



Red Hat Enterprise Linux 6 Fence Configuration Guide

Configuring and Managing Fence Devices for the High Availability Add-On

Red Hat Enterprise Linux 6 Fence Configuration Guide

Configuring and Managing Fence Devices for the High Availability Add-On

Legal Notice

Copyright © 2012 Red Hat, Inc. and others.

This document is licensed by Red Hat under the [Creative Commons Attribution-ShareAlike 3.0 Unported License](#). If you distribute this document, or a modified version of it, you must provide attribution to Red Hat, Inc. and provide a link to the original. If the document is modified, all Red Hat trademarks must be removed.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, JBoss, MetaMatrix, Fedora, the Infinity Logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux® is the registered trademark of Linus Torvalds in the United States and other countries.

Java® is a registered trademark of Oracle and/or its affiliates.

XFS® is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL® is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js® is an official trademark of Joyent. Red Hat Software Collections is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack® Word Mark and OpenStack Logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

Abstract

Fencing is the disconnection of a node from the cluster's shared storage. Fencing cuts off I/O from shared storage, thus ensuring data integrity. This manual documents the configuration of fencing on clustered systems using High Availability Add-On and details the configuration of supported fence devices.

Table of Contents

Preface	4
1. Document Conventions	4
1.1. Typographic Conventions	4
1.2. Pull-quote Conventions	5
1.3. Notes and Warnings	6
2. Getting Help and Giving Feedback	6
2.1. Do You Need Help?	6
2.2. We Need Feedback!	7
Chapter 1. Fencing Pre-Configuration	8
1.1. Configuring ACPI For Use with Integrated Fence Devices	8
1.1.1. Disabling ACPI Soft-Off with chkconfig Management	9
1.1.2. Disabling ACPI Soft-Off with the BIOS	9
1.1.3. Disabling ACPI Completely in the grub.conf File	10
Chapter 2. Configuring Fencing with the ccs Command	12
2.1. Configuring Fence Devices	12
2.2. Listing Fence Devices and Fence Device Options	14
2.3. Configuring Fencing for Cluster Members	16
2.3.1. Configuring a Single Power-Based Fence Device for a Node	16
2.3.2. Configuring a Single Storage-Based Fence Device for a Node	18
2.3.3. Configuring a Backup Fence Device	20
2.3.4. Configuring a Node with Redundant Power	24
2.3.5. Removing Fence Methods and Fence Instances	26
Chapter 3. Configuring Fencing with Conga	28
3.1. Configuring Fence Daemon Properties	28
3.2. Configuring Fence Devices	28
3.2.1. Creating a Fence Device	29
3.2.2. Modifying a Fence Device	30
3.2.3. Deleting a Fence Device	30
3.3. Configuring Fencing for Cluster Members	30
3.3.1. Configuring a Single Fence Device for a Node	30
3.3.2. Configuring a Backup Fence Device	31
3.3.3. Configuring a Node with Redundant Power	32
Chapter 4. Fence Devices	34
4.1. APC Power Switch over Telnet and SSH	36
4.2. APC Power Switch over Telnet and SSH	38
4.3. Brocade Fabric Switch	40
4.4. Cisco MDS	42
4.5. Cisco UCS	44
4.6. Dell Drac 5	45
4.7. Eaton Network Power Switch	47
4.8. Egenera BladeFrame	50
4.9. ePowerSwitch	51
4.10. Fence Virt	52
4.11. Fujitsu-Siemens RemoteView Service Board (RSB)	54
4.12. Hewlett-Packard BladeSystem	55
4.13. Hewlett-Packard iLO	56
4.14. Hewlett-Packard iLO MP	58
4.15. IBM BladeCenter	59
4.16. IBM BladeCenter over SNMP	61

4.17. IBM iPDU	64
4.18. IF-MIB	66
4.19. Intel Modular	68
4.20. IPMI over LAN	70
4.21. RHEV-M REST API	71
4.22. SCSI Persistent Reservations	73
4.23. VMWare over SOAP API	74
4.24. WTI Power Switch	76
Revision History	78
Index	78
A	78
B	78
C	78
D	78
E	78
F	78
H	81
I	81
R	81
S	81
T	81
V	81
W	81

Preface

1. Document Conventions

This manual uses several conventions to highlight certain words and phrases and draw attention to specific pieces of information.

In PDF and paper editions, this manual uses typefaces drawn from the [Liberation Fonts](#) set. The Liberation Fonts set is also used in HTML editions if the set is installed on your system. If not, alternative but equivalent typefaces are displayed. Note: Red Hat Enterprise Linux 5 and later include the Liberation Fonts set by default.

1.1. Typographic Conventions

Four typographic conventions are used to call attention to specific words and phrases. These conventions, and the circumstances they apply to, are as follows.

Mono-spaced Bold

Used to highlight system input, including shell commands, file names and paths. Also used to highlight keys and key combinations. For example:

To see the contents of the file **my_next_bestselling_novel** in your current working directory, enter the **cat my_next_bestselling_novel** command at the shell prompt and press **Enter** to execute the command.

The above includes a file name, a shell command and a key, all presented in mono-spaced bold and all distinguishable thanks to context.

Key combinations can be distinguished from an individual key by the plus sign that connects each part of a key combination. For example:

Press **Enter** to execute the command.

Press **Ctrl+Alt+F2** to switch to a virtual terminal.

The first example highlights a particular key to press. The second example highlights a key combination: a set of three keys pressed simultaneously.

If source code is discussed, class names, methods, functions, variable names and returned values mentioned within a paragraph will be presented as above, in **mono-spaced bold**. For example:

File-related classes include **filesystem** for file systems, **file** for files, and **dir** for directories. Each class has its own associated set of permissions.

Proportional Bold

This denotes words or phrases encountered on a system, including application names; dialog-box text; labeled buttons; check-box and radio-button labels; menu titles and submenu titles. For example:

Choose **System** → **Preferences** → **Mouse** from the main menu bar to launch **Mouse Preferences**. In the **Buttons** tab, select the **Left-handed mouse** check box and click **Close** to switch the primary mouse button from the left to the right (making the mouse suitable for use in the left hand).

To insert a special character into a **gedit** file, choose **Applications** → **Accessories** →

Character Map from the main menu bar. Next, choose **Search** → **Find...** from the **Character Map** menu bar, type the name of the character in the **Search** field and click **Next**. The character you sought will be highlighted in the **Character Table**. Double-click this highlighted character to place it in the **Text to copy** field and then click the **Copy** button. Now switch back to your document and choose **Edit** → **Paste** from the **gedit** menu bar.

The above text includes application names; system-wide menu names and items; application-specific menu names; and buttons and text found within a GUI interface, all presented in proportional bold and all distinguishable by context.

Mono-spaced Bold Italic* or *Proportional Bold Italic

Whether mono-spaced bold or proportional bold, the addition of italics indicates replaceable or variable text. Italics denotes text you do not input literally or displayed text that changes depending on circumstance. For example:

To connect to a remote machine using ssh, type **ssh *username@domain.name*** at a shell prompt. If the remote machine is **example.com** and your username on that machine is john, type **ssh *john@example.com***.

The **mount -o remount *file-system*** command remounts the named file system. For example, to remount the **/home** file system, the command is **mount -o remount */home***.

To see the version of a currently installed package, use the **rpm -q *package*** command. It will return a result as follows: ***package-version-release***.

Note the words in bold italics above: *username*, *domain.name*, *file-system*, *package*, *version* and *release*. Each word is a placeholder, either for text you enter when issuing a command or for text displayed by the system.

Aside from standard usage for presenting the title of a work, italics denotes the first use of a new and important term. For example:

Publican is a *DocBook* publishing system.

1.2. Pull-quote Conventions

Terminal output and source code listings are set off visually from the surrounding text.

Output sent to a terminal is set in **mono-spaced roman** and presented thus:

```
books      Desktop  documentation  drafts  mss    photos  stuff  svn
books_tests Desktop1  downloads      images  notes  scripts svgs
```

Source-code listings are also set in **mono-spaced roman** but add syntax highlighting as follows:

```
static int kvm_vm_ioctl_deassign_device(struct kvm *kvm,
                                       struct kvm_assigned_pci_dev *assigned_dev)
{
    int r = 0;
    struct kvm_assigned_dev_kernel *match;

    mutex_lock(&kvm->lock);

    match = kvm_find_assigned_dev(&kvm->arch.assigned_dev_head,
                                assigned_dev->assigned_dev_id);
    if (!match) {
        printk(KERN_INFO "%s: device hasn't been assigned before, "
                    "so cannot be deassigned\n", __func__);
        r = -EINVAL;
        goto out;
    }

    kvm_deassign_device(kvm, match);

    kvm_free_assigned_device(kvm, match);

out:
    mutex_unlock(&kvm->lock);
    return r;
}
```

1.3. Notes and Warnings

Finally, we use three visual styles to draw attention to information that might otherwise be overlooked.



Note

Notes are tips, shortcuts or alternative approaches to the task at hand. Ignoring a note should have no negative consequences, but you might miss out on a trick that makes your life easier.



Important

Important boxes detail things that are easily missed: configuration changes that only apply to the current session, or services that need restarting before an update will apply. Ignoring a box labeled “Important” will not cause data loss but may cause irritation and frustration.



Warning

Warnings should not be ignored. Ignoring warnings will most likely cause data loss.

2. Getting Help and Giving Feedback

2.1. Do You Need Help?

If you experience difficulty with a procedure described in this documentation, visit the Red Hat Customer

Portal at <http://access.redhat.com>. Through the customer portal, you can:

- search or browse through a knowledgebase of technical support articles about Red Hat products.
- submit a support case to Red Hat Global Support Services (GSS).
- access other product documentation.

Red Hat also hosts a large number of electronic mailing lists for discussion of Red Hat software and technology. You can find a list of publicly available mailing lists at <https://www.redhat.com/mailman/listinfo>. Click on the name of any mailing list to subscribe to that list or to access the list archives.

2.2. We Need Feedback!

If you find a typographical error in this manual, or if you have thought of a way to make this manual better, we would love to hear from you! Please submit a report in Bugzilla: <http://bugzilla.redhat.com/> against the product **Documentation**.

When submitting a bug report, be sure to mention the manual's identifier: *Fence_Configuration_Guide*

If you have a suggestion for improving the documentation, try to be as specific as possible when describing it. If you have found an error, please include the section number and some of the surrounding text so we can find it easily.

Chapter 1. Fencing Pre-Configuration

This chapter describes tasks to perform and considerations to make before deploying fencing on clusters using Red Hat High Availability Add-On, and consists of the following sections.

- [Section 1.1, “Configuring ACPI For Use with Integrated Fence Devices”](#)

1.1. Configuring ACPI For Use with Integrated Fence Devices

If your cluster uses integrated fence devices, you must configure ACPI (Advanced Configuration and Power Interface) to ensure immediate and complete fencing.



Note

For the most current information about integrated fence devices supported by Red Hat High Availability Add-On, refer to http://www.redhat.com/cluster_suite/hardware/.

If a cluster node is configured to be fenced by an integrated fence device, disable ACPI Soft-Off for that node. Disabling ACPI Soft-Off allows an integrated fence device to turn off a node immediately and completely rather than attempting a clean shutdown (for example, **shutdown -h now**). Otherwise, if ACPI Soft-Off is enabled, an integrated fence device can take four or more seconds to turn off a node (refer to note that follows). In addition, if ACPI Soft-Off is enabled and a node panics or freezes during shutdown, an integrated fence device may not be able to turn off the node. Under those circumstances, fencing is delayed or unsuccessful. Consequently, when a node is fenced with an integrated fence device and ACPI Soft-Off is enabled, a cluster recovers slowly or requires administrative intervention to recover.



Note

The amount of time required to fence a node depends on the integrated fence device used. Some integrated fence devices perform the equivalent of pressing and holding the power button; therefore, the fence device turns off the node in four to five seconds. Other integrated fence devices perform the equivalent of pressing the power button momentarily, relying on the operating system to turn off the node; therefore, the fence device turns off the node in a time span much longer than four to five seconds.

To disable ACPI Soft-Off, use **chkconfig** management and verify that the node turns off immediately when fenced. The preferred way to disable ACPI Soft-Off is with **chkconfig** management; however, if that method is not satisfactory for your cluster, you can disable ACPI Soft-Off with one of the following alternate methods:

- Changing the BIOS setting to "instant-off" or an equivalent setting that turns off the node without delay



Note

Disabling ACPI Soft-Off with the BIOS may not be possible with some computers.

- Appending **acpi=off** to the kernel boot command line of the **/boot/grub/grub.conf** file

**Important**

This method completely disables ACPI; some computers do not boot correctly if ACPI is completely disabled. Use this method *only* if the other methods are not effective for your cluster.

The following sections provide procedures for the preferred method and alternate methods of disabling ACPI Soft-Off:

- ▶ [Section 1.1.1, “Disabling ACPI Soft-Off with **chkconfig** Management”](#) — Preferred method
- ▶ [Section 1.1.2, “Disabling ACPI Soft-Off with the BIOS”](#) — First alternate method
- ▶ [Section 1.1.3, “Disabling ACPI Completely in the **grub.conf** File”](#) — Second alternate method

1.1.1. Disabling ACPI Soft-Off with **chkconfig** Management

You can use **chkconfig** management to disable ACPI Soft-Off either by removing the ACPI daemon (**acpid**) from **chkconfig** management or by turning off **acpid**.

**Note**

This is the preferred method of disabling ACPI Soft-Off.

Disable ACPI Soft-Off with **chkconfig** management at each cluster node as follows:

1. Run either of the following commands:
 - ▶ **chkconfig --del acpid** — This command removes **acpid** from **chkconfig** management.
 - OR —
 - ▶ **chkconfig --level 2345 acpid off** — This command turns off **acpid**.
2. Reboot the node.
3. When the cluster is configured and running, verify that the node turns off immediately when fenced.

**Note**

You can fence the node with the **fence_node** command or **Conga**.

1.1.2. Disabling ACPI Soft-Off with the BIOS

The preferred method of disabling ACPI Soft-Off is with **chkconfig** management ([Section 1.1.1, “Disabling ACPI Soft-Off with **chkconfig** Management”](#)). However, if the preferred method is not effective for your cluster, follow the procedure in this section.

**Note**

Disabling ACPI Soft-Off with the BIOS may not be possible with some computers.

You can disable ACPI Soft-Off by configuring the BIOS of each cluster node as follows:

1. Reboot the node and start the **BIOS CMOS Setup Utility** program.
2. Navigate to the **Power** menu (or equivalent power management menu).
3. At the **Power** menu, set the **Soft-Off by PWR-BTTN** function (or equivalent) to **Instant-Off** (or the equivalent setting that turns off the node via the power button without delay). [Example 1.1, “BIOS CMOS Setup Utility: Soft-Off by PWR-BTTN set to Instant-Off”](#) shows a **Power** menu with **ACPI Function** set to **Enabled** and **Soft-Off by PWR-BTTN** set to **Instant-Off**.

Note

The equivalents to **ACPI Function**, **Soft-Off by PWR-BTTN**, and **Instant-Off** may vary among computers. However, the objective of this procedure is to configure the BIOS so that the computer is turned off via the power button without delay.

4. Exit the **BIOS CMOS Setup Utility** program, saving the BIOS configuration.
5. When the cluster is configured and running, verify that the node turns off immediately when fenced.

Note

You can fence the node with the **fence_node** command or **Conga**.

Example 1.1. BIOS CMOS Setup Utility: Soft-Off by PWR-BTTN set to Instant-Off

		Item Help
	ACPI Function	[Enabled]
	ACPI Suspend Type	[S1(POS)]
x	Run VGABIOS if S3 Resume	[Auto]
	Suspend Mode	[Disabled]
	HDD Power Down	[Disabled]
	Soft-Off by PWR-BTTN	[Instant-Off]
	CPU THRM-Throttling	[50.0%]
	Wake-Up by PCI card	[Enabled]
	Power On by Ring	[Enabled]
	Wake Up On LAN	[Enabled]
x	USB KB Wake-Up From S3	[Disabled]
	Resume by Alarm	[Disabled]
x	Date(of Month) Alarm	0
x	Time(hh:mm:ss) Alarm	0 : 0 :
	POWER ON Function	[BUTTON ONLY]
x	KB Power ON Password	Enter
x	Hot Key Power ON	Ctrl-F1

This example shows **ACPI Function** set to **Enabled**, and **Soft-Off by PWR-BTTN** set to **Instant-Off**.

1.1.3. Disabling ACPI Completely in the grub . conf File

The preferred method of disabling ACPI Soft-Off is with **chkconfig** management ([Section 1.1.1](#),

[“Disabling ACPI Soft-Off with `chkconfig` Management”](#)). If the preferred method is not effective for your cluster, you can disable ACPI Soft-Off with the BIOS power management ([Section 1.1.2, “Disabling ACPI Soft-Off with the BIOS”](#)). If neither of those methods is effective for your cluster, you can disable ACPI completely by appending **`acpi=off`** to the kernel boot command line in the **`grub.conf`** file.



Important

This method completely disables ACPI; some computers do not boot correctly if ACPI is completely disabled. Use this method *only* if the other methods are not effective for your cluster.

You can disable ACPI completely by editing the **`grub.conf`** file of each cluster node as follows:

1. Open **`/boot/grub/grub.conf`** with a text editor.
2. Append **`acpi=off`** to the kernel boot command line in **`/boot/grub/grub.conf`** (refer to [Example 1.2, “Kernel Boot Command Line with `acpi=off` Appended to It”](#)).
3. Reboot the node.
4. When the cluster is configured and running, verify that the node turns off immediately when fenced.



Note

You can fence the node with the **`fence_node`** command or **Conga**.

Example 1.2. Kernel Boot Command Line with `acpi=off` Appended to It

```
# grub.conf generated by anaconda
#
# Note that you do not have to rerun grub after making changes to this file
# NOTICE: You have a /boot partition. This means that
#          all kernel and initrd paths are relative to /boot/, eg.
#          root (hd0,0)
#          kernel /vmlinuz-version ro root=/dev/mapper/vg_doc01-lv_root
#          initrd /initrd-[generic-]version.img
#boot=/dev/hda
default=0
timeout=5
serial --unit=0 --speed=115200
terminal --timeout=5 serial console
title Red Hat Enterprise Linux Server (2.6.32-193.el6.x86_64)
    root (hd0,0)
    kernel /vmlinuz-2.6.32-193.el6.x86_64 ro root=/dev/mapper/vg_doc01-
lv_root console=ttyS0,115200n8 acpi=off
    initrd /initramfs-2.6.32-131.0.15.el6.x86_64.img
```

In this example, **`acpi=off`** has been appended to the kernel boot command line — the line starting with “kernel /vmlinuz-2.6.32-193.el6.x86_64.img”.

Chapter 2. Configuring Fencing with the `ccs` Command

As of the Red Hat Enterprise Linux 6.1 release and later, the Red Hat High Availability Add-On provides support for the `ccs` cluster configuration command. The `ccs` command allows an administrator to create, modify and view the `cluster.conf` cluster configuration file. You can use the `ccs` command to configure a cluster configuration file on a local file system or on a remote node. Using the `ccs` command, an administrator can also start and stop the cluster services on one or all of the nodes in a configured cluster.

This chapter describes how to configure the Red Hat High Availability Add-On cluster configuration file using the `ccs` command.

This chapter consists of the following sections:

- [Section 2.1, “Configuring Fence Devices”](#)



Important

Make sure that your deployment of High Availability Add-On meets your needs and can be supported. Consult with an authorized Red Hat representative to verify your configuration prior to deployment. In addition, allow time for a configuration burn-in period to test failure modes.



Important

This chapter references commonly used `cluster.conf` elements and attributes. For a comprehensive list and description of `cluster.conf` elements and attributes, refer to the cluster schema at `/usr/share/cluster/cluster.rng`, and the annotated schema at `/usr/share/doc/cman-X.Y.ZZ/cluster_conf.html` (for example `/usr/share/doc/cman-3.0.12/cluster_conf.html`).

2.1. Configuring Fence Devices

Configuring fence devices consists of creating, updating, and deleting fence devices for the cluster. You must create and name the fence devices in a cluster before you can configure fencing for the nodes in the cluster. For information on configuring fencing for the individual nodes in the cluster, see [Section 2.3, “Configuring Fencing for Cluster Members”](#).

Before configuring your fence devices, you may want to modify some of the fence daemon properties for your system from the default values. The values you configure for the fence daemon are general values for the cluster. The general fencing properties for the cluster you may want to modify are summarized as follows:

- The `post_fail_delay` attribute is the number of seconds the fence daemon (**fenced**) waits before fencing a node (a member of the fence domain) after the node has failed. The `post_fail_delay` default value is `0`. Its value may be varied to suit cluster and network performance.
- The `post_join_delay` attribute is the number of seconds the fence daemon (**fenced**) waits before fencing a node after the node joins the fence domain. The `post_join_delay` default value is `6`. A typical setting for `post_join_delay` is between 20 and 30 seconds, but can vary according to cluster and network performance.

You reset the values of the **post_fail_delay** and **post_join_delay** attributes with the **--setfencedaemon** option of the **ccs** command. Note, however, that executing the **ccs --setfencedaemon** command overwrites all existing fence daemon properties.

For example, to configure a value for the **post_fail_delay** attribute, execute the following command. This command will overwrite the values of all other existing fence daemon properties that you can set with this command.

```
ccs -h host --setfencedaemon post_fail_delay=value
```

To configure a value for the **post_join_delay** attribute, execute the following command. This command will overwrite the values of all other existing fence daemon properties that you can set with this command.

```
ccs -h host --setfencedaemon post_join_delay=value
```

To configure a value for both the **post_join_delay** attribute and the **post_fail_delay** attribute, execute the following command:

```
ccs -h host --setfencedaemon post_fail_delay=value post_join_delay=value
```



Note

For more information about the **post_join_delay** and **post_fail_delay** attributes as well as the additional fence daemon properties you can modify, refer to the `fenced(8)` man page and refer to the cluster schema at `/usr/share/cluster/cluster.rng`, and the annotated schema at `/usr/share/doc/cman-X.Y.ZZ/cluster_conf.html`.

To configure a fence device for a cluster, execute the following command:

```
ccs -h host --addfencedev  
devicename  
[fencedeviceoptions]
```

For example, to configure an APC fence device in the configuration file on the cluster node **node1** named **myfence** with an IP address of **apc_ip_example**, a login of **login_example**, and a password of **password_example**, execute the following command:

```
ccs -h node1 --addfencedev myfence agent=fence_apc ipaddr=apc_ip_example  
login=login_example passwd=password_example
```

The following example shows the **fencedevices** section of the **cluster.conf** configuration file after you have added this APC fence device:

```
<fencedevices>  
  <fencedevice agent="fence_apc" ipaddr="apc_ip_example"  
  login="login_example" name="myfence" passwd="password_example"/>  
</fencedevices>
```

When configuring fence devices for a cluster, you may find it useful to see a listing of available devices for your cluster and the options available for each device. You may also find it useful to see a listing of fence devices currently configured for your cluster. For information on using the **ccs** command to print a list of available fence devices and options or to print a list of fence devices currently configured for your cluster, refer to [Section 2.2, “Listing Fence Devices and Fence Device Options”](#).

To remove a fence device from your cluster configuration, execute the following command:

```
ccs -h host --rmfencedev fence_device_name
```

For example, to remove a fence device that you have named **myfence** from the cluster configuration file on cluster node **node1**, execute the following command:

```
ccs -h node1 --rmfencedev myfence
```

If you need to modify the attributes of a fence device you have already configured, you must first remove that fence device then add it again with the modified attributes.

Note that when you have finished configuring all of the components of your cluster, you will need to sync the cluster configuration file to all of the nodes.

2.2. Listing Fence Devices and Fence Device Options

You can use the **ccs** command to print a list of available fence devices and to print a list of options for each available fence type. You can also use the **ccs** command to print a list of fence devices currently configured for your cluster.

To print a list of fence devices currently available for your cluster, execute the following command:

```
ccs -h host --lsfenceopts
```

For example, the following command lists the fence devices available on the cluster node **node1**, showing sample output.

```
[root@ask-03 ~]# ccs -h node1 --lsfenceopts
fence_rps10 - RPS10 Serial Switch
fence_vixel - No description available
fence_egenera - No description available
fence_xcat - No description available
fence_na - Node Assassin
fence_apc - Fence agent for APC over telnet/ssh
fence_apc_snmp - Fence agent for APC over SNMP
fence_bladecenter - Fence agent for IBM BladeCenter
fence_bladecenter_snmp - Fence agent for IBM BladeCenter over SNMP
fence_cisco_mds - Fence agent for Cisco MDS
fence_cisco_ucs - Fence agent for Cisco UCS
fence_drac5 - Fence agent for Dell DRAC CMC/5
fence_eps - Fence agent for ePowerSwitch
fence_ibmblade - Fence agent for IBM BladeCenter over SNMP
fence_ifmib - Fence agent for IF MIB
fence_ilo - Fence agent for HP iLO
fence_ilo_mp - Fence agent for HP iLO MP
fence_intelmodular - Fence agent for Intel Modular
fence_ipmilan - Fence agent for IPMI over LAN
fence_kdump - Fence agent for use with kdump
fence_rhevm - Fence agent for RHEV-M REST API
fence_rsa - Fence agent for IBM RSA
fence_sanbox2 - Fence agent for QLogic SANBox2 FC switches
fence_scsi - fence agent for SCSI-3 persistent reservations
fence_virsh - Fence agent for virsh
fence_virt - Fence agent for virtual machines
fence_vmware - Fence agent for VMware
fence_vmware_soap - Fence agent for VMware over SOAP API
fence_wti - Fence agent for WTI
fence_xvm - Fence agent for virtual machines
```

To print a list of the options you can specify for a particular fence type, execute the following command:

```
ccs -h host --lsfenceopts fence_type
```

For example, the following command lists the fence options for the **fence_wti** fence agent.

```
[root@ask-03 ~]# ccs -h node1 --lsfenceopts fence_wti
fence_wti - Fence agent for WTI
Required Options:
Optional Options:
  option: No description available
  action: Fencing Action
  ipaddr: IP Address or Hostname
  login: Login Name
  passwd: Login password or passphrase
  passwd_script: Script to retrieve password
  cmd_prompt: Force command prompt
  secure: SSH connection
  identity_file: Identity file for ssh
  port: Physical plug number or name of virtual machine
  inet4_only: Forces agent to use IPv4 addresses only
  inet6_only: Forces agent to use IPv6 addresses only
  ipport: TCP port to use for connection with device
  verbose: Verbose mode
  debug: Write debug information to given file
  version: Display version information and exit
  help: Display help and exit
  separator: Separator for CSV created by operation list
  power_timeout: Test X seconds for status change after ON/OFF
  shell_timeout: Wait X seconds for cmd prompt after issuing command
  login_timeout: Wait X seconds for cmd prompt after login
  power_wait: Wait X seconds after issuing ON/OFF
  delay: Wait X seconds before fencing is started
  retry_on: Count of attempts to retry power on
```

To print a list of fence devices currently configured for your cluster, execute the following command:

```
ccs -h host --lsfencedev
```

2.3. Configuring Fencing for Cluster Members

Once you have completed the initial steps of creating a cluster and creating fence devices, you need to configure fencing for the cluster nodes. To configure fencing for the nodes after creating a new cluster and configuring the fencing devices for the cluster, follow the steps in this section. Note that you must configure fencing for each node in the cluster.

This section documents the following procedures:

- ▶ [Section 2.3.1, “Configuring a Single Power-Based Fence Device for a Node”](#)
- ▶ [Section 2.3.2, “Configuring a Single Storage-Based Fence Device for a Node”](#)
- ▶ [Section 2.3.3, “Configuring a Backup Fence Device”](#)
- ▶ [Section 2.3.4, “Configuring a Node with Redundant Power”](#)
- ▶ [Section 2.3.5, “Removing Fence Methods and Fence Instances”](#)

2.3.1. Configuring a Single Power-Based Fence Device for a Node

Use the following procedure to configure a node with a single power-based fence device that uses a fence device named **apc**, which uses the **fence_apc** fencing agent.

1. Add a fence method for the node, providing a name for the fence method.

```
ccs -h host --addmethod method node
```

For example, to configure a fence method named **APC** for the node **node-01.example.com** in the configuration file on the cluster node **node-01.example.com**, execute the following command:

```
ccs -h node01.example.com --addmethod APC node01.example.com
```

2. Add a fence instance for the method. You must specify the fence device to use for the node, the node this instance applies to, the name of the method, and any options for this method that are specific to this node:

```
ccs -h host --addfenceinst fencedevicename node method [options]
```

For example, to configure a fence instance in the configuration file on the cluster node **node-01.example.com** that uses the APC switch power port 1 on the fence device named **apc** to fence cluster node **node-01.example.com** using the method named **APC**, execute the following command:

```
ccs -h node01.example.com --addfenceinst apc node01.example.com APC port=1
```

You will need to add a fence method for each node in the cluster. The following commands configure a fence method for each node with the method name **APC**. The device for the fence method specifies **apc** as the device name, which is a device previously configured with the **--addfencedev** option, as described in [Section 2.1, “Configuring Fence Devices”](#). Each node is configured with a unique APC switch power port number: The port number for **node-01.example.com** is **1**, the port number for **node-02.example.com** is **2**, and the port number for **node-03.example.com** is **3**.

```
ccs -h node01.example.com --addmethod APC node01.example.com
ccs -h node01.example.com --addmethod APC node02.example.com
ccs -h node01.example.com --addmethod APC node03.example.com
ccs -h node01.example.com --addfenceinst apc node01.example.com APC port=1
ccs -h node01.example.com --addfenceinst apc node02.example.com APC port=2
ccs -h node01.example.com --addfenceinst apc node03.example.com APC port=3
```

[Example 2.1, “**cluster.conf** After Adding Power-Based Fence Methods ”](#) shows a **cluster.conf** configuration file after you have added these fencing methods and instances to each node in the cluster.

Example 2.1. cluster.conf After Adding Power-Based Fence Methods

```

<cluster name="mycluster" config_version="3">
  <clusternodes>
    <clusternode name="node-01.example.com" nodeid="1">
      <fence>
        <method name="APC">
          <device name="apc" port="1"/>
        </method>
      </fence>
    </clusternode>
    <clusternode name="node-02.example.com" nodeid="2">
      <fence>
        <method name="APC">
          <device name="apc" port="2"/>
        </method>
      </fence>
    </clusternode>
    <clusternode name="node-03.example.com" nodeid="3">
      <fence>
        <method name="APC">
          <device name="apc" port="3"/>
        </method>
      </fence>
    </clusternode>
  </clusternodes>
  <fencedevices>
    <fencedevice agent="fence_apc" ipaddr="apc_ip_example"
login="login_example" name="apc" passwd="password_example"/>
  </fencedevices>
  <rm>
  </rm>
</cluster>

```

Note that when you have finished configuring all of the components of your cluster, you will need to sync the cluster configuration file to all of the nodes.

2.3.2. Configuring a Single Storage-Based Fence Device for a Node

When using non-power fencing methods (that is, SAN/storage fencing) to fence a node, you must configure *unfencing* for the fence device. This ensures that a fenced node is not re-enabled until the node has been rebooted. When you configure unfencing for a node, you specify a device that mirrors the corresponding fence device you have configured for the node with the notable addition of the explicit action of **on** or **enable**.

For more information about unfencing a node, refer to the **fence_node(8)** man page.

Use the following procedure to configure a node with a single storage-based fence device that uses a fence device named **sanswitch1**, which uses the **fence_sanbox2** fencing agent.

1. Add a fence method for the node, providing a name for the fence method.

```
ccs -h host --addmethod method node
```

For example, to configure a fence method named **SAN** for the node **node-01.example.com** in the configuration file on the cluster node **node-01.example.com**, execute the following command:

```
ccs -h node01.example.com --addmethod SAN node01.example.com
```

2. Add a fence instance for the method. You must specify the fence device to use for the node, the node this instance applies to, the name of the method, and any options for this method that are specific to this node:

```
ccs -h host --addfenceinst fencedevicename node method [options]
```

For example, to configure a fence instance in the configuration file on the cluster node **node-01.example.com** that uses the SAN switch power port 11 on the fence device named **sanswitch1** to fence cluster node **node-01.example.com** using the method named **SAN**, execute the following command:

```
ccs -h node01.example.com --addfenceinst sanswitch1 node01.example.com SAN port=11
```

3. To configure unfencing for the storage based fence device on this node, execute the following command:

```
ccs -h host --addunfence fencedevicename node action=on|off
```

You will need to add a fence method for each node in the cluster. The following commands configure a fence method for each node with the method name **SAN**. The device for the fence method specifies **sanswitch** as the device name, which is a device previously configured with the `--addfencedev` option, as described in [Section 2.1, “Configuring Fence Devices”](#). Each node is configured with a unique SAN physical port number: The port number for **node-01.example.com** is **11**, the port number for **node-02.example.com** is **12**, and the port number for **node-03.example.com** is **13**.

```
ccs -h node01.example.com --addmethod SAN node01.example.com
ccs -h node01.example.com --addmethod SAN node02.example.com
ccs -h node01.example.com --addmethod SAN node03.example.com
ccs -h node01.example.com --addfenceinst sanswitch1 node01.example.com SAN port=11
ccs -h node01.example.com --addfenceinst sanswitch1 node02.example.com SAN port=12
ccs -h node01.example.com --addfenceinst sanswitch1 node03.example.com SAN port=13
ccs -h node01.example.com --addunfence sanswitch1 node01.example.com port=11
action=on
ccs -h node01.example.com --addunfence sanswitch1 node02.example.com port=12
action=on
ccs -h node01.example.com --addunfence sanswitch1 node03.example.com port=13
action=on
```

[Example 2.2, “ccluster.conf After Adding Storage-Based Fence Methods ”](#) shows a **ccluster.conf** configuration file after you have added fencing methods, fencing instances, and unfencing to each node in the cluster.

Example 2.2. cluster.conf After Adding Storage-Based Fence Methods

```

<cluster name="mycluster" config_version="3">
  <clusternodes>
    <clusternode name="node-01.example.com" nodeid="1">
      <fence>
        <method name="SAN">
          <device name="sanswitch1" port="11"/>
        </method>
      </fence>
      <unfence>
        <device name="sanswitch1" port="11" action="on"/>
      </unfence>
    </clusternode>
    <clusternode name="node-02.example.com" nodeid="2">
      <fence>
        <method name="SAN">
          <device name="sanswitch1" port="12"/>
        </method>
      </fence>
      <unfence>
        <device name="sanswitch1" port="12" action="on"/>
      </unfence>
    </clusternode>
    <clusternode name="node-03.example.com" nodeid="3">
      <fence>
        <method name="SAN">
          <device name="sanswitch1" port="13"/>
        </method>
      </fence>
      <unfence>
        <device name="sanswitch1" port="13" action="on"/>
      </unfence>
    </clusternode>
  </clusternodes>
  <fencedevices>
    <fencedevice agent="fence_sanbox2" ipaddr="san_ip_example"
login="login_example" name="sanswitch1" passwd="password_example"/>
  </fencedevices>
  <rm>
  </rm>
</cluster>

```

Note that when you have finished configuring all of the components of your cluster, you will need to sync the cluster configuration file to all of the nodes.

2.3.3. Configuring a Backup Fence Device

You can define multiple fencing methods for a node. If fencing fails using the first method, the system will attempt to fence the node using the second method, followed by any additional methods you have configured. To configure a backup fencing method for a node, you configure two methods for a node, configuring a fence instance for each method.

**Note**

The order in which the system will use the fencing methods you have configured follows their order in the cluster configuration file. The first method you configure with the `ccs` command is the primary fencing method, and the second method you configure is the backup fencing method. To change the order, you can remove the primary fencing method from the configuration file, then add that method back.

Note that at any time you can print a list of fence methods and instances currently configured for a node by executing the following command. If you do not specify a node, this command will list the fence methods and instances currently configured for all nodes.

```
ccs -h host --lsfenceinst [node]
```

Use the following procedure to configure a node with a primary fencing method that uses a fence device named **apc**, which uses the **fence_apc** fencing agent, and a backup fencing device that uses a fence device named **sanswitch1**, which uses the **fence_sanbox2** fencing agent. Since the **sanswitch1** device is a storage-based fencing agent, you will need to configure unfencing for that device as well.

1. Add a primary fence method for the node, providing a name for the fence method.

```
ccs -h host --addmethod method node
```

For example, to configure a fence method named **APC** as the primary method for the node **node-01.example.com** in the configuration file on the cluster node **node-01.example.com**, execute the following command:

```
ccs -h node01.example.com --addmethod APC node01.example.com
```

2. Add a fence instance for the primary method. You must specify the fence device to use for the node, the node this instance applies to, the name of the method, and any options for this method that are specific to this node:

```
ccs -h host --addfenceinst fencedevicename node method [options]
```

For example, to configure a fence instance in the configuration file on the cluster node **node-01.example.com** that uses the APC switch power port 1 on the fence device named **apc** to fence cluster node **node-01.example.com** using the method named **APC**, execute the following command:

```
ccs -h node01.example.com --addfenceinst apc node01.example.com APC port=1
```

3. Add a backup fence method for the node, providing a name for the fence method.

```
ccs -h host --addmethod method node
```

For example, to configure a backup fence method named **SAN** for the node **node-01.example.com** in the configuration file on the cluster node **node-01.example.com**, execute the following command:

```
ccs -h node01.example.com --addmethod SAN node01.example.com
```

4. Add a fence instance for the backup method. You must specify the fence device to use for the node, the node this instance applies to, the name of the method, and any options for this method that are specific to this node:

```
ccs -h host --addfenceinst fencedevicename node method [options]
```

For example, to configure a fence instance in the configuration file on the cluster node **node-01.example.com** that uses the SAN switch power port 11 on the fence device named **sanswitch1** to fence cluster node **node-01.example.com** using the method named **SAN**, execute the following command:

```
ccs -h node01.example.com --addfenceinst sanswitch1 node01.example.com SAN  
port=11
```

5. Since the **sanswitch1** device is a storage-based device, you must configure unfencing for this device.

```
ccs -h node01.example.com --addunfence sanswitch1 node01.example.com port=11  
action=on
```

You can continue to add fencing methods as needed.

This procedure configures a fence device and a backup fence device for one node in the cluster. You will need to configure fencing for the other nodes in the cluster as well.

[Example 2.3, “**cluster.conf** After Adding Backup Fence Methods](#)” shows a **cluster.conf** configuration file after you have added a power-based primary fencing method and a storage-based backup fencing method to each node in the cluster.

Example 2.3. cluster.conf After Adding Backup Fence Methods

```

<cluster name="mycluster" config_version="3">
  <clusternodes>
    <clusternode name="node-01.example.com" nodeid="1">
      <fence>
        <method name="APC">
          <device name="apc" port="1"/>
        </method>
        <method name="SAN">
          <device name="sanswitch1" port="11"/>
        </method>
      </fence>
      <unfence>
        <device name="sanswitch1" port="11" action="on"/>
      </unfence>
    </clusternode>
    <clusternode name="node-02.example.com" nodeid="2">
      <fence>
        <method name="APC">
          <device name="apc" port="2"/>
        </method>
        <method name="SAN">
          <device name="sanswitch1" port="12"/>
        </method>
      </fence>
      <unfence>
        <device name="sanswitch1" port="12" action="on"/>
      </unfence>
    </clusternode>
    <clusternode name="node-03.example.com" nodeid="3">
      <fence>
        <method name="APC">
          <device name="apc" port="3"/>
        </method>
        <method name="SAN">
          <device name="sanswitch1" port="13"/>
        </method>
      </fence>
      <unfence>
        <device name="sanswitch1" port="13" action="on"/>
      </unfence>
    </clusternode>
  </clusternodes>
  <fencedevices>
    <fencedevice agent="fence_apc" ipaddr="apc_ip_example"
login="login_example" name="apc" passwd="password_example"/>
    <fencedevice agent="fence_sanbox2" ipaddr="san_ip_example"
login="login_example" name="sanswitch1" passwd="password_example"/>
  </fencedevices>
  <rm>
  </rm>
</cluster>

```

Note that when you have finished configuring all of the components of your cluster, you will need to sync

the cluster configuration file to all of the nodes.



Note

The order in which the system will use the fencing methods you have configured follows their order in the cluster configuration file. The first method you configure is the primary fencing method, and the second method you configure is the backup fencing method. To change the order, you can remove the primary fencing method from the configuration file, then add that method back.

2.3.4. Configuring a Node with Redundant Power

If your cluster is configured with redundant power supplies for your nodes, you must be sure to configure fencing so that your nodes fully shut down when they need to be fenced. If you configure each power supply as a separate fence method, each power supply will be fenced separately; the second power supply will allow the system to continue running when the first power supply is fenced and the system will not be fenced at all. To configure a system with dual power supplies, you must configure your fence devices so that both power supplies are shut off and the system is taken completely down. This requires that you configure two instances within a single fencing method, and that for each instance you configure both fence devices with an **action** attribute of **off** before configuring each of the devices with an **action** attribute of **on**.

To configure fencing for a node with dual power supplies, follow the steps in this section.

1. Before you can configure fencing for a node with redundant power, you must configure each of the power switches as a fence device for the cluster. For information on configuring fence devices, see [Section 2.1, “Configuring Fence Devices”](#).

To print a list of fence devices currently configured for your cluster, execute the following command:

```
ccs -h host --lsfencedev
```

2. Add a fence method for the node, providing a name for the fence method.

```
ccs -h host --addmethod method node
```

For example, to configure a fence method named **APC-dual** for the node **node-01.example.com** in the configuration file on the cluster node **node-01.example.com**, execute the following command:

```
ccs -h node01.example.com --addmethod APC-dual node01.example.com
```

3. Add a fence instance for the first power supply to the fence method. You must specify the fence device to use for the node, the node this instance applies to, the name of the method, and any options for this method that are specific to this node. At this point you configure the **action** attribute as **off**.

```
ccs -h host --addfenceinst fencedevicename node method [options] action=off
```

For example, to configure a fence instance in the configuration file on the cluster node **node-01.example.com** that uses the APC switch power port 1 on the fence device named **apc1** to fence cluster node **node-01.example.com** using the method named **APC-dual**, and setting

the **action** attribute to **off**, execute the following command:

```
ccs -h node01.example.com --addfenceinst apc1 node01.example.com APC-dual
port=1 action=off
```

4. Add a fence instance for the second power supply to the fence method. You must specify the fence device to use for the node, the node this instance applies to, the name of the method, and any options for this method that are specific to this node. At this point you configure the **action** attribute as **off** for this instance as well:

```
ccs -h host --addfenceinst fencedevicename node method [options] action=off
```

For example, to configure a second fence instance in the configuration file on the cluster node **node-01.example.com** that uses the APC switch power port 1 on the fence device named **apc2** to fence cluster node **node-01.example.com** using the same method as you specified for the first instance named **APC-dual**, and setting the **action** attribute to **off**, execute the following command:

```
ccs -h node01.example.com --addfenceinst apc2 node01.example.com APC-dual
port=1 action=off
```

5. At this point, add another fence instance for the first power supply to the fence method, configuring the **action** attribute as **on**. You must specify the fence device to use for the node, the node this instance applies to, the name of the method, and any options for this method that are specific to this node, and specifying the **action** attribute as **on**:

```
ccs -h host --addfenceinst fencedevicename node method [options] action=on
```

For example, to configure a fence instance in the configuration file on the cluster node **node-01.example.com** that uses the APC switch power port 1 on the fence device named **apc1** to fence cluster node **node-01.example.com** using the method named **APC-dual**, and setting the **action** attribute to **on**, execute the following command:

```
ccs -h node01.example.com --addfenceinst apc1 node01.example.com APC-dual
port=1 action=on
```

6. Add another fence instance for second power supply to the fence method, specifying the **action** attribute as **on** for this instance. You must specify the fence device to use for the node, the node this instance applies to, the name of the method, and any options for this method that are specific to this node as well as the **action** attribute of **on**.

```
ccs -h host --addfenceinst fencedevicename node method [options] action=on
```

For example, to configure a second fence instance in the configuration file on the cluster node **node-01.example.com** that uses the APC switch power port 1 on the fence device named **apc2** to fence cluster node **node-01.example.com** using the same method as you specified for the first instance named **APC-dual** and setting the **action** attribute to **on**, execute the following command:

```
ccs -h node01.example.com --addfenceinst apc2 node01.example.com APC-dual
port=1 action=on
```

[Example 2.4, “cluster.conf After Adding Dual-Power Fencing.”](#) shows a **cluster.conf**

configuration file after you have added fencing for two power supplies for each node in a cluster.

Example 2.4. `cluster.conf` After Adding Dual-Power Fencing

```
<cluster name="mycluster" config_version="3">
  <clusternodes>
    <clusternode name="node-01.example.com" nodeid="1">
      <fence>
        <method name="APC-dual">
          <device name="apc1" port="1"action="off"/>
          <device name="apc2" port="1"action="off"/>
          <device name="apc1" port="1"action="on"/>
          <device name="apc2" port="1"action="on"/>
        </method>
      </fence>
    </clusternode>
    <clusternode name="node-02.example.com" nodeid="2">
      <fence>
        <method name="APC-dual">
          <device name="apc1" port="2"action="off"/>
          <device name="apc2" port="2"action="off"/>
          <device name="apc1" port="2"action="on"/>
          <device name="apc2" port="2"action="on"/>
        </method>
      </fence>
    </clusternode>
    <clusternode name="node-03.example.com" nodeid="3">
      <fence>
        <method name="APC-dual">
          <device name="apc1" port="3"action="off"/>
          <device name="apc2" port="3"action="off"/>
          <device name="apc1" port="3"action="on"/>
          <device name="apc2" port="3"action="on"/>
        </method>
      </fence>
    </clusternode>
  </clusternodes>
  <fencedevices>
    <fencedevice agent="fence_apc" ipaddr="apc_ip_example"
login="login_example" name="apc1" passwd="password_example"/>
    <fencedevice agent="fence_apc" ipaddr="apc_ip_example"
login="login_example" name="apc2" passwd="password_example"/>
  </fencedevices>
  <rm>
  </rm>
</cluster>
```

Note that when you have finished configuring all of the components of your cluster, you will need to sync the cluster configuration file to all of the nodes.

2.3.5. Removing Fence Methods and Fence Instances

To remove a fence method from your cluster configuration, execute the following command:

```
ccs -h host --rmmethod method node
```

For example, to remove a fence method that you have named **APC** that you have configured for **node01.example.com** from the cluster configuration file on cluster node **node01.example.com**, execute the following command:

```
ccs -h node01.example.com --rmmethod APC node01.example.com
```

To remove all fence instances of a fence device from a fence method, execute the following command:

```
ccs -h host --rmfenceinst fencedevicename node method
```

For example, to remove all instances of the fence device named **apc1** from the method named **APC-dual** configured for **node01.example.com** from the cluster configuration file on cluster node **node01.example.com**, execute the following command:

```
ccs -h node01.example.com --rmfenceinst apc1 node01.example.com APC-dual
```

Chapter 3. Configuring Fencing with Conga

This chapter describes how to configure fencing in Red Hat High Availability Add-On using **Conga**.



Note

Conga is a graphical user interface that you can use to administer the Red Hat High Availability Add-On. Note, however, that in order to use this interface effectively you need to have a good and clear understanding of the underlying concepts. Learning about cluster configuration by exploring the available features in the user interface is not recommended, as it may result in a system that is not robust enough to keep all services running when components fail.

► [Section 3.2, “Configuring Fence Devices”](#)

3.1. Configuring Fence Daemon Properties

Clicking on the **Fence Daemon** tab displays the **Fence Daemon Properties** page, which provides an interface for configuring **Post Fail Delay** and **Post Join Delay**. The values you configure for these parameters are general fencing properties for the cluster. To configure specific fence devices for the nodes of the cluster, use the **Fence Devices** menu item of the cluster display, as described in [Section 3.2, “Configuring Fence Devices”](#).

- The **Post Fail Delay** parameter is the number of seconds the fence daemon (**fenced**) waits before fencing a node (a member of the fence domain) after the node has failed. The **Post Fail Delay** default value is **0**. Its value may be varied to suit cluster and network performance.
- The **Post Join Delay** parameter is the number of seconds the fence daemon (**fenced**) waits before fencing a node after the node joins the fence domain. The **Post Join Delay** default value is **6**. A typical setting for **Post Join Delay** is between 20 and 30 seconds, but can vary according to cluster and network performance.

Enter the values required and click **Apply** for changes to take effect.



Note

For more information about **Post Join Delay** and **Post Fail Delay**, refer to the **fenced(8)** man page.

3.2. Configuring Fence Devices

Configuring fence devices consists of creating, updating, and deleting fence devices for the cluster. You must configure the fence devices in a cluster before you can configure fencing for the nodes in the cluster.

Creating a fence device consists of selecting a fence device type and entering parameters for that fence device (for example, name, IP address, login, and password). Updating a fence device consists of selecting an existing fence device and changing parameters for that fence device. Deleting a fence device consists of selecting an existing fence device and deleting it.

This section provides procedures for the following tasks:

- Creating fence devices — Refer to [Section 3.2.1, “Creating a Fence Device”](#). Once you have created and named a fence device, you can configure the fence devices for each node in the cluster, as described in [Section 3.3, “Configuring Fencing for Cluster Members”](#).
- Updating fence devices — Refer to [Section 3.2.2, “Modifying a Fence Device”](#).
- Deleting fence devices — Refer to [Section 3.2.3, “Deleting a Fence Device”](#).

From the cluster-specific page, you can configure fence devices for that cluster by clicking on **Fence Devices** along the top of the cluster display. This displays the fence devices for the cluster and displays the menu items for fence device configuration: **Add** and **Delete**. This is the starting point of each procedure described in the following sections.



Note

If this is an initial cluster configuration, no fence devices have been created, and therefore none are displayed.

[Figure 3.1, “luci fence devices configuration page”](#) shows the fence devices configuration screen before any fence devices have been created.

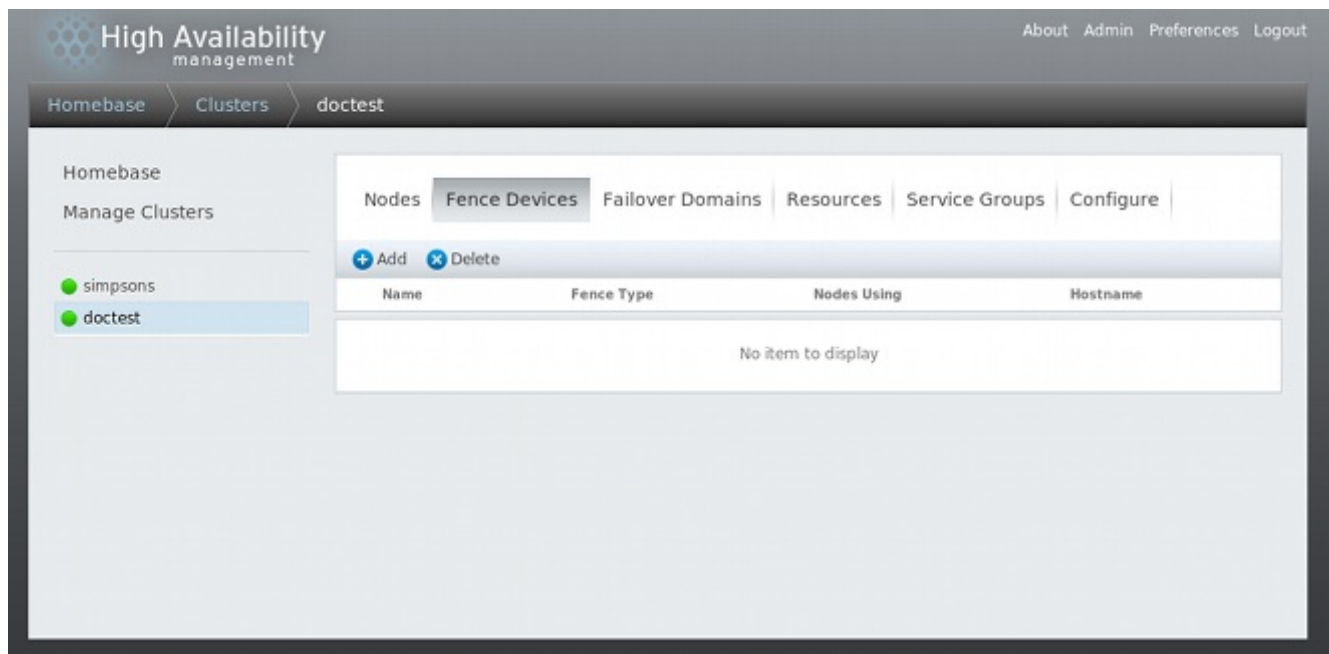


Figure 3.1. luci fence devices configuration page

3.2.1. Creating a Fence Device

To create a fence device, follow these steps:

1. From the **Fence Devices** configuration page, click **Add**. Clicking **Add** displays the **Add Fence Device (Instance)** dialog box. From this dialog box, select the type of fence device to configure.
2. Specify the information in the **Add Fence Device (Instance)** dialog box according to the type of fence device. In some cases you will need to specify additional node-specific parameters for the fence device when you configure fencing for the individual nodes.

3. Click **Submit**.

After the fence device has been added, it appears on the **Fence Devices** configuration page.

3.2.2. Modifying a Fence Device

To modify a fence device, follow these steps:

1. From the **Fence Devices** configuration page, click on the name of the fence device to modify. This displays the dialog box for that fence device, with the values that have been configured for the device.
2. To modify the fence device, enter changes to the parameters displayed.
3. Click **Apply** and wait for the configuration to be updated.

3.2.3. Deleting a Fence Device



Note

Fence devices that are in use cannot be deleted. To delete a fence device that a node is currently using, first update the node fence configuration for any node using the device and then delete the device.

To delete a fence device, follow these steps:

1. From the **Fence Devices** configuration page, check the box to the left of the fence device or devices to select the devices to delete.
2. Click **Delete** and wait for the configuration to be updated. A message appears indicating which devices are being deleted.

When the configuration has been updated, the deleted fence device no longer appears in the display.

3.3. Configuring Fencing for Cluster Members

Once you have completed the initial steps of creating a cluster and creating fence devices, you need to configure fencing for the cluster nodes. To configure fencing for the nodes after creating a new cluster and configuring the fencing devices for the cluster, follow the steps in this section. Note that you must configure fencing for each node in the cluster.

The following sections provide procedures for configuring a single fence device for a node, configuring a node with a backup fence device, and configuring a node with redundant power supplies:

- ▶ [Section 3.3.1, “Configuring a Single Fence Device for a Node”](#)
- ▶ [Section 3.3.2, “Configuring a Backup Fence Device”](#)
- ▶ [Section 3.3.3, “Configuring a Node with Redundant Power”](#)

3.3.1. Configuring a Single Fence Device for a Node

Use the following procedure to configure a node with a single fence device.

1. From the cluster-specific page, you can configure fencing for the nodes in the cluster by clicking on **Nodes** along the top of the cluster display. This displays the nodes that constitute the cluster. This is also the default page that appears when you click on the cluster name beneath **Manage Clusters** from the menu on the left side of the **luci Homebase** page.

2. Click on a node name. Clicking a link for a node causes a page to be displayed for that link showing how that node is configured.
The node-specific page displays any services that are currently running on the node, as well as any failover domains of which this node is a member. You can modify an existing failover domain by clicking on its name.
3. On the node-specific page, under **Fence Devices**, click **Add Fence Method**. This displays the **Add Fence Method to Node** dialog box.
4. Enter a **Method Name** for the fencing method that you are configuring for this node. This is an arbitrary name that will be used by Red Hat High Availability Add-On; it is not the same as the DNS name for the device.
5. Click **Submit**. This displays the node-specific screen that now displays the method you have just added under **Fence Devices**.
6. Configure a fence instance for this method by clicking the **Add Fence Instance** button that appears beneath the fence method. This displays the **Add Fence Device (Instance)** drop-down menu from which you can select a fence device you have previously configured, as described in [Section 3.2.1, “Creating a Fence Device”](#).
7. Select a fence device for this method. If this fence device requires that you configure node-specific parameters, the display shows the parameters to configure.



Note

For non-power fence methods (that is, SAN/storage fencing), **Unfencing** is selected by default on the node-specific parameters display. This ensures that a fenced node's access to storage is not re-enabled until the node has been rebooted. For information on unfencing a node, refer to the **fence_node(8)** man page.

8. Click **Submit**. This returns you to the node-specific screen with the fence method and fence instance displayed.

3.3.2. Configuring a Backup Fence Device

You can define multiple fencing methods for a node. If fencing fails using the first method, the system will attempt to fence the node using the second method, followed by any additional methods you have configured.

Use the following procedure to configure a backup fence device for a node.

1. Use the procedure provided in [Section 3.3.1, “Configuring a Single Fence Device for a Node”](#) to configure the primary fencing method for a node.
2. Beneath the display of the primary method you defined, click **Add Fence Method**.
3. Enter a name for the backup fencing method that you are configuring for this node and click **Submit**. This displays the node-specific screen that now displays the method you have just added, below the primary fence method.
4. Configure a fence instance for this method by clicking **Add Fence Instance**. This displays a drop-down menu from which you can select a fence device you have previously configured, as described in [Section 3.2.1, “Creating a Fence Device”](#).
5. Select a fence device for this method. If this fence device requires that you configure node-specific parameters, the display shows the parameters to configure.
6. Click **Submit**. This returns you to the node-specific screen with the fence method and fence instance displayed.

You can continue to add fencing methods as needed. You can rearrange the order of fencing methods that will be used for this node by clicking on **Move Up** and **Move Down**.

3.3.3. Configuring a Node with Redundant Power

If your cluster is configured with redundant power supplies for your nodes, you must be sure to configure fencing so that your nodes fully shut down when they need to be fenced. If you configure each power supply as a separate fence method, each power supply will be fenced separately; the second power supply will allow the system to continue running when the first power supply is fenced and the system will not be fenced at all. To configure a system with dual power supplies, you must configure your fence devices so that both power supplies are shut off and the system is taken completely down. When configuring your system using **Conga**, this requires that you configure two instances within a single fencing method.

To configure fencing for a node with dual power supplies, follow the steps in this section.

1. Before you can configure fencing for a node with redundant power, you must configure each of the power switches as a fence device for the cluster. For information on configuring fence devices, see [Section 3.2, “Configuring Fence Devices”](#).
2. From the cluster-specific page, click on **Nodes** along the top of the cluster display. This displays the nodes that constitute the cluster. This is also the default page that appears when you click on the cluster name beneath **Manage Clusters** from the menu on the left side of the **luci Homebase** page.
3. Click on a node name. Clicking a link for a node causes a page to be displayed for that link showing how that node is configured.
4. On the node-specific page, click **Add Fence Method**.
5. Enter a name for the fencing method that you are configuring for this node.
6. Click **Submit**. This displays the node-specific screen that now displays the method you have just added under **Fence Devices**.
7. Configure the first power supply as a fence instance for this method by clicking **Add Fence Instance**. This displays a drop-down menu from which you can select one of the power fencing devices you have previously configured, as described in [Section 3.2.1, “Creating a Fence Device”](#).
8. Select one of the power fence devices for this method and enter the appropriate parameters for this device.
9. Click **Submit**. This returns you to the node-specific screen with the fence method and fence instance displayed.
10. Under the same fence method for which you have configured the first power fencing device, click **Add Fence Instance**. This displays a drop-down menu from which you can select the second power fencing devices you have previously configured, as described in [Section 3.2.1, “Creating a Fence Device”](#).
11. Select the second of the power fence devices for this method and enter the appropriate parameters for this device.
12. Click **Submit**. This returns you to the node-specific screen with the fence methods and fence instances displayed, showing that each device will power the system off in sequence and power the system on in sequence. This is shown in [Figure 3.2, “Dual-Power Fencing Configuration”](#).

Fence Devices

Method

Method1 [Remove](#)

Name	Type/Values	
pwr01	APC Power Device port : 1 option : off	✕
pwr02	APC Power Device port : 1 option : off	✕
pwr01	APC Power Device port : 1 option : on	✕
pwr02	APC Power Device port : 1 option : on	✕

[Add Fence Instance](#)

[Add Fence Method](#)

Cluster Daemons

	Status
cman	Running
rgmanager	Not running

Figure 3.2. Dual-Power Fencing Configuration

Chapter 4. Fence Devices

This chapter documents the fence devices currently supported in Red Hat Enterprise Linux High-Availability Add-On.

[Table 4.1, “Fence Device Summary”](#) lists the fence devices, the fence device agents associated with the fence devices, and provides a reference to the table documenting the parameters for the fence devices.

Table 4.1. Fence Device Summary

Fence Device	Fence Agent	Reference to Parameter Description
APC Power Switch (telnet/SSH)	fence_apc	Section 4.2, “APC Power Switch over Telnet and SSH”
APC Power Switch (SNMP)	fence_apc_snmp	Section 4.1, “APC Power Switch over Telnet and SSH”
Brocade Fabric Switch	fence_brocade	Section 4.3, “Brocade Fabric Switch”
Cisco MDS	fence_cisco_mds	Section 4.4, “Cisco MDS”
Cisco UCS	fence_cisco_ucs	Section 4.5, “Cisco UCS”
Dell DRAC 5	fence_drac5	Section 4.6, “Dell Drac 5”
Eaton Network Power Switch	fence_eaton_snmp	Section 4.7, “Eaton Network Power Switch”
Egenera SAN Controller	fence_egenera	Section 4.8, “Egenera BladeFrame”
ePowerSwitch	fence_eps	Section 4.9, “ePowerSwitch”
Fence virt	fence_virt	Section 4.10, “Fence Virt”
Fujitsu Siemens Remoteview Service Board (RSB)	fence_rsb	Section 4.11, “Fujitsu-Siemens RemoteView Service Board (RSB)”
HP BladeSystem	fence_hpblade	Section 4.12, “Hewlett-Packard BladeSystem”
HP iLO/iLO2 (Integrated Lights Out)	fence_ilo	Section 4.13, “Hewlett-Packard iLO”
HP iLO (Integrated Lights Out) MP	fence_ilo_mp	Section 4.14, “Hewlett-Packard iLO MP”
IBM BladeCenter	fence_bladecenter	Section 4.15, “IBM BladeCenter”
IBM BladeCenter SNMP	fence_ibmblade	Section 4.16, “IBM BladeCenter over SNMP”
IBM iPDU	fence_ipdu	Section 4.17, “IBM iPDU”
IF MIB	fence_ifmib	Section 4.18, “IF-MIB”
Intel Modular	fence_intelmodular	Section 4.19, “Intel Modular”
IPMI (Intelligent Platform Management Interface) LAN	fence_ipmilan	Section 4.20, “IPMI over LAN”
RHEV-M REST API	fence_rhevm	Section 4.21, “RHEV-M REST API”
SCSI Fencing	fence_scsi	Section 4.22, “SCSI Persistent

		Reservations"
VMware Fencing (SOAP Interface)	fence_vmware_soap	Section 4.23, "VMWare over SOAP API"
WTI Power Switch	fence_wti	Section 4.24, "WTI Power Switch"

4.1. APC Power Switch over Telnet and SSH

[Table 4.2, "APC Power Switch over SNMP"](#) lists the fence device parameters used by **fence_apc_snmp**, the fence agent for APC that logs into the SNP device via the SNMP protocol.

Table 4.2. APC Power Switch over SNMP

luci Field	cluster.conf Attribute	Description
Name	name	A name for the APC device connected to the cluster into which the fence daemon logs via the SNMP protocol.
IP Address or Hostname	ipaddr	The IP address or hostname assigned to the device.
UDP/TCP port	udpport	The UDP/TCP port to use for connection with the device; the default value is 161.
Login	login	The login name used to access the device.
Password	passwd	The password used to authenticate the connection to the device.
Password Script (optional)	passwd_script	The script that supplies a password for access to the fence device. Using this supersedes the Password parameter.
SNMP Version	snmp_version	The SNMP version to use (1, 2c, 3); the default value is 1.
SNMP Community	community	The SNMP community string; the default value is private .
SNMP Security Level	snmp_sec_level	The SNMP security level (noAuthNoPriv, authNoPriv, authPriv).
SNMP Authentication Protocol	snmp_auth_protocol	The SNMP authentication protocol (MD5, SHA).
SNMP Privacy Protocol	snmp_priv_protocol	The SNMP privacy protocol (DES, AES).
SNMP Privacy Protocol Password	snmp_priv_protocol_password	The SNMP privacy protocol password.
SNMP Privacy Protocol Script	snmp_priv_protocol_password_script	The script that supplies a password for SNMP privacy protocol. Using this supersedes the SNMP privacy protocol password parameter.
Power Wait	power_wait	Number of seconds to wait after issuing a power off or power on command.
Port (Outlet) Number	port	The port.

[Figure 4.1, “APC Power Switch over SNMP”](#) shows the configuration screen for adding an APC Power Switch fence device.

Add Fence Device (Instance)

APC Power Switch (SNMP interface) ⬇	
Fence Type	APC Power Switch (SNMP interface)
Name	
IP Address or Hostname	
UDP/TCP Port (optional, defaults to 161)	
Login	
Password	
Password Script (optional)	
SNMP Version	Default ⬇
SNMP Community	
SNMP Security Level	Default ⬇
SNMP Authentication Protocol	Default ⬇
SNMP Privacy Protocol	Default ⬇
SNMP Privacy Protocol Password	
SNMP Privacy Protocol Script	
Power Wait (seconds)	

Figure 4.1. APC Power Switch over SNMP

The following is the **cluster.conf** entry for the **fence_apc_snmp** device:

```
<fencedevice>
  <fencedevice agent="fence_apc_snmp" community="private" ipaddr="192.168.0.1"
login="root" \
  name="apcpwsnmptst1" passwd="password123" power_wait="60"
snmp_priv_passwd="password123"/>
</fencedevices>
```

4.2. APC Power Switch over Telnet and SSH

[Table 4.3, “APC Power Switch \(telnet/SSH\)”](#) lists the fence device parameters used by **fence_apc**, the fence agent for APC over telnet/SSH.

Table 4.3. APC Power Switch (telnet/SSH)

luci Field	cluster.conf Attribute	Description
Name	name	A name for the APC device connected to the cluster into which the fence daemon logs via telnet/ssh.
IP Address or Hostname	ipaddr	The IP address or hostname assigned to the device.
IP Port (optional)	ipport	The TCP port to use to connect to the device..
Login	login	The login name used to access the device.
Password	passwd	The password used to authenticate the connection to the device.
Password Script (optional)	passwd_script	The script that supplies a password for access to the fence device. Using this supersedes the Password parameter.
Power wait	power_wait	Number of seconds to wait after issuing a power off or power on command.
Port	port	The port.
Switch (optional)	switch	The switch number for the APC switch that connects to the node when you have multiple daisy-chained switches.
Use SSH	secure	Indicates that system will use SSH to access the device. When using SSH, you must specify either a password, a password script, or an identity file.
Path to SSH Identity File	identity_file	The identity file for SSH.

[Figure 4.2, “APC Power Switch”](#) shows the configuration screen for adding an APC Power Switch fence device.

Add Fence Device (Instance)

APC Power Switch	
Fence Type	APC Power Switch
Name	
IP Address or Hostname	
IP Port (optional)	
Login	
Password	
Password Script (optional)	
Power Wait (seconds)	

Figure 4.2. APC Power Switch

The following command creates a fence device instance for a APC device:

```
ccs -f cluster.conf --addfencedev apc agent=fence_apc ipaddr=192.168.0.1  
login=root passwd=password123
```

The following is the **cluster.conf** entry for the **fence_apc** device:

```
<fencedevices>  
  <fencedevice agent="fence_apc" name="apc" ipaddr="apc-telnet.example.com"  
login="root" passwd="password123"/>  
</fencedevices>
```

4.3. Brocade Fabric Switch

[Table 4.4, “Brocade Fabric Switch”](#) lists the fence device parameters used by **fence_brocade**, the fence agent for Brocade FC switches.

Table 4.4. Brocade Fabric Switch

luci Field	cluster.conf Attribute	Description
Name	name	A name for the Brocade device connected to the cluster.
IP Address or Hostname	ipaddr	The IP address assigned to the device.
Login	login	The login name used to access the device.
Password	passwd	The password used to authenticate the connection to the device.
Password Script (optional)	passwd_script	The script that supplies a password for access to the fence device. Using this supersedes the Password parameter.
Port	port	The switch outlet number.

Figure 4.3, “Brocade Fabric Switch” shows the configuration screen for adding an Brocade Fabric Switch fence device.

Add Fence Device (Instance)

Brocade Fabric Switch
↕

Fence Type	Brocade Fabric Switch
Name	
IP Address or Hostname	
Login	
Password	
Password Script (optional)	

Figure 4.3. Brocade Fabric Switch

The following command creates a fence device instance for a Brocade device:

```
ccs -f cluster.conf --addfencedev brocadetest agent=fence_brocade
ipaddr=brocadetest.example.com login=root \
passwd=password123
```

The following is the **cluster.conf** entry for the **fence_brocade** device:

```
<fencedevices>
  <fencedevice agent="fence_brocade" ipaddr="brocadetest.example.com"
login="brocadetest" \
  name="brocadetest" passwd="brocadetest"/>
</fencedevices>
```

4.4. Cisco MDS

[Table 4.5, “Cisco MDS”](#) lists the fence device parameters used by **fence_cisco_mds**, the fence agent for Cisco MDS.

Table 4.5. Cisco MDS

luci Field	cluster.conf Attribute	Description
Name	name	A name for the Cisco MDS 9000 series device with SNMP enabled.
IP Address or Hostname	ipaddr	The IP address or hostname assigned to the device.
UDP/TCP port	udpport	The UDP/TCP port to use for connection with the device; the default value is 161.
Login	login	The login name used to access the device.
Password	passwd	The password used to authenticate the connection to the device.
Password Script (optional)	passwd_script	The script that supplies a password for access to the fence device. Using this supersedes the Password parameter.
Port (Outlet) Number	port	The port.
SNMP Version	snmp_version	The SNMP version to use (1, 2c, 3).
SNMP Community	community	The SNMP community string.
SNMP Security Level	snmp_sec_level	The SNMP security level (noAuthNoPriv, authNoPriv, authPriv).
SNMP Authentication Protocol	snmp_auth_protocol	The SNMP authentication protocol (MD5, SHA).
SNMP Privacy Protocol	snmp_priv_protocol	The SNMP privacy protocol (DES, AES).
SNMP Privacy Protocol Password	snmp_priv_protocol_password	The SNMP privacy protocol password.
SNMP Privacy Protocol Script	snmp_priv_protocol_script	The script that supplies a password for SNMP privacy protocol. Using this supersedes the SNMP privacy protocol password parameter.
Power Wait	power_wait	Number of seconds to wait after issuing a power off or power on command.

Figure 4.4. “Cisco MDS” shows the configuration screen for adding an Cisco MDS fence device.

Add Fence Device (Instance)

Cisco MDS
⌵

Fence Type	Cisco MDS
Name	
IP Address or Hostname	
UDP/TCP Port (optional, defaults to 161)	
Login	
Password	
Password Script (optional)	
SNMP Version	Default ⌵
SNMP Community	
SNMP Security Level	Default ⌵
SNMP Authentication Protocol	Default ⌵
SNMP Privacy Protocol	Default ⌵
SNMP Privacy Protocol Password	
SNMP Privacy Protocol Script	
Power Wait (seconds)	

Figure 4.4. Cisco MDS

The following command creates a fence device instance for a Cisco MDS device:

```
ccs -f cluster.conf --addfencedev mds agent=fence_cisco_mds ipaddr=192.168.0.1
name=ciscomdstest1 login=root \
passwd=password123 power_wait=60 snmp_priv_passwd=password123 udpport=161
```

The following is the **cluster.conf** entry for the **fence_cisco_mds** device:

```
<fencedevices>
  <fencedevice agent="fence_cisco_mds" community="private" ipaddr="192.168.0.1"
login="root" \
    name="ciscomdstest1" passwd="password123" power_wait="60"
snmp_priv_passwd="password123" \
    udpport="161"/>
</fencedevices>
```

4.5. Cisco UCS

[Table 4.6, “Cisco UCS”](#) lists the fence device parameters used by **fence_cisco_ucs**, the fence agent for Cisco UCS.

Table 4.6. Cisco UCS

luci Field	cluster.conf Attribute	Description
Name	name	A name for the Cisco UCS device.
IP Address or Hostname	ipaddr	The IP address or hostname assigned to the device.
IP port (optional)	ipport	The TCP port to use to connect to the device.
Login	login	The login name used to access the device.
Password	passwd	The password used to authenticate the connection to the device.
Password Script (optional)	passwd_script	The script that supplies a password for access to the fence device. Using this supersedes the Password parameter.
Use SSL	ssl	Use SSL connections to communicate with the device.
Sub-Organization	suborg	Additional path needed to access suborganization.
Port (Outlet) Number	port	Name of virtual machine.
Power Wait	power_wait	Number of seconds to wait after issuing a power off or power on command.

[Figure 4.5, “Cisco UCS”](#) shows the configuration screen for adding a Cisco UCS fence device.

Add Fence Device (Instance)

Cisco UCS	
Fence Type	Cisco UCS
Name	
IP Address or Hostname	
IP Port (optional)	
Login	
Password	
Password Script (optional)	
Use SSL	☐
Sub-Organization	
Power Wait (seconds)	

Figure 4.5. Cisco UCS

The following command creates a fence device instance for a Cisco UCS device:

```
ccs -f cluster.conf --addfencedev ucs agent=fence_cisco_ucs ipaddr=192.168.0.1
login=root passwd=password123 \
suborg=/org-RHEL/org-Fence/
```

The following is an example **cluster.conf** entry for the **fence_cisco_ucs** device as created using either Conga or **ccs**:

```
<fencedevices>
  <fencedevice agent="fence_cisco_ucs" ipaddr="192.168.0.1" login="root"
name="ciscoucstest1" \
  passwd="password123" power_wait="60" ssl="on" suborg="/org-RHEL/org-Fence/" />
</fencedevices>
```

4.6. Dell Drac 5

[Table 4.7, “Dell DRAC 5”](#) lists the fence device parameters used by **fence_drac5**, the fence agent for Dell DRAC 5.

Table 4.7. Dell DRAC 5

luci Field	cluster.conf Attribute	Description
Name	name	The name assigned to the DRAC.
IP Address or Hostname	ipaddr	The IP address or hostname assigned to the DRAC.
IP Port (optional)	ipport	The TCP port to use to connect to the device.
Login	login	The login name used to access the DRAC.
Password	passwd	The password used to authenticate the connection to the DRAC.
Password Script (optional)	passwd_script	The script that supplies a password for access to the fence device. Using this supersedes the Password parameter.
Use SSH	secure	Indicates that the system will use SSH to access the device.
Path to SSH Identity File	identity_file	The identity file for SSH.
Module Name	module_name	(optional) The module name for the DRAC when you have multiple DRAC modules.
Force Command Prompt	cmd_prompt	The command prompt to use. The default value is '\$'.
Power Wait	power_wait	Number of seconds to wait after issuing a power off or power on command.

[Figure 4.6, “Dell Drac 5”](#) shows the configuration screen for adding a Dell Drac 5 device

Add Fence Device (Instance)

Dell DRAC 5

Fence Type	Dell DRAC 5
Name	
IP Address or Hostname	
IP Port (optional)	
Login	
Password	
Password Script (optional)	
SSH	☐ Use SSH
Path to SSH Identity File	
Module Name	
Force Command Prompt	
Power Wait (seconds)	

Figure 4.6. Dell Drac 5

The following command creates a fence device instance for a Dell Drac 5 device:

```
ccs -f cluster.conf --addfencedev delldrac5test1 agent=fence_drac5
ipaddr=192.168.0.1 login=root passwd=password123\
module_name=drac1 power_wait=60
```

The following is the **cluster.conf** entry for the **fence_drac5** device:

```
<fencedevices>
  <fencedevice agent="fence_drac5" cmd_prompt="\$" ipaddr="192.168.0.1"
login="root" module_name="drac1" \
  name="delldrac5test1" passwd="password123" power_wait="60"/>
</fencedevices>
```

4.7. Eaton Network Power Switch

[Table 4.8 “Eaton Network Power Controller \(SNMP Interface\) \(Red Hat Enterprise Linux 6.4 and later\)”](#) lists the fence device parameters used by **fence_eaton_snmp**, the fence agent for the Eaton over SNMP network power switch.

Table 4.8. Eaton Network Power Controller (SNMP Interface) (Red Hat Enterprise Linux 6.4 and later)

luci Field	cluster.conf Attribute	Description
Name	name	A name for the Eaton network power switch connected to the cluster.
IP Address or Hostname	ipaddr	The IP address or hostname assigned to the device.
UDP/TCP Port (optional)	udpport	The UDP/TCP port to use for connection with the device; the default value is 161.
Login	login	The login name used to access the device.
Password	passwd	The password used to authenticate the connection to the device.
Password Script (optional)	passwd_script	The script that supplies a password for access to the fence device. Using this supersedes the Password parameter.
SNMP Version	snmp_version	The SNMP version to use (1, 2c, 3); the default value is 1.
SNMP Community	community	The SNMP community string; the default value is private .
SNMP Security Level	snmp_sec_level	The SNMP security level (noAuthNoPriv, authNoPriv, authPriv).
SNMP Authentication Protocol	snmp_auth_protocol	The SNMP authentication protocol (MD5, SHA).
SNMP Privacy Protocol	snmp_priv_protocol	The SNMP privacy protocol (DES, AES).
SNMP Privacy Protocol Password	snmp_priv_protocol_passwd	The SNMP privacy protocol password.
SNMP Privacy Protocol Script	snmp_priv_protocol_passwd_script	The script that supplies a password for SNMP privacy protocol. Using this supersedes the SNMP privacy protocol password parameter.
Power wait (seconds)	power_wait	Number of seconds to wait after issuing a power off or power on command.
Port (Outlet) Number	port	Physical plug number or name of virtual machine. This parameter is always required.

[Figure 4.7, “Eaton Network Power Switch”](#) shows the configuration screen for adding an Eaton Network Power Switch fence device.

Add Fence Device (Instance)

Fence Type	Eaton Network Power Switch (SNMP interface)
Name	
IP Address or Hostname	
UDP/TCP Port (optional, defaults to 161)	
Login	
Password	
Password Script (optional)	
SNMP Version	Default v
SNMP Community	
SNMP Security Level	Default v
SNMP Authentication Protocol	Default v
SNMP Privacy Protocol	Default v
SNMP Privacy Protocol Password	
SNMP Privacy Protocol Script	
Power Wait (seconds)	

Figure 4.7. Eaton Network Power Switch

The following command creates a fence device instance for an Eaton Network Power Switch device:

```
ccs -f cluster.conf --addfencedev eatontest agent=fence_eaton_snmp
ipaddr=192.168.0.1 login=root \
passwd=password123 power_wait=60 snmp_priv_passwd=eatonpassword123 udpport=161
```

The following is the **cluster.conf** entry for the **fence_eaton_snmp** device:

```
<fencedevices>
  <fencedevice agent="fence_eaton_snmp" community="private" ipaddr="eatonhost"
login="eatonlogin" \
  name="eatontest" passwd="password123" passwd_script="eatonpwscr"
power_wait="3333" \
  snmp_priv_passwd="eatonprivprotpass"
snmp_priv_passwd_script="eatonprivprotpwscr" udpport="161"/>
</fencedevices>
```

4.8. Egenera BladeFrame

[Table 4.9, “Egenera SAN Controller”](#) lists the fence device parameters used by **fence_egenera**, the fence agent for the Egenera BladeFrame.

Table 4.9. Egenera SAN Controller

luci Field	cluster.conf Attribute	Description
Name	name	A name for the Egenera BladeFrame device connected to the cluster.
CServer	cserver	The hostname (and optionally the username in the form of username@hostname) assigned to the device. Refer to the <code>fence_egenera(8)</code> man page for more information.
ESH Path (optional)	esh	The path to the esh command on the cserver (default is <code>/opt/panmgr/bin/esh</code>)
Username	user	The login name. The default value is root .
lpan	lpan	The logical process area network (LPAN) of the device.
pserver	pserver	The processing blade (pserver) name of the device.

[Figure 4.8, “Egenera BladeFrame”](#) shows the configuration screen for adding an Egenera BladeFrame fence device.

Add Fence Device (Instance)

Egenera SAN Controller

Fence TypeEgenera SAN Controller

Name

CServer

ESH Path (optional)

Username

Figure 4.8. Egenera BladeFrame

The following command creates a fence device instance for an Egenera BladeFrame device:

```
ccs -f cluster.conf --addfencedev egeneratest agent=fence_egenera user=root
cserver=cservertest
```

The following is the **cluster.conf** entry for the **fence_egenera** device:

```
<fencedevices>
  <fencedevice agent="fence_egenera" cserver="cservertest" name="egeneratest"
user="root"/>
</fencedevices>
```

4.9. ePowerSwitch

[Table 4.10, “ePowerSwitch”](#) lists the fence device parameters used by **fence_eps**, the fence agent for ePowerSwitch.

Table 4.10. ePowerSwitch

luci Field	cluster.conf Attribute	Description
Name	name	A name for the ePowerSwitch device connected to the cluster.
IP Address or Hostname	ipaddr	The IP address or hostname assigned to the device.
Login	login	The login name used to access the device.
Password	passwd	The password used to authenticate the connection to the device.
Password Script (optional)	passwd_script	The script that supplies a password for access to the fence device. Using this supersedes the Password parameter.
Name of Hidden Page	hidden_page	The name of the hidden page for the device.
Port (Outlet) Number	port	Physical plug number or name of virtual machine.

[Figure 4.9, “ePowerSwitch”](#) shows the configuration screen for adding an ePowerSwitch fence device.

Add Fence Device (Instance)

ePowerSwitch	
Fence Type	ePowerSwitch
Name	
IP Address or Hostname	
Login	
Password	
Password Script (optional)	
Name of Hidden Page	

Figure 4.9. ePowerSwitch

The following command creates a fence device instance for an ePowerSwitch device:

```
ccs -f cluster.conf --addfencedev epstest1 agent=fence_eps ipaddr=192.168.0.1
login=root passwd=password123 \
hidden_page=hidden.htm
```

The following is the **cluster.conf** entry for the **fence_eps** device:

```
<fencedevices>
  <fencedevice agent="fence_eps" hidden_page="hidden.htm" ipaddr="192.168.0.1"
login="root" name="epstest1" \
  passwd="password123"/>
</fencedevices>
```

4.10. Fence Virt

[Table 4.11, “Fence virt”](#) lists the fence device parameters used by **fence_virt**, the fence agent for a Fence virt fence device.

Table 4.11. Fence virt

luci Field	cluster.conf Attribute	Description
Name	name	A name for the Fence virt fence device.
Serial Device	serial_device	On the host, the serial device must be mapped in each domain's configuration file. For more information, see the fence_virt.conf man page. If this field is specified, it causes the fence_virt fencing agent to operate in serial mode. Not specifying a value causes the fence_virt fencing agent to operate in VM channel mode.
Serial Parameters	serial_params	The serial parameters. The default is 115200, 8N1.
VM Channel IP Address	channel_address	The channel IP. The default value is 10.0.2.179.
Port or Domain (deprecated)	port	Virtual machine (domain UUID or name) to fence.
	ipport	The channel port. The default value is 1229, which is the value used when configuring this fence device with luci .

[Figure 4.10, “Fence Virt”](#) shows the configuration screen for adding a Fence Virt device.

Add Fence Device (Instance)

Fence virt (Multicast Mode) ⌵

Fence Type

Fence xvm

Name

Figure 4.10. Fence Virt

The following command creates a fence device instance for the Fence Virt device:

```
ccs -f cluster.conf --addfencedev fencevirt1 agent=fence_virt
serial_device=/dev/ttyS1 serial_params=19200, 8N1
```

The following is the **cluster.conf** entry for the **fence_virt** device:

```
<fencedevices>
  <fencedevice agent="fence_virt" name="fencevirt1" serial_device="/dev/ttyS1"
serial_params="19200, 8N1"/>
</fencedevices>
```

4.11. Fujitsu-Siemens RemoteView Service Board (RSB)

Table 4.12, “Fujitsu Siemens RemoteView Service Board (RSB)” lists the fence device parameters used by **fence_rsb**, the fence agent for Fujitsu-Siemens RemoteView Service Board (RSB).

Table 4.12. Fujitsu Siemens RemoteView Service Board (RSB)

luci Field	cluster.conf Attribute	Description
Name	name	A name for the RSB to use as a fence device.
IP Address or Hostname	ipaddr	The hostname assigned to the device.
Login	login	The login name used to access the device.
Password	passwd	The password used to authenticate the connection to the device.
Password Script (optional)	passwd_script	The script that supplies a password for access to the fence device. Using this supersedes the Password parameter.
TCP Port	ipport	The port number on which the telnet service listens. The default value is 3172.

Figure 4.11, “Fujitsu-Siemens RSB” shows the configuration screen for adding an Fujitsu-Siemens RSB fence device.

Add Fence Device (Instance)

Fujitsu Siemens RemoteView Service Board
⌵

Fence Type	Fujitsu Siemens RemoteView Service Board (RSB)
Name	
IP Address or Hostname	
Login	
Password	
Password Script (optional)	
TCP Port	

Figure 4.11. Fujitsu-Siemens RSB

The following command creates a fence device instance for a Fujitsu-Siemens RSB device:

```
ccs -f cluster.conf --addfencedev fsrbtest1 agent=fence_rsb ipaddr=192.168.0.1
login=root passwd=password123 \
telnet_port=3172
```

The following is the **cluster.conf** entry for the **fence_rsb** device:

```
<fencedevices>
  <fencedevice agent="fence_rsb" ipaddr="192.168.0.1" login="root"
name="fsrsbtest1" passwd="password123" telnet_port="3172"/>
</fencedevices>
```

4.12. Hewlett-Packard BladeSystem

[Table 4.13, “HP BladeSystem \(Red Hat Enterprise Linux 6.4 and later\)”](#) lists the fence device parameters used by **fence_hpbld**, the fence agent for HP BladeSystem.

Table 4.13. HP BladeSystem (Red Hat Enterprise Linux 6.4 and later)

luci Field	cluster.conf Attribute	Description
Name	name	The name assigned to the HP Bladesystem device connected to the cluster.
IP Address or Hostname	ipaddr	The IP address or hostname assigned to the HP BladeSystem device.
IP Port (optional)	ipport	The TCP port to use to connect to the device.
Login	login	The login name used to access the HP BladeSystem device. This parameter is required.
Password	passwd	The password used to authenticate the connection to the fence device.
Password Script (optional)	passwd_script	The script that supplies a password for access to the fence device. Using this supersedes the Password parameter.
Force Command Prompt	cmd_prompt	The command prompt to use. The default value is '/\$'.
Missing port returns OFF instead of failure	missing_as_off	Missing port returns OFF instead of failure.
Power Wait (seconds)	power_wait	Number of seconds to wait after issuing a power off or power on command.
Use SSH	secure	Indicates that the system will use SSH to access the device.
Path to SSH Identity File	identity_file	The identity file for SSH.

[Figure 4.12, “HP BladeSystem”](#) shows the configuration screen for adding an HP BladeSystem fence device.

Add Fence Device (Instance)

Fence Type	HP BladeSystem
Name	
IP Address or Hostname	
IP Port (optional)	
Login	
Password	
Password Script (optional)	
Force Command Prompt	
Missing port returns OFF instead of failure	☐
Power Wait (seconds)	

Figure 4.12. HP BladeSystem

The following command creates a fence device instance for a BladeSystem device:

```
ccs -f cluster.conf --addfencedev hpbladetest1 agent=fence_hpblade
cmd_prompt=c7000oa> ipaddr=192.168.0.1 \
login=root passwd=password123 missing_as_off=on power_wait=60
```

The following is the **cluster.conf** entry for the **fence_hpblade** device:

```
<fencedevices>
  <fencedevice agent="fence_hpblade" cmd_prompt="c7000oa>" ipaddr="hpbladeaddr"
ipport="13456" \
  login="root" missing_as_off="on" name="hpbladetest1" passwd="password123"
passwd_script="hpbladepwscr" \
  power_wait="60"/>
</fencedevices>
```

4.13. Hewlett-Packard iLO

[Table 4.14, "HP iLO/iLO2 \(Integrated Lights Out\)"](#) lists the fence device parameters used by **fence_ilo**, the fence agent for HP iLO devices.

Table 4.14. HP iLO/iLO2 (Integrated Lights Out)

luci Field	cluster.conf Attribute	Description
Name	name	A name for the server with HP iLO support.
IP Address or Hostname	ipaddr	The IP address or hostname assigned to the device.
IP Port (optional)	ipport	TCP port to use for connection with the device.
Login	login	The login name used to access the device.
Password	passwd	The password used to authenticate the connection to the device.
Password Script (optional)	passwd_script	The script that supplies a password for access to the fence device. Using this supersedes the Password parameter.
Power Wait	power_wait	Number of seconds to wait after issuing a power off or power on command.

Figure 4.13, “HP iLO” shows the configuration screen for adding an HP iLO fence device.

Add Fence Device (Instance)

HP iLO Device

Fence Type	HP iLO / iLO2
Name	
IP Address or Hostname	
IP Port (optional)	
Login	
Password	
Password Script (optional)	
Power Wait (seconds)	

Figure 4.13. HP iLO

The following command creates a fence device instance for a HP iLO device:

```
ccs -f cluster.conf --addfencedev hpilotest1 agent=fence_hpilo ipaddr=192.168.0.1
login=root passwd=password123 \
power_wait=60
```

The following is the **cluster.conf** entry for the **fence_ilo** device:

```
<fencedevices>
  <fencedevice agent="fence_ilo" ipaddr="192.168.0.1" login="root"
name="hpilotest1" passwd="password123" \
  power_wait="60"/>
</fencedevices>
```

4.14. Hewlett-Packard iLO MP

[Table 4.15, “HP iLO \(Integrated Lights Out\) MP”](#) lists the fence device parameters used by **fence_ilo_mp**, the fence agent for HP iLO MP devices.

Table 4.15. HP iLO (Integrated Lights Out) MP

luci Field	cluster.conf Attribute	Description
Name	name	A name for the server with HP iLO support.
IP Address or Hostname	ipaddr	The IP address or hostname assigned to the device.
IP Port (optional)	ipport	TCP port to use for connection with the device.
Login	login	The login name used to access the device.
Password	passwd	The password used to authenticate the connection to the device.
Password Script (optional)	passwd_script	The script that supplies a password for access to the fence device. Using this supersedes the Password parameter.
Use SSH	secure	Indicates that the system will use SSH to access the device.
Path to SSH Identity File	identity_file	The Identity file for SSH.
Force Command Prompt	cmd_prompt	The command prompt to use. The default value is MP>, hpiLO->.
Power Wait	power_wait	Number of seconds to wait after issuing a power off or power on command.

[Figure 4.14, “HP iLO MP”](#) shows the configuration screen for adding an HP iLO MPfence device.

Add Fence Device (Instance)

Fence Type	HP iLO MP
Name	
IP Address or Hostname	
IP Port (optional)	
Login	
Password	
Password Script (optional)	
SSH	☐ Use SSH
Path to SSH Identity File	
Force Command Prompt	
Power Wait (seconds)	

Figure 4.14. HP iLO MP

The following command creates a fence device instance for a HP iLO MP device:

```
ccs -f cluster.conf --addfencedev hpilomptest1 agent=fence_hpilo
cmd_prompt=hpilo-> ipaddr=192.168.0.1 \
login=root passwd=password123 power_wait=60
```

The following is the **cluster.conf** entry for the **fence_hpilo_mp** device:

```
<fencedevices>
<fencedevice agent="fence_ilo_mp" cmd_prompt="hpilo->" ipaddr="192.168.0.1"
login="root" name="hpilomptest1" passwd="password123" power_wait="60"/>
</fencedevices>
```

4.15. IBM BladeCenter

[Table 4.16, “IBM BladeCenter”](#) lists the fence device parameters used by **fence_bladecenter**, the fence agent for IBM BladeCenter.

Table 4.16. IBM BladeCenter

luci Field	cluster.conf Attribute	Description
Name	name	A name for the IBM BladeCenter device connected to the cluster.
IP Address or Hostname	ipaddr	The IP address or hostname assigned to the device.
IP port (optional)	ipport	TCP port to use for connection with the device.
Login	login	The login name used to access the device.
Password	passwd	The password used to authenticate the connection to the device.
Password Script (optional)	passwd_script	The script that supplies a password for access to the fence device. Using this supersedes the Password parameter.
Power Wait	power_wait	Number of seconds to wait after issuing a power off or power on command.
Use SSH	secure	Indicates that system will use SSH to access the device.
Path to SSH Identity File	identity_file	The identity file for SSH.

[Figure 4.15, “IBM BladeCenter”](#) shows the configuration screen for adding an IBM BladeCenter fence device.

Add Fence Device (Instance)

IBM BladeCenter
⌵

Fence Type	IBM Blade Center
Name	
IP Address or Hostname	
IP Port (optional)	
Login	
Password	
Password Script (optional)	
Power Wait (seconds)	

Figure 4.15. IBM BladeCenter

The following command creates a fence device instance for an IBM BladeCenter device:

```
ccs -f cluster.conf --addfencedev bladecentertest1 agent=fence_bladecenter  
ipaddr=192.168.0.1 login=root \  
passwd=password123 power_wait=60
```

The following is the **cluster.conf** entry for the **fence_bladecenter** device:

```
<fencedevices>  
  <fencedevice agent="fence_bladecenter" ipaddr="192.168.0.1" login="root"  
name="bladecentertest1" passwd="password123" \  
  power_wait="60"/>  
</fencedevices>
```

4.16. IBM BladeCenter over SNMP

[Table 4.17, “IBM BladeCenter SNMP”](#) lists the fence device parameters used by **fence_ibmblade**, the fence agent for IBM BladeCenter over SNMP.

Table 4.17. IBM BladeCenter SNMP

luci Field	cluster.conf Attribute	Description
Name	name	A name for the IBM BladeCenter SNMP device connected to the cluster.
IP Address or Hostname	ipaddr	The IP address or hostname assigned to the device.
UDP/TCP Port (optional)	udpport	UDP/TCP port to use for connections with the device; the default value is 161.
Login	login	The login name used to access the device.
Password	passwd	The password used to authenticate the connection to the device.
Password Script (optional)	passwd_script	The script that supplies a password for access to the fence device. Using this supersedes the Password parameter.
SNMP Version	snmp_version	The SNMP version to use (1, 2c, 3); the default value is 1.
SNMP Community	community	The SNMP community string.
SNMP Security Level	snmp_sec_level	The SNMP security level (noAuthNoPriv, authNoPriv, authPriv).
SNMP Authentication Protocol	snmp_auth_protocol	The SNMP authentication protocol (MD5, SHA).
SNMP Privacy Protocol	snmp_priv_protocol	The SNMP privacy protocol (DES, AES).
SNMP privacy protocol password	snmp_priv_protocol_password	The SNMP Privacy Protocol Password.
SNMP Privacy Protocol Script	snmp_priv_protocol_password_script	The script that supplies a password for SNMP privacy protocol. Using this supersedes the SNMP privacy protocol password parameter.
Power Wait	power_wait	Number of seconds to wait after issuing a power off or power on command.
Port	port	Physical plug number or name of virtual machine.

[Figure 4.16, “IBM BladeCenter SNMP”](#) shows the configuration screen for adding an IBM BladeCenter SNMP fence device.

Add Fence Device (Instance)

Fence Type	IBM BladeCenter SNMP
Name	
IP Address or Hostname	
UDP/TCP Port (optional, defaults to 161)	
Login	
Password	
Password Script (optional)	
SNMP Version	Default v
SNMP Community	
SNMP Security Level	Default v
SNMP Authentication Protocol	Default v
SNMP Privacy Protocol	Default v
SNMP Privacy Protocol Password	
SNMP Privacy Protocol Script	
Power Wait (seconds)	

Figure 4.16. IBM BladeCenter SNMP

The following command creates a fence device instance for an IBM BladeCenter SNMP device:

```
ccs -f cluster.conf --addfencedev bladesnmp1 agent=fence_ibmblade
community=private ipaddr=192.168.0.1 login=root \
passwd=password123 snmp_priv_passwd=snmpasswd123 power_wait=60
```

The following is the **cluster.conf** entry for the **fence_ibmblade** device:

```
<fencedevices>
  <fencedevice agent="fence_ibmblade" community="private" ipaddr="192.168.0.1"
login="root" name="bladesnmp1" \
  passwd="password123" power_wait="60" snmp_priv_passwd="snmpasswd123"
udpport="161"/>
</fencedevices>
```

4.17. IBM iPDU

[Table 4.18, “IBM iPDU \(Red Hat Enterprise Linux 6.4 and later\)”](#) lists the fence device parameters used by **fence_ipdu**, the fence agent for iPDU over SNMP devices.

Table 4.18. IBM iPDU (Red Hat Enterprise Linux 6.4 and later)

luci Field	cluster.conf Attribute	Description
Name	name	A name for the IBM iPDU device connected to the cluster into which the fence daemon logs via the SNMP protocol.
IP Address or Hostname	ipaddr	The IP address or hostname assigned to the device.
UDP/TCP Port	udpport	The UDP/TCP port to use for connection with the device; the default value is 161.
Login	login	The login name used to access the device.
Password	passwd	The password used to authenticate the connection to the device.
Password Script (optional)	passwd_script	The script that supplies a password for access to the fence device. Using this supersedes the Password parameter.
SNMP Version	snmp_version	The SNMP version to use (1, 2c, 3); the default value is 1.
SNMP Community	community	The SNMP community string; the default value is private .
SNMP Security Level	snmp_sec_level	The SNMP security level (noAuthNoPriv, authNoPriv, authPriv).
SNMP Authentication Protocol	snmp_auth_prot	The SNMP Authentication Protocol (MD5, SHA).
SNMP Privacy Protocol	snmp_priv_prot	The SNMP privacy protocol (DES, AES).
SNMP Privacy Protocol Password	snmp_priv_password	The SNMP privacy protocol password.
SNMP Privacy Protocol Script	snmp_priv_password_script	The script that supplies a password for SNMP privacy protocol. Using this supersedes the SNMP privacy protocol password parameter.
Power Wait	power_wait	Number of seconds to wait after issuing a power off or power on command.
Port	port	The port.

[Figure 4.17, “IBM iPDU”](#) shows the configuration screen for adding an IBM iPDU fence device.

Add Fence Device (Instance)

Fence Type	IBM BladeCenter SNMP
Name	
IP Address or Hostname	
UDP/TCP Port (optional, defaults to 161)	
Login	
Password	
Password Script (optional)	
SNMP Version	Default v
SNMP Community	
SNMP Security Level	Default v
SNMP Authentication Protocol	Default v
SNMP Privacy Protocol	Default v
SNMP Privacy Protocol Password	
SNMP Privacy Protocol Script	
Power Wait (seconds)	

Figure 4.17. IBM iPDU

The following command creates a fence device instance for an IBM iPDU device:

```
ccs -f cluster.conf --addfencedev ipdutest1 agent=fence_ipdu community=ipdusnmpcom
ipaddr=192.168.0.1 login=root \
  passwd=password123 snmp_priv_passwd=snmpasswd123 power_wait=60 snmp_priv_prot=AES
udpport=111
```

The following is the **cluster.conf** entry for the **fence_ipdu** device:

```
<fencedevices>
  <fencedevice agent="fence_ipdu" community="ipdusnmpcom" ipaddr="ipduhost"
login="root" name="ipdutest1" \
  passwd="password123" power_wait="60" snmp_priv_passwd="ipduprivprotpasswd"
snmp_priv_prot="AES" \
  udpport="111"/>
</fencedevices>
```

4.18. IF-MIB

[Table 4.19, “IF MIB”](#) lists the fence device parameters used by **fence_ifmib**, the fence agent for IF-MIB devices.

Table 4.19. IF MIB

luci Field	cluster.conf Attribute	Description
Name	name	A name for the IF MIB device connected to the cluster.
IP Address or Hostname	ipaddr	The IP address or hostname assigned to the device.
UDP/TCP Port (optional)	udpport	The UDP/TCP port to use for connection with the device; the default value is 161.
Login	login	The login name used to access the device.
Password	passwd	The password used to authenticate the connection to the device.
Password Script (optional)	passwd_script	The script that supplies a password for access to the fence device. Using this supersedes the Password parameter.
SNMP Version	snmp_version	The SNMP version to use (1, 2c, 3); the default value is 1.
SNMP Community	community	The SNMP community string.
SNMP Security Level	snmp_sec_level	The SNMP security level (noAuthNoPriv, authNoPriv, authPriv).
SNMP Authentication Protocol	snmp_auth_protocol	The SNMP authentication protocol (MD5, SHA).
SNMP Privacy Protocol	snmp_priv_protocol	The SNMP privacy protocol (DES, AES).
SNMP Privacy Protocol Password	snmp_priv_protocol_passwd	The SNMP privacy protocol password.
SNMP Privacy Protocol Script	snmp_priv_protocol_passwd_script	The script that supplies a password for SNMP privacy protocol. Using this supersedes the SNMP privacy protocol password parameter.
Power Wait	power_wait	Number of seconds to wait after issuing a power off or power on command.
Port	port	Physical plug number or name of virtual machine.

[Figure 4.18, “IF-MIB”](#) shows the configuration screen for adding an IF-MIB fence device.

Add Fence Device (Instance)

Fence Type	IF MIB
Name	
IP Address or Hostname	
UDP/TCP Port (optional, defaults to 161)	
Login	
Password	
Password Script (optional)	
SNMP Version	Default v
SNMP Community	
SNMP Security Level	Default v
SNMP Authentication Protocol	Default v
SNMP Privacy Protocol	Default v
SNMP Privacy Protocol Password	
SNMP Privacy Protocol Script	
Power Wait (seconds)	

Figure 4.18. IF-MIB

The following command creates a fence device instance for an IF-MIB device:

```
ccs -f cluster.conf --addfencedev ifmib1 agent=fence_ifmib community=private
ipaddr=192.168.0.1 login=root \
passwd=password123 snmp_priv_passwd=snmpasswd123 power_wait=60 udpport=161
```

The following is the **cluster.conf** entry for the **fence_ifmib** device:

```
<fencedevices>
  <fencedevice agent="fence_ifmib" community="private" ipaddr="192.168.0.1"
login="root" name="ifmib1" \
  passwd="password123" power_wait="60" snmp_priv_passwd="snmpasswd123"
udpport="161"/>
</fencedevices>
```

4.19. Intel Modular

[Table 4.20, “Intel Modular”](#) lists the fence device parameters used by `fence_intelmodular`, the fence agent for Intel Modular.

Table 4.20. Intel Modular

luci Field	cluster.conf Attribute	Description
Name	name	A name for the Intel Modular device connected to the cluster.
IP Address or Hostname	ipaddr	The IP address or hostname assigned to the device.
Login	login	The login name used to access the device.
Password	passwd	The password used to authenticate the connection to the device.
Password Script (optional)	passwd_script	The script that supplies a password for access to the fence device. Using this supersedes the Password parameter.
SNMP Version	snmp_version	The SNMP version to use (1, 2c, 3); the default value is 1.
SNMP Community	community	The SNMP community string; the default value is private .
SNMP Security Level	snmp_sec_level	The SNMP security level (noAuthNoPriv, authNoPriv, authPriv).
SNMP Authentication Protocol	snmp_auth_protocol	The SNMP authentication protocol (MD5, SHA).
SNMP Privacy Protocol	snmp_priv_protocol	The SNMP privacy protocol (DES, AES).
SNMP Privacy Protocol Password	snmp_priv_protocol_password	The SNMP privacy protocol password.
SNMP Privacy Protocol Script	snmp_priv_protocol_password_script	The script that supplies a password for SNMP privacy protocol. Using this supersedes the SNMP privacy protocol password parameter.
Power Wait	power_wait	Number of seconds to wait after issuing a power off or power on command.
Port	port	Physical plug number or name of virtual machine.

[Figure 4.19, “Intel Modular”](#) shows the configuration screen for adding an Intel Modular fence device.

Add Fence Device (Instance)

Fence Type	Intel Modular
Name	
IP Address or Hostname	
UDP/TCP Port (optional, defaults to 161)	
Login	
Password	
Password Script (optional)	
SNMP Version	Default v
SNMP Community	
SNMP Security Level	Default v
SNMP Authentication Protocol	Default v
SNMP Privacy Protocol	Default v
SNMP Privacy Protocol Password	
SNMP Privacy Protocol Script	
Power Wait (seconds)	

Figure 4.19. Intel Modular

The following command creates a fence device instance for an Intel Modular device:

```
ccs -f cluster.conf --addfencedev intelmodular1 agent=fence_intelmodular
community=private ipaddr=192.168.0.1 login=root \
passwd=password123 snmp_priv_passwd=snmpasswd123 power_wait=60 udpport=161
```

The following is the **cluster.conf** entry for the **fence_intelmodular** device:

```
<fencedevices>
  <fencedevice agent="fence_intelmodular" community="private"
ipaddr="192.168.0.1" login="root" name="intelmodular1" \
  passwd="password123" power_wait="60" snmp_priv_passwd="snmpasswd123"
udpport="161"/>
</fencedevices>
```

4.20. IPMI over LAN

[Table 4.21, “IPMI \(Intelligent Platform Management Interface\) LAN, Dell iDrac, IBM Integrated Management Module, HP iLO3, HP iLO4”](#) lists the fence device parameters used by **fence_ipmilan**, the fence agent for IPMI over LAN. *Note:* The fence agents for IPMI over LAN (**fence_ipmilan**), Dell iDRAC (**fence_idrac**), IBM Integrated Management Module (**fence_imm**), HP iLO3 devices (**fence_ilo3**), and HP iLO4 devices (**fence_ilo4**) share the same implementation, so they share the same configuration as shown in [Table 4.21, “IPMI \(Intelligent Platform Management Interface\) LAN, Dell iDrac, IBM Integrated Management Module, HP iLO3, HP iLO4”](#).

Table 4.21. IPMI (Intelligent Platform Management Interface) LAN, Dell iDrac, IBM Integrated Management Module, HP iLO3, HP iLO4

luci Field	cluster.conf Attribute	Description
Name	name	A name for the IPMI LAN device connected to the cluster.
IP Address or Hostname	ipaddr	The IP address or hostname assigned to the device.
Login	login	The login name of a user capable of issuing power on/off commands to the given IPMI port.
Password	passwd	The password used to authenticate the connection to the IPMI port.
Password Script (optional)	passwd_script	The script that supplies a password for access to the fence device. Using this supersedes the Password parameter.
Authentication Type	auth	IPMI LAN authentication type: none , password , or md5 .
Use Lanplus	lanplus	True or 1 . If blank, then value is False . It is recommended that you enable Lanplus to improve the security of your connection if your hardware supports it.
Ciphersuite to use	cipher	The remote server authentication, integrity, and encryption algorithms to use for IPMIv2 lanplus connections.
Privilege level	privlvl	The privilege level on the IPMI device.
IPMI Operation Timeout	timeout	Timeout in seconds for IPMI operation.
Power Wait (seconds)	power_wait	Number of seconds to wait after issuing a power off or power on command.
Delay (optional)	delay	The number of seconds to wait before fencing is started. The default value is 0.

[Figure 4.20, “IPMI over LAN”](#) shows the configuration screen for adding an IPMI over LAN device

Add Fence Device (Instance)

IPMI Lan

Fence Type	IPMI Lan
Name	
IP Address or Hostname	
Login	
Password	
Password Script (optional)	
Authentication Type	None
Use Lanplus	☐
Ciphersuite to use	
Privilege Level	Default

Figure 4.20. IPMI over LAN

The following command creates a fence device instance for an IPMI over LAN device:

```
ccs -f cluster.conf --addfencedev ipmitest1 agent=fence_ipmilan auth=password
cipher=3 ipaddr=192.168.0.1 \
lanplus=on login=root passwd=password123
```

The following is the **cluster.conf** entry for the **fence_ipmilan** device:

```
<fencedevices>
  <fencedevice agent="fence_ipmilan" auth="password" cipher="3"
ipaddr="192.168.0.1" lanplus="on" login="root" \
  name="ipmitest1" passwd="password123"/>
</fencedevices>
```

4.21. RHEV-M REST API

[Table 4.22 “RHEV-M REST API \(RHEL 6.2 and later against RHEV 3.0 and later\)”](#) lists the fence device parameters used by **fence_rhevm**, the fence agent for RHEV-M REST API.

Table 4.22. RHEV-M REST API (RHEL 6.2 and later against RHEV 3.0 and later)

luci Field	cluster.conf Attribute	Description
Name	name	Name of the RHEV-M REST API fencing device.
IP Address or Hostname	ipaddr	The IP address or hostname assigned to the device.
IP Port (optional)	ipport	The TCP port to use for connection with the device.
Login	login	The login name used to access the device.
Password	passwd	The password used to authenticate the connection to the device.
Password Script (optional)	passwd_script	The script that supplies a password for access to the fence device. Using this supersedes the Password parameter.
Use SSL	ssl	Use SSL connections to communicate with the device.
Power Wait	power_wait	Number of seconds to wait after issuing a power off or power on command.
Port	port	Physical plug number or name of virtual machine.

Figure 4.21, “RHEV-M REST API” shows the configuration screen for adding an RHEV-M REST API device

Add Fence Device (Instance)

RHEV-M fencing

Fence Type	RHEV-M fencing
Name	
IP Address or Hostname	
IP Port (optional)	
Login	
Password	
Password Script (optional)	
Use SSL	☐
Power Wait (seconds)	

Figure 4.21. RHEV-M REST API

The following command creates a fence device instance for an RHEV-M REST API device:

```
ccs -f cluster.conf --addfencedev rhevmtest1 agent=fence_rhevm ipaddr=192.168.0.1
login=root passwd=password123 \
power_wait=60 ssl=on
```

The following is the **cluster.conf** entry for the **fence_rhevm** device:

```
<fencedevices>
  <fencedevice agent="fence_rhevm" ipaddr="192.168.0.1" login="root"
name="rhevmtest1" passwd="password123" \
  power_wait="60" ssl="on"/>
</fencedevices>
```

4.22. SCSI Persistent Reservations

[Table 4.23, “SCSI Fencing”](#) lists the fence device parameters used by **fence_scsi**, the fence agent for SCSI persistent reservations.



Note

Use of SCSI persistent reservations as a fence method is supported with the following limitations:

- » When using SCSI fencing, all nodes in the cluster must register with the same devices so that each node can remove another node's registration key from all the devices it is registered with.
- » Devices used for the cluster volumes should be a complete LUN, not partitions. SCSI persistent reservations work on an entire LUN, meaning that access is controlled to each LUN, not individual partitions.

Table 4.23. SCSI Fencing

luci Field	cluster.conf Attribute	Description
Name	name	A name for the SCSI fence device.
Node name		
Key for current action		(overrides node name)

[Figure 4.22, “SCSI Fencing”](#) shows the configuration screen for adding an SCSI fence device

Add Fence Device (Instance)



SCSI Reservation Fencing

Fence Type SCSI Reservation Fencing

Name

Figure 4.22. SCSI Fencing

The following command creates a fence device instance for a SCSI Fence device:

```
ccs -f cluster.conf --addfencedev scsifencetest1 agent=fence_scsi
```

The following is the **cluster.conf** entry for the **fence_scsi** device:

```
<fencedevices>
  <<fencedevice agent="fence_scsi" name="scsifencetest1"/>
</fencedevices>
```

4.23. VMWare over SOAP API

[Table 4.24, “VMware Fencing \(SOAP Interface\) \(Red Hat Enterprise Linux 6.2 and later\)”](#) lists the fence device parameters used by **fence_vmware_soap**, the fence agent for VMWare over SOAP API.

Table 4.24. VMware Fencing (SOAP Interface) (Red Hat Enterprise Linux 6.2 and later)

luci Field	cluster.conf Attribute	Description
Name	name	Name of the virtual machine fencing device.
IP Address or Hostname	ipaddr	The IP address or hostname assigned to the device.
IP Port (optional)	ipport	The TCP port to use for connection with the device.
Login	login	The login name used to access the device.
Password	passwd	The password used to authenticate the connection to the device.
Password Script (optional)	passwd_script	The script that supplies a password for access to the fence device. Using this supersedes the Password parameter.
Separator	separator	Separator for CSV created by operation list. The default value is a comma(,).
Power Wait	power_wait	Number of seconds to wait after issuing a power off or power on command.
VM name	port	Name of virtual machine in inventory path format (e.g., /datacenter/vm/Discovered_virtual_machine/myMachine).
VM UUID	uuid	The UUID of the virtual machine to fence.
Use SSL	ssl	Use SSL connections to communicate with the device.

Figure 4.23, “VMWare over SOAP Fencing” shows the configuration screen for adding a VMWare over SOAP fence device

Add Fence Device (Instance)

VMware Fencing (SOAP Interface) 

Fence Type

VMware (SOAP Interface)

Name

IP Address or Hostname

IP Port (optional)

Login

Password

Password Script (optional)

Separator

Power Wait (seconds)

Figure 4.23. VMWare over SOAP Fencing

The following command creates a fence device instance for a VMWare over SOAP fence device:

```
ccs -f cluster.conf --addfencedev vmwaresoaptest1 agent=fence_vmware_soap
login=root passwd=password123 power_wait=60 \
separator=,
```

The following is the **cluster.conf** entry for the **fence_vmware_soap** device:

```
<fencedevices>
  <fencedevice agent="fence_vmware_soap" ipaddr="192.168.0.1" login="root"
name="vmwaresoaptest1" passwd="password123" \
  power_wait="60" separator="."/>
</fencedevices>
```

4.24. WTI Power Switch

[Table 4.25, “WTI Power Switch”](#) lists the fence device parameters used by **fence_wti**, the fence agent for the WTI network power switch.

Table 4.25. WTI Power Switch

luci Field	cluster.conf Attribute	Description
Name	name	A name for the WTI power switch connected to the cluster.
IP Address or Hostname	ipaddr	The IP or hostname address assigned to the device.
IP Port (optional)	ipport	The TCP port to use to connect to the device.
Login	login	The login name used to access the device.
Password	passwd	The password used to authenticate the connection to the device.
Password Script (optional)	passwd_script	The script that supplies a password for access to the fence device. Using this supersedes the Password parameter.
Port	port	Physical plug number or name of virtual machine.
Force command prompt	cmd_prompt	The command prompt to use. The default value is ['RSM>', '>MPC', 'IPS>', 'TPS>', 'NBB>', 'NPS>', 'VMR>']
Power Wait	power_wait	Number of seconds to wait after issuing a power off or power on command.
Use SSH	secure	Indicates that system will use SSH to access the device.
Path to SSH Identity File	identity_file	The identity file for SSH.

[Figure 4.24, “WTI Fencing”](#) shows the configuration screen for adding a WTI fence device

Add Fence Device (Instance)

WTI Power Switch

⌵

Fence Type	WTI Power Switch
Name	
IP Address or Hostname	
IP Port (optional)	
Login	
Password	
Password Script (optional)	
Force Command Prompt	
Power Wait (seconds)	

Figure 4.24. WTI Fencing

The following command creates a fence device instance for a WTI fence device:

```
ccs -f cluster.conf --addfencedev wtipwrs1 agent=fence_wti cmd_prompt=VMR>
login=root passwd=password123 \
power_wait=60
```

The following is the **cluster.conf** entry for the **fence_wti** device:

```
<fencedevices>
  <fencedevice agent="fence_wti" cmd_prompt="VMR>" ipaddr="192.168.0.1"
login="root" name="wtipwrs1" \
  passwd="password123" power_wait="60"/>
</fencedevices>
```

Revision History

Revision 1-9.404 Rebuild with Publican 4.0.0	Mon Nov 25 2013	Rüdiger Landmann
Revision 1-9 Release for GA of Red Hat Enterprise Linux 6.5	Wed Nov 20 2013	John Ha
Revision 1-4 Release for Beta of Red Hat Enterprise Linux 6.5	Mon Nov 28 2012	John Ha
Revision 1-2 Release for Beta of Red Hat Enterprise Linux 6.4	Mon Nov 28 2012	John Ha

Index

A

ACPI

- configuring, [Configuring ACPI For Use with Integrated Fence Devices](#)

APC power switch over SNMP fence device , [APC Power Switch over Telnet and SSH](#)

APC power switch over telnet/SSH fence device , [APC Power Switch over Telnet and SSH](#)

B

Brocade fabric switch fence device , [Brocade Fabric Switch](#)

C

CISCO MDS fence device , [Cisco MDS](#)

Cisco UCS fence device , [Cisco UCS](#)

cluster administration

- configuring ACPI, [Configuring ACPI For Use with Integrated Fence Devices](#)

D

Dell DRAC 5 fence device , [Dell Drac 5](#)

E

Eaton network power switch, [Eaton Network Power Switch](#)

Egenera SAN controller fence device , [Egenera BladeFrame](#)

ePowerSwitch fence device , [ePowerSwitch](#)

F

feedback

- contact information for this manual, [We Need Feedback!](#)

fence

- configuration, [Fencing Pre-Configuration](#)
- devices, [Fence Devices](#)

fence agent

- fence_apc, [APC Power Switch over Telnet and SSH](#)
- fence_apc_snmp, [APC Power Switch over Telnet and SSH](#)
- fence_bladecenter, [IBM BladeCenter](#)
- fence_brocade, [Brocade Fabric Switch](#)
- fence_cisco_mds, [Cisco MDS](#)
- fence_cisco_ucs, [Cisco UCS](#)
- fence_drac5, [Dell Drac 5](#)
- fence_eaton_snmp, [Eaton Network Power Switch](#)
- fence_egenera, [Egenera BladeFrame](#)
- fence_eps, [ePowerSwitch](#)
- fence_hpblade, [Hewlett-Packard BladeSystem](#)
- fence_ibmblade, [IBM BladeCenter over SNMP](#)
- fence_ifmib, [IF-MIB](#)
- fence_ilo, [Hewlett-Packard iLO](#)
- fence_ilo_mp, [Hewlett-Packard iLO MP](#)
- fence_intelmodular, [Intel Modular](#)
- fence_ipdu, [IBM iPDU](#)
- fence_ipmilan, [IPMI over LAN](#)
- fence_rhevm, [RHEV-M REST API](#)
- fence_rsb, [Fujitsu-Siemens RemoteView Service Board \(RSB\)](#)
- fence_scsi, [SCSI Persistent Reservations](#)
- fence_virt, [Fence Virt](#)
- fence_vmware_soap, [VMWare over SOAP API](#)
- fence_wti, [WTI Power Switch](#)

fence configuration, [Fencing Pre-Configuration](#), [Configuring Fencing with Conga](#)

fence device

- APC power switch over SNMP, [APC Power Switch over Telnet and SSH](#)
- APC power switch over telnet/SSH, [APC Power Switch over Telnet and SSH](#)
- Brocade fabric switch, [Brocade Fabric Switch](#)
- Cisco MDS, [Cisco MDS](#)
- Cisco UCS, [Cisco UCS](#)
- Dell DRAC 5, [Dell Drac 5](#)
- Eaton network power switch, [Eaton Network Power Switch](#)
- Egenera SAN controller, [Egenera BladeFrame](#)
- ePowerSwitch, [ePowerSwitch](#)
- Fence virt, [Fence Virt](#)
- Fujitsu Siemens RemoteView Service Board (RSB), [Fujitsu-Siemens RemoteView Service Board \(RSB\)](#)
- HP BladeSystem, [Hewlett-Packard BladeSystem](#)
- HP iLO MP, [Hewlett-Packard iLO MP](#)
- HP iLO/iLO2, [Hewlett-Packard iLO](#)
- IBM BladeCenter, [IBM BladeCenter](#)

- IBM BladeCenter SNMP, [IBM BladeCenter over SNMP](#)
- IBM iPDU, [IBM iPDU](#)
- IF MIB, [IF-MIB](#)
- Intel Modular, [Intel Modular](#)
- IPMI LAN, [IPMI over LAN](#)
- RHEV-M REST API, [RHEV-M REST API](#)
- SCSI fencing, [SCSI Persistent Reservations](#)
- VMware (SOAP interface), [VMWare over SOAP API](#)
- WTI power switch, [WTI Power Switch](#)

fence devices, [Fence Devices](#)

Fence virt fence device , [Fence Virt](#)

fence_apc fence agent, [APC Power Switch over Telnet and SSH](#)

fence_apc_snmp fence agent, [APC Power Switch over Telnet and SSH](#)

fence_bladecenter fence agent, [IBM BladeCenter](#)

fence_brocade fence agent, [Brocade Fabric Switch](#)

fence_cisco_mds fence agent, [Cisco MDS](#)

fence_cisco_ucs fence agent, [Cisco UCS](#)

fence_drac5 fence agent, [Dell Drac 5](#)

fence_eaton_snmp fence agent, [Eaton Network Power Switch](#)

fence_egenera fence agent, [Egenera BladeFrame](#)

fence_eps fence agent, [ePowerSwitch](#)

fence_hpblade fence agent, [Hewlett-Packard BladeSystem](#)

fence_ibmblade fence agent, [IBM BladeCenter over SNMP](#)

fence_ifmib fence agent, [IF-MIB](#)

fence_ilo fence agent, [Hewlett-Packard iLO](#)

fence_ilo_mp fence agent, [Hewlett-Packard iLO MP](#)

fence_intelmodular fence agent, [Intel Modular](#)

fence_ipdu fence agent, [IBM iPDU](#)

fence_ipmilan fence agent, [IPMI over LAN](#)

fence_rhevm fence agent, [RHEV-M REST API](#)

fence_rsb fence agent, [Fujitsu-Siemens RemoteView Service Board \(RSB\)](#)

fence_scsi fence agent, [SCSI Persistent Reservations](#)

fence_virt fence agent, [Fence Virt](#)

fence_vmware_soap fence agent, [VMWare over SOAP API](#)

fence_wti fence agent, [WTI Power Switch](#)

fencing

- configuration, [Configuring Fencing with the ccs Command](#), [Configuring Fencing with Conga](#)

fencing configuration, [Configuring Fencing with the ccs Command](#)

Fujitsu Siemens RemoteView Service Board (RSB) fence device, [Fujitsu-Siemens RemoteView Service Board \(RSB\)](#)

H

help

- getting help, [Do You Need Help?](#)

HP Bladesystem fence device , [Hewlett-Packard BladeSystem](#)

HP iLO MP fence device , [Hewlett-Packard iLO MP](#)

HP iLO/iLO2 fence device, [Hewlett-Packard iLO](#)

I

IBM BladeCenter fence device , [IBM BladeCenter](#)

IBM BladeCenter SNMP fence device , [IBM BladeCenter over SNMP](#)

IBM iPDU fence device , [IBM iPDU](#)

IF MIB fence device , [IF-MIB](#)

integrated fence devices

- configuring ACPI, [Configuring ACPI For Use with Integrated Fence Devices](#)

Intel Modular fence device , [Intel Modular](#)

IPMI LAN fence device , [IPMI over LAN](#)

R

RHEV-M REST API fence device , [RHEV-M REST API](#)

S

SCSI fencing, [SCSI Persistent Reservations](#)

T

tables

- fence devices, parameters, [Fence Devices](#)

V

VMware (SOAP interface) fence device , [VMWare over SOAP API](#)

W

WTI power switch fence device , [WTI Power Switch](#)