**RHEL-8 Pacemaker cluster on VMware.**



RHEL8

Pacemaker setup guide V 1.9

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | | |
| **Document name** | HCL Technical documentation for Pacemaker setup V1.8.docx | | |
|  |  | | |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
| **Date** | 2023-06-07 | **Approved date** | 2023-07-04 |

Table of Contents

[1. About this document. 2](#_Toc140486422)

[2. Revision history. 2](#_Toc140486423)

[3. Pacemaker Cluster installation instruction on RHEL8 vm running on vmware. 2](#_Toc140486424)

[4. Prerequisite for installing 2 Node Pacemaker cluster. 2](#_Toc140486425)

[4.1 Ports to Enable for High Availability Add-On. 4](#_Toc140486426)

[4.2 Prepare All Nodes: 5](#_Toc140486427)

[4.3 Install the required packages on All Nodes 6](#_Toc140486428)

[5. Begin the Cluster setup 6](#_Toc140486429)

[6. Configure Fencing for the cluster. 7](#_Toc140486430)

[6.1 What is Fencing. 7](#_Toc140486431)

[6.2 Add the fence\_vmware\_rest stonith to pacemaker. 8](#_Toc140486432)

[6.3 Test the service account access to act for cluster node. 9](#_Toc140486433)

[6.4 Test to Fence the cluster nodes one by one. 10](#_Toc140486434)

[6.5 Delay fencing to prevent fence races when using a shared stonith device in a two-node cluster by enabling pcmk\_delay\_max 13](#_Toc140486435)

[7. Configure shared storage disk. 13](#_Toc140486436)

[7.1 Configure the iSCSI initiator nodes initiator’s details on the Cluster Nodes. 14](#_Toc140486437)

[7.2 Discover and login to the iScsi target on both cluster nodes 14](#_Toc140486438)

[7.3 Verify the shared disk coming from NETAPP storage. 15](#_Toc140486439)

[8. Configure DM Multipath. 16](#_Toc140486440)

[Checking for the device-mapper-multipath package. 16](#_Toc140486441)

[Setting up DM Multipath for a basic failover configuration. 16](#_Toc140486442)

[9. Cluster filesystem creation 18](#_Toc140486443)

[Configure HA-LVM “active/passive” shared storage using system\_id. 18](#_Toc140486444)

[Create a filesystem for the Application 18](#_Toc140486445)

[10. Storing Web content on shared storage. 21](#_Toc140486446)

[11. Create PCS cluster resources. 22](#_Toc140486447)

[12. PCS resource constraints. 23](#_Toc140486448)

[13. Testing the cluster resource failover. 25](#_Toc140486449)

# About this document.

This Document guide you to install Pacemaker cluster on RHE8 OS running on vmware, along with following functionality:

* Installing Pacmaker, highavailability and other required packages for pacmaker cluster setup.
* Setup Fencing device through fence\_vmware\_rest.
* Listing fence\_vmware\_rest stonith device configuration.
* Testing fencing functionality by fencing the other node of cluster
* Configure the iscsi initiator and target.
* Configure Multipath for iscsi shared storage.

# Revision history.

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Version | Description | Changed by |
| 2022-08-24 | 1.1 | Initial Draft | Anuj S Chauhan |
| 2023-07-04 | 1.8 | Fencing device creation update | Anuj S Chauhan |
| 2023-07-17 | 1.8 | Adding details regarding disk mgmt | Anuj S Chauhan |
| 2023-08-02 | 1.9 | Fix some detail | Benoit G |

# Pacemaker Cluster installation instruction on RHEL8 vm running on vmware.

Introduction: Follow below mentioned instruction to download the required packages and do the required configuration for setup the pacemaker cluster on RHEL8 vm running on Vmware. Note: hostname/computer name that we are using in this document is our test machine name, so in your case it will be your Server hostname name on which you are going to install the PCS cluster.

# Prerequisite for installing 2 Node Pacemaker cluster.

Following is environment specifications which is going to use throughout this documentation.

Note: ESX host must be in different Data Center and each Vm should be on different ESX host.

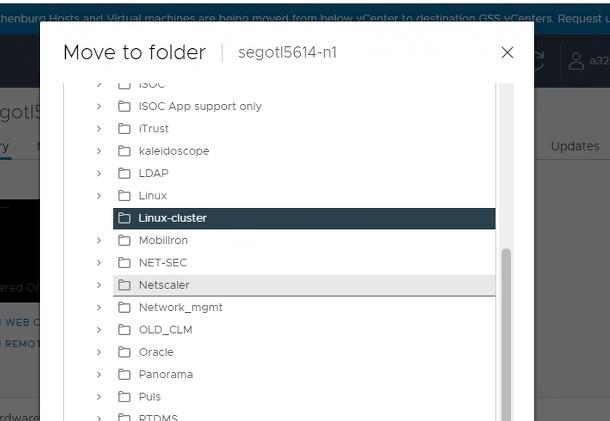
|  |  |  |  |
| --- | --- | --- | --- |
|  | Linux VM1 DC-1 | Linux VM2 DC-2 | VLAN |
|  |  |  |  |
| Hostname | segotlxxxx-n1 | Segotlxxxx-n2 |  |
|  |  |  |  |
| IP Address Nic1 | PROD IP | PROD IP | OS VLAN |
| IP Address Nic2 | IP A1 | IP B1 | Storage IP VLAN1 |
| IP Address Nic3 | IP A2 | IP B2 | Storage IP VLAN2 |
| VIP | Floating VIP | |  |
| OS | RHEL-8 | RHEL-8 |
| Shared Disk | n Gb | |
| /etc/hosts | Should have entrée for each cluster nodes or have DNS entrée for cluster nodes | |  |

Prerequisite  **Table 4.1.**

**Requesting VMs in vcenter:**

VMs for PCS cluster must be requested under the Linux-cluster folder of Vcenter

Refer the following captured screenshot showing the actual folder name configured for Linux-cluster

****

**Configure the PROD – IP and A1 & A2 Ips on each vms.**

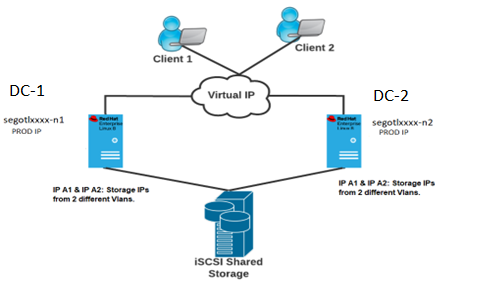
PROD IP: Production IP for VM

IP A1 & IP A2: Storage IPs from 2 different Vlans.

VIP: 1 Floating IP that will be used for application service configuration.

OS: Operating system RHEL8

N Gb: Requested storage size



**PCS: HLD fig 4.2.**

## **4.1 Ports to Enable for High Availability Add-On**.

[**Table 4.3**, “**Ports to Enable for High Availability Add-On**”](https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/7/html/high_availability_add-on_reference/s1-firewalls-HAAR#tb-portenable-HAAR) shows the ports to enable for the Red Hat High Availability Add-On and provides an explanation for what the port is used for.

**Table 4.3. Ports to Enable for High Availability Add-On**

| **Port** | **When Required** |
| --- | --- |
| TCP 2224 | Required on all nodes (needed by the pcsd Web UI and required for node-to-node communication)  It is crucial to open port 2224 in such a way that pcs from any node can talk to all nodes in the cluster, including itself. When using the Booth cluster ticket manager or a quorum device you must open port 2224 on all related hosts, such as Booth arbiters or the quorum device host. |
| TCP 3121 | Required on all nodes if the cluster has any Pacemaker Remote nodes  Pacemaker's crmd daemon on the full cluster nodes will contact the pacemaker\_remoted daemon on Pacemaker Remote nodes at port 3121. If a separate interface is used for cluster communication, the port only needs to be open on that interface. At a minimum, the port should open on Pacemaker Remote nodes to full cluster nodes. Because users may convert a host between a full node and a remote node, or run a remote node inside a container using the host's network, it can be useful to open the port to all nodes. It is not necessary to open the port to any hosts other than nodes. |
| TCP 5403 | Required on the quorum device host when using a quorum device with corosync-qnetd. The default value can be changed with the -p option of the corosync-qnetd command. |
| UDP 5404 | Required on corosync nodes if corosync is configured for multicast UDP |
| UDP 5405 | Required on all corosync nodes (needed by corosync) |
| TCP 21064 | Required on all nodes if the cluster contains any resources requiring DLM (such as clvm or GFS2) |
| TCP 9929, UDP 9929 | Required to be open on all cluster nodes and booth arbitrator nodes to connections from any of those same nodes when the Booth ticket manager is used to establish a multi-site cluster. |

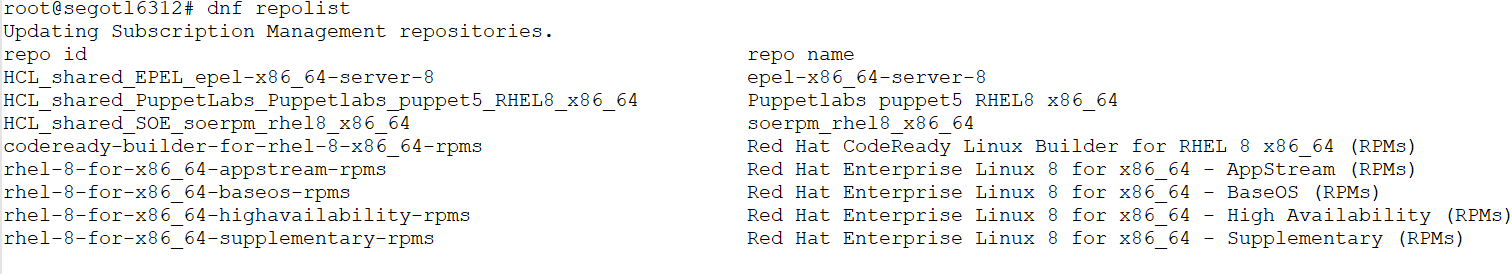
## **4.2** **Prepare All Nodes:**

On all nodes, registered system to RHSM or the required package manger server to have the Following Repositories.

* **rhel-8-for-x86\_64-appstream-rpms**
* **rhel-8-for-x86\_64-highavailability-rpms**
* **rhel-8-for-x86\_64-appstream-rpms**
* **rhel-8-for-x86\_64-baseos-rpms**

# subscription-manager register

# subscription-manager repos --enable=rhel-8-for-x86\_64-highavailability-rpms



## **4.3 Install the required packages on All Nodes**

Register all nodes system to RHSM or available package manager. Install the Red Hat High Availability Add-On software packages from the High Availability channel, and start and enable the pcsd service, along with other required packages.

# dnf install -y pcs pacemaker fence-agents-all pcp-zeroconf psmisc policycoreutils-python-utils lvm2 chrony iscsi-initiator-utils

# systemctl start pcsd.service

# systemctl enable pcsd.service

* If you are running the firewalld daemon, enable the ports that are required by the Red Hat High Availability Add-On.

# firewall-cmd --permanent --add-service=high-availability

# firewall-cmd --add-service=high-availability

# firewall-cmd --reload

# Begin the Cluster setup

* Set a password for user hacluster on each node in the cluster and authenticate user hacluster for each node in the cluster on the node from which you will be running the pcs commands. This example is using only a single node, the node from which you are running the commands, but this step is included here since it is a necessary step in configuring a supported Red Hat High Availability multi-node cluster.

1. # passwd hacluster
2. # root@segotlxxxx-n1# pcs host auth segotlxxxx-n1 segotlxxxx-n2

Username: hacluster

Password:<give the password just setup in step 1>

segotlxxxx-n1: Authorized

segotlxxxx-n2: Authorized

1. # pcs cluster setup cdopcs --start segotlxxxx-n1 segotlxxxx-n2
2. root@segotlxxxx-n1# pcs cluster start –all

segotlxxxx-n1: Cluster Enabled

segotlxxxx-n2: Cluster Enabled

**(We mustn’t enable cluster service because pcsd service must be start manually in case of reboot or failure)**

1. root@segotlxxxx-n1# pcs cluster status

Cluster Status:

Cluster Summary:

\* Stack: corosync

\* Current DC: segotlxxxx-n2 (version 2.1.0-8.el8-7c3f660707) - partition with quorum

\* Last updated: Thu Feb 3 08:43:48 2022

\* Last change: Thu Feb 3 08:43:22 2022 by hacluster via crmd on segotlxxxx-n2

\* 2 nodes configured

\* 0 resource instances configured

Node List:

\* Online: [ segotlxxxx-n1 segotlxxxx-n2 ]

PCSD Status:

segotlxxxx-n1: Online

segotlxxxx-n2: Online

# Configure Fencing for the cluster.

## **What is Fencing.**

Red Hat High Availability cluster requires that you configure fencing for the cluster, If communication with a single node in the cluster fails, then other nodes in the cluster must be able to restrict or release access to resources that the failed cluster node may have access to. This cannot be accomplished by contacting the cluster node itself as the cluster node may not be responsive. Instead, you must provide an external method, which is called fencing with a fence agent. A fence device is an external device that can be used by the cluster to restrict access to shared resources by an errant node, or to issue a hard reboot on the cluster node.

Without a fence device configured you do not have a way to know that the resources previously used by the disconnected cluster node have been released, and this could prevent the services from running on any of the other cluster nodes. Conversely, the system may assume erroneously that the cluster node has released its resources and this can lead to data corruption and data loss. Without a fence device configured data integrity cannot be guaranteed and the cluster configuration will be unsupported.

The key aspect of Red Hat cluster design is that a system must be configured with at least one fencing device to ensure that the services that the cluster provides remain available when a node in the cluster encounters a problem. Fencing is the mechanism that the cluster uses to resolve issues and failures that occur. When you design your cluster services to take advantage of fencing, you can ensure that a problematic cluster node will be cut off quickly and the remaining nodes in the cluster can take over those services, making for a more resilient and stable cluster.

## **Add the fence\_vmware\_rest stonith to pacemaker.**

* Should have one service account/user. At Vcenter where VMs are running. (Must be requested via <https://npam.srv.volvo.com> for volvo.)
* Note: Service account password must avoid some special characters (it trigs authentication issue). So, **request password with only following special characters**

! ; - \_`

* Suggestion for the service account name should be something like

**“cs-ws-s-*clustername*fe”**

* Add service account to **CS-XS-VM-Linux-cluster-fencing** group.
* Put the Linux operational team functional mailbox *SupportJavaServer.O@hcl.com*. As the backup owner and backup password owner while requesting service account.

Due to limitation on npam tool we can’t define our hcl functional mailbox as owner of the account and password.

* configuredvCenter user account needs to have role with the following four permissions set in vSphere. This is in place when you add service account in the cs-xs-vm-Linux-cluster-fencing group. This group is map with the role **role-Linux-cluster-fencing** in vcenter which provides required authorization.

System.Anonymous

System.View

VirtualMachine.Interact.PowerOff

VirtualMachine.Interact.PowerOn

## **Test the service account access to act for cluster node.**

**Uses:**

fence\_vmware\_soap -a Vcenter ip or url -l 'SERVICEACCOUNT@FQDN' -p 'XXXXXXX' --ssl-insecure -o list

root@segotlxxxx-n1# fence\_vmware\_soap -a segotvxxxxx -l 'SERVICEACCOUNT@FQDN' -p 'XXXXXXX' --ssl-insecure -o list

segotlxxxx-n1,423912cd-36a5-af85-637c-0c86731d54d2

segotlxxxx-n2,423914e3-13ba-1d47-8947-e9cfe0a5f506

root@segotlxxxx-n1# fence\_vmware\_soap -a segotv00818.it.hclgss.com -l 'SERVICE ACCOUNT@FQDN' -p 'XXXXXXXX' --ssl-insecure -o reboot -n segotlxxxx-n2

Success: Rebooted

Once the above testing is good then you can, use the fence\_vmware\_soap to fence the nodes, because both nodes will be fenced by the same fencing agent, you can configure both fencing devices as a single resource, using the pcmk\_host\_map option. pcmk\_host\_map use to specify node mappings to stonith devices.

Create a fencing device by configuring the device as a stonith resource with the pcs stonith create command.

The following command configures a stonith resource named "vmfence" that uses the fence\_vmware\_soap for nodes segotlxxxx-n1 and segotlxxxx-n2.

In the following steps we are giving 1st value as the stonith device name at the cluster level and 2nd value is the actual name of the server that is to be used to map with fencing device.

pcmk\_host\_map="segotlxxxx-n1:segotlxxxx-n1;segotlxxxx-n2:segotlxxxx-n2"

The login value and password for vmware vcenter device.

ip=**vcenterurl/ip** ssl\_insecure=1 username=<serviceaccount@fqdn > password=XXXXX

root@segotlxxxx-n1# pcs property set stonith-enabled=true

Set the vmfence pcmk\_reboot\_action=off, so that the fenced node should be shutdown instead of reboot

root@segotlxxxx-n1#pcs stonith update vmfence pcmk\_reboot\_action=off

root@segotlxxxx-n1# pcs stonith create vmfence fence\_vmware\_soap pcmk\_host\_map="segotlxxxx-n1:segotlxxxx-n1;segotlxxxx-n2:segotlxxxx-n2" ip=170.102.128.47 ssl\_insecure=1 username=<serviceaccount@domain > password=XXXXX

**# Check the status of stonith device.**

root@segotlxxxx-n1# pcs stonith status

\* vmfence (stonith:fence\_vmware\_soap): Started segotlxxxx-n1

**# To see the configuration attributes of the stonith device.**

root@segotlxxxx-n1# pcs stonith config vmfence

Resource: vmfence (class=stonith type=fence\_vmware\_soap)

Attributes: ipaddr=170.102.128.47 login=serviceaccount passwd=XXXXX pcmk\_host\_map=segotlxxxx-n1:segotlxxxx-n1;segotlxxxx-n2:segotlxxxx-n2 ssl\_insecure=1

Operations: monitor interval=60s (vmfence-monitor-interval-60s)

**# Check the cluster status we should have fencing device configured and running on one of the node of cluster.**

root@segotlxxxx-n1# pcs status

Cluster name: cdopcs

Cluster Summary:

\* Stack: corosync

\* Current DC: segotlxxxx-n2 (version 2.1.2-4.el8\_6.2-ada5c3b36e2) - partition with quorum

\* Last updated: Thu Aug 4 12:06:28 2022

\* Last change: Thu Aug 4 12:04:34 2022 by root via cibadmin on segotlxxxx-n1

\* 2 nodes configured

\* 1 resource instance configured

Node List:

\* Online: [ segotlxxxx-n1 segotlxxxx-n2 ]

Full List of Resources:

\* vmfence (stonith:fence\_vmware\_soap): Started segotlxxxx-n1

Daemon Status:

corosync: active/enabled

pacemaker: active/enabled

pcsd: active/enabled

## **Test to Fence the cluster nodes one by one.**

root@segotlxxxx-n2# pcs stonith fence segotlxxxx-n1

Node: segotlxxxx-n1 fenced

root@segotlxxxx-n1# pcs status

Cluster name: cdopcs

Cluster Summary:

\* Stack: corosync

\* Current DC: segotlxxxx-n2 (version 2.1.2-4.el8\_6.2-ada5c3b36e2) - partition with quorum

\* Last updated: Thu Aug 4 12:10:39 2022

\* Last change: Thu Aug 4 12:04:34 2022 by root via cibadmin on segotlxxxx-n1

\* 2 nodes configured

\* 1 resource instance configured

Node List:

\* OFFLINE: [ segotlxxxx-n1 ]

\* Online: [ segotlxxxx-n2 ]

Full List of Resources:

\* vmfence (stonith:fence\_vmware\_soap): Started segotlxxxx-n2

Daemon Status:

corosync: active/enabled

pacemaker: active/enabled

pcsd: active/enabled

**Login to the Vcenter and power on the vm manually.**

***ssh to the node and start the cluster services, to join the node in cluster.***

root@segotlxxxx-n1# **pcs cluster start**

**Starting Cluster...**

**root@segotl6313#**

root@segotlxxxx-n1# pcs stonith fence segotlxxxx-n2

Node: segotlxxxx-n2 fenced

root@segotlxxxx-n1#

root@segotlxxxx-n1# pcs status

Cluster name: cdopcs

Cluster Summary:

\* Stack: corosync

\* Current DC: segotlxxxx-n1 (version 2.1.2-4.el8\_6.2-ada5c3b36e2) - partition with quorum

\* Last updated: Thu Aug 4 12:16:38 2022

\* Last change: Thu Aug 4 12:16:14 2022 by root via cibadmin on segotlxxxx-n1

\* 2 nodes configured

\* 1 resource instance configured

Node List:

\* Online: [ segotlxxxx-n1 ]

\* Online: [ segotlxxxx-n2 ]

Full List of Resources:

\* vmfence (stonith:fence\_vmware\_soap): Started segotlxxxx-n1

# Test the other node to Fence.

root@segotlxxxx-n1# pcs stonith fence segotlxxxx-n2

Node: segotlxxxx-n2 fenced

root@segotlxxxx-n1#

root@segotlxxxx-n1# pcs status

Cluster name: cdopcs

Cluster Summary:

\* Stack: corosync

\* Current DC: segotlxxxx-n1 (version 2.1.2-4.el8\_6.2-ada5c3b36e2) - partition with quorum

\* Last updated: Thu Aug 4 12:16:38 2022

\* Last change: Thu Aug 4 12:16:14 2022 by root via cibadmin on segotlxxxx-n1

\* 2 nodes configured

\* 1 resource instance configured

Node List:

\* Online: [ segotlxxxx-n1 ]

\* OFFLINE: [ segotlxxxx-n2 ]

Full List of Resources:

\* vmfence (stonith:fence\_vmware\_soap): Started segotlxxxx-n1

Follow the same procedure, as mentioned for the segotlxxxx-n1 to join the node back cluster.

# Following are the logs visible in /var/log/messages to know how fencing working at backend.

# root@segotlxxxx-n1# tail -100 /var/log/messages

Aug 4 12:23:27 segotlxxxx-n1 pacemaker-fenced[1693]: notice: Client stonith\_admin.6152 wants to fence (reboot) segotlxxxx-n2 using any device

Aug 4 12:23:27 segotlxxxx-n1 pacemaker-fenced[1693]: notice: Requesting peer fencing (reboot) targeting segotlxxxx-n2

Aug 4 12:23:27 segotlxxxx-n1 pacemaker-fenced[1693]: notice: vmfence is eligible to fence (reboot) segotlxxxx-n2: static-list

Aug 4 12:23:27 segotlxxxx-n1 pacemaker-fenced[1693]: notice: Requesting that segotlxxxx-n1 perform 'reboot' action targeting segotlxxxx-n2

=== snip ====

Aug 4 12:23:33 segotlxxxx-n1 pacemaker-fenced[1693]: notice: Operation 'reboot' [6153] targeting segotlxxxx-n2 using vmfence returned 0

Aug 4 12:23:34 segotlxxxx-n1 corosync[1336]: [TOTEM ] A processor failed, forming new configuration: token timed out (3000ms), waiting 3600ms for consensus.

Node: segotlxxxx-n2 fenced

Aug 4 12:23:37 segotlxxxx-n1 corosync[1336]: [QUORUM] Sync members[1]: 1

Aug 4 12:23:37 segotlxxxx-n1 corosync[1336]: [QUORUM] Sync left[1]: 2

Aug 4 12:23:37 segotlxxxx-n1 corosync[1336]: [TOTEM ] A new membership (1.43) was formed. Members left: 2

Aug 4 12:23:37 segotlxxxx-n1 corosync[1336]: [TOTEM ] Failed to receive the leave message. failed: 2

Aug 4 12:23:37 segotlxxxx-n1 corosync[1336]: [QUORUM] Members[1]: 1

Aug 4 12:23:37 segotlxxxx-n1 corosync[1336]: [MAIN ] Completed service synchronization, ready to provide service.

Aug 4 12:23:37 segotlxxxx-n1 pacemaker-attrd[1704]: notice: Node segotlxxxx-n2 state is now lost

Aug 4 12:23:37 segotlxxxx-n1 pacemaker-attrd[1704]: notice: Removing all segotlxxxx-n2 attributes for peer loss

Aug 4 12:23:37 segotlxxxx-n1 pacemaker-attrd[1704]: notice: Purged 1 peer with id=2 and/or uname=segotlxxxx-n2 from the membership cache

**Aug 4 12:23:37 segotlxxxx-n1 pacemaker-based[1692]: notice: Node segotlxxxx-n2 state is now lost**

Aug 4 12:23:37 segotlxxxx-n1 pacemaker-based[1692]: notice: Purged 1 peer with id=2 and/or uname=segotlxxxx-n2 from the membership cache

Aug 4 12:23:37 segotlxxxx-n1 pacemaker-controld[1706]: notice: Node segotlxxxx-n2 state is now lost

=== snip ======

Aug 4 12:23:38 segotlxxxx-n1 pacemaker-controld[1706]: notice: Peer segotlxxxx-n2 was terminated (reboot) by segotlxxxx-n1 on behalf of stonith\_admin.6152@segotlxxxx-n1: OK

## **Delay fencing to prevent fence races when using a shared stonith device in a two-node cluster by enabling pcmk\_delay\_max**

In a two-node cluster, both nodes can attempt to fence each other simultaneously, causing both nodes to reboot or power off. This happens most commonly when there is an issue with the heartbeat network, where both nodes are healthy but cannot communicate with each other.

Enable a random delay of up to the time specified before executing stonith actions. This is sometimes used in two-node clusters to ensure that the nodes don’t fence each other at the same time. The overall delay introduced by pacemaker is derived from this random delay value adding a static delay so that the sum is kept below the maximum delay. In many cases, the delay attribute can prevent these "fence races.

# pcs stonith update vmfence pcmk\_delay\_max=15

At This point we are good to go for cluster resource configuration

# Configure shared storage disk.

Perquisite please refer the fig,

- You have to secure two storage ips on each node from two different vlans.

- Configure the both ips on both storage nics.

- Configure the iSCSI initiator that will make connection to the storage using the iSCSI protocol.

- Ask storage team to assign one shared disk and map to the both nodes iscsi initiator.

## **Configure the iSCSI initiator nodes initiator’s details on the Cluster Nodes.**

ON segotlxxxx-n1 just change the hostname at the end of the string as per your hostname.

root@segotlxxxx-n1# **cat /etc/iscsi/initiatorname.iscsi**

InitiatorName=iqn.1994-05.com.redhat:segotlxxxx-n1

ON segotlxxxx-n2 just change the hostname at the end of the string as per your hostname.

root@segotlxxxx-n2# **cat /etc/iscsi/initiatorname.iscsi**

InitiatorName=iqn.1994-05.com.redhat:segotlxxxx-n2

## **Discover and login to the iScsi target on both cluster nodes**

Note: Following steps to be performed on both nodes

**iscsiadm -m discovery -t st -p < storage IP >:3260 –l**

root@segotlxxxx-n1# iscsiadm -m session --rescan

Rescanning session [sid: 1, target: iqn.1992-08.com.netapp:sn.c30db59246f911e98b7f00a098c5187f:vs.137, portal: 172.20.56.22,3260]

Rescanning session [sid: 2, target: iqn.1992-08.com.netapp:sn.c30db59246f911e98b7f00a098c5187f:vs.137, portal: 172.20.56.23,3260]

Rescanning session [sid: 3, target: iqn.1992-08.com.netapp:sn.c30db59246f911e98b7f00a098c5187f:vs.137, portal: 172.20.56.55,3260]

Rescanning session [sid: 4, target: iqn.1992-08.com.netapp:sn.c30db59246f911e98b7f00a098c5187f:vs.137, portal: 172.20.56.54,3260]

root@segotlxxxx-n1#

root@segotlxxxx-n1# **iscsiadm -m discovery -t st -p 172.20.56.22:3260 -l**

172.20.56.22:3260,1328 iqn.1992-08.com.netapp:sn.c30db59246f911e98b7f00a098c5187f:vs.137

172.20.56.23:3260,1336 iqn.1992-08.com.netapp:sn.c30db59246f911e98b7f00a098c5187f:vs.137

172.20.56.55:3260,1335 iqn.1992-08.com.netapp:sn.c30db59246f911e98b7f00a098c5187f:vs.137

172.20.56.54:3260,1329 iqn.1992-08.com.netapp:sn.c30db59246f911e98b7f00a098c5187f:vs.137

172.20.56.39:3260,1191 iqn.1992-08.com.netapp:sn.c30db59246f911e98b7f00a098c5187f:vs.137

172.20.56.38:3260,1190 iqn.1992-08.com.netapp:sn.c30db59246f911e98b7f00a098c5187f:vs.137

172.20.56.7:3260,1189 iqn.1992-08.com.netapp:sn.c30db59246f911e98b7f00a098c5187f:vs.137

172.20.56.6:3260,1188 iqn.1992-08.com.netapp:sn.c30db59246f911e98b7f00a098c5187f:vs.137

Logging in to [iface: default, target: iqn.1992-08.com.netapp:sn.c30db59246f911e98b7f00a098c5187f:vs.137, portal: 172.20.56.39,3260]

Logging in to [iface: default, target: iqn.1992-08.com.netapp:sn.c30db59246f911e98b7f00a098c5187f:vs.137, portal: 172.20.56.38,3260]

Logging in to [iface: default, target: iqn.1992-08.com.netapp:sn.c30db59246f911e98b7f00a098c5187f:vs.137, portal: 172.20.56.7,3260]

Logging in to [iface: default, target: iqn.1992-08.com.netapp:sn.c30db59246f911e98b7f00a098c5187f:vs.137, portal: 172.20.56.6,3260]

root@segotlxxxx-n1# systemctl enable --now iscsid

Created symlink /etc/systemd/system/multi-user.target.wants/iscsid.service → /usr/lib/systemd/system/iscsid.service.

root@segotlxxxx-n1#

## **Verify the shared disk coming from NETAPP storage.**

NETAPP storage should be visible on both nodes.

root@segotlxxxx-n1# grep "Attached SCSI" /var/log/messages

May 4 12:40:19 segotl6312 kernel: [3803401.904493] sd 35:0:0:0: [sdc] Attached SCSI disk

May 4 12:40:19 segotl6312 kernel: [3803401.907540] sd 33:0:0:0: [sdb] Attached SCSI disk

May 4 12:40:19 segotl6312 kernel: [3803401.925179] sd 34:0:0:0: [sdd] Attached SCSI disk

May 4 12:40:19 segotl6312 kernel: [3803401.935844] sd 36:0:0:0: [sde] Attached SCSI disk

May 4 12:40:19 segotl6312 kernel: sd 35:0:0:0: [sdc] Attached SCSI disk

May 4 12:40:19 segotl6312 kernel: sd 33:0:0:0: [sdb] Attached SCSI disk

May 4 12:40:19 segotl6312 kernel: sd 34:0:0:0: [sdd] Attached SCSI disk

May 4 12:40:19 segotl6312 kernel: sd 36:0:0:0: [sde] Attached SCSI disk

root@segotlxxxx-n1 ~]$ lsscsi

[0:0:0:0] disk VMware Virtual disk 2.0 /dev/sda

[3:0:0:0] cd/dvd NECVMWar VMware SATA CD00 1.00 /dev/sr0

[33:0:0:0] disk NETAPP LUN C-Mode 9300 /dev/sdc

[34:0:0:0] disk NETAPP LUN C-Mode 9300 /dev/sdb

[35:0:0:0] disk NETAPP LUN C-Mode 9300 /dev/sdd

[36:0:0:0] disk NETAPP LUN C-Mode 9300 /dev/sde

# Configure DM Multipath.

You can set up DM Multipath with the mpathconf utility. This utility creates or edits the /etc/multipath.conf multipath configuration file based on the following scenarios:

* If the /etc/multipath.conf file already exists, the mpathconf utility will edit it.
* If the /etc/multipath.conf file does not exist, the mpathconf utility will create the /etc/multipath.conf file from scratch

## Checking for the device-mapper-multipath package.

Before setting up DM Multipath on your system, ensure that your system is up-to-date and includes the device-mapper-multipath package.

Procedure.

1. Check if your system includes the the device-mapper-multipath package:

# rpm -q device-mapper-multipath

device-mapper-multipath-*current-package-version*

If your system does not include the package, it prints the following:

**package** device-mapper-multipath is not installed

1. If your system does not include the package, install it by running the following command:

# {PackageManager} install device-mapper-multipath

## Setting up DM Multipath for a basic failover configuration.

Use the following procedure to set up DM Multipath for a basic failover configuration if you need to edit the /etc/multipath.conf file before starting the multipathd daemon.

**Procedure.**

1. Enable the multipath configuration file:

# mpathconf --enable

1. Edit the /etc/multipath.conf file if necessary to give user custom name to shared storage, add following lines to

root@segotlxxxx-n1 ~]$ tail -5 /etc/multipath.conf

multipath {

wwid "3600xxxxxxxxxxxxxxx66687xxx"

alias mpatha

}

}

The default settings for DM Multipath are compiled into the system and do not need to be explicitly set in the /etc/multipath.conf file.

The default value of path\_grouping\_policy is set to failover, so in this example you do not need to edit the /etc/multipath.conf file.

The initial defaults section of the configuration file configures your system so that the names of the multipath devices are of the form /dev/mapper/mpathn; without this setting, the names of the multipath devices would be aliased to the WWID of the device. If you do not want to use user friendly names, you can enter the following command:

# mpathconf --enable --user\_friendly\_names n

If you need to edit the multipath configuration file after you have started the multipath daemon, you must execute the systemctl reload multipathd.service command for the changes to take effect.

1. Save the configuration file and exit the editor, if necessary.
2. Start the multipath daemon and create the multipath devices:

# systemctl start multipathd.service

root@segotlxxxx-n1# multipath -ll

mpatha (3600xxxxxxxxxxxxxxx66687xxx) dm-5 NETAPP,LUN C-Mode

size=100G features='3 queue\_if\_no\_path pg\_init\_retries 50' hwhandler='1 alua' wp=rw

|-+- policy='service-time 0' prio=50 status=active

| |- 33:0:0:0 sdb 8:16 active ready running

| `- 35:0:0:0 sdc 8:32 active ready running

`-+- policy='service-time 0' prio=10 status=enabled

|- 34:0:0:0 sdd 8:48 active ready running

`- 36:0:0:0 sde 8:64 active ready running

# Cluster filesystem creation

## Configure HA-LVM “active/passive” shared storage using system\_id.

1. Add following on all the cluster nodes, to enable system-id for mounting the shared volume exclusively on one node at a time.

#vi /etc/lvm/lvm.conf

system\_id\_source = "uname"

1. Take backup of current initramfs file on all the cluster nodes.

# cp /boot/initramfs-$(uname -r).img /boot/initramfs-$(uname -r).img.$(date +%m-%d-%H%M%S).bak

1. Rebuild initramfs file on both cluster nodes.

# dracut -f -v

1. Reboot all the cluster nodes.

Once all the cluster nodes are back online, proceed with creation on PV/VG/LV from any one of the cluster node

## Create a filesystem for the Application

Could be for Apache webserver to hold the website files. On any one of your nodes (Ex, segotlxxxx-n1) create a filesystem with LVM.

root@segotlxxxx-n1# pvcreate /dev/mapper/mpatha

root@segotlxxxx-n1# vgcreate --setautoactivation n vg\_apache /dev/mapper/mpatha

Volume group "vg\_apache" successfully created with system ID segotlxxxx-n1

root@segotlxxxx-n1# vgs -o+systemid

VG #PV #LV #SN Attr VSize VFree System ID

rootvg 1 5 0 wz--n- <29.02g <9.02g

vg\_apache 1 0 0 wz--n- <100.00g <100.00g segotlxxxx-n1

root@segotlxxxx-n1# lvcreate -n lv\_apache -l 100%FREE vg\_apache

WARNING: ext4 signature detected on /dev/vg\_apache/lv\_apache at offset 1080. Wipe it? [y/n]: y

Wiping ext4 signature on /dev/vg\_apache/lv\_apache.

Logical volume "lv\_apache" created.

root@segotlxxxx-n1#

root@segotlxxxx-n1#

root@segotlxxxx-n1# lvs

LV VG Attr LSize Pool Origin Data% Meta% Move Log Cpy%Sync Convert

homevol rootvg -wi-ao---- 2.00g

rootvol rootvg -wi-ao---- 6.00g

swap01 rootvg -wi-ao---- 2.00g

tmpvol rootvg -wi-ao---- 4.00g

varvol rootvg -wi-ao---- 6.00g

lv\_apache vg\_apache -wi-a----- <100.00g

root@segotlxxxx-n1# mkfs.ext4 /dev/vg\_apache/lv\_apache

mke2fs 1.45.6 (20-Mar-2020)

Creating filesystem with 26213376 4k blocks and 6553600 inodes

Filesystem UUID: dea22953-2892-4e62-9637-914197c6590a

Superblock backups stored on blocks:

32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632, 2654208, 4096000, 7962624, 11239424, 20480000, 23887872

Allocating group tables: done

Writing inode tables: done

Creating journal (131072 blocks): done

Writing superblocks and filesystem accounting information: done

# Ensuring a volume group is not activated on multiple cluster nodes (RHEL 8.4 and earlier)

This procedure ensures that volume groups that are managed by Pacemaker in a cluster will not be automatically activated on startup. If a volume group is automatically activated on startup rather than by Pacemaker, there is a risk that the volume group will be active on multiple nodes at the same time, which could corrupt the volume group’s metadata.

This procedure modifies the auto\_activation\_volume\_list entry in the /etc/lvm/lvm.conf configuration file. The auto\_activation\_volume\_list entry is used to limit autoactivation to specific logical volumes. Setting auto\_activation\_volume\_list to an empty list disables autoactivation entirely.

Any local volumes that are not shared and are not managed by Pacemaker should be included in the auto\_activation\_volume\_list entry, including volume groups related to the node’s local root and home directories. All volume groups managed by the cluster manager must be excluded from the auto\_activation\_volume\_list entry.

**Procedure**

Perform the following procedure on each node in the cluster.

1. Determine which volume groups are currently configured on your local storage with the following command. This will output a list of the currently-configured volume groups. If you have space allocated in separate volume groups for root and for your home directory on this node, you will see those volumes in the output, as in this example.

# **vgs --noheadings -o vg\_name**

vg\_apache

rootvg

Note: From step 2 to step 4 must performed on both cluster nodes.

1. Add the volume groups other than vg\_apache (the volume group you have just defined for the cluster) as entries to auto\_activation\_volume\_list in the /etc/lvm/lvm.conf configuration file.

For example, if you have space allocated in separate volume groups for root and for your home directory, you would uncomment the auto\_activation\_volume\_list line of the lvm.conf file and add these volume groups as entries to auto\_activation\_volume\_list as follows.

**Note:** that the volume group you have just defined for the cluster (vg\_apache in this example) is not in this list.

root@segotlxxxx-n1# grep -i rootvg /etc/lvm/lvm.conf

auto\_activation\_volume\_list = [ "rootvg" ]

**Note:** If no local volume groups are present on a node to be activated outside of the cluster manager, you must still initialize the auto\_activation\_volume\_list entry as auto\_activation\_volume\_list = [].

1. Rebuild the initramfs boot image to guarantee that the boot image will not try to activate a volume group controlled by the cluster. Update the initramfs device with the following command. This command can take up to a minute to complete.

root@segotlxxxx-n1# dracut -H -f /boot/initramfs-$(uname -r).img $(uname -r)

1. Reboot the node.

root@segotlxxxx-n1# reboot

# Storing Web content on shared storage.

root@segotlxxxx-n1# mount /dev/vg\_apache/lv\_apache /var/www

root@segotlxxxx-n1# mkdir /var/www/html

root@segotlxxxx-n1# mkdir /var/www/cgi-bin

root@segotlxxxx-n1# mkdir /var/www/error

root@segotlxxxx-n1# restorecon -R /var/www

root@segotlxxxx-n1# cat <<-END >/var/www/html/index.html

<html>

<body>Welcome this page is hosted by RHEL PCS HA cluster </body>

</html>

END

root@segotlxxxx-n1# umount /var/www

# 11. Create PCS cluster resources.

Let`s create a filesystem resource for the Apache server.

root@segotlxxxx-n1# pcs resource create lv\_apache ocf:heartbeat:LVM-activate vgname=vg\_apache vg\_access\_mode=system\_id --group apache

root@segotlxxxx-n1# pcs resource create httpd\_fs Filesystem device="/dev/mapper/vg\_apache-lv\_apache" directory="/var/www" fstype="ext4" --group apache

Assumed agent name 'ocf:heartbeat:Filesystem' (deduced from 'Filesystem')

root@segotlxxxx-n1# pcs resource create httpd\_vip IPaddr2 ip=10.189.70.186 cidr\_netmask=27 --group apache

Assumed agent name 'ocf:heartbeat:IPaddr2' (deduced from 'IPaddr2')

root@segotlxxxx-n1# pcs resource create httpd\_monitor apache configfile="/etc/httpd/conf/httpd.conf" statusurl="http://127.0.0.1/server-status" --group apache

Assumed agent name 'ocf:heartbeat:apache' (deduced from 'apache')

segotlxxxx-n1#

segotlxxxx-n1#

segotlxxxx-n1# pcs status

Cluster name: cdopcs

Cluster Summary:

\* Stack: corosync

\* Current DC: segotlxxxx-n1 (version 2.1.2-4.el8\_6.2-ada5c3b36e2) - partition with quorum

\* Last updated: Sun Sep 18 09:01:27 2022

\* Last change: Sun Sep 18 09:01:23 2022 by root via cibadmin on segotlxxxx-n2

\* 2 nodes configured

\* 5 resource instances configured

Node List:

\* Online: [ segotlxxxx-n1 segotlxxxx-n2 ]

Full List of Resources:

\* vmfence (stonith:fence\_vmware\_soap): Started segotlxxxx-n1

\* Resource Group: apache:

\* lv\_apache (ocf::heartbeat:LVM-activate): Started segotlxxxx-n2

\* httpd\_fs (ocf::heartbeat:Filesystem): Started segotlxxxx-n2

\* httpd\_vip (ocf::heartbeat:IPaddr2): Started segotlxxxx-n2

\* httpd\_monitor (ocf::heartbeat:apache): Started segotlxxxx-n2

Daemon Status:

corosync: active/enabled

pacemaker: active/enabled

pcsd: active/enabled

# 12. PCS resource constraints.

You can determine the behavior of a resource in a cluster by configuring constraints for that resource.

location constraints — A location constraint determines which nodes a resource can run on.

You can configure a basic location constraint to specify whether a resource prefers or avoid a node, with an optional score value to indicate the relative degree of preference for the constraint.

The following command creates a location constraint for a resource to prefer the specified node.

root@segotlxxxx-n1# pcs constraint location apache prefers segotlxxxx-n1=200

root@segotlxxxx-n1# pcs constraint location apache prefers segotlxxxx-n2=100

root@segotlxxxx-n1# pcs constraint

Location Constraints:

Resource: apache

Enabled on:

Node: segotlxxxx-n1 (score:200)

Node: segotlxxxx-n2 (score:100)

Ordering Constraints:

Colocation Constraints:

Ticket Constraints:

Resource-stickiness resource constraints.

After a node is fenced, its resources relocate to another node as expected. But when that node rejoins the resources move back to it the node, where it was running before the node fenced.

So to avoid this set the preference for the resource to stay where it is rather than fail back to its original preferred location, then adjust either the value of the relevant [resource-stickiness](https://access.redhat.com/solutions/1299213) setting Value to indicate how much the resource prefers to stay where it is.  or the score of the location constraint such that the stickiness is higher than the location score.

root@segotlxxxx-n1# pcs resource defaults resource-stickiness=500

Warning: This command is deprecated and will be removed. Please use 'pcs resource defaults update' instead.

Warning: Defaults do not apply to resources which override them with their own defined values

segotlxxxx-n1# pcs resource defaults

Meta Attrs: rsc\_defaults-meta\_attributes

resource-stickiness=500

# 13. T**esting the cluster resource failover.**

# Check the cluster status.

root@segotlxxxx-n1# pcs status

Cluster name: cdopcs

Cluster Summary:

\* Stack: corosync

\* Current DC: segotlxxxx-n1 (version 2.1.2-4.el8\_6.2-ada5c3b36e2) - partition with quorum

\* Last updated: Sun Sep 18 09:26:02 2022

\* Last change: Sun Sep 18 09:12:17 2022 by root via cibadmin on segotl6313

\* 2 nodes configured

\* 5 resource instances configured

Node List:

\* Online: [ segotlxxxx-n1 segotlxxxx-n2 ]

Full List of Resources:

\* vmfence (stonith:fence\_vmware\_soap): Started segotlxxxx-n1

\* Resource Group: apache:

\* lv\_apache (ocf::heartbeat:LVM-activate): Started segotlxxxx-n2

\* httpd\_fs (ocf::heartbeat:Filesystem): Started segotlxxxx-n2

\* httpd\_vip (ocf::heartbeat:IPaddr2): Started segotlxxxx-n2

\* httpd\_monitor (ocf::heartbeat:apache): Started segotlxxxx-n2

Daemon Status:

corosync: active/enabled

pacemaker: active/enabled

pcsd: active/enabled

# Fence the current active cluster node and all resource should fail-over to the secondary node.

root@segotlxxxx-n1# pcs stonith fence segotlxxxx-n2

# Check the cluster status all resources should switch to the **segotlxxxx-n1**

root@segotlxxxx-n1# pcs status

Cluster name: cdopcs

Cluster Summary:

\* Stack: corosync

\* Current DC: segotlxxxx-n1(version 2.1.2-4.el8\_6.2-ada5c3b36e2) - partition with quorum

\* Last updated: Sun Sep 18 09:43:00 2022

\* Last change: Sun Sep 18 09:12:17 2022 by root via cibadmin on segotlxx-n2

\* 2 nodes configured

\* 5 resource instances configured

Node List:

\* Online: [ segotlxxxx-n1 ]

\* OFFLINE: [ segotlxxxx-n2 ]

Full List of Resources:

\* vmfence (stonith:fence\_vmware\_soap): Started segotlxxxx-n1

\* Resource Group: apache:

\* lv\_apache (ocf::heartbeat:LVM-activate): Started segotlxxxx-n1

\* httpd\_fs (ocf::heartbeat:Filesystem): Started segotlxxxx-n1

\* httpd\_vip (ocf::heartbeat:IPaddr2): Started segotlxxxx-n1

\* httpd\_monitor (ocf::heartbeat:apache): Started segotlxxxx-n1

Daemon Status:

corosync: active/enabled

pacemaker: active/enabled

pcsd: active/enabled