



Parallels Virtuozzo Containers 4.7 for Linux

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 Clustering Software.....	7
Installing Parallels Virtuozzo Containers 4.7	8
Deploying a Failover Cluster.....	9
Cluster Configuration Options.....	9
Configuring Shared Storage for Failover Clusters.....	10
Configuring iSCSI Data Storages.....	11
Creating and Configuring a Failover Cluster.....	13
Deploying a Data Sharing Cluster.....	19
Cluster Configuration Options.....	20
Configuring Shared Storages for Data Sharing Clusters.....	20
Configuring Shared Storages on NFS.....	21
Creating and Configuring a Data Sharing Cluster.....	23
Managing Clusters	29
Index	30

CHAPTER 1

Introduction

In Parallels Virtuozzo Containers 4.7, 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 Virtuozzo Containers installations. In a failover cluster, each *vz* and *parallels-server* 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.
- *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. In Parallels Virtuozzo Containers 4.7, a shared file system is Network Files System (NFS).

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 Clustering Software	7
Installing Parallels Virtuozzo Containers 4.7	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 service act as clustered services. 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 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 Virtuozzo Containers 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 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 Virtuozzo Containers 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 Virtuozzo Containers.
 - *Storage* usually represented by a common disk array on a SAN and used to store all Parallels Virtuozzo Containers-related data (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.

Note: You will also need a server running RHEL or CentOS 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, see the clustering documentation on the Red Hat web site (available at http://docs.redhat.com/docs/en-US/Red_Hat_Enterprise_Linux/).

Installing Clustering Software

Once you set up hardware for your cluster, you need to install the clustering software from Red Hat Cluster Suite. The clustering software must be installed on the server before you start installing Parallels Virtuozzo Containers 4.7.

Consult the clustering documentation on the Red Hat web site (available at http://docs.redhat.com/docs/en-US/Red_Hat_Enterprise_Linux/) to learn what software components you need to install on your server.

Installing Parallels Virtuozzo Containers 4.7

After you set up cluster hardware and install clustering software, you can proceed with installing Parallels Virtuozzo Containers 4.7. The configuration and partitioning requirements for cluster nodes slightly differ from those for standalone servers. Use the guidelines below when installing Parallels Virtuozzo Containers 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 Virtuozzo Containers, we highly recommend that you update your software to its latest version (for example, 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 Parallels servers.

In This Chapter

Cluster Configuration Options	9
Configuring Shared Storage for Failover Clusters.....	10
Creating and Configuring a Failover Cluster.....	13

Cluster Configuration Options

Parallels Virtuozzo Containers 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 Virtuozzo Containers 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

After installing the clustering software, you need to configure the shared storage which is to be used as the `/vz` partition. The shared storage can be configured from a single server or from several servers (it does not really matter). The space requirements for this partition should be calculated on the basis of the space demands for your Containers and Parallels Virtuozzo Containers templates. The number of partitions should correspond to the number of clustered services and be less than the number of physical servers in your cluster.

Please keep in mind that all your servers should 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 the mount points which are always `/vz` on all the servers in the cluster.

After the corresponding LUNs have been successfully created, make the 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:

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

- 1 Disable the automatic start of the *vz* and *parallels-server* service (it will be managed by the cluster script):

```
# service vz 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 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 an RHEL or CentOS operating system (for example, RHEL 5 or CentOS 5) 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:



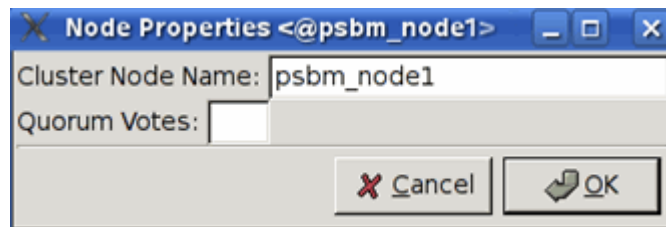
- b** In the **Fence Device Configuration** window, select the type of fence device to configure, and provide the necessary information in the appropriate fields.

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 Parallel servers to the cluster:

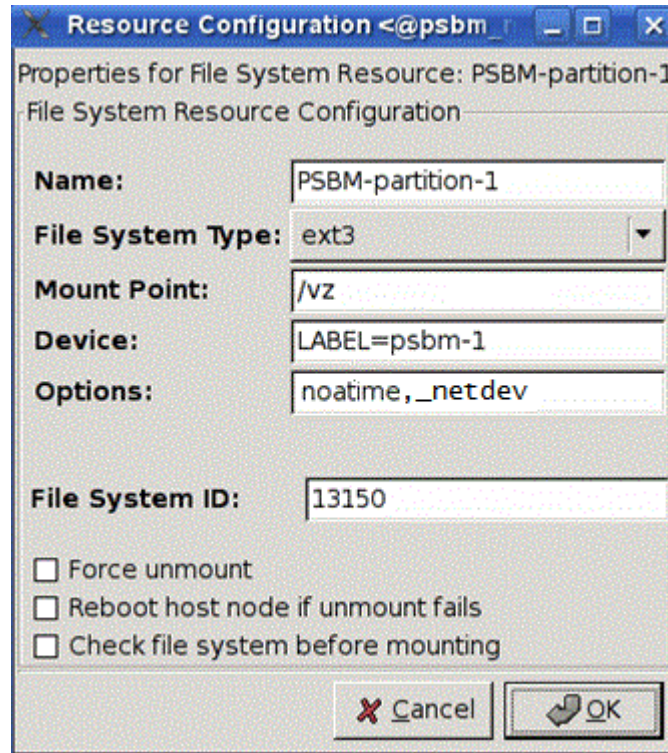
- a** Select the **Cluster Nodes** item, and click **Add a Cluster Node** at the bottom of the **Cluster Configuration** window.
- b** In the **Node Properties** window, type the name of the server you want to add to the cluster (or its IP address on the cluster subnet), and click **OK**. For example:



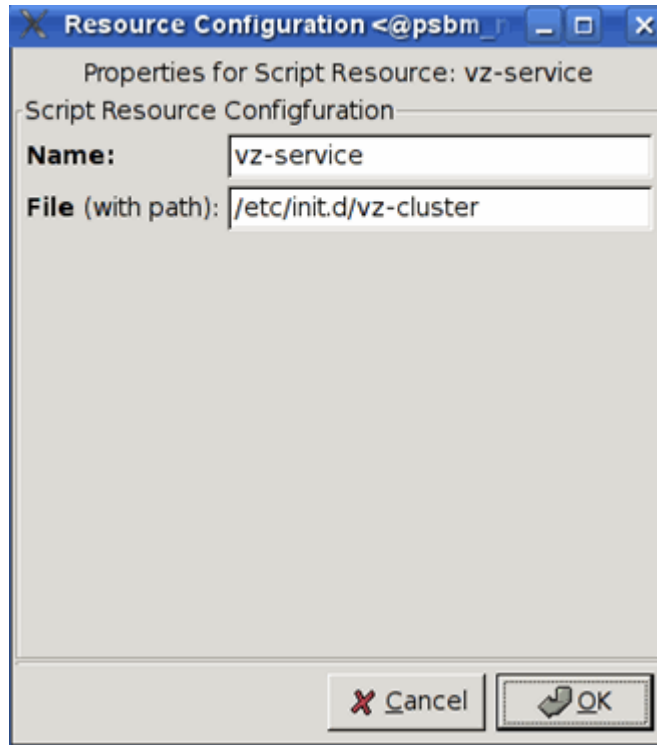
- c** Configure fencing for the added node using the **Manage Fencing For This Node** button at the bottom of the **Cluster Configuration** right frame.
- d** Repeat the aforementioned operations for each server to be included in the cluster.

6 Create resources:

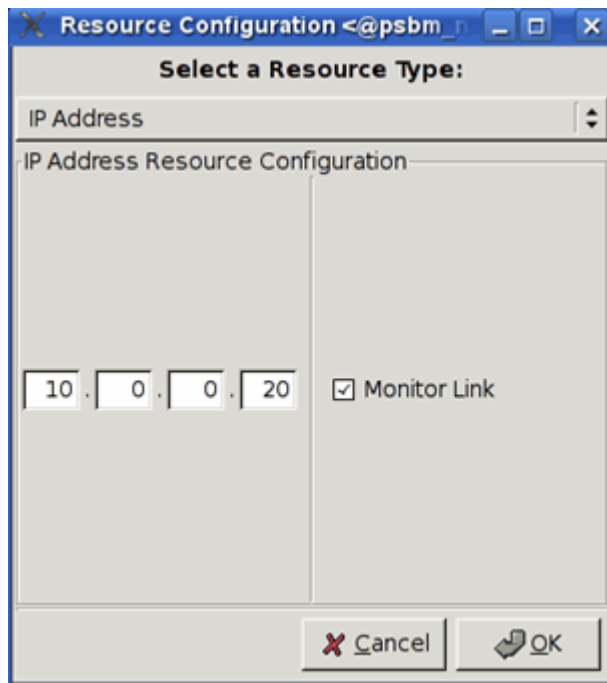
- *File system*: one resource per each shared SCSI partition. Make sure that the partition is referred to by the volume label rather than by the device name. Also, do not forget to use the `noatime` and `_netdev` options. See the picture below:



- *Script*: one script per cluster:

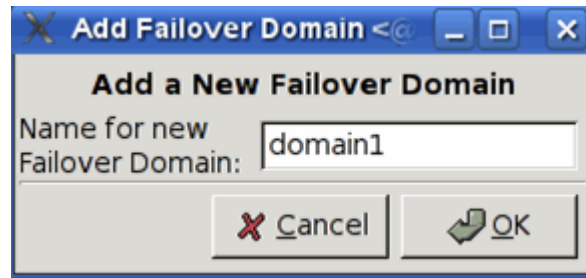


- (Optional) *IP address*: an IP address is needed for each vz service (it will be used for a direct SSH connection to the host). Note that the IP address will be managed by the cluster, so it must not be already in use or assigned to the nodes directly:

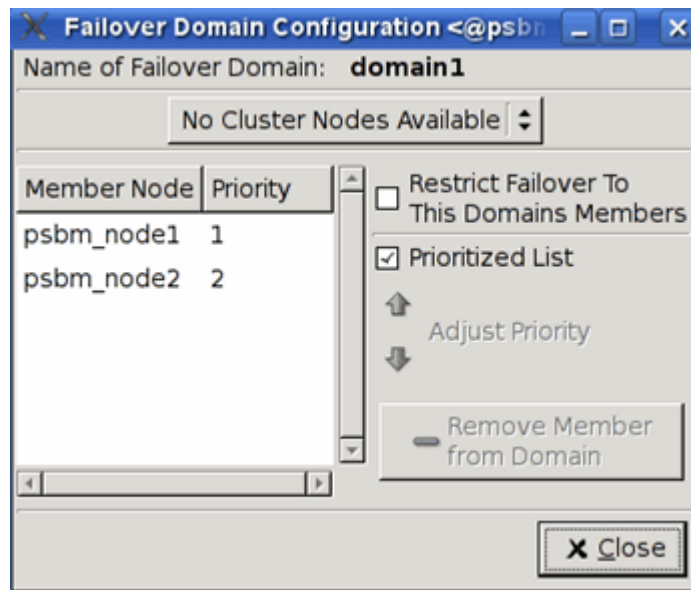


7 Configure failover domains:

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

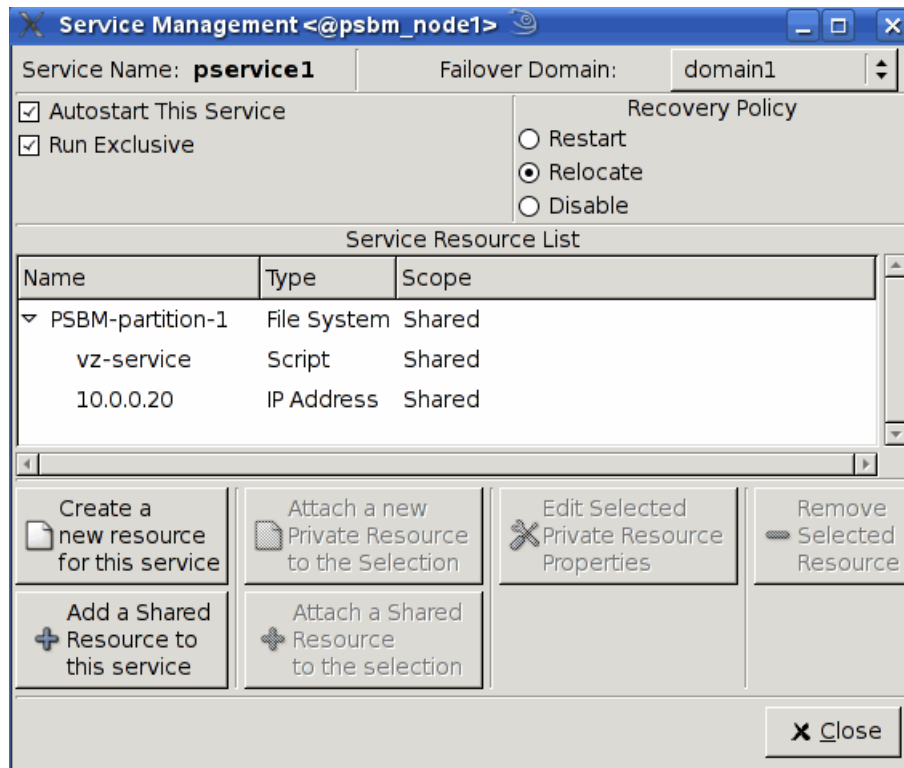


- Configure the list of cluster nodes that will be able to run cluster services from these domains:

**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 (`psservice1` in our example) is attached to the filesystem resource (`PSBM-partition-1`). This ensures the proper start 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 Virtuozzo Containers 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 templates because 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 migration.
- Provide failover support for Parallels Virtuozzo Containers vital services and Containers. Each server in the cluster is running the clustering software responsible for monitoring the health of Parallels Virtuozzo Containers 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.

In This Chapter

Cluster Configuration Options	20
Configuring Shared Storages for Data Sharing Clusters	20

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 (NFS) is used.
- One and the same version of the Parallels Virtuozzo Containers 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 Virtuozzo Containers on each cluster node, you need to configure a data storage (iSCSI, Fibre Channel, or NFS) for the cluster. This data storage will then 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 NFS partitions only.

Configuring Shared Storages on NFS

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 all NFS-related services are enabled on the node:

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

- 3 Stop the *vz* and *parallels-server* 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 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 _netdev,defaults,noatime 0 0
```

Be sure to add the *_netdev* and *noatime* options to the entry.

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 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* and *parallels-server* service:

```
# service vz 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 _netdev,defaults,noatime 0 0
```

Be sure to add the *_netdev* and *noatime* options to the entry.

Once you configure all nodes, reboot them and make sure each node has the shared data storage mounted to */vz*.

Creating and Configuring a Data Sharing Cluster

Before setting up a data sharing cluster, make sure that the automatic startup of the `vz` service is disabled on each of the cluster nodes. If it is not, run this command on the respective nodes:

```
# service vz off
```

Once the automatic startup 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 Cluster Configuration Tool:

- 1 Log in to any sever that runs an RHEL or CentOS operating system (e.g., RHEL 5 or CentOS 5) 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** if you are running the Cluster Configuration Tool for the first time.
 - Click **File > New** if you have already used the Cluster Configuration Tool before.
- 3 In the **New Configuration** window, enter a descriptive name for the cluster in the field provided, and click **OK**.

Note: The `d1m 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:



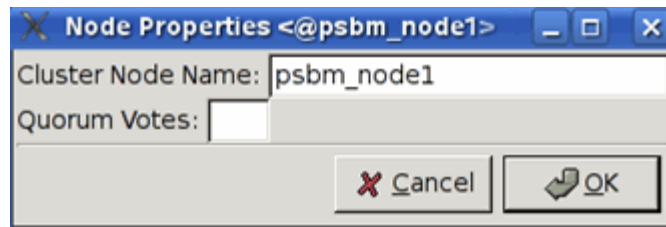
- b** In the **Fence Device Configuration** window, select the type of fence device to configure, and provide the necessary information in the appropriate fields.

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:

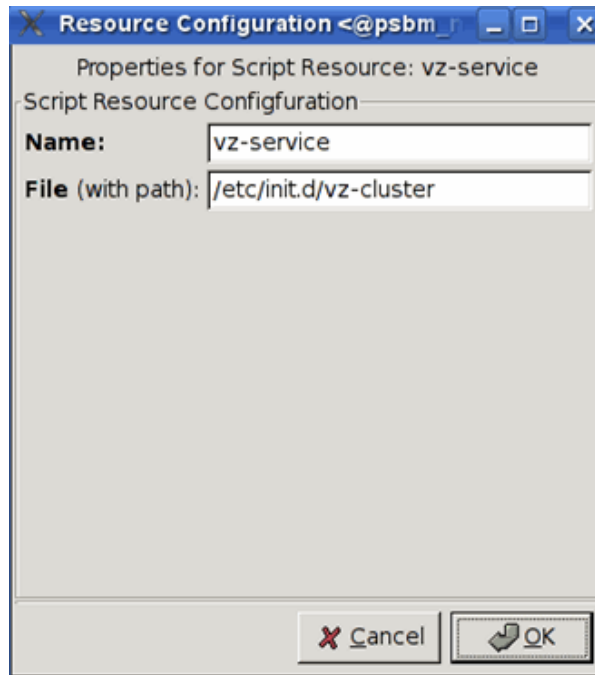
- a** Select the **Cluster Nodes** item, and click **Add a Cluster Node** at the bottom of the **Cluster Configuration** window.
- b** In the **Node Properties** window, type the name of the server you wish to add to the cluster (or its IP address on the cluster subnet), and click **OK**. For example:



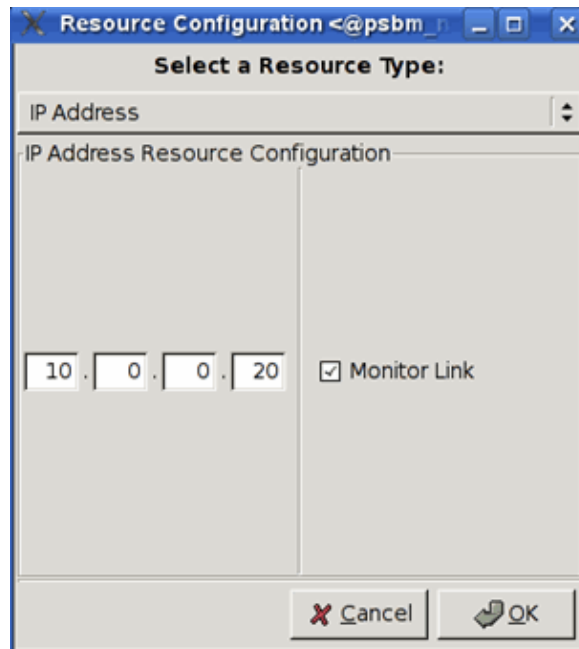
- c** Configure fencing for the added node using the **Manage Fencing For This Node** button at the bottom of the **Cluster Configuration** right frame.
- d** Repeat the operations above for each server to be included in the cluster.

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:

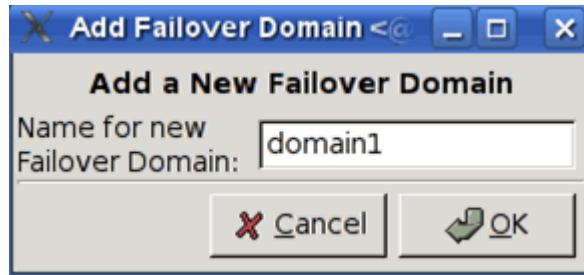


- (Optional) *IP address*: an IP address is needed for each vz service (it will be used for a direct SSH connection to the server). Note that the IP address will be managed by the cluster; so, it must not be already in use and assigned to the nodes directly:

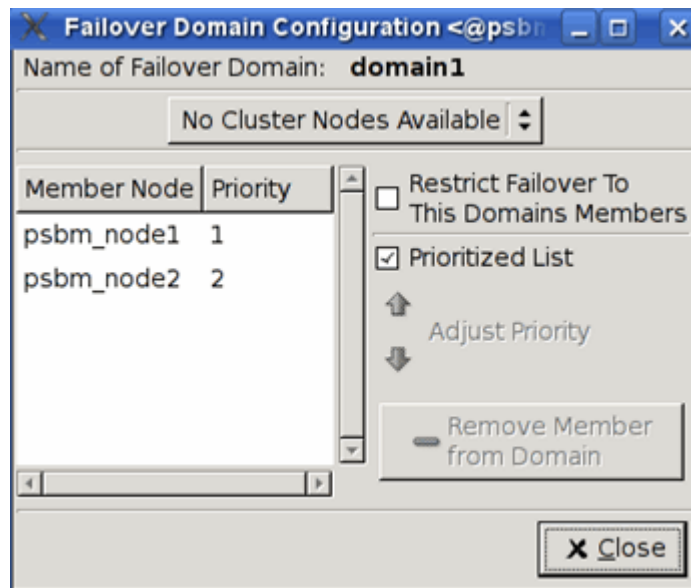


7 Configure failover domains:

- Create one failover domain per cluster service that will be managed by the cluster. For example, when creating a two-node active/passive cluster, you can add only one failover domain because the cluster will host only one cluster service.



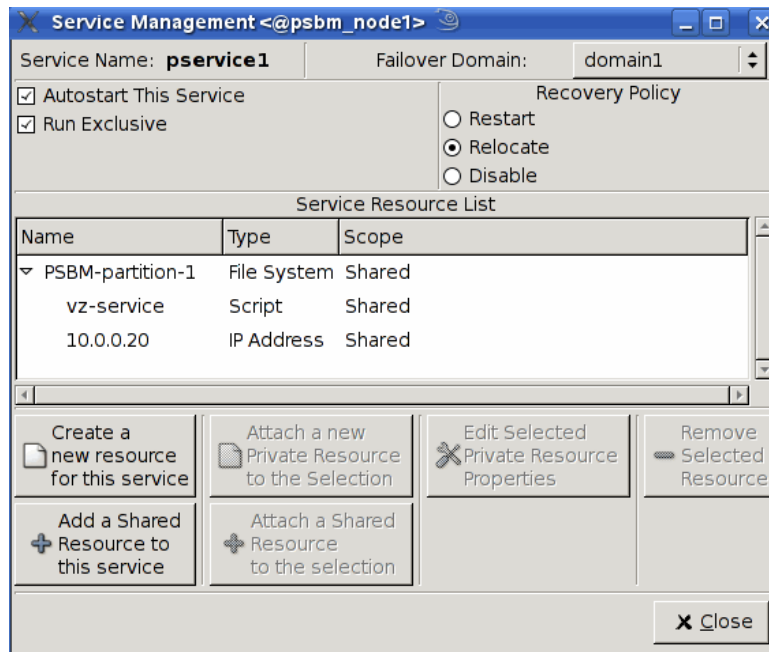
- Configure the list of cluster nodes that will be able to run the cluster service associated with the respective failover domain.



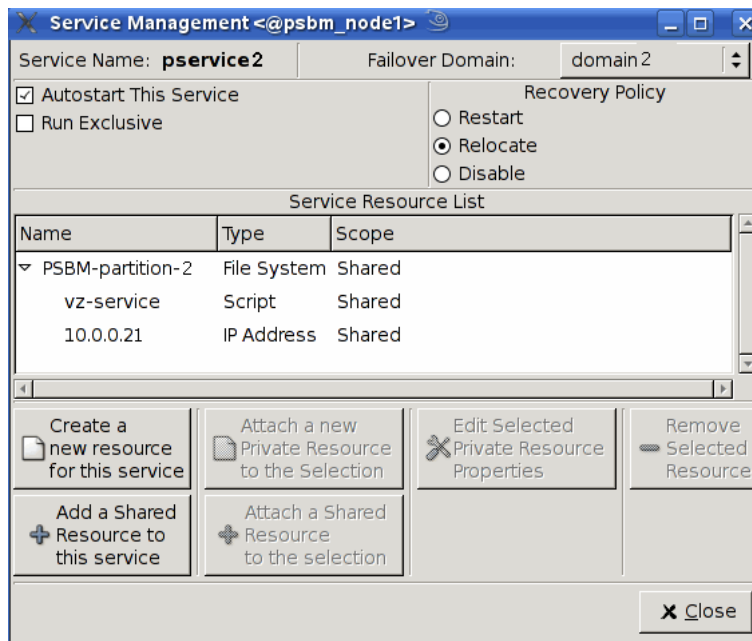
8 Create the corresponding clustered services. The number of services must correspond to the number of active servers and shared partitions. Also make sure that:

- The service autostart is enabled.
- The service is configured to run exclusively so that 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 (`p-service1` in our example) is attached to the File System resource (`PSBM-partition-1`). This ensures the proper start-up order of the services:



When configuring more than one clustered service, make sure that each service is associated with the respective failover domain. For example, the picture above shows the process of configuring the `p-service1` service that is associated with the `domain1` failover domain (see the **Service Name** and **Failover Domain** fields in the picture). So when configuring another service (e.g., `p-service2`) that is associated with another failover domain (e.g., `domain2`), your **Service Management** window should look like the following:



- 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.

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 `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, 20
- Configuring iSCSI Data Storages - 11
- Configuring Shared Storage for Failover Clusters - 10
- Configuring Shared Storages for Data Sharing Clusters - 20
- Configuring Shared Storages on NFS - 21
- Creating and Configuring a Data Sharing Cluster - 23
- Creating and Configuring a Failover Cluster - 13

D

- Deploying a Data Sharing Cluster - 19
- Deploying a Failover Cluster - 9

I

- Installing Clustering Software - 7
- Installing Parallels Virtuozzo Containers 4.7 - 8
- Introduction - 4

M

- Managing Clusters - 29

P

- Preparing to Create a Cluster - 5

S

- Setting Up Cluster Hardware - 7

U

- Understanding the Cluster Configuration - 6