

NFS on ZFS HA Cluster

Build a high-available dual-controller storage array using open-source technologies. This solution should be capable of presenting shared storage to NFS client and can be expanded to iSCSI and FCoE.

What is needed:

- 2 nodes with A LOT of memory.
- Fast storage that is directly attached to each node.
- Root access.
- The ZFS filesystem configured via the [ZFS on Linux repository](#).
- Familiarity with basic ZFS operations: zpool/zfs filesystem creation and modification.
- Understanding of network under RHEL Linux.

These steps describe the construction of a two host, single JBOD cluster that manages a single ZFS pool.

On both nodes:

Get EPEL and ZFS repositories

```
yum install -y epel-release
yum install -y http://download.zfsonlinux.org/epel/zfs-release.el7_8.noarch.rpm
yum update -y
yum install -y kernel-devel zfs
systemctl preset zfs-import-cache zfs-import-scan zfs-mount zfs-share zfs-zed zfs.target
systemctl enable zfs-import-scan
systemctl start zfs-import-scan
```

Add some clustering requirements:

```
yum install -y pcs fence-agents-all device-mapper-multipath nfs-utils
touch /etc/multipath.conf
systemctl start multipathd
systemctl enable multipathd
```

Add each node to the /etc/hosts file.

```
echo "172.16.100.142 node1" >> /etc/hosts
echo "172.16.100.144 node2" >> /etc/hosts
```

Set the hacluster password:

```
passwd hacluster wljgnFAW4fgwEGF21
```

Add needed heartbeat files

```
cd /usr/lib/ocf/resource.d/heartbeat/
wget https://raw.githubusercontent.com/clusterapps/stmf-ha/master/heartbeat/ZFS
wget https://github.com/ClusterLabs/resource-agents/raw/master/heartbeat/iSCSITarget
wget https://github.com/ClusterLabs/resource-agents/raw/master/heartbeat/iSCSILogicalUnit
chmod a+x ./ZFS
chmod a+x ./iSCSILogicalUnit
chmod a+x ./iSCSITarget
```

Update the firewall configuration.

```
firewall-cmd --add-service=nfs --permanent
firewall-cmd --add-service=high-availability --permanent
firewall-cmd --permanent --add-service=nfs
firewall-cmd --permanent --add-service=mountd
firewall-cmd --permanent --add-service=rpc-bind
firewall-cmd --reload
```

Enable the services and reboot. This step helps verify all of the services are properly configured on boot.

```
systemctl enable pcsd
systemctl enable corosync
systemctl enable pacemaker
reboot
```

On the Primary

Create the pool. You will need the path to the devices. Get them from:

```
ls -l /dev/disk/by-id/
```

Run `zpool create` (Your devices will be different!)

This pool has similar qualities to a RAID5+0. You should build what you need.

```
zpool create array1 -o ashift=12 -o autoexpand=on -o autoreplace=on -o cachefile=none \  
raidz1 /dev/disk/by-id/scsi-35000c500740a6277 /dev/disk/by-id/scsi-35000c5007411d36b /dev/disk/by-id/scsi-35000c5007411cb1b \  
raidz1 /dev/disk/by-id/scsi-35000c50076703a9f /dev/disk/by-id/scsi-35000c50070d3a853 /dev/disk/by-id/scsi-35000c50076701f57 \  
raidz1 /dev/disk/by-id/scsi-35000c5007411cfa7 /dev/disk/by-id/scsi-35000c5007411d0bf /dev/disk/by-id/scsi-35000c500740a109f \  
log mirror /dev/disk/by-id/scsi-35000c5007670107f /dev/disk/by-id/scsi-35000c5007411d467 spare /dev/disk/by-id/scsi-35000c50076bd0c03
```

Update a few ZFS settings

```
zfs set acltype=posixacl array1  
zfs set atime=off array1  
zfs set xattr=sa array1  
zfs set compression=lz4 array1
```

Authorize Cluster

```
pcs cluster auth node1 node2  
pcs cluster setup --start --name NASOne node1 node2
```

Set some cluster properties and add the resources

```
pcs property set no-quorum-policy=ignore  
pcs stonith create fence-array1 fence_scsi pcmk_monitor_action="metadata" pcmk_host_list="node1 node2" \  
devices="/dev/mapper/35000c5007670107f,/dev/mapper/35000c5007411d467,/dev/mapper/35000c50076bd0c03" \  
meta provides=unfencing --group=group-array1  
pcs resource create array1-ip IPAddr2 ip=172.16.100.99 cidr_netmask=24 --group group-array1  
pcs resource create array1 ZFS pool="array1" importargs="-d /dev/mapper/" op start timeout="90" op stop timeout="90" --group=group-array1  
pcs resource defaults resource-stickiness=100
```

Create and share a ZFS directory.

```
zfs create array1/nfs1  
zfs set sharenfs=rw=@172.16.100.0/24,sync,no_root_squash,no_wdelay array1/nfs1
```

Enable and start NFS related services.

```
systemctl enable rpcbind nfs-server
```

```
systemctl start rpcbind nfs-server
```

Check the status of the cluster.

```
pcs cluster status
```

```
pcs status resources
```

```
showmount -e localhost
```

Revision #3

Created 17 March 2019 16:33:36 by Michael Cleary

Updated 16 February 2022 23:59:58 by Michael Cleary