# RHCS: Install a two-node basic cluster

Article Number: 192 | Rating: Unrated | Last Updated: Sun, Jun 3, 2018 9:26 AM

## RHCS: Install a two-node basic cluster

```
# Tested on CentOS 7




# Notes mainly from http://clusterlabs.org/pacemaker




# Note: For the commands here after, [ALL] indicates that command has
to be run on the two

#  nodes and [ONE] indicates that one needs to run it only on one of
the hosts.
```

```
# The cluster installed here uses Pacemaker and Corosync to provide
resource management

# and messaging.

#

# Pacemaker is a resource manager which, among other capabilities, is
able to detect and

# recover from the failure of various nodes, resources and services
under its control by

# using the messaging and membership capabilities provided by the
choses cluster

# infrastructure (either Corosync or Heartbeat).

#

# Pacemaker main features:

#

# - Detection and recovery of node and service-level failures

# - Storage agnostic, no requirement for shared storage

# - Resource agnostic, anything that can be scripted can be clustered

# - Supports fencing for ensuring data integrity

# - Supports large and small clusters

# - Supports both quorate and resource-driven clusters

# - Supports practically any redundancy configuration
```

```
# - Automatically replicated configuration that can be updated from
any node

# - Ability to specify cluster-wide service ordering, colocation and
anti-colocation

# - Support for advanced service types

#  - Clones: for services which need to be active on multiple nodes

#  - Multi-state: for services with multiple modes

#  (e.g. master/slave, primary/secondary)

# - Unified, scriptable cluster management tools

#

# Pacemaker components:

#

# - Cluster Information Base (CIB)

# - Cluster Resource Management daemon (CRMd)

# - Local Resource Management daemon (LRMd)

# - Policy Engine (PEngine or PE)

# - Fencing daemon (STONITHd - "Shoot-The-Other-Node-In-The-Head")
```

```
# QUORUM

# ------------------------------------------------------------------------
------------------------




# If a cluster splits into two (or more) groups of nodes that can no
longer communicate

# with each other, quorum is used to prevent resources from starting
on more nodes than

# desired, which would risk data corruption.

# A cluster has quorum when more than half of all known nodes are
online in the same

# partition (group of nodes).

# For example, if a 5-node cluster split into 3- and 2-node
partitions, the 3-node

# partition would have quorum and could continue serving resources.
If a 6-node cluster

# split into two 3-node partitions, neither partition would have
quorum; pacemaker’s

# default behavior in such cases is to stop all resources, in order
to prevent data
```

```
# corruption.

# Two-node clusters are a special case. By the above definition, a
two-node cluster would

# only have quorum when both nodes are running. This would make the
creation of a two-node

# cluster pointless, but corosync has the ability to treat two-node
clusters as if only

# one node is required for quorum.

# The pcs cluster setup command will automatically configure
two_node: 1 in corosync.conf,

# so a two-node cluster will "just work".

# Depending of the versions of corosync, if may be that you will have
to ignore quorum at

# the pacemaker level, using pcs property set no-quorum-
policy=ignore.









# INSTALLATION
```

```
# ------------------------------------------------------------------------
------------------------



# First of all, make sure that the two nodes are reachable on their
IP addresses and that

# they are known by their names:



root@nodeA:/root#> cat /etc/hosts | egrep "nodeA|nodeB"

192.168.56.101 nodeA

192.168.56.102 nodeB



root@nodeA:/root#> ssh nodeB

root@nodeB's password:

Last login: Wed Jan 24 14:13:38 2018 from 192.168.56.101

root@nodeB:/root#>



root@nodeB:/root#> ssh nodeA

root@nodeA's password:
```

```
Last login: Wed Jan 24 14:13:38 2018 from 192.168.56.102

root@nodeA:/root#>
```

```
# In order to facilitate communications, de-activate SELinux and
firewall service
```

```
# This may create significant security issues and should not be
performed on machines

# that may be exposed to the outside world, but may be appropriate
during development and

# testing on a protected host.
```

```
[ALL] sed -i 's/SELINUX=enforcing/SELINUX=disabled/'
/etc/selinux/config

[ALL] setenforce 0
```

```
[ALL] systemctl stop firewalld

[ALL] systemctl disable firewalld
```

```
# Install the needed packages

[ALL] yum install pacemaker pcs resource-agents

# Start (and enable) pcs daemon on both nodes

[ALL] systemctl start pcsd.service

[ALL] systemctl enable pcsd.service

# Configure pcs authentication
```

```
[ALL] echo "mypassword" | passwd --stdin hacluster

Changing password for user hacluster.

passwd: all authentication tokens updated successfully.



[ONE] pcs cluster auth nodeA nodeB -u hacluster -p mypassword --force

nodeA: Authorized

nodeB: Authorized







# Create the cluster and populate it with the nodes



[ONE] pcs cluster setup --force --name lar_cluster nodeA nodeB

Destroying cluster on nodes: nodeA, nodeB...

nodeA: Stopping Cluster (pacemaker)...

nodeB: Stopping Cluster (pacemaker)...

nodeA: Successfully destroyed cluster

nodeB: Successfully destroyed cluster
```

```
Sending 'pacemaker_remote authkey' to 'nodeA', 'nodeB'
nodeA: successful distribution of the file 'pacemaker_remote authkey'
nodeB: successful distribution of the file 'pacemaker_remote authkey'
Sending cluster config files to the nodes...
nodeA: Succeeded
nodeB: Succeeded


Synchronizing pcsd certificates on nodes nodeA, nodeB...
nodeA: Success
nodeB: Success
Restarting pcsd on the nodes in order to reload the certificates...
nodeA: Success
nodeB: Success



# Start the cluster
```

```
[ONE] pcs cluster start --all

nodeA: Starting Cluster...

nodeB: Starting Cluster...







# Enable necessary daemons so the cluster starts automatically on
boot-up



[ALL] systemctl enable corosync.service

[ALL] systemctl enable pacemaker.service







# Verify corosync installation



[ONE] corosync-cfgtool -s

Printing ring status.
```

```
Local node ID 1

RING ID 0

  id  = 192.168.56.101

  status  = ring 0 active with no faults



[ONE] corosync-cmapctl | grep members

runtime.totem.pg.mrp.srp.members.1.config_version (u64) = 0

runtime.totem.pg.mrp.srp.members.1.ip (str) = r(0) ip(192.168.56.101)

runtime.totem.pg.mrp.srp.members.1.join_count (u32) = 1

runtime.totem.pg.mrp.srp.members.1.status (str) = joined

runtime.totem.pg.mrp.srp.members.2.config_version (u64) = 0

runtime.totem.pg.mrp.srp.members.2.ip (str) = r(0) ip(192.168.56.102)

runtime.totem.pg.mrp.srp.members.2.join_count (u32) = 1

runtime.totem.pg.mrp.srp.members.2.status (str) = joined




# To check the status of the cluster run either one the two following
commands
```

```
[ONE] pcs status

Cluster name: lar_cluster

WARNING: no stonith devices and stonith-enabled is not false

Stack: corosync

Current DC: nodeB (version 1.1.16-12.el7-94ff4df) - partition with
quorum

Last updated: Thu Feb  8 17:39:33 2018

Last change: Thu Feb  8 17:37:46 2018 by hacluster via crmd on nodeB



2 nodes configured

0 resources configured



Online: [ nodeA nodeB ]



No resources
```

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




[ONE] crm_mon -1
Stack: corosync
Current DC: nodeB (version 1.1.16-12.el7-94ff4df) - partition with
quorum
Last updated: Thu Feb  8 17:39:46 2018
Last change: Thu Feb  8 17:37:46 2018 by hacluster via crmd on nodeB



2 nodes configured
0 resources configured
```

```
Online: [ nodeA nodeB ]



No active resources




# Voilà! We have installed our basic cluster




# Raw cluster configuration can be shown by using following command:



[ONE] pcs cluster cib

<cib crm_feature_set="3.0.12" validate-with="pacemaker-2.8" epoch="5"
num_updates="7" admin_epoch="0" cib-last-written="Thu Feb  8 17:37:46
2018" update-origin="nodeB" update-client="crmd" update-



user="hacluster" have-quorum="1" dc-uuid="2">

  <configuration>

  <crm_config>
```

```
  <cluster_property_set id="cib-bootstrap-options">

  <nvpair id="cib-bootstrap-options-have-watchdog" name="have-
watchdog" value="false"/>

  <nvpair id="cib-bootstrap-options-dc-version" name="dc-version"
value="1.1.16-12.el7-94ff4df"/>

  <nvpair id="cib-bootstrap-options-cluster-infrastructure"
name="cluster-infrastructure" value="corosync"/>

  <nvpair id="cib-bootstrap-options-cluster-name" name="cluster-name"
value="lar_cluster"/>

  </cluster_property_set>

  </crm_config>

  <nodes>

  <node id="1" uname="nodeA"/>

  <node id="2" uname="nodeB"/>

  </nodes>

  <resources/>

  <constraints/>

  </configuration>

  <status>

  <node_state id="2" uname="nodeB" in_ccm="true" crmd="online" crm-
debug-origin="do_state_transition" join="member" expected="member">

  <lrm id="2">
```

```
  <lrm_resources/>

  </lrm>

  <transient_attributes id="2">

  <instance_attributes id="status-2">

  <nvpair id="status-2-shutdown" name="shutdown" value="0"/>

  </instance_attributes>

  </transient_attributes>

  </node_state>

  <node_state id="1" uname="nodeA" in_ccm="true" crmd="online" crm-
debug-origin="do_state_transition" join="member" expected="member">

  <lrm id="1">

  <lrm_resources/>

  </lrm>

  <transient_attributes id="1">

  <instance_attributes id="status-1">

  <nvpair id="status-1-shutdown" name="shutdown" value="0"/>

  </instance_attributes>

  </transient_attributes>

  </node_state>

  </status>
```

```
</cib>
```

```
# If ever we made some changes to the configuration manually, we can
check the correction

# of the XML file by running this command:
```

[ONE] **crm_verify -L -V**

```
  error: unpack_resources:  Resource start-up disabled since no
STONITH resources have been defined

  error: unpack_resources:  Either configure some or disable STONITH
with the stonith-enabled option

  error: unpack_resources:  NOTE: Clusters with shared data need
STONITH to ensure data integrity

Errors found during check: config not valid
```

```
# These errors will be ignored after disabling fencing.
```

```
# Cluster logs can be found
in /var/log/pacemaker.log and /var/log/cluster/corosync.log
```

root@nodeA:/root#> **cat /var/log/pacemaker.log**

Set r/w permissions for uid=189, gid=189 on /var/log/pacemaker.log

Feb 08 17:37:24 [5487] nodeA pacemakerd: info: crm_log_init: Changed active directory to /var/lib/pacemaker/cores

Feb 08 17:37:24 [5487] nodeA pacemakerd: info: get_cluster_type: Detected an active 'corosync' cluster

Feb 08 17:37:24 [5487] nodeA pacemakerd: info: mcp_read_config: Reading configure for stack: corosync

Feb 08 17:37:24 [5487] nodeA pacemakerd: notice: crm_add_logfile: Switching to /var/log/cluster/corosync.log

root@nodeA:/root#> **tail -20 /var/log/cluster/corosync.log**

Feb 08 17:37:46 [5488] nodeA cib: info: cib_perform_op: Diff: +++ 0.5.6 (null)

Feb 08 17:37:46 [5488] nodeA cib: info: cib_perform_op: + /cib: @num_updates=6

Feb 08 17:37:46 [5488] nodeA cib: info: cib_process_request: Completed cib_modify operation for

section status: OK (rc=0, origin=nodeB/attrd/3, version=0.5.6)

Feb 08 17:37:46 [5488] nodeA cib: info: cib_process_request: Forwarding cib_delete operation for section //node_state[@uname='nodeA']/transient_attributes to all (origin=local/crmd/13)

Feb 08 17:37:46 [5488] nodeA cib: info: cib_process_request: Completed cib_delete operation for section //node_state[@uname='nodeA']/transient_attributes: OK (rc=0, origin=nodeA/crmd/13, version=0.5.6)

Feb 08 17:37:46 [5488] nodeA cib: info: cib_perform_op: Diff: --- 0.5.6 2

Feb 08 17:37:46 [5488] nodeA cib: info: cib_perform_op: Diff: +++ 0.5.7 (null)

Feb 08 17:37:46 [5488] nodeA cib: info: cib_perform_op: + /cib: @num_updates=7

Feb 08 17:37:46 [5488] nodeA cib: info: cib_perform_op: ++ /cib/status/node_state[@id='1']: <transient_attributes id="1"/>

Feb 08 17:37:46 [5488] nodeA cib: info: cib_perform_op: ++ <instance_attributes id="status-1">

Feb 08 17:37:46 [5488] nodeA cib: info: cib_perform_op: ++ <nvpair id="status-1-shutdown" name="shutdown" value="0"/>

Feb 08 17:37:46 [5488] nodeA cib: info: cib_perform_op: ++ </instance_attributes>

Feb 08 17:37:46 [5488] nodeA cib: info: cib_perform_op: ++ </transient_attributes>

Feb 08 17:37:46 [5488] nodeA cib: info: cib_process_request: Completed cib_modify operation for section status: OK (rc=0, origin=nodeB/attrd/4, version=0.5.7)

Feb 08 17:37:46 [5488] nodeA cib: info: cib_file_backup: Archived previous version as /var/lib/pacemaker/cib/cib-3.raw

Feb 08 17:37:46 [5488] nodeA m01 cib: info: cib_file_write_with_digest: Wrote version 0.5.0 of the CIB to disk (digest: ef4905fd38cc2a926d5b6c686d3ab21e)

Feb 08 17:37:46 [5488] nodeA cib: info: cib_file_write_with_digest: Reading cluster configuration

file /var/lib/pacemaker/cib/cib.clWuxF (digest: /var/lib/pacemaker/cib/cib.gJSY1F)

Feb 08 17:37:51 [5488] nodeA cib: info: cib_process_ping: Reporting our current digest to nodeB: 2cdda87849aa421905eb3901c98cb8c1 for 0.5.7 (0x563aecfc0970 0)

Feb 08 17:37:55 [5493] nodeA crmd: info: crm_procfs_pid_of: Found cib active as process 5488

Feb 08 17:37:55 [5493] nodeA crmd: info: throttle_send_command: New throttle mode: 0000 (was ffffffff)