Red Hat Enterprise Linux 6

Fence Configuration Guide

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

Last Updated: 2018-02-08

Red Hat Enterprise Linux 6 Fence Configuration Guide

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

Steven Levine
Red Hat Customer Content Services
slevine@redhat.com

John Ha
Red Hat Customer Content Services

Legal Notice

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

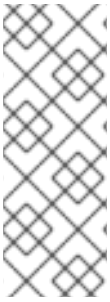
CHAPTER 1. FENCING PRE-CONFIGURATION

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.

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.

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:

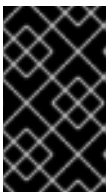
- Changing the BIOS setting to "instant-off" or an equivalent setting that turns off the node without delay

  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

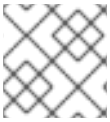
  This method completely disables ACPI; some computers do not boot correctly if ACPI is completely disabled. Use this method *only* if the other methods are not effective for your cluster.

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

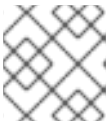
This is the preferred method of disabling ACPI Soft-Off.

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.

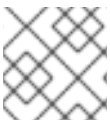


NOTE

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.



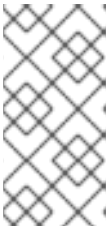
NOTE

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 by means of 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**.

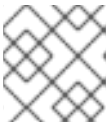


NOTE

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 by means of 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

```
+---------------------------------------|-----------------+
|   ACPI Function             [Enabled]    |  Item Help      |
|   ACPI Suspend Type         [S1(POS)]    |-----------------|
| x Run VGABIOS if S3 Resume  [Auto]       | Menu Level   *  |
|   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

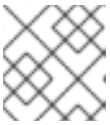
ACPI Soft-Off with the BIOS"). If neither of those methods is effective for your cluster, you can disable ACPI completely by appending **acpi=off** to the kernel boot command line in the **grub.conf** file.

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** (see 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 /initramrs-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".

1.2. SELINUX

The High Availability Add-On for Red Hat Enterprise Linux 6 supports SELinux in the **enforcing** state with the SELinux policy type set to **targeted**.

NOTE

When using SELinux with the High Availability Add-On in a VM environment, you should ensure that the SELinux boolean **fenced_can_network_connect** is persistently set to **on**. This allows the **fence_xvm** fencing agent to work properly, enabling the system to fence virtual machines.

For more information about SELinux, see *Deployment Guide* for Red Hat Enterprise Linux 6.

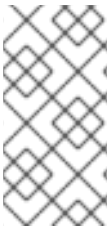
CHAPTER 2. CONFIGURING FENCING WITH THE CCS COMMAND

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”



NOTE

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.

NOTE

This chapter references commonly used **cluster.conf** elements and attributes. For a comprehensive list and description of **cluster.conf** elements and attributes, see 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.

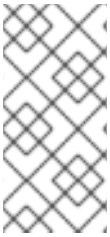
```
ccs -h host --setfencedaemon post_fail_delay=value
```

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, see the fenced(8) man page and see 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 **node-01** 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 node-01 --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, see 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 **node-01**, execute the following command:

```
ccs -h node-01 --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 **node-01**, showing sample output.

```
[root@ask-03 ~]# ccs -h node-01 --lsfenceopts
fence_apc - Fence agent for APC over telnet/ssh
fence_apc_snmp - Fence agent for APC, Tripplite PDU over SNMP
fence_bladecenter - Fence agent for IBM BladeCenter
fence_bladecenter_snmp - Fence agent for IBM BladeCenter over SNMP
fence_brocade - Fence agent for HP Brocade over telnet/ssh
fence_cisco_mds - Fence agent for Cisco MDS
fence_cisco_ucs - Fence agent for Cisco UCS
fence_drac - fencing agent for Dell Remote Access Card
fence_drac5 - Fence agent for Dell DRAC CMC/5
fence_eaton_snmp - Fence agent for Eaton over SNMP
fence_egenera - I/O Fencing agent for the Egenera BladeFrame
fence_emerson - Fence agent for Emerson over SNMP
fence_eps - Fence agent for ePowerSwitch
fence_hpblade - Fence agent for HP BladeSystem
fence_ibmblade - Fence agent for IBM BladeCenter over SNMP
fence_idrac - Fence agent for IPMI
fence_ifmib - Fence agent for IF MIB
fence_ilo - Fence agent for HP iLO
fence_ilo2 - Fence agent for HP iLO
fence_ilo3 - Fence agent for IPMI
fence_ilo3_ssh - Fence agent for HP iLO over SSH
```

```
fence_ilo4 - Fence agent for IPMI
fence_ilo4_ssh - Fence agent for HP iLO over SSH
fence_ilo_moonshot - Fence agent for HP Moonshot iLO
fence_ilo_mp - Fence agent for HP iLO MP
fence_ilo_ssh - Fence agent for HP iLO over SSH
fence_imm - Fence agent for IPMI
fence_intelmodular - Fence agent for Intel Modular
fence_ipdu - Fence agent for iPDU over SNMP
fence_ipmilan - Fence agent for IPMI
fence_kdump - Fence agent for use with kdump
fence_mpath - Fence agent for multipath persistent reservation
fence_pcmk - Helper that presents a RHCS-style interface to stonith-ng for
CMAN based clusters
fence_rhevm - Fence agent for RHEV-M REST API
fence_rsa - Fence agent for IBM RSA
fence_rsb - I/O Fencing agent for Fujitsu-Siemens RSB
fence_sanbox2 - Fence agent for QLogic SANBox2 FC switches
fence_sanlock - Fence agent for watchdog and shared storage
fence_scsi - fence agent for SCSI-3 persistent reservations
fence_tripplite_snmp - Fence agent for APC, Tripplite PDU over SNMP
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
[root@host-138 ~]# ccs -h host-138 --lsfenceopts
fence_apc - Fence agent for APC over telnet/ssh
fence_apc_snmp - Fence agent for APC, Tripplite PDU over SNMP
fence_bladecenter - Fence agent for IBM BladeCenter
fence_bladecenter_snmp - Fence agent for IBM BladeCenter over SNMP
fence_brocade - Fence agent for HP Brocade over telnet/ssh
fence_cisco_mds - Fence agent for Cisco MDS
fence_cisco_ucs - Fence agent for Cisco UCS
fence_drac - fencing agent for Dell Remote Access Card
fence_drac5 - Fence agent for Dell DRAC CMC/5
fence_eaton_snmp - Fence agent for Eaton over SNMP
fence_egenera - I/O Fencing agent for the Egenera BladeFrame
fence_emerson - Fence agent for Emerson over SNMP
fence_eps - Fence agent for ePowerSwitch
fence_hpblade - Fence agent for HP BladeSystem
fence_ibmblade - Fence agent for IBM BladeCenter over SNMP
fence_idrac - Fence agent for IPMI
fence_ifmib - Fence agent for IF MIB
fence_ilo - Fence agent for HP iLO
fence_ilo2 - Fence agent for HP iLO
fence_ilo3 - Fence agent for IPMI
fence_ilo3_ssh - Fence agent for HP iLO over SSH
fence_ilo4 - Fence agent for IPMI
fence_ilo4_ssh - Fence agent for HP iLO over SSH
fence_ilo_moonshot - Fence agent for HP Moonshot iLO
fence_ilo_mp - Fence agent for HP iLO MP
fence_ilo_ssh - Fence agent for HP iLO over SSH
fence_imm - Fence agent for IPMI
fence_intelmodular - Fence agent for Intel Modular
fence_ipdu - Fence agent for iPDU over SNMP
```

```
fence_ipmilan - Fence agent for IPMI
fence_kdump - Fence agent for use with kdump
fence_mpath - Fence agent for multipath persistent reservation
fence_pcmk - Helper that presents a RHCS-style interface to stonith-ng for
CMAN based clusters
fence_rhevm - Fence agent for RHEV-M REST API
fence_rsa - Fence agent for IBM RSA
fence_rsb - I/O Fencing agent for Fujitsu-Siemens RSB
fence_sanbox2 - Fence agent for QLogic SANBox2 FC switches
fence_sanlock - Fence agent for watchdog and shared storage
fence_scsi - fence agent for SCSI-3 persistent reservations
fence_tripplite_snmp - Fence agent for APC, Tripplite PDU over SNMP
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 node-01 --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 2.3.6, “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, see 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, "**cluster.conf** After Adding Storage-Based Fence Methods " shows a **cluster.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.

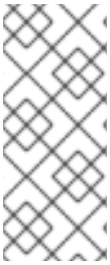
Example 2.3, "**cluster.conf** After Adding Backup Fence Methods " shows a **cluster.conf** configuration file after you have added a power-based primary fencing method and a storage-based backup fencing method to each node in the cluster.

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. Testing the Fence Configuration

As of Red Hat Enterprise Linux Release 6.4, you can test the fence configuration for each node in a cluster with the **fence_check** utility.

The following example shows the output of a successful execution of this command.

```
[root@host-098 ~]# fence_check
fence_check run at Wed Jul 23 09:13:57 CDT 2014 pid: 4769
Testing host-098 method 1: success
Testing host-099 method 1: success
Testing host-100 method 1: success
```

For information on this utility, see the **fence_check**(8) man page.

2.3.6. 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”

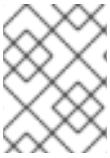
3.1. CONFIGURING FENCE DAEMON PROPERTIES

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**, see 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.

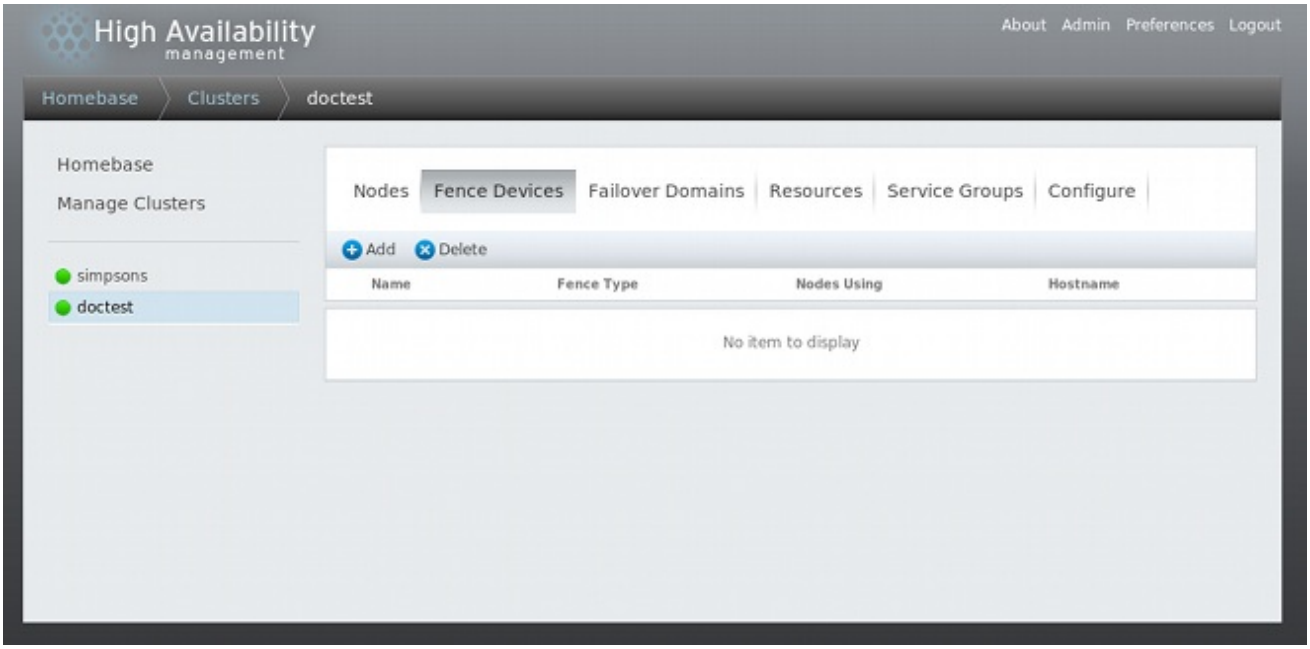


Figure 3.1. luci fence devices configuration page

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.

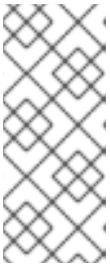
3. On the node-specific page, under **Fence Devices**, click **`Add Fence Method`**. This displays the **`Add Fence Method to Node`** dialog box.

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, see 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”.

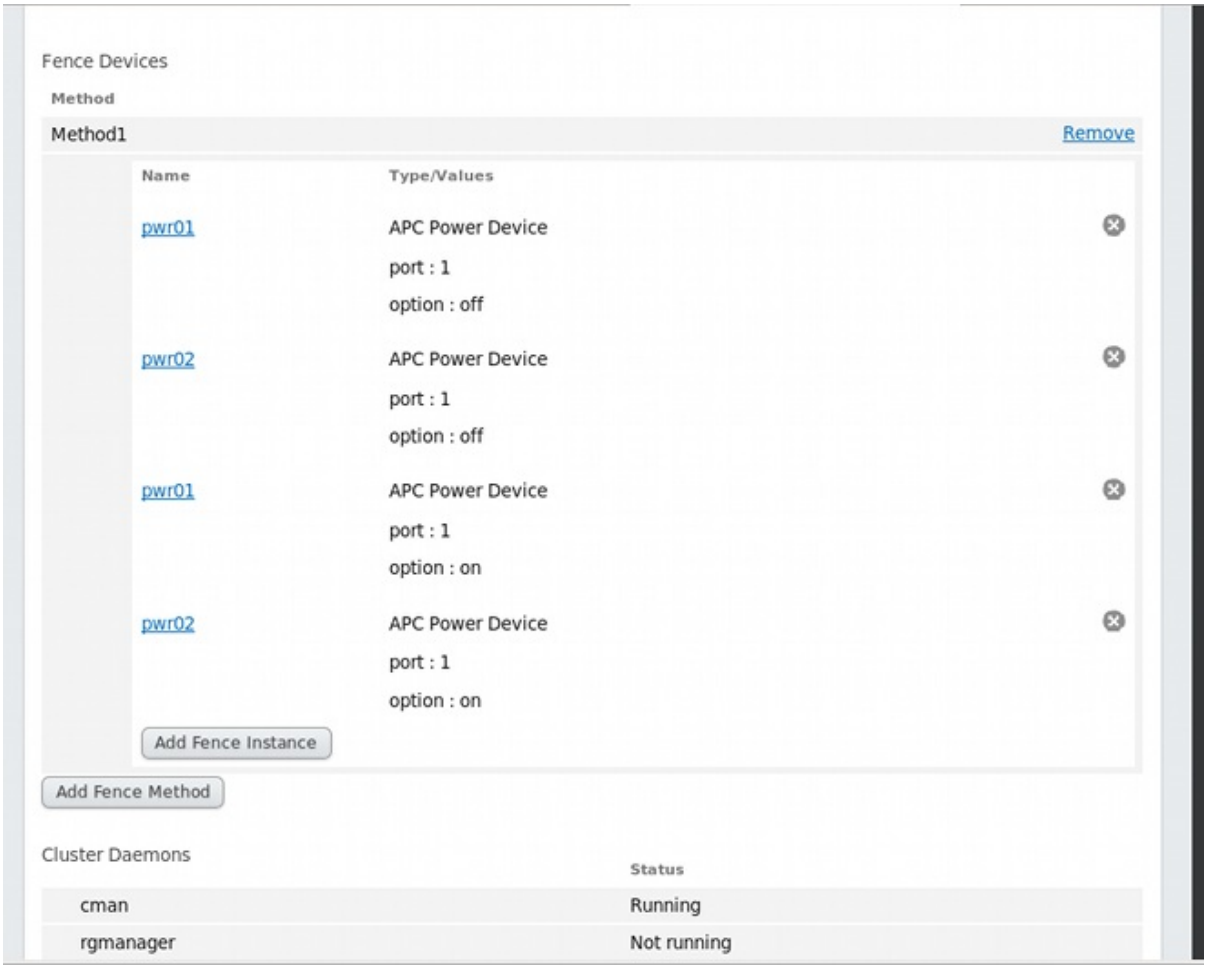


Figure 3.2. Dual-Power Fencing Configuration

3.3.4. Testing the Fence Configuration

As of Red Hat Enterprise Linux Release 6.4, you can test the fence configuration for each node in a cluster with the **fence_check** utility.

The following example shows the output of a successful execution of this command.

```
[root@host-098 ~]# fence_check
fence_check run at Wed Jul 23 09:13:57 CDT 2014 pid: 4769
Testing host-098 method 1: success
Testing host-099 method 1: success
Testing host-100 method 1: success
```

For information on this utility, see the **fence_check**(8) man page.

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	Table 4.2, “APC Power Switch (telnet/SSH)”
APC Power Switch over SNMP	fence_apc_snmp	Table 4.3, “APC Power Switch over SNMP”
Brocade Fabric Switch	fence_brocade	Table 4.4, “Brocade Fabric Switch”
Cisco MDS	fence_cisco_mds	Table 4.5, “Cisco MDS”
Cisco UCS	fence_cisco_ucs	Table 4.6, “Cisco UCS”
Dell DRAC 5	fence_drac5	Table 4.7, “Dell DRAC 5”
Dell iDRAC	fence_idrac	Table 4.25, “IPMI (Intelligent Platform Management Interface) LAN, Dell iDrac, IBM Integrated Management Module, HPiLO3, HPiLO4”
Eaton Network Power Switch (SNMP Interface)	fence_eaton_snmp	Table 4.8, “Eaton Network Power Controller (SNMP Interface) (Red Hat Enterprise Linux 6.4 and later)”
Egenera BladeFrame	fence_egenera	Table 4.9, “Egenera BladeFrame”
Emerson Network Power Switch (SNMP Interface)	fence_emerson	Table 4.10, “Emerson Network Power Switch (SNMP interface) (Red Hat Enterprise Linux 6.7 and later)”
ePowerSwitch	fence_eps	Table 4.11, “ePowerSwitch”
Fence virt (Serial/VMChannel Mode)	fence_virt	Table 4.12, “Fence virt (Serial/VMChannel Mode)”
Fence virt (fence_xvm/Multicast Mode)	fence_xvm	Table 4.13, “Fence virt (Multicast Mode) ”

Fence Device	Fence Agent	Reference to Parameter Description
Fujitsu Siemens Remoteview Service Board (RSB)	fence_rsb	Table 4.14, “Fujitsu Siemens Remoteview Service Board (RSB)”
HP BladeSystem	fence_hpblade	Table 4.15, “HP BladeSystem (Red Hat Enterprise Linux 6.4 and later)”
HP iLO Device	fence_ilo	Table 4.16, “HP iLO (Integrated Lights Out) and HP iLO2”
HP iLO over SSH Device	fence_ilo3_ssh	Table 4.17, “HP iLO over SSH, HP iLO3 over SSH, HPiLO4 over SSH (Red Hat Enterprise Linux 6.7 and later)”
HP iLO4 Device	fence_ilo4	Table 4.25, “IPMI (Intelligent Platform Management Interface) LAN, Dell iDrac, IBM Integrated Management Module, HPiLO3, HPiLO4”
HP iLO4 over SSH Device	fence_ilo4_ssh	Table 4.17, “HP iLO over SSH, HP iLO3 over SSH, HPiLO4 over SSH (Red Hat Enterprise Linux 6.7 and later)”
HP iLO MP	fence_ilo_mp	Table 4.18, “HP iLO (Integrated Lights Out) MP”
HP Moonshot iLO	fence_ilo_moonshot	Table 4.19, “HP Moonshot iLO (Red Hat Enterprise Linux 6.7 and later)”
IBM BladeCenter	fence_bladecenter	Table 4.20, “IBM BladeCenter”
IBM BladeCenter SNMP	fence_ibmblade	Table 4.21, “IBM BladeCenter SNMP”
IBM Integrated Management Module	fence_imm	Table 4.25, “IPMI (Intelligent Platform Management Interface) LAN, Dell iDrac, IBM Integrated Management Module, HPiLO3, HPiLO4”
IBM iPDU	fence_ipdu	Table 4.22, “IBM iPDU (Red Hat Enterprise Linux 6.4 and later)”
IF MIB	fence_ifmib	Table 4.23, “IF MIB”
Intel Modular	fence_intelmodular	Table 4.24, “Intel Modular”
IPMI (Intelligent Platform Management Interface) Lan	fence_ipmilan	Table 4.25, “IPMI (Intelligent Platform Management Interface) LAN, Dell iDrac, IBM Integrated Management Module, HPiLO3, HPiLO4”
Fence kdump	fence_kdump	Table 4.26, “Fence kdump”

Fence Device	Fence Agent	Reference to Parameter Description
Multipath Persistent Reservation Fencing	fence_mpath	Table 4.27, “Multipath Persistent Reservation Fencing (Red Hat Enterprise Linux 6.7 and later)”
RHEV-M fencing	fence_rhevm	Table 4.28, “RHEV-M REST API (RHEL 6.2 and later against RHEV 3.0 and later)”
SCSI Fencing	fence_scsi	Table 4.29, “SCSI Reservation Fencing”
VMware Fencing (SOAP Interface)	fence_vmware_soap	Table 4.30, “VMware Fencing (SOAP Interface) (Red Hat Enterprise Linux 6.2 and later)”
WTI Power Switch	fence_wti	Table 4.31, “WTI Power Switch”

4.1. APC POWER SWITCH OVER TELNET AND SSH

Table 4.2, “APC Power Switch (telnet/SSH)” lists the fence device parameters used by **fence_apc**, the fence agent for APC over telnet/SSH.

Table 4.2. 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 by means of telnet/ssh.
IP Address or Hostname	**ipaddr**	The IP address or host name assigned to the device.
IP Port (optional)	**ipport**	The TCP port to use to connect to the device. The default port is 23, or 22 if **Use SSH** is selected.
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 (seconds)	**power_wait**	Number of seconds to wait after issuing a power off or power on command.
Power Timeout (seconds)	**power_timeout**	Number of seconds to continue testing for a status change after issuing a power off or power on command. The default value is 20.

luci Field	cluster.conf Attribute	Description
Shell Timeout (seconds)	**shell_timeout**	Number of seconds to wait for a command prompt after issuing a command. The default value is 3.
Login Timeout (seconds)	**login_timeout**	Number of seconds to wait for a command prompt after login. The default value is 5.
Times to Retry Power On Operation	**retry_on**	Number of attempts to retry a power on operation. The default value is 1.
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.
Delay (optional)	**delay**	The number of seconds to wait before fencing is started. The default value is 0.
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.
SSH Options	**ssh_options**	SSH options to use. The default value is **-1 -c blowfish**.
Path to SSH Identity File	**identity_file**	The identity file for SSH.

Figure 4.1, “APC Power Switch” shows the configuration screen for adding an APC Power Switch fence device.

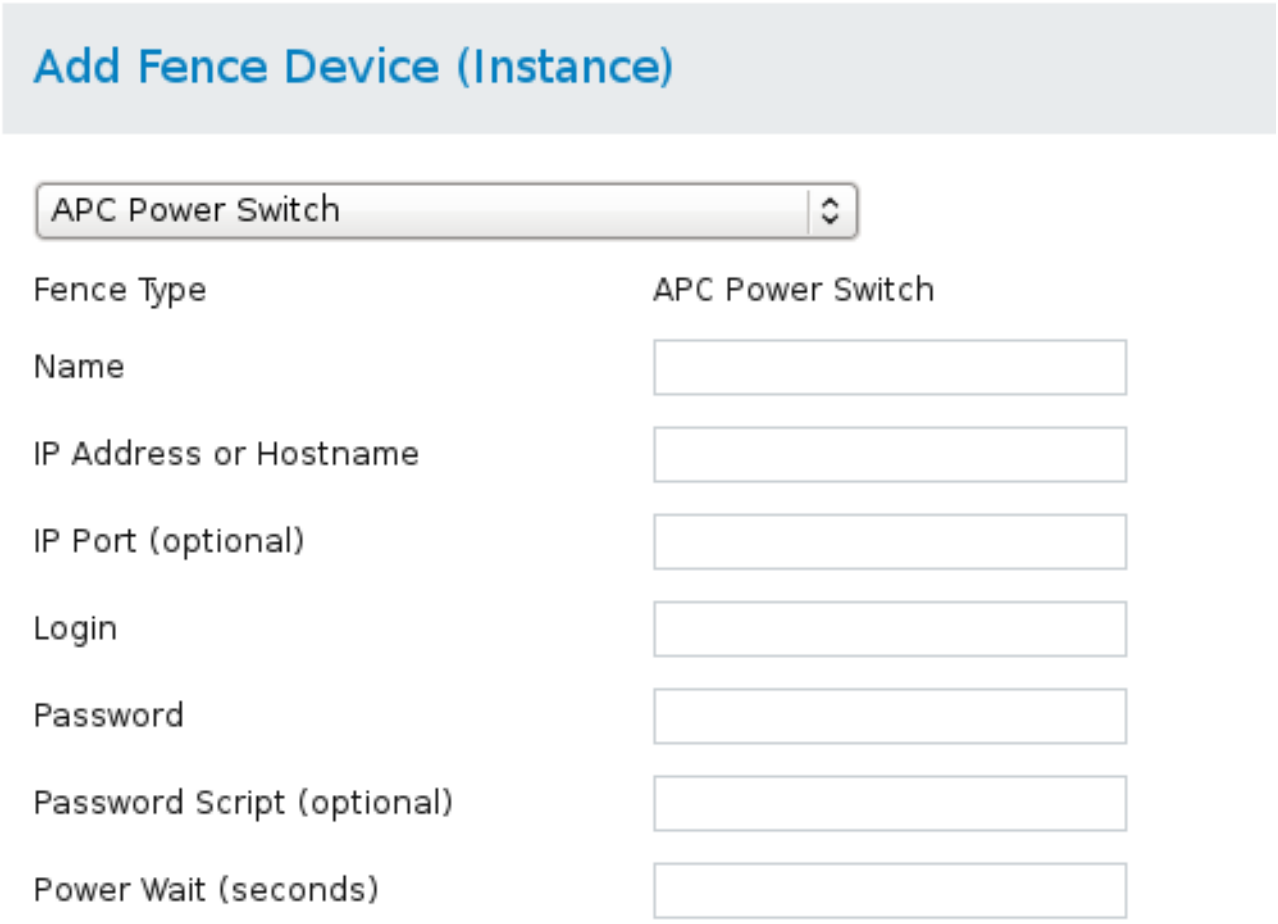


Figure 4.1. APC Power Switch

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>
```

4.2. APC POWER SWITCH OVER SNMP

Table 4.3, “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 by means of the SNMP protocol.

Table 4.3. 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 by means of the SNMP protocol.
IP Address or Hostname	**ipaddr**	The IP address or host name 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_passwd**	The SNMP privacy protocol password.
SNMP Privacy Protocol Script	**snmp_priv_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.
Power Timeout (seconds)	**power_timeout**	Number of seconds to continue testing for a status change after issuing a power off or power on command. The default value is 20.

luci Field	cluster.conf Attribute	Description
Shell Timeout (seconds)	**shell_timeout**	Number of seconds to wait for a command prompt after issuing a command. The default value is 3.
Login Timeout (seconds)	**login_timeout**	Number of seconds to wait for a command prompt after login. The default value is 5.
Times to Retry Power On Operation	**retry_on**	Number of attempts to retry a power on operation. The default value is 1.
Port (Outlet) Number	**port**	The port.
Delay (optional)	**delay**	The number of seconds to wait before fencing is started. The default value is 0.

Figure 4.2, “APC Power Switch over SNMP” shows the configuration screen for adding an APC Power Switch fence device.

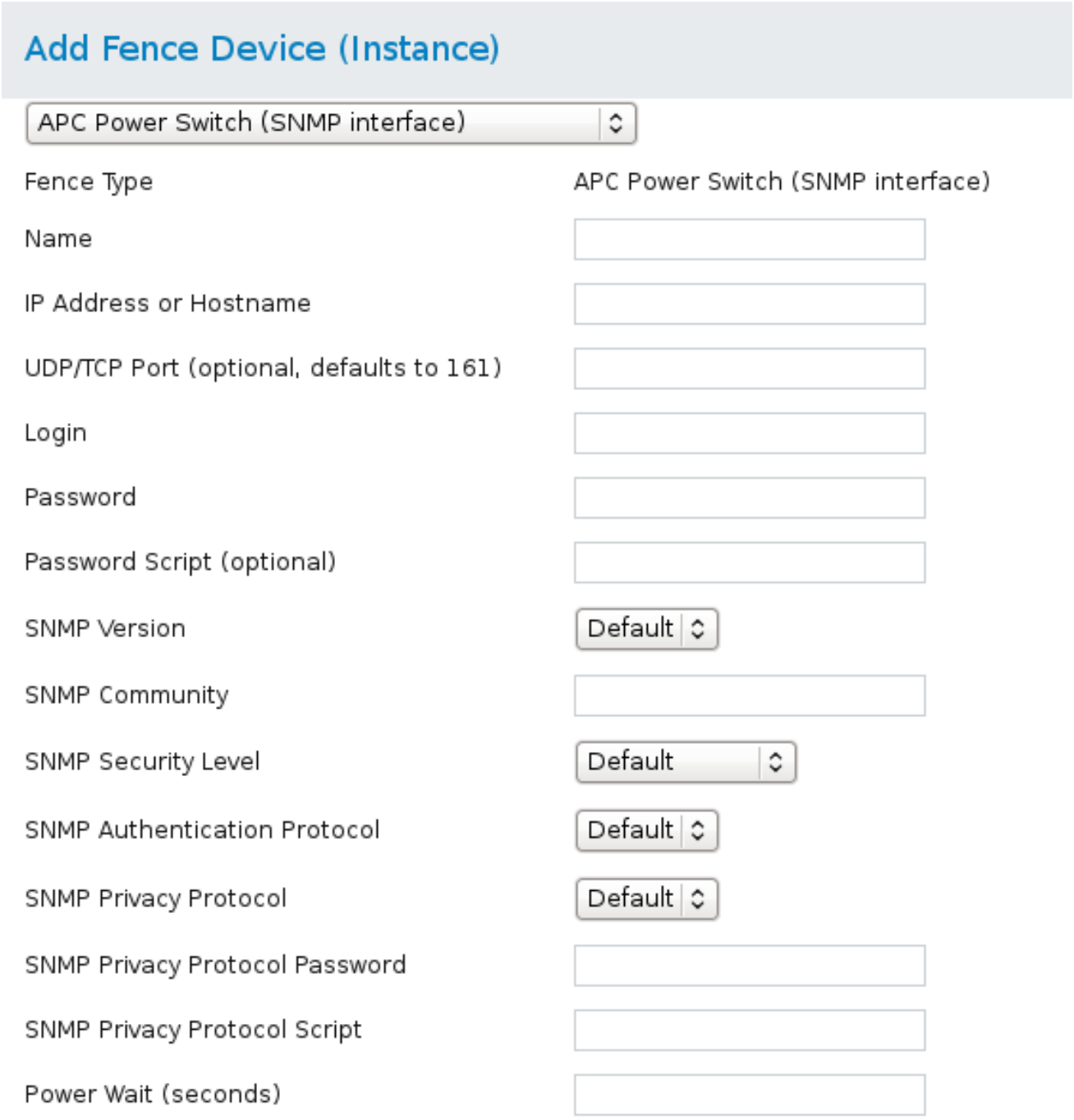


Figure 4.2. APC Power Switch over SNMP

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>
```

4.3. 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 or host name 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.
Force IP Family	**inet4_only**, **inet6_only**	Force the agent to use IPv4 or IPv6 addresses only
Force Command Prompt	**cmd_prompt**	The command prompt to use. The default value is '\$'.
Power Wait (seconds)	**power_wait**	Number of seconds to wait after issuing a power off or power on command.
Power Timeout (seconds)	**power_timeout**	Number of seconds to continue testing for a status change after issuing a power off or power on command. The default value is 20.
Shell Timeout (seconds)	**shell_timeout**	Number of seconds to wait for a command prompt after issuing a command. The default value is 3.
Login Timeout (seconds)	**login_timeout**	Number of seconds to wait for a command prompt after login. The default value is 5.
Times to Retry Power On Operation	**retry_on**	Number of attempts to retry a power on operation. The default value is 1.
Port	**port**	The switch outlet number.
Delay (optional)	**delay**	The number of seconds to wait before fencing is started. The default value is 0.

luci Field	cluster.conf Attribute	Description
Use SSH	**secure**	Indicates that the system will use SSH to access the device. When using SSH, you must specify either a password, a password script, or an identity file.
SSH Options	**ssh_options**	SSH options to use. The default value is **-1 -c blowfish**.
Path to SSH Identity File	**identity_file**	The identity file for SSH.
Unfencing	**unfence** section of the cluster configuration file	When enabled, this ensures that a fenced node is not re-enabled until the node has been rebooted. This is necessary for non-power fence methods (that is, SAN/storage fencing). When you configure a device that requires unfencing, the cluster must first be stopped and the full configuration including devices and unfencing must be added before the cluster is started. For more information about unfencing a node, see the **fence_node**(8) man page.

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

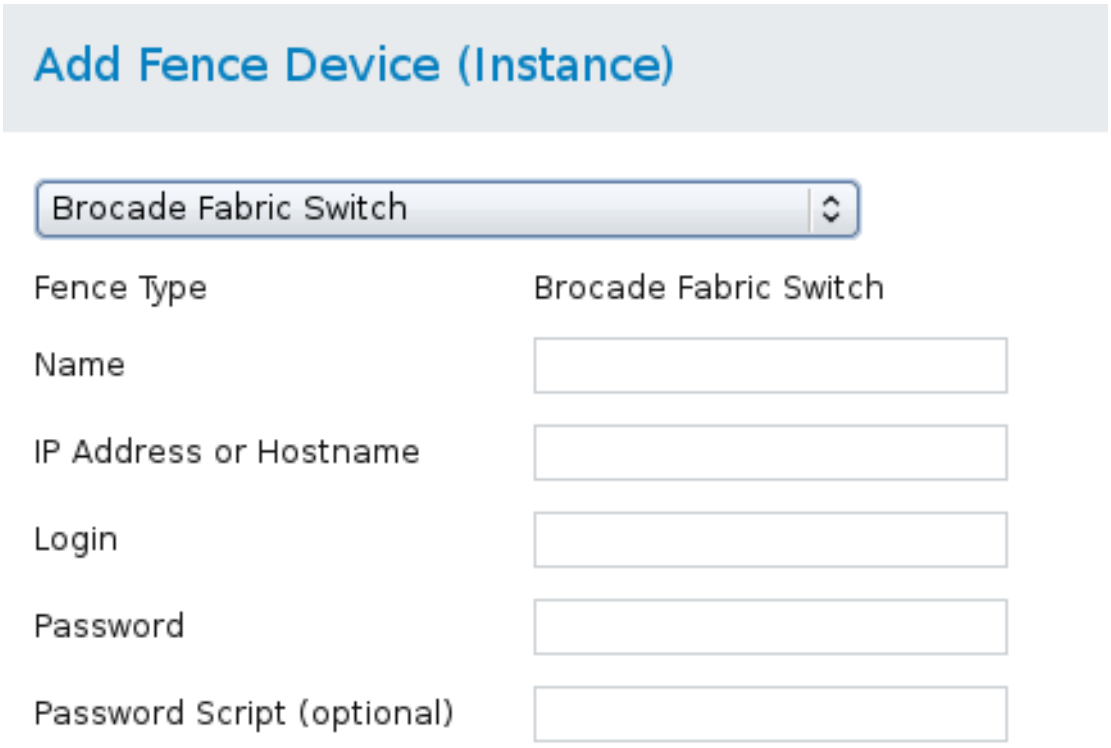


Figure 4.3. Brocade Fabric Switch

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 host name 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).
SNMP Community	**community**	The SNMP community string.
SNMP Security Level	**snmp_sec_level**	The SNMP security level (noAuthNoPriv, authNoPriv, authPriv).

luci Field	cluster.conf Attribute	Description
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_passwd	The SNMP privacy protocol password.
SNMP Privacy Protocol Script	snmp_priv_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.
Power Timeout (seconds)	power_timeout	Number of seconds to continue testing for a status change after issuing a power off or power on command. The default value is 20.
Shell Timeout (seconds)	shell_timeout	Number of seconds to wait for a command prompt after issuing a command. The default value is 3.
Login Timeout (seconds)	login_timeout	Number of seconds to wait for a command prompt after login. The default value is 5.
Times to Retry Power On Operation	retry_on	Number of attempts to retry a power on operation. The default value is 1.
Port (Outlet) Number	port	The port.
Delay (optional)	delay	The number of seconds to wait before fencing is started. The default value is 0.

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

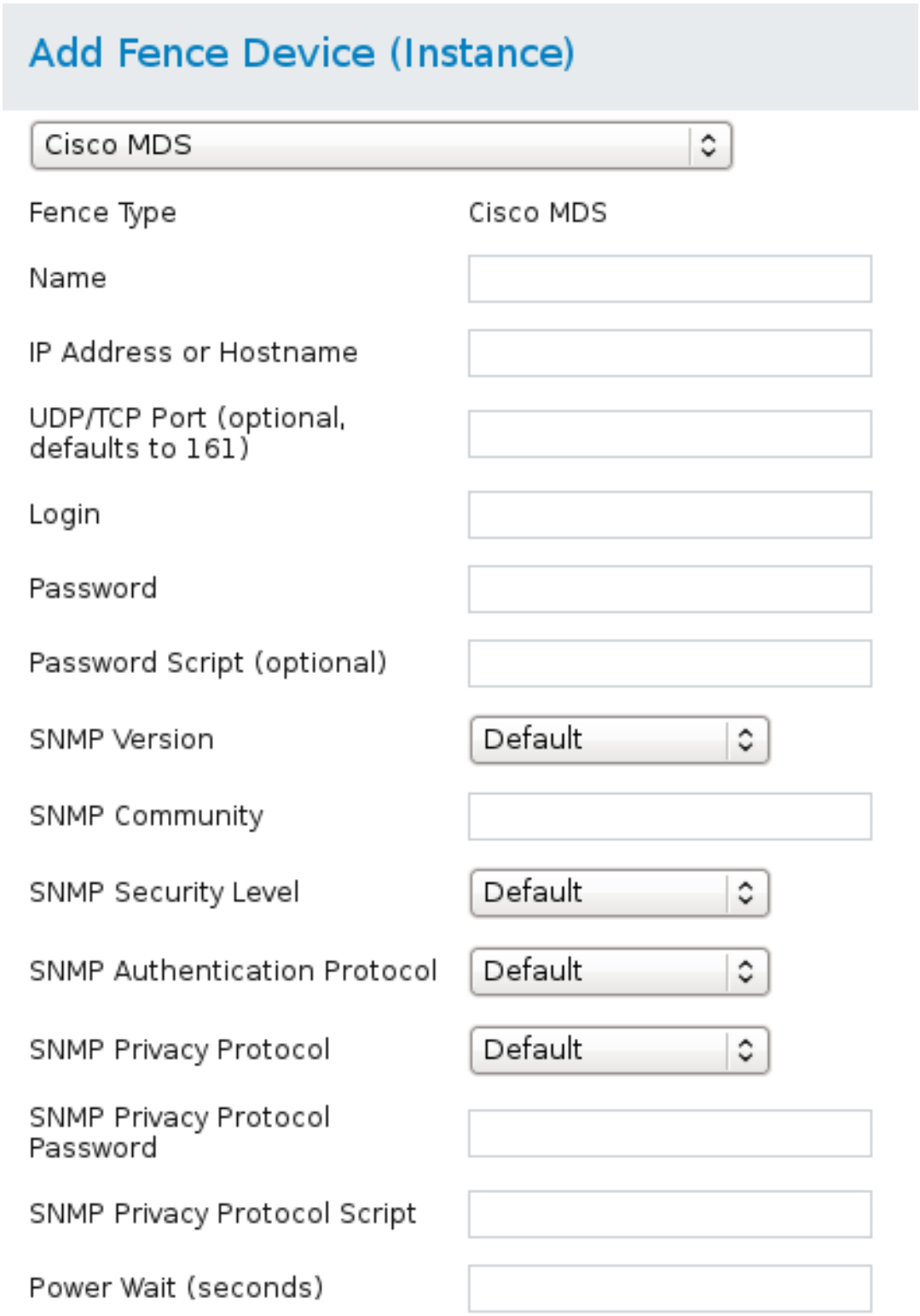


Figure 4.4. Cisco MDS

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 host name 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.
Power Wait (seconds)	**power_wait**	Number of seconds to wait after issuing a power off or power on command.
Power Timeout (seconds)	**power_timeout**	Number of seconds to continue testing for a status change after issuing a power off or power on command. The default value is 20.
Shell Timeout (seconds)	**shell_timeout**	Number of seconds to wait for a command prompt after issuing a command. The default value is 3.
Login Timeout (seconds)	**login_timeout**	Number of seconds to wait for a command prompt after login. The default value is 5.

luci Field	cluster.conf Attribute	Description
Times to Retry Power On Operation	retry_on	Number of attempts to retry a power on operation. The default value is 1.
Port (Outlet) Number	port	Name of virtual machine.
Delay (optional)	delay	The number of seconds to wait before fencing is started. The default value is 0.

Figure 4.5, “Cisco UCS” shows the configuration screen for adding a Cisco UCS fence device.

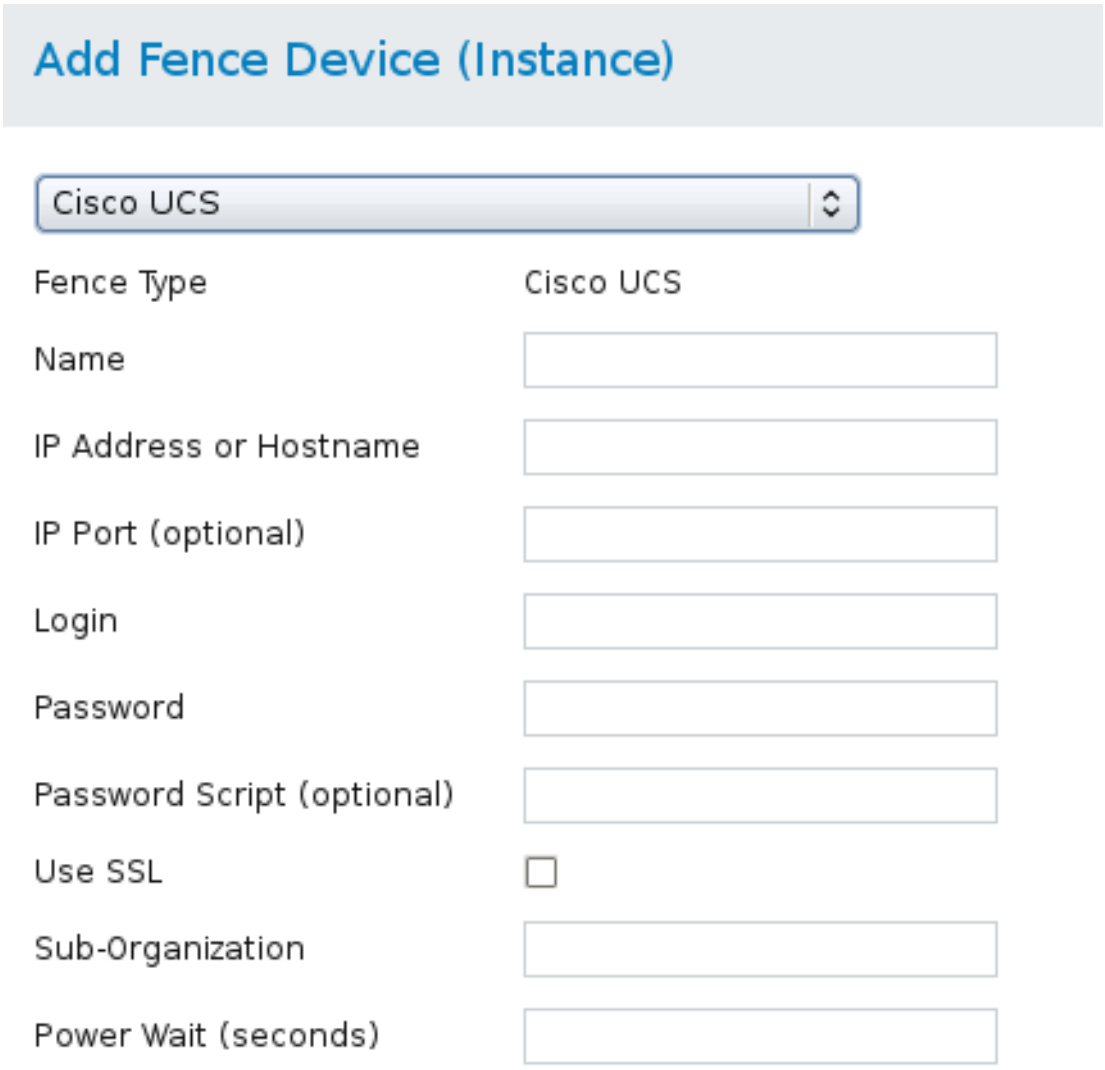


Figure 4.5. Cisco UCS

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 host name 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. When using SSH, you must specify either a password, a password script, or an identity file.
SSH Options	**ssh_options**	SSH options to use. The default value is **-1 -c blowfish**.
Path to SSH Identity File	**identity_file**	The identity file for SSH.

luci Field	cluster.conf Attribute	Description
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 (seconds)	power_wait	Number of seconds to wait after issuing a power off or power on command.
Delay (seconds)	delay	The number of seconds to wait before fencing is started. The default value is 0.
Power Timeout (seconds)	power_timeout	Number of seconds to continue testing for a status change after issuing a power off or power on command. The default value is 20.
Shell Timeout (seconds)	shell_timeout	Number of seconds to wait for a command prompt after issuing a command. The default value is 3.
Login Timeout (seconds)	login_timeout	Number of seconds to wait for a command prompt after login. The default value is 5.
Times to Retry Power On Operation	retry_on	Number of attempts to retry a power on operation. The default value is 1.

Figure 4.6, “Dell Drac 5” shows the configuration screen for adding a Dell Drac 5 device

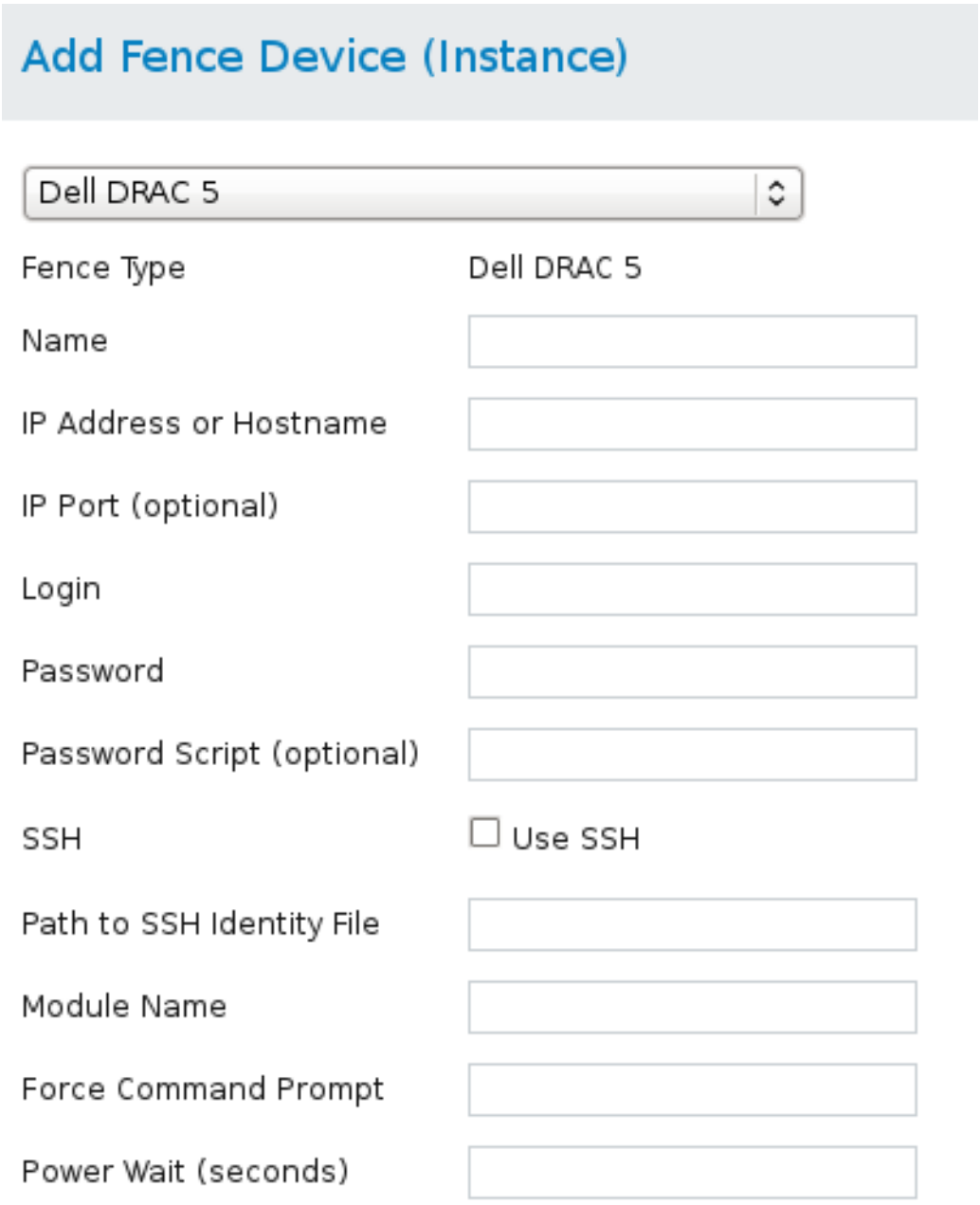


Figure 4.6. Dell Drac 5

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 host name 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_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_passwd**	The SNMP privacy protocol password.
SNMP Privacy Protocol Script	**snmp_priv_passwd_script**	The script that supplies a password for SNMP privacy protocol. Using this supersedes the **SNMP privacy protocol password** parameter.

luci Field	cluster.conf Attribute	Description
Power wait (seconds)	power_wait	Number of seconds to wait after issuing a power off or power on command.
Power Timeout (seconds)	power_timeout	Number of seconds to continue testing for a status change after issuing a power off or power on command. The default value is 20.
Shell Timeout (seconds)	shell_timeout	Number of seconds to wait for a command prompt after issuing a command. The default value is 3.
Login Timeout (seconds)	login_timeout	Number of seconds to wait for a command prompt after login. The default value is 5.
Times to Retry Power On Operation	retry_on	Number of attempts to retry a power on operation. The default value is 1.
Port (Outlet) Number	port	Physical plug number or name of virtual machine. This parameter is always required.
Delay (optional)	delay	The number of seconds to wait before fencing is started. The default value is 0.

Figure 4.7, “Eaton Network Power Switch” shows the configuration screen for adding an Eaton Network Power Switch fence device.

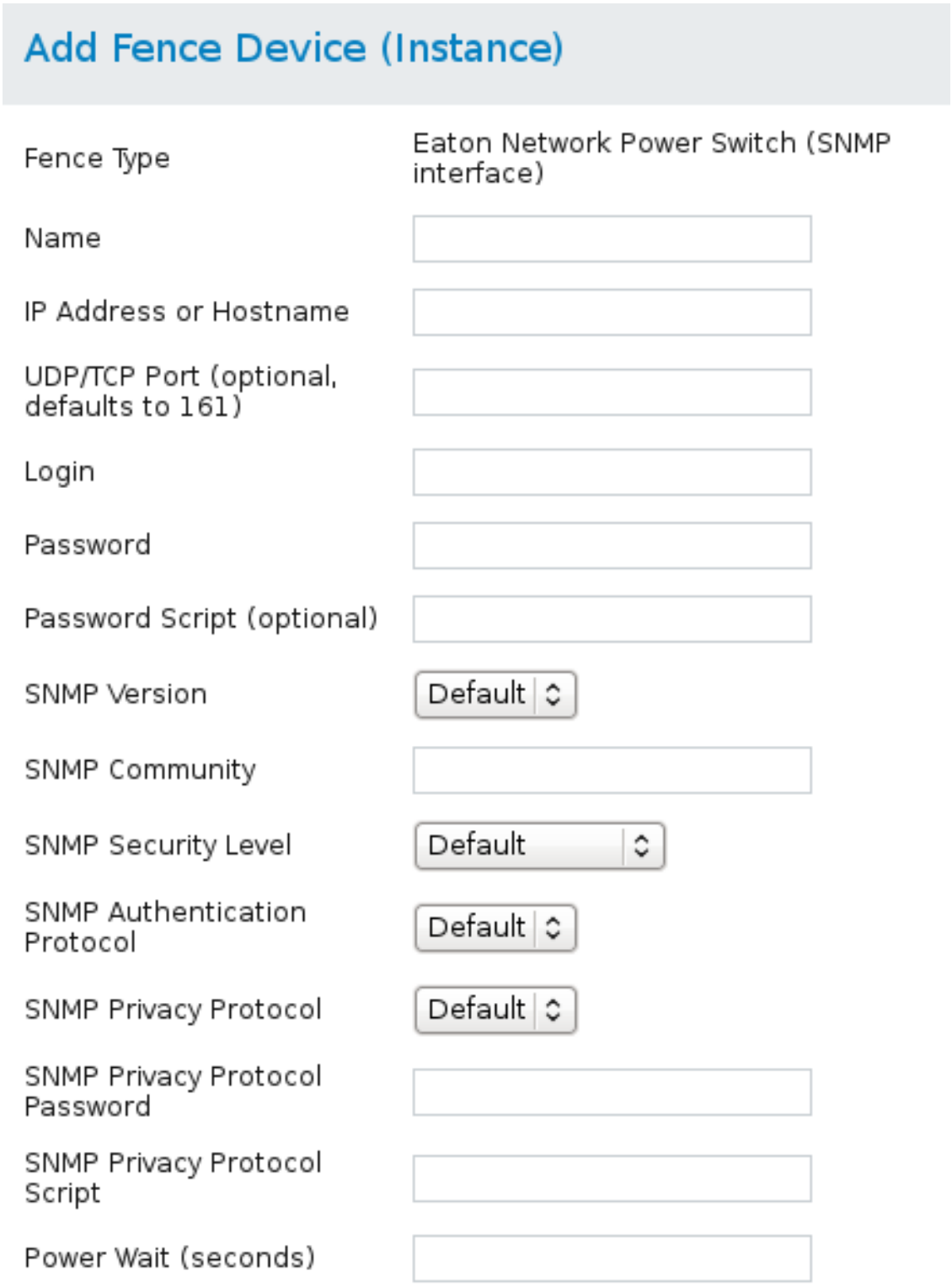


Figure 4.7. Eaton Network Power Switch

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 BladeFrame” lists the fence device parameters used by **fence_egenera**, the fence agent for the Egenera BladeFrame.

Table 4.9. Egenera BladeFrame

luci Field	cluster.conf Attribute	Description
Name	**name**	A name for the Egenera BladeFrame device connected to the cluster.
CServer	**cserver**	The host name (and optionally the user name in the form of **username@hostname**) assigned to the device. Refer to the fence_egenera(8) man page for more information.
ESH Path (optional)	**esh**	The path to the esh command on the cserver (default is /opt/panmgr/bin/esh)
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.
Delay (optional)	**delay**	The number of seconds to wait before fencing is started. The default value is 0.
Unfencing	**unfence** section of the cluster configuration file	When enabled, this ensures that a fenced node is not re-enabled until the node has been rebooted. This is necessary for non-power fence methods (that is, SAN/storage fencing). When you configure a device that requires unfencing, the cluster must first be stopped and the full configuration including devices and unfencing must be added before the cluster is started. For more information about unfencing a node, see the **fence_node**(8) man page.

Figure 4.8, “Egenera BladeFrame” shows the configuration screen for adding an Egenera BladeFrame fence device.

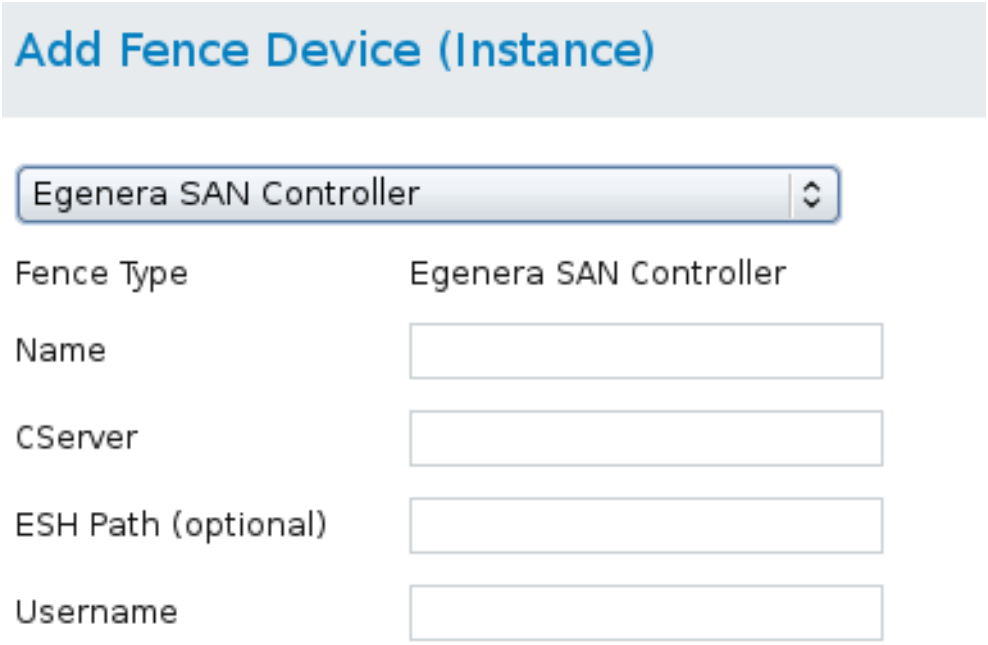


Figure 4.8. Egenera BladeFrame

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. EMERSON NETWORK POWER SWITCH (SNMP INTERFACE)

Table 4.10, “Emerson Network Power Switch (SNMP interface) (Red Hat Enterprise Linux 6.7 and later)” lists the fence device parameters used by **fence_emerson**, the fence agent for Emerson over SNMP.

Table 4.10. Emerson Network Power Switch (SNMP interface) (Red Hat Enterprise Linux 6.7 and later)

luci Field	cluster.conf Attribute	Description
Name	**name**	A name for the Emerson Network Power Switch device connected to the cluster.
IP Address or Hostname	**ipaddr**	The IP address or host name assigned to the device.

luci Field	cluster.conf Attribute	Description
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_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_passwd	The SNMP Privacy Protocol Password.
SNMP Privacy Protocol Script	snmp_priv_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.
Power Timeout (seconds)	power_timeout	Number of seconds to continue testing for a status change after issuing a power off or power on command. The default value is 20.
Shell Timeout (seconds)	shell_timeout	Number of seconds to wait for a command prompt after issuing a command. The default value is 3.
Login Timeout (seconds)	login_timeout	Number of seconds to wait for a command prompt after login. The default value is 5.

luci Field	cluster.conf Attribute	Description
Times to Retry Power On Operation	**retry_on**	Number of attempts to retry a power on operation. The default value is 1.
Port (Outlet) Number	**port**	Physical plug number or name of virtual machine.
Delay (optional)	**delay**	The number of seconds to wait before fencing is started. The default value is 0.

4.10. EPOWERSWITCH

Table 4.11, "ePowerSwitch" lists the fence device parameters used by **fence_eps**, the fence agent for ePowerSwitch.

Table 4.11. 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 host name 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.
Times to Retry Power On Operation	**retry_on**	Number of attempts to retry a power on operation. The default value is 1.
Port (Outlet) Number	**port**	Physical plug number or name of virtual machine.
Delay (optional)	**delay**	The number of seconds to wait before fencing is started. The default value is 0.

Figure 4.9, “ePowerSwitch” shows the configuration screen for adding an ePowerSwitch fence device.

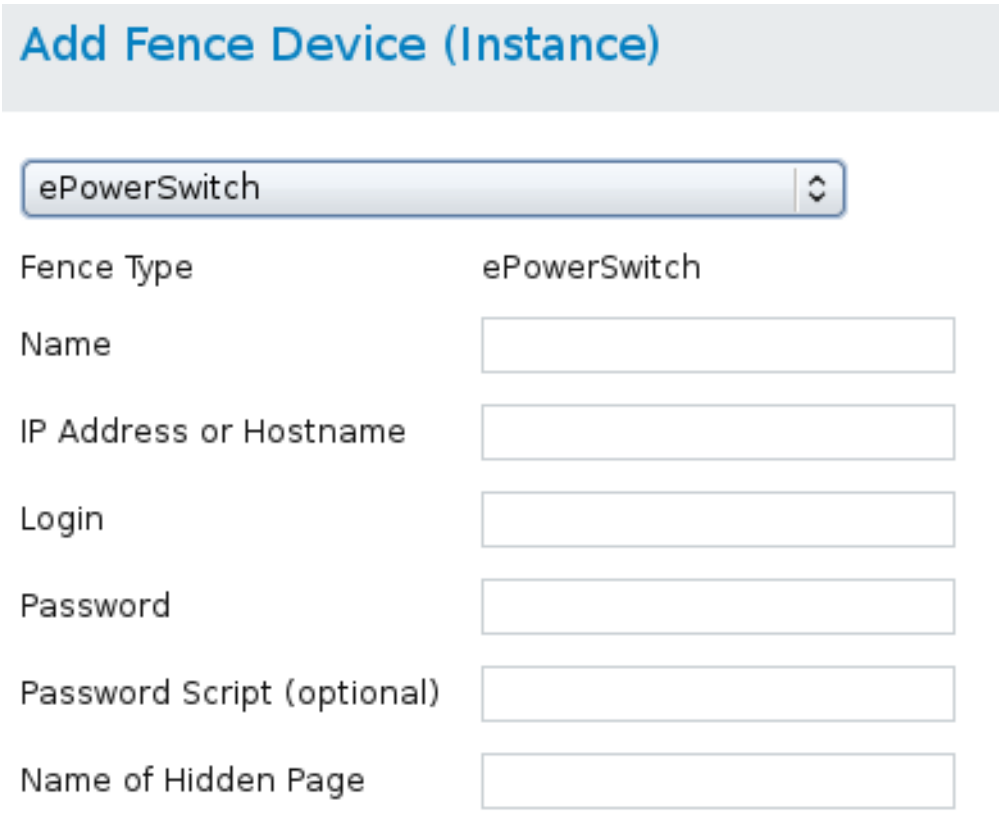


Figure 4.9. ePowerSwitch

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>
```

4.11. FENCE VIRT (SERIAL/VMCHANNEL MODE)

Table 4.12, “Fence virt (Serial/VMChannel Mode)” lists the fence device parameters used by **`fence_virt`**, the fence agent for virtual machines using VM channel or serial mode .

Table 4.12. Fence virt (Serial/VMChannel Mode)

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** 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.
Timeout (optional)	**timeout**	Fencing timeout, in seconds. The default value is 30.
Domain	**port** (formerly **domain**)	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**.
Delay (optional)	**delay**	Fencing delay, in seconds. The fence agent will wait the specified number of seconds before attempting a fencing operation. The default value is 0.

The following command creates a fence device instance for virtual machines using serial mode.

```
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>
```

4.12. FENCE VIRT (MULTICAST MODE)

Table 4.13, “Fence virt (Multicast Mode) ” lists the fence device parameters used by **fence_xvm**, the fence agent for virtual machines using multicast.

Table 4.13. Fence virt (Multicast Mode)

luci Field	cluster.conf Attribute	Description
Name	**name**	A name for the Fence virt fence device.
Timeout	**timeout**	Fencing timeout, in seconds. The default value is 30.
Domain	**port** (formerly **domain**)	Virtual machine (domain UUID or name) to fence.
Delay (optional)	**delay**	Fencing delay, in seconds. The fence agent will wait the specified number of seconds before attempting a fencing operation. The default value is 0.

4.13. FUJITSU-SIEMENS REMOTEVIEW SERVICE BOARD (RSB)

Table 4.14, "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 4.14. 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 host name 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.
Force Command Prompt	**cmd_prompt**	The command prompt to use. The default value is '\$'.
Power Wait (seconds)	**power_wait**	Number of seconds to wait after issuing a power off or power on command.
Delay (seconds)	**delay**	The number of seconds to wait before fencing is started. The default value is 0.

luci Field	cluster.conf Attribute	Description
Power Timeout (seconds)	power_timeout	Number of seconds to continue testing for a status change after issuing a power off or power on command. The default value is 20.
Shell Timeout (seconds)	shell_timeout	Number of seconds to wait for a command prompt after issuing a command. The default value is 3.
Login Timeout (seconds)	login_timeout	Number of seconds to wait for a command prompt after login. The default value is 5.
Times to Retry Power On Operation	retry_on	Number of attempts to retry a power on operation. The default value is 1.

Figure 4.10, “Fujitsu-Siemens RSB” shows the configuration screen for adding an Fujitsu-Siemens RSB fence device.

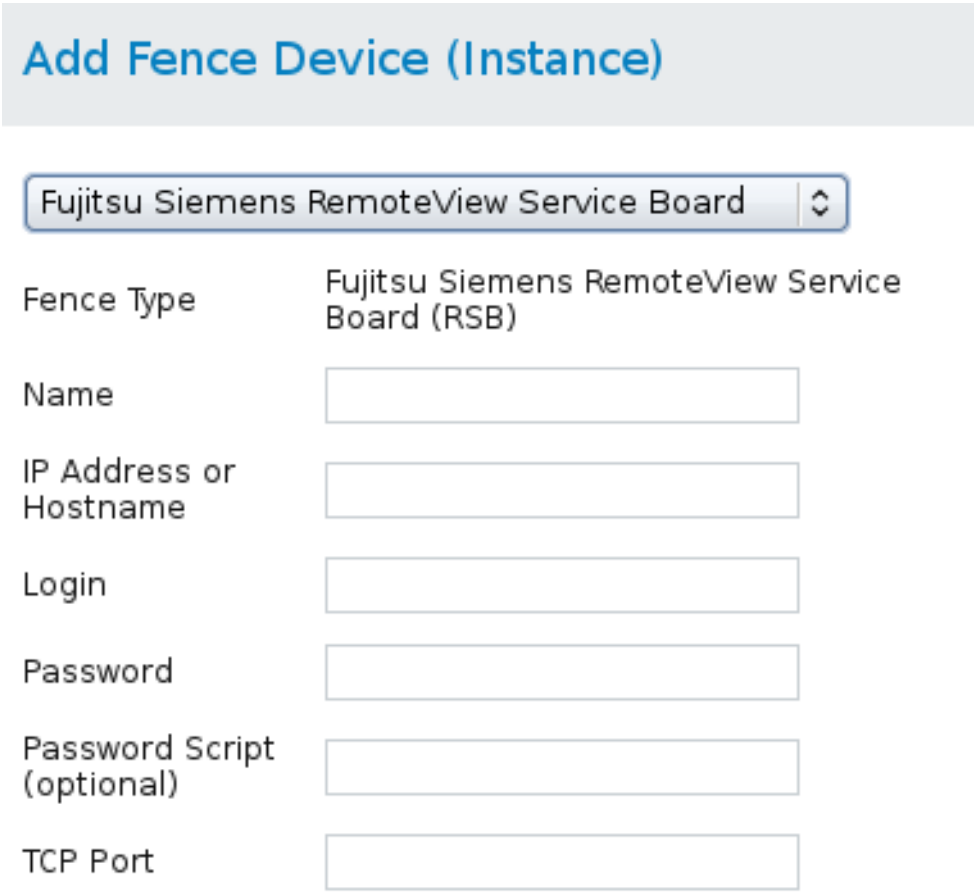


Figure 4.10. Fujitsu-Siemens RSB

The following command creates a fence device instance for a Fujiitsu-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>
```

4.14. HEWLETT-PACKARD BLADESYSTEM

Table 4.15, “HP BladeSystem (Red Hat Enterprise Linux 6.4 and later)” lists the fence device parameters used by **fence_hpblade**, the fence agent for HP BladeSystem.

Table 4.15. 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 host name 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.
Power Timeout (seconds)	**power_timeout**	Number of seconds to continue testing for a status change after issuing a power off or power on command. The default value is 20.

luci Field	cluster.conf Attribute	Description
Shell Timeout (seconds)	**shell_timeout**	Number of seconds to wait for a command prompt after issuing a command. The default value is 3.
Login Timeout (seconds)	**login_timeout**	Number of seconds to wait for a command prompt after login. The default value is 5.
Times to Retry Power On Operation	**retry_on**	Number of attempts to retry a power on operation. The default value is 1.
Use SSH	**secure**	Indicates that the system will use SSH to access the device. When using SSH, you must specify either a password, a password script, or an identity file.
SSH Options	**ssh_options**	SSH options to use. The default value is **-1 -c blowfish**.
Path to SSH Identity File	**identity_file**	The identity file for SSH.

Figure 4.11, “HP BladeSystem” shows the configuration screen for adding an HP BladeSystem fence device.

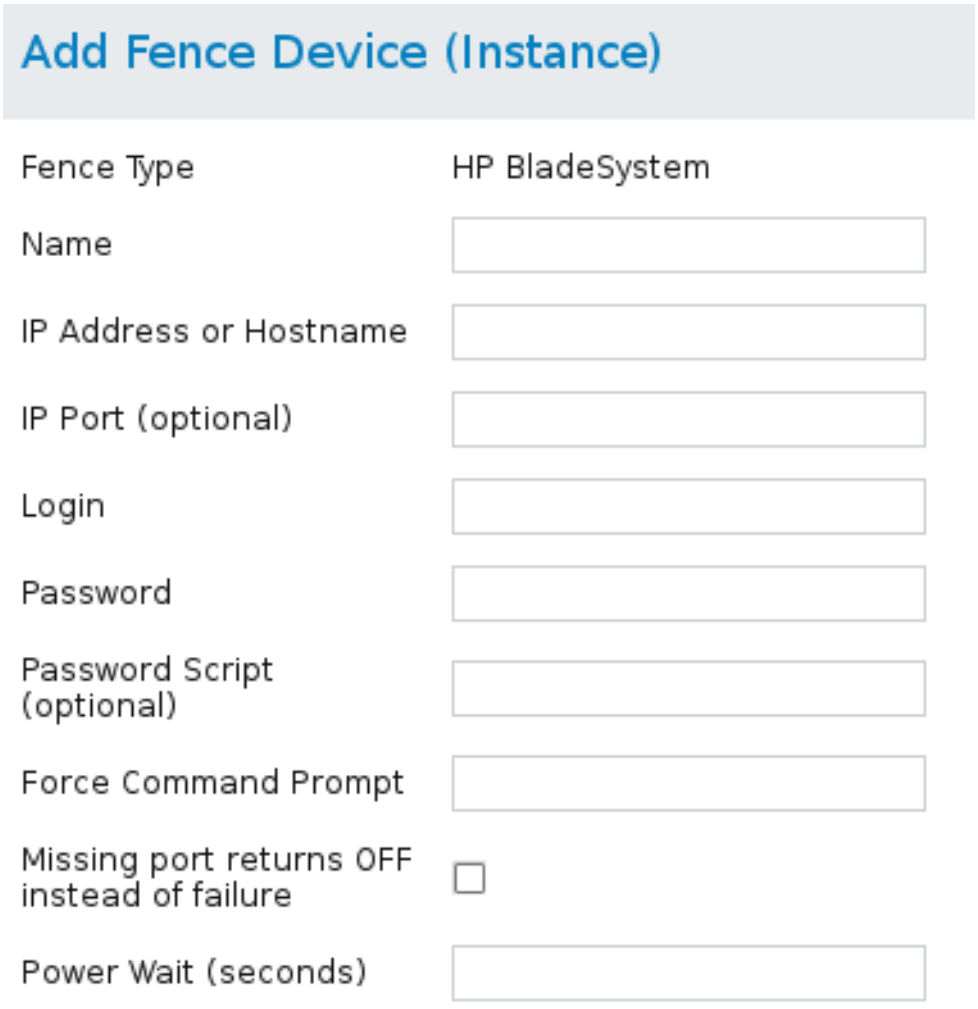


Figure 4.11. HP BladeSystem

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>
```

4.15. HEWLETT-PACKARD ILO

The fence agents for HP iLO devices **`fence_ilo`** and HP iLO2 devices **`fence_ilo2`**. share the same implementation. Table 4.16, “HP iLO (Integrated Lights Out) and HP iLO2” lists the fence device parameters used by these agents.

Table 4.16. HP iLO (Integrated Lights Out) and HP iLO2

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 host name assigned to the device.
IP Port (optional)	**`ipport`**	TCP port to use for connection with the device. The default value is 443.
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 (seconds)	**`power_wait`**	Number of seconds to wait after issuing a power off or power on command.
Delay (seconds)	**`delay`**	The number of seconds to wait before fencing is started. The default value is 0.
Power Timeout (seconds)	**`power_timeout`**	Number of seconds to continue testing for a status change after issuing a power off or power on command. The default value is 20.
Shell Timeout (seconds)	**`shell_timeout`**	Number of seconds to wait for a command prompt after issuing a command. The default value is 3.
Login Timeout (seconds)	**`login_timeout`**	Number of seconds to wait for a command prompt after login. The default value is 5.
Times to Retry Power On Operation	**`retry_on`**	Number of attempts to retry a power on operation. The default value is 1.

Figure 4.12, “HP iLO” shows the configuration screen for adding an HP iLO fence device.

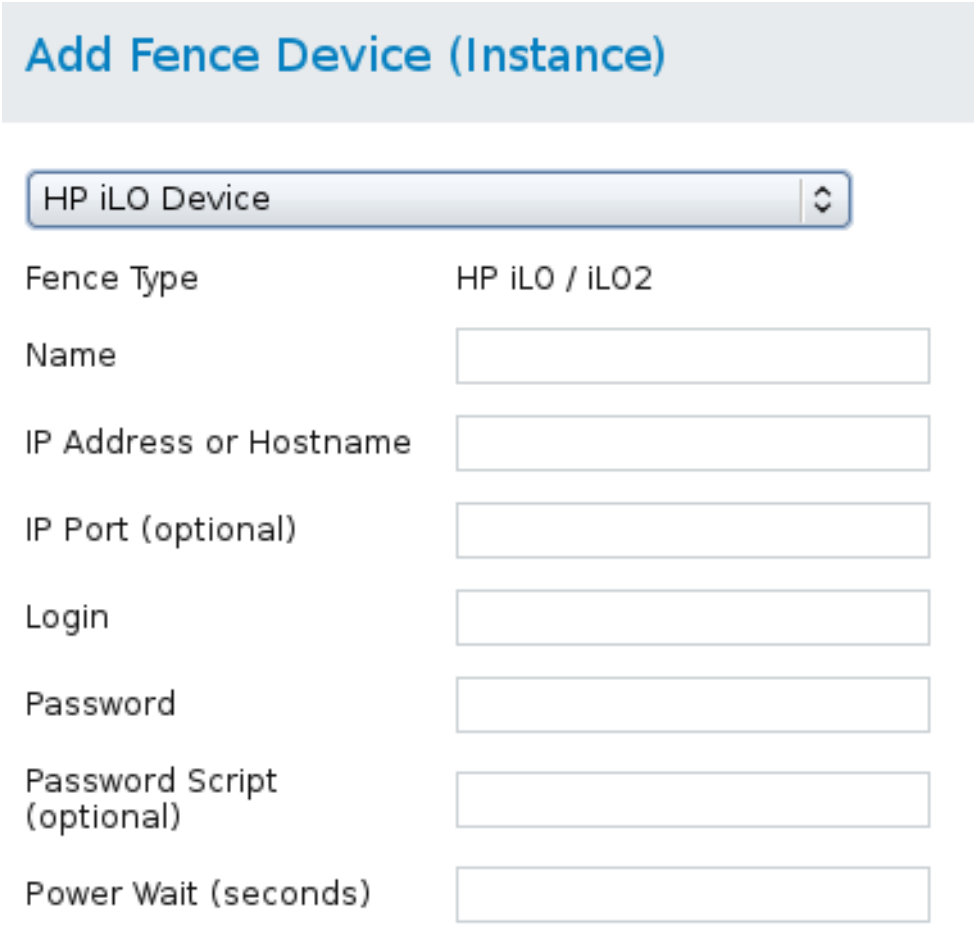


Figure 4.12. HP iLO

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>
```

4.16. HP ILO OVER SSH

The fence agents for HP iLO devices over SSH (**fence_ilo_ssh**), HP iLO3 devices over SSH (**fence_ilo3_ssh**), and HP iLO4 devices over SSH (**fence_ilo4_ssh**) share the same implementation. Table 4.17, “HP iLO over SSH, HP iLO3 over SSH, HPiLO4 over SSH (Red Hat Enterprise Linux 6.7 and later)” lists the fence device parameters used by these agents.

Table 4.17. HP iLO over SSH, HP iLO3 over SSH, HPiLO4 over SSH (Red Hat Enterprise Linux 6.7 and later)

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 host name 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.
Use SSH	**secure**	Indicates that the 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.
TCP Port	**ipport**	UDP/TCP port to use for connections with the device; the default value is 23.
Force Command Prompt	**cmd_prompt**	The command prompt to use. The default value is 'MP>', 'hpiLO->'.
Method to Fence	**method**	The method to fence: **on/off** or **cycle**
Power Wait (seconds)	**power_wait**	Number of seconds to wait after issuing a power off or power on command.
Delay (seconds)	**delay**	The number of seconds to wait before fencing is started. The default value is 0.
Power Timeout (seconds)	**power_timeout**	Number of seconds to continue testing for a status change after issuing a power off or power on command. The default value is 20.
Shell Timeout (seconds)	**shell_timeout**	Number of seconds to wait for a command prompt after issuing a command. The default value is 3.
Login Timeout (seconds)	**login_timeout**	Number of seconds to wait for a command prompt after login. The default value is 5.

luci Field	cluster.conf Attribute	Description
Times to Retry Power On Operation	**retry_on**	Number of attempts to retry a power on operation. The default value is 1.

4.17. HEWLETT-PACKARD ILO MP

Table 4.18, “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 4.18. 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 host name 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. When using SSH, you must specify either a password, a password script, or an identity file.
SSH Options	**ssh_options**	SSH options to use. The default value is **-1 -c blowfish**.
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 (seconds)	**power_wait**	Number of seconds to wait after issuing a power off or power on command.

luci Field	cluster.conf Attribute	Description
Delay (seconds)	**delay**	The number of seconds to wait before fencing is started. The default value is 0.
Power Timeout (seconds)	**power_timeout**	Number of seconds to continue testing for a status change after issuing a power off or power on command. The default value is 20.
Shell Timeout (seconds)	**shell_timeout**	Number of seconds to wait for a command prompt after issuing a command. The default value is 3.
Login Timeout (seconds)	**login_timeout**	Number of seconds to wait for a command prompt after login. The default value is 5.
Times to Retry Power On Operation	**retry_on**	Number of attempts to retry a power on operation. The default value is 1.

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

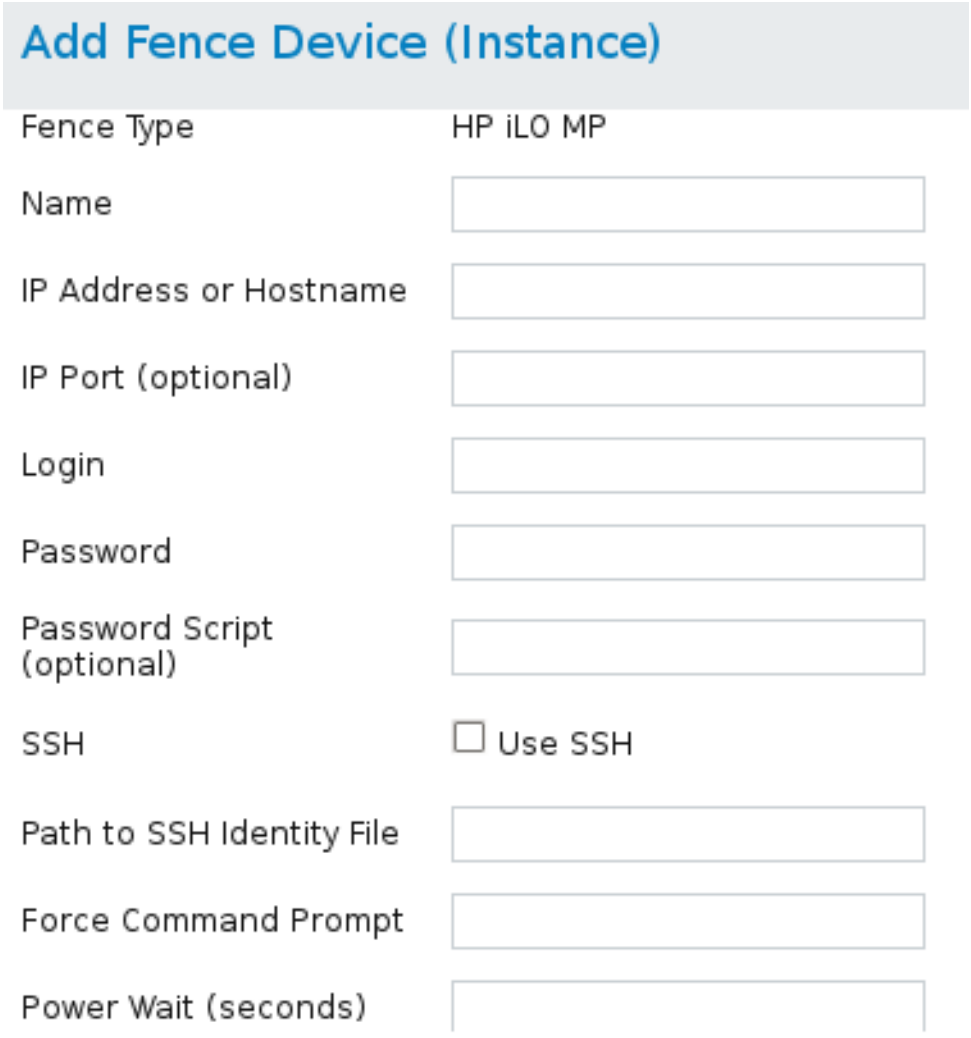


Figure 4.13. HP iLO MP

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-&gt;"
ipaddr="192.168.0.1" login="root" name="hpilomptest1" passwd="password123"
power_wait="60"/>
 </fencedevices>
```

4.18. HP MOONSHOT ILO

Table 4.19, “HP Moonshot iLO (Red Hat Enterprise Linux 6.7 and later)” lists the fence device parameters used by **fence_ilo_moonshot**, the fence agent for HP Moonshot iLO devices.

Table 4.19. HP Moonshot iLO (Red Hat Enterprise Linux 6.7 and later)

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 host name 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.
Use SSH	**secure**	Indicates that the 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.
TCP Port	**ipport**	UDP/TCP port to use for connections with the device; the default value is 22.

luci Field	cluster.conf Attribute	Description
Force Command Prompt	cmd_prompt	The command prompt to use. The default value is 'MP>', 'hpiLO->'.
Power Wait (seconds)	power_wait	Number of seconds to wait after issuing a power off or power on command.
Delay (seconds)	delay	The number of seconds to wait before fencing is started. The default value is 0.
Power Timeout (seconds)	power_timeout	Number of seconds to continue testing for a status change after issuing a power off or power on command. The default value is 20.
Shell Timeout (seconds)	shell_timeout	Number of seconds to wait for a command prompt after issuing a command. The default value is 3.
Login Timeout (seconds)	login_timeout	Number of seconds to wait for a command prompt after login. The default value is 5.
Times to Retry Power On Operation	retry_on	Number of attempts to retry a power on operation. The default value is 1.

4.19. IBM BLADECENTER

Table 4.20, “IBM BladeCenter” lists the fence device parameters used by **fence_bladecenter**, the fence agent for IBM BladeCenter.

Table 4.20. 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 host name 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.

luci Field	cluster.conf Attribute	Description
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 (seconds)	**power_wait**	Number of seconds to wait after issuing a power off or power on command.
Power Timeout (seconds)	**power_timeout**	Number of seconds to continue testing for a status change after issuing a power off or power on command. The default value is 20.
Shell Timeout (seconds)	**shell_timeout**	Number of seconds to wait for a command prompt after issuing a command. The default value is 3.
Login Timeout (seconds)	**login_timeout**	Number of seconds to wait for a command prompt after login. The default value is 5.
Times to Retry Power On Operation	**retry_on**	Number of attempts to retry a power on operation. The default value is 1.
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.
SSH Options	**ssh_options**	SSH options to use. The default value is **-1 -c blowfish**.
Path to SSH Identity File	**identity_file**	The identity file for SSH.

Figure 4.14, “IBM BladeCenter” shows the configuration screen for adding an IBM BladeCenter fence device.

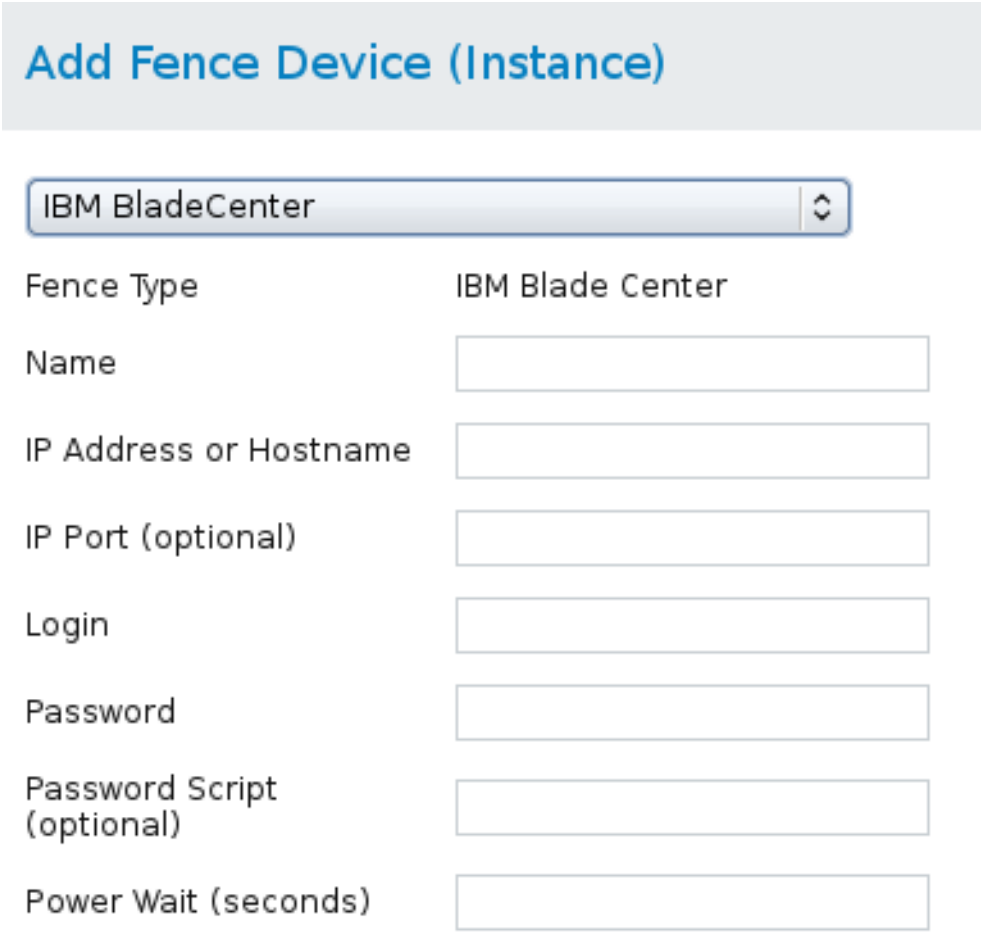


Figure 4.14. IBM BladeCenter

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>
```

4.20. IBM BLADECENTER OVER SNMP

Table 4.21, "IBM BladeCenter SNMP" lists the fence device parameters used by **fence_ibmblade**, the fence agent for IBM BladeCenter over SNMP.

Table 4.21. 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 host name 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_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_passwd**	The SNMP Privacy Protocol Password.
SNMP Privacy Protocol Script	**snmp_priv_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.
Power Timeout (seconds)	**power_timeout**	Number of seconds to continue testing for a status change after issuing a power off or power on command. The default value is 20.

luci Field	cluster.conf Attribute	Description
Shell Timeout (seconds)	shell_timeout	Number of seconds to wait for a command prompt after issuing a command. The default value is 3.
Login Timeout (seconds)	login_timeout	Number of seconds to wait for a command prompt after login. The default value is 5.
Times to Retry Power On Operation	retry_on	Number of attempts to retry a power on operation. The default value is 1.
Port (Outlet) Number	port	Physical plug number or name of virtual machine.
Delay (optional)	delay	The number of seconds to wait before fencing is started. The default value is 0.

Figure 4.15, “IBM BladeCenter SNMP” shows the configuration screen for adding an IBM BladeCenter SNMP fence device.

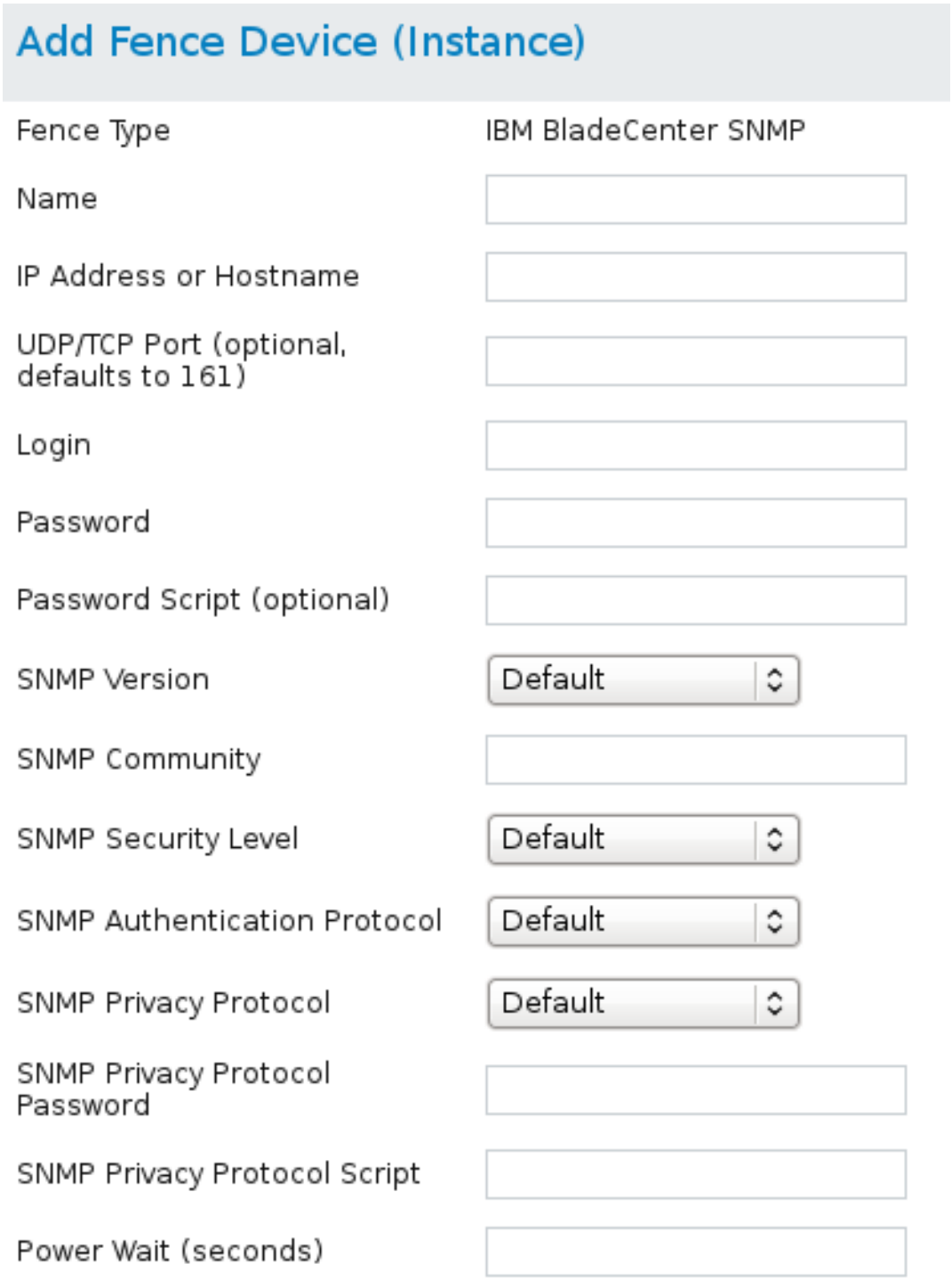


Figure 4.15. IBM BladeCenter SNMP

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>
```

4.21. IBM IPDU

Table 4.22, "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 4.22. 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 by means of the SNMP protocol.
IP Address or Hostname	**ipaddr**	The IP address or host name 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_passwd**	The SNMP privacy protocol password.

luci Field	cluster.conf Attribute	Description
SNMP Privacy Protocol Script	snmp_priv_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.
Power Timeout (seconds)	power_timeout	Number of seconds to continue testing for a status change after issuing a power off or power on command. The default value is 20.
Shell Timeout (seconds)	shell_timeout	Number of seconds to wait for a command prompt after issuing a command. The default value is 3.
Login Timeout (seconds)	login_timeout	Number of seconds to wait for a command prompt after login. The default value is 5.
Times to Retry Power On Operation	retry_on	Number of attempts to retry a power on operation. The default value is 1.
Port (Outlet) Number	port	Physical plug number or name of virtual machine.
Delay (optional)	delay	The number of seconds to wait before fencing is started. The default value is 0.

Figure 4.16, “IBM iPDU” shows the configuration screen for adding an IBM iPDU fence device.

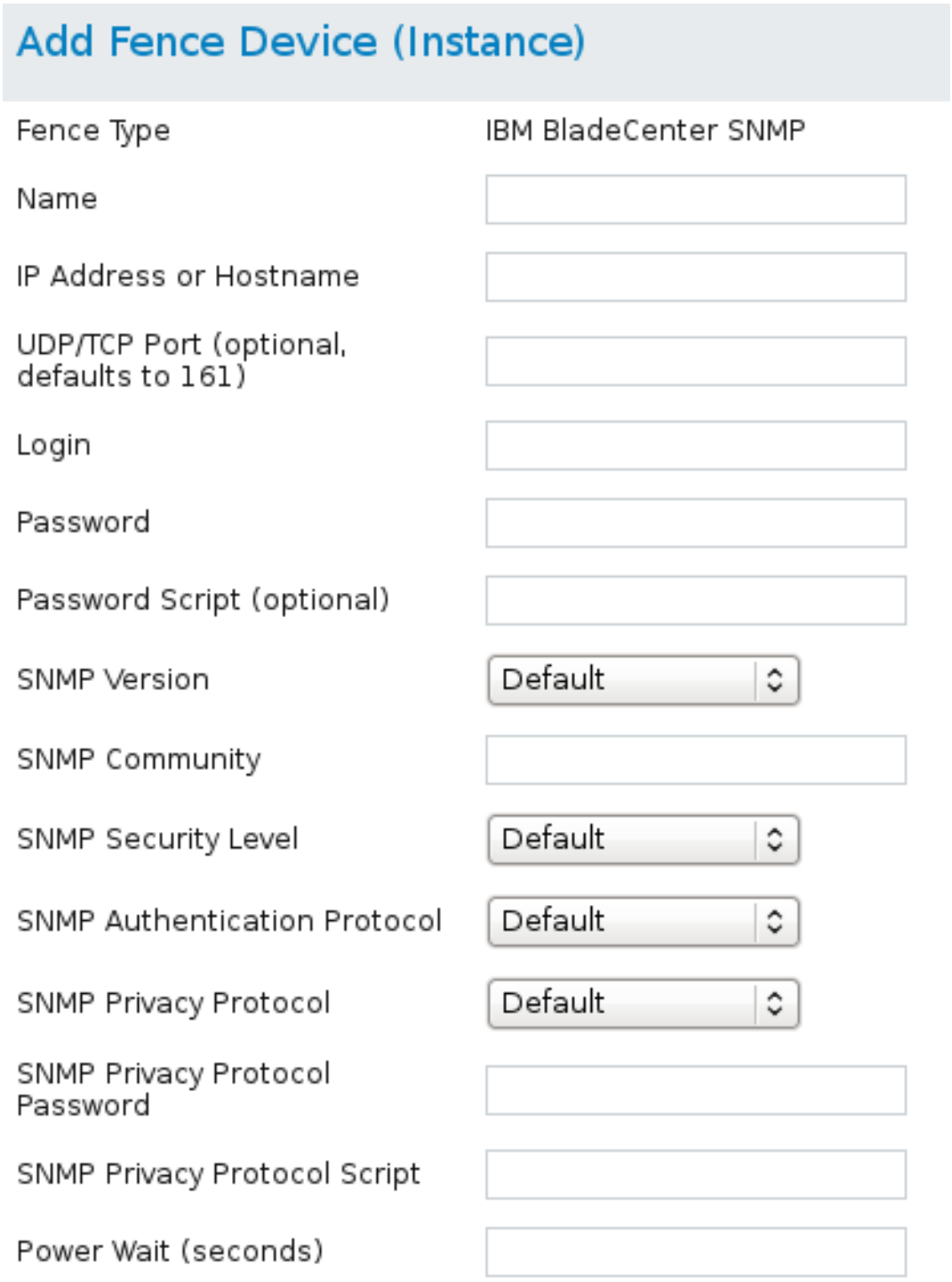


Figure 4.16. IBM iPDU

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>
```

4.22. IF-MIB

Table 4.23, “IF MIB” lists the fence device parameters used by **fence_ifmib**, the fence agent for IF-MIB devices.

Table 4.23. 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 host name 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_prot**	The SNMP authentication protocol (MD5, SHA).
SNMP Privacy Protocol	**snmp_priv_prot**	The SNMP privacy protocol (DES, AES).

luci Field	cluster.conf Attribute	Description
SNMP Privacy Protocol Password	snmp_priv_passwd	The SNMP privacy protocol password.
SNMP Privacy Protocol Script	snmp_priv_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.
Power Timeout (seconds)	power_timeout	Number of seconds to continue testing for a status change after issuing a power off or power on command. The default value is 20.
Shell Timeout (seconds)	shell_timeout	Number of seconds to wait for a command prompt after issuing a command. The default value is 3.
Login Timeout (seconds)	login_timeout	Number of seconds to wait for a command prompt after login. The default value is 5.
Times to Retry Power On Operation	retry_on	Number of attempts to retry a power on operation. The default value is 1.
Port (Outlet) Number	port	Physical plug number or name of virtual machine.
Delay (optional)	delay	The number of seconds to wait before fencing is started. The default value is 0.

Figure 4.17, “IF-MIB” shows the configuration screen for adding an IF-MIB fence device.

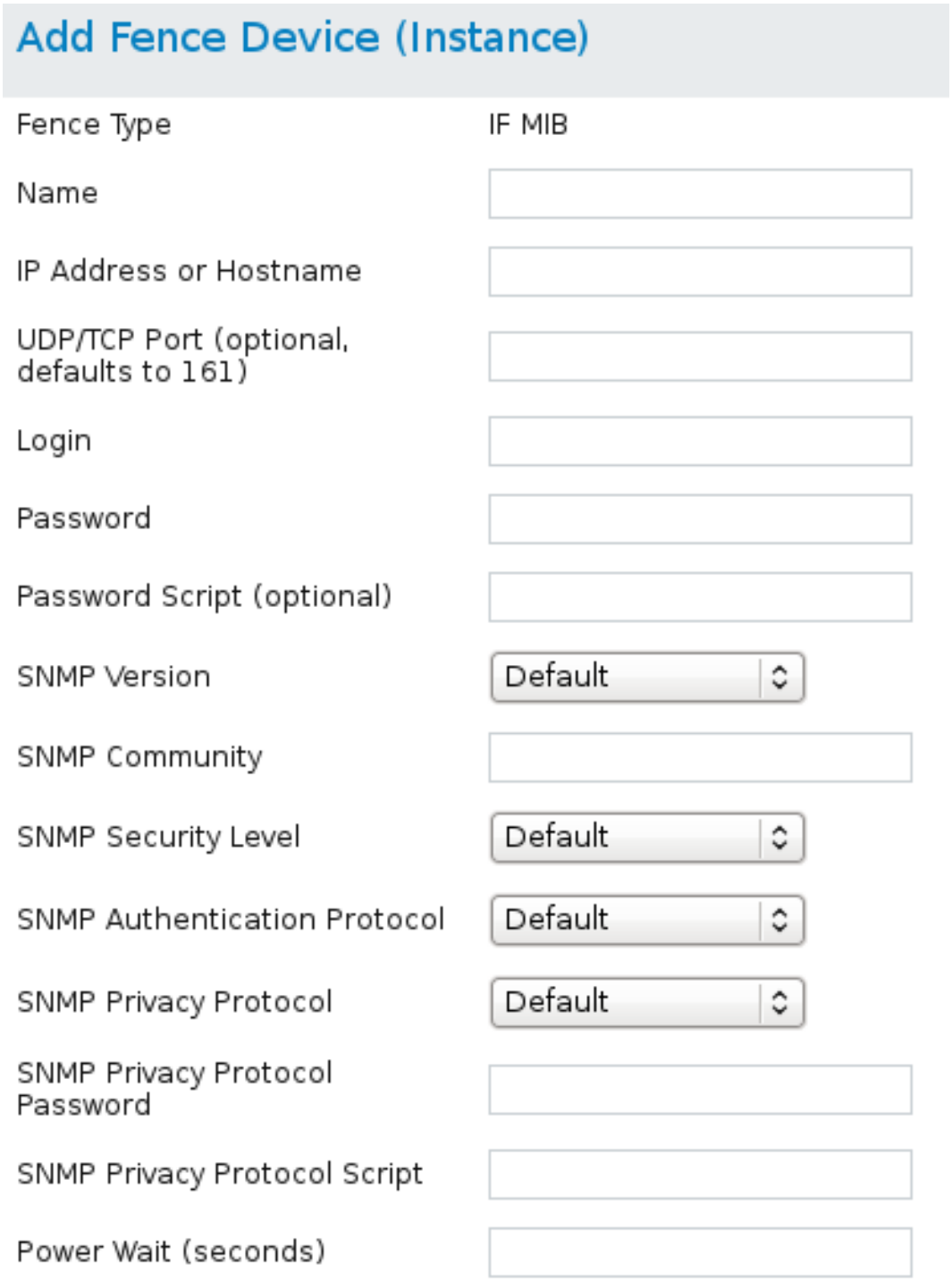


Figure 4.17. IF-MIB

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>
```

4.23. INTEL MODULAR

Table 4.24, “Intel Modular” lists the fence device parameters used by **fence_intelmodular**, the fence agent for Intel Modular.

Table 4.24. 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 host name 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_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_passwd**	The SNMP privacy protocol password.

luci Field	cluster.conf Attribute	Description
SNMP Privacy Protocol Script	snmp_priv_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.
Power Timeout (seconds)	power_timeout	Number of seconds to continue testing for a status change after issuing a power off or power on command. The default value is 20.
Shell Timeout (seconds)	shell_timeout	Number of seconds to wait for a command prompt after issuing a command. The default value is 3.
Login Timeout (seconds)	login_timeout	Number of seconds to wait for a command prompt after login. The default value is 5.
Times to Retry Power On Operation	retry_on	Number of attempts to retry a power on operation. The default value is 1.
Port (Outlet) Number	port	Physical plug number or name of virtual machine.
Delay (optional)	delay	The number of seconds to wait before fencing is started. The default value is 0.

Figure 4.18, “Intel Modular” shows the configuration screen for adding an Intel Modular fence device.

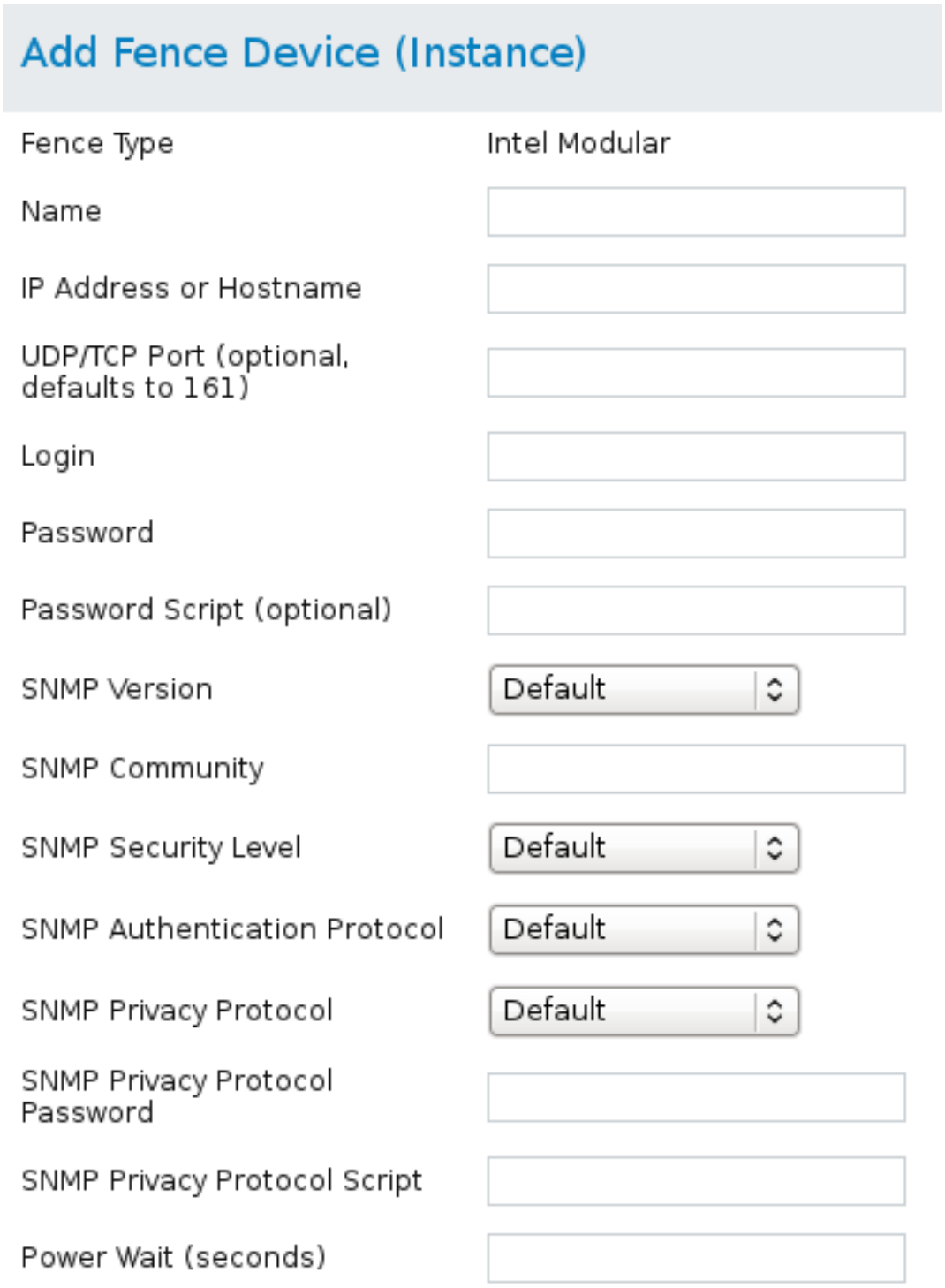


Figure 4.18. Intel Modular

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>
```

4.24. IPMI OVER LAN

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. Table 4.25, "IPMI (Intelligent Platform Management Interface) LAN, Dell iDrac, IBM Integrated Management Module, HPiLO3, HPiLO4" lists the fence device parameters used by these agents.

Table 4.25. IPMI (Intelligent Platform Management Interface) LAN, Dell iDrac, IBM Integrated Management Module, HPiLO3, HPiLO4

luci Field	cluster.conf Attribute	Description
Name	**name**	A name for the fence device connected to the cluster.
IP Address or Hostname	**ipaddr**	The IP address or host name assigned to the device.
Login	**login**	The login name of a user capable of issuing power on/off commands to the given port.
Password	**passwd**	The password used to authenticate the connection to the 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**	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 device.
IPMI Operation Timeout	**timeout**	Timeout in seconds for IPMI operation.
Power Timeout (seconds)	**power_timeout**	Number of seconds to continue testing for a status change after issuing a power off or power on command. The default value is 20.

luci Field	cluster.conf Attribute	Description
Shell Timeout (seconds)	**shell_timeout**	Number of seconds to wait for a command prompt after issuing a command. The default value is 3.
Login Timeout (seconds)	**login_timeout**	Number of seconds to wait for a command prompt after login. The default value is 5.
Times to Retry Power On Operation	**retry_on**	Number of attempts to retry a power on operation. The default value is 1.
Power Wait (seconds)	**power_wait**	Number of seconds to wait after issuing a power off or power on command. The default value is 2 seconds for **fence_ipmilan**, **fence_idrac**, **fence_imm**, and **fence_ilo4**. The default value is 4 seconds for **fence_ilo3**.
Delay (optional)	**delay**	The number of seconds to wait before fencing is started. The default value is 0.
Method to Fence	**method**	The method to fence: **on/off** or **cycle**

Figure 4.19, “IPMI over LAN” shows the configuration screen for adding an IPMI over LAN device

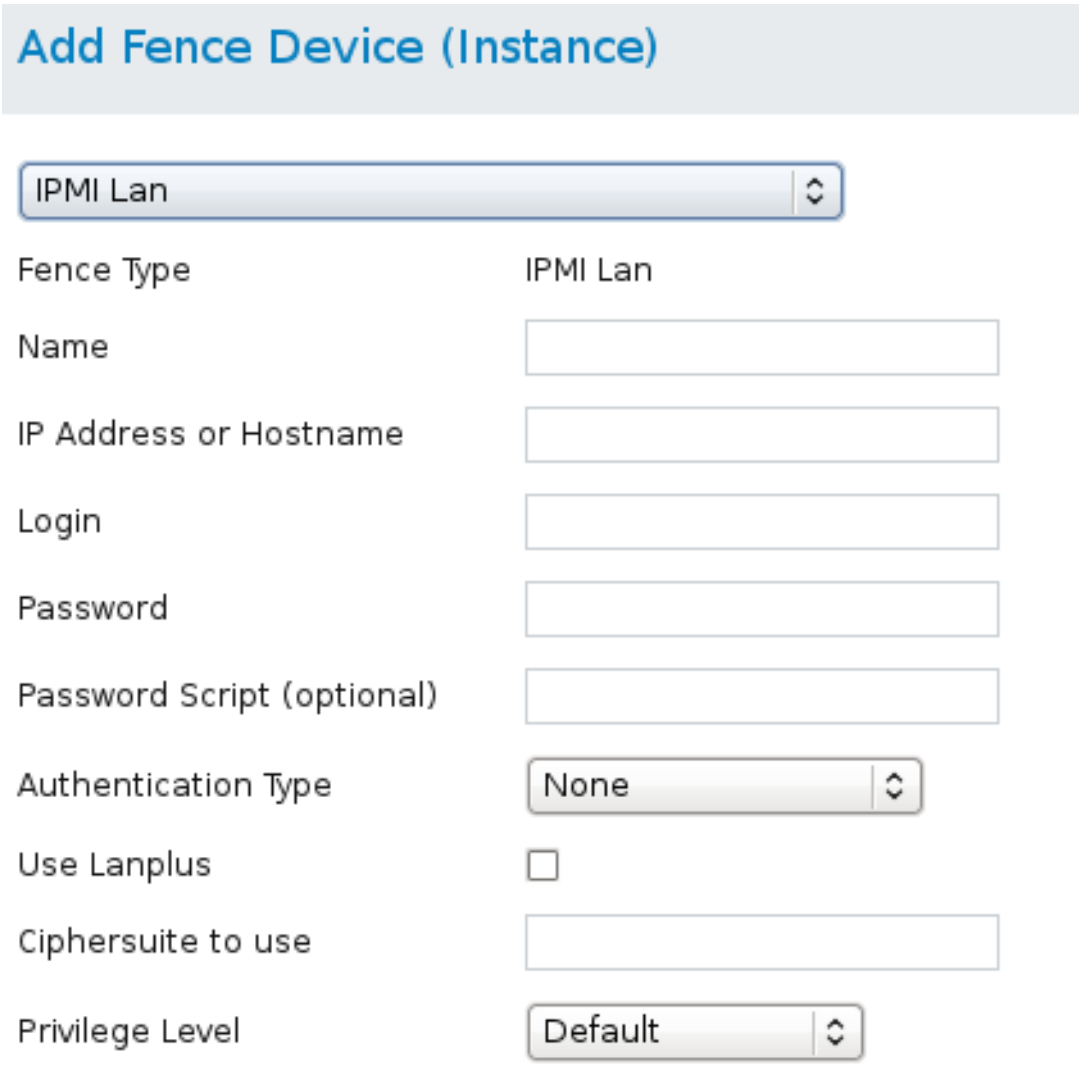


Figure 4.19. IPMI over LAN

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>
```

4.25. FENCE KDUMP

Table 4.26, “Fence kdump” lists the fence device parameters used by **fence_dkump**, the fence agent for **kdump** crash recovery service. Note that **fence_kdump** is not a replacement for traditional fencing

methods; The **fence_kdump** agent can detect only that a node has entered the **kdump** crash recovery service. This allows the **kdump** crash recovery service to complete without being preempted by traditional power fencing methods.

Table 4.26. Fence kdump

luci Field	cluster.conf Attribute	Description
Name	**name**	A name for the **fence_kdump** device.
IP Family	**family**	IP network family. The default value is **auto**.
IP Port (optional)	**ipport**	IP port number that the **fence_kdump** agent will use to listen for messages. The default value is 7410.
Operation Timeout (seconds) (optional)	**timeout**	Number of seconds to wait for message from failed node.
Node name	**nodename**	Name or IP address of the node to be fenced.

4.26. MULTIPATH PERSISTENT RESERVATION FENCING (RED HAT ENTERPRISE LINUX 6.7 AND LATER)

Table 4.27, "Multipath Persistent Reservation Fencing (Red Hat Enterprise Linux 6.7 and later)" lists the fence device parameters used by **fence_mpath**, the fence agent for multipath persistent reservation fencing.

Table 4.27. Multipath Persistent Reservation Fencing (Red Hat Enterprise Linux 6.7 and later)

luci Field	cluster.conf Attribute	Description
Name	**name**	A name for the **fence_mpath** device.
Devices (Comma delimited list)	**devices**	Comma-separated list of devices to use for the current operation. Each device must support SCSI-3 persistent reservations.
Use sudo when calling third-party software	**sudo**	Use sudo (without password) when calling 3rd party software.
Path to sudo binary (optional)	**sudo_path**	Path to sudo binary (default value is **/usr/bin/sudo**.

luci Field	cluster.conf Attribute	Description
Path to mpathpersist binary (optional)	**mpathpersist_path**	Path to mpathpersist binary (default value is **/sbin/mpathpersist**.
Path to a directory where the fence agent can store information (optional)	**store_path**	Path to directory where fence agent can store information (default value is **/var/run/cluster**.
Power Wait (seconds)	**power_wait**	Number of seconds to wait after issuing a power off or power on command.
Power Timeout (seconds)	**power_timeout**	Number of seconds to continue testing for a status change after issuing a power off or power on command. The default value is 20.
Shell Timeout (seconds)	**shell_timeout**	Number of seconds to wait for a command prompt after issuing a command. The default value is 3.
Login Timeout (seconds)	**login_timeout**	Number of seconds to wait for a command prompt after login. The default value is 5.
Times to Retry Power On Operation	**retry_on**	Number of attempts to retry a power on operation. The default value is 1.
Unfencing	**unfence** section of the cluster configuration file	When enabled, this ensures that a fenced node is not re-enabled until the node has been rebooted. This is necessary for non-power fence methods. When you configure a device that requires unfencing, the cluster must first be stopped and the full configuration including devices and unfencing must be added before the cluster is started. For more information about unfencing a node, see the **fence_node**(8) man page.
Key for current action	**key**	Key to use for the current operation. This key should be unique to a node and written in **/etc/multipath.conf**. For the "on" action, the key specifies the key use to register the local node. For the "off" action, this key specifies the key to be removed from the device(s). This parameter is always required.
Delay (optional)	**delay**	The number of seconds to wait before fencing is started. The default value is 0.

4.27. RHEV-M REST API

Table 4.28, “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 4.28. 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 host name 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 (seconds)	**power_wait**	Number of seconds to wait after issuing a power off or power on command.
Power Timeout (seconds)	**power_timeout**	Number of seconds to continue testing for a status change after issuing a power off or power on command. The default value is 20.
Shell Timeout (seconds)	**shell_timeout**	Number of seconds to wait for a command prompt after issuing a command. The default value is 3.
Login Timeout (seconds)	**login_timeout**	Number of seconds to wait for a command prompt after login. The default value is 5.
Times to Retry Power On Operation	**retry_on**	Number of attempts to retry a power on operation. The default value is 1.
Port (Outlet) Number	**port**	Physical plug number or name of virtual machine.
Delay (optional)	**delay**	The number of seconds to wait before fencing is started. The default value is 0.

Figure 4.20, “RHEV-M REST API” shows the configuration screen for adding an RHEV-M REST API device

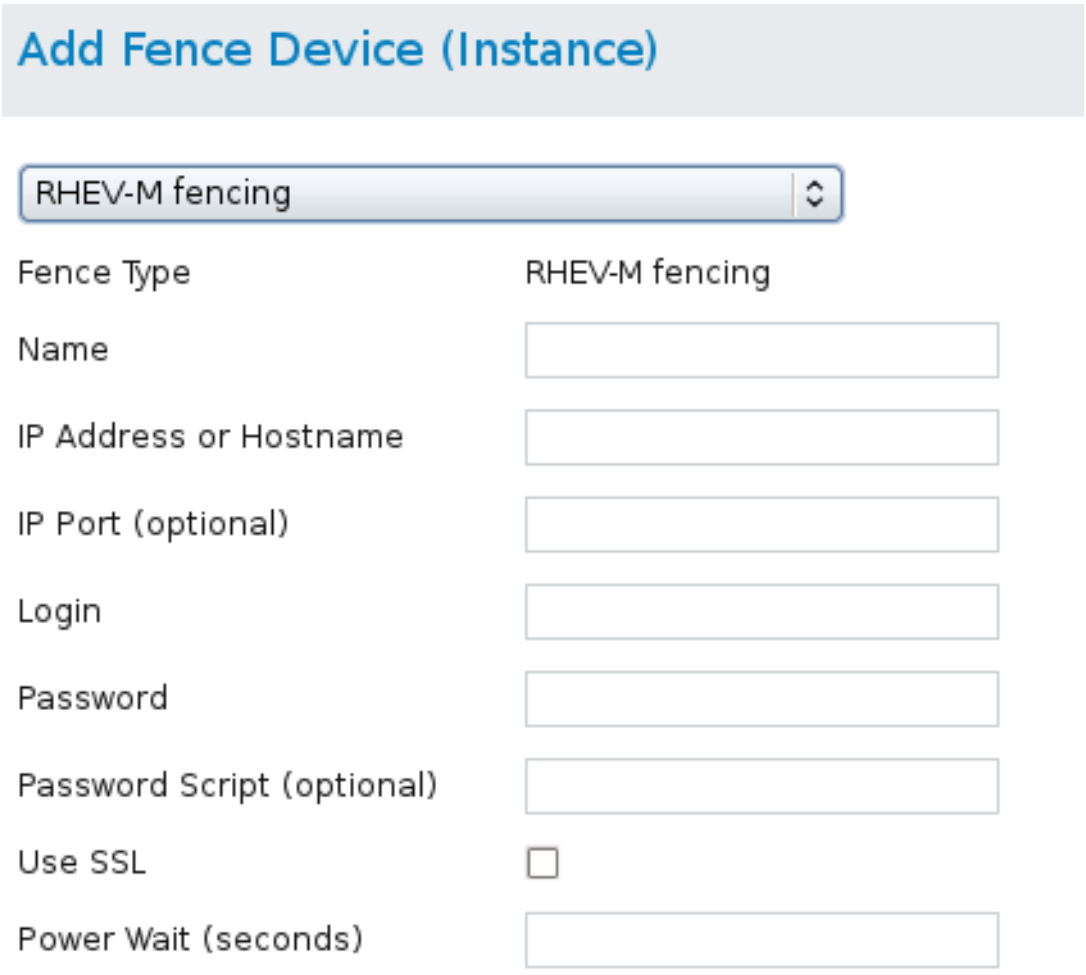


Figure 4.20. RHEV-M REST API

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>
```

4.28. SCSI PERSISTENT RESERVATIONS

Table 4.29, “SCSI Reservation 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.

It is recommended that devices used for the cluster volumes be specified in the format **/dev/disk/by-id/***xxx* where possible. Devices specified in this format are consistent among all nodes and will point to the same disk, unlike devices specified in a format such as **/dev/sda** which can point to different disks from machine to machine and across reboots.

Table 4.29. SCSI Reservation Fencing

luci Field	cluster.conf Attribute	Description
Name	**name**	A name for the SCSI fence device.
Unfencing	**unfence** section of the cluster configuration file	When enabled, this ensures that a fenced node is not re-enabled until the node has been rebooted. This is necessary for non-power fence methods (that is, SAN/storage fencing). When you configure a device that requires unfencing, the cluster must first be stopped and the full configuration including devices and unfencing must be added before the cluster is started. For more information about unfencing a node, see the **fence_node**(8) man page.
Node name	**nodename**	The node name is used to generate the key value used for the current operation.
Key for current action	**key**	(overrides node name) Key to use for the current operation. This key should be unique to a node. For the "on" action, the key specifies the key use to register the local node. For the "off" action,this key specifies the key to be removed from the device(s).
Delay (optional)	**delay**	The number of seconds to wait before fencing is started. The default value is 0.

Figure 4.21, “SCSI Fencing” shows the configuration screen for adding an SCSI fence device

Figure 4.21. 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>
```

4.29. VMWARE OVER SOAP API

Table 4.30, “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 4.30. 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 host name assigned to the device.
IP Port (optional)	**ipport**	The TCP port to use for connection with the device. The default port is 80, or 443 if **Use SSL** is selected.
Login	**login**	The login name used to access the device.
Password	**passwd**	The password used to authenticate the connection to the device.

luci Field	cluster.conf Attribute	Description
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 (seconds)	**power_wait**	Number of seconds to wait after issuing a power off or power on command.
Power Timeout (seconds)	**power_timeout**	Number of seconds to continue testing for a status change after issuing a power off or power on command. The default value is 20.
Shell Timeout (seconds)	**shell_timeout**	Number of seconds to wait for a command prompt after issuing a command. The default value is 3.
Login Timeout (seconds)	**login_timeout**	Number of seconds to wait for a command prompt after login. The default value is 5.
Times to Retry Power On Operation	**retry_on**	Number of attempts to retry a power on operation. The default value is 1.
VM name	**port**	Name of virtual machine in inventory path format (for example, /datacenter/vm/Discovered_virtual_machine/myMachine).
VM UUID	**uuid**	The UUID of the virtual machine to fence.
Delay (optional)	**delay**	The number of seconds to wait before fencing is started. The default value is 0.
Use SSL	**ssl**	Use SSL connections to communicate with the device.

Figure 4.22, “VMware over SOAP Fencing” shows the configuration screen for adding a VMware over SOAP fence device

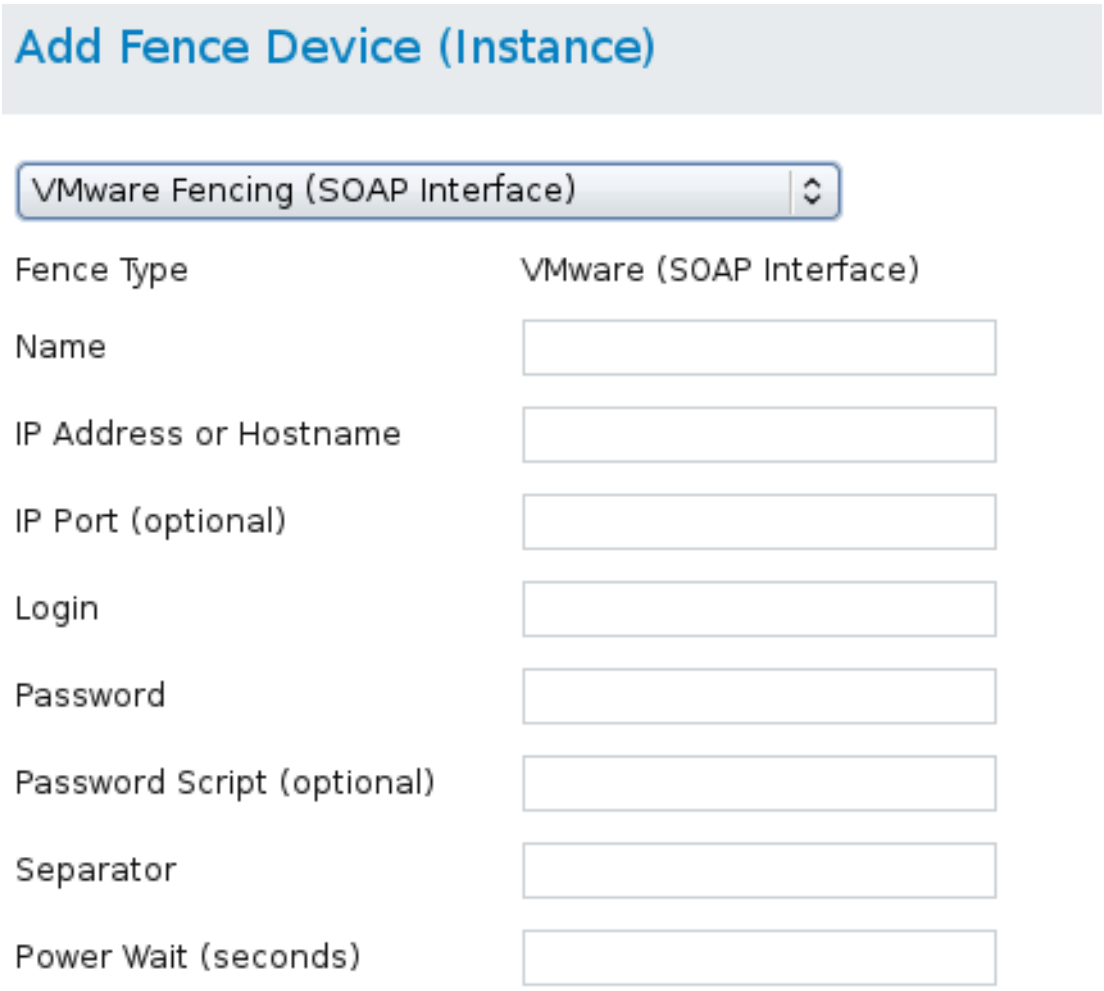


Figure 4.22. VMware over SOAP Fencing

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>
```

4.30. WTI POWER SWITCH

Table 4.31, “WTI Power Switch” lists the fence device parameters used by **fence_wti**, the fence agent for the WTI network power switch.

Table 4.31. 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 address or host name 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.
Force command prompt	**cmd_prompt**	The command prompt to use. The default value is ['RSM>', '>MPC', 'IPS>', 'TPS>', 'NBB>', 'NPS>', 'VMR>']
Power Wait (seconds)	**power_wait**	Number of seconds to wait after issuing a power off or power on command.
Power Timeout (seconds)	**power_timeout**	Number of seconds to continue testing for a status change after issuing a power off or power on command. The default value is 20.
Shell Timeout (seconds)	**shell_timeout**	Number of seconds to wait for a command prompt after issuing a command. The default value is 3.
Login Timeout (seconds)	**login_timeout**	Number of seconds to wait for a command prompt after login. The default value is 5.
Times to Retry Power On Operation	**retry_on**	Number of attempts to retry a power on operation. The default value is 1.
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.
SSH Options	**ssh_options**	SSH options to use. The default value is **-1 -c blowfish**.
Path to SSH Identity File	**identity_file**	The identity file for SSH.

luci Field	cluster.conf Attribute	Description
Port	port	Physical plug number or name of virtual machine.

Figure 4.23, “WTI Fencing” shows the configuration screen for adding a WTI fence device

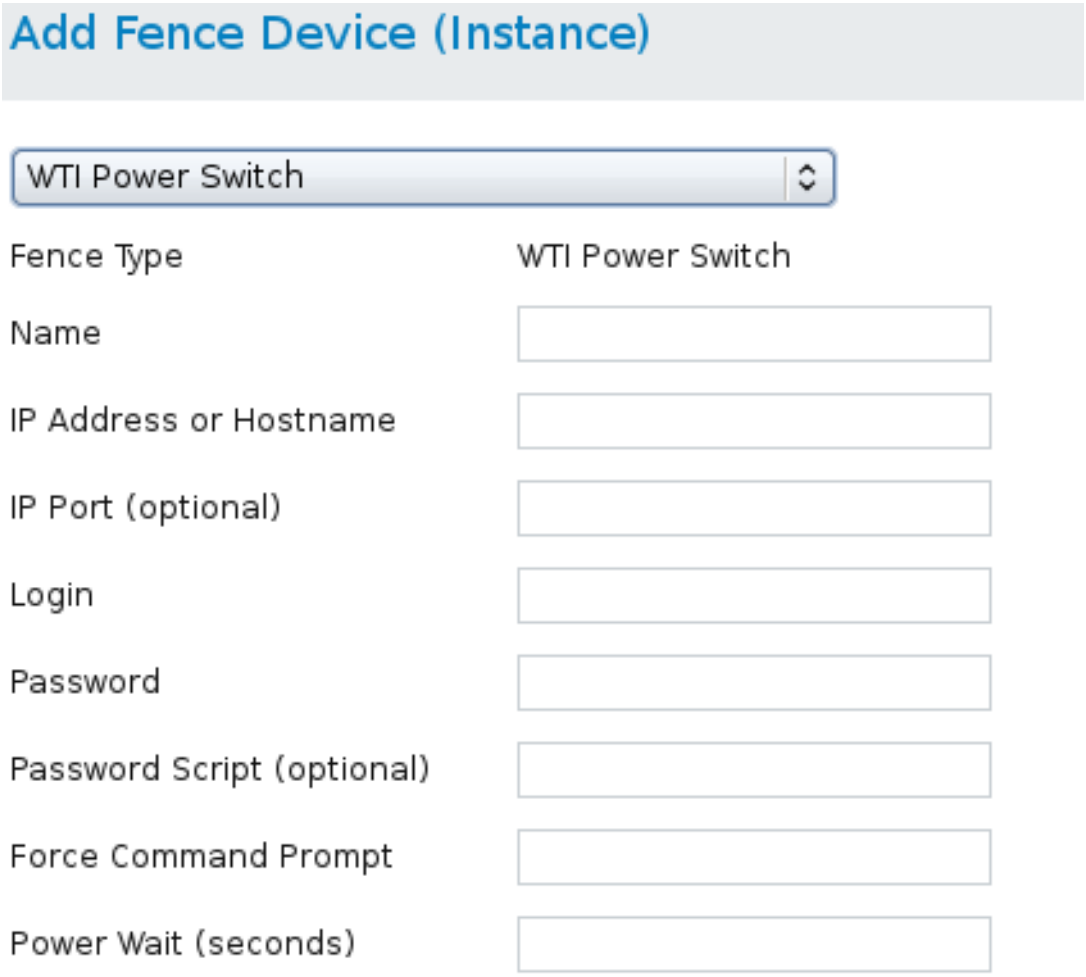


Figure 4.23. WTI Fencing

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

```
ccs -f cluster.conf --addfencedev wtipwrsw1 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&gt;" ipaddr="192.168.0.1"
login="root" name="wtipwrsw1" \
   passwd="password123" power_wait="60"/>
 </fencedevices>
```

APPENDIX A. REVISION HISTORY

Revision 4-2 **Wed Mar 8 2017** **Steven Levine**
Version for 6.9 GA publication.

Revision 4-1 **Fri Dec 16 2016** **Steven Levine**
Version for 6.9 Beta publication.

Revision 3-5 **Wed Apr 27 2016** **Steven Levine**
Preparing document for 6.8 GA publication.

Revision 3-4 **Wed Mar 9 2016** **Steven Levine**
Initial revision for Red Hat Enterprise Linux 6.8 Beta release

Revision 2-2 **Wed Jul 8 2015** **Steven Levine**
Initial revision for Red Hat Enterprise Linux 6.7

Revision 2-1 **Thu Apr 16 2015** **Steven Levine**
Initial revision for Red Hat Enterprise Linux 6.7 Beta release

Revision 1-13 **Wed Oct 8 2014** **Steven Levine**
Initial revision for Red Hat Enterprise Linux 6.6

Revision 1-11 **Thu Aug 7 2014** **Steven Levine**
Initial revision for Red Hat Enterprise Linux 6.6 Beta release

Revision 1-9 **Wed Nov 20 2013** **John Ha**
Initial revision for Red Hat Enterprise Linux 6.5

Revision 1-4 **Mon Nov 28 2012** **John Ha**
Initial revision for Red Hat Enterprise Linux 6.5 Beta release

Revision 1-2 **Mon Nov 28 2012** **John Ha**
Initial revision for Red Hat Enterprise Linux 6.4 Beta release

INDEX

H