Oracle Linux 10

Setting Up High Availability Clustering

G14605-02
July 2025

Oracle Linux 10 Setting Up High Availability Clustering,

G14605-02

Contents

5 Configuring Fencing (stonith)

6 Working With Quorum Devices

7 Using the Cockpit HA Cluster Management Web UI

8 More Information

Preface

Oracle Linux 10: Setting Up High Availability Clustering describes how to install and configure high availability clustering in Oracle Linux using Corosync and Pacemaker, which are tools that enable you to achieve high availability for applications and services that are running on Oracle Linux.

Documentation License

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
italic	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
`monospace`	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at https://www.oracle.com/corporate/accessibility/.

Access to Oracle Support for Accessibility

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit https://www.oracle.com/corporate/accessibility/learning-support.html#support-tab.

Diversity and Inclusion

Oracle is fully committed to diversity and inclusion. Oracle respects and values having a diverse workforce that increases thought leadership and innovation. As part of our initiative to build a more inclusive culture that positively impacts our employees, customers, and partners,

we are working to remove insensitive terms from our products and documentation. We are also mindful of the necessity to maintain compatibility with our customers' existing technologies and the need to ensure continuity of service as Oracle's offerings and industry standards evolve. Because of these technical constraints, our effort to remove insensitive terms is ongoing and will take time and external cooperation.

1

About High Availability Clustering

This chapter describes how to set up and configure the Pacemaker and Corosync technologies to create a high availability (HA) cluster that delivers continuous access to services that are running across multiple nodes.

High availability services in Oracle Linux consist of several open source packages and components, including the following:

Pacemaker
Pacemaker is an open source high availability cluster resource manager that's responsible for managing the life cycle of software that's deployed on a cluster. Pacemaker provides high availability services by detecting and recovering from node and resource-level failures by using the API that's provided by the cluster engine Corosync.

Corosync
Corosync, which is included in the Pacemaker package, is an open source cluster engine that includes an API to implement several high availability features. Corosync provides an availability manager that can restart a process when it fails, a configuration and statistics database, and a quorum system that can notify applications when quorum is achieved or lost.

Pacemaker and Corosync Configuration System (pcs)
The Pacemaker and Corosync Configuration System (`pcs`) includes the `pcs` command line interface to manage a cluster and its resources. The `pcsd` daemon runs as a service on each node in the cluster, making it possible to synchronize configuration changes across all the nodes in the cluster.
The `pcs` package also includes a web user interface for managing clusters.

You can download Corosync, Pacemaker, their dependencies, and related packages, from the Unbreakable Linux Network (ULN) at https://linux.oracle.com or the Oracle Linux yum server at https://yum.oracle.com.

Oracle provides support for Corosync and Pacemaker that's used for an active-passive 2-node (1:1) cluster configuration on Oracle Linux 10. Note that support for clustering services doesn't imply support for Oracle products that are clustered by using these services.

Oracle also provides Oracle Clusterware for high availability clustering with Oracle Database. You can find more information at https://www.oracle.com/database/technologies/rac/clusterware.html.

2

Installing and Configuring Pacemaker and Corosync

This chapter describes how to set up and configure the Pacemaker and Corosync features to create a high availability (HA) cluster that delivers continuous access to services running across multiple nodes.

Enabling Access to the Pacemaker and Corosync Packages

The Pacemaker and Corosync packages are available on the Oracle Linux yum server in the `ol10_addons` repository, or on the Unbreakable Linux Network (ULN) in the `ol10_arch_addons` channel.

Some dependency packages might be required from the `ol10_appstream` and `ol10_baseos_latest` yum repositories, or from the `ol10_arch_appstream` and `ol10_arch_baseos_latest` channels on ULN.

Enabling Repositories With ULN

If you're registered to use ULN, use the ULN web interface to subscribe the system to the appropriate channels.

To subscribe a system to the ULN channels:

1. Sign in to https://linux.oracle.com with your ULN username and password.
2. On the **Systems** tab, select the link corresponding to the system whose subscription you're configuring.
3. On the System Details page, select **Manage Subscriptions**.
4. On the System Summary page, select the following channels:
   - `ol10_arch_appstream`
   - `ol10_arch_baseos_latest`
   - `ol10_arch_addons`
5. Select **Save Subscriptions**.

Enabling Repositories With the Oracle Linux Yum Server

If you're using the Oracle Linux yum server for system updates, enable the appropriate Oracle Linux yum repositories.

To enable the yum repositories:

1. Enable the following yum repositories:
   - `ol10_appstream`
   - `ol10_baseos_latest`

- `ol10_addons`

Use the `dnf config-manager` tool to enable the yum repositories:

```
sudo dnf config-manager --enable ol10_appstream ol10_baseos_latest
ol10_addons
```

Installing and Enabling the Pacemaker and Corosync Service

Install and enable Pacemaker and Corosync on each node in the cluster.

To install and configure Pacemaker and Corosync for an HA cluster, complete the following steps on each cluster node:

1. Install the `pcs` and `pacemaker` software packages, and the required resource and fence agents, for example, by running the following command:

   ```
   sudo dnf install pcs pacemaker resource-agents fence-agents-all
   ```

2. Configure the firewall so that the service components can communicate across the network. For example, if you're using `firewalld`, run the following commands:

   ```
   sudo firewall-cmd --permanent --add-service=high-availability
   sudo firewall-cmd --add-service=high-availability
   ```

   The precding commands typically enable the following ports:

   - TCP ports 2224 (used by the `pcs` daemon), 3121 (for Pacemaker Remote nodes), and 21064 (for DLM resources).
   - UDP ports 5405 (for Corosync clustering) and 5404 (for Corosync multicast, if configured).

3. Set a password for the `hacluster` account so you can use the `pcs` command to configure and manage the cluster:

   ```
   sudo passwd hacluster
   ```

   > **Tip:**
   >
   > Set the same same password on each node to avoid authorization issues when running `pcs` commands on different nodes within the same cluster.

4. Set the `pcsd` service to run and to start at boot by running the following command:

   ```
   sudo systemctl enable --now pcsd.service
   ```

> **Note:**
>
> For running High Availability Clustering in the cloud, see the following documents:
>
> - Create a High Availability Cluster on Oracle Cloud Infrastructure (OCI)

3
Configuring a Test Cluster and Service

This chapter provides step-by-step instructions on configuring a test cluster across two nodes that are hosted on systems with the resolvable host names `node1` and `node2`. Each system is installed and configured by using the instructions that are provided in Installing and Configuring Pacemaker and Corosync.

The test cluster is configured to run a service, `Dummy`, that is included in the `resource-agents` package for testing purposes. The Dummy resource agent keeps track of whether the service is or is not running.

To demonstrate a successful failover, the cluster service is stopped on the node the Dummy service is initially running on, and the `pcs status` command is used to confirm the service has successfully started on the other node that is still on line.

Creating the Cluster

To create the cluster:

1. Authenticate the `pcs` cluster configuration tool for the `hacluster` user on each node in your configuration by running the following command on one of the nodes that will form part of the cluster:

   ```
   sudo pcs host auth node1 node2 -u hacluster
   ```

   Replace *node1* and *node2* with the resolvable hostnames of the nodes that will form part of the cluster.

   Alternately, if the node names are not resolvable, specify the IP addresses where the nodes can be accessed, as shown in the following example:

   ```
   sudo pcs host auth node1 addr=192.0.2.1 node2 addr=192.0.2.2 -u hacluster
   ```

   Replace *192.0.2.1* and *192.0.2.2* with the IP addresses of each of the respective hosts in the cluster.

   The tool prompts you to provide a password for the `hacluster` user. Provide the password that you set for this user when you installed and configured the Pacemaker software on each node.

2. Create the cluster by using the `pcs cluster setup` command. You must specify a name for the cluster and the node names and IP addresses for each node in the cluster. For example, run the following command:

   ```
   sudo pcs cluster setup pacemaker1 node1 addr=192.0.2.1 node2 addr=192.0.2.2
   ```

   Replace *pacemaker1* with an appropriate name for the cluster. Replace *node1* and *node2* with the resolvable hostnames of the nodes in the cluster. Replace *192.0.2.1* and *192.0.2.2* with the IP addresses of each of the respective hosts in the cluster.

Note that if you used the `addr` option to specify the IP addresses when authenticated the nodes, you do not need to specify them again when running the `pcs cluster setup` command.

The cluster setup process destroys any existing cluster configuration on the specified nodes and creates a configuration file for the Corosync service that's copied to each of the nodes within the cluster.

You can, optionally, use the `--start` option when running the `pcs cluster setup` command to automatically start the cluster after it has been created.

3. If you have not already started the cluster as part of the cluster setup command, start the cluster on all the nodes. To start the cluster manually, run the following `pcs` command on one of the nodes:

```
sudo pcs cluster start --all
```

Alternatively, you can start `pacemaker` service from `systemd` by running the following command on all the nodes:

```
sudo systemctl start pacemaker.service
```

4. Optionally, you can enable these services to start at boot time so that if a node reboots, it automatically rejoins the cluster. To do this, run the following command on one of the nodes:

```
sudo pcs cluster enable --all
```

Alternatively, you can enable the `pacemaker` service from `systemd`. Run the following command on all the nodes:

```
sudo systemctl enable pacemaker.service
```

> **Note:**
>
> Some users prefer not to enable these services so that a node failure resulting in a full system reboot can be properly debugged before it rejoins the cluster.

Setting Cluster Parameters for Testing Purposes

Fencing is an important part of setting up a production-level HA cluster. For simplicity and testing purposes, it's disabled in this example. Fencing is covered in a later chapter. For more information see About Fencing Configuration (stonith).

> **WARNING:**
>
> **Fencing must not be disabled in a production cluster.**
>
> Fencing is only disabled in test clusters so that resources can be started (for initial testing) without completing setup and configuration of fencing devices.

To set parameters for the test cluster:

1. Disable the fencing feature by running the following command:

   ```
   sudo pcs property set stonith-enabled=false
   ```

2. Optionally, configure the cluster to ignore the quorum state by running the following command:

   ```
   sudo pcs property set no-quorum-policy=ignore
   ```

   Because this example uses a two-node cluster, disabling the no-quorum policy makes the most sense, as quorum technically requires a minimum of three nodes to be a viable configuration. Quorum is only achieved when more than half of the nodes agree on the status of the cluster.

   In the current release of Corosync, this issue is treated specially for two-node clusters, where the quorum value is artificially set to 1 so that the primary node is always considered in quorum. In the case where a network outage results in both nodes going offline for a period, the nodes race to fence each other and the first to succeed wins quorum. The fencing agent can usually be configured to give one node priority so that it is more likely to win quorum if this is preferred.

Creating a Service and Testing Failover

To create a service and test failover:

Services are created and usually configured to run a resource agent that is responsible for starting and stopping processes. Most resource agents are created according to the OCF (Open Cluster Framework) specification, which is defined as an extension for the Linux Standard Base (LSB). Many handy resource agents for commonly used processes are included in the `resource-agents` packages, including various heartbeat agents that track whether commonly used daemons or services are still running.

In the following example, a service is set up that uses a Dummy resource agent created precisely to test Pacemaker. This agent is used because it requires a basic configuration and doesn't make any assumptions about the environment or the types of services that you intend to run with Pacemaker.

1. Add the service as a resource by using the `pcs resource create` command:

   ```
   sudo pcs resource create dummy_service ocf:pacemaker:Dummy
   ```

   In the previous example, *dummy_service* is the name that is provided for the service for this resource:

   To invoke the Dummy resource agent, a notation (`ocf:pacemaker:Dummy`) is used to specify that it conforms to the OCF standard, that it runs in the pacemaker namespace, and that the Dummy script is used. If you were configuring a heartbeat monitor service for a clustered file system, you might use the `ocf:heartbeat:Filesystem` resource agent.

   When you create a service, the cluster starts the resource on a node by using the resource agent's start command.

2. Use the `pcs status` command to view the status of the cluster and its resources:

   ```
   sudo pcs status
   ```

   The following sample extract shows the command output:

   ```
   Cluster name: pacemaker1
   Cluster Summary:
     * Stack: corosync (Pacemaker is running)
     * Current DC: node1 (version version_information) - partition with quorum
     * Last updated: Fri Mar  7 13:31:26 2025 on node1
     * Last change:  Tue Mar  4 14:55:20 2025 by root via root on node1
     * 2 nodes configured
     * 1 resource instance configured

   Node List:
     * Online: [ node2 node1 ]

   Full List of Resources:
     * dummy_service    (ocf:pacemaker:Dummy):     Started node1

   Daemon Status:
     corosync: active/enabled
     pacemaker: active/enabled
     pcsd: active/enabled
   ...
   ```

   The preceding sample output shows both nodes are online, and the service is started on *node1*.

3. Trigger a failover by stopping the cluster on the node where the service is running:

   ```
   sudo pcs cluster stop node1
   ```

   The command reports that Pacemaker and Corosync are being stopped on the node, as shown in the following sample output:

   ```
   node1: Stopping Cluster (pacemaker)...
   node1: Stopping Cluster (corosync)...
   ```

4. Verify the cluster has been stopped, for example, by running `pcs cluster status` command on the node (*node1* in this example):

   ```
   sudo pcs cluster status
   ```

   The command reports the cluster is no longer running, as shown in the following sample output:

   ```
   Error: cluster is not currently running on this node
   ```

5. Sign in to the other node, *node2* in this example, and run the `pcs status` command to confirm that the service has been started on that node:

   ```
   sudo pcs status
   ```

   The following sample extract shows the command output:

   ```
   Cluster name: pacemaker1
   Cluster Summary:
     * Stack: corosync (Pacemaker is running)
     * Current DC: node2 (version version_information) - partition with quorum
     * Last updated: Fri Mar  7 13:35:11 2025 on node2
     * Last change:  Tue Mar  4 14:55:20 2025 by root via root on node1
     * 2 nodes configured
     * 1 resource instance configured

   Node List:
     * Online: [ node2 ]
     * OFFLINE: [ node1 ]

   Full List of Resources:
     * dummy_service    (ocf:pacemaker:Dummy):     Started node2

   Daemon Status:
     corosync: active/enabled
     pacemaker: active/enabled
     pcsd: active/enabled
   ...
   ```

   The preceding sample output shows that *node1* is offline and that the service has successfully failed over and is now started on *node2*.

6. Use the `pcs cluster start` command to start the cluster on the offline node:

   ```
   sudo pcs cluster start node1
   ```

7. Confirm the node is back online:

   ```
   sudo pcs status
   ```

   The following sample extract shows the expected command output:

   ```
   Cluster name: pacemaker1
   Cluster Summary:
     * Stack: corosync (Pacemaker is running)
     * Current DC: node2 (version version_information) - partition with quorum
     * Last updated: Fri Mar  7 13:38:36 2025 on node1
     * Last change:  Tue Mar  4 14:55:20 2025 by root via root on node1
     * 2 nodes configured
     * 1 resource instance configured

   Node List:
     * Online: [ node2 node1]
   ```

```
Full List of Resources:
  * dummy_service    (ocf:pacemaker:Dummy):     Started node2

Daemon Status:
  corosync: active/enabled
  pacemaker: active/enabled
  pcsd: active/enabled
...
```

The preceding sample output shows *node1* is back online.

4

Configuring Pacemaker Resources and Resource Groups

Pacemaker enables you to create groups of resources, such as database and file system services, so that they start in the order you specify and stop in the reverse of that order. Adding resources to the same resource group also ensures that they run on the same node.

About Pacemaker Resources and Resource Groups

A Pacemaker resource, at its most basic, is a service managed by Pacemaker. Example of resources include services for the following:

- IP address
- File system
- Website
- Database

The resources in the preceding list are known as **primitive** resources. Primitive resources can be grouped and cloned into more complex resources called **groups** and **clones**.

Resource Agents

Each primitive resource is managed by a **resource agent** script that provides Pacemaker with access to the resource through a standardized interface. Some of the standards compatible with Pacemaker are shown in the following list:

- Open Cluster Framework (OCF)
- Systemd
- System Services
- STONITH
- Linux Standard Base (LSB)

> **WARNING:**
>
> LSB scripts do not always comply with the standard. See https://clusterlabs.org/ for more information.

To see the resource agents on your system, you can run the `pcs resource agents` command as shown in the following example (lots of lines have been omitted from the sample output for brevity):

```
pcs resource agents


.
.
.
apache
.
.
.
IPaddr
IPaddr2
.
.
.
oracle
oralsnr
pacemaker
pacemaker
.
.
.
pcsd
pcsd
.
.
.
```

To see the interface standards on your system you can run the `pcs resource standards` command as shown in the following example:

```
pcs resource standards


lsb
ocf
service
systemd
```

Resource Properties

Resource properties determine which resource agent manages the resource, where to find that resource agent and which standards it conforms to. The following list describes the properties of a primitive resource:

id
This is the resource name of your choice.

class
The standard the resource agent conforms to.

For example: `ocf`, `stonith`, or `systemd`.

description
Description of the resource agent.

type
The type of the resource agent you need to use.
For example, `IPaddr2`, `Filesystem`, or `Website`.

provider
With OCF agents you can specify the provider to use for a resource (the OCF spec allows multiple vendors to supply the same resource agent).
Example values: `heartbeat`, `pacemaker`.

Creating Resources

You can create a resource by using the `pcs resource create` command as shown in the following example that creates a resource for a virtual IP address:

```
sudo pcs resource create MyVirtualIP ocf:heartbeat:IPaddr2 ip=192.0.2.3 \
cidr_netmask=24 nic=eth1 \
op monitor interval=1s --group myapachegroup
```

In the preceding example:

- *MyVirtualIP* is the **id**, or name of the resource.
- `ocf` is the **class**, or the standard, the resource agent conforms to.
- `heartbeat` is the **provider** of the resource.
- `IPaddr2` is the type of resource that is to be created.

  The following options are specified for the `IPaddr2` resource:

  - `ip=`*192.0.2.3* specifies IP address to use for the Virtual IP Address.
  - `cidr_netmask=`*24* specifies the netmask for the interface in CIDR format.
  - `nic=`*eth1* The base network interface on which the IP address will be brought online.

  The following operation is specified for the `IPaddr2` resource:

  - `op monitor interval=`*1s* specifies that the system checks every second whether the resource is running.
- `--group` *myapachegroup* specifies the resource group to which the resource is to be added. If the group does not exist, it is created.

Following on from the preceding example, you can create and add a second resource, this time of type `apache`, as shown in the following example:

```
sudo pcs resource create MyApacheWebsite ocf:heartbeat:apache \
configfile="/etc/httpd/conf/httpd.conf" \
statusurl="http://192.0.2.3/server-status" --group myapachegroup
```

You can confirm the status of your resources by running the `pcs status` command. Depending upon your configuration, you r command output will be similar to the following sample code block:

```
sudo pcs status
Cluster name: my_cluster
Cluster Summary:
* Stack: corosync (Pacemaker is running)
* Current DC: node1 (version 2.1.6-9.1.0.1.el8_9-
6fdc9deea29) - partition with quorum
* Last updated: Sat Feb 3 22:35:58 2024 on node1
* Last change: Sat Feb 3 22:33:37 2024 by root via
cibadmin on node1
* 2 nodes configured
* 4 resource instances configured
Node List:
* Online: [ node1 ]
* OFFLINE: [ node2 ]
Full List of Resources:
* Resource Group: myapachegroup:
 * MyVirtualIP (ocf::heartbeat:IPaddr2): Started node1
 * MyApacheWebsite (ocf::heartbeat:apache): Started node1
...
...
Daemon Status:
corosync: active/disabled
pacemaker: active/enabled
pcsd: active/enabled
```

Resource Start and Stop Order in a Resource Group

If the resource group in the preceding examples, *myapachegroup*, is started from a stopped status, the resources will be started in the order they were added to the group:

1. *MyVirtualIP* , the first resource to be added to the resource group, will be started first.
2. *MyApacheWebsite*, the second resource to be added to the resource group, will be started second.

Conversely, when the resource group is stopped, the resources will be stopped in the opposite order of that in which they were added to the group:

1. *MyApacheWebsite*, the last resource added to the resource group in the preceding examples, will be stopped first.
2. *MyVirtualIP* , the first resource added to the resource group in the preceding examples, will be stopped last.

You can also explicitly set the start order by using options `--before` or `--after` when creating a resource with the `pcs resource create` command to specify the position of the resource being created relative to a resource that already exists in the group.

The `pcs resource group add` command also enables you to specify resource sequence. See Creating a Resource Group for more information.

Creating a Resource Group

As documented in the preceding sections, using the `pcs resource create` command will create the group specified with the `--group` option if it does not already exist. A second way of creating a resource group is by using the `pcs resource group add` command, as shown in the following example:

```
sudo pcs resource group add mygroup MyVirtualIP MyApacheWebsite
```

The preceding command creates group *mygroup* If it does not already exist, and adds existing resources *MyVirtualIP* and *MyApacheWebsite* to the group. If the resources specified in the command are in another group, they will be moved to the new group *mygroup*.

You can also use options `--before` or `--after` with the `pcs resource group add` command to specify the position of the resource being created relative to a resource that already exists in the group.

See `pcs(8)` and https://clusterlabs.org/ for more information on configuring resources and resource groups.

5

Configuring Fencing (stonith)

This chapter describes how to configure fencing (`stonith`).

About Fencing Configuration (stonith)

Fencing, or `stonith` (shoot the other node in the head), is used to protect data in the event that nodes become unresponsive. If a node fails to respond, it may still be accessing data. To ensure that your data is safe, you can use fencing to prevent a live node from accessing data until the original node is truly offline. To accomplish this task, you must configure a device that can ensure a node is taken offline. There are a number of available fencing agents that can be configured for this purpose. In general, `stonith` relies on particular hardware and service protocols that can force reboot or shutdown nodes physically to protect the cluster.

The following are different configurations that use some available fencing agents. Note that these examples make certain presumptions about hardware and assume that you already know how to set up, configure, and use the affected hardware. The following examples are provided for basic guidance only. It is recommended that you also refer to upstream documentation to familiarize yourself with some of the concepts that are presented in this documentation.

Before proceeding with any of the following configurations, ensure that `stonith` is enabled for your cluster configuration:

```
sudo pcs property set stonith-enabled=true
```

After configuring `stonith`, run the following commands to check your configuration and ensure that it is set up correctly:

```
sudo pcs stonith config
sudo pcs cluster verify --full
```

To check the status of your `stonith` configuration, run the following command:

```
sudo pcs stonith
```

To check the status of your cluster, run the following command:

```
sudo pcs status
```

Fencing Configuration Examples

The following examples describe the various types of fencing configurations that you can implement.

IPMI LAN Fencing

Intelligent Platform Management Interface (IPMI) is an interface to a subsystem that provides management features of the host system's hardware and firmware and includes facilities to power cycle a system over a dedicated network without any requirement to access the system's operating system. You can configure the `fence_ipmilan` fencing agent for the cluster so that `stonith` can be achieved across the IPMI LAN.

If your systems are configured for IPMI, you can run the following commands on one of the nodes in the cluster to enable the `ipmilan` fencing agent and configure `stonith` for both nodes, for example:

```
sudo pcs stonith create ipmilan_n1_fencing fence_ipmilan pcmk_host_list=node1
delay=5 \
ipaddr=203.0.113.1 login=root passwd=password lanplus=1 op monitor
interval=60s
sudo pcs stonith create ipmilan_n2_fencing fence_ipmilan pcmk_host_list=node2
\
ipaddr=203.0.113.2 login=root passwd=password lanplus=1 op monitor
interval=60s
```

In the example, *node1* is a host that has an IPMI LAN interface configured on the IP address *203.0.113.1*. The host named *node2* has an IPMI LAN interface that is configured on the IP *203.0.113.2*. The `root` user password for the IPMI login on both systems is specified in this example as *password*. In each instance. You should replace these configuration variables with the appropriate values for your particular environment.

Note that the delay option should only be set to one node. This setting ensures that in the rare case of a fence race condition that only one node is killed and the other continues to run. Without this option set, it is possible that both nodes make the assumption that they are the only surviving node and then simultaneously reset each other.

> **NOT_SUPPORTED:**
>
> The IPMI LAN agent exposes the login credentials of the IPMI subsystem in plain text. Your security policy should ensure that it is acceptable for users with access to the Pacemaker configuration and tools to also have access to these credentials and the underlying subsystems that are involved.

SCSI Fencing

The SCSI Fencing agent is used to provide storage-level fencing. This configuration protects storage resources from being written to by two nodes simultaneously by using SCSI-3 PR (Persistent Reservation). Used in conjunction with a watchdog service, a node can be reset automatically by using `stonith` when it attempts to access the SCSI resource without a reservation.

To configure an environment in this way:

1. Install the watchdog service on *both* nodes and then copy the provided `fence_scsi_check` script to the watchdog configuration before enabling the service, as shown in the following example:

```
sudo dnf install watchdog
sudo cp /usr/share/cluster/fence_scsi_check /etc/watchdog.d/
sudo systemctl enable --now watchdog
```

2. Enable the `iscsid` service that is provided in the `iscsi-initiator-utils` package on both nodes:

```
sudo dnf install -y iscsi-initiator-utils
sudo systemctl enable --now iscsid
```

3. After both nodes are configured with the watchdog service and the `iscsid` service, you can configure the `fence_scsi` fencing agent on one of the cluster nodes to monitor a shared storage device, such as an iSCSI target, for example:

```
sudo pcs stonith create scsi_fencing fence_scsi pcmk_host_list="node1
node2" \
 devices="/dev/sdb" meta provides="unfencing"
```

In the example, *node1* and *node2* represent the hostnames of the nodes in the cluster and */dev/sdb* is the shared storage device. Replace these variables with the appropriate values for your particular environment.

SBD Fencing

The Storage Based Death (SBD) daemon can run on a system and monitor shared storage. The SBD daemon can use a messaging system to track cluster health. SBD can also trigger a reset if the appropriate fencing agent determines that `stonith` needs to be implemented.

> **Note:**
>
> SBD Fencing is the method used with Oracle Linux HA clusters running on Oracle Cloud Infrastructure, as documented in Create a High Availability Cluster on Oracle Cloud Infrastructure (OCI) .

To set up and configure SBD fencing:

1. Stop the cluster by running the following command on one of the nodes:

```
sudo pcs cluster stop --all
```

2. On each node, install and configure the SBD daemon:

```
sudo dnf install sbd
```

3. Enable the `sbd` systemd service:

```
sudo systemctl enable sbd
```

Note that the `sbd` systemd service is automatically started and stopped as a dependency of the `pacemaker` service, you do not need to run this service independently. Attempting to start or stop the `sbd` systemd service fails and returns an error indicating that it is controlled as a dependency service.

4. Edit the `/etc/sysconfig/sbd` file and set the `SBD_DEVICE` parameter to identify the shared storage device. Use a persistent device path, such as a link within the `/dev/disk/by-id/` directory system, to do this. For example, if the shared storage device is available on */dev/disk/by-id/wwn-0x6d401fd54x2339544b8040d88087Z4*, set the parameter as follows:

   ```
   SBD_DEVICE="/dev/disk/by-id/wwn-0x6d401fd54x2339544b8040d88087Z4"
   ```

5. On one of the nodes, create the SBD messaging layout on the shared storage device and confirm that it is in place. For example, to set up and verify messaging on the shared storage device at */dev/disk/by-id/wwn-0x6d401fd54x2339544b8040d88087Z4*, run the following commands:

   ```
   sudo sbd -d /dev/disk/by-id/wwn-0x6d401fd54x2339544b8040d88087Z4 create
   sudo sbd -d /dev/disk/by-id/wwn-0x6d401fd54x2339544b8040d88087Z4 list
   ```

6. Finally, start the cluster, and configure the `fence_sbd` fencing agent for the shared storage device. For example, to configure the shared storage device, */dev/disk/by-id/wwn-0x6d401fd54x2339544b8040d88087Z4*, run the following commands on one of the nodes:

   ```
   sudo pcs cluster start --all
   sudo pcs stonith create sbd_fencing fence_sbd devices=/dev/disk/by-id/
   wwn-0x6d401fd54x2339544b8040d88087Z4
   ```

IF-MIB Fencing

IF-MIB fencing takes advantage of SNMP to access the IF-MIB on an Ethernet network switch and to also shutdown the port on the switch, which effectively takes a host offline. This configuration leaves the host running, while disconnecting it from the network. Bear in mind that any FibreChannel or InfiniBand connections could remain intact, even after the Ethernet connection has been stopped, which means that any data made available on these connections could still be at risk. Thus, consider configuring this fencing method as a fallback fencing mechanism. See Configuring Fencing Levels for more information about how to use multiple fencing agents in combination to maximize `stonith` success.

To configure IF-MIB fencing:

1. Configure the switch for SNMP v2c, at minimum, and ensure that SNMP SET messages are enabled. For example, on an Oracle Switch, by using the ILOM CLI, you could run the following commands:

   ```
   sudo set /SP/services/snmp/ sets=enabled
   sudo set /SP/services/snmp/ v2c=enabled
   ```

2. On one of the nodes in the cluster, configure the `fence_ifmib` fencing agent for each node in the environment, as shown in the following example:

   ```
   sudo pcs stonith create ifmib_n1_fencing fence_ifmib pcmk_host_list=node1 \
   ipaddr=203.0.113.10 community=private port=1 delay=5 op monitor
   ```

```
interval=60s
sudo pcs stonith create ifmib_n2_fencing fence_ifmib pcmk_host_list=node2 \
ipaddr=203.0.113.10 community=private port=2 op monitor interval=60s
```

In the example, the SNMP IF-MIB switch is accessible at the IP address *203.0.113.10*; the *node1* host is connected to port *1* on the switch, and the *node2* host is connected to port *2* on the switch. Replace these variables with the appropriate values for the particular environment.

Configuring Fencing Levels

If you have configured multiple fencing agents, you may want to set different fencing levels. Fencing levels enable you to prioritize different approaches to fencing and can provide a valuable mechanism for fallback options should your default fencing mechanism fail.

Each fencing level is attempted in ascending order, starting from level 1. If the fencing agent that is configured for a particular level fails, the fencing agent from the next level is then attempted, and so on.

For example, you may wish to configure IPMI-LAN fencing at level 1, but fallback to IF-MIB fencing as a level 2 option. Using the example configurations from Fencing Configuration Examples, you would run the following commands on one of the nodes to set the fencing levels for each configured agent:

```
sudo pcs stonith level add 1 node1 ipmilan_n1_fencing
sudo pcs stonith level add 1 node2ipmilan_n2_fencing
sudo pcs stonith level add 2 node1ifmib_n1_fencing
sudo pcs stonith level add 2 node2ifmib_n2_fencing
```

6
Working With Quorum Devices

A quorum device acts as a third-party arbitrator in the event where standard quorum rules might not adequately cater for node failure. A quorum device is typically used where there may be an even number of nodes in a cluster. For example, in a cluster that contains two nodes failure of the nodes to communicate can result in a split-brain issue where both nodes function as primary at the same time, which results in possible data corruption. By using a quorum device, quorum arbitration can be achieved and a selected node survives.

A quorum device is a service that ideally runs on a separate physical network to the cluster itself. It should run on a system that's not a node in the cluster. Although the quorum device can service multiple clusters at the same time, it should be the only quorum device for each cluster that it serves. Each node in the cluster is configured for the quorum device. The quorum device is installed and run as a network bound service on a system outside of the cluster network.

Installing and Enabling a Quorum Device

Installation of the quorum device requires that you install the `pcs` and `corosync-qnetd` packages on the system where you intend to run the quorum device service and then install the `corosync-qdevice` package on each of the nodes in the existing cluster.

1. On the system assigned to run the quorum device service, run:

   ```
   sudo dnf install -y pcs corosync-qnetd
   ```

2. Enable and start the systemd `pcsd` service by running:

   ```
   sudo systemctl enable --now pcsd
   ```

3. If you're running a firewall on the quorum device service host, you must open the firewall ports to allow the host to communicate with the cluster. For example, run:

   ```
   sudo firewall-cmd --permanent --add-service=high-availability
   sudo systemctl restart firewalld
   ```

4. On the quorum device service host, enable and start the quorum device service by setting the Pacemaker configuration for the node to use the `net` model. Run:

   ```
   sudo pcs qdev setup model net --enable --start
   ```

   This command creates a configuration for the host and names the node `qdev`. It sets the model to `net` and enables and starts the node. The command triggers the `corosync-qnetd` daemon to load and run at boot.

5. On each of the nodes within the existing cluster, install the `corosync-qdevice` package by running:

   ```
   sudo dnf install -y corosync-qdevice
   ```

Configuring the Cluster for a Quorum Device

The node running the quorum device service must be authenticated to the rest of the cluster and must then be added to the cluster. When you add the quorum device service node, you can set configuration options such as which algorithm to use to determine quorum. After the quorum device is added to the cluster you can verify the quorum device status to check that the device is functioning correctly.

1. Authenticate the quorum device service node to the cluster. On a node within the existing cluster, to authenticate the node named *qdev*, run:

   ```
   sudo pcs host auth qdev
   ```

   You're prompted for the cluster username and password.

2. Check that no quorum device is already configured for the cluster. A cluster must never have more than one quorum device configured. On a node within the existing cluster, run:

   ```
   sudo pcs quorum status
   ```

   Note that the output includes membership information:

   ```
   ...
   Membership information
   ----------------------
       Nodeid      Votes    Qdevice Name
            1          1         NR node1 (local)
            2          1         NR node2
   ```

   Under the Qdevice column, the value NR is displayed. The NR value indicates that no quorum devices are registered with any of the nodes within the cluster. If any other value is displayed, don't proceed with adding another quorum device to the cluster without removing the existing device first.

3. Add the quorum device to the cluster. On one of the nodes within the existing cluster, run:

   ```
   sudo pcs quorum device add model net host=qdev algorithm=ffsplit
   ```

   Note that you specify the host to match the host where you're running the quorum device service, in this case named *qdev*; and the algorithm that you want to use to determine quorum, in this case *ffsplit*.
   Algorithm options are:

   - `ffsplit`: is a fifty-fifty split algorithm that favors the partition with the highest number of active nodes in the cluster.
   - `lms`: is a last-man-standing algorithm that returns a vote for the nodes that are still able to connect to the quorum device service node. If a single node is still active and it can connect to the quorum device service, the cluster remains quorate. If none of the nodes can connect to the quorum device service and any one node loses connection with the rest of the cluster, the cluster becomes inquorate.

   See the `corosync-qdevice(8)` manual page for more information.

4. Verify that the quorum device is configured within the cluster. On any node in the existing cluster, run:

```
sudo pcs quorum config
```

The output displays that a quorum device is configured and indicates the algorithm that is in use:

```
Options:
Device:
  Model: net
    algorithm: ffsplit
    host: qdev
```

You can also query the quorum status for the cluster by running:

```
sudo pcs quorum status
```

The output displays the quorum status.

```
Quorum information
------------------
Date:             Fri Jul 15 14:19:07 2022
Quorum provider:  corosync_votequorum
Nodes:            2
Node ID:          1
Ring ID:          1/8272
Quorate:          Yes

Votequorum information
----------------------
Expected votes:   3
Highest expected: 3
Total votes:      3
Quorum:           2
Flags:            Quorate Qdevice

Membership information
----------------------
    Nodeid      Votes    Qdevice Name
         1          1    A,V,NMW node1 (local)
         2          1    A,V,NMW node2
         0          1            Qdevice
```

Note that the membership information now displays values A,V,NMW for the Qdevice field. Values for this field can be equal to any of the following:

- `A/NA`: indicates that the quorum device is alive or not alive to each node in the cluster.
- `V/NV`: indicates whether the quorum device has provided a vote to a node. In the case where the cluster is split, one node would be set to V and the other to NV.
- `MW/NMW`: indicates whether the quorum device `master_wins` flag is set. Any node with an active quorum device that also has the `master_wins` flag set becomes quorate regardless of the node votes of the cluster. By default the option is unset.

Managing Quorum Devices

The quorum device service must be managed from the host system where the quorum device service is running.

Quorum configuration for the cluster and the configuration of the quorum device on the cluster nodes is performed by running operations on any of the nodes within the cluster itself.

Controlling the Quorum Device Service

You can perform various operations to directly control the quorum device service. Commands that control the quorum device service must be run on the host where the quorum device service is running.

- To view the full status for the service, run:

  ```
  sudo pcs qdevice status net --full
  ```

  Output similar to the following is displayed:

  ```
  QNetd address:                  *:5403
  TLS:                            Supported (client certificate required)
  Connected clients:              2
  Connected clusters:             1
  Maximum send/receive size:      32768/32768 bytes
  Cluster "test":
      Algorithm:          ffsplit
      Tie-breaker:        Node with lowest node ID
      Node ID 2:
          Client address:         ::ffff:192.168.2.25:33526
          HB interval:            8000ms
          Configured node list:   1, 2
          Ring ID:                1.16
          Membership node list:   1, 2
          TLS active:             Yes (client certificate verified)
          Vote:                   ACK (ACK)
      Node ID 1:
          Client address:         ::ffff:192.168.2.26:48786
          HB interval:            8000ms
          Configured node list:   1, 2
          Ring ID:                1.16
          Membership node list:   1, 2
          TLS active:             Yes (client certificate verified)
          Vote:                   ACK (ACK)
  ```

- To start the service, run:

  ```
  sudo pcs qdevice start net
  ```

- To stop the service, run:

  ```
  sudo pcs qdevice stop net
  ```

- To enable the service so that it runs at boot time, run:

  ```
  sudo pcs qdevice enable net
  ```

- To disable the service to prevent it from restarting at boot, run:

  ```
  sudo pcs qdevice disable net
  ```

- To force the service to stop if the normal stop process is not working, run:

  ```
  sudo pcs qdevice kill net
  ```

Updating Quorum Device Settings

The quorum device can be updated in the cluster configuration at any time. Modifications to the quorum device configuration must be performed on a node within the cluster. Typically modifications to the quorum device involve changing the algorithm, however you can modify other options that are available for a quorum device in the same way.

To update the algorithm used for the quorum device, run:

```
sudo pcs quorum device update model algorithm=lms
```

The example changes the algorithm to use the `lms` or last-man-standing algorithm.

> **Note:**
>
> You can't update the host for a quorum device. You must remove the device and add it back into the cluster if you need to change the host.

Removing the Quorum Device From the Cluster

To remove the quorum device from the cluster, run the following command on a node within the cluster:

```
sudo pcs quorum device remove
```

Removing the quorum device updates the cluster configuration to remove any configuration entries for the quorum device, reloads the cluster configuration into the cluster and then disables and stops the quorum device on each node.

Because you might use the same quorum device service across multiple clusters, removing the quorum device from the cluster doesn't affect the quorum device service in any way. The service continues to run on the service host, but no longer serves the cluster where it has been removed.

Destroying the Quorum Device Service

You can destroy the quorum device service on the host where the service is running. This action stops the service and removes any configuration for the service from the host.

```
sudo pcs qdevice destroy net
```

> **Note:**
>
> Remove the quorum device from any clusters that it services before destroying the quorum device service.

7

Using the Cockpit HA Cluster Management Web UI

In Oracle Linux 10, the web UI for creating and managing HA clusters is provided by the Cockpit HA Cluster Management add-on. The add-on replaces the `pcsd` web UI used in earlier releases of Oracle Linux.

The examples and procedures in this chapter assume you have completed the following prerequisites for nodes you plan to configure with the HA Cluster Management tool:

- **Pacemaker and Corosync Configuration**

  You need to complete the tasks described in Installing and Enabling the Pacemaker and Corosync Service for each node.

- **`pcs` and `hacluster` Authentication**

  For information about authentication and configuring `hacluster` credentials, see Step 1 of Creating the Cluster.

- This chapter also assumes that you have configured resolvable names for all the nodes.

Installing and Accessing the Cockpit HA Cluster Management Add-on

Install Cockpit, and the Cockpit HA Cluster Management add-on that implements the web UI application for managing clusters.

What do you need?

The prerequisites for installing and using the HA Cluster Management application on a server include the following:

- The Cockpit web console must be installed and enabled.
- You must have administrative access to the Cockpit web console.
- The `pcsd` service must be installed and have access to the nodes you're planning to manage.

Steps

To install and access the Cockpit HA Cluster Management add-on, complete the following steps:

1. Install and enable Cockpit on one of the nodes to be included in a cluster, and verify you have administrative access to the Cockpit web console.

   Upon completion of a standard install and setup, the URL of the Cockpit web console is in the following format:

   ```
   https://clusternode.example.com:9090
   ```

Where *clusternode.example.com* is the FQDN hostname or IP address of the node on which Cockpit has been installed.

For more information on installing Cockpit , and signing in to the Cockpit web console with administrative access, see Oracle Linux: Using the Cockpit Web Console .

2. In the Cockpit web console, select the **Terminal** link in the navigation panel to display the **Terminal shell** page.
3. Run the following command in the **Terminal shell** to install the Cockpit HA Cluster Management add-on application:

```
sudo dnf install -y cockpit-ha-cluster
```

4. Verify the `pcsd.service` is running and enabled by running the following command:

```
sudo systemctl status pcsd.service
```

If it's not running and enabled then run the following command:

```
sudo systemctl enable --now pcsd.service
```

5. In the Cockpit web console, select the **HA Cluster Management** link in the navigation panel to start the application. The **Clusters** page appears.

> **Note:**
>
> You might need to refresh the browser page before you can see the **HA Cluster Management** link.

Managing Clusters With the Web UI

Use the Cockpit HA Cluster Management web UI to create HA clusters and perform cluster management tasks such as stopping and starting clusters, and setting cluster level properties.

Creating a New Cluster

Use the Cockpit HA Cluster Management web UI to create a cluster from nodes on which Pacemaker and Corosync configuration and `hacluster` user authentication tasks have been completed.

Steps

To create cluster in the HA Cluster Management web UI, perform the following steps:

1. In the Cockpit navigation pane, select **HA Cluster Management** .

   The **Clusters** page appears.
2. In the **Clusters** page, select **Setup cluster**.

   The first page of the **Setup cluster** workflow appears.
3. Specify the cluster name, and the nodes to include in the cluster, and then select **Next**.

   A message appears to confirm the nodes are prepared and ready to form a cluster.
4. Select **Next** to acknowledge the message and continue to the next step in the workflow.

The first of the **Advance options** page is displayed.

5. The **Advance options** steps in the workflow include the following pages: **Transport links**, **Transport Options**, **Quorum**, and **Totem**. Optionally set values the on each successive page and select **Next**.

   The **Review settings** page appears.

6. Review the cluster settings you have selected in the preceding steps and select **Setup cluster**.

   A message is displayed to confirm the setup of the cluster is complete.

7. Select **Start cluster and close** to start the cluster and complete the workflow. Alternatively, select **Close** to complete the workflow without starting the cluster.

> **Note:**
>
> If you plan to enable the `cluster` and `pacemaker` services so they start automatically at boot time, you need to do this at the command prompt. See Creating the Cluster for more information on how to do this.

Adding an Existing Cluster

Add an existing cluster to the Cockpit HA Cluster Management web UI application.

Steps

To add an existing cluster to the HA Cluster Management web UI application, perform the following steps:

1. In the Cockpit navigation pane, select **HA Cluster Management** .

   The **Clusters** page appears.

2. In the **Clusters** page, select **Add existing cluster**.

   The first page of the workflow for adding an exising cluster appears.

3. Enter the name of a node from the cluster you intend to add, and then select **Check authentication**.

   The **Prepare node** page appears.

4. In the **Prepare node** page, enter the password for the `hacluster` user, and then select **Authenticate**.

   A message confirms that the node has been authenticated. The option **Add existing cluster** appears.

5. Select **Add existing cluster**.

   The final page of the workflow appears with a message to confirm that the cluster has been successfully added to the web UI.

6. Select **Close** to complete the workflow.

   The **Clusters** page appears and contains the name of the newly added cluster in its list.

Setting Cluster Properties

Use the Cockpit HA Cluster Management web UI application to set cluster proeprties.

The following table gives an overview of some cluster properties you can set in the HA Cluster Management web UI:

Property	Default	Overview
cluster-recheck-interval	`15min`	A value of `0` disables polling. A positive value sets an interval in seconds, unless other units are specified, for example,`5min`.
Enable ACLs	`false`	Boolean value that sets whether Access Control Lists (ACLs) can be used in the cluster to control user and group access.
Stonith Enabled	`true`	Sets whether the cluster can fence failed nodes and nodes with resources that cannot be stopped. To avoid data loss and "split-brain" situation in a cluster. When set to true, at least one fence device must be configured before resources are allowed to run.
No Quorum Policy	`stop`	What to do when the cluster does not have quorum. Possible values include the following: • `ignore`: continue all resource management. • `freeze`: continue resource management, but do not recover resources from nodes that are not in the affected partition. • `stop`: stop all resources in the affected cluster partition. • `demote`: demote promotable resources and stop all other resources in the affected cluster partition. • `fence`: fence all nodes in the affected cluster partition. • `suicide` (deprecated since 2.1.9 and replaced by `fence`): fence all nodes in the affected cluster partition.

Property	Default	Overview
Stonith Action	`reboot`	Action to send to fence device when a node must be fenced. Allowed values are `reboot` and `off`.

For a more complete list of cluster properties, run `pcs property describe`.

Steps

To set a cluster's properties in the HA Cluster Management web UI application, perform the following steps:

1. In the Cockpit navigation pane, select **HA Cluster Management** .

   The **Clusters** page appears.

2. In the **Clusters** page, select the cluster you're configuring.

   A tabbed page displaying the cluster information appears, initially with the **Overview** tab active.

3. Select the **Properties** tab.

   The **Properties** tab becomes active and displays a list of cluster properties.

   > **Tip:**
   >
   > Select the **information** icon that follows each property to display information about role and function of that property.

4. Use the **search box** and the **properties** filter to display only those properties you plan to configure.

5. Select **Edit Attributes**.

   The properties become available for editing.

6. Assign new values to the properties, and select **Save properties**.

   A message to confirm the changes have been successful appears. The properties are displayed with their newly assigned values.

Stopping and Starting Clusters

Stop and start clusters in the Cockpit HA Cluster Management web UI application.

Stopping a cluster in the web UI stops the `pacemaker` and `corosync` services on each node in the cluster. Conversely, starting a cluster in the web UI starts those services on each node.

The example procedure that follows assumes you intend to stop a running cluster and then start it again.

Steps

To stop and start a cluster in the HA Cluster Management web UI application, perform the following steps:

1. In the Cockpit navigation pane, select **HA Cluster Management** .

The **Clusters** page appears.

2. In the **Clusters** page, find the row for the cluster you want to stop and select the **Actions** button.

   A menu of actions appears.

3. Select **Stop** from the list of actions, and in the confirmation message that appears, select **Stop**.

   A message appears to confirm the task has been completed successfully. Upon acknowledging the message, the **Clusters** page appears and the stopped cluster is shown as **offline**.

4. To start the cluster again, select the **Actions** button, and this time select **Start** from the menu that appears.

   A message appears to confirm the task has been completed successfully.

> **Note:**
>
> The cluster status is not shown as **running** until all the nodes are online and some nodes have quorum.

> **Tip:**
>
> Double-click the cluster status to display a summary of the different statuses a cluster can be in.

Removing a Cluster

This procedure shows you how to remove a cluster from the Cockpit HA Cluster Management web UI application.

Removing a cluster from an instance of the Cockpit HA Cluster Management application removes it from the web UI. However, the cluster continues to run and you can contiue administering it through the command line.

Steps

To remove a cluster from an instance of the HA Cluster Management web UI application, perform the following steps:

1. In the Cockpit navigation pane, select **HA Cluster Management** .

   The **Clusters** page appears.

2. In the **Clusters** page, find the row for the cluster you want to stop and select the **Actions** button.

   A menu of actions appears.

3. Select **Remove** from the list of actions, and in the confirmation message that appears, select **Remove**.

   A message confirms the cluster has been removed, and the **Clusters** page appears with the removed cluster no longer showing in the list.

Managing Nodes With the Web UI

Use the Cockpit HA Cluster Management web UI to perform node management tasks, such as adding and removing nodes from a cluster or stopping and starting nodes.

Adding Nodes to a Cluster

Use the Cockpit HA Cluster Management web UI to add nodes to a cluster.

The following procedure assumes that Pacemaker and Corosync have been configured on the node that's being added to the cluster, and that the `hacluster` user credentials have been authenticated for the node.

Steps

To add a node to a cluster in the HA Cluster Management web UI, perform the following steps:

1. In the Cockpit navigation pane, select **HA Cluster Management** .

   The **Clusters** page appears.
2. In the **Clusters** page, select the cluster you're configuring.

   A tabbed page displaying the cluster information appears, initially with the **Overview** tab active.
3. Select the **Nodes** tab.

   The **Nodes** tab becomes active.
4. Select **Add Node**.

   The first page of the **Add Node** workflow appears.
5. Specify the name of the node to add to the cluster, and then select **Next**.

   A message appears to confirm the node is prepared and ready to be added to the cluster.
6. Select **Next** to acknowledge the message and continue to the next step in the workflow.

   The **Specify node addresses** page is displayed.
7. In the **Specify node addresses** page, you can enter multiple addresses for the node if the cluster uses multiple links. Select **Next**.

   The **Configure sbd** page is appears.
8. In the **Configure sbd** page, configure any Storage Based Death devices (also called STONITH block devices) that have been set up, and select **Next**.

   The **Review settings** page is appears.
9. Review the settings you have selected in the preceding steps and select **Add node**.

   The final page of the workflow appears with a message that confirms the node has been added successfully.
10. Select **Start node and close** to start the node and complete the workflow. Alternatively, select **Close** to complete the workflow without starting the node.

> **Note:**
>
> If you plan to enable the `cluster` and `pacemaker` services on the node so they start automatically at boot time, you need to do this at the command prompt. See Creating the Cluster for more information on how to do this.

Removing Nodes from a Cluster

Use the Cockpit HA Cluster Management web UI to remove nodes from a cluster.

Steps

To shut down and remove a node from a cluster in the HA Cluster Management web UI, perform the following steps:

1. In the Cockpit navigation pane, select **HA Cluster Management** .

   The **Clusters** page appears.
2. In the **Clusters** page, select the cluster you're configuring.

   A tabbed page displaying the cluster information appears, initially with the **Overview** tab active.
3. Select the **Nodes** tab.

   The **Nodes** tab becomes active.
4. Select the node you're planning to remove from the cluster.

   A node information panel, with details of resources and services running the node, appears.
5. In the panel, select the **Actions** button.

   A menu of actions appears.
6. Select **Remove** from the list of actions, and in the confirmation message that appears, select **Remove** to confirm the action.

   A message appears and confirms the task has been completed successfully.

Stopping and Starting Nodes

Stop and start cluster nodes in the Cockpit HA Cluster Management web UI application.

Stopping a node in the web UI stops the `pacemaker` and `corosync` services on that node. Conversely, starting a node in the web UI starts those services.

The example procedure that follows assumes you intend to stop a running node and then start it again.

Steps

To stop and start a node in the HA Cluster Management web UI application, perform the following steps:

1. In the Cockpit navigation pane, select **HA Cluster Management** .

   The **Clusters** page appears.
2. In the **Clusters** page, select the cluster whose node you're planning to stop.

A tabbed page displaying the cluster information appears, initially with the **Overview** tab active.

3. Select the **Nodes** tab.

   The **Nodes** tab becomes active.

4. Select the node that's to be stopped.

   A node information panel, with details of resources and services running the node, appears.

5. Select **Stop**, and in the confirmation message that follows, select **Stop** to confirm the action.

   A message appears and confirms the node has been stopped. The node status is shown as **Offline**.

6. To start the node again, select **Start**, and in the confirmation message that follows, select **Start** to confirm the action.

   A message appears and confirms the node has been started. The node status is shown as **Online**.

Putting Nodes into Standby Mode

Use the Cockpit HA Cluster Management web UI application to move nodes into and out of standby mode.

Putting a node in standby moves resources away from that node. The following procedure shows you how to use the web UI put a node into standby mode, and how you might optionally take it out of standby mode when required.

Steps

To put a node into standby mode using the HA Cluster Management web UI, perform the following steps:

1. In the Cockpit navigation pane, select **HA Cluster Management** .

   The **Clusters** page appears.

2. In the **Clusters** page, select the cluster you're configuring.

   A tabbed page displaying the cluster information appears, initially with the **Overview** tab active.

3. Select the **Nodes** tab.

   The **Nodes** tab becomes active.

4. Select the node you're planning to put into standby mode.

   A node information panel, with details of resources and services running the node, appears.

5. In the panel, select the **Actions** button.

   A menu of actions appears.

6. Select **Standby** from the list of actions, and in the confirmation message that appears, select **Standby** to confirm the action.

   A message appears and confirms the task has been completed successfully. The node status is updated to **Standby**.

7. Optionally, you can take the node out of standby mode by selecting **Unstandby** option that now appears in the **Actions** menu.

Putting Nodes into Maintenance Mode

Use the Cockpit HA Cluster Management web UI application to move nodes into and out of standby mode.

Putting a node in maintenance puts the resources hosted on the node into an unmanaged state. Unmanaged resources are not stopped or started by the cluster.

The following procedure shows you how to use the web UI put a node into maintenance mode, and how take it out of maintenance mode when required.

Steps

To put a node into maintenance mode using the HA Cluster Management web UI, perform the following steps:

1. In the Cockpit navigation pane, select **HA Cluster Management** .

   The **Clusters** page appears.
2. In the **Clusters** page, select the cluster you're configuring.

   A tabbed page displaying the cluster information appears, initially with the **Overview** tab active.
3. Select the **Nodes** tab.

   The **Nodes** tab becomes active.
4. Select the node you're planning to put into maintenance mode.

   A node information panel, with details of resources and services running the node, appears.
5. In the panel, select the **Actions** button.

   A menu of actions appears.
6. Select **Maintenance** from the list of actions, and in the confirmation message that appears, select **Maintenance** to confirm the action.
   - A message appears and confirms the task has been completed successfully.
   - The **Pacemaker** section in the node information panel shows the status to be set to **Maintenance**.
   - Any resources running on the node are shown as **Unmanaged**
7. When you have completed maintenance tasks, you can take the node out of maintenance mode and by the selecting **Unmaintenace** option that now appears in the **Actions** menu.

Configuring Fencing in the Web UI

You can use the web UI to configure different kinds of fencing configurations.

The configuration options that become available in the UI depend on the fencing type you create. For example, if you configure SBD fencing for a cluster, the web UI might present different options to those you see when you configure IPMI LAN fencing.

For the purposes of providing an overview of configuring fencing, the following sections show you how to configure SDB fencing in a cluster. The examples assume you have completed the steps discussed in SBD Fencing.

Creating a Fencing Device

Use the Cockpit HA Cluster Management web UI to create a fencing device for a cluster. For the purposes of the example, the procedure steps shown are those for creating device for SDB fencing.

What do you need?

The steps in the following procedure assume you have completed the following:

- Configured a shared storage device to be accessible from all nodes that in the HA cluster. A shared storage device is needed for cluster service and application messaging, and for cluster SBD fencing.
- Completed the steps discussed in SBD Fencing.

Steps

To create a fencing device using the HA Cluster Management web UI application, perform the following steps:

1. In the Cockpit navigation pane, select **HA Cluster Management**.

   The **Clusters** page appears.

2. In the **Clusters** page, select the cluster you're configuring.

   A tabbed page displaying the cluster information appears, initially with the **Overview** tab active.

3. Select the **Fence devices** tab.

   The **Fence devices** tab becomes active.

4. In the **Fence devices** page, select **Create Fence Device**.

   The first page of the workflow for creating fence devices appears.

5. Specify a name for the device you're creating, and from the **Fence device type** list, select `fence_sbd` for the purposes of this example.

   The **Instance attributes** page appears.

6. In the **devices** field, enter the path to the shared storage device that's been set up for the SBD fencing.

   Use a persistent device path, for example a link within the `/dev/disk/by-id/` directory system in the following format:

   `/dev/disk/by-id/`*`wwn-0x6d401fd54x2...44b8040d88087Z4`*
   Optionally, you can use the use the search box and the **More attributes** filter to display other properties you might want to set for the device. You can also do this later, after you have created the device, as discussed in Configuring Fence Device Arguments.

   The **Settings** page appears.

7. On the **Settings** page, select **Start automatically** (the preselected default) or **Disabled**, depending upon whether you want the device to be started after it's created.

   The **Review new fence device configuration** page appears.

8. Select **Create fence device**.

A message confirming the device has been created appears. Upon acknowledging the message, the workflow closes, and the **Fence devices** page appears with the newly created device listed.

Configuring Fence Device Arguments

Use the Cockpit HA Cluster Management web UI to configure parameters for a fencing device.

What do you need?

- The following proedure assumes you have created a fencing device for the cluster.
- For more information on creating fencing devices, see About Fencing Configuration (stonith) and Creating a Fencing Device.

Configuring arguments for a fencing device is a task for advance use cases, and the precise configuration required depends upoon the cluster environment, the device hardware, and so on.

The following table gives an overview of some file device arguments you can set in the HA Cluster Management web UI:

> **Note:**
>
> The properties available in the UI can vary depending upon the type of fencing device you're configuring.

Property	Default	Overview
`pcmk_reboot_timeout`	`60s`	*Advanced use only*. Specifies a device-specific timeout value to use for reboot actions instead of that specified by `stonith-timeout`.
`pcmk_off_timeout`	`60s`	*Advanced use only*. Specifies a device-specific timeout value to use for off actions instead of that specified by `stonith-timeout`.
`pcmk_reboot_retries`	`2`	*Advanced use only*. The maximum number of times to retry the reboot command within the timeout period.
`pcmk_status_retries`	`2`	*Advanced use only*. The maximum number of times to try the `status` command within the timeout period.

Steps

To configure the arguments for a fencing device in the HA Cluster Management web UI application, perform the following steps:

1. In the Cockpit navigation pane, select **HA Cluster Management**.

   The **Clusters** page appears.

2. In the **Clusters** page, select the cluster you're configuring.

   A tabbed page displaying the cluster information appears, initially with the **Overview** tab active.

3. Select the **Fence devices** tab.

   The **Fence devices** tab becomes active.

4. In the **Fence devices** page, select the fence device you're configuring.

   A panel with an overview of the selected fence device appears.

5. Select **Arguments**.

   The panel shows a selection of arguments that can be configured for the device. The panel contains search box and filter to control items the list includes.

   > **Tip:**
   >
   > Select the **information** icon that follows each property to display information about role and function of that property.

6. Select **Edit Arguments**

   The **Edit fence device arguments** page appears.

7. Use the search box and the filter to display the arguments you need to configure, and set the values as required. Select **Save arguments**.

   A message confirming the successful update of the parameters appears. Upon acknowledging the message the workflow closes and the **Fence devices** page appears.

Managing Resources in the web UI

Use the Cockpit HA Cluster Management web UI to create and manage resources and resource groups.

Creating Resources in a Cluster

Use the Cockpit HA Cluster Management web UI to create resources in a cluster.

Steps

To create a cluster resource in the HA Cluster Management web UI application, perform the following steps:

1. In the Cockpit navigation pane, select **HA Cluster Management**.

   The **Clusters** page appears.

2. In the **Clusters** page, select the cluster you're configuring.

   A tabbed page displaying the cluster information appears, initially with the **Overview** tab active.

3. Select the **Resources** tab.

   The **Resources** tab becomes active.

4. In the **Resources** page, select **Create Resource**.

   The first page of the workflow for creating resources appears.

5. Specify a name for the resource you're creating, and from the **Resource type** list, select the type of resource to be created. For example, to create a resource of type `ocf:heartbeat:Filesystem`, select **Filesystem**.

   The **Instance attributes** page appears.

6. In the **Instance attributes** page, complete the mandatory fields and any other attributes you need to specify.

   The attributes available depend upon the type of resource you're creating. For example, if you're creating a resource of type `ocf:heartbeat:Filesystem`, the mandatory attributes include **device**, **directory**, and **fstype**.

   Optionally, you can use the use the search box and the **More attributes** filter to display other properties you might want to set for the resource.

   The **Settings** page appears.

7. On the **Settings** page, configure any settings required for the resource.

   The settings are typically arranged in the following sections:

   - **Multiple instances**:

     This section contains the **Clone** and **Promotable** settings used when creating resources that run concurrently on multiple nodes.

   - **Group**:

     The **Group** section contains options for creating a resource as a member of a resource group.

   - **Start automatically**:

     This section contains the option to specify whether the resource is to be started automatically (the default) or created in a **Disabled** state.

   The **Review new resource configuration** page appears.

8. Select **Create resource**.

   A message confirming the resource has been created appears. Upon acknowledging the message, the workflow closes, and the **Resources** page appears with the newly created resource listed.

Creating Resource Groups

Use the Cockpit HA Cluster Management web UI to create resource groups in a cluster.

Steps

To create a resource group in the HA Cluster Management web UI application, perform the following steps:

1. In the Cockpit navigation pane, select **HA Cluster Management**.

   The **Clusters** page appears.

2. In the **Clusters** page, select the cluster you're configuring.

   A tabbed page displaying the cluster information appears, initially with the **Overview** tab active.

3. Select the **Resources** tab.

   The **Resources** tab becomes active.

4. In the **Resources** page, select **Create group**.

   The **Create group** page appears.

5. Specify a name for the group you're creating, and then move the resources to be added to the group from the **Available resources** list to the **Chosen resources** list.

   > **Note:**
   >
   > - The order in which you add each resource to the **Chosen resources** list sets that resource's start and stop positions in the group that's created. The position of a resource can also be changed after the group has been created as explained in Changing Resource Position in Group.

   The **Create group** option becomes available.

6. Select **Create group**.

   A message confirming the group has been created appears. Upon acknowledging the message, the workflow closes, and the **Resources** page appears with the newly created resource group listed.

Changing Resource Position in Group

Use the Cockpit HA Cluster Management web UI to change a resource's stop and start position within the group to which it belongs.

Steps

To change a resource's position in a group, perform the following steps:

1. In the Cockpit navigation pane, select **HA Cluster Management**.

   The **Clusters** page appears.

2. In the **Clusters** page, select the cluster you're configuring.

   A tabbed page displaying the cluster information appears, initially with the **Overview** tab active.

3. Select the **Resources** tab.

   The **Resources** tab becomes active.

4. In the **Resources** page, select the resource you're configuring.

   A panel containing an overview of the resource's properties appears.

5. In the panel select the **Actions** button.

   A menu of actions appears.

6. Select **Change Group** from the list of actions.

   A page appears with options to change the group-related attributes of the resource.

7. Select the option to change the position of the resource within its current group. Set the resource's position relative to other resources in the group.

   To set the position of the resource, select the **before** or the **after** option, and then select the resource it is to precede or follow.

8. Select **Change group**.

A message confirming the change has been completed appears. Upon acknowledging the message, the workflow closes, and the **Resources** page appears with the group listed with the new resource order.

Changing Resource Group Membership

Use the Cockpit HA Cluster Management web UI to assign a resource to a group. If the resource is already a member of a group, it is taken out of that group and moved to the new one you're assigning the resource to.

Steps

To assign a resource to a resource group, perform the following steps:

1. In the Cockpit navigation pane, select **HA Cluster Management**.

   The **Clusters** page appears.
2. In the **Clusters** page, select the cluster you're configuring.

   A tabbed page displaying the cluster information appears, initially with the **Overview** tab active.
3. Select the **Resources** tab.

   The **Resources** tab becomes active.
4. In the **Resources** page, select the resource you're configuring.

   A panel containing an overview of the resource's properties appears.
5. In the panel select the **Actions** button.

   A menu of actions appears.
6. Select **Change Group** from the list of actions.

   The first page of the workflow to change a resource's group-related attributes appears.
7. Select the option to move the resource to a group.

   The **Select a group** list appears.
8. From the list, select the group to which the resource is to be assigned.

   The current list of resources in the destination group appears, with **before** and **after** options for you to select the position the resource its new group.
9. Select the position of the resource you're moving and select **Change group**.

   A message confirming the change has been completed appears. Upon acknowledging the message, the workflow closes, and the **Resources** page appears with the resource in the group to which it has been moved.

Configuring ACL Permissions in the web UI

Use Pacemaker access control lists (ACLs) to provide local groups and users with role-based access to perform cluster configuration tasks.

To use ACLs in an HA cluster, the following prerequisites are required:

- The **Enable ACLs** cluster property must be set to true for the cluster you're configuring. See Setting Cluster Properties for information on how to do this.
- The local users and groups being configured with ACL permissions must exist on each node in the cluster.

- The local users must be assigned to the `haclient` group on each node in the cluster.

Configuring ACL permissions in the web UI involves working with the following `pcs` entities:

Cluster Information Base
The Cluster Information Base (CIB) is an XML representation of the cluster configuration and the current state of its resources. To view the XML from the CIB run the following command:

```
sudo pcs cluster cib
```

Configuring ACL permissions in HA clusters involves configuring `read`, `write`, and `deny` rules for accessing different parts of the CIB XML. You can use `xpath` values, or the `id` values of XML elements, to specify the CIB XML elements to which permissions are to be applied. For more information on the CIB, and the way permissions are applied to its XML, see manual page for `pcs(8)`.

> **Caution:**
>
> Don't edit the CIB directly. Instead, use the UI, or the `pcs` interface, to configure the cluster.

Roles
You create roles in the UI to define permissions needed by local groups and users. For example, you might create a role named `resource_manager_role` with write access to the `/cib/configuration/resources` XML subtree of the CIB, and assign this role to groups and users that need to manage cluster resources.

Groups
To assign role permissions to a local group on the cluster nodes, you create a corresponding `pcs` group in the web UI with the same name as the local group. You assign the role to the `pcs` group created in the UI, and the users in the corresponding local groups receive the ACL permissions defined in the role.

Users
To assign role permissions to a local user on the cluster nodes, you need to create a corresponding `pcs` user with a matching username in the web UI. You assign the role to the `pcs` user created in the UI, and the corresponding local user on each node is assigned the ACL permissions defined in the role.

Creating Roles

Use the Cockpit HA Cluster Management web UI to create roles with ACL permissions to perform cluster configuration tasks.

Use the web UI to create roles to configure `read`, `write`, and `deny` rules for accessing different parts of the CIB XML. You can use `xpath` values, or the `id` values of XML elements, to specify the CIB XML elements to which permissions are to be applied.

The following example procedure shows how you might create a role with permission to read the CIB and configure the resources.

Steps

To create a role in the HA Cluster Management web UI application, perform the following steps:

1. In the Cockpit navigation pane, select **HA Cluster Management**.

   The **Clusters** page appears.

2. In the **Clusters** page, select the cluster you're configuring.

   A tabbed page displaying the cluster information appears, initially with the **Overview** tab active.

3. Select the **ACL** tab.

   The **ACL** tab becomes active.

4. Select **Create Role**.

   The first page of the **create acl workflow** appears.

5. Specify a name, for example, `resource_manager_role`, and optionally add a description.

   > **Note:**
   >
   > Role names must not conflict with group names. Use a naming convention that pevents such a conflict, for example by using different prefixes for group and role names.

   The **Add permissions** page appears.

6. In the **Add permissions** page, configure **read** permission for the whole CIB xml by selecting **xpath**, and entering a value of `/cib` (the root element), and selecting **read** from the list of permission types. Select **Add permission**.

   A new row of permission fields appears.

7. In the new row, configure **write** permission for resources by selecting **xpath**, and entering a value of `/cib/configuration/resources`, and selecting **write** from the list of permission types.

   Upon completing this step the **Review Settings page** appears.

8. Select **Create role**.

   A message confirming the role has been created appears. Upon acknowledging the message, the workflow closes, and the ACL tab appears with the newly created role listed in the **Roles** section.

Creating Groups

Use the Cockpit HA Cluster Management web UI to create `pcs` groups that correspond to local groups on the cluster nodes.

What do you need?

The steps in the following procedure assume the following prerequisites have been met:

- On each cluster node, a local group has been created with member users that require the same `pcs` ACL permissions. For example, a group named `group_resource_users` whose member users require write access to the resource configuration.

- The user accounts must also be members of the `haclient` group on each node.
- In the UI, you have created a role with the ACL permissions the local users require. See Creating Roles for information on how to do this.

Steps

To create a `pcs` group and assign a role to it, perform the following steps in the HA Cluster Management web UI application:

1. In the Cockpit navigation pane, select **HA Cluster Management**.

   The **Clusters** page appears.
2. In the **Clusters** page, select the cluster you're configuring.

   A tabbed page displaying the cluster information appears, initially with the **Overview** tab active.
3. Select the **ACL** tab.

   The **ACL** tab becomes active.
4. From the **Actions** menu, select **Create Group**.

   The first page of the **Create group** workflow appears.
5. In the first page of the **Create group** workflow, in the **Name** field, enter a value that exactly matches the name of the corresponding local group on each node.

   For example, if the local group on each cluster is named `group_resource_users`, enter this value in the **Name** field.

   The **Assign ACL roles** page appears.
6. In the **Assign ACL roles** page, select the role you have created for this group and move it from the **Available roles** list to the **Chosen roles** list .

   The **Review settings** page appears.
7. Select **Create group**.

   A message confirming the group has been created appears. Upon acknowledging the message, the workflow closes, and the ACL tab appears with the newly created group listed in the **Groups** section.
8. Sign in to the HA Cluster Management web UI application using one of the user accounts belonging to the local group, and verify the role-defined permissions have been assigned to it.

Creating Users

Use the Cockpit HA Cluster Management web UI to create `pcs` users that correspond to local users on the cluster nodes.

What do you need?

The steps in the following procedure assume the following prerequisites have been met:

- On each cluster node, there is a user account that requires `pcs` ACL permissions. The user accounts must be a member of the `haclient` group on each node.
- In the UI, you have created a role with the ACL permissions the local user requires. See Creating Roles for information on how to do this.

Steps

To create a `pcs` user and assign a role to it, perform the following steps in the HA Cluster Management web UI application:

1. In the Cockpit navigation pane, select **HA Cluster Management**.

   The **Clusters** page appears.

2. In the **Clusters** page, select the cluster you're configuring.

   A tabbed page displaying the cluster information appears, initially with the **Overview** tab active.

3. Select the **ACL** tab.

   The **ACL** tab becomes active.

4. Select **Create User**.

   The first page of the **Create user** workflow appears.

5. In the first page of the **Create user** workflow, in the **Name** field, enter a value that exactly matches the username of the corresponding local user on each node.

   For example, if the local user is named `user1`, enter this value in the **Name** field.

   The **Assign ACL roles** page appears.

6. In the **Assign ACL roles** page, select the role you have created for this group and move it from the **Available roles** list to the **Chosen roles** list .

   The **Review settings** page appears.

7. Select **Create user**.

   A message confirming the user has been created appears. Upon acknowledging the message, the workflow closes, and the ACL tab appears with the newly created user listed in the **Users** section.

8. Sign in to the HA Cluster Management web UI application as the local user and verify the role-defined permissions have been assigned to it.

8
More Information

For more information and documentation on Pacemaker and Corosync, see https://clusterlabs.org/pacemaker/doc/.