**RedHat HA Setup**

**(3 node cluster)**

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***). When setting up virtual servers with shared storage, configure all vdisks first before starting HA setup. All server names in this documentation are used as an example only. The HA setup may different in the actual environments.

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**

\*\*\*\*\*\*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**

pcs cluster auth dwauslhacmppoc010cl dwauslhacmppoc011cl dwauslhacmppoc012cl

**5. Create the pacemaker cluster**

pcs cluster setup --start --name dwauslhacmppoc\_cluster dwauslhacmppoc010cl dwauslhacmppoc011cl dwauslhacmppoc012cl

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

pcs cluster enable --all; pcs cluster status

**7. Set up mpath fencing (physical servers only)**

pcs stonith create twauslodsdb\_mpath fence\_mpath pcmk\_host\_map=" dwauslhacmppoc010cl:1; dwauslhacmppoc011cl:2; dwauslhacmppoc012cl:3" pcmk\_host\_argument="key" pcmk\_monitor\_action="metadata" pcmk\_reboot\_action="off" meta provides=unfencing

**8. Set up ilo fencing (physical servers only)**

*Test that you can communicate with iLO:*

fence\_ilo5\_ssh -a dwauslhacmppoc010-ilo -x -l 'ADMIN' -p 'B4YOUGO2' -o status

pcs stonith create dwauslhacmppoc010\_ilo fence\_ilo5\_ssh ipaddr="dwauslhacmppoc010-ilo" login="ADMIN" passwd="B4YOUGO2" pcmk\_host\_list=”dwauslhacmppoc010cl” secure="true" op monitor interval=120s on-fail=restart timeout=120s

pcs stonith create dwauslhacmppoc011\_ilo fence\_ilo5\_ssh ipaddr="dwauslhacmppoc011-ilo" login="ADMIN" passwd="B4YOUGO2" pcmk\_host\_list=“dwauslhacmppoc011cl” secure="true" op monitor interval=120s on-fail=restart timeout=120s

pcs stonith create dwauslhacmppoc012\_ilo fence\_ilo5\_ssh ipaddr="dwauslhacmppoc012-ilo" login="ADMIN" passwd="B4YOUGO2" pcmk\_host\_list=”dwauslhacmppoc012cl” secure="true" op monitor interval=120s on-fail=restart timeout=120s

**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)"

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 (physical servers only)**

pcs stonith level add 2 dwauslhacmppoc010cl dwauslhacmppoc\_mpath; pcs stonith level add 2 dwauslhacmppoc011cl dwauslhacmppoc\_mpath; pcs stonith level add 2 dwauslhacmppoc012cl dwauslhacmppoc\_mpath

pcs stonith level add 1 dwauslhacmppoc010cl dwauslhacmppoc\_ilo; pcs stonith level add 1 dwauslhacmppoc011cl dwauslhacmppoc\_ilo; pcs stonith level add 1 dwauslhacmppoc012cl dwauslhacmppoc\_ilo

**11. Restart the cluster**

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

Multiple server commands (each node)

**12. Unmount any NFS mountpoints and “shared” volume group filesystems**

umount /data-mirror/p1

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

vgchange -an data{1,2,3}vg

**14. Deactivate LVM management**

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

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

vim /etc/lvm/lvm.conf

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

**16. Create a backup initramFS and create haLVM-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)

**17. Reboot**

Single server commands (one node)

**18. Put secondary nodes on standby**

pcs cluster standby dwauslhacmppoc011cl dwauslhacmppoc012cl

**19. Create the 3 HA-LVM resources inside 3 different resource groups**

pcs resource create dwauslhacmppoc\_lvm1 ocf:heartbeat:LVM-activate vgname=data1vg activation\_mode=exclusive vg\_access\_mode=tagging tag=rhel7 --group db2\_group1

pcs resource create dwauslhacmppoc\_lvm2 ocf:heartbeat:LVM-activate vgname=data2vg activation\_mode=exclusive vg\_access\_mode=tagging tag=rhel7 --group db2\_group2

pcs resource create dwauslhacmppoc\_lvm3 ocf:heartbeat:LVM-activate vgname=data3vg activation\_mode=exclusive vg\_access\_mode=tagging tag=rhel7 --group db2\_group3

**20. Create all Volume Groups and Logical Volumes that are going to be managed by HA (in this case data1vg, data2vg, data3vg).**

**20. Create all db2 filesystem mounts inside the resource (db2) groups**

pcs resource create app-instance1-logs Filesystem device="/dev/data1vg/app-instance1-logslv" directory="/app-instance1/logs" fstype="ext4" --group db2\_group1

pcs resource create app-instance1-backups Filesystem device="/dev/data1vg/app-instance1-backupslv" directory="/app-instance1/backups" fstype="ext4" --group db2\_group1

pcs resource create app-instance1-data Filesystem device="/dev/data1vg/app-instance1-datalv" directory="/app-instance1/data" fstype="ext4" --group db2\_group1

pcs resource create app-instance2-logs Filesystem device="/dev/data2vg/app-instance2-logslv" directory="/app-instance2/logs" fstype="ext4" --group db2\_group2

pcs resource create app-instance2-backups Filesystem device="/dev/data2vg/app-instance2-backupslv" directory="/app-instance2/backups" fstype="ext4" --group db2\_group2

pcs resource create app-instance2-data Filesystem device="/dev/data2vg/app-instance2-datalv" directory="/app-instance2/data" fstype="ext4" --group db2\_group2

pcs resource create app-instance3-logs Filesystem device="/dev/data3vg/app-instance3-logslv" directory="/app-instance3/logs" fstype="ext4" --group db2\_group3

pcs resource create app-instance3-backups Filesystem device="/dev/data3vg/app-instance3-backupslv" directory="/app-instance3/backups" fstype="ext4" --group db2\_group3

pcs resource create app-instance3-data Filesystem device="/dev/data3vg/app-instance3-datalv" directory="/app-instance3/data" fstype="ext4" --group db2\_group3

**21. Create three floating VIPs inside each resource (db2) group**

pcs resource create floating\_vip1 IPaddr2 ip=<ip addr> cidr\_netmask=22 --group db2\_group1

pcs resource create floating\_vip1 IPaddr2 ip=<ip addr> cidr\_netmask=22 --group db2\_group2

pcs resource create floating\_vip1 IPaddr2 ip=<ip addr> cidr\_netmask=22 --group db2\_group3

**22. Add devices only managed by HA-LVM to the mpath fence resource (physical servers only)**

Example: pcs stonith update dwauslhacmppoc\_mpath devices="/dev/mapper/data1vg-app--instance1--backupslv,/dev/mapper/data1vg-app--instance1--datalv,/dev/mapper/data1vg-app--instance1--logslv"

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

**23. Add constraints for resources**

pcs constraint location db2\_group1 prefers dwauslhacmppoc010cl

pcs constraint location db2\_group2 prefers dwauslhacmppoc011cl

pcs constraint location db2\_group3 prefers dwauslhacmppoc012cl

NOTE: constraints can be score=1 through INFINITY

FYI: pcs resource defaults resource-stickiness=1

pcs constraint list

pcs constraint show --full

**24. Restart the cluster**

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

Application installation

Additional resources will depend on the application needs (ex/ symlinks, cron jobs, NFS)

(see doc: ***RedHat HA Setup for ODS – db2 (2 node cluster and quorum device)*** for help with additional steps.

Multiple server commands (each node)

**25. 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")

**26. 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

**27. 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/\*