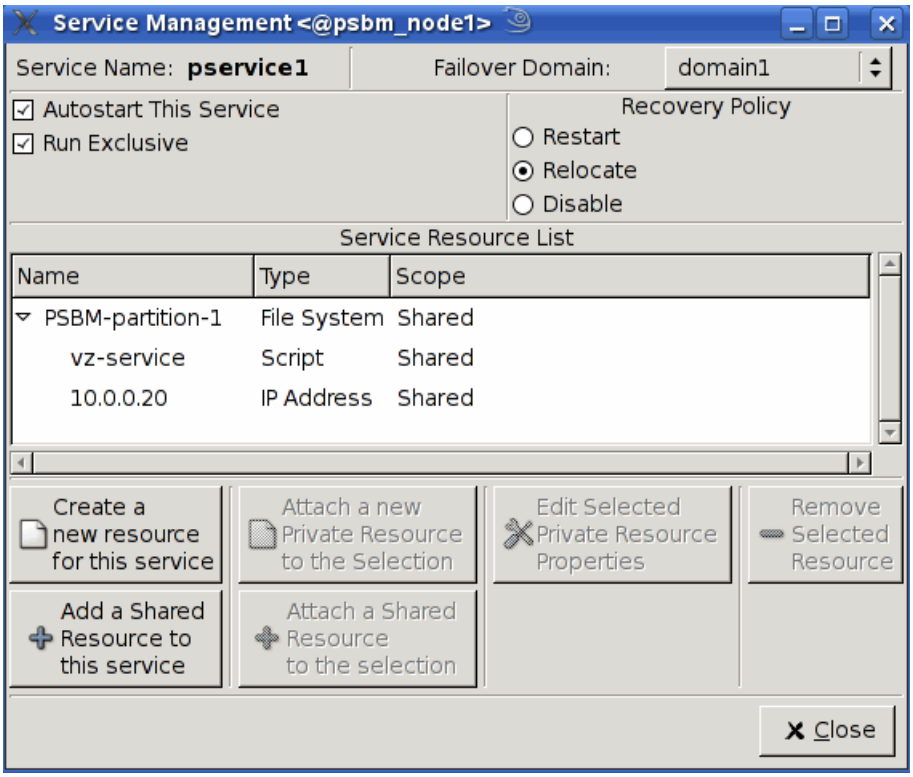
- Create one failover domain per cluster service managed by the cluster:

- Configure the list of cluster nodes that will be able to run cluster services from these domains:



8 Create the corresponding clustered services. The number of services corresponds to the number of active servers and shared partitions. Also make sure that:

- The service autostart is enabled.
- The service is configured to run exclusively; thus, the cluster will not attempt to run more than one `vz` service on the same physical server.
- The service recovery policy is `Relocate` or `Restart`. In the latter case, if the `vz` service is stopped for some reason, the cluster will attempt to restart this service on the same server before relocating it to another one.
- Make sure that the proper domain is specified.
- The `Script` resource (`pservice1` in our example) is attached to the `File System` resource (`PSBM-partition-1`). This ensures the proper start-up order of the services:



9 Distribute the configuration file (`/etc/cluster/cluster.conf`) to all the servers, and start the clustering service:

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

10 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
```

11 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.

Testing a Cluster

To test the created cluster, you can shut down or unplug a power cord on one of the servers in the cluster which currently runs the Parallels Server Bare Metal software and in less than a minute, the `vz` service should be automatically relocated from the failed server to a standby one (if you are using manual fencing, you may need to use the `fence_ack_manual` tool to inform the cluster of the fenced server). 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 servers in the cluster.

Index