RHCS6: Install a two-node basic cluster

RHCS: Install a two-node basic cluster

```
# Tested on RHEL 6


# Red Hat Cluster is quite complex as to explain every and all
functionalities in a simple
# recipe like this. There are many considerations that should be
taken into account as
# network interfaces to use, fence type (depending on hw), etc. I
won't spend much time to
# explain all these options and functionalities. There are lots of
documentation about the
# subject. Do not hesitate to go and search on Red Hat official
documentation or any other
# web site in order to configure more complex clusters

# Main components of the Red Hat Cluster
#
# rgmanager: handles management of user-defined cluster services
(resource groups) upon


#   user request or in the event of failures.
#
# ricci: cluster management and configuration daemon. It dispatches
incoming messages to
#        underlying management modules.
#
# ccs: allows an administrator to create, modify and view a cluster
```

configuration file.
Using ccs an administrator can also start and stop the cluster services on one or
all of the nodes in a configured cluster.
#
cman: kernel-based cluster manager. It handles membership, messaging, quorum, event
notification and transitions.

Let's name my servers "**nodeA**" and "**nodeB**".

Note: <u>"ccs" commands are run only on one cluster node</u> (I"ll execute them on "nodeA").
All the rest must be executed on each node forming the cluster

As recommended by Red Hat, in order to power off immediately server via the fencing
device, instead of doing a clean shutdown, 'acpi' should be disabled on all nodes

service acpid stop
chkconfig --del acpid

Also, we must ensure that all nodes in the cluster have exactly the same time. Apart
from basic ntp options, I like to add following configuration:

```
echo "UTC=true" >> /etc/sysconfig/clock
sed -i.bak 's/OPTIONS="/OPTIONS="-x /' /etc/sysconfig/ntpd
sed -i.bak 's/SYNC_HWCLOCK=no/SYNC_HWCLOCK=yes/' /etc/sysconfig/ntpdate
```

```
# We have to know that the use of NetworkManager is not compatible
with cluster
# operations, so better disable or remove it, and that when using
bonding devices
# for intra-cluster connections, only active-backup mode is
supported.

# Apart from that, we have to take into account that the following
ports must be opened
# on the private network:
#
#     5404/UDP, 5405/UDP: cman
#     11111/TCP: ricci
#     21064/TCP: dlm (Distributed Lock Manager)
#     16861/tcp: modclusterd
#
# For practical reasons, I will fully disable systems' firewall as
well as SELinux,
# even if the use of SELinux in 'enforcing' mode is fully supported
when using the
# 'targeted' policy (These actions should never be performed on
servers that will
# be exposed to the outside world):

chkconfig iptables off
service iptables stop

sed -i.bak "s/SELINUX=enforcing/SELINUX=disabled/g"
/etc/selinux/config

shutdown -r now




# First of all we install the needed packages for the cluster layer
(depending on
```

```
# cluster type):


yum install ricci cman rgmanager ccs



# Then we start the ricci daemon, necessary in each cluster node for
the cluster to be able
# to propagate updated cluster configuration. This synchronization
can be done via the
# "cman_tool version -r", the "ccs" command or the "luci" user
interface server

service ricci start

# Let's set a password for "ricci" user

echo "ricci:myriccipasswd" | chpasswd  # or # echo "myriccipasswd" |
passwd --stdin ricci




# Create a basic cluster configuration. We have to provide a cluster
name, a multicast IP
# and the number of expected votes. Usually the number of expected
votes would match the
# number of nodes forming the cluster (+1 if quorum disk added);
nevertheless for a two-node
# cluster we"ll set "expected_votes" to "1" as we want the cluster to
keep on running
# in the eventuality of a node's failure.
# Note: Private network must support multicast and IGMP; if network
equipment do not
# support multicast and IGMP we can use UDP unicast communications by
adding following
# directive:
#       <cman transport="udpu"/>


ccs -f /etc/cluster/cluster.conf --createcluster mycluster
```

```
ccs -f /etc/cluster/cluster.conf --setmulticast 239.192.0.111
ccs -f /etc/cluster/cluster.conf --setcman expected_votes="1"
two_node="1"



# At any moment, we can check the configuration made so far by
running following command
# (configuration is stored in /etc/cluster/cluster.conf):

ccs -f /etc/cluster/cluster.conf --getconf



#  I add my nodes to the cluster

ccs -f /etc/cluster/cluster.conf --addnode nodeA --nodeid 1 --votes 1
ccs -f /etc/cluster/cluster.conf --addnode nodeB --nodeid 2 --votes 1



# We spread the configuration to the rest of nodes forming the
cluster.
# Do not forget to add the IPs used for cluster communications to
/etc/hosts

ccs -h nodeA -p myriccipasswd --sync --activate



# and start "cman" deamon, needed for the cluster to run. cman is a
distributed cluster
# manager and runs in each cluster node; cluster management is
distributed across all
# nodes in the cluster. It keeps track of membership by monitoring
messages from other
# cluster nodes.

service cman start
```

```
chkconfig cman on
chkconfig ricci on




# Voilà! We have installed our basic cluster

ccs -h nodeA -p myriccipasswd --getconf

<cluster config_version="1" name="mycluster">
<clusternodes>
<clusternode name="nodeA" nodeid="1" votes="1"/>
<clusternode name="nodeB" nodeid="2" votes="1"/>
</clusternodes>
<cman expected_votes="1" two_node="1">
<multicast addr="239.192.0.111"/>
</cman>
<rm/>
</cluster>



# To run a basic check of our new cluster we can use following
commands:

clustat
Cluster Status for mycluster @ Wed Jul 30 15:22:40 2014
Member Status: Quorate

Member Name                                                  ID
Status
------ ----                                                  ----
------
nodeA                                                         1
Online, Local
nodeB                                                         2
Online
```

```
cman_tool status
Version: 6.2.0
Config Version: 1
Cluster Name: mycluster
Cluster Id: 65461
Cluster Member: Yes
Cluster Generation: 68
Membership state: Cluster-Member
Nodes: 2
Expected votes: 1
Total votes: 2
Node votes: 1
Quorum: 1
Active subsystems: 8
Flags: 2node
Ports Bound: 0
Node name: nodeA
Node ID: 1
Multicast addresses: 239.192.0.111
Node addresses: 192.168.54.102
```

\# Cluster logs can be found in **/var/log/messages** and under **/var/log/cluster**

```
root@nodeA:/root#> ll /var/log/cluster
total 20
-rw-r--r--. 1 root root  531 Jul 30 12:19 dlm_controld.log
-rw-r--r--. 1 root root  423 Jul 30 12:19 fenced.log
-rw-r--r--. 1 root root  531 Jul 30 12:19 gfs_controld.log
```

\# For the higher level of logging, we can add **<rm log_level="7"/>** directive to our cluster configuration

Posted - Sun, Jun 3, 2018 9:33 AM. This article has been viewed 3126 times.

Online URL: http://kb.ictbanking.net/article.php?id=195