

RHCS: Configure an active/backup pacemaker cluster

Article Number: 193 | Rating: Unrated | Last Updated: Sun, Jun 3, 2018 9:28 AM

RHCS: Configure an active/backup pacemaker cluster

```
# Tested on CentOS 7
```

```
# 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.
```

```
# This recipe focuses on the shared storage case where two nodes have access to the shared
```

```
# volume but only one node can actively mount and write to it at a time.
```

```
# I'll work on a two-node basic pacemaker cluster like the one we  
built on:
```

```
https://sites.google.com/site/syscookbook/rhel/RHCS/pacemaker/rhcs-  
cluster-pacemaker-install
```

```
# I'll create a simple active/passive cluster. For that I will add a  
simple resource like
```

```
# an IP, that will allow me to connect to either of the nodes where  
IP will be active, a
```

```
# web service serving pages on the previously configured IP and a  
filesystem where will
```

```
# reside the web pages being served.
```

```
# Fencing
```

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

```
# There are many different topologies for fencing as to explain each  
of them in this recipe
```

```
# so, for the moment, I will disable it:
```

```
[ONE] pcs property set stonith-enabled=false
```

```
# Clustered Volume
```

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

```
# The first step is to configure a volume that will be able to  
failover between nodes. For
```

```
# that, we have to install the extensions to LVM2 to support  
clusters:
```

```
[ALL] yum install lvm2-cluster
```

```
# Then, create the PV and VG for the clustered filesystem:
```

```
[ONE] pvcreate /dev/sdb
```

```
[ONE] vgcreate vg_shared /dev/sdb
```

```
[ONE] lvcreate -l 100%VG -n lv_shared vg_shared
```

```
# Deactivate and reactivate shared volume group with an exclusive lock
```

```
[ONE] vgchange -an vg_shared
```

```
[ONE] vgchange -aey vg_shared
```

```
# And format the new logical volume
```

```
[ONE] mkfs.ext4 /dev/vg_shared/lv_shared
```

```
# Now, we have to update the /etc/lvm/lvm.conf configuration file so  
the CLVM volume is
```

```
# never auto-mounted. For that we will limit the volume list to  
be auto-mounted to 'rootvg'
```

```
# (this is the name of my rootvg) and the volume groups tagged with the hostname  
(cluster
```

```
# will take care of tagging volume groups at activation time):
```

```
[ALL] LINE=$(grep -n "# volume_list" /etc/lvm/lvm.conf | cut -f1  
-d':'`+1))
```

```
[ALL] sed -i.bak "${LINE}i volume_list = [ "rootvg", "@`hostname  
-s`" ]"
```

```
/etc/lvm/lvm.conf
```

```
# Update initramfs and reboot
```

```
[ALL] cp /boot/initramfs-$(uname -r).img /boot/initramfs-$(uname -r).img.bak
```

```
[ALL] dracut -H -f /boot/initramfs-$(uname -r).img $(uname -r)
```

```
[ALL] shutdown -r now
```

```
# Now we have a logical volume that is accessible from both nodes but inactive for both
```

```
# of them
```

```
root@nodeA:/root#> lvs
```

```
LV VG Attr LSize Pool Origin Data% Meta% Move Log [...]
```

```
lv_depot rootvg -wi-ao---- 5.00g
```

```
lv_root   rootvg   -wi-ao----   2.00g
lv_swap   rootvg   -wi-ao----   2.00g
lv_var    rootvg   -wi-ao----   2.00g
lv_shared vg_shared -wi----- 1020.00m
```

```
root@nodeB:/root#> lvs
```

```
LV VG Attr LSize Pool Origin Data% Meta% Move Log [...]
lv_depot rootvg -wi-ao---- 5.00g
lv_root   rootvg -wi-ao---- 2.00g
lv_swap   rootvg -wi-ao---- 2.00g
lv_var    rootvg -wi-ao---- 2.00g
lv_shared vg_shared -wi----- 1020.00m
```

```
# We are ready to create the LVM resource ("lar_cl-lv01" is the name
of the new resource
```

```
# and "lar_cl-rg01" the name of the new Resource Group cluster)
```

```
[ONE] pcs resource create lar_cl-lv01 ocf:heartbeat:LVM  
volgrpname=vg_shared
```

```
exclusive=true --group lar_cl-rg01
```

```
# Look at this, resource has been started on one of the nodes of the  
cluster:
```

```
[ONE] pcs status | grep lar_cl-lv01
```

```
lar_cl-lv01 (ocf::heartbeat:LVM): Started nodeA
```

```
# and, therefore, logical volume should be active on that node. Let's  
check it:
```

```
root@nodeA:/root#> lvs
```

```
LV VG Attr LSize Pool Origin Data% Meta% Move Log [...]  
  
lv_depot rootvg -wi-ao---- 5.00g  
  
lv_root rootvg -wi-ao---- 2.00g  
  
lv_swap rootvg -wi-ao---- 2.00g  
  
lv_var rootvg -wi-ao---- 2.00g  
  
lv_shared vg_shared -wi-a----- 1020.00m
```

```
root@nodeB:/root#> lvs
```

```
LV VG Attr LSize Pool Origin Data% Meta% Move Log [...]  
  
lv_depot rootvg -wi-ao---- 5.00g  
  
lv_root rootvg -wi-ao---- 2.00g  
  
lv_swap rootvg -wi-ao---- 2.00g  
  
lv_var rootvg -wi-ao---- 2.00g  
  
lv_shared vg_shared -wi----- 1020.00m
```

```
# Well, so far so good.
```

```
# Let's do the same thing for the filesystem resource (pay attention
to the Resource Group
```

```
# name, it must be the the one used before for the LVM resource):
```

```
[ONE] pcs resource create lar_cl-fs01 ocf:heartbeat:Filesystem
```

```
    device="/dev/vg_shared/lv_shared" directory="/cluster_fs_01"
fstype="ext4"
```

```
    --group lar_cl-rg01
```

```
[ONE] pcs status | grep "lar_cl-"
```

```
Resource Group: lar_cl-rg01
```

```
lar_cl-lv01 (ocf::heartbeat:LVM): Started nodeA
```

```
lar_cl-fs01 (ocf:heartbeat:Filesystem): Started nodeA <---
```

```
# Filesystem has been mounted on the right node (obvious):
```

```
root@nodeA:/root#> df -h | grep shared
```

```
/dev/mapper/vg_shared-lv_shared 988M 2.6M 919M 1% /cluster_fs_01
```

```
root@nodeB:/root#> df -h | grep shared
```

```
root@nodeB:/root#>
```

```
# Let's check if manual failover works fine:
```

```
[ONE] pcs status | grep "lar_cl-"
```

```
Resource Group: lar_cl-rg01
```

```
lar_cl-lv01 (ocf::heartbeat:LVM): Started nodeA
```

```
lar_cl-fs01 (ocf::heartbeat:Filesystem): Started nodeA
```

```
[ONE] pcs resource move lar_cl-fs01 nodeB
```

```
[ONE] pcs status | grep "lar_cl-"
```

```
Resource Group: lar_cl-rg01
```

```
lar_cl-lv01 (ocf::heartbeat:LVM): Started nodeB
```

```
lar_cl-fs01 (ocf::heartbeat:Filesystem): Started nodeB
```

```
root@nodeA:/root#> df -h | grep shared
```

```
root@nodeA:/root#>
```

```
root@nodeB:/root#> df -h | grep shared
```

```
/dev/mapper/vg_shared-lv_shared 988M 2.6M 919M 1% /cluster_fs_01
```

```
# And now, to check automatic failover, I will forcefully poweroff  
nodeB:
```

```
root@nodeB:/root#> echo 'o' > /proc/sysrq-trigger
```

```
# we can see that Resource Group failed over back to nodeA:
```

```
root@nodeA:/root#> pcs status | grep "lar_cl-"
```

```
Resource Group: lar_cl-rg01
```

```
lar_cl-lv01 (ocf::heartbeat:LVM): Started nodeA
```

```
lar_cl-fs01 (ocf::heartbeat:Filesystem): Started nodeA
```

```
root@nodeA:/root#> df -h | grep lv_shared
```

```
/dev/mapper/vg_shared-lv_shared 988M 2.6M 919M 1% /cluster_fs_01
```

```
# Everything worked as expected ...the only thing is that as soon as  
powered-off node
```

```
# becomes available again, Resource Group will switch back to it. In  
order to minimize the
```

```
# unavailability of services we may not want Resource Group switching  
back automatically. I
```

```
# will take care of this behaviour on a different recipe.
```

```
# Let's continue.
```

```
# Now I'm creating an IP resource, for our future clustered web server:
```

```
[ONE] pcs resource create lar_cl-ip01 ocf:heartbeat:IPaddr2  
ip=192.168.56.111
```

```
    cidr_netmask=24 op monitor interval=30s
```

```
# After a little while IP becomes available at cluster level:
```

```
[ONE] pcs status
```

```
Cluster name: lar_cluster
```

```
Stack: corosync
```

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

Last updated: Tue Feb 13 16:57:16 2018

Last change: Tue Feb 13 16:57:02 2018 by root via cibadmin on nodeA

2 nodes configured

3 resources configured

Online: [nodeA nodeB]

Full list of resources:

Resource Group: lar_cl-rg01

lar_cl-lv01 (ocf::heartbeat:LVM): Started nodeB

lar_cl-fs01 (ocf::heartbeat:Filesystem): Started nodeB

lar_cl-ip01 (ocf::heartbeat:IPaddr2): Started nodeA <---

Daemon Status:

corosync: active/enabled

pacemaker: active/enabled

pcsd: active/enabled

And is mounted

root@pacem01:/root#> **grep lar_cl-ip01 /etc/hosts**

192.168.56.111 lar_cl-ip01

root@nodeA:/root#> **ip a s eth0**

2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast
state UP qlen 1000

link/ether 08:00:27:5f:e2:2c brd ff:ff:ff:ff:ff:ff

inet 192.168.56.101/24 brd 192.168.56.255 scope global eth0

valid_lft forever preferred_lft forever

inet **192.168.56.111**/24 brd 192.168.56.255 scope global secondary
eth0

```
valid_lft forever preferred_lft forever
```

```
inet6 fe80::a00:27ff:fe5f:e22c/64 scope link
```

```
valid_lft forever preferred_lft forever
```

```
# What we've done here is to define a resource of type  
ocf:heartbeat:IPaddr2 this is:
```

```
# ocf: is the standard to which the resource script conforms and  
where to find it.
```

```
# heartbeat: is standard-specific; for OCF resources, it tells the  
cluster which OCF
```

```
# namespace the resource script is in.
```

```
# IPaddr2: is the name of the resource script.
```

```
# To obtain a list of the available resource standards, OCF providers  
and resource agents,
```

```
# run:
```

[ONE] pcs resource standards

ocf

lsb

service

systemd

stonith

[ONE] pcs resource providers

heartbeat

openstack

pacemaker

[ONE] pcs resource agents ocf:heartbeat

CTDB

Delay

Dummy

Filesystem

IPaddr

```
IPaddr2
```

```
[...]
```

```
rsyncd
```

```
slapd
```

```
symlink
```

```
tomcat
```

```
# The only thing is that IP resource doesn't belong to the  
LVM/filesystem Resource Group
```

```
# (I forgot to indicate "--group lar_cl-rg01" when creating the  
resource and we need
```

```
# everything to work together, so I will move IP resource to the  
right Resource Group by
```

```
# running following command:
```

```
[ONE] pcs resource group add lar_cl-rg01 lar_cl-ip01
```

```
# In addition, this will stop/start IP resource in the right node if
necessary:
```

```
[ONE] pcs status
```

```
[...]
```

```
Full list of resources:
```

```
Resource Group: lar_cl-rg01
```

```
lar_cl-lv01 (ocf::heartbeat:LVM): Started nodeB
```

```
lar_cl-fs01 (ocf::heartbeat:Filesystem): Started nodeB
```

```
lar_cl-ip01 (ocf::heartbeat:IPaddr2): Started nodeB <---
```

```
[...]
```

```
# Ok, we have a Resource Group with a filesystem and an IP. Let's
create an associated
```

```
# service on the top of that. I will add an Apache HTTP Server as a
service.
```

```
[ALL] yum install httpd wget
```

```
# Create a page to serve. As long as the site must be available on a
service that may
```

```
# failover between nodes, we have to create the file on the clustered
filesystem we
```

```
# created previously. Then, on the right node (cluster filesystem
must be mounted):
```

```
[ONE] mkdir -p /cluster_fs_01/www/html
```

```
[ONE] cat << EOF >/cluster_fs_01/www/html/index.html
```

```
<html>
```

```
<body>My clustered Apache service</body>
```

```
</html>
```

```
EOF
```

```
# And as the default Apache document root is /var/www/html:
```

```
[ALL] ln -s /cluster_fs_01/www/html/index.html  
/var/www/html/index.html
```

```
# Then, in order to monitor the health of my Apache instance, and  
recover it if it fails,
```

```
# the resource agent used by Pacemaker assumes the server-status URL  
is available:
```

```
[ALL] cat << EOF >/etc/httpd/conf.d/status.conf
```

```
<Location /server-status>
```

```
    SetHandler server-status
```

```
    Require local
```

```
</Location>
```

```
EOF
```

```
# Finally, I create the resource for my web site in the right  
Resource Group
```

```
[ONE] pcs resource create lar_cl-wb01 ocf:heartbeat:apache
```

```
configfile=/etc/httpd/conf/httpd.conf
```

```
statusurl="http://localhost/server-status"
```

```
Op monitor interval=1min --group lar_cl-rg01
```

```
[ONE] pcs status | grep "lar_cl-"
```

```
Resource Group: lar_cl-rg01
```

```
lar_cl-lv01 (ocf::heartbeat:LVM): Started nodeB
```

```
lar_cl-fs01 (ocf::heartbeat:Filesystem): Started nodeB
```

```
lar_cl-ip01 (ocf::heartbeat:IPaddr2): Started nodeB
```

```
lar_cl-wb01 (ocf::heartbeat:apache): Started nodeB <---
```

```
# Who is automatically started on the right node:
```

```
root@nodeA:/root#> ps -ef | grep -v grep | grep httpd
```

```
root@nodeA:/root#>
```

```
root@nodeB:/root#> ps -ef | grep -v grep | grep httpd
```

```
root  20504  1  0 17:36 ?  00:00:00 /sbin/httpd -DSTATUS -f
/etc/httpd/conf/httpd[...]
```

```
apache  20505 20504  0 17:36 ?  00:00:00 /sbin/httpd -DSTATUS -f
/etc/httpd/conf/httpd[...]
```

```
apache  20506 20504  0 17:36 ?  00:00:00 /sbin/httpd -DSTATUS -f
/etc/httpd/conf/httpd[...]
```

```
apache  20507 20504  0 17:36 ?  00:00:00 /sbin/httpd -DSTATUS -f
/etc/httpd/conf/httpd[...]
```

```
apache  20508 20504  0 17:36 ?  00:00:00 /sbin/httpd -DSTATUS -f
/etc/httpd/conf/httpd[...]
```

```
apache  20509 20504  0 17:36 ?  00:00:00 /sbin/httpd -DSTATUS -f
/etc/httpd/conf/httpd[...]
```

[ONE] **pcs status**

Cluster name: lar_cluster

Stack: corosync

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

Last updated: Tue Feb 13 17:40:34 2018

Last change: Tue Feb 13 17:36:24 2018 by root via cibadmin on nodeA

2 nodes configured

4 resources configured

Online: [nodeA nodeB]

Full list of resources:

Resource Group: lar_cl-rg01

lar_cl-lv01 (ocf::heartbeat:LVM): Started nodeB

```
lar_cl-fs01 (ocf::heartbeat:Filesystem): Started nodeB  
lar_cl-ip01 (ocf::heartbeat:IPaddr2): Started nodeB  
lar_cl-wb01 (ocf::heartbeat:apache): Started nodeB
```

Daemon Status:

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

```
# A test on the URL shows that our active/passive cluster configuration is ok and  
that the
```

```
# service is accesible:
```

```
[ONE] wget --spider http://192.168.56.111/index.html
```

```
Spider mode enabled. Check if remote file exists.
```

```
--2018-02-14 15:31:02-- http://192.168.56.111/index.html
```

```
Connecting to 192.168.56.111:80... connected.
```

```
HTTP request sent, awaiting response... 200 OK
```

```
Length: 62 [text/html]
```

```
Remote file exists and could contain further links,  
but recursion is disabled -- not retrieving.
```

```
# so it is after switching the service to the other node (first I had  
to clear some
```

```
# remaining constraints - we will see that in a different recipe)
```

```
[ONE] pcs resource move lar_cl-rg01 nodeA
```

```
[ONE] pcs status
```

```
Cluster name: lar_cluster
```

```
Stack: corosync
```

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

```
Last updated: Wed Feb 14 16:36:42 2018
```

Last change: Wed Feb 14 16:36:33 2018 by root via crm_resource on nodeA

2 nodes configured

4 resources configured

Online: [nodeA nodeB]

Full list of resources:

Resource Group: lar_cl-rg01

lar_cl-lv01 (ocf::heartbeat:LVM): Started nodeA

lar_cl-fs01 (ocf::heartbeat:Filesystem): Started nodeA

lar_cl-ip01 (ocf::heartbeat:IPaddr2): Started nodeA

lar_cl-wb01 (ocf::heartbeat:apache): Started nodeA

Daemon Status:

```
corosync: active/enabled
```

```
pacemaker: active/enabled
```

```
pcsd: active/enabled
```

```
[ONE] wget --spider http://192.168.56.111/index.html
```

```
Spider mode enabled. Check if remote file exists.
```

```
--2018-02-14 16:39:04-- http://192.168.56.111/index.html
```

```
Connecting to 192.168.56.111:80... connected.
```

```
HTTP request sent, awaiting response... 200 OK
```

```
Length: 62 [text/html]
```

```
Remote file exists and could contain further links,
```

```
but recursion is disabled -- not retrieving.
```

```
# My active/passive cluster is ready to work!
```

Posted - Sun, Jun 3, 2018 9:28 AM. This article has been viewed 10146 times.

Online URL: <http://kb.ictbanking.net/article.php?id=193>