



How to test fence devices and fencing configuration in a Red Hat High Availability cluster?

🔒 **SOLUTION VERIFIED** - Updated February 7 2019 at 4:55 AM - English ▼

Environment

- Red Hat Enterprise Linux (RHEL) 5, 6, 7, or 8 with the High Availability Add On

Issue

- We had a failed fence event in our cluster and need to test whether the fence devices and configuration are working.
- For compliance with support policies for Red Hat Enterprise Linux High Availability cluster software, we need to ensure fencing is working in our cluster.
- How can I test my fence device to ensure it works properly?

Resolution

When testing whether a fence or `stonith` device is working properly, it is important to test all applicable actions that the cluster may call against that device.

If attempting to diagnose a failure that has occurred, then the diagnostic process should first identify where in the sequence of events it is having a problem in the specific action the is requested, such as: `status` , `off` , `on` , or `reboot` on a cluster node. This process may involve:

- Using `ssh` , `telnet` , or whatever remote protocol is used to connect to the device to manually login and test the fence device or see what output is given.
- Test the fence agent manually.

- Test that the options for the fencing agent in the `/etc/cluster/cluster.conf` are correct, and match those that were observed to work properly when used on the command line.

Use the remote protocol to manual login and test the fence device

Try manually logging into the fence device with the protocol that the agent uses, such as `ssh`, `telnet`, `HTTP`, etc, that the fencing agent uses for remote access. For example, if there is a problem with the fencing agent `fence_ipmilan` then try to remotely log in with `ipmitool`. Take note of the options used when logging in manually because those options might be needed when using the fencing agent.

If you are unable to login to the fence device, then verify that there is nothing preventing access (firewall, is pingable, etc) to the fence device, remote access is enabled on the fencing agent, the credentials are correct, etc.

Test the fence agent manually

When testing the fence agent manually you will usually need to know the login/password for the fence device and other information related to the fence device. It is recommend that the fencing agents `man` page is consulted to see what options are available for the fencing agent. For example, on HP machines with iLO 1 or 2 you would use `fence_ilo`, where iLO 3 or 4 uses `fence_ipmilan`.

When testing a fence device using the fence agent script, you do not need to have cluster services running. However, you must have credentials to sign in to your fence device. Check the vendor information for your selected fence device for more information.

```
# man fence_ilo
```

Then, construct commands similar to the following to reboot one node from the other. Replace the IP address for the node you want to fence, and provide the user name and password of an iLO user that has power on/off permissions on the iLO device. Reboot is the default action for this agent, but you can add `-o reboot` if desired:

```
# fence_ilo -a <ipaddress> -l <username> -p <password>
```

You can also use `-o status` to check the status of the other node's fence device interface without actually fencing it:

```
# fence_ilo -a <ipaddress> -l <username> -p <password> -o status
```

If the fence agent failed to properly do a *status*, *off*, *on*, or *reboot* action then, check the hardware. the configuration of the fence device, and check the syntax of your commands. In addition, get the debug output that can be enabled with the fencing agent script. The debug output is useful for some fencing agents to see where the fencing agent script is failing when logging into the fence device.

```
# fence_ilo -a <ipaddress> -l <username> -p <password> -o status -D /tmp/$(hostname)-<fence_agent>.debug
```

For `fence_vmware_soap` there is an article that describes how to enable enhanced debugging: Is there a way to get more debugging output from the fencing agent `fence_vmware_soap` ?.

NOTE: Use the "STDIN PARAMETERS" section in the agent's man page to determine what XML attribute to use in `/etc/cluster/cluster.conf` for each argument used at the command line.

Test the configured fencing agent for `cman` or `pacemaker` clusters

`cman` -based clusters (RHEL 6 only)

Once the fence device is configured either in `Conga` or manually in `/etc/cluster/cluster.conf`, you should have the following:

- At least one `fencedevice` configured, depending on how your selected agent works.
- A fence method configured for each node that uses a configured fence device.

This test uses the tool `fence_node` and requires that `cman` is running, The `fence_node` command reads the cluster configuration from `cman` and calls the fence agent as configured to execute the fence action.

To check the status of a cluster node which will not cause any change to the cluster, use `fence_node -S` followed by the node name from the cluster configuration::

```
# fence_node -S <node_name>
```

To reboot a specific node, use `fence_node` followed by the node name from the cluster configuration:

```
# fence_node <node_name>
```

If `fence_node` works properly, that means the fencing configuration for the cluster should work when a fence event occurs. If it fails, update your cluster configuration as needed to fix the issues.

pacemaker -based clusters

NOTE: Technically in RHEL 6, pacemaker clusters are also cman -based, however in these configurations pacemaker still manages the fencing configuration. So, this section applies to RHEL 6 pacemaker clusters with cman in addition to RHEL 7 pacemaker clusters with corosync , and the above section is for other cman -based clusters without pacemaker .

Once stonith devices have been created in the CIB using pcs stonith commands or the pcsd web interface, a fence action can be triggered manually to test whether the configuration works.

To check the status of a cluster node which will not cause any change to the cluster, use fence_node -S (on RHEL 6 only) followed by the node name from the cluster configuration (fence_node is not included on RHEL 7+):

```
# fence_node -S <node_name>
```

To reboot a specific node:

```
# fence_node <node_name> or # pcs stonith fence <node_name>
```

The pcs stonith fence command reads the cluster configuration from cib and calls the fence agent as configured to execute the fence action.

If pcs stonith fence works properly, that means the fencing configuration for the cluster should work when a fence event occurs. If it fails, update your cluster configuration as needed to fix the issues.

The fencing agent continues to fail after all the debugging steps have been done

In some situations, the fence agent scripts may work successfully, but fence_node or pcs stonith fence fails. When this happens, it means that cluster management can not invoke fence device through the configuration it has retrieved. This may be a problem with the settings used on the command line not matching what's in the configuration, special characters or could sometimes be attributed to SELinux.

In addition, if the fence agent that is being tested is a fence_drac , fence_ilo , or some other fencing agent for a systems management device that continues to fail then fall back to trying fence_ipmilan . Most systems management cards support IPMI remote login and the only supported fencing agent is fence_ipmilan . The following articles explain why the only supported fencing agent is fence_ipmilan for certain versions of a fencing device.

- How to configure `fence_ipmilan` with Red Hat Enterprise Linux High Availability Add On
- `fence_drac` and `fence_drac5` will not fence my cluster node with an iDRAC6.
- The fencing agent `fence_idrac` fails to work with iDRAC7

NOTE: This is not an exhaustive list of fence devices where `fence_ipmilan` is the only supported fencing agent.

Test whether the cluster will respond correctly during a fence event

This testing ensures that a node will fence successfully when it is triggered automatically during a cluster fence event. To do this, take an action in the cluster that should initiate a token loss, such as pulling the network cable, disabling a switch port, or simulating a crash with `sysrqtrigger` (do note that triggering a kernel panic can cause data lost and recommended to have your clustered resources disabled).

Debugging Articles for Specific Fencing Agents

- How can I diagnose `fence_ipmilan` failures in RHEL 5, 6, or 7?
- How can I diagnose `fence_ilo` failures when connecting to an iLO version 1 or 2 device in RHEL 5, 6, or 7?
- How can I diagnose `fence_vmware` failures in RHEL 5, 6, or 7?
- Is there a way to get more debugging output from the fencing agent `fence_vmware_soap` ?

Fencing Articles

- Fence Device and Agent Information for Red Hat Enterprise Linux
- Fencing in Red Hat Enterprise Linux: Methods, Use Cases, and Failover
- Running a fence agent from the command line works, but `fence_node` fails to fence a node in a RHEL High Availability cluster
- How do you manually call fencing agents from the commandline?

Root Cause

A cluster node is not being fenced properly with either `fence_node` or by a fencing agent script.

Fencing is the disconnection of a node from a cluster's shared storage or the removal of power from the cluster node (by powering off or rebooting) which ensures that the shared storage data integrity is saved or the cluster is in a sane state. The fencing is performed by a fencing agent which is a script or program used to interact with a specific class of fence devices, sometimes those being of a particular brand of hardware (`ilo`, `drac`, etc) or sometimes those implementing a particular standard (`ipmi`, etc).

When a cluster node is elected to be evicted from the cluster, another surviving member of the cluster will fence that member off from the rest of the cluster. In some instances the fencing agent will fail to fence the evicted cluster node off from the cluster which could be because of a configuration issue in the `cluster.conf`, the wrong fencing agent, network issue, etc.

Product(s) Red Hat Enterprise Linux **Component** cluster **Category** Troubleshoot

Tags cluster ha high availability fence_agent fencing rhel rhel_5 rhel_6 rhel_7 rhel_8

This solution is part of Red Hat's fast-track publication program, providing a huge library of solutions that Red Hat engineers have created while supporting our customers. To give you the knowledge you need the instant it becomes available, these articles may be presented in a raw and unedited form.

People who viewed this solution also viewed

[How to create/use custom fence agent with Red Hat Cluster?](#)

Solution - 8 thg 4, 2013

[How do I configure my highly available cluster to enable access to storage devices managed by fence agents in RHEL 6 or 7?](#)

Solution - 23 thg 6, 2014

[When a power_wait is specified for the fencing agent in a node can fail to rejoin the cluster](#)

Solution - 31 thg 10, 2017

2 Comments



ACTIVE
CONTRIBUTOR

143 Points

8 March 2016 4:36 AM

erxu su

Sometimes failed ,because our BMC only supports lanplus .

↩ Reply



NEWBIE

5 Points

27 March 2018 5:12 PM

Andrew Kelling

Link to "specific class of fence devices, " is broken

↩ Reply

Copyright © 2021 Red Hat, Inc.