**RedHat HA Setup for ODS – db2**

**(2 node cluster and quorum device)**

LBOM / Infrastructure Diagram Link --> [SHAREPOINT LINK](https://myfyi.sharepoint.com/:f:/r/teams/infrase/Shared%20Documents/2020/20203075?csf=1&web=1&e=CyLmvj)

Make sure all IP’s are entered into /etc/hosts.

**Node1**

10.136.164.254 pwauslodsdb0101.app.hcscint.net pwauslodsdb0101

192.168.84.2 pwauslodsdb0101cl.app.hcscint.net pwauslodsdb0101cl

10.215.24.122 pwauslodsdb0101-ilo.oob.hcscint.net pwauslodsdb0101-ilo

**Node2**

10.136.165.3 pwauslodsdb0102.app.hcscint.net pwauslodsdb0102

192.168.84.3 pwauslodsdb0101cl.app.hcscint.net pwauslodsdb0102cl

10.215.24.124 pwauslodsdb0102-ilo.oob.hcscint.net pwauslodsdb0102-ilo

**Quorum Device**

10.136.165.4 pwauslodsdb0103.app.hcscint.net pwauslodsdb0103

192.168.84.4 pwauslodsdb0103cl.app.hcscint.net pwauslodsdb0103cl

192.168.44.129 pwauslodsdb0103.oob.hcscint.net

**HADR Standby**

10.136.165.2 pwauslodsdb03.app.hcscint.net pwauslodsdb03

192.168.44.127 pwauslodsdb03.oob.hcscint.net

10.215.24.123 pwauslodsdb03-ilo.oob.hcscint.net pwauslodsdb03-ilo

**VIP**

10.136.165.25 pwauslodsdb01a.app.hcscint.net

When setting up physical servers with shared storage, configure all LUNs first before starting HA setup (see doc: ***Adding LUNs to ODS – db2 and multipath.conf Setup***). Then build out all Volume Groups and Logical Volumes that aren’t being managed by HA (in this case rootvg and appsvg).

Multiple server commands (each node)

**1. Install all required packages**

NOTE: The rhel-ha-for-rhel-7-server-rpms repo must be enabled to install pcs and the corosync packages.

Example:

subscription-manager repos --enable rhel-ha-for-rhel-7-server-rpms

\*\*\*\*\*\*Virtual Servers only\*\*\*\*\*\*\*

yum install -y pcs fence-agents-vmware-rest corosync corosync-qdevice

\*\*\*\*\*\*Physical HPE Servers only\*\*\*\*\*\*\*

yum install -y pcs fence-agents-ilo-ssh.x86\_64 fence-agents-mpath.x86\_64 corosync corosync-qdevice

**2. Set HA clustering password**

passwd hacluster

$L0wLyW0rk1ng1!

**3. Start and enable the pcs server**

systemctl start pcsd.service; systemctl enable pcsd.service

Single server commands (one node)

**4. Authorize each client in the entire cluster (this includes Quorum device)**

pcs cluster auth pwauslodsdb0101cl pwauslodsdb0102cl pwauslodsdb0103cl

**5. Create the pacemaker cluster (Quorum device not included)**

pcs cluster setup --start --name pwauslodsdb\_cluster pwauslodsdb0101cl pwauslodsdb0102cl

**6. Enable the pacemaker cluster and check status**

pcs cluster enable --all; pcs cluster status

**7. Set up mpath fencing**

pcs stonith create pwauslodsdb\_mpath fence\_mpath pcmk\_host\_map="pwauslodsdb0101cl:1; pwauslodsdb0102cl:2" pcmk\_host\_argument="key" pcmk\_monitor\_action="metadata" pcmk\_reboot\_action="off" meta provides=unfencing

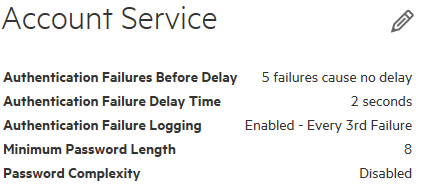
**8. Set up ilo fencing**

pcs stonith create pwauslodsdb0101\_ilo fence\_ilo5\_ssh ipaddr="pwauslodsdb0101-ilo" login="ADMIN" passwd="B4YOUGO2" pcmk\_host\_list="pwauslodsdb0101cl" secure="true" op monitor interval=120s on-fail=restart timeout=120s

pcs stonith create pwauslodsdb0102\_ilo fence\_ilo5\_ssh ipaddr="pwauslodsdb0102-ilo" login="ADMIN" passwd="B4YOUGO2" pcmk\_host\_list="pwauslodsdb0102cl" secure="true" op monitor interval=120s on-fail=restart timeout=120s

**IMPORTANT**

In order for ilo5\_ssh fencing to function properly, you need to modify settings inside of the iLO web interface. Inside “Security” => “Access Settings” => “Account Service”. Make sure all parameters are modified to match the below settings:



FYI: If these settings are not modified, you will receive fencing failure alerts through ‘pcs status’ and the fencing agent will stop.

Note: You can now test to see if fencing is working properly.

halt pwauslodsdb0102

pcs stonith confirm pwauslodsdb\_mpath

pcs stonith confirm pwauslodsdb0102\_ilo

fence\_ilo5\_ssh -a pwauslodsdb0102-ilo -x -l 'ADMIN' -p 'B4YOUGO2' -o reboot

Note: To modify Fencing (this command must be run before modifying an existing stonith resource)

pcs property set stonith-enabled=false

pcs property set stonith-enabled=true (default)

**9. Set up vmware fencing (virtual servers only)**

*Test that you can communicate with the Hypervisor:*

fence\_vmware\_rest -a dchivavmvc01.adhcscdev.net -l 'rh\_fence\_usr@vsphere.local' -p 'F3nc3U$rR0ck$' --ssl-insecure -o list | egrep "(dwauslhacmppoc010|dwauslhacmppoc011|dwauslhacmppoc012)"

\*\*NOTE: stopped here due to firewall issue, REQ0461570-RITM0511651 to open port 443,80 from Nodes to Vcenter server (i.e. prod/dev/test). Also verify NPID for connection.

pcs stonith create dwauslhacmppoc\_vmware fence\_vmware\_rest pcmk\_host\_map="dwauslhacmppoc010cl:dwauslhacmppoc010;dwauslhacmppoc011cl: dwauslhacmppoc011:dwauslhacmppoc012cl:dwauslhacmppoc012" ipaddr=dchivavmvc01.adhcscdev.net ssl\_insecure=1 login='rh\_fence\_usr@vsphere.local' passwd='F3nc3U$rR0ck$'

\*\*\*\*\*\*To test Fencing\*\*\*\*\*\*\*

halt dwauslhacmppoc011

pcs stonith confirm dwauslhacmppoc\_mpath

pcs stonith confirm dwauslhacmppoc011\_ilo

fence\_ilo5\_ssh -a dwauslhacmppoc011-ilo -x -l 'ADMIN' -p 'B4YOUGO2' -o reboot

\*\*\*\*\*\*To modify Fencing \*\*\*\*\*\*\*

pcs property set stonith-enabled=false

pcs property set stonith-enabled=true (default)

**10. Set up fence (execution) ordering**

pcs stonith level add 2 pwauslodsdb0101cl pwauslodsdb\_mpath; pcs stonith level add 2 pwauslodsdb0102cl pwauslodsdb\_mpath

pcs stonith level add 1 pwauslodsdb0101cl pwauslodsdb0101\_ilo; pcs stonith level add 1 pwauslodsdb0102cl pwauslodsdb0102\_ilo

**11. Restart the cluster**

pcs cluster stop --all; pcs cluster start --all

Quorum Device commands

**12. Install the required Quorum device package**

yum install -y corosync-qnetd.x86\_64

**13. Enable the Quorum device and check status**

pcs qdevice setup model net --enable --start; pcs qdevice status net --full

Single server commands

**14. Adding Quorum function to the cluster and check status**

pcs quorum device add model net host=pwauslodsdb0103cl algorithm=ffsplit; pcs quorum status

**15. Show Quorum and device status**

pcs quorum config; pcs quorum device status

Quorum Device commands

**16. Show status after Quorum device is added to the cluster**

pcs qdevice status net --full

Multiple server commands (each node)

**17. Unmount any (local) NFS mountpoints**

umount /data-mirror/p1

**18.** **If the datavg volume group was accidently created, it will need to be deactivated.**

vgchange -an datavg

**19. Deactivate LVM management**

lvmconf --enable-halvm --services --startstopservices

**20. Exclude** **datavg volume group from LVM management**

vim /etc/lvm/lvm.conf

insert volume\_list = [ "rootvg", "appsvg" ]

**21. Create a backup initramFS and create HA-LVM-enabled initramFS**

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

**22. Reboot**

Single server commands (one node)

**23. Put second node on standby**

pcs cluster standby pwauslodsdb0102cl

**24. Create all Volume Groups and Logical Volumes that are going to be managed by HA (in this case datavg).**

**NOTE: You will need to temporarily comment out** volume\_list = [ "rootvg", "appsvg" ] **inside** /etc/lvm/lvm.conf **in order to create datavg.**

**25. Create the HA-LVM resource inside the resource (db2) group**

pcs resource create pwauslodsdb\_lvm ocf:heartbeat:LVM-activate vgname=datavg activation\_mode=exclusive vg\_access\_mode=tagging tag=rhel7 --group db2\_group

**26. Create all db2 filesystem mounts inside the resource (db2) group**

Example: pcs resource create db2dba Filesystem device="/dev/datavg/db2dbalv" directory="/db2dba/p1" fstype="ext4" --group db2\_group

**27. Create floating VIP inside the resource (db2) group**

pcs resource create floating\_vip IPaddr2 ip=10.136.165.25 cidr\_netmask=22 --group db2\_group

**28. Add devices only managed by HA-LVM to the mpath fence resource**

Example: pcs stonith update pwauslodsdb\_mpath devices="/dev/mapper/datavg\_db2dba"

Note: All devices must be added at once - No whitespaces. No extra commas.

**29. Restart the cluster**

pcs cluster stop --all; pcs cluster start –all

**DB2 installation**

**------------------------------------------**

**30. Modify/verify floating VIP**

vim /db2home/p1/odsprdi1/sqllib/db2nodes.cfg

replace current hostname with floating VIP (pwauslodsdb01a)

**31. Create database instance resource inside the resource (db2) group**

pcs resource create odsprdi1\_db2 db2 instance=odsprdi1 dblist=odsprdda op start timeout=600s --group db2\_group

NOTE: Check instance script status for dblist name (script provided by db team)

su - odsprdi1 -c /db2dba/p1/master\_scripts/db2\_status.ksh

**32. Create the symlink resources inside the resource (db2) group**

pcs resource create cronsym\_odsprdi1 symlink target=/db2home/p1/odsprdi1/FAILOVER\_CRON/odsprfi1 link=/var/spool/cron/odsprdi1 backup\_suffix=".disabled" --group db2\_group

pcs resource create cronsym\_datadir1 symlink target=/data-mirror/p1/FAILOVER\_CRON/datadir1 link=/var/spool/cron/datadir1 backup\_suffix=".disabled" --group db2\_group

**33. Create the cron resource inside the resource (db2) group**

pcs resource create systemd\_db2 systemd:crond op monitor interval="10" --group db2\_group

**34. Create the NFS resource inside the resource (db2) group**

pcs resource create db2bkup\_nfs nfsserver nfs\_shared\_infodir=/db2bkup1/p1/nfsinfo nfs\_no\_notify=true op monitor on-fail=ignore --group db2\_group

NOTE: Only one NFS directory (/db2bkup1/p1) needs to be created to share any NFS filesystems.

**35. Create the NFS export resources inside the resource (db2) group**

pcs resource create db2bkup1\_nfs\_export exportfs clientspec="10.136.165.25" options=rw,sync,no\_root\_squash directory=/db2bkup1/p1 fsid=1 op monitor on-fail=ignore start on-fail=ignore --group db2\_group

pcs resource create db2bkup2\_nfs\_export exportfs clientspec="10.136.165.25" options=rw,sync,no\_root\_squash directory=/db2bkup2/p1 fsid=2 op monitor on-fail=ignore start on-fail=ignore --group db2\_group

pcs resource create db2bkup3\_nfs\_export exportfs clientspec="10.136.165.25" options=rw,sync,no\_root\_squash directory=/db2bkup3/p1 fsid=3 op monitor on-fail=ignore start on-fail=ignore --group db2\_group

pcs resource create db2bkup4\_nfs\_export exportfs clientspec="10.136.165.25" options=rw,sync,no\_root\_squash directory=/db2bkup4/p1 fsid=4 op monitor on-fail=ignore start on-fail=ignore --group db2\_groupq

pcs resource create db2bkup5\_nfs\_export exportfs clientspec="10.136.165.25" options=rw,sync,no\_root\_squash directory=/db2bkup5/p1 fsid=5 op monitor on-fail=ignore start on-fail=ignore --group db2\_group

**36. Add the mounts onto the HADR server**

On pwauslodsdb03:

vi /etc/fstab

pwauslodsdb01a.app.hcscint.net:/db2bkup1/p1 /db2bkup1/p1 nfs rw,sync,hard,\_netdev 0 0

pwauslodsdb01a.app.hcscint.net:/db2bkup2/p1 /db2bkup2/p1 nfs rw,sync,hard,\_netdev 0 0

pwauslodsdb01a.app.hcscint.net:/db2bkup3/p1 /db2bkup3/p1 nfs rw,sync,hard,\_netdev 0 0

pwauslodsdb01a.app.hcscint.net:/db2bkup4/p1 /db2bkup4/p1 nfs rw,sync,hard,\_netdev 0 0

pwauslodsdb01a.app.hcscint.net:/db2bkup5/p1 /db2bkup5/p1 nfs rw,sync,hard,\_netdev 0 0

**37. Create the NFS notify resource inside the resource (db2) group**

pcs resource create nfs\_notify nfsnotify source\_host=10.134.167.134 --group db2\_group

Multiple server commands (each node)

**38. Create the HA configuration backup directory and cron job**

mkdir -p /opt/ha/backups; crontab -e

### backup RedHat HA settings - run weekly

0 03 \* \* 6 /sbin/pcs config backup /opt/ha/backups/pcs\_$(date "+%F-%H-%M-%S")

**39. Create HA alert log and log rotate**

NOTE: location of alert scripts - /usr/share/pacemaker/alerts

cp /usr/share/pacemaker/alerts/alert\_file.sh.sample /usr/share/pacemaker/alerts/alert\_HCSC.sh.sample

install --mode=0755 /usr/share/pacemaker/alerts/alert\_HCSC.sh.sample /var/lib/pacemaker/alert\_HCSC.sh

vim /var/lib/pacemaker/alert\_HCSC.sh

if [ "${debug\_exec\_order}" = "true" ]; then

tstamp=`printf "%04d. " "$CRM\_alert\_node\_sequence"`

if [ ! -z "$CRM\_alert\_timestamp" ]; then

tstamp="${tstamp} $CRM\_alert\_timestamp (`date "+%b %d %T "`): "

fi

else

if [ ! -z "$CRM\_alert\_timestamp" ]; then

# tstamp="$CRM\_alert\_timestamp: "

tstamp=`date "+%b %d %T "`

fi

fi

touch /var/log/pcmk\_alert\_file.log; chown hacluster:haclient /var/log/pcmk\_alert\_file.log; chmod 600 /var/log/pcmk\_alert\_file.log

vim /etc/logrotate.d/pcmk

/var/log/pcmk\_alert\_file.log {

missingok

compress

copytruncate

monthly

rotate 12

notifempty

}

Single server commands (one node)

pcs alert create id=alert\_file description="Log events to a file." path=/var/lib/pacemaker/alert\_HCSC.sh; pcs alert recipient add alert\_file id=my-alert\_logfile value=/var/log/pcmk\_alert\_file.log

**40. Create PowerBroker scripts for db2 team (these scripts allow them to check status)**

mkdir -p /opt/ha/pb-scripts

vim pcs\_status.sh

pcs status

vim pcs\_resource\_show\_odsprfi1\_db2.sh

pcs resource show odsprfi1\_db2

vim systemctl\_restart\_cron.sh

systemctl restart cron.d

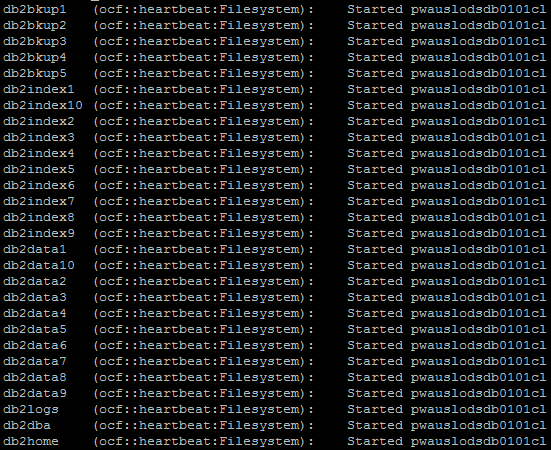
chmod 555 /opt/ha/pb-scripts/\*

**ODS HA Resource Group Overview (db2\_group)**

This resource controls lvm through HA-LVM for datavg (instead of local LVM management)



These resources are the shared filesystems managed by HA (instead of /etc/fstab)



This resource allows connectivity to the application regardless of location of the current cluster node.



This resource controls the start/stop of the application and monitors the instance and database.



This resource creates a symlink from a crontab file residing in /db2home/p1 and /data-mirror/p1/ and links it to /var/spool/cron/ of location of the current cluster node.



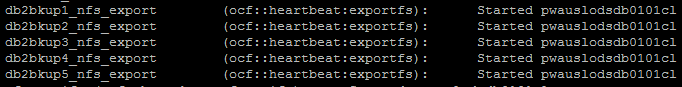
This resource restarts the cron service to reread the crontabs.



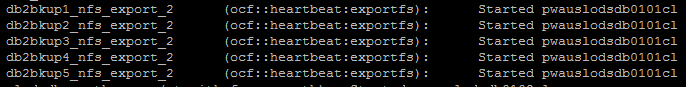
This resource sets up the NFS filesystems for export (instead of /etc/exports).



These resources export the NFS filesystems to pwauslodsdb03 (instead of /etc/exports).



These resources export the NFS filesystems to pftwslodsdb01 (instead of /etc/exports).



This resource notifies any NFS clients when the NFS service will be temporarily unavailable/available.



This resource reserves the LUNs (SAN) to the current cluster node location and can also prevent a node from keeping the reservation (fencing).



These resources control the power/reset of the physical servers in the cluster.

