Red Hat Enterprise Linux 6

フェンス設定ガイド

High Availability Add-On 向けフェンスデバイスの設定と管理

Last Updated: 2017-10-12

Red Hat Enterprise Linux 6 フェンス設定ガイド

High Availability Add-On 向けフェンスデバイスの設定と管理

.

法律上の通知

概要

フェンシングとは、 クラスターの共有ストレージから任意のノードを切断することを指します。 フェンシングによって共有ストレージからの I/O を遮断することでデータの整合性を確保します。 本ガイドでは、 High Availability Add-On を使用してクラスター化したシステム群でフェンシングを設定する方法を説明していきます。 また、 対応するフェンスデバイスの設定についても記載しています。

目次

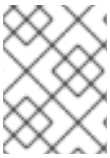
第1章 フェンシング設定の前に行なうべき準備

本章では、 Red Hat High Availability Add-On を使用したクラスターへのフェンシング導入を行なう前に実行すべき作業および注意事項について述べておきます。 以下のセクションで構成されています。

- 「統合フェンスデバイスで使用するための ACPI の設定」

1.1. 統合フェンスデバイスで使用するための ACPI の設定

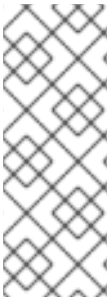
クラスターで統合フェンスデバイスを使用している場合には、 フェンシングが直ちにまた完全に機能するよう ACPI (Advanced Configuration and Power Interface) の設定を行なう必要があります。



注記

Red Hat High Availability Add-On で対応している統合フェンスデバイスの最新情報については、 http://www.redhat.com/cluster_suite/hardware/ をご覧ください。

クラスターノードを統合フェンスデバイスで隔離するよう設定している場合は、そのノードの ACPI Soft-Off を無効にします。ACPI Soft-Off を無効にすると、明示的なシャットダウン (**shutdown -h now** など) をしなくても統合フェンスデバイス側でノードを直ちに完全電源オフにすることができます。これに対して、ACPI Soft-Off が有効な場合、統合フェンスデバイスによるノードの電源オフに 4 秒以上かかることがあります (以下の注記参照)。さらに、ACPI Soft-Off が有効な状態でシャットダウン中にノードがパニックやフリーズを起こすと、統合フェンスデバイスがノードの電源を切れなくなる場合があります。このような場合、フェンシングの動作が遅れる、または失敗することになります。結果、統合フェンスデバイスを使ってノードのフェンシングを行うよう設定する一方、ACPI Soft-Off を有効にしておくと、クラスターの復帰に時間がかかったり、管理者の介入が必要になります。

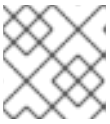


注記

ノードのフェンシングにかかる時間は使用する統合フェンスデバイスによって異なります。 電源ボタンを押し続けて電源を切るのと同じくらいの時間となる統合フェンスデバイスもあれば (統合フェンスデバイスによるノードの電源オフにかかる時間が4 秒ないしは 5 秒以内)、 電源ボタンを一度押してあとはオペレーティングシステムに任せて電源を切るのと同じくらいの時間となる統合フェンスデバイスもあります (統合フェンスデバイスによるノードの電源オフが前述のような 4、 5 秒に比べかなり時間がかかる)。

ACPI Soft-Off を無効にする場合、 **chkconfig** 管理を使用して行い、 フェンシングの際にノードが直ちに電源オフになることを確認します。 ACPI Soft-Off は **chkconfig** 管理を使用して無効にする方法をお勧めします。 ただし、 この方法がクラスターに適していない場合には次のような代替方法を使って ACPI Soft-Off を無効にすることもできます。

- BIOS の設定を "instant-off" または「遅延なくノードの電源をオフにする」に相当する設定に変更する

  注記

  BIOS では ACPI Soft-Off を無効にできないコンピュータもあります。

- **/boot/grub/grub.conf** ファイルのカーネルブートコマンドラインに **acpi=off** を追加する

重要

この方法を使用すると ACPI が完全に無効になります。 ACPI が完全に無効になっていると正しく起動しないコンピュータがあります。 この方法を使用するのは **他の方法がクラスターに適さない場合に限ってください** 。

ACPI Soft-Off を無効にする方法として推奨している方法、 およびその代替となる方法について次のセクションで説明していきます。

- 「**chkconfig** 管理を使って ACPI Soft-Off を無効にする」 — 推奨している方法
- 「BIOS を使って ACPI Soft-Off を無効にする」 — 代替方法 1
- 「**grub.conf** ファイル内で ACPI を完全に無効にする」 — 代替方法 2

1.1.1. chkconfig 管理を使って ACPI Soft-Off を無効にする

chkconfig 管理を使って ACPI Soft-Off を無効にする場合は、 ACPI デーモン (**acpid**) を **chkconfig** 管理から削除するか、 **acpid** をオフにします。

注記

この方法が ACPI Soft-Off を無効にする方法として推奨している方法になります。

以下のようにして、 各クラスターノードで **chkconfig** を使い ACPI Soft-Off を無効にします。

1. 次のいずれかのコマンドを実行します。
   - **chkconfig --del acpid** — このコマンドにより **acpid** が **chkconfig** 管理から削除されます。

     — または —

   - **chkconfig --level 2345 acpid off** — このコマンドにより **acpid** がオフになります。
2. ノードを再起動します。
3. クラスターを設定して実行を開始したら、 フェンシングの際にノードの電源が直ちにオフになることを確認します。

注記

ノードのフェンシングは **fence_node** コマンドや **Conga** を使っても行なうことができます。

1.1.2. BIOS を使って ACPI Soft-Off を無効にする

ACPI Soft-Off を無効にする場合に推奨している方法は **chkconfig** 管理を使用する方法です (「**chkconfig** 管理を使って ACPI Soft-Off を無効にする」)。 ただし、 推奨している方法がクラスターに適していない場合には、 このセクションの手順に従ってください。

注記

BIOS では ACPI Soft-Off を無効にできないコンピュータもあります。

以下のようにして、 各クラスターノードの BIOS を設定し ACPI Soft-Off を無効にします。

1. ノードを再起動して、 **BIOS CMOS Setup Utility** プログラムを開始します。
2. **Power** メニュー (または電源管理に相当するメニュー) に移動します。
3. **Power** メニューで **Soft-Off by PWR-BTTN** の機能 (またはこれに相当する機能) を **Instant-Off** (またはこれに相当し遅延なく電源ボタンでノードの電源を切る設定) に設定します。 例 1.1「**BIOS CMOS Setup Utility**: **Soft-Off by PWR-BTTN** が **Instant-Off** に設定されている状態」 では、 **Power** メニューで **ACPI Function** が **Enabled** に、 **Soft-Off by PWR-BTTN** が **Instant-Off** に設定されている例を示しています。

   注記

   ACPI Function、 **Soft-Off by PWR-BTTN**、 **Instant-Off** はそれぞれコンピュータにより機能名が異なる場合があります。 ただし、 記載している手順の目的は、 遅延なく電源ボタンを使ってコンピュータの電源が切れるよう BIOS を設定することです。

4. **BIOS CMOS Setup Utility** プログラムを終了して BIOS の設定を保存します。
5. クラスターを設定して実行を開始したら、 フェンシングの際にノードの電源が直ちにオフになることを確認します。

   注記

   ノードのフェンシングは **fence_node** コマンドや **Conga** を使っても行なうことができます。

例1.1 BIOS CMOS Setup Utility: Soft-Off by PWR-BTTN が Instant-Off に設定されている状態

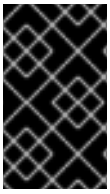
```
+---------------------------------------------|-----------------+
|   ACPI Function              [Enabled]      |  Item Help      |
|   ACPI Suspend Type          [S1(POS)]      |-----------------|
| x Run VGABIOS if S3 Resume   [Auto]         | Menu Level   *  |
|   Suspend Mode               [Disabled]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

この例では、 **ACPI Function** が **Enabled** に、 **Soft-Off by PWR-BTTN** が **Instant-Off** に設定されています。

1.1.3. `grub.conf` ファイル内で ACPI を完全に無効にする

推奨している方法は **chkconfig** 管理を使って ACPI Soft-Off を無効にする方法です (「**chkconfig** 管理を使って ACPI Soft-Off を無効にする」)。 推奨している方法がクラスターには適さない場合には、BIOS の電源管理を使って ACPI Soft-Off を無効にすることができます (「BIOS を使って ACPI Soft-Off を無効にする」)。 これらいずれの方法もクラスターに適さない場合には、 **grub.conf** ファイル内のカーネルブートコマンドラインに **acpi=off** を追加して ACPI を完全に無効にします。



重要

この方法を使用すると ACPI が完全に無効になります。 ACPI が完全に無効になっていると正しく起動しないコンピュータがあります。 この方法を使用するのは **他の方法がクラスターに適さない場合に限ってください** 。

以下のようにして、 各クラスターノードの **grub.conf** ファイルを編集し ACPI を完全に無効にします。

1. テキストエディタで **/boot/grub/grub.conf** を開きます。
2. **acpi=off** を **/boot/grub/grub.conf** 内のカーネルブートコマンドラインに追加します (例 1.2「カーネルブートコマンドラインに **acpi=off** を追加した状態」 を参照)。
3. ノードを再起動します。
4. クラスターを設定して実行を開始したら、 フェンシングの際にノードの電源が直ちにオフになることを確認します。

注記

ノードのフェンシングは **fence_node** コマンドや **Conga** を使っても行なうことができます。

例1.2 カーネルブートコマンドラインに acpi=off を追加した状態

```
# 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
```

この例では、 **acpi=off** がカーネルブートコマンドライン ("kernel /vmlinuz-2.6.32-193.el6.x86_64.img" で始まる行) に追加されています。

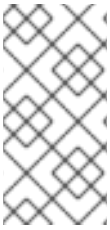
第2章 CCS コマンドを使ってフェンシングを設定する

現在の Red Hat Enterprise Linux 6.1 以降のリリースより、 Red Hat High Availability Add-On では **ccs** クラスター設定コマンドに対応するようになります。**ccs** コマンドを使用することにより **cluster.conf** クラスター設定ファイルの作成や変更、表示などを行なうことができるようになります。**ccs** コマンドを使用するとローカルのファイルシステムや遠隔にあるノードのクラスター設定ファイルの設定を行なうことができます。また、**ccs** コマンドを使って、設定したクラスター内の任意のノードあるいはすべてのノードでクラスターサービスの起動や停止を行うこともできます。

本章では **ccs** コマンドを使った Red Hat High Availability Add-On のクラスター設定ファイルの設定方法について説明しています。

次のようなセクションで構成されます。

- 「フェンスデバイスを設定する」



注記

High Availability Add-On の導入が確かにニーズに適合しサポートされるのか必ず確認してください。導入を行なう前に、 Red Hat 認定担当者に連絡して導入予定の構成の確認を行なうようにしてください。 また、 バーンイン期間を設けて障害モードのテストを実施するようにしてください。

注記

本章では、 よく使用される **cluster.conf** のエレメントや属性について記載しています。 **cluster.conf** のエレメントおよび属性の全一覧とその詳細については **/usr/share/cluster/cluster.rng** のクラスタースキーマおよび **/usr/share/doc/cman-X.Y.ZZ/cluster_conf.html** の注釈付きスキーマ (**/usr/share/doc/cman-3.0.12/cluster_conf.html** など) を参照してください。

2.1. フェンスデバイスを設定する

フェンスデバイスの設定とは、クラスターにフェンスデバイスを作成する、フェンスデバイスを更新する、 フェンスデバイスを削除することを指します。クラスター内のノードに対して排他処理 (フェンシング) を設定する前に、まずクラスター内にフェンスデバイスを作成して名前を付ける必要があります。クラスター内の各ノードに対してフェンシングを設定する方法については 「クラスターのメンバーにフェンシングを設定する」 を参照してください。

フェンスデバイスを設定する前に、フェンスデーモンプロパティの一部をシステムに合わせてデフォルト値から変更したい場合があるかもしれません。 フェンスデーモンに設定する値はクラスター全体に適用されます。変更が可能な汎用フェンスプロパティを以下に簡単に説明しておきます。

- **post_fail_delay** 属性は、 ノードに障害が発生した場合にそのノード (フェンスドメインのメンバー) を排他処理するまでにフェンスデーモン (**fenced**) を待機させる秒数です。 **post_fail_delay** のデフォルト値は **0** です。 この値はクラスターとネットワークのパフォーマンスに合わせて変更できます。

- **post-join_delay** 属性は、 ノードがフェンスドメインに参加した後そのノードを排他処理するまでにフェンスデーモン (**fenced**) を待機させる秒数です。 **post_join_delay** のデフォルト値は **6**です。 **post_join_delay** に見られる一般的な設定は 20 秒から 30 秒ですが、 クラスターやネットワークのパフォーマンスにより異なります。

post_fail_delay 属性と **post_join_delay** 属性の値をリセットする場合は **ccs** コマンドの **--setfencedaemon** オプションを使用します。 ただし、**ccs --setfencedaemon** コマンドを実行すると既存のフェンスデーモンプロパティがすべて上書きされるため注意してください。

たとえば、 **post_fail_delay** 属性の値を設定する場合は次のコマンドを実行します。 このコマンドで設定できるその他すべての既存フェンスデーモンプロパティの値が上書きされます。

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

post_join_delay 属性の値を設定する場合は次のコマンドを実行します。 このコマンドで設定できるその他すべての既存フェンスデーモンプロパティの値が上書きされます。

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

post_join_delay 属性と **post_fail_delay** 属性の両方の値を設定する場合は次のコマンドを実行します。

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

注記

post_join_delay 属性、**post_fail_delay** 属性、変更可能なフェンスデーモンのプロパティの詳細は fenced(8) の man ページおよび **/usr/share/cluster/cluster.rng** のクラスタースキーマ、**/usr/share/doc/cman-X.Y.ZZ/cluster_conf.html** の注釈付きスキーマなどを参照してください。

クラスターにフェンスデバイスを設定する場合は次のコマンドを実行します。

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

たとえば、 **node1** と言うクラスターノードの設定ファイル内に **myfence** と言う APC フェンスデバイスを設定する場合は、 次のコマンドを実行します。 IP アドレスは **apc_ip_example**、 ログインは **login_example**、 パスワードは **password_example** とします。

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

APC フェンスデバイスを追加した後の **cluster.conf** 設定ファイル内の **fencedevices** セクションの例を示します。

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

クラスターにフェンスデバイスを設定する場合、クラスターに使用できるデバイスとそのデバイスのオプション一覧を表示できると便利な場合があります。また、クラスターに現在設定しているフェンスデバイスを表示させたい場合もあります。**ccs** コマンドを使って使用できるフェンスデバイスやオプショ

ンの一覧を表示させたり、現在設定しているフェンスデバイス一覧を表示させる方法については 「フェンスデバイスとフェンスデバイスのオプションの一覧を表示する」 を参照してください。

クラスター設定からフェンスデバイスを削除する場合は次のコマンドを実行します。

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

たとえば、 **myfence** と言う名前を付けたフェンスデバイスを **node1** という名前のクラスターノードにあるクラスター設定ファイルから削除する場合は次のコマンドを実行します。

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

すでに設定済みのフェンスデバイスの属性を変更する必要がある場合は、 まずそのフェンスデバイスを削除して属性の変更を行なってから再びそのフェンスデバイスの追加を行ないます。

クラスターのコンポーネントの設定がすべて終了したら、 クラスター設定ファイルを全ノードに対して同期する必要があります。

2.2. フェンスデバイスとフェンスデバイスのオプションの一覧を表示する

ccs コマンドを使って、 使用できるフェンスデバイスの一覧を出力させたり、 フェンスタイプごとに使用できるオプション一覧を出力させたりすることができます。 また、 クラスターに現在設定しているフェンスデバイスの一覧を出力させることもできます。

現在、 クラスターに使用できるフェンスデバイスの一覧を出力させる場合は次のコマンドを実行します。

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

たとえば、 次のコマンドではクラスターノード **node1** で使用できるフェンスデバイスの一覧が表示されます。 サンプルの出力を示します。

```
[root@ask-03 ~]# ccs -h node1 --lsfenceopts
fence_rps10 - RPS10 Serial Switch
fence_vixel - No description available
fence_egenera - No description available
fence_xcat - No description available
fence_na - Node Assassin
fence_apc - Fence agent for APC over telnet/ssh
fence_apc_snmp - Fence agent for APC over SNMP
fence_bladecenter - Fence agent for IBM BladeCenter
fence_bladecenter_snmp - Fence agent for IBM BladeCenter over SNMP
fence_cisco_mds - Fence agent for Cisco MDS
fence_cisco_ucs - Fence agent for Cisco UCS
fence_drac5 - Fence agent for Dell DRAC CMC/5
fence_eps - Fence agent for ePowerSwitch
fence_ibmblade - Fence agent for IBM BladeCenter over SNMP
fence_ifmib - Fence agent for IF MIB
fence_ilo - Fence agent for HP iLO
fence_ilo_mp - Fence agent for HP iLO MP
fence_intelmodular - Fence agent for Intel Modular
fence_ipmilan - Fence agent for IPMI over LAN
fence_kdump - Fence agent for use with kdump
fence_rhevm - Fence agent for RHEV-M REST API
```

```
fence_rsa - Fence agent for IBM RSA
fence_sanbox2 - Fence agent for QLogic SANBox2 FC switches
fence_scsi - fence agent for SCSI-3 persistent reservations
fence_virsh - Fence agent for virsh
fence_virt - Fence agent for virtual machines
fence_vmware - Fence agent for VMware
fence_vmware_soap - Fence agent for VMware over SOAP API
fence_wti - Fence agent for WTI
fence_xvm - Fence agent for virtual machines
```

特定のフェンスタイプに指定できるオプションの一覧を出力させる場合は次のコマンドを実行します。

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

たとえば、 次のコマンドでは **fence_wti** フェンスエージェントのフェンスオプションが表示されます。

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

現在、 クラスターに設定しているフェンスデバイスの一覧を出力させる場合は次のコマンドを実行します。

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

2.3. クラスターのメンバーにフェンシングを設定する

クラスターを作成しフェンスデバイスを作成する最初の手順が完了したら、クラスターノードにフェン

シングを設定する必要があります。新しいクラスターを作成しそのクラスターにフェンスデバイスを設定した後、本セクションの手順に従って ノードにフェンシングを設定します。フェンシングの設定はクラスター内の各ノードに対してそれぞれ行わなければなりません。

このセクションでは次のような手順について説明していきます。

- 「ノードに対して電源ベースのフェンスデバイスを一つ設定する」
- 「ノードに対してストレージベースのフェンスデバイスを一つ設定する」
- 「バックアップ用のフェンスデバイスを設定する」
- 「冗長電源を備えたノードの設定」
- 「フェンスメソッドとフェンスインスタンスを削除する」

2.3.1. ノードに対して電源ベースのフェンスデバイスを一つ設定する

次の手順に従って、 **apc** と言う名前のフェンスデバイスを使用する電源ベースのフェンスデバイスをノードに設定します。 **fence_apc** というフェンスエージェントを使用します。

1. フェンスメソッドの名前を入力してノードにフェンスメソッドを追加します。

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

   たとえば、 **node-01.example.com** というクラスターノードの設定ファイル内で **node-01.example.com** というノードに **APC** というフェンスメソッドを設定する場合は次のコマンドを実行します。

   ```
   ccs -h node-01.example.com --addmethod APC node-01.example.com
   ```

2. このメソッドにフェンスインスタンスを追加します。 このノードに使用するフェンスデバイス、 このインスタンスを適用させるノード、 メソッド名、 このノードに対して固有となるこのメソッドのオプションなどを指定する必要があります。

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

   たとえば、クラスターノード **node-01.example.com** の設定ファイルにフェンスインスタンスを設定します。ここで使用しているフェンスデバイスは **apc**、インスタンスを適用するノードは **node-01.example.com**、メソッド名は **APC** とし、ノードに固有となるメソッド用のオプションにはフェンスデバイスの APC スイッチの電源ポート 1 を指定しています。

   ```
   ccs -h node-01.example.com --addfenceinst apc node-01.example.com
   APC port=1
   ```

フェンスメソッドはクラスター内の各ノードに対してそれぞれ追加しなければなりません。**APC** というメソッド名で各ノードにフェンスメソッドを設定するコマンドを以下に示します。フェンスメソッドのデバイスには **apc** というデバイス名を指定しています。「フェンスデバイスを設定する」 で説明したように **--addfencedev** オプションを付けて設定します。ノードにそれぞれ固有の APC スイッチ電源ポート番号を指定します。 **node-01.example.com** のポート番号は **1**、 **node-02.example.com** のポート番号は **2**、 **node-03.example.com** のポート番号は **3** に指定しています。

```
ccs -h node-01.example.com --addmethod APC node-01.example.com
```

```
ccs -h node-01.example.com --addmethod APC node-02.example.com
ccs -h node-01.example.com --addmethod APC node-03.example.com
ccs -h node-01.example.com --addfenceinst apc node-01.example.com APC
port=1
ccs -h node-01.example.com --addfenceinst apc node-02.example.com APC
port=2
ccs -h node-01.example.com --addfenceinst apc node-03.example.com APC
port=3
```

例2.1「電源ベースのフェンスメソッドを追加した状態の **cluster.conf**」 に、 クラスター内の各ノードにフェンシングメソッドとインスタンスを追加した後の **cluster.conf** 設定ファイルを示します。

例2.1 電源ベースのフェンスメソッドを追加した状態の cluster.conf

```
<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>
```

クラスターのコンポーネントの設定がすべて終了したら、 クラスター設定ファイルを全ノードに対して同期する必要があります。

2.3.2. ノードに対してストレージベースのフェンスデバイスを一つ設定する

電源フェンシング以外のメソッド（SAN/ストレージフェンシング）を使用してノードのフェンシング

を行なう場合、 フェンスデバイスには *unfencing（アンフェンシング）* の設定を行なう必要があります。 これにより、 フェンシングされたノードは再起動が行なわれるまでは、 再び有効にならないようにします。 ノードにアンフェンシングを設定する場合は、 そのノードに設定したフェンスデバイスを反映しているデバイスを指定し、 action を **on** または **enable** の設定で明示的に付け加えます。

ノードをアンフェンシングする方法については、 **fence_node**(8) の man ページを参照してください。

次の手順に従って、 **sanswitch1** と言う名前のフェンスデバイスを使用するストレージベースのフェンスデバイスをノードに設定します。 **fence_sanbox2** というフェンスエージェントを使用します。

1. フェンスメソッドの名前を入力してノードにフェンスメソッドを追加します。

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

   例えば、 クラスターノード **node-01.example.com** にある設定ファイル内のノード **node-01.example.com** に **SAN** と呼ばれるフェンスメソッドを設定する場合は、 次のコマンドを実行します。

   ```
   ccs -h node-01.example.com --addmethod SAN  node-01.example.com
   ```

2. このメソッドにフェンスインスタンスを追加します。 このノードに使用するフェンスデバイス、 このインスタンスを適用させるノード、 メソッド名、 このノードに対して固有となるこのメソッドのオプションなどを指定する必要があります。

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

   例えば、 クラスターノード **node-01.example.com** の設定ファイルにフェンスインスタンスを設定します。ここで使用しているフェンスデバイスは **sanswitch1**、適用するクラスターノードは **node-01.example.com**、メソッド名は **SAN** とし、ノードに固有となるオプションにはフェンスデバイスの SAN スイッチ電源ポート 11 を指定しています。

   ```
   ccs -h node-01.example.com --addfenceinst sanswitch1 node-
   01.example.com SAN port=11
   ```

3. このノードのストレージベースのフェンスデバイスにアンフェンシングを設定する場合は、 次のコマンドを実行します。

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

クラスター内の各ノードに対してフェンスメソッドを追加する必要があります。 次のコマンドでは **SAN** と言うメソッド名を付けてフェンスメソッドをノードに設定しています。 フェンスメソッドのデバイスには **sanswitch** と言うデバイス名が付けられています。 前回、 --addfencedev オプションを付けて設定したデバイスです。 「フェンスデバイスを設定する」 で説明しています。 ノードはそれぞれ固有の SAN 物理ポート番号で設定されます。 **node-01.example.com** ならポート番号は **11**、 **node-02.example.com** なら **12**、 **node-03.example.com** なら **13** になります。

```
ccs -h node-01.example.com --addmethod SAN node-01.example.com
ccs -h node-01.example.com --addmethod SAN node-02.example.com
ccs -h node-01.example.com --addmethod SAN node-03.example.com
ccs -h node-01.example.com --addfenceinst sanswitch1 node-01.example.com
SAN port=11
ccs -h node-01.example.com --addfenceinst sanswitch1 node-02.example.com
SAN port=12
```

```
ccs -h node-01.example.com --addfenceinst sanswitch1 node-03.example.com
SAN port=13
ccs -h node-01.example.com --addunfence sanswitch1 node-01.example.com
port=11 action=on
ccs -h node-01.example.com --addunfence sanswitch1 node-02.example.com
port=12 action=on
ccs -h node-01.example.com --addunfence sanswitch1 node-03.example.com
port=13 action=on
```

例2.2「ストレージベースのフェンスメソッドを追加した状態の **cluster.conf**」 では、 クラスター内の各ノードにフェンシングメソッド、 フェンスインスタンス、 アンフェンシングを追加した後の **cluster.conf** 設定ファイルを示します。

例2.2 ストレージベースのフェンスメソッドを追加した状態の cluster.conf

```
<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>
```

クラスターのコンポーネントの設定がすべて終了したら、 クラスター設定ファイルを全ノードに対して同期する必要があります。

2.3.3. バックアップ用のフェンスデバイスを設定する

1 つのノードに対して複数のフェンシングメソッドを定義することが可能です。 最初のメソッドで失敗すると、 2 番目のメソッドでノードのフェンシングを試行します。 さらにメソッドを設定していればそれらのメソッドが順次試行されていきます。 ノードにバックアップ用のフェンシングメソッドを設定する場合は、 1 ノードに 2 種類のメソッドを設定し、 各メソッドに対してフェンスインスタンスを設定します。

注記

設定したフェンシングメソッドが使用される順序は、 クラスター設定ファイルに記載されている順序に従います。 **ccs** コマンドで設定する一番目のメソッドが第一フェンシングメソッドになり、 2 番目に設定するメソッドがバックアップ用のフェンシングメソッドになります。 順序を変更する場合は、 一旦、 設定ファイルから第一フェンシングメソッドを削除し、 そのメソッドを追加し直します。

次のコマンドを実行すると、 ノードに現在設定しているフェンスメソッドとインスタンスの一覧をいつでも出力させることができます。 ノードを指定しないでコマンドを実行すると、 全ノードに設定されているフェンスメソッドとインスタンスが出力されます。

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

次の手順にしたがい、 ノードに第一フェンシングメソッドを設定します。 使用するフェンスデバイスは **apc**、 フェンスエージェントは **fence_apc** です。 また、 バックアップ用のフェンスデバイスは **sanswitch1**、 フェンスエージェントは **fence_sanbox2** を使用します。 **sanswitch1** デバイスはストレージベースのフェンスエージェントのため、 このデバイスにはアンフェンシングの設定も行なう必要があります。

1. ノードにフェンスメソッド名を与え、 第一フェンスメソッドを追加します。

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

   たとえば、 **node-01.example.com** と言うクラスターノードの設定ファイル内で **APC** という第一フェンスメソッドを **node-01.example.com** と言うノードに対して設定する場合は次のようなコマンドになります。

   ```
   ccs -h node-01.example.com --addmethod APC node-01.example.com
   ```

2. 第一メソッドにフェンスインスタンスを追加します。 ノードに使用するフェンスデバイス、 このインスタンスを適用するノード、 メソッド名、 このノードに固有となるこのメソッドのオプションなどを指定する必要があります。

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

   たとえば、クラスターノード **node-01.example.com** の設定ファイルにフェンスインスタンスを設定します。ここで使用しているフェンスデバイスは **apc**、インスタンスを適用するノードは **node-01.example.com**、メソッド名は **APC** とし、ノードに固有となるメソッド用のオ

プションにはフェンスデバイスの APC スイッチの電源ポート 1 を指定しています。

```
ccs -h node-01.example.com --addfenceinst apc node-01.example.com
APC port=1
```

3. フェンスメソッド名を与え、 ノードにバックアップ用のフェンスメソッドを追加します。

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

たとえば、 **node-01.example.com** というクラスターノードの設定ファイル内で **SAN** というバックアップ用のフェンスメソッドを **node-01.example.com** というノードに対して設定する場合は次のようなコマンドになります。

```
ccs -h node-01.example.com --addmethod SAN  node-01.example.com
```

4. バックアップ用のメソッドにフェンスインスタンスを追加します。 ノードに使用するフェンスデバイス、 このインスタンスを適用するノード、 メソッド名、 このノードに固有となるこのメソッドのオプションなどを指定する必要があります。

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

例えば、 クラスターノード **node-01.example.com** の設定ファイルにフェンスインスタンスを設定します。ここで使用しているフェンスデバイスは **sanswitch1**、適用するクラスターノードは **node-01.example.com**、メソッド名は **SAN** とし、ノードに固有となるオプションにはフェンスデバイスの SAN スイッチ電源ポート 11 を指定しています。

```
ccs -h node-01.example.com --addfenceinst sanswitch1 node-
01.example.com SAN port=11
```

5. **sanswitch1** デバイスはストレージベースのデバイスのためアンフェンシングの設定を行なう必要があります。

```
ccs -h node-01.example.com --addunfence sanswitch1 node-
01.example.com port=11 action=on
```

必要に応じて引き続きフェンシングメソッドを追加します。

この手順では、 クラスター内の一つのノードに対してフェンスデバイスおよびバックアップ用のフェンスデバイスを設定しています。 このクラスター内の他のノードに対してもフェンシングを設定する必要があります。

電源ベースの第一フェンシングメソッドとストレージベースのバックアップ用フェンシングメソッドをクラスター内の各ノードに追加した後の **cluster.conf** 設定ファイルを 例2.3「バックアップ用のフェンスメソッドを追加した状態の **cluster.conf**」 に示します。

例2.3 バックアップ用のフェンスメソッドを追加した状態の cluster.conf

```
<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>
```

クラスターのコンポーネントの設定がすべて終了したら、 クラスター設定ファイルを全ノードに対して同期する必要があります。

注記

設定したフェンシングメソッドが使用される順序は、 クラスター設定ファイルに記載されている順序に従います。 一番目に設定するメソッドが第一フェンシングメソッドになり、 2 番目に設定するメソッドがバックアップ用のフェンシングメソッドになります。順序を変更する場合は、 一旦、 設定ファイルから第一フェンシングメソッドを削除し、そのメソッドを追加し直します。

2.3.4. 冗長電源を備えたノードの設定

ノード用の冗長電源装置をクラスター設定している場合は、ノードの排他処理を行う必要がある場合にそのノードが完全にシャットダウンするようフェンシングが正しく設定されているか確認してください。各電源装置を別々のフェンスメソッドとして設定するとフェンシングも別々に行われます。つまり、1 つ目の電源装置が排他処理されても 2 つ目の電源装置でシステムは稼働し続けることができるため、排他処理が行なわれないことになります。二重に電源装置を備えたシステムの設定を行なう場合には、両方の電源装置が遮断されシステムが完全に停止するようフェンスデバイスを設定する必要があります。そのためには、単一のフェンスメソッド内に 2 つのインスタンスを設定する必要があります。また、各インスタンスに対して **action** 属性が **off** のデバイスと **on** のデバイス 2 種類を設定します。設定順序は **action** 属性が **off** のデバイスを先に、 そのあと **on** のデバイスを設定します。

二重電源装置を備えたノードにフェンシングを設定する場合は、 本セクションの手順に従ってください。

1. 冗長電源装置を備えたノードにフェンシングを設定する場合は、 まず先に各電源スイッチをクラスターのフェンスデバイスとして設定しておく必要があります。フェンスデバイスの設定に関する詳細は 「フェンスデバイスを設定する」 を参照してください。

   現在、 クラスターに設定しているフェンスデバイスの一覧を出力させる場合は次のコマンドを実行します。

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

2. フェンスメソッドの名前を入力してノードにフェンスメソッドを追加します。

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

   例えば、クラスターノード **node-01.example.com** の設定ファイル内で **APC-dual** という名前のフェンスメソッドをノード **node-01.example.com** に対して設定する場合は次のようなコマンドになります。

   ```
   ccs -h node-01.example.com --addmethod APC-dual node-01.example.com
   ```

3. 1 つ目の電源装置用のフェンスインスタンスをフェンスメソッドに追加します。ノードに使用するフェンスデバイス、このインスタンスを適用するノード、メソッド名、このノードに固有となるこのメソッド用のオプションなどを指定します。ここでは **action** 属性は **off** に設定します。

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

   例えば、 クラスターノード **node-01.example.com** の設定ファイルにフェンスインスタンスを設定するには 以下のコマンドを実行します。ここで使用しているフェンスデバイスは **apc1**、適用するノードは **node-01.example.com**、メソッド名は **APC-dual** とし、ノードに

固有となるオプションにはフェンスデバイスの APC スイッチ電源ポート 1 を指定、**action** 属性は **off** に設定しています。

```
ccs -h node-01.example.com --addfenceinst apc1 node-01.example.com
APC-dual port=1 action=off
```

4. 2 つ目の電源装置用のフェンスインスタンスをフェンスメソッドに追加します。 ノードに使用するフェンスデバイス、 このインスタンスを適用するノード、 メソッド名、 このノードに固有となるこのメソッド用のオプションなどを指定する必要があります。 ここでも、 インスタンスの **action** 属性は **off** に設定します。

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

例えば、 クラスターノード **node-01.example.com** の設定ファイルに 2 つ目のフェンスインスタンスを設定するには以下のコマンドを実行します。ここで使用しているフェンスデバイスは **apc2**、適用するノードは **node-01.example.com**、メソッド名は 1 つ目のインスタンスと同じ **APC-dual** とし、ノードに固有となるオプションにはフェンスデバイスの APC スイッチ電源ポート 1 を指定、**action** 属性は **off** に設定しています。

```
ccs -h node-01.example.com --addfenceinst apc2 node-01.example.com
APC-dual port=1 action=off
```

5. 次に、1 つ目の電源装置用にフェンスインスタンスをもうひとつフェンスメソッドに追加、**action** 属性は **on** に設定します。ノードに使用するフェンスデバイス、このインスタンスを適用するノード、メソッド名、このノードに固有となるこのメソッド用のオプションなどを指定する必要があります。また、**action** 属性は **on** に設定します。

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

例えば、 クラスターノード **node-01.example.com** の設定ファイルにフェンスインスタンスを設定するには以下のコマンドを実行します。ここで使用しているフェンスデバイスは **apc1**、適用するノードは **node-01.example.com**、メソッド名は **APC-dual** とし、ノードに固有となるオプションにはフェンスデバイスの APC スイッチ電源ポート 1 を指定、**action** 属性は **on** に設定しています。

```
ccs -h node-01.example.com --addfenceinst apc1 node-01.example.com
APC-dual port=1 action=on
```

6. 2 つ目の電源装置用にもフェンスインスタンスをもうひとつフェンスメソッドに追加、**action** 属性は **on** に設定します。ノードに使用するフェンスデバイス、このインスタンスを適用するノード、メソッド名、このノードに固有となるこのメソッド用のオプションなどを指定する必要があります。また、**action** 属性は **on** に設定します。

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

例えば、 クラスターノード **node-01.example.com** の設定ファイルに 2 つ目のフェンスインスタンスを設定するには以下のコマンドを実行します。ここで使用しているフェンスデバイスは **apc2**、適用するノードは **node-01.example.com**、メソッド名は 1 つ目のインスタンスと

同じ **APC-dual** とし、ノードに固有となるオプションにはフェンスデバイスの APC スイッチ電源ポート 1 を指定、**action** 属性は **on** に設定しています。

```
ccs -h node-01.example.com --addfenceinst apc2 node-01.example.com
APC-dual port=1 action=on
```

二重電源装置用のフェンシングをクラスターの各ノードに追加した後の **cluster.conf** 設定ファイルを 例2.4「二重電源装置用フェンシングを追加した状態の **cluster.conf**」 に示します。

例2.4 二重電源装置用フェンシングを追加した状態の cluster.conf

```
<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>
```

クラスターのコンポーネントの設定がすべて終了したら、 クラスター設定ファイルを全ノードに対して同期する必要があります。

2.3.5. フェンスの設定をテストする

Red Hat Enterprise Linux Release 6.4 からは **fence_check** ユーティリティーを使用すると、クラスター内の各ノードに対しフェンスの設定をテストすることができます。

このコマンドを正常に実行できた場合の出力を以下に示します。

```
[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
```

このユーティリティーの詳細については **fence_check**(8) man ページを参照してください。

2.3.6. フェンスメソッドとフェンスインスタンスを削除する

クラスター設定からフェンスメソッドを削除する場合は次のコマンドを実行します。

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

例えば、 **node01.example.com** に設定した **APC** と言うフェンスメソッドをクラスターノード **node01.example.com** のクラスター設定ファイルから削除するには以下のコマンドを実行します。

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

任意のフェンスデバイスの全フェンスインスタンスをフェンスメソッドから削除する場合は以下のコマンドを実行します。

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

例えば、クラスターノード **node01.example.com** のクラスター設定ファイルから **node01.example.com** ノードに対して設定した **APC-dual** と言うメソッド内の **apc1** と言うフェンスデバイスの全インスタンスを削除する場合は以下のコマンドを実行します。

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

第3章 CONGA を使ってフェンシングを設定する

本章では、 **Conga** を使って Red Hat High Availability Add-On にフェンシングを設定する方法を説明していきます。

注記

Conga とは、Red Hat High Availability Add-On の管理に使用できるグラフィカルユーザーインターフェースです。ただし、このインターフェースを実質的に使用するためにはユーザー側が基本概念を充分に理解していなければなりません。ユーザーインターフェースで利用できる機能を試行錯誤しながらクラスターの設定について学ぼうとするのはお勧めできません。十分な知識がないまま手探りで設定を行うと、堅牢性が不十分なシステムとなる可能性があり、コンポーネントに障害が発生した場合には実行している全サービスを維持できなくなる恐れがあります。

- 「フェンスデバイスを設定する」

3.1. フェンスデーモンのプロパティを設定する

Fence Daemon タブをクリックすると、**Fence Daemon Properties** ページに **Post Fail Delay** と **Post Join Delay** を設定するインターフェースが表示されます。このパラメータに設定する値はクラスター全体に適用されるフェンスプロパティです。クラスター内のノードに特定のフェンスデバイスを設定する場合は、「フェンスデバイスを設定する」 の説明にしたがいクラスター表示の **Fence Devices** メニューアイテムを使用してください。

- **Post Fail Delay** パラメータは、ノードに障害が発生した場合にそのノード (フェンスドメインのメンバー) を排他処理するまでにフェンスデーモン (**fenced**) を待機させる秒数です。 **Post Fail Delay** のデフォルト値は **0** です。 この値はクラスターとネットワークのパフォーマンスに合わせて変更できます。

- **Post Join Delay** パラメータは、ノードがフェンスドメインに参加した後そのノードを排他処理するまでにフェンスデーモン (**fenced**) を待機させる秒数です。 **Post Join Delay** のデフォルト値は **6**です。 **Post Join Delay** は 20 秒から 30 秒が一般的な設定ですが、 クラスターやネットワークのパフォーマンスによ変更することができます。

適切な値を入力して **Apply** をクリックすると変更が反映されます。

注記

Post Join Delay および **Post Fail Delay** の詳細は fenced(8) の man ページを参照してください。

3.2. フェンスデバイスを設定する

フェンスデバイスの設定とは、クラスターのフェンスデバイスの作成、 更新、 削除などを指します。クラスター内のノードに対してフェンシングを設定する前に、 まずクラスター内にフェンスデバイスを設定する必要があります。

フェンスデバイスを作成するには、 フェンスデバイスのタイプを選択し、 そのフェンスデバイスのパラメータ (名前、 IP アドレス、 ログイン、 パスワードなど) を入力します。 フェンスデバイスを更新するには、 既存のフェンスデバイスを選択して、 そのフェンスデバイスのパラメータを変更します。フェンスデバイスを削除するには、 既存のフェンスデバイスを選択して削除します。

このセクションでは、 以下の作業についての手順を説明しています。

- フェンスデバイスを作成する — 「フェンスデバイスを作成する」 を参照してください。 フェンスデバイスを作成してデバイス名を付けたら、 「クラスターメンバーにフェンシングを設定する」 の説明にしたがいクラスター内の各ノードに対してフェンスデバイスを設定できるようになります。
- フェンスデバイスを更新する — 「フェンスデバイスを修正する」 を参照してください。
- フェンスデバイスを削除する — 「フェンスデバイスを削除する」 を参照してください。

クラスター固有のページからクラスター表示の上部にある **Fence Devices** をクリックするとそのクラスターのフェンスデバイスを設定することができます。フェンスデバイスとフェンスデバイス設定用のメニュー項目 **Add** と **Delete** が表示されます。以下のセクションで説明する各手順はここから開始することになります。

注記

初めてクラスターの設定を行なう場合、まだフェンスデバイスを作成していないため何も表示されません。

またフェンスデバイスを作成していない状態のフェンスデバイス設定画面を 図3.1「luci フェンスデバイス設定ページ」 を示します。

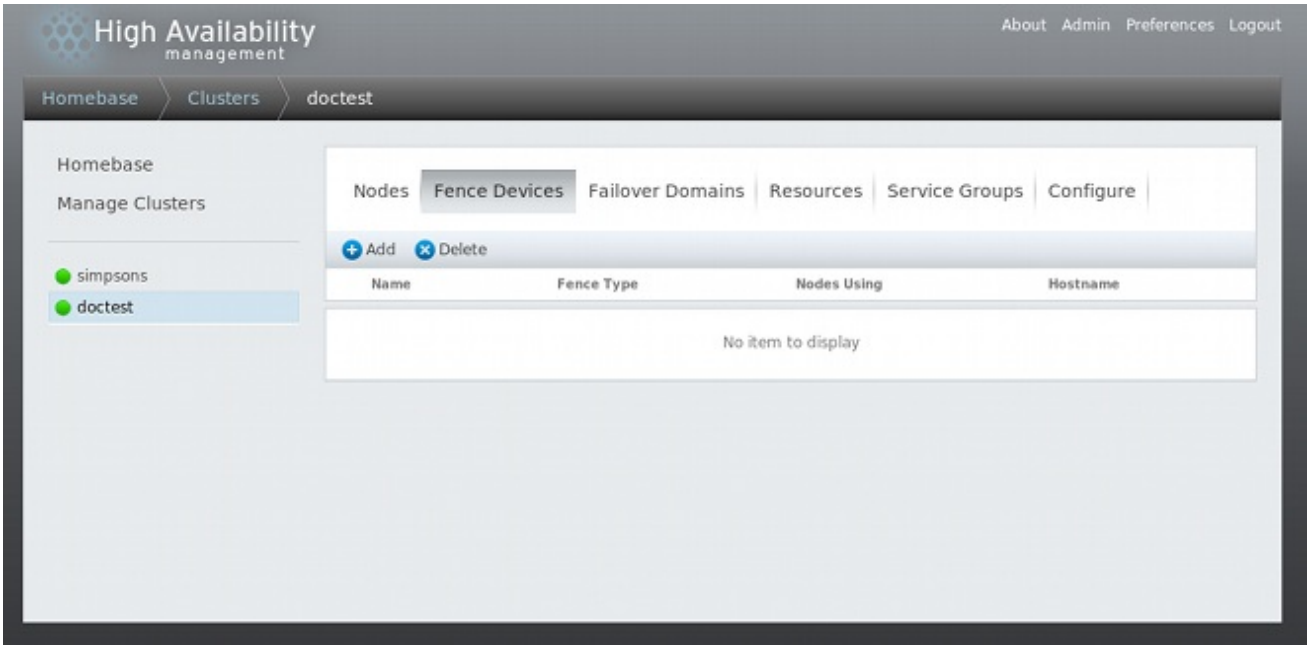


図3.1 luci フェンスデバイス設定ページ

3.2.1. フェンスデバイスを作成する

次の手順にしたがいフェンスデバイスを作成します。

1. **Fence Devices** 設定ページから、**Add** をクリックします。**Add** をクリックすると、**Add Fence Device (Instance)** ダイアログボックスが表示されます。このダイアログボックスから、設定するフェンスデバイスのタイプを選択します。

2. **Add Fence Device (Instance)** ダイアログボックスにフェンスデバイス名とタイプを指定します。 ノードごとにフェンシングを設定するなどの場合、 フェンスデバイスにノード固有のパラメータを追加で指定する必要があるかもしれません。

3. **Submit** をクリックします。

フェンスデバイスの追加が完了すると、 **Fence Devices** 設定ページ上にフェンスデバイスが表示されます。

3.2.2. フェンスデバイスを修正する

次の手順にしたがいフェンスデバイスを修正します。

1. **Fence Devices** 設定ページから、 修正を行なうフェンスデバイス名をクリックします。 そのデバイスに設定した値を示しているフェンスデバイスのダイアログボックスが表示されます。

2. フェンスデバイスを修正するため、 表示パラメータに対する変更を入力します。

3. **Apply** をクリックし、 設定が更新されるのを待ちます。

3.2.3. フェンスデバイスを削除する

注記

使用中のフェンスデバイスは削除できません。 ノードが現在使用しているフェンスデバイスを削除するには、 まずこのデバイスを使用しているすべてのノードのフェンス設定を更新してからそのデバイスを削除します。

次の手順にしたがいフェンスデバイスを削除します。

1. **Fence Devices** 設定のページから、 フェンスデバイスの左にあるボックスにチェックマークを入れて、 削除するデバイスを選択します。

2. **Delete** をクリックして、 設定が更新されるのを待ちます。 デバイスが削除中であることを示すメッセージが表示されます。

設定が更新されると、 削除されたデバイスは表示されなくなります。

3.3. クラスターメンバーにフェンシングを設定する

クラスターの作成、 フェンスデバイスの作成など最初の手順が完了したら、 クラスターノードにフェンシングを設定する必要があります。 新しいクラスターを作成しクラスターにフェンシングのデバイスを設定した後、 ノードにフェンシングを設定するには、 このセクションに記載している手順に従って行ないます。 フェンシングはクラスター内の各ノードに対してそれぞれ行なう必要があるため注意してください。

以下のセクションでは、 ノードにフェンスデバイスをひとつ設定する手順、 ノードにバックアップ用のフェンスデバイスを設定する手順、 冗長電源装置を備えたノードの設定を行う手順について説明していきます。

- 「ノードにフェンスデバイスをひとつ設定する」
- 「バックアップ用のフェンスデバイスを設定する」

- 「冗長電源を備えたノードの設定」

3.3.1. ノードにフェンスデバイスをひとつ設定する

次の手順にしたがいノードにフェンスデバイスを一つ設定します。

1. クラスター固有のページから、 クラスター表示の上部にある **Nodes** をクリックしてクラスター内のノードにフェンシングを設定します。クラスターを構成しているノード群が表示されます。 このページは、 **luci Homebase** ページの左側のメニューの **Manage Clusters** の下に表示されるクラスター名をクリックした場合にも表示されるデフォルトのページになります。

2. ノード名をクリックします。 ノードのリンクをクリックすると、 そのノードの設定詳細を示すリンク先のページが表示されます。

   ノード固有のページには、 ノードで現在実行中の全サービスの他、 このノードがメンバーとなっているフェールオーバードメインも表示されます。 ドメイン名をクリックすると既存のフェールオーバードメインを変更することができます。

3. ノード固有のページの **Fence Devices** の下にある **Add Fence Method** をクリックします。 **Add Fence Method to Node** ダイアログボックスが表示されます。

4. このノードに設定するフェンスメソッドの **Method Name** を入力します。Red Hat High Availability Add-On で使用される任意の名前になります。デバイスの DNS 名とは異なります。

5. **Submit** をクリックします。ノード固有の画面が表示され、**Fence Devices** の下に追加したメソッドが表示されます。

6. このメソッドにフェンスインスタンスを設定するには、 フェンスメソッドの下に表示される **Add Fence Instance** ボタンをクリックします。 「フェンスデバイスを作成する」 の説明にしたがい前に設定したフェンスデバイスが **Add Fence Device (Instance)** ドロップダウンメニューに表示されます。

7. このメソッドのフェンスデバイスを選択します。 フェンスデバイスにノード固有のパラメータを設定する必要がある場合には設定すべきパラメータが表示されます。

   注記

   電源フェンシング以外のメソッド (SAN/ストレージのフェンシング) の場合、ノード固有のパラメータ表示には **Unfencing** がデフォルトで選択されています。 フェンシングしたノードはその再起動が行なわれるまでストレージには再度アクセスさせないようにするためです。 ノードにアンフェンシングを設定する方法については **fence_node**(8) の man ページを参照してください。

8. **Submit** をクリックします。 ノード固有の画面に戻り、 フェンスメソッドとフェンスインスタンスが表示されます。

3.3.2. バックアップ用のフェンスデバイスを設定する

1 つのノードに対して複数のフェンシングメソッドを定義することが可能です。 最初のメソッドでフェンシングに失敗すると、 2 番目のメソッドでノードのフェンシングを試行します。 さらにメソッドを設定していればそれらのメソッドが順次試行されていきます。

以下の手順にしたがいノードにバックアップ用のフェンスデバイスを設定します。

1. 「ノードにフェンスデバイスをひとつ設定する」 の記載通り、 ノードに第一のフェンシングメソッドを設定します。

2. 定義した第一メソッドの表示で、 **Add Fence Method** をクリックします。

3. このノードに設定するバックアップ用のフェンシングメソッドの名前を入力し、 **Submit** をクリックします。ノード固有の画面が表示され、 追加したメソッドが第一フェンシングメソッドの下に表示されるようになります。

4. このメソッドにフェンスインスタンスを設定するには、 **Add Fence Instance** をクリックします。 「フェンスデバイスを作成する」 の説明にしたがい前に設定したフェンスデバイスがドロップダウンメニューに表示されます。

5. このメソッドのフェンスデバイスを選択します。 フェンスデバイスにノード固有のパラメータを設定する必要がある場合には設定すべきパラメータが表示されます。

6. **Submit** をクリックします。 ノード固有の画面に戻り、 フェンスメソッドとフェンスインスタンスが表示されます。

必要に応じて引き続きフェンシングメソッドを追加します。**Move Up** や **Move Down** をクリックすると、このノードに使用するフェンスメソッドの順序を並べ替えることができます。

3.3.3. 冗長電源を備えたノードの設定

ノード用の冗長電源装置をクラスター設定している場合は、ノードの排他処理を行う必要がある場合にそのノードが完全にシャットダウンするようフェンシングが正しく設定されているか確認してください。各電源装置を別々のフェンスメソッドとして設定するとフェンシングも別々に行われます。つまり、1 つ目の電源装置が排他処理されても 2 つ目の電源装置でシステムは稼働し続けることができるため、排他処理が行なわれないことになります。二重に電源装置を備えたシステムの設定を行なう場合には、両方の電源装置が遮断されシステムが完全に停止するようフェンスデバイスを設定する必要があります。**Conga** でシステムの設定を行なう場合、ひとつのフェンスメソッド内に 2 種類のインスタンスを設定する必要があります。

二重電源装置を備えたノードにフェンシングを設定する場合は、 本セクションの手順に従ってください。

1. 冗長電源装置を備えたノードにフェンシングを設定する前に、 まず各電源スイッチをクラスターのフェンスデバイスとして設定しておく必要があります。 フェンスデバイスの設定方法については、 「フェンスデバイスを設定する」 を参照してください。

2. クラスター固有のページから、 クラスター表示の上部にある **Nodes** をクリックします。 クラスターを構成しているノード群が表示されます。 このページは、 **luci Homebase** ページの左側のメニューの **Manage Clusters** の下に表示されるクラスター名をクリックした場合にも表示されるデフォルトのページになります。

3. ノード名をクリックします。 ノードのリンクをクリックすると、 そのノードの設定詳細を示すリンク先のページが表示されます。

4. ノード固有のページで、 **Add Fence Method** をクリックします。

5. このノードに設定するフェンスメソッドの名前を入力します。

6. **Submit** をクリックします。ノード固有の画面が表示され、**Fence Devices** の下に追加したメソッドが表示されます。

7. **Add Fence Instance** をクリックし、このメソッドに 1 つ目の電源装置をフェンスインスタンスとして設定します。「フェンスデバイスを作成する」 の説明にしたがい前に設定した電源フェンスデバイスがドロップダウンメニューに表示されます。

8. このメソッドに電源フェンスデバイスを一つ選択し適切なパラメータを入力します。

9. **Submit** をクリックします。 ノード固有の画面に戻り、 フェンスメソッドとフェンスインスタンスが表示されます。

10. 1 つ目の電源フェンスデバイスを設定した同じフェンスメソッドで **Add Fence Instance** をクリックします。「フェンスデバイスを作成する」 の説明にしたがい前に設定した電源フェンスデバイスがドロップダウンメニューに表示されます。

11. このメソッドに 2 つ目の電源フェンスデバイスを選択して適切なパラメータを入力します。

12. **Submit** をクリックします。 ノード固有の画面に戻り、 フェンスメソッドとフェンスインスタンスが表示されます。 各デバイスにより順番にシステムの電源が切られ、 また順番にシステムの電源が入れられるのが分かります。 図3.2「二重電源装置を備えている場合のフェンシング設定」 をご覧ください。

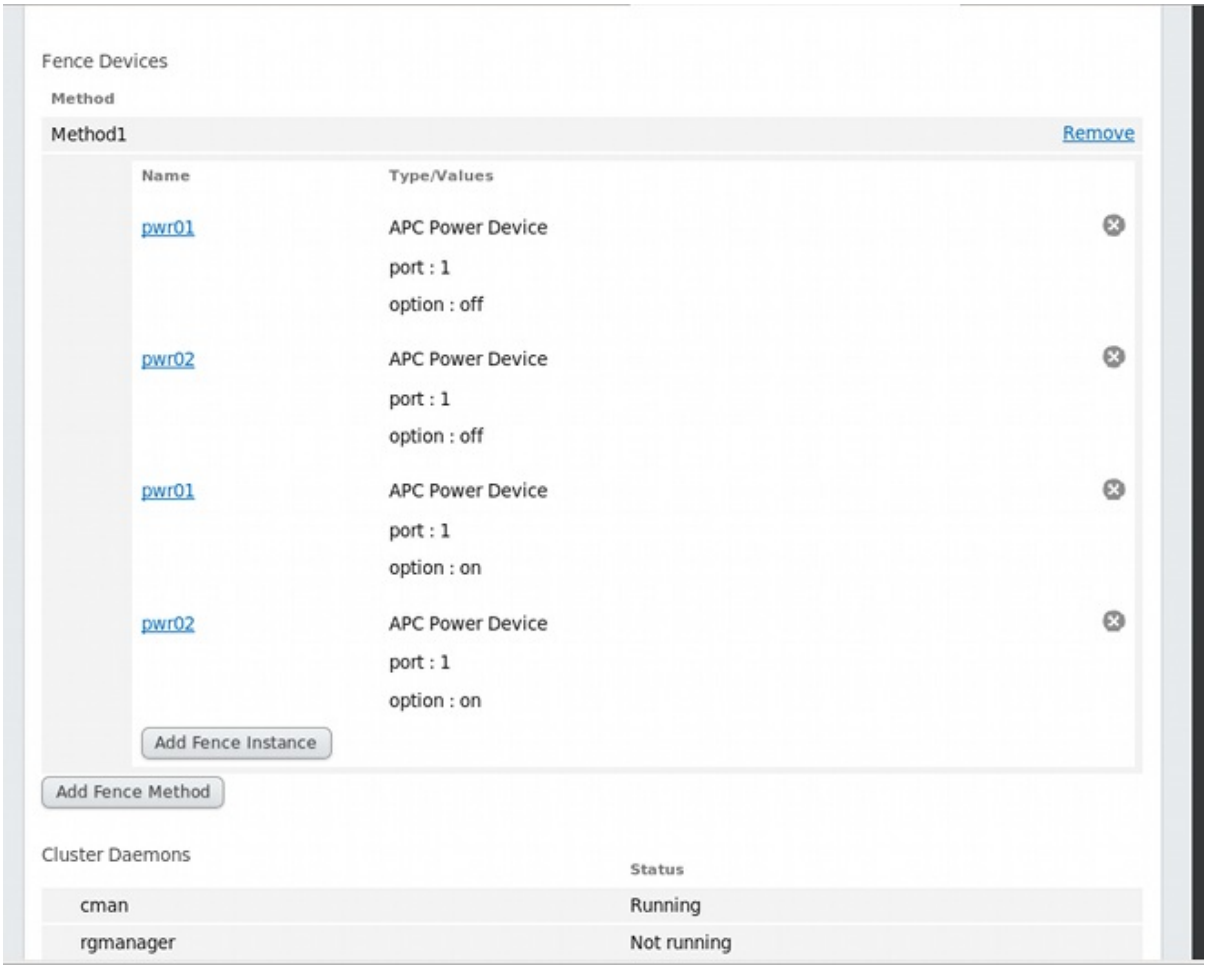


図3.2 二重電源装置を備えている場合のフェンシング設定

3.3.4. フェンスの設定をテストする

Red Hat Enterprise Linux Release 6.4 からは **fence_check** ユーティリティーを使用すると、クラスター内の各ノードに対しフェンスの設定をテストすることができます。

このコマンドを正常に実行できた場合の出力を以下に示します。

```
[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
```

このユーティリティーの詳細については **fence_check**(8) man ページを参照してください。

第4章 フェンスデバイス

本章では、 Red Hat Enterprise Linux High-Availability Add-On で現在対応しているフェンスデバイスについて説明します。

フェンスデバイス、 フェンスデバイスと関連のあるフェンスデバイスのエージェント、 およびフェンスデバイスのパラメータについて解説している表へのリンクを 表4.1「フェンスデバイス要約」 に示します。

表4.1 フェンスデバイス要約

フェンスデバイス	フェンスエージェント	パラメータ詳細へのリンク
APC 電源スイッチ (telnet/SSH)	fence_apc	表4.2「APC 電源スイッチ (telnet/SSH)」
SNMP 経由の APC 電源スイッチ	fence_apc_snmp	表4.3「SNMP 経由の APC 電源スイッチ」
Brocade ファブリックスイッチ	fence_brocade	表4.4「Brocade ファブリックスイッチ」
Cisco MDS	fence_cisco_mds	表4.5「Cisco MDS」
Cisco UCS	fence_cisco_ucs	表4.6「Cisco UCS」
Dell DRAC 5	fence_drac5	表4.7「Dell DRAC 5」
Dell iDRAC	fence_idrac	表4.22「IPMI (Intelligent Platform Management Interface) LAN、Dell iDrac、IBM Integrated Management Module、HPiLO3、HPiLO4」
Eaton Network Power Switch (SNMP Interface)	fence_eaton_snmp	表4.8「Eaton ネットワーク電源コントローラー (SNMP インターフェース) (Red Hat Enterprise Linux 6.4 以降)」
Egenera BladeFrame	fence_egenera	表4.9「Egenera BladeFrame」
ePowerSwitch	fence_eps	表4.10「ePowerSwitch」
Fence kdump	fence_kdump	表4.11「Fence kdump」
Fence virt	fence_virt	表4.12「Fence virt」

フェンスデバイス	フェンスエージェント	パラメータ詳細へのリンク
Fujitsu Siemens Remoteview Service Board (RSB)	fence_rsb	表4.13「Fujitsu Siemens Remoteview Service Board (RSB)」
HP BladeSystem	fence_hpblade	表4.14「HP BladeSystem (Red Hat Enterprise Linux 6.4 以降)」
HP iLO Device (Integrated Lights Out),	fence_ilo	表4.15「HP iLO (Integrated Lights Out) および HP iLO2」
HP iLO2	fence_ilo2	表4.15「HP iLO (Integrated Lights Out) および HP iLO2」
HPiLO3	fence_ilo3	表4.22「IPMI (Intelligent Platform Management Interface) LAN、Dell iDrac、IBM Integrated Management Module、HPiLO3、HPiLO4」
HPiLO4	fence_ilo4	表4.22「IPMI (Intelligent Platform Management Interface) LAN、Dell iDrac、IBM Integrated Management Module、HPiLO3、HPiLO4」
HP iLO (Integrated Lights Out) MP	fence_ilo_mp	表4.16「HP iLO (Integrated Lights Out) MP」
IBM BladeCenter	fence_bladecenter	表4.17「IBM BladeCenter」
IBM BladeCenter SNMP	fence_ibmblade	表4.18「IBM BladeCenter SNMP」
IBM Integrated Management Module (統合管理モジュール)	fence_imm	表4.22「IPMI (Intelligent Platform Management Interface) LAN、Dell iDrac、IBM Integrated Management Module、HPiLO3、HPiLO4」
IBM iPDU	fence_ipdu	表4.19「IBM iPDU (Red Hat Enterprise Linux 6.4 以降)」

フェンスデバイス	フェンスエージェント	パラメータ詳細へのリンク
IF MIB	fence_ifmib	表4.20「IF MIB」
Intel Modular	fence_intelmodular	表4.21「Intel Modular」
IPMI (Intelligent Platform Management Interface) Lan	fence_ipmilan	表4.22「IPMI (Intelligent Platform Management Interface) LAN、Dell iDrac、IBM Integrated Management Module、HPiLO3、HPiLO4」
RHEV-M REST API	fence_rhevm	表4.23「RHEV-M REST API (RHEL 6.2 以降及び RHEV 3.0 以降)」
SCSI フェンシング	fence_scsi	表4.24「SCSI 予約フェンシング」
VMware フェンシング (SOAP インターフェース)	fence_vmware_soap	表4.25「VMware フェンシング (SOAP インターフェース) (Red Hat Enterprise Linux 6.2 以降)」
WTI 電源スイッチ	fence_wti	表4.26「WTI 電源スイッチ」

4.1. TELNET および SSH 経由の APC 電源スイッチ

telnet または SSH 経由の APC 用フェンスエージェント **fence_apc** で使用するフェンスデバイスのパラメータを 表4.2「APC 電源スイッチ (telnet/SSH)」 に示します。

表4.2 APC 電源スイッチ (telnet/SSH)

luci フィールド	cluster.conf 属性	詳細
Name	**name**	APC デバイス名、フェンスデーモンが telnet または ssh 経由でログインするクラスターに接続
IP Address or Hostname	**ipaddr**	デバイスに割り当てている IP アドレスまたはホスト名
IP Port (optional)	**ipport**	デバイスへの接続に使用する TCP ポート、デフォルトのポートは 23 ですが **Use SSH** を選択した場合は 22 になります
Login	**login**	デバイスへのアクセスに使用するログイン名

luci フィールド	cluster.conf 属性	詳細
Password	**passwd**	デバイスへの接続を認証する際に使用するパスワード
Password Script (optional)	**passwd_script**	フェンスデバイスへのアクセス用パスワードを与えるスクリプト (これを使用するとスクリプトの方が **Password** パラメータより優先される)
Power Wait (seconds)	**power_wait**	電源オフまたは電源オンのコマンド発行後の待機秒数
Power Timeout (seconds)	**power_timeout**	電源オンまたは電源オフのコマンドを発行後、状態変更のテストまで待機させる秒数、デフォルト値は 20 です
Shell Timeout (seconds)	**shell_timeout**	コマンド発行後にコマンドプロンプトを待機させる秒数、デフォルト値は 3 です
Login Timeout (seconds)	**login_timeout**	ログイン後にコマンドプロンプトを待機させる秒数、デフォルト値は 5 です
Times to Retry Power On Operation	**retry_on**	電源オンの動作を再試行する回数、デフォルト値は 1 です
Port	**port**	ポート
Switch (optional)	**switch**	ノードに接続している APCスイッチのスイッチ番号、デイジーチェーン配線でスイッチが複数ある場合
Delay (optional)	**delay**	フェンシング開始までに待機させる秒数、デフォルト値は 0 です
Use SSH	**secure**	デバイスへのアクセスに SSH を使用するかどうかを指定します、SSH を使用する場合はパスワード、パスワードスクリプト、識別ファイルのいずれかを指定する必要があります
SSH オプション	**ssh_options**	使用する SSH オプション、デフォルト値は **-1 -c blowfish** です
Path to SSH Identity File	**identity_file**	SSH の識別ファイル

APC 電源スイッチのフェンスデバイスを追加する際に使用する設定画面を 図4.1「APC 電源スイッチ」 に示します。

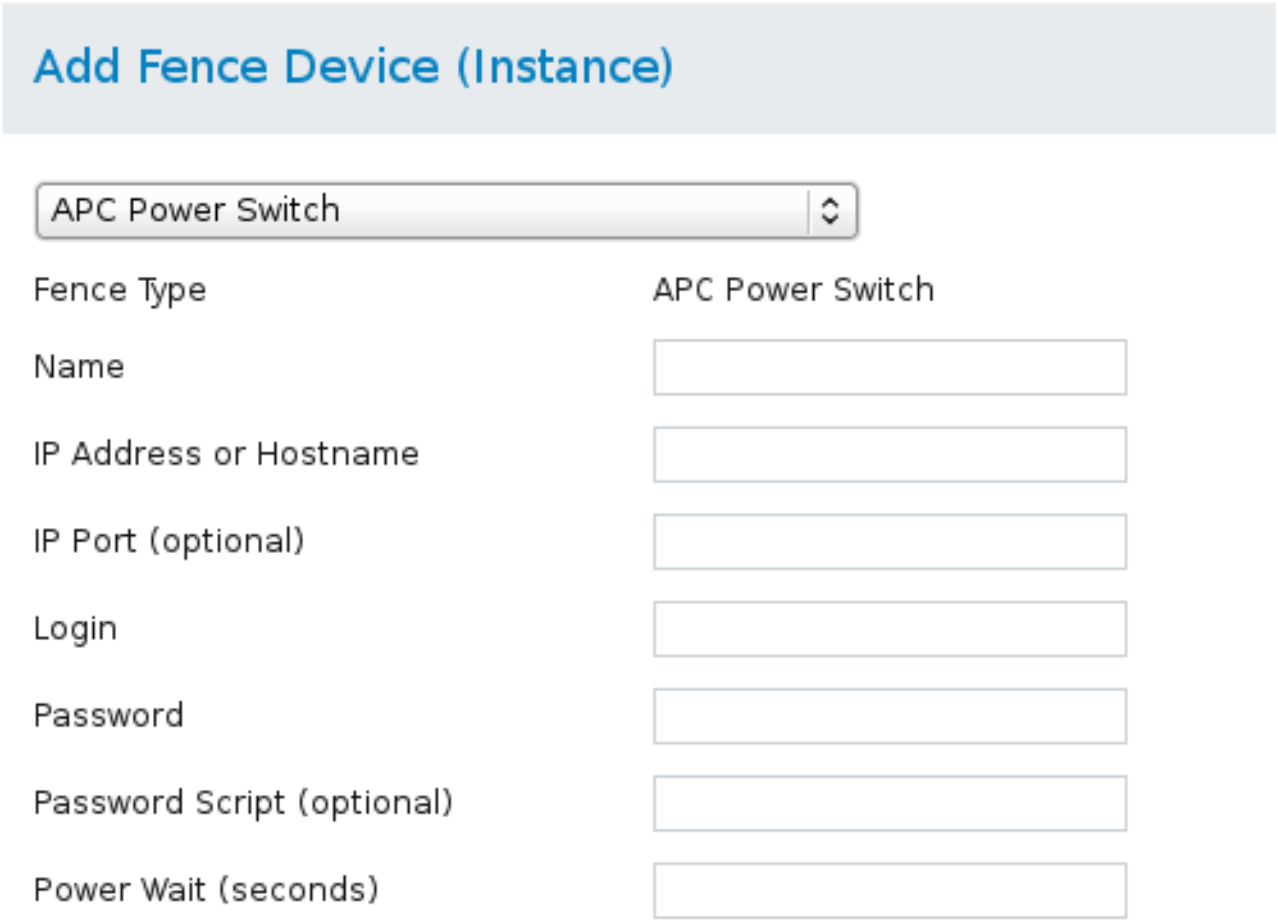


図4.1 APC 電源スイッチ

APC デバイス用のフェンスデバイスインスタンスを作成するコマンドです。

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

cluster.conf ファイル内の **fence_apc** デバイス用のエントリーを以下に示します。

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

4.2. SNMP 経由の APC 電源スイッチ

SNMP プロトコル経由で SNP デバイスにログインする APC のフェンスエージェント **fence_apc_snmp** で使用するフェンスデバイスのパラメータを 表4.3「SNMP 経由の APC 電源スイッチ」 に示します。

表4.3 SNMP 経由の APC 電源スイッチ

luci フィールド	cluster.conf 属性	詳細
Name	**name**	APC デバイス名、フェンスデーモンが SNMP プロトコル経由でログインするクラスターに接続
IP Address or Hostname	**ipaddr**	デバイスに割り当てている IP アドレスまたはホスト名
UDP/TCP Port	**udpport**	デバイスとの接続に使用する UDP/TCP ポート、 デフォルト値は 161
Login	**login**	デバイスへのアクセスに使用するログイン名
Password	**passwd**	デバイスへの接続を認証する際に使用するパスワード
Password Script (optional)	**passwd_script**	フェンスデバイスへのアクセス用パスワードを与えるスクリプト (これを使用するとスクリプトの方が **Password** パラメータより優先される)
SNMP Version	**snmp_version**	使用する SNMP バージョン (1、2c、3)、デフォルト値は 1
SNMP Community	**community**	SNMP コミュニティ文字列、デフォルト値は **private**
SNMP Security Level	**snmp_sec_level**	SNMP セキュリティレベル (noAuthNoPriv、authNoPriv、authPriv)
SNMP Authentication Protocol	**snmp_auth_prot**	SNMP 認証プロトコル (MD5、SHA)
SNMP Privacy Protocol	**snmp_priv_prot**	SNMP プライバシープロトコル (DES、AES)
SNMP Privacy Protocol Password	**snmp_priv_passwd**	SNMP プライバシープロトコルのパスワード
SNMP Privacy Protocol Script	**snmp_priv_passwd_script**	SNMP プライバシープロトコル用パスワードを与えるスクリプト (これを使用するとスクリプトの方が **SNMP privacy protocol password** パラメータより優先される)
Power Wait (seconds)	**power_wait**	電源オフまたは電源オンのコマンド発行後の待機秒数
Power Timeout (seconds)	**power_timeout**	電源オンまたは電源オフのコマンドを発行後、状態変更のテストまで待機させる秒数、デフォルト値は 20 です

luci フィールド	cluster.conf 属性	詳細
Shell Timeout (seconds)	**shell_timeout**	コマンド発行後にコマンドプロンプトを待機させる秒数、デフォルト値は 3 です
Login Timeout (seconds)	**login_timeout**	ログイン後にコマンドプロンプトを待機させる秒数、デフォルト値は 5 です
Times to Retry Power On Operation	**retry_on**	電源オンの動作を再試行する回数、デフォルト値は 1 です
Port (Outlet) Number	**port**	ポート
Delay (optional)	**delay**	フェンシング開始までに待機させる秒数、デフォルト値は 0 です

APC 電源スイッチのフェンスデバイスを追加する際に使用する設定画面を 図4.2「SNMP 経由の APC 電源スイッチ」 に示します。

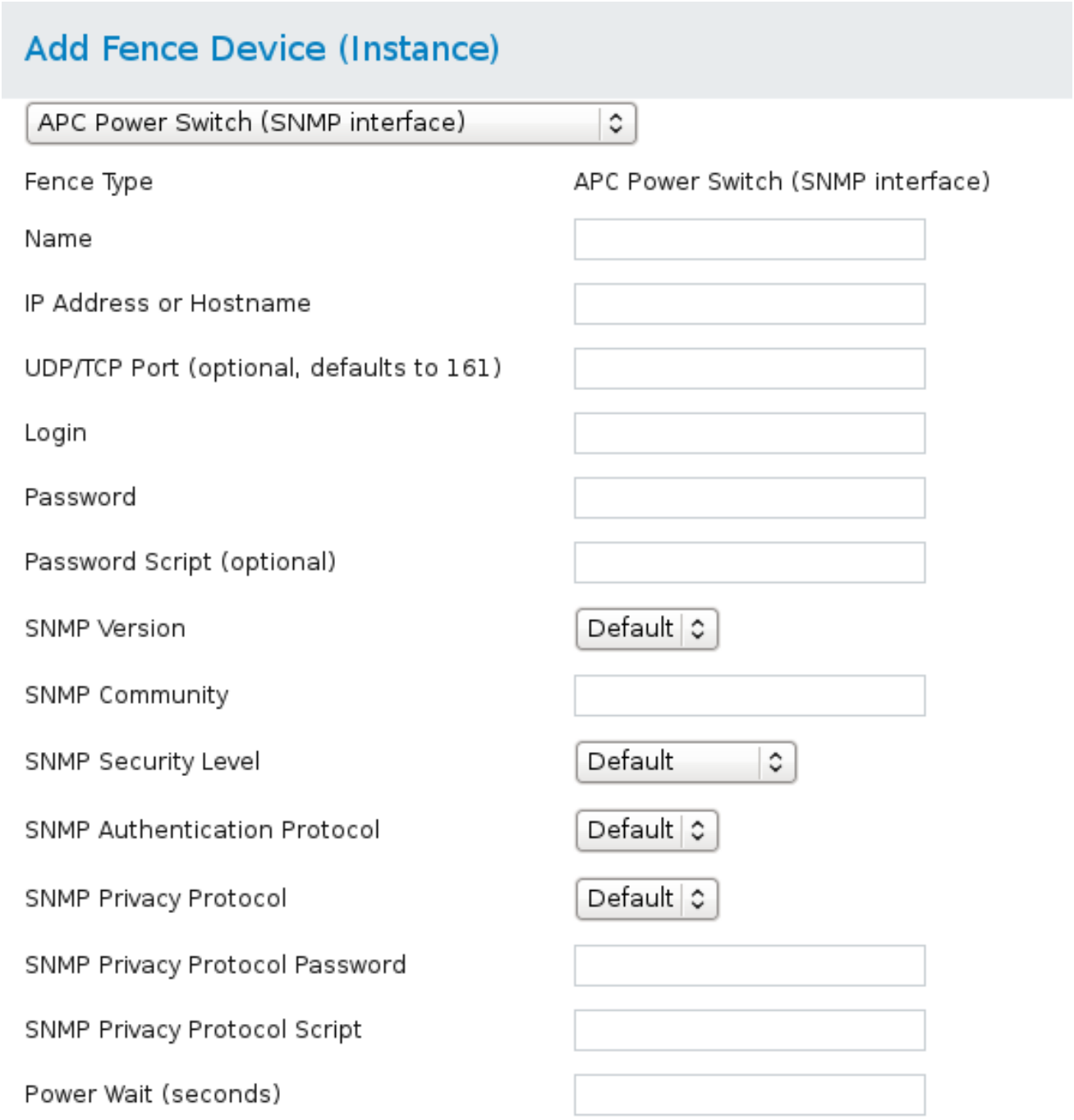


図4.2 SNMP 経由の APC 電源スイッチ

cluster.conf ファイル内の **fence_apc_snmp** デバイス用のエントリーを以下に示します。

```
 <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 ファブリックスイッチ

Brocade FC スイッチのフェンスエージェント **fence_brocade** で使用するフェンスデバイスのパラメータを 表4.4「Brocade ファブリックスイッチ」 に示します。

表4.4 Brocade ファブリックスイッチ

luci フィールド	cluster.conf 属性	詳細
Name	**name**	クラスターに接続している Brocade デバイス名
IP Address or Hostname	**ipaddr**	デバイスに割り当てている IP アドレス
Login	**login**	デバイスへのアクセスに使用するログイン名
Password	**passwd**	デバイスへの接続を認証する際に使用するパスワード
Password Script (optional)	**passwd_script**	フェンスデバイスへのアクセス用パスワードを与えるスクリプト (これを使用するとスクリプトの方が **Password** パラメータより優先される)
Force IP Family	**inet4_only**, **inet6_only**	エージェントのアドレスの使用を IPv4 または IPv6 に制限します
Force Command Prompt	**cmd_prompt**	使用するコマンドプロンプト、 デフォルト値は '\$'
Power Wait (seconds)	**power_wait**	電源オフまたは電源オンのコマンド発行後の待機秒数
Power Timeout (seconds)	**power_timeout**	電源オンまたは電源オフのコマンドを発行後、状態変更のテストまで待機させる秒数、デフォルト値は 20 です
Shell Timeout (seconds)	**shell_timeout**	コマンド発行後にコマンドプロンプトを待機させる秒数、デフォルト値は 3 です
Login Timeout (seconds)	**login_timeout**	ログイン後にコマンドプロンプトを待機させる秒数、デフォルト値は 5 です
Times to Retry Power On Operation	**retry_on**	電源オンの動作を再試行する回数、デフォルト値は 1 です
Port	**port**	スイッチ出口番号
Delay (optional)	**delay**	フェンシング開始までに待機させる秒数、デフォルト値は 0 です

luci フィールド	cluster.conf 属性	詳細
Use SSH	**secure**	デバイスへのアクセスに SSH を使用するかどうかを指定します、SSH を使用する場合はパスワード、パスワードスクリプト、識別ファイルのいずれかを指定する必要があります
SSH オプション	**ssh_options**	使用する SSH オプション、デフォルト値は **-1 -c blowfish** です
Path to SSH Identity File	**identity_file**	SSH の識別ファイル
Unfencing	**unfence** section of the cluster configuration file	有効にすると、フェンス済みのノードは再起動が完了するまで再度有効にならないようにします。電源フェンス以外の方法を使用する場合 (SAN ストレージフェンシング) に必要なパラメータです。アンフェンシングを必要とするデバイスを設定する際には、最初にクラスターを停止し、デバイスおよびアンフェンシングを含むすべての設定を追加してから、その後クラスターを開始しなければなりません。ノードにアンフェンシングを設定する方法については **fence_node**(8) の man ページを参照してください。

Brocade ファブリックスイッチのフェンスデバイスを追加する際に使用する設定画面を 図 4.3「Brocade ファブリックスイッチ」 に示します。

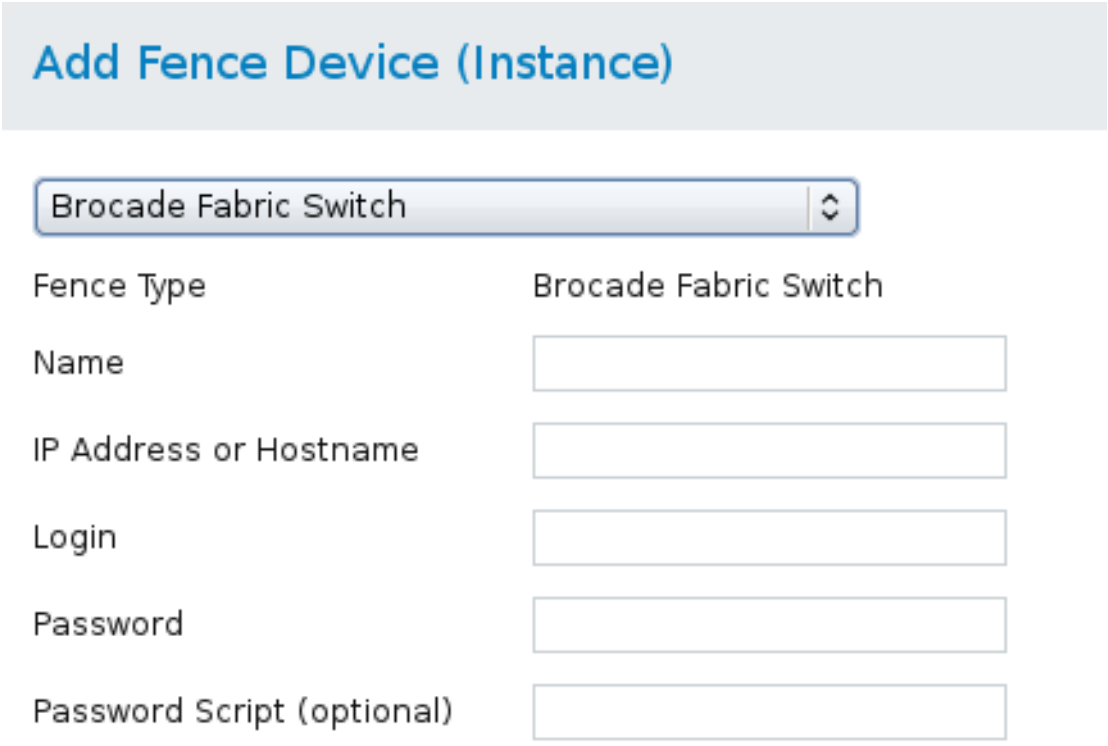


図4.3 Brocade ファブリックスイッチ

Brocade デバイス用のフェンスデバイスインスタンスを作成するコマンドです。

-

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

cluster.conf ファイル内の **fence_brocade** デバイス用のエントリーを以下に示します。

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

4.4. CISCO MDS

Cisco MDS のフェンスエージェント **fence_cisco_mds** で使用するフェンスデバイスのパラメータを 表4.5「Cisco MDS」 に示します。

表4.5 Cisco MDS

luci フィールド	cluster.conf 属性	詳細
Name	**name**	SNMP が有効になっている Cisco MDS 9000 シリーズデバイスの名前
IP Address or Hostname	**ipaddr**	デバイスに割り当てている IP アドレスまたはホスト名
UDP/TCP port (optional)	**udpport**	デバイスとの接続に使用する UDP/TCP ポート、 デフォルト値は 161
Login	**login**	デバイスへのアクセスに使用するログイン名
Password	**passwd**	デバイスへの接続を認証する際に使用するパスワード
Password Script (optional)	**passwd_script**	フェンスデバイスへのアクセス用パスワードを与えるスクリプト (これを使用するとスクリプトの方が **Password** パラメータより優先される)
SNMP Version	**snmp_version**	使用する SNMP バージョン (1、2c、3)
SNMP Community	**community**	SNMP コミュニティ文字列
SNMP Security Level	**snmp_sec_level**	SNMP セキュリティレベル (noAuthNoPriv、authNoPriv、authPriv)

luci フィールド	cluster.conf 属性	詳細
SNMP Authentication Protocol	snmp_auth_prot	SNMP 認証プロトコル (MD5、SHA)
SNMP Privacy Protocol	snmp_priv_prot	SNMP プライバシープロトコル (DES、AES)
SNMP Privacy Protocol Password	snmp_priv_passwd	SNMP プライバシープロトコルのパスワード
SNMP Privacy Protocol Script	snmp_priv_passwd_script	SNMP プライバシープロトコル用パスワードを与えるスクリプト (これを使用するとスクリプトの方が **SNMP privacy protocol password** パラメータより優先される)
Power Wait (seconds)	power_wait	電源オフまたは電源オンのコマンド発行後の待機秒数
Power Timeout (seconds)	power_timeout	電源オンまたは電源オフのコマンドを発行後、状態変更のテストまで待機させる秒数、デフォルト値は 20 です
Shell Timeout (seconds)	shell_timeout	コマンド発行後にコマンドプロンプトを待機させる秒数、デフォルト値は 3 です
Login Timeout (seconds)	login_timeout	ログイン後にコマンドプロンプトを待機させる秒数、デフォルト値は 5 です
Times to Retry Power On Operation	retry_on	電源オンの動作を再試行する回数、デフォルト値は 1 です
Port (Outlet) Number	port	ポート
Delay (optional)	delay	フェンシング開始までに待機させる秒数、デフォルト値は 0 です

Cisco MDS のフェンスデバイスを追加する際に使用する設定画面を 図4.4「Cisco MDS」 に示します。

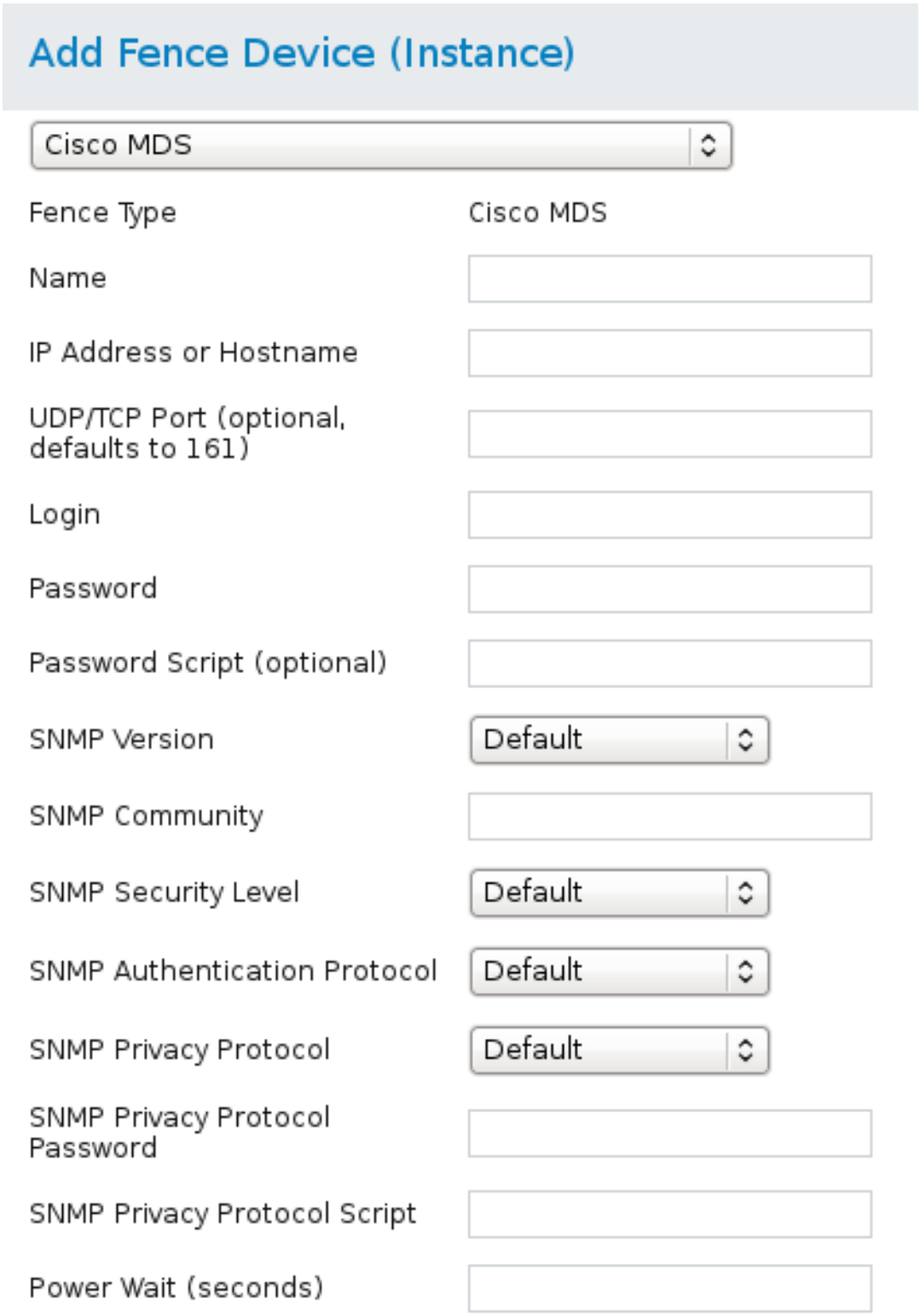


図4.4 Cisco MDS

Cisco MDS デバイス用のフェンスデバイスインスタンスを作成するコマンドです。

```
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
```

cluster.conf ファイル内の **fence_cisco_mds** デバイス用のエントリーを以下に示します。

```
```

```
 <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

Cisco UCS のフェンスエージェント **fence_cisco_ucs** で使用するフェンスデバイスのパラメータを 表4.6「Cisco UCS」 に示します。

表4.6 Cisco UCS

luci フィールド	cluster.conf 属性	詳細
Name	**name**	Cisco UCS デバイスの名前
IP Address or Hostname	**ipaddr**	デバイスに割り当てている IP アドレスまたはホスト名
IP Port (optional)	**ipport**	デバイスへの接続に使用する TCP ポート
Login	**login**	デバイスへのアクセスに使用するログイン名
Password	**passwd**	デバイスへの接続を認証する際に使用するパスワード
Password Script (optional)	**passwd_script**	フェンスデバイスへのアクセス用パスワードを与えるスクリプト (これを使用するとスクリプトの方が **Password** パラメータより優先される)
Use SSL	**ssl**	デバイスとの通信に SSL 接続を使用する
Sub-Organization	**suborg**	サブ組織へのアクセスに必要な追加のパス
Power Wait (seconds)	**power_wait**	電源オフまたは電源オンのコマンド発行後の待機秒数
Power Timeout (seconds)	**power_timeout**	電源オンまたは電源オフのコマンドを発行後、状態変更のテストまで待機させる秒数、デフォルト値は 20 です
Shell Timeout (seconds)	**shell_timeout**	コマンド発行後にコマンドプロンプトを待機させる秒数、デフォルト値は 3 です
Login Timeout (seconds)	**login_timeout**	ログイン後にコマンドプロンプトを待機させる秒数、デフォルト値は 5 です

luci フィールド	cluster.conf 属性	詳細
Times to Retry Power On Operation	retry_on	電源オンの動作を再試行する回数、デフォルト値は 1 です
Port (Outlet) Number	port	仮想マシン名
Delay (optional)	delay	フェンシング開始までに待機させる秒数、デフォルト値は 0 です

Cisco UCS のフェンスデバイスを追加する際に使用する設定画面を 図4.5「Cisco UCS」 に示します。

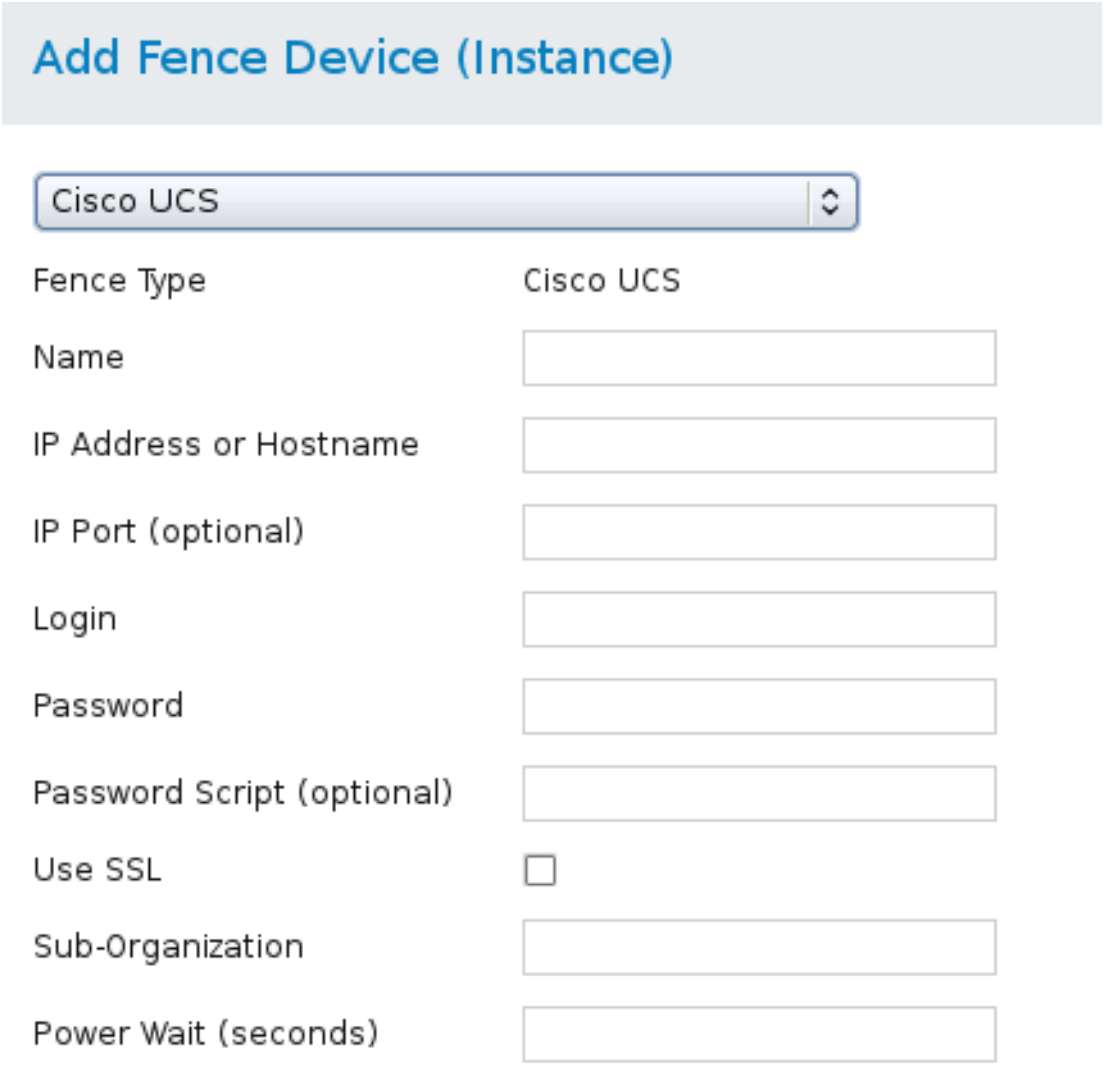


図4.5 Cisco UCS

Cisco UCS デバイス用のフェンスデバイスインスタンスを作成するコマンドです。

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

Conga または **ccs** で作成した **cluster.conf** ファイル内の **fence_cisco_ucs** デバイス用のエントリーを以下に示します。

```
 <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

Dell DRAC 5 のフェンスエージェント **fence_drac5** で使用するフェンスデバイスのパラメータを 表4.7「Dell DRAC 5」 に示します。

表4.7 Dell DRAC 5

luci フィールド	cluster.conf 属性	詳細
Name	**name**	DRAC に割り当てる名前
IP Address or Hostname	**ipaddr**	DRAC に割り当てている IP アドレスまたはホスト名
IP Port (optional)	**ipport**	デバイスへの接続に使用する TCP ポート
Login	**login**	DRAC へのアクセスに使用するログイン名
Password	**passwd**	DRAC への接続を認証する際に使用するパスワード
Password Script (optional)	**passwd_script**	フェンスデバイスへのアクセス用パスワードを与えるスクリプト (これを使用するとスクリプトの方が **Password** パラメータより優先される)
Use SSH	**secure**	デバイスへのアクセスに SSH を使用するかどうかを指定、SSH を使用する場合はパスワード、パスワードスクリプト、識別ファイルのいずれかを指定する必要があります
SSH オプション	**ssh_options**	使用する SSH オプション、デフォルト値は **-1 -c blowfish** です
Path to SSH Identity File	**identity_file**	SSH の識別ファイル

luci フィールド	cluster.conf 属性	詳細
Module Name	module_name	オプション: この DRAC 用のモジュール名、複数の DRAC モジュールを使用する場合
Force Command Prompt	cmd_prompt	使用するコマンドプロンプト、 デフォルト値は '\$'
Power Wait (seconds)	power_wait	電源オフまたは電源オンのコマンド発行後の待機秒数
Delay (seconds)	delay	フェンシング開始までに待機させる秒数、デフォルト値は 0 です
Power Timeout (seconds)	power_timeout	電源オンまたは電源オフのコマンドを発行後、状態変更のテストまで待機させる秒数、デフォルト値は 20 です
Shell Timeout (seconds)	shell_timeout	コマンド発行後にコマンドプロンプトを待機させる秒数、デフォルト値は 3 です
Login Timeout (seconds)	login_timeout	ログイン後にコマンドプロンプトを待機させる秒数、デフォルト値は 5 です
Times to Retry Power On Operation	retry_on	電源オンの動作を再試行する回数、デフォルト値は 1 です

Dell Drac 5 のデバイスを追加する際に使用する設定画面を 図4.6「Dell Drac 5」 に示します。

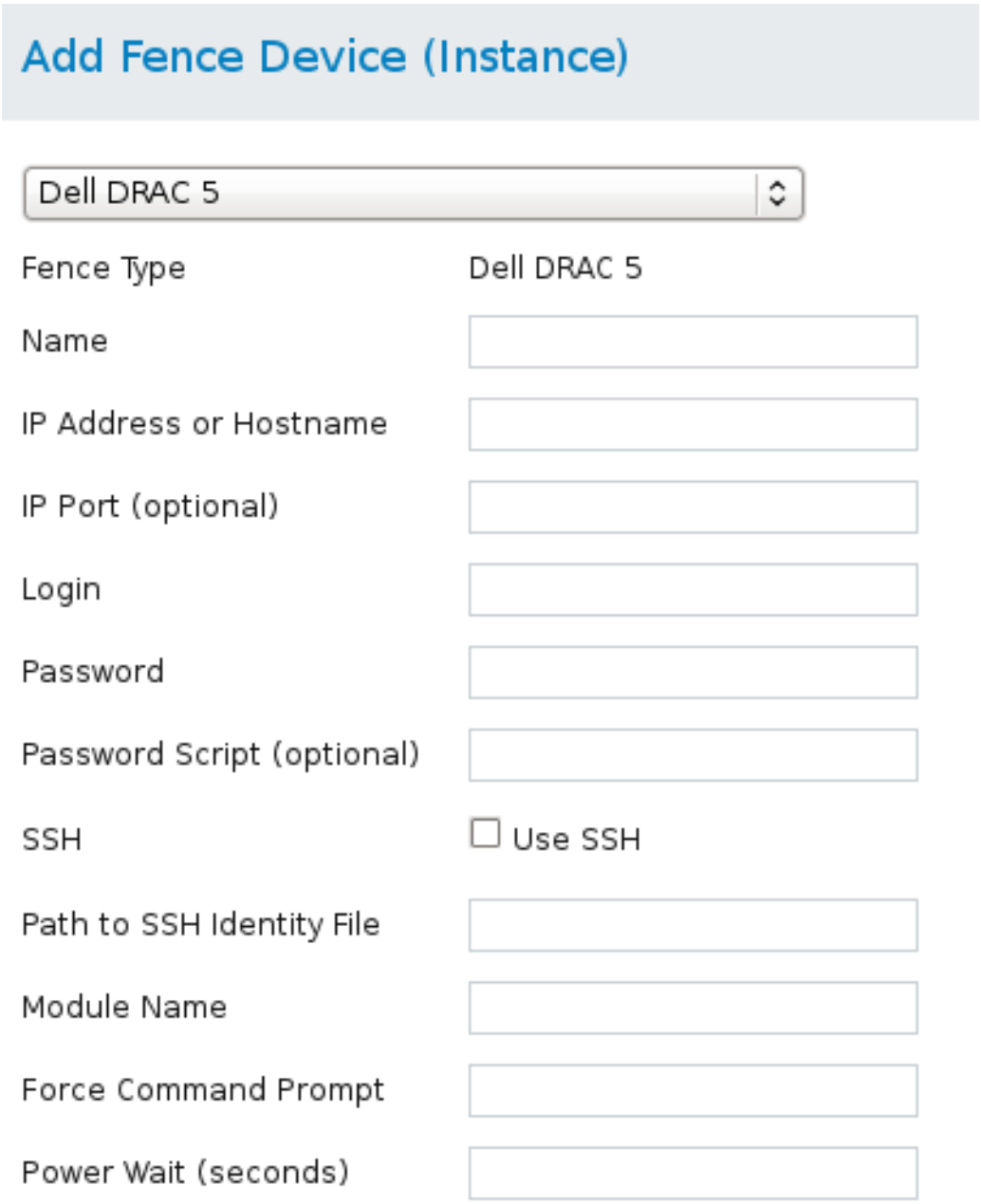


図4.6 Dell Drac 5

Dell Drac 5 デバイス用のフェンスデバイスインスタンスを作成するコマンドです。

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

cluster.conf ファイル内の **fence_drac5** デバイス用のエントリーを以下に示します。

```
 <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 ネットワーク電源スイッチ

SNMP 経由の Eaton ネットワーク電源スイッチのフェンスエージェント **fence_eaton_snmp** で使用するフェンスデバイスのパラメータを 表4.8「Eaton ネットワーク電源コントローラー (SNMP インターフェース) (Red Hat Enterprise Linux 6.4 以降)」 に示します。

表4.8 Eaton ネットワーク電源コントローラー (SNMP インターフェース) (Red Hat Enterprise Linux 6.4 以降)

luci フィールド	cluster.conf 属性	詳細
Name	**name**	クラスターに接続している Eaton ネットワーク電源スイッチの名前
IP Address or Hostname	**ipaddr**	デバイスに割り当てている IP アドレスまたはホスト名
UDP/TCP Port (optional)	**udpport**	デバイスとの接続に使用する UDP/TCP ポート、 デフォルト値は 161
Login	**login**	デバイスへのアクセスに使用するログイン名
Password	**passwd**	デバイスへの接続を認証する際に使用するパスワード
Password Script (optional)	**passwd_script**	フェンスデバイスへのアクセス用パスワードを与えるスクリプト (これを使用するとスクリプトの方が **Password** パラメータより優先される)
SNMP Version	**snmp_version**	使用する SNMP バージョン (1、 2c、 3)、 デフォルト値は 1
SNMP Community	**community**	SNMP コミュニティ文字列、 デフォルト値は **private**
SNMP Security Level	**snmp_sec_level**	SNMP セキュリティレベル (noAuthNoPriv、authNoPriv、authPriv)
SNMP Authentication Protocol	**snmp_auth_prot**	SNMP 認証プロトコル (MD5、SHA)
SNMP Privacy Protocol	**snmp_priv_prot**	SNMP プライバシープロトコル (DES、AES)
SNMP Privacy Protocol Password	**snmp_priv_passwd**	SNMP プライバシープロトコルのパスワード
SNMP Privacy Protocol Script	**snmp_priv_passwd_script**	SNMP プライバシープロトコル用パスワードを与えるスクリプト (これを使用するとスクリプトの方が **SNMP privacy protocol password** パラメータより優先される)

luci フィールド	cluster.conf 属性	詳細
Power wait (seconds)	power_wait	電源オフまたは電源オンのコマンド発行後の待機秒数
Power Timeout (seconds)	power_timeout	電源オンまたは電源オフのコマンドを発行後、状態変更のテストまで待機させる秒数、デフォルト値は 20 です
Shell Timeout (seconds)	shell_timeout	コマンド発行後にコマンドプロンプトを待機させる秒数、デフォルト値は 3 です
Login Timeout (seconds)	login_timeout	ログイン後にコマンドプロンプトを待機させる秒数、デフォルト値は 5 です
Times to Retry Power On Operation	retry_on	電源オンの動作を再試行する回数、デフォルト値は 1 です
Port (Outlet) Number	port	物理的なプラグ番号または仮想マシン名、 このパラメータは常に必須
Delay (optional)	delay	フェンシング開始までに待機させる秒数、デフォルト値は 0 です

Eaton ネットワーク電源スイッチのフェンスデバイスを追加する際に使用する設定画面を 図4.7「Eaton ネットワーク電源スイッチ」 に示します。

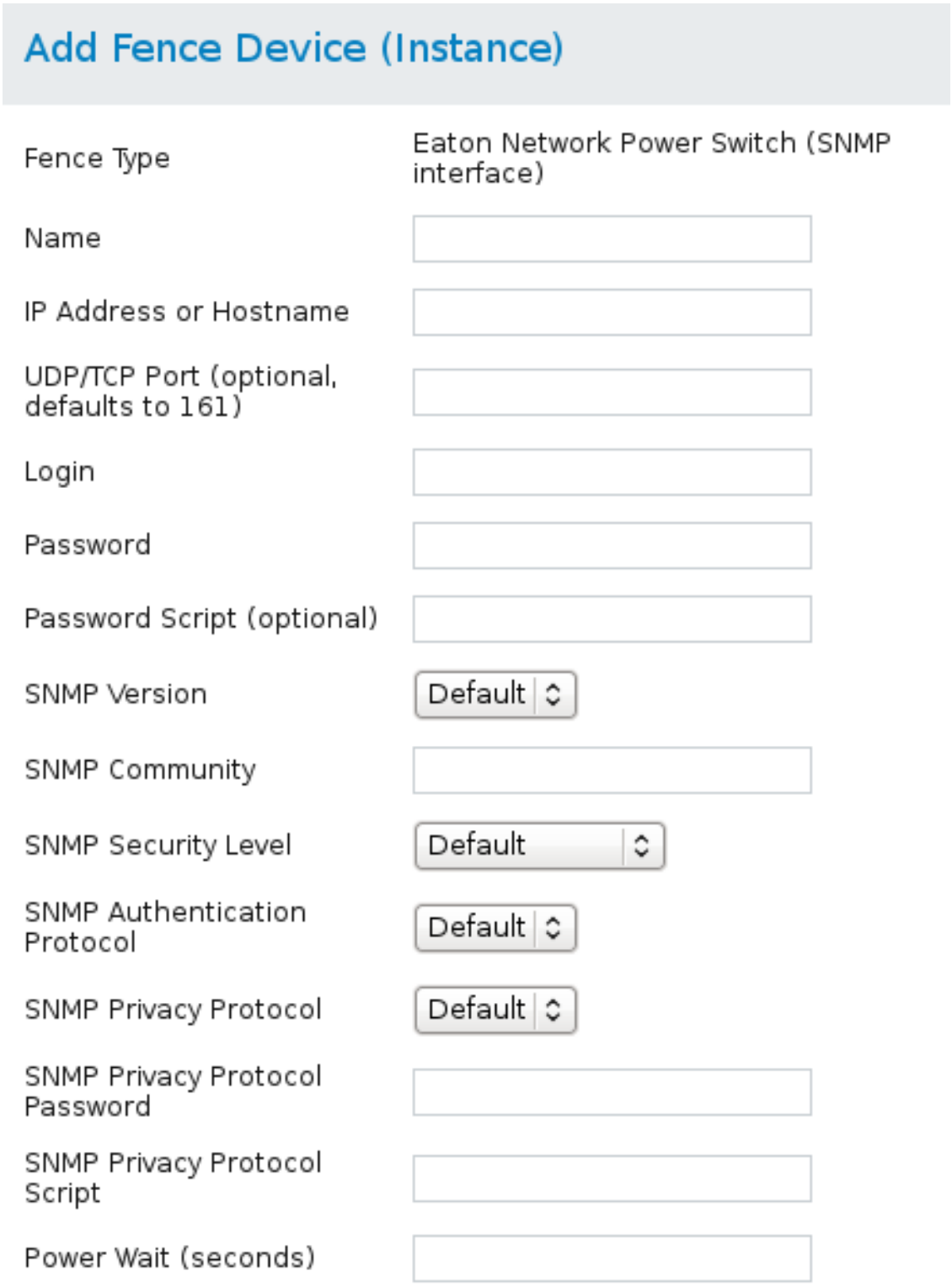


図4.7 Eaton ネットワーク電源スイッチ

Eaton ネットワーク電源スイッチのデバイス用にフェンスデバイスインスタンスを作成するコマンドです。

```
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
```

cluster.conf ファイル内の **fence_eaton_snmp** デバイス用のエントリーを以下に示します。

```
 <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

Egenera BladeFrame のフェンスエージェント **fence_egenera** で使用するフェンスデバイスのパラメータを 表4.9「Egenera BladeFrame」 に示します。

表4.9 Egenera BladeFrame

luci フィールド	cluster.conf 属性	説明
Name	**name**	クラスターに接続している Egenera BladeFrame デバイス名
CServer	**cserver**	デバイスに割り当てているホスト名とオプションとして **username@hostname** 形式のユーザー名 (fence_egenera(8) の man ページを参照)
ESH Path (optional)	**esh**	cserver 上の esh コマンドへのパス (デフォルトは /opt/panmgr/bin/esh)
Username	**user**	ログイン名、 デフォルト値は **root**
lpan	**lpan**	デバイスの論理プロセスエリアネットワーク (LPAN)
pserver	**pserver**	デバイスのプロセッシングブレード (pserver) 名
Delay (optional)	**delay**	フェンシング開始までに待機させる秒数、デフォルト値は 0 です
Unfencing	**unfence** section of the cluster configuration file	有効にすると、フェンス済みのノードは再起動が完了するまで再度有効にならないようにします。電源フェンス以外の方法を使用する場合 (SAN/ストレージフェンシング) に必要なパラメータです。アンフェンシングを必要とするデバイスを設定する際には、最初にクラスターを停止し、デバイスおよびアンフェンシングを含むすべての設定を追加してから、その後クラスターを開始しなければなりません。ノードにアンフェンシングを設定する方法については **fence_node**(8) の man ページを参照してください。

Egenera BladeFrame のフェンスデバイスを追加する際に使用する設定画面を図4.8「Egenera BladeFrame」 に示します。

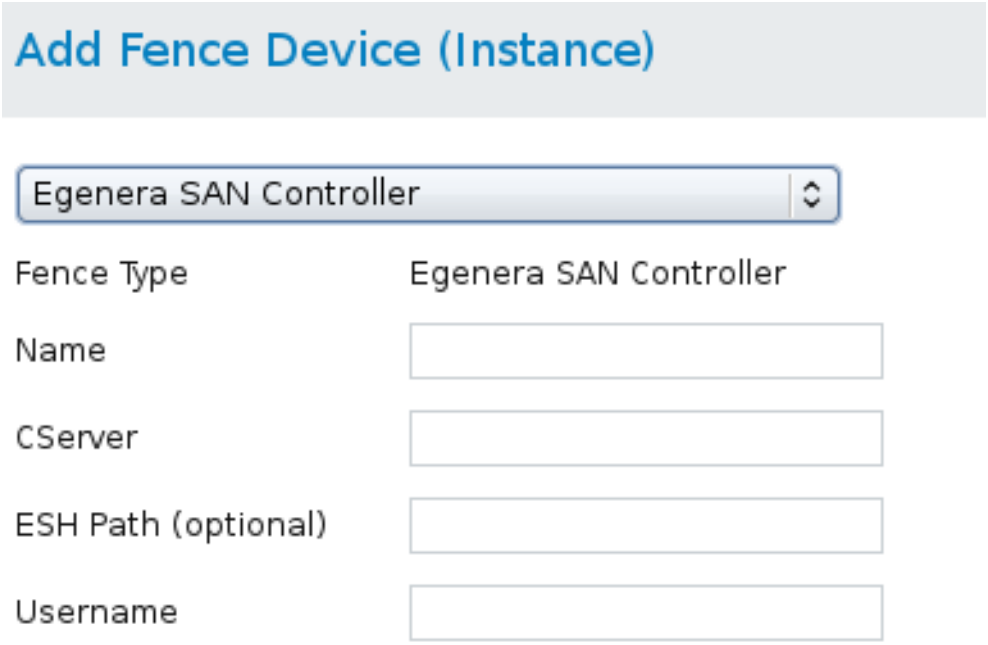


図4.8 Egenera BladeFrame

Egenera BladeFrame デバイス用のフェンスデバイスインスタンスを作成するコマンドです。

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

cluster.conf ファイル内の **fence_egenera** デバイス用のエントリーを以下に示します。

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

4.9. EPOWERSWITCH

ePowerSwitch のフェンスエージェント **fence_eps** で使用するフェンスデバイスのパラメータを 表4.10「ePowerSwitch」 に示します。

表4.10 ePowerSwitch

luci フィールド	cluster.conf 属性	説明
Name	**name**	クラスターに接続している ePowerSwitch デバイスの名前
IP Address or Hostname	**ipaddr**	デバイスに割り当てている IP アドレスまたはホスト名
Login	**login**	デバイスへのアクセスに使用するログイン名

luci フィールド	cluster.conf 属性	説明
Password	passwd	デバイスへの接続を認証する際に使用するパスワード
Password Script (optional)	passwd_script	フェンスデバイスへのアクセス用パスワードを与えるスクリプト (これを使用するとスクリプトの方が **Password** パラメータより優先される)
Name of Hidden Page	hidden_page	デバイス用の非表示ページの名前
Times to Retry Power On Operation	retry_on	電源オンの動作を再試行する回数、デフォルト値は 1 です
Port (Outlet) Number	port	物理的なプラグ番号または仮想マシン名
Delay (optional)	delay	フェンシング開始までに待機させる秒数、デフォルト値は 0 です

ePowerSwitch のフェンスデバイスを追加する際に使用する設定画面を図4.9「ePowerSwitch」 に示します。

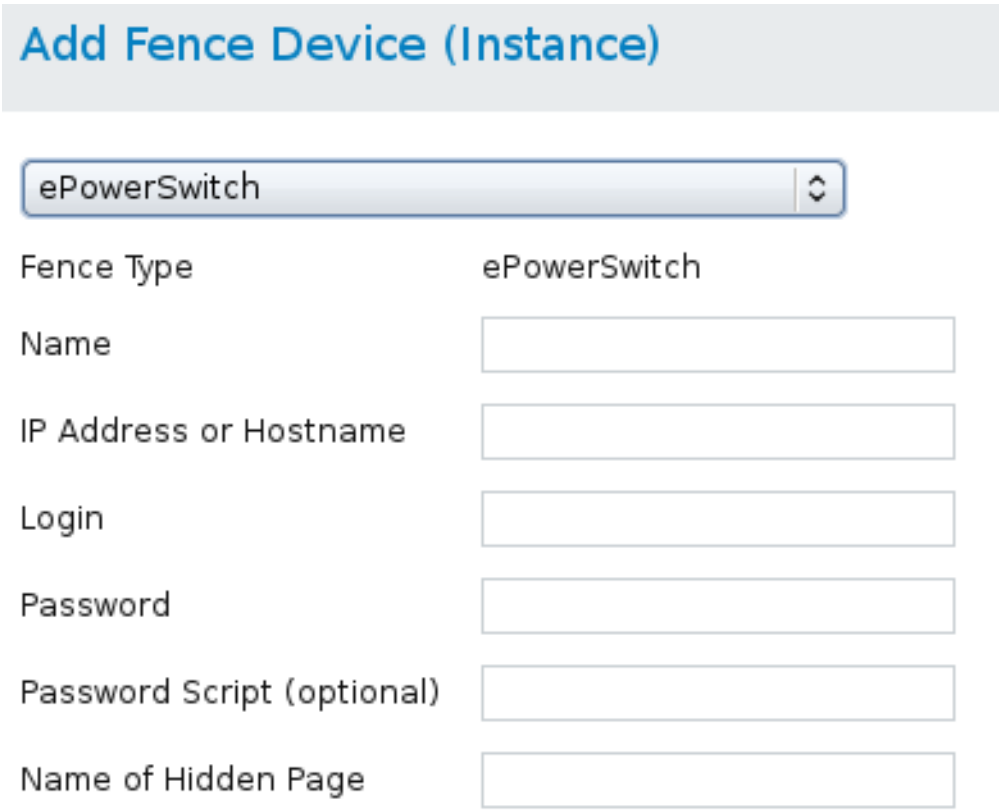


図4.9 ePowerSwitch

ePowerSwitch デバイス用のフェンスデバイスインスタンスを作成するコマンドです。

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

cluster.conf ファイル内の **fence_eps** デバイス用のエントリーを以下に示します。

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

4.10. FENCE KDUMP

kdump クラッシュリカバリーサービスのフェンスエージェント **fence_kdump** で使用されるフェンスデバイスのパラメータを 表4.11「Fence kdump」 に示します。**fence_kdump** は従来のフェンシングメソッドの代替ではありません。**fence_kdump** で行えるのはノードが **kdump** クラッシュリカバリーサービスに入ったことを検知するだけです。従来の電源フェンスメソッドで排他処理が行われる前に **kdump** クラッシュリカバリーサービスを完了させることができます。

表4.11 Fence kdump

luci フィールド	cluster.conf 属性	詳細
Name	**name**	**fence_kdump** デバイスの名前です
IP Family	**family**	IP ネットワークファミリー、デフォルト値は **auto** です
IP Port (optional)	**ipport**	**fence_kdump** エージェントによってメッセージのリッスンに使用される IP ポート番号、デフォルト値は 7410 です
Operation Timeout (秒) (オプション)	**timeout**	障害が発生したノードからのメッセージを待機する秒数です
Node name	**nodename**	フェンシングを行うノードの名前または IP アドレスです

4.11. FENCE VIRT

Fence virt フェンスデバイスのフェンスエージェント **fence_virt** で使用するフェンスデバイスのパラメータを 表4.12「Fence virt」 に示します。

表4.12 Fence virt

luci フィールド	cluster.conf 属性	説明
Name	**name**	Fence virt フェンスデバイスの名前です
Serial Device	**serial_device**	ホスト側で各ドメインの設定ファイル内にシリアルデバイスをマッピングする必要があります (**fence_virt.conf** の man ページを参照)、このフィールドを指定すると **fence_virt** フェンシングエージェントはシリアルモードで動作し、指定しないと VM チャンネルモードで動作します
Serial Parameters	**serial_params**	シリアルパラメータ、 デフォルトは 115200、 8N1 です
VM Channel IP Address	**channel_address**	チャンネル IP、 デフォルト値は 10.0.2.179 です
Port or Domain (deprecated)	**port**	フェンシングを行なう仮想マシンです (ドメインの UUID または名前)
	ipport	チャンネルポート、 デフォルト値は 1229 (**luci** でこのフェンスデバイスを設定する場合に使用される値) です
Timeout	**timeout**	フェンシングのタイムアウト、秒単位、デフォルト値は 30 です

Fence Virt のフェンスデバイスを追加する際に使用する設定画面を 図4.10「Fence Virt」 に示します。

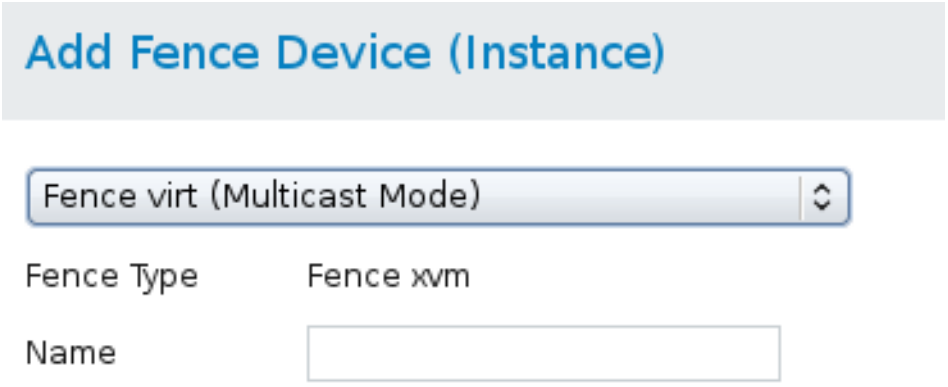


図4.10 Fence Virt

Fence Virt デバイス用のフェンスデバイスインスタンスを作成するコマンドです。

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

cluster.conf ファイル内の **fence_virt** デバイス用のエントリーを以下に示します。

```
```

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

4.12. FUJITSU-SIEMENS REMOTEVIEW SERVICE BOARD (RSB)

Fujitsu-Siemens RemoteView Service Board (RSB) のフェンスエージェント **fence_rsb** で使用するフェンスデバイスのパラメータを 表4.13「Fujitsu Siemens Remoteview Service Board (RSB)」 に示します。

表4.13 Fujitsu Siemens Remoteview Service Board (RSB)

luci フィールド	cluster.conf 属性	説明
Name	**name**	フェンスデバイスとして使用する RSB 名
IP Address or Hostname	**ipaddr**	デバイスに割り当てているホスト名
Login	**login**	デバイスへのアクセスに使用するログイン名
Password	**passwd**	デバイスへの接続を認証する際に使用するパスワード
Password Script (optional)	**passwd_script**	フェンスデバイスへのアクセス用パスワードを与えるスクリプト (これを使用するとスクリプトの方が **Password** パラメータより優先される)
TCP Port	**ipport**	telnet サービスがリッスンするポート番号、 デフォルト値は 3172
Force Command Prompt	**cmd_prompt**	使用するコマンドプロンプト、 デフォルト値は '\$'
Power Wait (seconds)	**power_wait**	電源オフまたは電源オンのコマンド発行後の待機秒数
Delay (seconds)	**delay**	フェンシング開始までに待機させる秒数、デフォルト値は 0 です
Power Timeout (seconds)	**power_timeout**	電源オンまたは電源オフのコマンドを発行後、状態変更のテストまで待機させる秒数、デフォルト値は 20 です
Shell Timeout (seconds)	**shell_timeout**	コマンド発行後にコマンドプロンプトを待機させる秒数、デフォルト値は 3 です
Login Timeout (seconds)	**login_timeout**	ログイン後にコマンドプロンプトを待機させる秒数、デフォルト値は 5 です

luci フィールド	cluster.conf 属性	説明
Times to Retry Power On Operation	retry_on	電源オンの動作を再試行する回数、デフォルト値は 1 です

Fujitsu-Siemens RSB のフェンスデバイスを追加する際に使用する設定画面を図4.11「Fujitsu-Siemens RSB」 に示します。

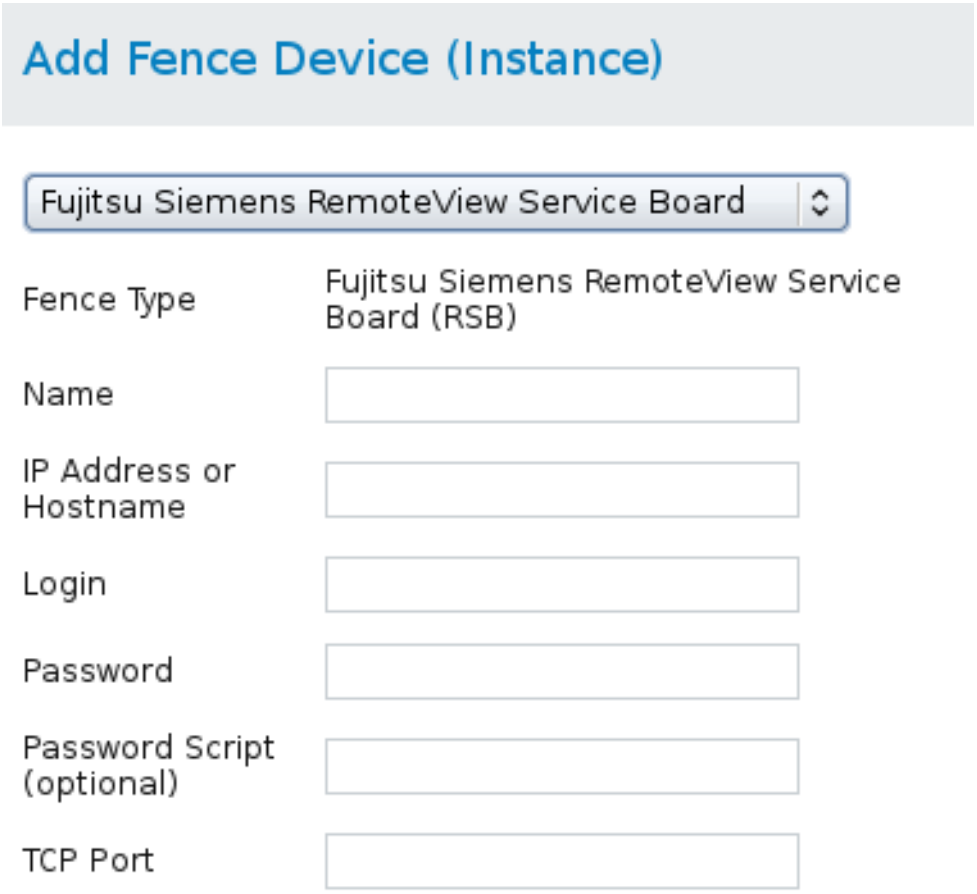


図4.11 Fujitsu-Siemens RSB

Fujitsu-Siemens RSB デバイス用のフェンスデバイスインスタンスを作成するコマンドです。

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

cluster.conf ファイル内の **fence_rsb** デバイス用のエントリーを以下に示します。

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

4.13. HEWLETT-PACKARD BLADESYSTEM

HP Bladesystem のフェンスエージェント **fence_hpblade** で使用するフェンスデバイスのパラメータを 表4.14「HP BladeSystem (Red Hat Enterprise Linux 6.4 以降)」 に示します。

表4.14 HP BladeSystem (Red Hat Enterprise Linux 6.4 以降)

luci フィールド	cluster.conf 属性	詳細
Name	**name**	クラスターに接続している HP Bladesystem デバイスに割り当てる名前
IP Address or Hostname	**ipaddr**	HP BladeSystem デバイスに割り当てている IP アドレスまたはホスト名
IP Port (optional)	**ipport**	デバイスへの接続に使用する TCP ポート
Login	**login**	HP BladeSystem デバイスへのアクセスに使用するログイン名、このパラメータは必須です
Password	**passwd**	フェンスデバイスへの接続を認証する際に使用するパスワード
Password Script (optional)	**passwd_script**	フェンスデバイスへのアクセス用パスワードを与えるスクリプト (これを使用するとスクリプトの方が **Password** パラメータより優先される)
Force Command Prompt	**cmd_prompt**	使用するコマンドプロンプト、 デフォルト値は '\$'
Missing port returns OFF instead of failure	**missing_as_off**	ポートが見つからない場合は障害を発生させずに電源をオフにします
Power Wait (seconds)	**power_wait**	電源オフまたは電源オンのコマンド発行後の待機秒数
Power Timeout (seconds)	**power_timeout**	電源オンまたは電源オフのコマンドを発行後、状態変更のテストまで待機させる秒数、デフォルト値は 20 です
Shell Timeout (seconds)	**shell_timeout**	コマンド発行後にコマンドプロンプトを待機させる秒数、デフォルト値は 3 です
Login Timeout (seconds)	**login_timeout**	ログイン後にコマンドプロンプトを待機させる秒数、デフォルト値は 5 です

luci フィールド	cluster.conf 属性	詳細
Times to Retry Power On Operation	**retry_on**	電源オンの動作を再試行する回数、デフォルト値は 1 です
Use SSH	**secure**	デバイスへのアクセスに SSH を使用するかどうかを指定します、SSH を使用する場合はパスワード、パスワードスクリプト、識別ファイルのいずれかを指定する必要があります
SSH オプション	**ssh_options**	使用する SSH オプション、デフォルト値は **-1 -c blowfish** です
Path to SSH Identity File	**identity_file**	SSH の識別ファイル

HP BladeSystem のフェンスデバイスを追加する際に使用する設定画面を図4.12「HP BladeSystem」に示します。

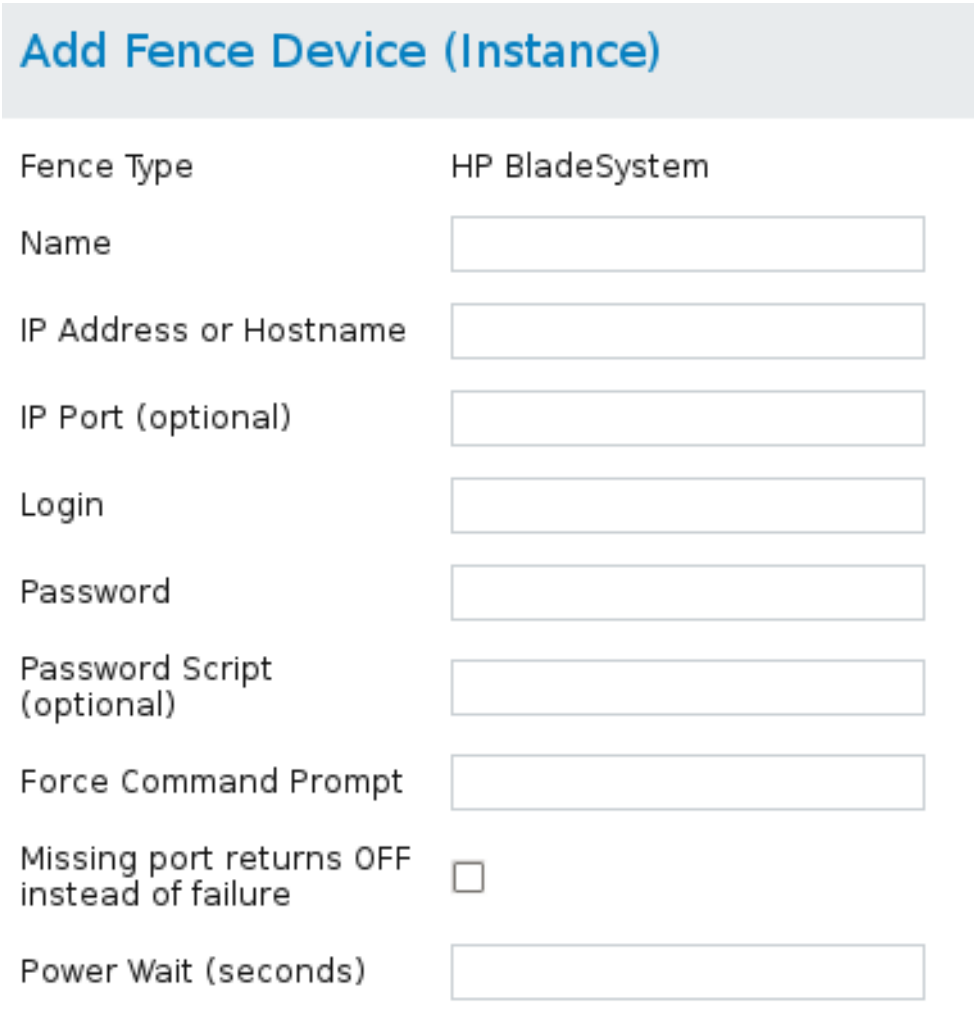


図4.12 HP BladeSystem

HP BladeSystem デバイス用のフェンスデバイスインスタンスを作成するコマンドです。

-

```
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
```

cluster.conf ファイル内の **fence_hpblade** デバイス用のエントリーを以下に示します。

```
 <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>
```

4.14. HEWLETT-PACKARD ILO

HP iLO デバイスのフェンスエージェント **fence_ilo** および HP iLO2 デバイスのフェンスエージェント **fence_ilo2** は同一実装を共有します。これらのエージェントが使用するフェンスデバイスのパラメータを 表4.15「HP iLO (Integrated Lights Out) および HP iLO2」 に示します。

表4.15 HP iLO (Integrated Lights Out) および HP iLO2

luci フィールド	cluster.conf 属性	詳細
Name	**name**	HP iLO 対応のサーバー名です
IP Address or Hostname	**ipaddr**	デバイスに割り当てる IP アドレスまたはホスト名です
IP Port (optional)	**ipport**	デバイスとの接続に使用する TCP ポート、デフォルト値は 443 です
Login	**login**	デバイスへのアクセスに使用するログイン名です
Password	**passwd**	デバイスへの接続を認証する際に使用するパスワードです
Password Script (optional)	**passwd_script**	フェンスデバイスへのアクセス用パスワードを与えるスクリプトです (これを使用するとスクリプトの方が **Password** パラメータより優先されます)
Power Wait (seconds)	**power_wait**	電源オフまたは電源オンのコマンドを発行後に待機させる秒数です
Delay (seconds)	**delay**	フェンシング開始までに待機させる秒数です、デフォルト値は 0 です
Power Timeout (seconds)	**power_timeout**	電源オンまたは電源オフのコマンドを発行後、状態変更のテストまで待機させる秒数です、デフォルト値は 20 です

luci フィールド	cluster.conf 属性	詳細
Shell Timeout (seconds)	**shell_timeout**	コマンド発行後にコマンドプロンプトを待機させる秒数です、デフォルト値は 3 です
Login Timeout (seconds)	**login_timeout**	ログイン後にコマンドプロンプトを待機させる秒数です、デフォルト値は 5 です
Times to Retry Power On Operation	**retry_on**	電源オンの動作を再試行させる回数です、デフォルト値は 1 です

HP iLO のフェンスデバイスを追加する際に使用する設定画面を 図4.13「HP iLO」 に示します。

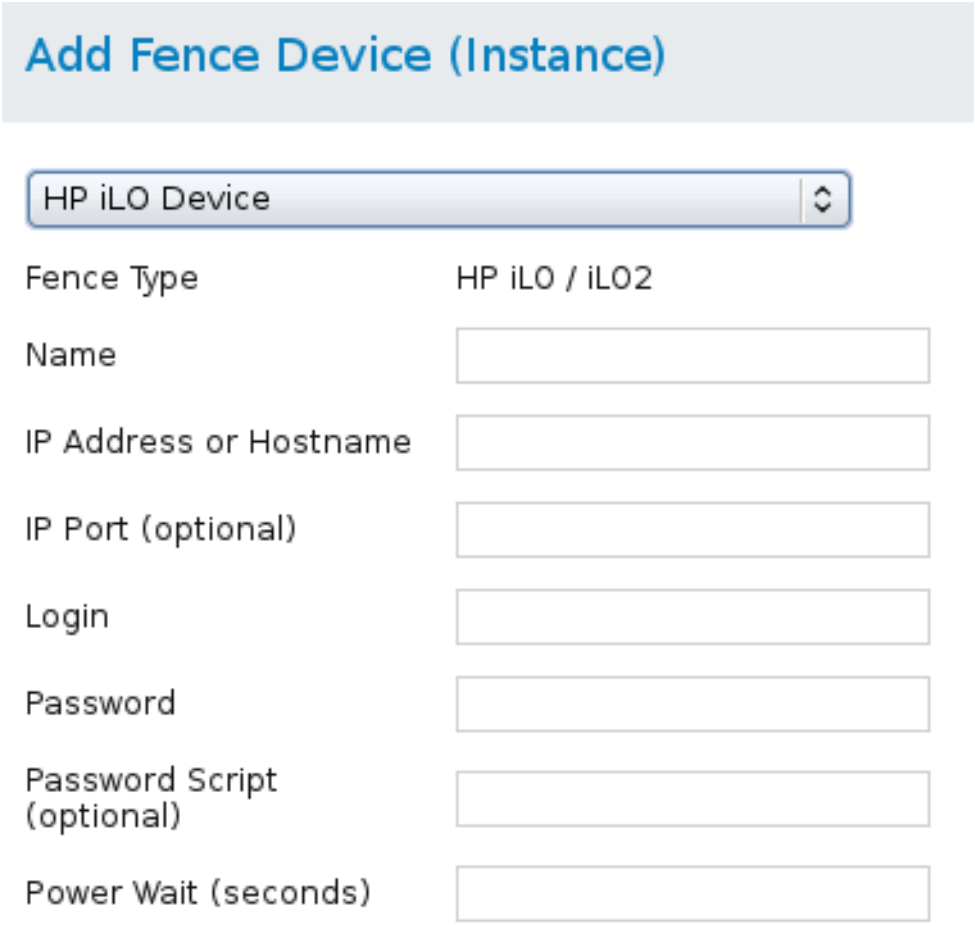


図4.13 HP iLO

HP iLO デバイス用のフェンスデバイスインスタンスを作成するコマンドです。

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

cluster.conf ファイル内の **fence_ilo** デバイス用のエントリーを以下に示します。

-

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

4.15. HEWLETT-PACKARD ILO MP

HP iLO MP デバイスのフェンスエージェント **fence_ilo_mp** で使用するフェンスデバイスのパラメータを 表4.16「HP iLO (Integrated Lights Out) MP」 に示します。

表4.16 HP iLO (Integrated Lights Out) MP

luci フィールド	cluster.conf 属性	詳細
Name	**name**	HP iLO 対応のサーバー名
IP Address or Hostname	**ipaddr**	デバイスに割り当てている IP アドレスまたはホスト名
IP Port (optional)	**ipport**	デバイスへの接続に使用する TCP ポート
Login	**login**	デバイスへのアクセスに使用するログイン名
Password	**passwd**	デバイスへの接続を認証する際に使用するパスワード
Password Script (optional)	**passwd_script**	フェンスデバイスへのアクセス用パスワードを与えるスクリプト (これを使用するとスクリプトの方が **Password** パラメータより優先される)
Use SSH	**secure**	デバイスへのアクセスに SSH を使用するかどうかを指定します、SSH を使用する場合はパスワード、パスワードスクリプト、識別ファイルのいずれかを指定する必要があります
SSH オプション	**ssh_options**	使用する SSH オプション、デフォルト値は **-1 -c blowfish** です
Path to SSH Identity File	**identity_file**	SSH の識別ファイル
Force Command Prompt	**cmd_prompt**	使用するコマンドプロンプト、デフォルト値は 'MP>', 'hpiLO->' です
Power Wait (seconds)	**power_wait**	電源オフまたは電源オンのコマンド発行後の待機秒数

luci フィールド	cluster.conf 属性	詳細
Delay (seconds)	**delay**	フェンシング開始までに待機させる秒数、デフォルト値は 0 です
Power Timeout (seconds)	**power_timeout**	電源オンまたは電源オフのコマンドを発行後、状態変更のテストまで待機させる秒数、デフォルト値は 20 です
Shell Timeout (seconds)	**shell_timeout**	コマンド発行後にコマンドプロンプトを待機させる秒数、デフォルト値は 3 です
Login Timeout (seconds)	**login_timeout**	ログイン後にコマンドプロンプトを待機させる秒数、デフォルト値は 5 です
Times to Retry Power On Operation	**retry_on**	電源オンの動作を再試行する回数、デフォルト値は 1 です

HP iLO MP のフェンスデバイスを追加する際に使用する設定画面を 図4.14「HP iLO MP」 に示します。

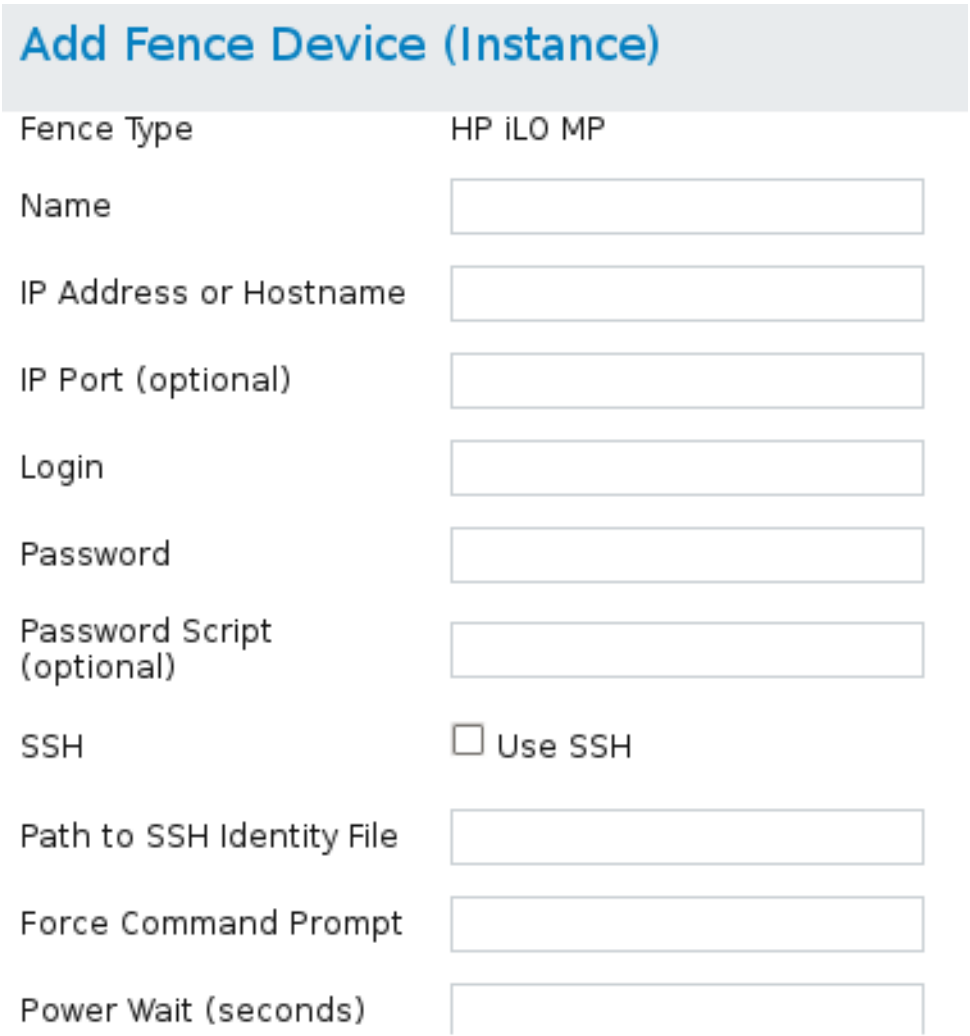


図4.14 HP iLO MP

HP iLO MP デバイス用のフェンスデバイスインスタンスを作成するコマンドです。

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

cluster.conf ファイル内の **fence_hpilo_mp** デバイス用のエントリーを以下に示します。

```
 <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>
```

4.16. IBM BLADECENTER

IBM BladeCenter のフェンスエージェント **fence_bladecenter** で使用するフェンスデバイスのパラメータを 表4.17「IBM BladeCenter」 に示します。

表4.17 IBM BladeCenter

luci フィールド	cluster.conf 属性	詳細
Name	**name**	クラスターに接続している IBM BladeCenter デバイスの名前です
IP Address or Hostname	**ipaddr**	デバイスに割り当てる IP アドレスまたはホスト名です
IP Port (optional)	**ipport**	デバイスで接続に使用する TCP ポートです
Login	**login**	デバイスへのアクセスに使用するログイン名です
Password	**passwd**	デバイスへの接続を認証する際に使用するパスワードです
Password Script (optional)	**passwd_script**	フェンスデバイスへのアクセス用パスワードを与えるスクリプトです (これを使用するとスクリプトの方が **Password** パラメータより優先されます)
Power Wait (seconds)	**power_wait**	電源オフまたは電源オンのコマンドを発行後に待機させる秒数です
Power Timeout (seconds)	**power_timeout**	電源オンまたは電源オフのコマンドを発行後、状態変更のテストまで待機させる秒数、デフォルト値は 20 です
Shell Timeout (seconds)	**shell_timeout**	コマンド発行後にコマンドプロンプトを待機させる秒数、デフォルト値は 3 です

luci フィールド	cluster.conf 属性	詳細
Login Timeout (seconds)	**login_timeout**	ログイン後にコマンドプロンプトを待機させる秒数、デフォルト値は 5 です
Times to Retry Power On Operation	**retry_on**	電源オンの動作を再試行する回数、デフォルト値は 1 です
Use SSH	**secure**	デバイスへのアクセスに SSH を使用するかどうかを指定、SSH を使用する場合はパスワード、パスワードスクリプト、識別ファイルのいずれかを指定する必要があります
SSH オプション	**ssh_options**	使用する SSH オプション、デフォルト値は **-1 -c blowfish** です
Path to SSH Identity File	**identity_file**	SSH の 識別ファイル

IBM BladeCenter のフェンスデバイスを追加する際に使用する設定画面を 図4.15「IBM BladeCenter」に示します。

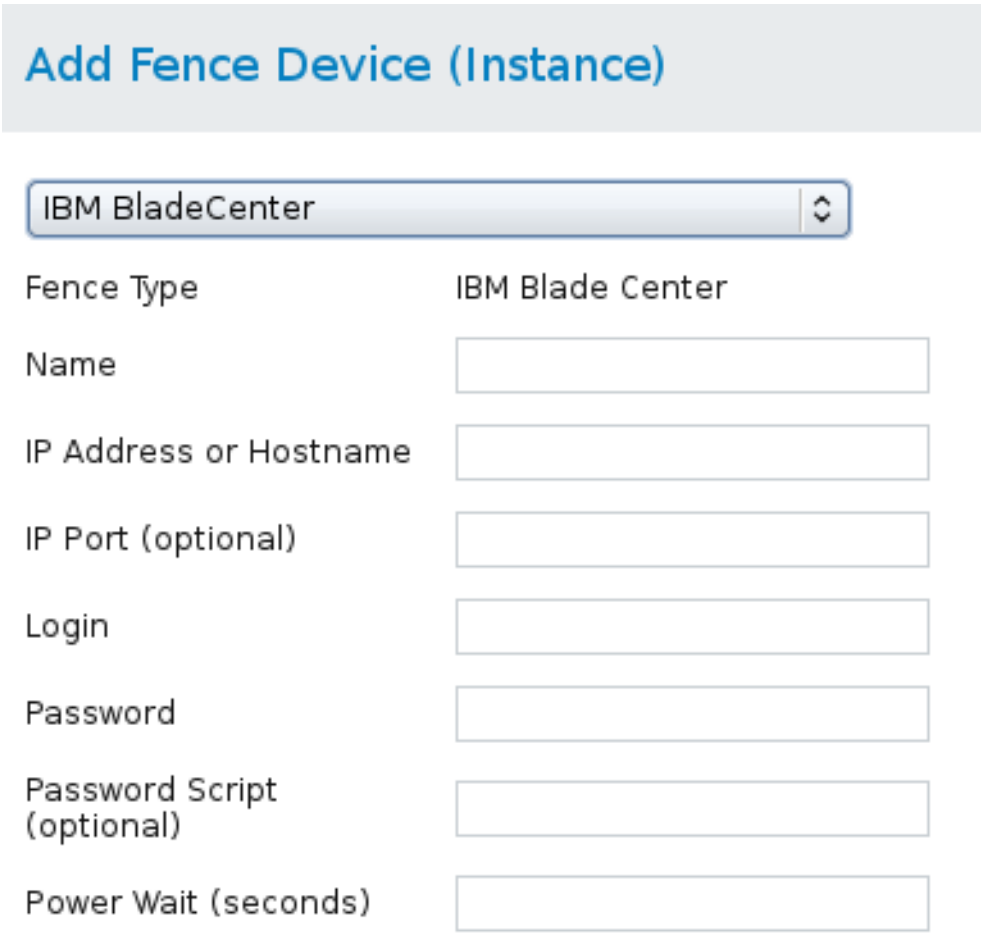


図4.15 IBM BladeCenter

IBM BladeCenter デバイス用のフェンスデバイスインスタンスを作成するコマンドです。

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

cluster.conf ファイルの **fence_bladecenter** デバイス用のエントリーを以下に示します。

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

4.17. SNMP 経由の IBM BLADECENTER

SNMP 経由の IBM BladeCenter のフェンスエージェント **fence_ibmblade** で使用するフェンスデバイスのパラメータを 表4.18「IBM BladeCenter SNMP」 に示します。

表4.18 IBM BladeCenter SNMP

luci フィールド	cluster.conf 属性	詳細
Name	**name**	クラスターに接続している IBM BladeCenter SNMP デバイスの名前
IP Address or Hostname	**ipaddr**	デバイスに割り当てている IP アドレスまたはホスト名
UDP/TCP Port (optional)	**udpport**	デバイスとの接続に使用する UDP/TCP ポート、 デフォルト値は 161
Login	**login**	デバイスへのアクセスに使用するログイン名
Password	**passwd**	デバイスへの接続を認証する際に使用するパスワード
Password Script (optional)	**passwd_script**	フェンスデバイスへのアクセス用パスワードを与えるスクリプト (これを使用するとスクリプトの方が **Password** パラメータより優先される)
SNMP Version	**snmp_version**	使用する SNMP バージョン (1、2c、3)、デフォルト値は 1
SNMP Community	**community**	SNMP コミュニティ文字列
SNMP Security Level	**snmp_sec_level**	SNMP セキュリティレベル (noAuthNoPriv、authNoPriv、authPriv)

luci フィールド	cluster.conf 属性	詳細
SNMP Authentication Protocol	snmp_auth_prot	SNMP 認証プロトコル (MD5、SHA)
SNMP Privacy Protocol	snmp_priv_prot	SNMP プライバシープロトコル (DES、AES)
SNMP Privacy Protocol Password	snmp_priv_passwd	SNMP プライバシープロトコルのパスワード
SNMP Privacy Protocol Script	snmp_priv_passwd_script	SNMP プライバシープロトコル用パスワードを与えるスクリプト (これを使用するとスクリプトの方が **SNMP privacy protocol password** パラメータより優先される)
Power Wait (seconds)	power_wait	電源オフまたは電源オンのコマンド発行後の待機秒数
Power Timeout (seconds)	power_timeout	電源オンまたは電源オフのコマンドを発行後、状態変更のテストまで待機させる秒数、デフォルト値は 20 です
Shell Timeout (seconds)	shell_timeout	コマンド発行後にコマンドプロンプトを待機させる秒数、デフォルト値は 3 です
Login Timeout (seconds)	login_timeout	ログイン後にコマンドプロンプトを待機させる秒数、デフォルト値は 5 です
Times to Retry Power On Operation	retry_on	電源オンの動作を再試行する回数、デフォルト値は 1 です
Port (Outlet) Number	port	物理的なプラグ番号または仮想マシン名
Delay (optional)	delay	フェンシング開始までに待機させる秒数、デフォルト値は 0 です

IBM BladeCenter SNMP のフェンスデバイスを追加する際に使用する設定画面を 図4.16「IBM BladeCenter SNMP」 に示します。

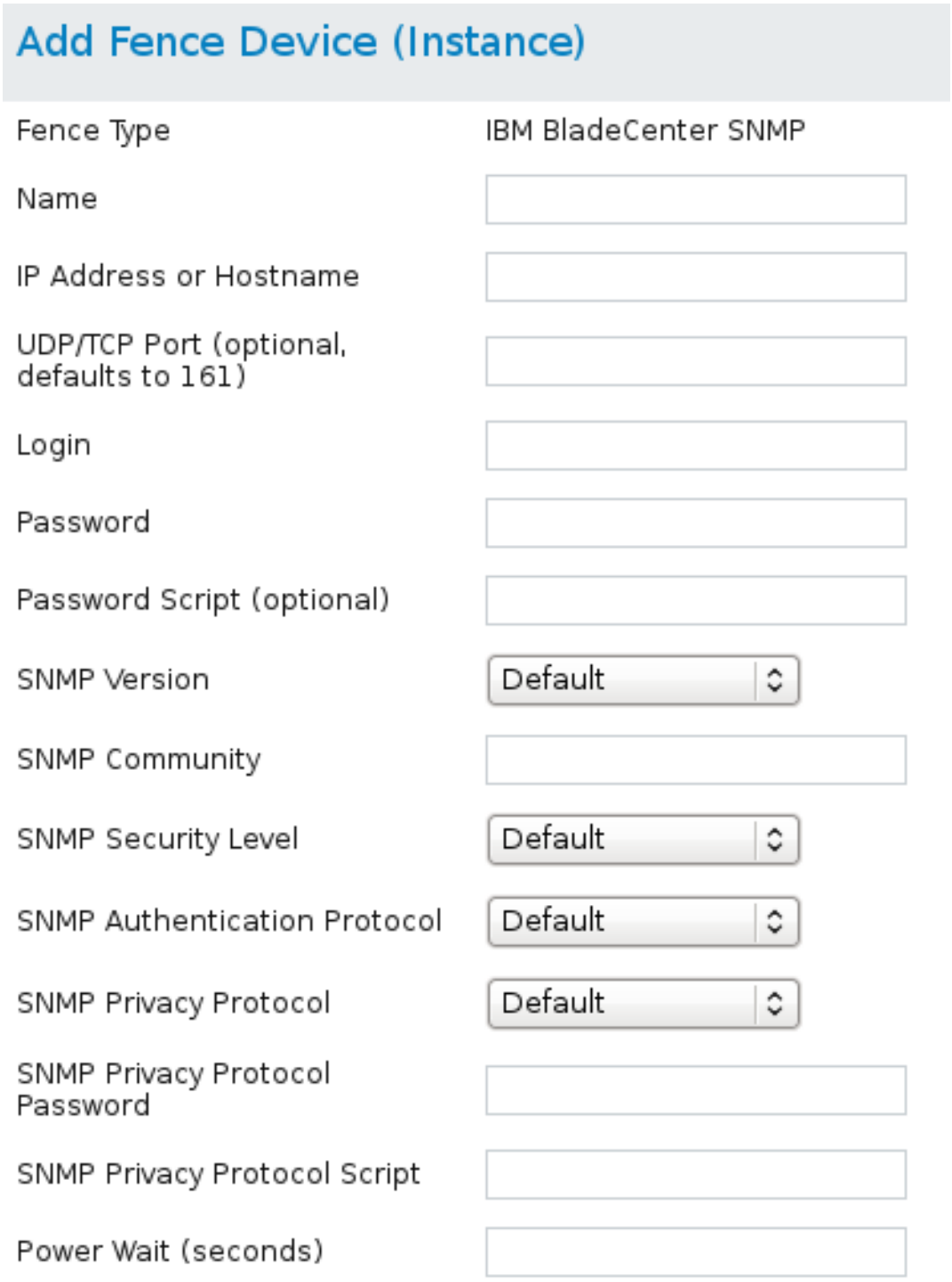


図4.16 IBM BladeCenter SNMP

IBM BladeCenter SNMP デバイス用のフェンスデバイスインスタンスを作成するコマンドです。

```
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
```

cluster.conf ファイル内の **fence_ibmblade** デバイス用のエントリーを以下に示します。

```
 <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>
```

4.18. IBM IPDU

SNMP 経由の iPDU デバイスのフェンスエージェント **fence_ipdu** で使用するフェンスデバイスのパラメータを 表4.19「IBM iPDU (Red Hat Enterprise Linux 6.4 以降)」 に示します。

表4.19 IBM iPDU (Red Hat Enterprise Linux 6.4 以降)

luci フィールド	cluster.conf 属性	詳細
Name	**name**	IBM iPDU デバイスの名前、フェンスデーモンが SNMP プロトコル経由でログインするクラスターに接続する
IP Address or Hostname	**ipaddr**	デバイスに割り当てている IP アドレスまたはホスト名
UDP/TCP Port	**udpport**	デバイスとの接続に使用する UDP/TCP ポート、 デフォルト値は 161
Login	**login**	デバイスへのアクセスに使用するログイン名
Password	**passwd**	デバイスへの接続を認証する際に使用するパスワード
Password Script (optional)	**passwd_script**	フェンスデバイスへのアクセス用パスワードを与えるスクリプト (これを使用するとスクリプトの方が **Password** パラメータより優先される)
SNMP Version	**snmp_version**	使用する SNMP バージョン (1、2c、3)、デフォルト値は 1
SNMP Community	**community**	SNMP コミュニティ文字列、デフォルト値は **private**
SNMP Security Level	**snmp_sec_level**	SNMP セキュリティレベル (noAuthNoPriv、authNoPriv、authPriv)
SNMP Authentication Protocol	**snmp_auth_prot**	SNMP 認証プロトコル (MD5、SHA)
SNMP Privacy Protocol	**snmp_priv_prot**	SNMP プライバシープロトコル (DES、AES)
SNMP Privacy Protocol Password	**snmp_priv_passwd**	SNMP プライバシープロトコルのパスワード

luci フィールド	cluster.conf 属性	詳細
SNMP Privacy Protocol Script	**snmp_priv_passwd_script**	SNMP プライバシープロトコル用パスワードを与えるスクリプト (これを使用するとスクリプトの方が **SNMP privacy protocol password** パラメータより優先される)
Power Wait (seconds)	**power_wait**	電源オフまたは電源オンのコマンド発行後の待機秒数
Power Timeout (seconds)	**power_timeout**	電源オンまたは電源オフのコマンドを発行後、状態変更のテストまで待機させる秒数、デフォルト値は 20 です
Shell Timeout (seconds)	**shell_timeout**	コマンド発行後にコマンドプロンプトを待機させる秒数、デフォルト値は 3 です
Login Timeout (seconds)	**login_timeout**	ログイン後にコマンドプロンプトを待機させる秒数、デフォルト値は 5 です
Times to Retry Power On Operation	**retry_on**	電源オンの動作を再試行する回数、デフォルト値は 1 です
Port (Outlet) Number	**port**	物理的なプラグ番号または仮想マシン名
Delay (optional)	**delay**	フェンシング開始までに待機させる秒数、デフォルト値は 0 です

IBM iPDU のフェンスデバイスを追加する際に使用する設定画面を 図4.17「IBM iPDU」 に示します。

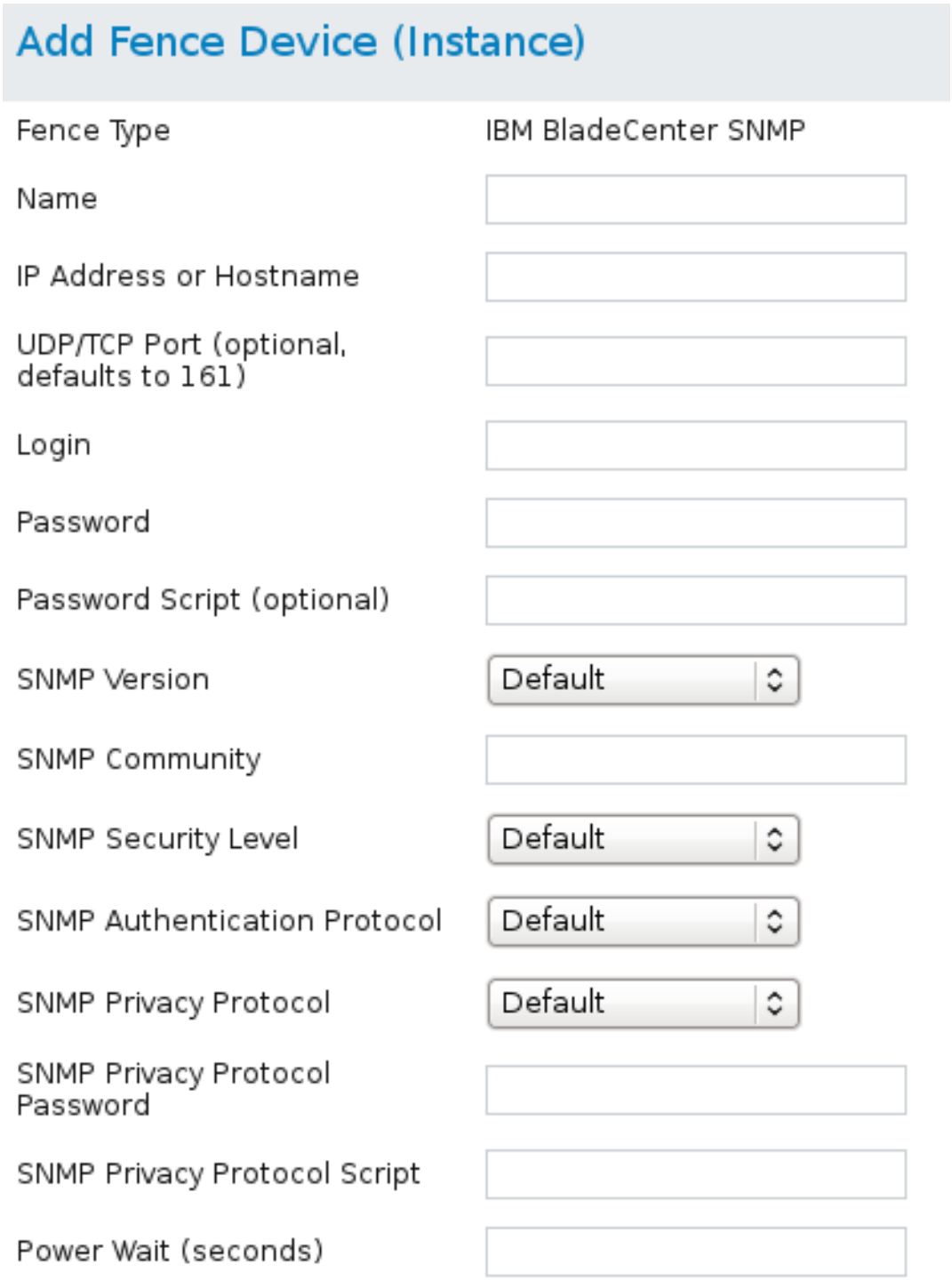


図4.17 IBM iPDU

IBM iPDU デバイス用のフェンスデバイスインスタンスを作成するコマンドです。

```
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
```

cluster.conf ファイル内の **fence_ipdu** デバイス用のエントリーを以下に示します。

```
 <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>
```

4.19. IF-MIB

IF-MIB デバイスのフェンスエージェント **fence_ifmib** で使用するフェンスデバイスのパラメータを 表4.20「IF MIB」 に示します。

表4.20 IF MIB

luci フィールド	cluster.conf 属性	詳細
Name	**name**	クラスターに接続している IF MIB デバイス名
IP Address or Hostname	**ipaddr**	デバイスに割り当てている IP アドレスまたはホスト名
UDP/TCP Port (optional)	**udpport**	デバイスとの接続に使用する UDP/TCP ポート、 デフォルト値は 161
Login	**login**	デバイスへのアクセスに使用するログイン名
Password	**passwd**	デバイスへの接続を認証する際に使用するパスワード
Password Script (optional)	**passwd_script**	フェンスデバイスへのアクセス用パスワードを与えるスクリプト (これを使用するとスクリプトの方が **Password** パラメータより優先される)
SNMP Version	**snmp_version**	使用する SNMP バージョン (1、2c、3)、デフォルト値は 1
SNMP Community	**community**	SNMP コミュニティ文字列
SNMP Security Level	**snmp_sec_level**	SNMP セキュリティレベル (noAuthNoPriv、authNoPriv、authPriv)
SNMP Authentication Protocol	**snmp_auth_prot**	SNMP 認証プロトコル (MD5、SHA)
SNMP Privacy Protocol	**snmp_priv_prot**	SNMP プライバシープロトコル (DES、AES)

luci フィールド	cluster.conf 属性	詳細
SNMP Privacy Protocol Password	**snmp_priv_passwd**	SNMP プライバシープロトコルのパスワード
SNMP Privacy Protocol Script	**snmp_priv_passwd_script**	SNMP プライバシープロトコル用パスワードを与えるスクリプト (これを使用するとスクリプトの方が **SNMP privacy protocol password** パラメータより優先される)
Power Wait (seconds)	**power_wait**	電源オフまたは電源オンのコマンド発行後の待機秒数
Power Timeout (seconds)	**power_timeout**	電源オンまたは電源オフのコマンドを発行後、状態変更のテストまで待機させる秒数、デフォルト値は 20 です
Shell Timeout (seconds)	**shell_timeout**	コマンド発行後にコマンドプロンプトを待機させる秒数、デフォルト値は 3 です
Login Timeout (seconds)	**login_timeout**	ログイン後にコマンドプロンプトを待機させる秒数、デフォルト値は 5 です
Times to Retry Power On Operation	**retry_on**	電源オンの動作を再試行する回数、デフォルト値は 1 です
Port (Outlet) Number	**port**	物理的なプラグ番号または仮想マシン名
Delay (optional)	**delay**	フェンシング開始までに待機させる秒数、デフォルト値は 0 です

IF-MIB のフェンスデバイスを追加する際に使用する設定画面を図4.18「IF-MIB」 に示します。

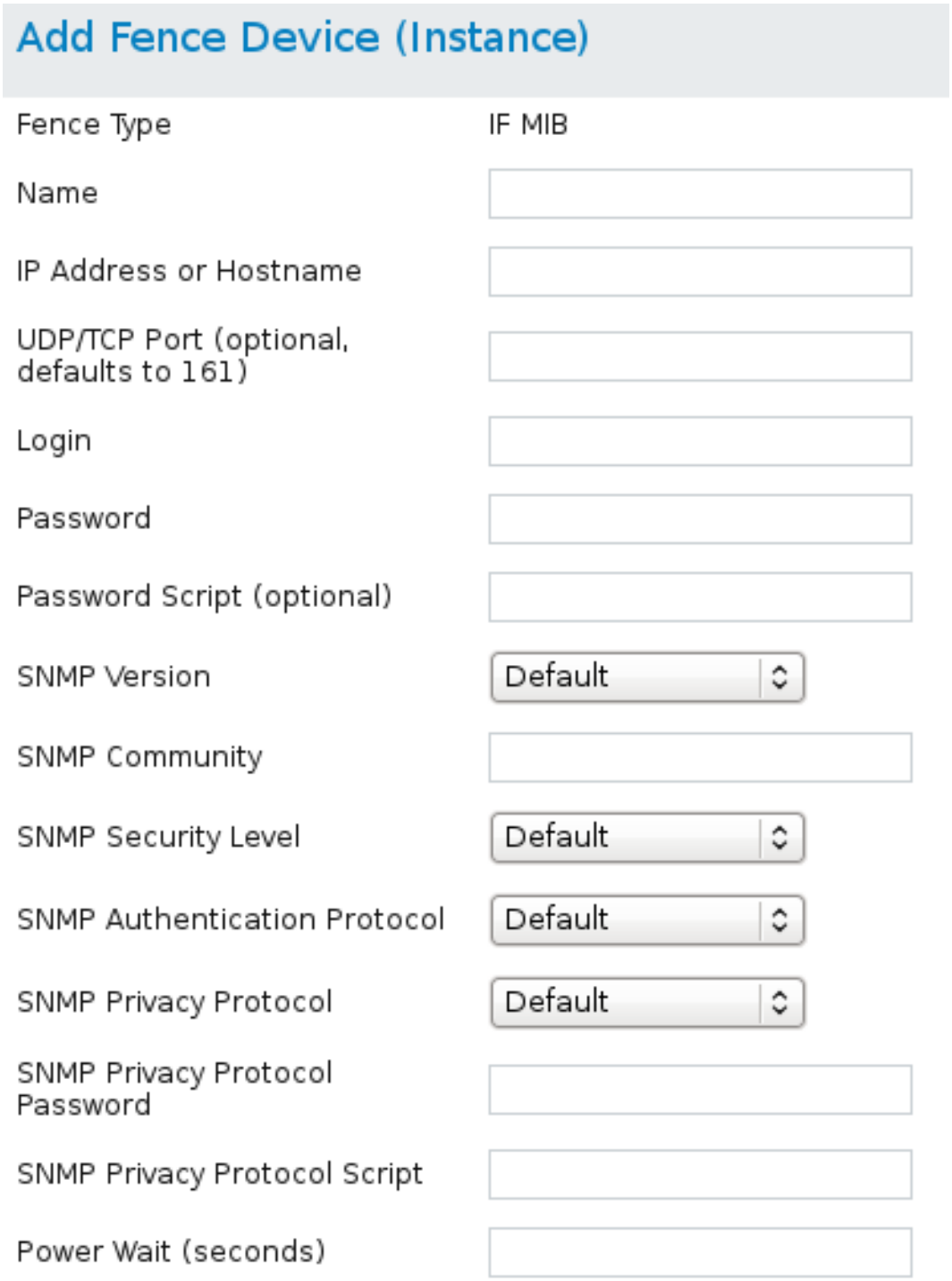


図4.18 IF-MIB

IF-MIB デバイス用のフェンスデバイスインスタンスを作成するコマンドです。

```
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
```

cluster.conf ファイル内の **fence_ifmib** デバイス用のエントリーを以下に示します。

```
 <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>
```

4.20. INTEL MODULAR

Intel Modular のフェンスエージェント **fence_intelmodular** で使用するフェンスデバイスのパラメータを 表4.21「Intel Modular」 に示します。

表4.21 Intel Modular

luci フィールド	cluster.conf 属性	詳細
Name	**name**	クラスターに接続している Intel Modular デバイス名
IP Address or Hostname	**ipaddr**	デバイスに割り当てている IP アドレスまたはホスト名
UDP/TCP Port (optional)	**udpport**	デバイスとの接続に使用する UDP/TCP ポート、 デフォルト値は 161
Login	**login**	デバイスへのアクセスに使用するログイン名
Password	**passwd**	デバイスへの接続を認証する際に使用するパスワード
Password Script (optional)	**passwd_script**	フェンスデバイスへのアクセス用パスワードを与えるスクリプト (これを使用するとスクリプトの方が **Password** パラメータより優先される)
SNMP Version	**snmp_version**	使用する SNMP バージョン (1、 2c、 3)、 デフォルト値は 1
SNMP Community	**community**	SNMP コミュニティ文字列、 デフォルト値は **private**
SNMP Security Level	**snmp_sec_level**	SNMP セキュリティレベル (noAuthNoPriv、 authNoPriv、 authPriv)
SNMP Authentication Protocol	**snmp_auth_prot**	SNMP 認証プロトコル (MD5、 SHA)
SNMP Privacy Protocol	**snmp_priv_prot**	SNMP プライバシープロトコル (DES、 AES)
SNMP Privacy Protocol Password	**snmp_priv_passwd**	SNMP プライバシープロトコルのパスワード

luci フィールド	cluster.conf 属性	詳細
SNMP Privacy Protocol Script	snmp_priv_passwd_script	SNMP プライバシープロトコル用パスワードを与えるスクリプト (これを使用するとスクリプトの方が **SNMP privacy protocol password** パラメータより優先される)
Power Wait (seconds)	power_wait	電源オフまたは電源オンのコマンド発行後の待機秒数
Power Timeout (seconds)	power_timeout	電源オンまたは電源オフのコマンドを発行後、状態変更のテストまで待機させる秒数、デフォルト値は 20 です
Shell Timeout (seconds)	shell_timeout	コマンド発行後にコマンドプロンプトを待機させる秒数、デフォルト値は 3 です
Login Timeout (seconds)	login_timeout	ログイン後にコマンドプロンプトを待機させる秒数、デフォルト値は 5 です
Times to Retry Power On Operation	retry_on	電源オンの動作を再試行する回数、デフォルト値は 1 です
Port (Outlet) Number	port	物理的なプラグ番号または仮想マシン名
Delay (optional)	delay	フェンシング開始までに待機させる秒数、デフォルト値は 0 です

Intel Modular のフェンスデバイスを追加する際に使用する設定画面を 図4.19「Intel Modular」 に示します。

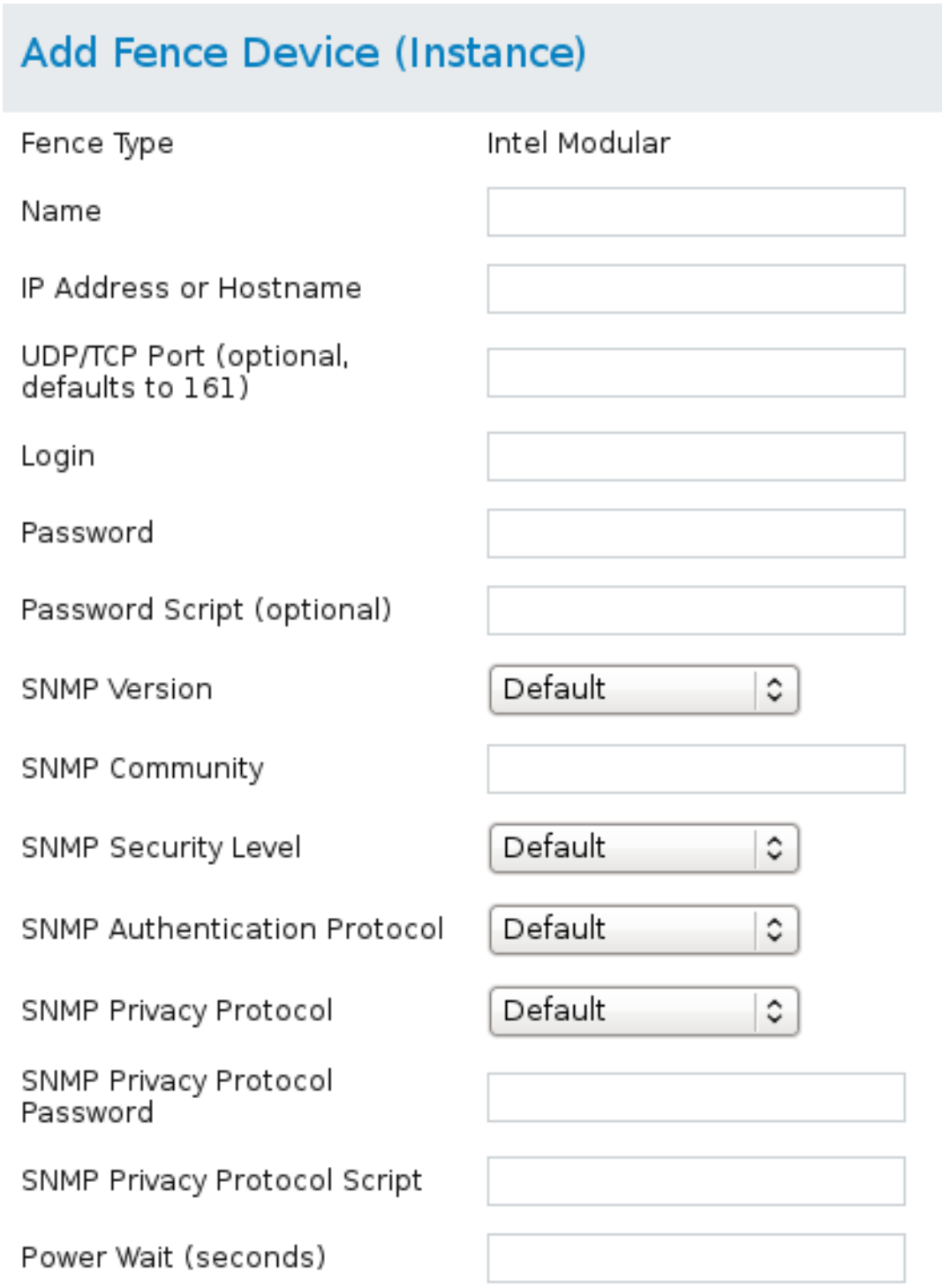


図4.19 Intel Modular

Intel Modular デバイス用のフェンスデバイスインスタンスを作成するコマンドです。

```
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
```

cluster.conf ファイル内の **fence_intelmodular** デバイス用のエントリーを以下に示します。

```
 <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>
```

4.21. IPMI OVER LAN

IPMI over LAN のフェンスエージェント (**fence_ipmilan**)、Dell iDRAC のフェンスエージェント (**fence_idrac**)、IBM Integrated Management Module のフェンスエージェント (**fence_imm**)、HP iLO3 デバイスのフェンスエージェント (**fence_ilo3**)、HP iLO4 デバイスのフェンスエージェント (**fence_ilo4**) は同じ実装を共有します。これらのエージェントで使用されるフェンスデバイスのパラメータを 表4.22「IPMI (Intelligent Platform Management Interface) LAN、Dell iDrac、IBM Integrated Management Module、HPiLO3、HPiLO4」 に示します。

表4.22 IPMI (Intelligent Platform Management Interface) LAN、Dell iDrac、IBM Integrated Management Module、HPiLO3、HPiLO4

luci フィールド	cluster.conf 属性	詳細
Name	**name**	クラスターに接続されるフェンスデバイスの名前
IP Address or Hostname	**ipaddr**	デバイスに割り当てている IP アドレスまたはホスト名
Login	**login**	所定のポートに対し電源オンまたはオフのコマンドを発行できるユーザーのログイン名
Password	**passwd**	ポートへの接続の認証に使用されるパスワード
Password Script (optional)	**passwd_script**	フェンスデバイスへのアクセス用パスワードを与えるスクリプト (これを使用するとスクリプトの方が **Password** パラメータより優先される)
Authentication Type	**auth**	認証タイプ: **none**、**password**、**MD5** のいずれか
Use Lanplus	**lanplus**	**True** または **1** (空白にした場合は **False**)、ハードウェアが対応している場合は Lanplus を有効にして接続の安全性の強化を図ることを推奨しています
Ciphersuite to use	**cipher**	IPMIv2 lanplus の接続に使用するリモートサーバー認証、整合性、及び暗号化のアルゴリズム
Privilege level	**privlvl**	デバイスの権限レベル
IPMI Operation Timeout	**timeout**	IPMI オペレーションのタイムアウト (秒単位)

luci フィールド	cluster.conf 属性	詳細
Power Wait (seconds)	**power_wait**	電源オンまたは電源オフのコマンドを発行後に待機させる秒数です (**fence_ipmilan**、**fence_idrac**、**fence_imm**、**fence_ilo4** のデフォルト値は 2 秒、**fence_ilo3** のデフォルト値は 4 秒です)
Delay (optional)	**delay**	フェンシング開始までに待機させる秒数、デフォルト値は 0 です

LIPMI over LAN デバイスを追加する際に使用する設定画面を 図4.20「IPMI over LAN」 に示します。

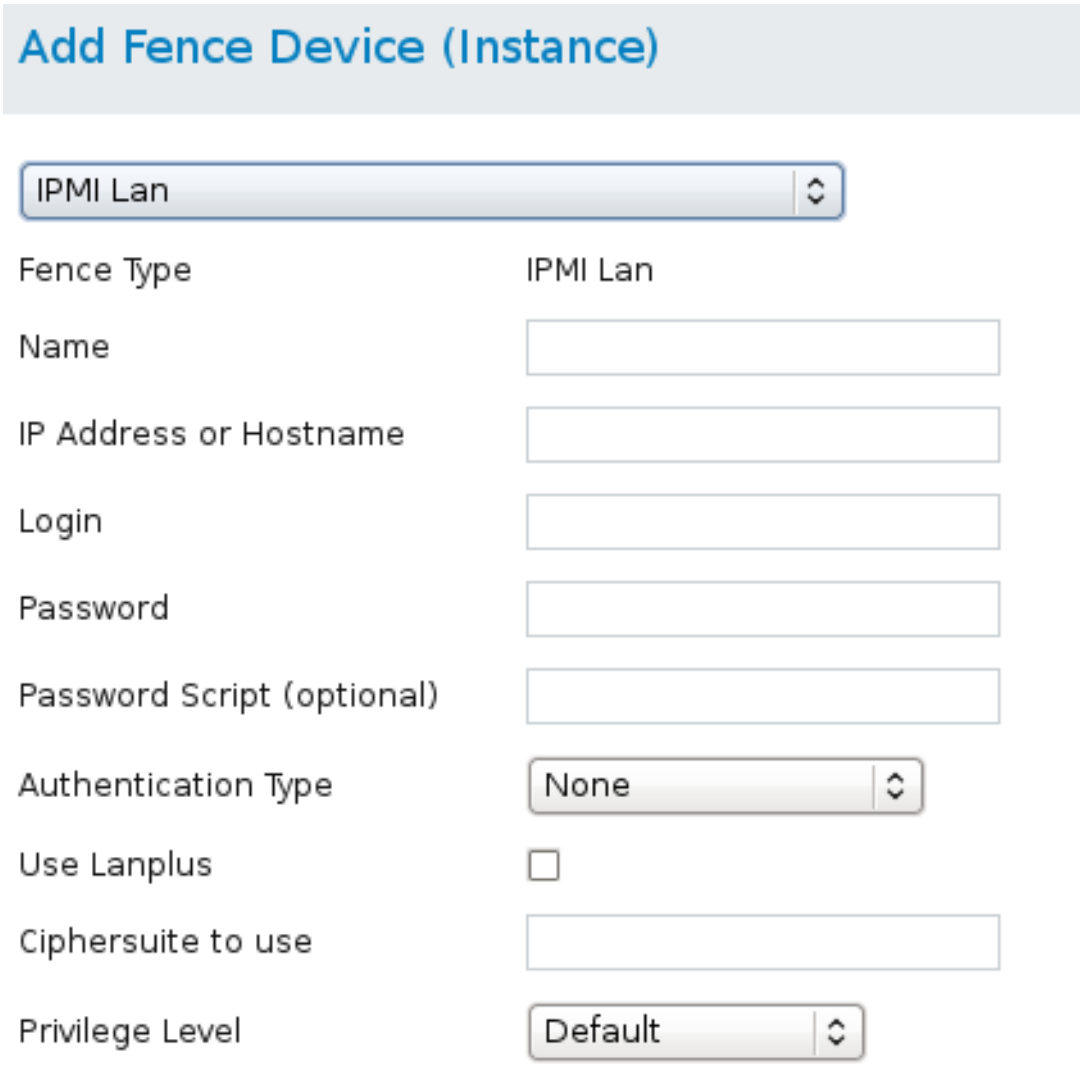


図4.20 IPMI over LAN

IPMI over LAN デバイス用のフェンスデバイスインスタンスを作成するコマンドです。

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

cluster.conf ファイル内の **fence_ipmilan** デバイス用のエントリーを以下に示します。

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

4.22. RHEV-M REST API

RHEV-M REST API のフェンスエージェント **fence_rhevm** で使用するフェンスデバイスのパラメータを 表4.23「RHEV-M REST API (RHEL 6.2 以降及び RHEV 3.0 以降)」 に示します。

表4.23 RHEV-M REST API (RHEL 6.2 以降及び RHEV 3.0 以降)

luci フィールド	cluster.conf 属性	詳細
Name	**name**	RHEV-M REST API フェンスデバイス名
IP Address or Hostname	**ipaddr**	デバイスに割り当てている IP アドレスまたはホスト名
IP Port (optional)	**ipport**	デバイスへの接続に使用する TCP ポート
Login	**login**	デバイスへのアクセスに使用するログイン名
Password	**passwd**	デバイスへの接続を認証する際に使用するパスワード
Password Script (optional)	**passwd_script**	フェンスデバイスへのアクセス用パスワードを与えるスクリプト (これを使用するとスクリプトの方が **Password** パラメータより優先される)
Use SSL	**ssl**	デバイスとの通信に SSL 接続を使用する
Power Wait (seconds)	**power_wait**	電源オフまたは電源オンのコマンド発行後の待機秒数
Power Timeout (seconds)	**power_timeout**	電源オンまたは電源オフのコマンドを発行後、状態変更のテストまで待機させる秒数、デフォルト値は 20 です
Shell Timeout (seconds)	**shell_timeout**	コマンド発行後にコマンドプロンプトを待機させる秒数、デフォルト値は 3 です
Login Timeout (seconds)	**login_timeout**	ログイン後にコマンドプロンプトを待機させる秒数、デフォルト値は 5 です

luci フィールド	cluster.conf 属性	詳細
Times to Retry Power On Operation	retry_on	電源オンの動作を再試行する回数、デフォルト値は 1 です
Port (Outlet) Number	port	物理的なプラグ番号または仮想マシン名
Delay (optional)	delay	フェンシング開始までに待機させる秒数、デフォルト値は 0 です

RHEV-M REST API のデバイスを追加する際に使用する設定画面を図4.21「RHEV-M REST API」 に示します。

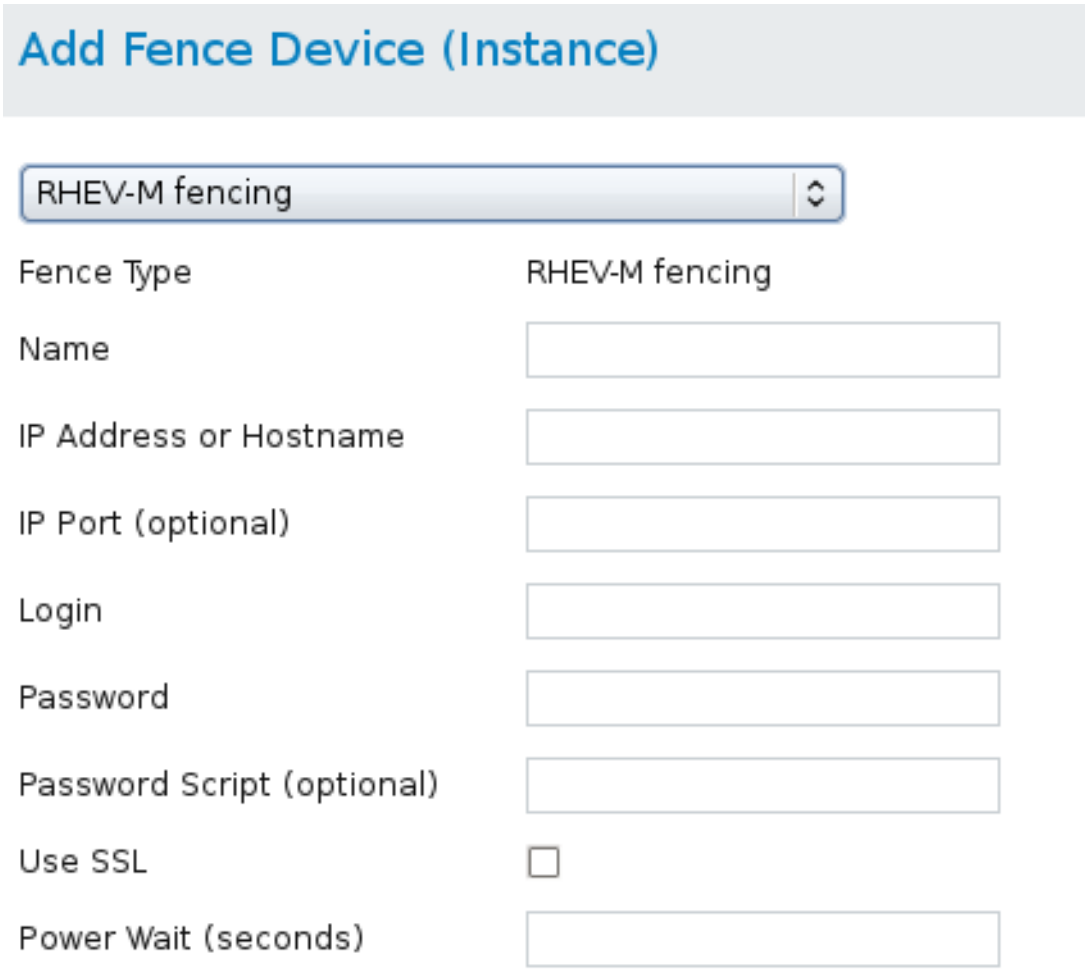


図4.21 RHEV-M REST API

RHEV-M REST API デバイス用のフェンスデバイスインスタンスを作成するコマンドです。

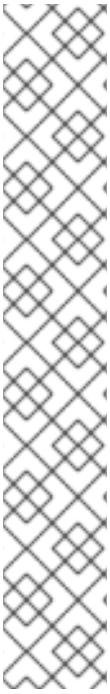
```
ccs -f cluster.conf --addfencedev rhevmtest1 agent=fence_rhevm
ipaddr=192.168.0.1 login=root passwd=password123 \
 power_wait=60 ssl=on
```

cluster.conf ファイル内の **fence_rhevm** デバイス用のエントリーを以下に示します。

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

4.23. SCSI 永続予約

SCSI 永続予約のフェンスエージェント **fence_scsi** で使用するフェンスデバイスのパラメータを 表4.24「SCSI 予約フェンシング」 に示します。



注記

フェンスメソッドとしての SCSI 永続予約を使用する場合、以下のような制限があります。

- SCSI フェンシングを使用する場合は、クラスター内の全ノードを同じデバイスで登録するようにしてください。これにより、各ノードが互いに別のノードの登録キーを登録している全デバイスから削除することができるようになります。
- クラスターボリュームに使用するデバイスは、パーティションではなく 1 つの LUN になるはずです。 SCSI 永続予約は 1 つの LUN 全体で機能します。 つまり、 アクセスは個別のパーティションではなく LUN 単位で制御されることになります。

クラスターのボリュームに使用するデバイスはできる限り **/dev/disk/by-id/***xxx* の形式で指定することを推奨しています。この形式で指定したデバイスは全ノードで整合性を維持するため同じディスクをポイントすることになります。**/dev/sda** などの形式を使用すると、マシンによって異なるディスクを指したり、再起動によって別のディスクをポイントすることになってしまう可能性があります。

表4.24 SCSI 予約フェンシング

luci フィールド	cluster.conf 属性	詳細
Name	**name**	SCSI フェンスデバイス名
Unfencing	**unfence** section of the cluster configuration file	有効にすると、フェンス済みのノードは再起動が完了するまで再度有効にならないようにします。電源フェンス以外の方法を使用する場合 (SAN/ストレージフェンシング) に必要なパラメータです。アンフェンシングを必要とするデバイスを設定する際には、最初にクラスターを停止し、デバイスおよびアンフェンシングを含むすべての設定を追加してから、その後クラスターを開始しなければなりません。ノードにアンフェンシングを設定する方法については **fence_node**(8) の man ページを参照してください。
Node name	**nodename**	このノード名を使って現在の動作に使用するキー値を生成します

luci フィールド	cluster.conf 属性	詳細
Key for current action	key	(ノード名に優先) 現在の動作に使用するキー、ノードに対して固有でなければなりません。"on" 動作の場合はローカルノードの登録に使用するキーを指定し、"off" 動作の場合はデバイスから削除するキーを指定します
Delay (optional)	delay	フェンシング開始までに待機させる秒数、デフォルト値は 0 です

SCSI のフェンスデバイスを追加する際に使用する設定画面を 図4.22「SCSI フェンシング」 に示します。

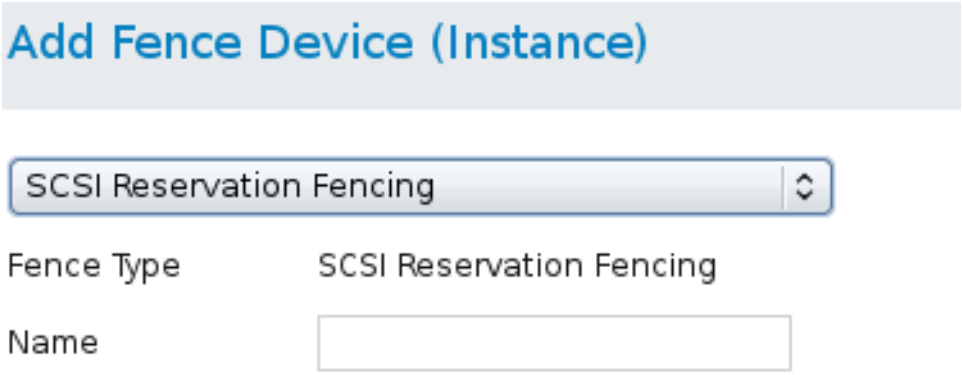


図4.22 SCSI フェンシング

SCSI フェンスデバイス用のフェンスデバイスインスタンスを作成するコマンドです。

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

cluster.conf ファイル内の **fence_scsi** デバイス用のエントリーを以下に示します。

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

4.24. VMWARE OVER SOAP API

VMWare over SOAP API のフェンスエージェント **fence_vmware_soap** で使用するフェンスデバイスのパラメータを表4.25「VMware フェンシング (SOAP インターフェース) (Red Hat Enterprise Linux 6.2 以降)」 に示します。

表4.25 VMware フェンシング (SOAP インターフェース) (Red Hat Enterprise Linux 6.2 以降)

luci フィールド	cluster.conf 属性	詳細
Name	**name**	仮想マシンフェンスデバイスの名前
IP Address or Hostname	**ipaddr**	デバイスに割り当てている IP アドレスまたはホスト名
IP Port (optional)	**ipport**	デバイスとの接続に使用する TCP ポート、デフォルトのポートは 80 です (**Use SSL** を選択すると 443 になります)
Login	**login**	デバイスへのアクセスに使用するログイン名
Password	**passwd**	デバイスへの接続を認証する際に使用するパスワード
Password Script (optional)	**passwd_script**	フェンスデバイスへのアクセス用パスワードを与えるスクリプト (これを使用するとスクリプトの方が **Password** パラメータより優先される)
Power Wait (seconds)	**power_wait**	電源オフまたは電源オンのコマンド発行後の待機秒数
Power Timeout (seconds)	**power_timeout**	電源オンまたは電源オフのコマンドを発行後、状態変更のテストまで待機させる秒数、デフォルト値は 20 です
Shell Timeout (seconds)	**shell_timeout**	コマンド発行後にコマンドプロンプトを待機させる秒数、デフォルト値は 3 です
Login Timeout (seconds)	**login_timeout**	ログイン後にコマンドプロンプトを待機させる秒数、デフォルト値は 5 です
Times to Retry Power On Operation	**retry_on**	電源オンの動作を再試行する回数、デフォルト値は 1 です
VM name	**port**	インベントリのパス形式による仮想マシン名 (/datacenter/vm/Discovered_virtual_machine/myMachine など)
VM UUID	**uuid**	フェンシングする仮想マシンの UUID
Delay (optional)	**delay**	フェンシング開始までに待機させる秒数、デフォルト値は 0 です
Use SSL	**ssl**	デバイスとの通信に SSL 接続を使用する

VMWare over SOAP のフェンスデバイスを追加する際に使用する設定画面を 図4.23「VMWare over SOAP フェンシング」 に示します。

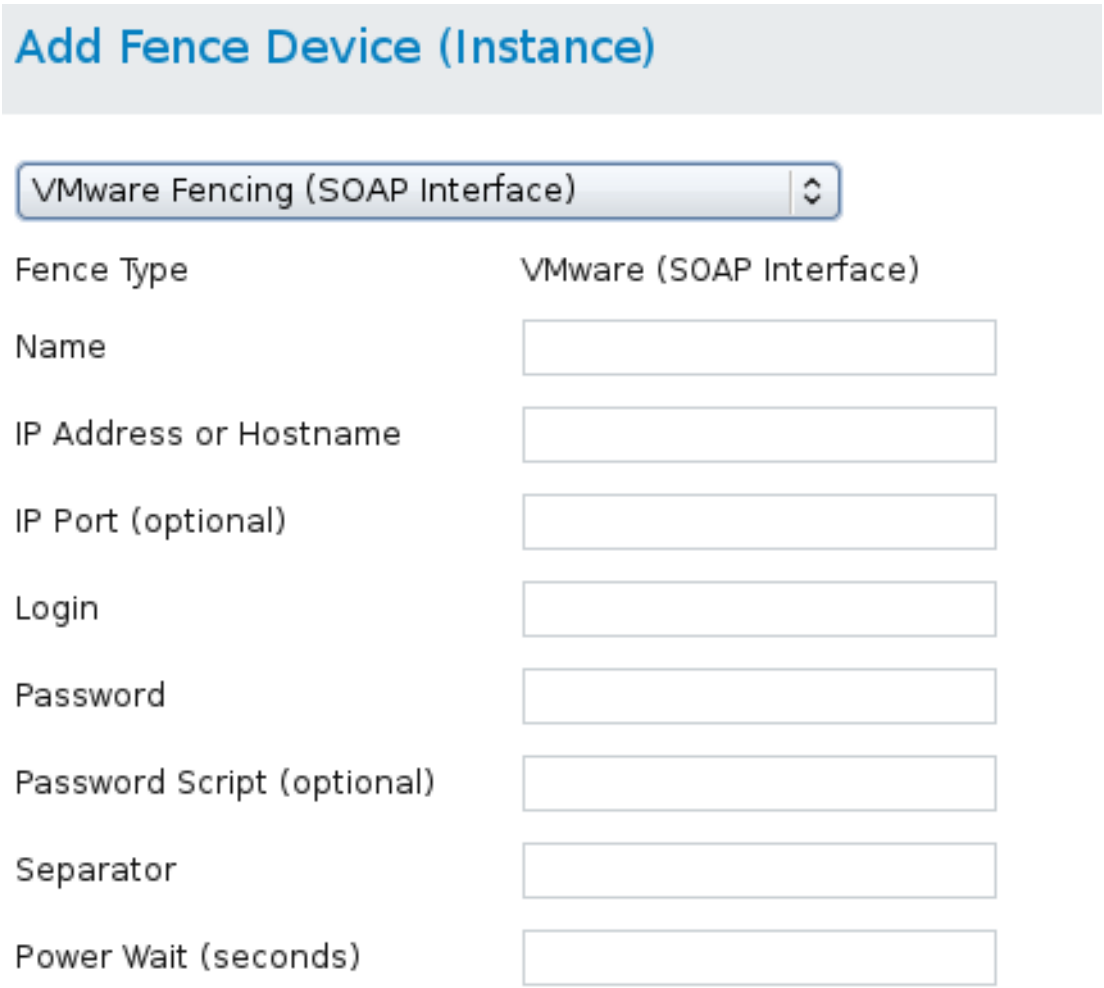


図4.23 VMWare over SOAP フェンシング

VMWare over SOAP フェンスデバイス用のフェンスデバイスインスタンスを作成するコマンドです。

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

cluster.conf ファイル内の **fence_vmware_soap** デバイス用のエントリーを以下に示します。

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

4.25. WTI 電源スイッチ

WTI ネットワーク電源スイッチのフェンスエージェント **fence_wti** で使用するフェンスデバイスのパラメータを 表4.26「WTI 電源スイッチ」 に示します。

表4.26 WTI 電源スイッチ

luci フィールド	cluster.conf 属性	詳細
Name	**name**	クラスターに接続している WTI 電源スイッチの名前です
IP Address or Hostname	**ipaddr**	デバイスに割り当てている IP またはホスト名のアドレスです
IP Port (optional)	**ipport**	デバイスへの接続に使用する TCP ポート
Login	**login**	デバイスへのアクセスに使用するログイン名です
Password	**passwd**	デバイスへの接続を認証する際に使用するパスワードです
Password Script (optional)	**passwd_script**	フェンスデバイスへのアクセス用パスワードを与えるスクリプトです (これを使用するとスクリプトの方が **Password** パラメータより優先される)
Force command prompt	**cmd_prompt**	使用するコマンドプロンプト、 デフォルト値は ['RSM>', '>MPC', 'IPS>', 'TPS>', 'NBB>', 'NPS>', 'VMR>'] です
Power Wait (seconds)	**power_wait**	電源オフまたは電源オンのコマンド発行後に待機させる秒数です
Power Timeout (seconds)	**power_timeout**	電源オンまたは電源オフのコマンドを発行後、状態変更のテストまで待機させる秒数、デフォルト値は 20 です
Shell Timeout (seconds)	**shell_timeout**	コマンド発行後にコマンドプロンプトを待機する秒数、デフォルト値は 3 です
Login Timeout (seconds)	**login_timeout**	ログイン後にコマンドプロンプトを待機する秒数、デフォルト値は 5 です
Times to Retry Power On Operation	**retry_on**	電源オンの動作を再試行する回数、デフォルト値は 1 です
Use SSH	**secure**	デバイスへのアクセスに SSH を使用するかどうかを指定、SSH を使用する場合はパスワード、パスワードスクリプト、識別ファイルのいずれかを指定する必要があります
SSH オプション	**ssh_options**	使用する SSH オプション、デフォルト値は **-1 -c blowfish** です
Path to SSH Identity File	**identity_file**	SSH の識別ファイルです

luci フィールド	cluster.conf 属性	詳細
Port	**port**	物理的なプラグ番号または仮想マシン名です

WTI のフェンスデバイスを追加する際に使用する設定画面を 図4.24「WTI フェンシング」 に示します。

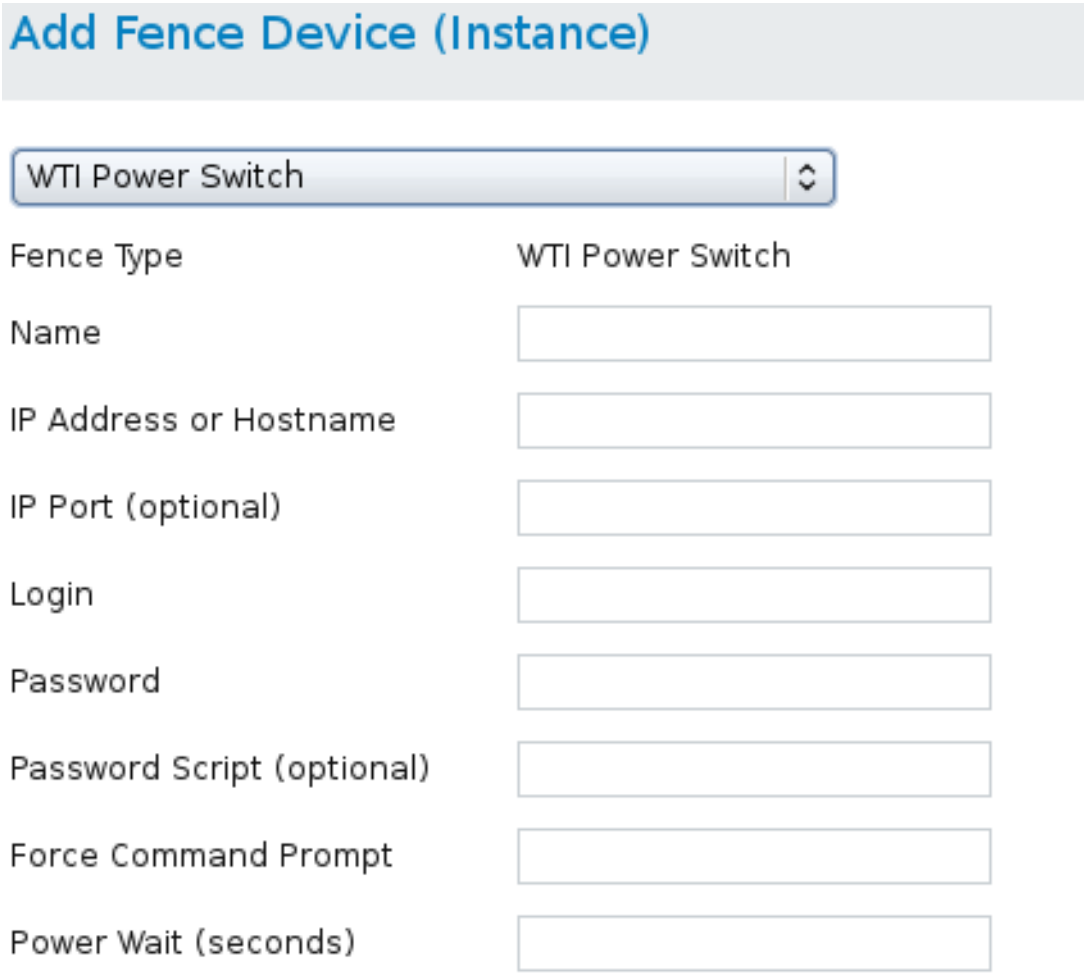


図4.24 WTI フェンシング

WTI フェンスデバイス用のフェンスデバイスインスタンスを作成するコマンドです。

```
ccs -f cluster.conf --addfencedev wtipwrsw1 agent=fence_wti
cmd_prompt=VMR> login=root passwd=password123 \
 power_wait=60
```

cluster.conf ファイル内の **fence_wti** デバイス用のエントリーを以下に示します。

```
 <fencedevices>
  <fencedevice agent="fence_wti" cmd_prompt="VMR&gt;" ipaddr="192.168.0.1"
```

```
login="root" name="wtipwrsw1" \
   passwd="password123" power_wait="60"/>
 </fencedevices>
```

付録A 改訂履歴

改訂 1-15.2 **Wed Feb 18 2015** **Noriko Mizumoto**
翻訳および査読完了

改訂 1-15.1 **Wed Feb 18 2015** **Noriko Mizumoto**
翻訳ファイルを XML ソースバージョン 1-15 と同期

改訂 1-15 **Tue Dec 16 2014** **Steven Levine**
RHEL 6 スプラッシュページに sort_order を実装するため更新

改訂 1-13 **Wed Oct 8 2014** **Steven Levine**
6.6 GA リリースバージョン

改訂 1-11 **Thu Aug 7 2014** **Steven Levine**
Red Hat Enterprise Linux 6.6 の Beta リリース

改訂 1-10 **Thu Jul 31 2014** **Steven Levine**
解決済み: #856311
fence_check の man ページについて記載

解決済み: #1104910
フェンスのパラメータ表を新しいフェンスデバイスのパラメータに更新

改訂 1-9 **Wed Nov 20 2013** **John Ha**
Red Hat Enterprise Linux 6.5 の GA リリース

改訂 1-4 **Mon Nov 28 2012** **John Ha**
Red Hat Enterprise Linux 6.5 の Beta リリース

改訂 1-2 **Mon Nov 28 2012** **John Ha**
Red Hat Enterprise Linux 6.4 Beta 向けにリリース

索引

シンボル

A

H

I

R

S

T

V

W