

# Deployment of storage manual

## Overview

SUPFile is based on a **Highly Available NFS Storage with DRBD and Pacemaker** infrastructure.

This manual will help you to deploy the whole system on your own server.

- Two nodes: `alice` (IP: 192.168.1.1 ) and `bob` (IP: 192.168.1.2 ).
- Two floating, virtual IP addresses ( 192.168.1.10 and 192.168.2.1 ).One for cluster administration, the other for the NFS exports.
- A shared storage device, used as an SBD fencing mechanism.
- **Active/Passive Setup**
- Local storage on each host.
- A file system exported through NFS.

## Step-by-step

### Installing a Basic Two-Node Cluster

Install and set up a basic two-node cluster for further usage,using `ha-cluster-bootstrap` package.

### Creating an LVM Device

1. Create an LVM volume group and replace `/dev/sdb1` with your corresponding device for LVM:

```
root # pvcreate /dev/sdb1
```

2. Create an LVM Volume Group `nfs` that includes this physical volume:

```
root # vgcreate nfs /dev/sdb1
```

3. Create one or more logical volumes in the volume group `nfs` .

```
root # lvcreate -n work -L 20G nfs
```

4. Activate the volume group:

```
root # vgchange -ay nfs
```

## Creating a DRBD Device

### Creating DRBD Configuration

1. Create the file `/etc/drbd.d/nfs.res` .

```
resource nfs {
    device /dev/drbd0; 1
    disk   /dev/nfs/work; 2
    meta-disk internal; 3

    net {
        protocol C; 4
    }

    connection-mesh { 5
        hosts      alice bob;
    }
    on alice { 6
        address     192.168.1.1:7790;
        node-id     0;
    }
    on bob { 6
        address     192.168.1.2:7790;
        node-id     1;
    }
}
```

2. Open `/etc/csync2/csync2.cfg` and add these lines if they are not exist:

```
include /etc/drbd.conf;
include /etc/drbd.d/*.res;
```

3. Copy the file to the other nodes:

```
root # csync2 -xv
```

## Activating the DRBD Device

1. Open port `7790` in your firewall.

2. Run this on two nodes:

```
root # drbdadm create-md nfs
root # drbdadm up nfs
```

3. Make `alice` primary:

```
root # drbdadm primary --force nfs
```

4. Check the DRBD status:

```
root # drbdadm status nfs
```

## Creating the File System

```
root # mkfs.ext3 /dev/drbd0
```

## Adjusting Pacemaker's Configuration

```
root # crm configure
crm(live)configure# rsc_defaults resource-stickiness="200"
crm(live)configure# commit
```

## Creating Cluster Resources

### DRBD Primitive and Multi-state Resource

```
crm(live)# configure
crm(live)configure# primitive drbd_nfs \
  ocf:linbit:drbd \
    params drbd_resource="nfs" \
    op monitor interval="15" role="Master" \
    op monitor interval="30" role="Slave"
crm(live)configure# ms ms-drbd_nfs drbd_nfs \
  meta master-max="1" master-node-max="1" clone-max="2" \
  clone-node-max="1" notify="true"
crm(live)configure# commit
```

### NFS Kernel Server Resource

```
crm(live)configure# primitive nfsserver \  
    systemd:nfs-server \  
    op monitor interval="30s"  
crm(live)configure# clone cl-nfsserver nfsserver  
crm(live)configure# commit
```

## File System Resource

1. Configure file system type resource.

```
crm(live)configure# primitive fs_work \  
    ocf:heartbeat:Filesystem \  
    params device=/dev/drbd0 \  
    directory=/srv/nfs/work \  
    fstype=ext3 \  
    op monitor interval="10s"
```

2. Combine these resources into a Pacemaker resource *group*:

```
crm(live)configure# group g-nfs fs_work
```

3. Add the following constraints:

```
crm(live)configure# order o-drbd_before_nfs inf: \  
    ms-drbd_nfs:promote g-nfs:start  
crm(live)configure# colocation col-nfs_on_drbd inf: \  
    g-nfs ms-drbd_nfs:Master
```

4. Commit this configuration:

```
crm(live)configure# commit
```

## NFS Export Resources

1. Create NFS exports.

```
crm(live)configure# primitive exportfs_work \  
  ocf:heartbeat:exportfs \  
    params directory="/srv/nfs/work" \  
      options="rw,mountpoint" \  
      clientspec="192.168.2.0/24" \  
      wait_for_leasetime_on_stop=true \  
  op monitor interval="30s"
```

2. Append them to the existing `g-nfs` resource group:

```
crm(live)configure# modgroup g-nfs add exportfs_work
```

3. Commit:

```
crm(live)configure# commit
```

4. Confirm:

```
root # exportfs -v  
/srv/nfs/work    IP_ADDRESS_OF_CLIENT(OPTIONS)
```

## Virtual IP Address for NFS Exports

```
crm(live)configure# primitive vip_nfs IPAddr2 \  
  params ip=192.168.2.1 cidr_netmask=24 \  
  op monitor interval=10 timeout=20  
crm(live)configure# modgroup g-nfs add vip_nfs  
crm(live)configure# commit
```

## Using the NFS Service

Mount the NFS export:

```
root # mount -t nfs 192.168.2.1:/srv/nfs/work /home/work
```

