

# ???? ?????

- [centos6.x drbd](#)
- [Ceph](#)
- [DRBD](#)
- [Glusterfs rpm](#)
- [RHEL VDO](#)

# centos6.x drbd??

Centos6  DRBD rpm  

```
[root@localhost ~]# rpm -ivh http://elrepo.org/elrepo-release-6-5.el6.elrepo.noarch.rpm
[root@localhost ~]# yum install *drbd84* -y
[root@localhost ~]# modprobe drbd
[root@localhost ~]# lsmod | grep drbd
drbd 297925 0
libcrc32c 841 1 drbd
```

# Ceph ????? ??

## ceph ??

- 1. ?? ?? ???? ???? OSS
- 2. ?? ???? OSD, Monitor, Manager, MDS ???? ??
- 3. ???? ???? Storage pool???? ???? ???? ???? . Crush???? ????  
???? ???? ??

## ????

- 1. Component ??
  - 1. ceph-mon(???? ) : ???? ???? ???? , ???? ???? ???? ???? / HA???? 3? ??
  - 2. ceph-mgr(???? ) : ???? ???? / ???? ? ???? ???? (dashboard ? RestAPI ? ) / HA???? 2? ?? (Active / Standby)
  - 3. ceph-osd(???? ? ) : ???? ???? ? / ???? ???? ???? (OSD???? 1TB? ???? 1G???? ???? ? ) , HA???? ? 3? ??
  - 4. ceph-mds : CEPH FS? ???? ? ???? ???? ???? . = Block Devices / Object Storage???? MDS? ???? ?

### 2. Component Hardware Spec

Component	Hardward	Spec
osd	CPU	OSD? 2 Core
osd	MEM	???? 4GB??
osd	DISK	?? 1TB?? ,(SSD ?? ) ?? ???? ? OSD ???? ???? ?? ???? osd+mon+mds ???? ???? OSD? ???? OS? ???? ? (???? ? )
osd	NIC	10G??
mon	CPU	2?? ?
mon	MEM	???? 24GB??

Component	Hardware	Spec
mon	DISK	100 60GB
mds	CPU	2 4核
mds	MEM	16GB 2GB
mds	DISK	100 1MB
mds	NIC	1Gb

\* OSD RAID10 100 100 100 BMT 100 100 100

## ??? ??

### 1. OSD Backend

#### 1. Bluestore

- Ceph 12.2 默认 storage
- 支持 多种 文件系统 - XFS 文件系统 支持
- RocksDB 支持 多种 文件系统
- 支持 多种 文件系统 checksum 支持 - 支持
- inline - 支持 多种 文件系统 支持
- 支持 多种 文件系统 - journal 支持 多种 文件系统 支持
- CoW 支持 多种 文件系统 IO

#### 2. Filestore

- Ceph 支持 多种 文件系统
- 支持 多种 文件系统 LevelDB 支持 key/value 支持
- 支持 多种 文件系统 btrfs / ext4 支持 多种 文件系统 支持 (XFS 支持)

### 2. Pool

#### 1. 支持 多种 文件系统 支持

- Recovery : 支持 多种 文件系统 OSD
- PG : Pool 支持 多种 文件系统 (支持 OSD 100 PG 支持)
- Cursh Rule : 支持 多种 文件系统 Crush Rule 支持
- Snapshot : 支持 多种 文件系统 支持

#### 2. Pool 支持 多种 文件系统 支持 , RBD 支持 多种 文件系统 RBD

支持 多种 文件系统 (cephfs / rbd / rgw 支持 1)

### 3. CephFS

1. 安装 RADOS 存储池
2. 安装 HA 节点
3. CephFS 存储池配置  
• 配置 pool 名称  
• 配置 pool 的 SSD 大小  
• 配置 inode 大小

### 4. NFSExport

1. NFS-Ganesha NFS 配置 CephFS 存储池 export 配置

## Ceph 部署 (ansible 部署 ceph)

1. 安装 cephadm / Rook / ansible 依赖  
1. cephadm - 安装 binary container python3  
2. Rook - kubernetes 部署 ceph k8s join Rook  
3. ceph-deploy 安装依赖  
2. ceph-ansible 安装 python 依赖

```
$ yum install -y python3 python3-pip sshpass  
$ pip3 install --upgrade setuptools pip --ignore-installed
```

### 3. ceph-ansible 安装

```
$ git clone https://github.com/ceph/ceph-ansible.git -b "v6.0.13"  
$ cd ceph-ansible
```

- ceph-ansible 安装

c	c	a
e	e	n
p	p	si
h-	h	bl
a		e
n		
si		
bl		
e		

3. 0	je w el / lu m in o u s	2. 4
3. 1	lu m in o u s / m i m ic	2. 4
3. 2	lu m in o u s / m i m ic	2. 6
4. 0	n a ut il u s	2. 9

5. 0	o ct  o p u s	2. 9
6. 0	p a c i f i c	2. 9

4. dependency ☐ ☐

```
$ pip3 install -r requirements.txt
```

5. ☐ ☐ ☐ ☐ ☐

```
$ vi hosts

[mons]
192.168.100.41

[osds]
192.168.100.41
192.168.100.42

[mdss]

[rgws]

[nfss]
192.168.100.41

[rbdmirrors]

[clients]
192.168.100.41

[mgrs]
```

```
192.168.100.41
```

```
[iscsigws]
```

```
[iscsi-gws]
```

```
[grafana-server]
```

```
[rgwloadbalancers]
```

```
[monitoring]
```

```
192.168.100.41
```

```
[all:vars]
```

```
ansible_become=true
```

```
ansible_user=root
```

```
ansible_ssh_pass=root
```

#### 6. 配置 (systemd 单元 文件 )

```
$ cp site.yml.sample site.yml
$ cp group_vars/all.yml.sample group_vars/all.yml
$ cp group_vars/osds.yml.sample group_vars/osds.yml
```

#### 7. 配置 (container 单元 文件 )

```
$ cp site-container.yml.sample site.yml
$ cp group_vars/all.yml.sample group_vars/all.yml
$ cp group_vars/osds.yml.sample group_vars/osds.yml
```

#### 8. config 配置 (systemd 单元 文件 )

```
$ vi group_vars/all.yml
...
osd_objectstore: bluestore
monitor_interface: ens3f0
public_network: 192.168.100.0/24
ntp_service_enabled: true
ntp_daemon_type: chronyd
...
```



```
#####  
# DASHBOARD #  
#####  
dashboard_enabled: false  
dashboard_protocol: http  
dashboard_port: 8081  
dashboard_admin_user: admin  
dashboard_admin_password: adminpassword  
containerized_deployment: false  
...  
configure_firewall: false  
...  
ceph_origin: repository  
...  
ceph_repository: community  
...  
ceph_stable_release: octopus
```

```
$ vi group_vars/osds.yml  
...  
devices:  
  - /dev/sdb  
...
```

```
$ vi roles/ceph-validate/tasks/main.yml  
...  
#[] name [] []  
- name: validate ceph_repository_community  
  fail:  
    msg: "ceph_stable_release must be 'pacific'"  
  when:  
    - ceph_origin == 'repository'  
    - ceph_repository == 'community'  
    - ceph_stable_release not in ['pacific']  
...
```

```
Centos7[] systemd [][] [][] dashboard[] [][] [] false[] [][] []
ceph [][] [] [] pacific [][] Centos7[] nfs export[] [] [] octopus[]
[] [] [] []
config [] (ceph[] container[] [][] )
```

```
$ vi group_vars/all.yml
...
osd_objectstore: bluestore
monitor_interface: ens3f0
public_network: 192.168.100.0/24
ntp_service_enabled: true
ntp_daemon_type: chronyd
...
#####
# DASHBOARD #
#####
dashboard_enabled: false
containerized_deployment: true
...
```

```
$ vi group_vars/osds.yml
...
devices:
  - /dev/sdb
...
```

## 9. []

```
$ ansible-playbook -i hosts site.yml -b -v
```

## 10. cluster health check[] warn [][]

```
#Cluster [][] [][] , health check warn[] [][] [] [][] (ceph[]
[] [] )
```

```
$ ceph config set mon auth_allow_insecure_global_id_reclaim false
```

????

## 1. ceph cluster `❏` `❏`

```
$ ceph status
cluster:
  id:      ca96d48d-1c9d-4168-9f21-ffda54a5cd9c
  health: HEALTH_OK

services:
  mon: 2 daemons, quorum openstack-dev1,openstack-dev2 (age 87m)
  mgr: openstack-dev1(active, since 78m), standbys: openstack-dev2
  osd: 3 osds: 3 up (since 83m), 3 in (since 2h)

data:
  pools:   5 pools, 105 pgs
  objects: 49 objects, 5.3 KiB
  usage:    41 MiB used, 300 GiB / 300 GiB avail
  pgs:     105 active+clean
```

## 2. ceph osd `❏` `❏`

```
$ ceph osd tree
```

ID	CLASS	WEIGHT	TYPE NAME	STATUS	REWEIGHT	PRI-AFF
-1		0.29306	root default			
-5		0.09769	host dev1			
2	hdd	0.09769	osd.2	up	1.00000	1.00000
-3		0.09769	host dev2			
0	hdd	0.09769	osd.0	up	1.00000	1.00000
-7		0.09769	host dev3			
1	hdd	0.09769	osd.1	up	1.00000	1.00000

## 3. ceph `❏` latency `❏❏❏`

```
$ ceph osd perf
osd  commit_latency(ms)  apply_latency(ms)
```

2	0	0
0	0	0
1	0	0

# commit 0 0 call 0 0 0 100 ~ 600ms 0 0 0

0

# 0 0 0 0 0 0 (ms 0 , 0 0 0 0 )

#### 4. nfs 0 0 0

```
$ cephadm logs --fsid <fsid> --name nfs.{{ clusteid }}.hostname
```

#### 5. 0 0 0 1. CephFS - Pool 0

```
$ ceph osd lspools
```

##### 1. Pool 0

#Pool 0

```
$ ceph osd pool create {{ DATA_POOL_NAME }}
$ ceph osd pool create {{ METADATA_POOL_NAME }}
```

#CephFS 0 0 0 2 0 RADOS 0 0

##### 2. 0 0 Pool 0 0 0 (cephfs 0 0 )

```
$ ceph osd pool application enable {{ DATA_POOL_NAME }} cephfs
```

##### 3. 0 0 0 0

```
$ ceph fs new {{ FS_NAME }} {{ METADATA_POOL_NAME }} {{DATANAME }}
```

##### 4. NFS export

# 1. nfs module 0

```
$ ceph mgr module enable nfs
```

# 2. nfs ganesha `[[[[]]]` `[[[]]]`

```
$ ceph nfs cluster create {{ clusterid }}
```

# 3. nfs export

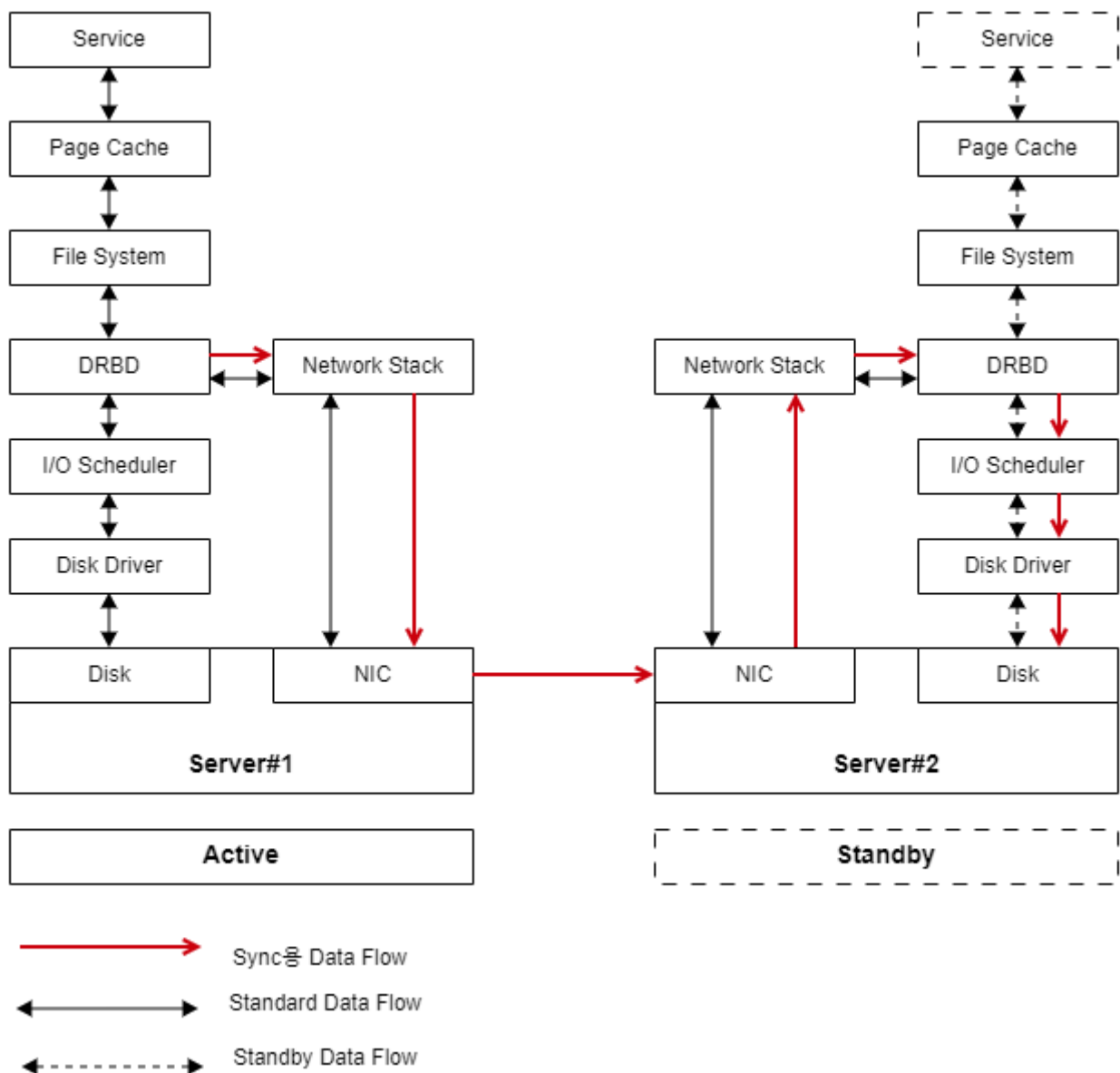
```
$ ceph nfs export create cephfs {{ NAME }} {{ clusterid }}
```

*reference*

- <https://docs.ceph.com/en/latest/architecture/>
- <https://www.slideshare.net/jenshadlich/ceph-object-storage-at-spreadshirt-july-2015-ceph-berlin-meetup>

# DRBD ????

1. Opensource [ ] : DRBD / DRBDUtils (<https://www.linbit.com/en/drbd-oss-distribution/>)
2. DRBD Stack (Active / Passive [ ] )
3. [ ] [ ] [ ] [ ] Mantec [ ] [ ] [ ] :  
<https://mantech.jira.com/wiki/spaces/WDRBDV9/pages/170098908/1>.
4. Mantech[ ] MCCS[ ] [ ] [ ] DRBD[ ] [ ] [ ] [ ]
5. DRBD Data Flow



DRBD????

## 1. DRBD 安装 (8.x, 9.x 版本)

```
> tar -zxvf drbd-8.4.5.tar.gz
> cd drbd-8.4.5
> make;make install
> echo "modprobe drbd" > /etc/sysconfig/modules/drbd.modules
> chmod +x /etc/sysconfig/modules/drbd.modules
> modprobe drbd
> lsmod | grep drbd

drbd                  568788  0
libcrc32c             12644  4 xfs,drbd,nf_nat,nf_contrack
```

## 2. DRBD Util 安装 (8.x, 9.x 版本)

```
> yum install -y libxslt libxml2
> tar -zxvf drbd-utils-8.9.1.tar.gz
> cd drbd-utils-8.9.1
> ./configure --prefix=/usr/local/drbd --sysconfdir=/etc/ --with-initscripttype=sysv
> ./configure --sysconfdir=/etc/ --with-initscripttype=systemd --with-
pacemaker (Centos7)
> make; make install
> mkdir -p /usr/local/drbd/var/run
```

# DRBD????

## 1. Target 磁盘 分区

```
># fdisk /dev/xvdb

Device contains neither a valid DOS partition table, nor Sun, SGI or OSF disklabel
Building a new DOS disklabel with disk identifier 0xcce44d25.

Changes will remain in memory only, until you decide to write them.
After that, of course, the previous content won't be recoverable.

\\Warning: invalid flag 0x0000 of partition table 4 will be corrected by w(rite)
\\WARNING: DOS-compatible mode is deprecated. It's strongly recommended to
switch off the mode (command 'c') and change display units to
sectors (command 'u').

\\Command (m for help): n

Command action
   e   extended
```

```

    p    primary partition (1-4)
p ( )
Partition number (1-4): 1 ( )
First cylinder (1-2080500, default 1):
Using default value 1 ( )
Last cylinder, +cylinders or +size{K,M,G} (1-2080500, default 2080500):
Using default value 2080500
\\Command (m for help): w ( )
The partition table has been altered!
\\Calling ioctl() to re-read partition table.
Syncing disks.

```

## 2. DRBD

```

># cat /etc/drbd.d/drbd.res
resource drbd0
{
    startup {
        wfc-timeout 30;
        outdated-wfc-timeout 20;
        degr-wfc-timeout 30;
    }
    \\ syncer {
        rate 1000M;
        verify-alg sha1;
    }
    \\ on web1 {
        device drbd0;
        disk /dev/xvdb1;
        address {1IP}:7789;
        meta-disk internal;
    }
    \\ on web2 {
        device drbd0;
        disk /dev/xvdb1;
        address {2IP}:7789;
        meta-disk internal;
    }
}

```



### 3. DRBD 11 111 11 1 111 11

```
># drbdadm create-md drbd0
># /etc/init.d/drbd start
```

### 4. Disk Sync 11 (11 1111 11 ) (DRBD 8.x 11 )

```
># drbdadm -- --overwrite-data-of-peer primary drbd0
```

### 5. drbd 11 11 primary 11 (111111 11 )

```
># drbdadm invalidate drbd0
># drbdadm primary --force drbd0
```

### 6. 111 111 11 11 DRBD 111 11 (DRBD 8.x 11 )

```
># cat /proc/drbd
version: 8.4.5 (api:1/proto:86-101)
GIT-hash: 1d360bde0e095d495786eaeb2a1ac76888e4db96 build by root@web2, 2015-03-
09 14:49:34
0: cs:SyncTarget ro:Secondary/Primary ds:Inconsistent/UpToDate C r-----
    ns:0 nr:491520 dw:487424 dr:0 al:0 bm:0 lo:16 pe:0 ua:16 ap:0 ep:1 wo:f oos:99869
56
    [>.....] sync'ed: 4.7% (9752/10228)M
    finish: 0:04:05 speed: 40,616 (40,616) want: 102,400 K/sec
```

### 7. 1111 11 (DRBD 9.0)

```
$> drbdadm status drbd0
drbd0 role:Primary
    disk:UpToDate
    repotx-gitlab-dev02.tx.skp role:Secondary
    replication:SyncSource
    peer-disk:Inconsistent done:8.53
```

## DRBD ?? ???

### 1. DRBD 11 Primary 11 (master 1111 )

```
$> drbdadm primary drbd0
$> cat /proc/drbd    *Primary/Secondary11 1111 111 1
```

```
version: 8.4.5 (api:1/proto:86-101)
GIT-hash: 1d360bde0e095d495786eaeb2a1ac76888e4db96 build by root@techtx-base-
dev02, 2019-04-26 17:54:37
0: cs:Connected ro:Primary/Secondary ds:UpToDate/UpToDate C r-----
   ns:0 nr:0 dw:0 dr:664 al:0 bm:0 lo:0 pe:0 ua:0 ap:0 ep:1 wo:f oos:0
```

2. `mkfs.ext4` (master `mkfs`)

```
># mkfs.ext4 /dev/drbd0
mke2fs 1.41.12 (17-May-2010)
Filesystem label=
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
Stride=0 blocks, Stripe width=0 blocks
327680 inodes, 1310595 blocks
65529 blocks (5.00%) reserved for the super user
First data block=0
Maximum filesystem blocks=1342177280
40 block groups
32768 blocks per group, 32768 fragments per group
8192 inodes per group
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376, 294912, 819200, 884736
Writing inode tables: done
Creating journal (32768 blocks): done
Writing superblocks and filesystem accounting information: done
This filesystem will be automatically checked every 26 mounts or
180 days, whichever comes first. Use tune2fs -c or -i to override.
```

3. `mount` (master `mount`)

```
$> mount /dev/drbd0 /mnt/
$> df -h /mnt
```

Filesystem	Size	Used	Avail	Use%	Mounted on
/dev/drbd0	4.8G	10M	4.6G	1%	/mnt

4. `df` (master `df`)

```
$> touch 123
$> ls -l 123
-rw-r--r-- 1 root root 0 Apr 29 13:04 123
```

## 5. 111 11 1 DRBD 1 secondary 11 (master 1111 11 )

```
$> umount /mnt
$> drbdadm secondary drbd0
$> cat /proc/drbd
version: 8.4.5 (api:1/proto:86-101)
GIT-hash: 1d360bde0e095d495786eaeb2a1ac76888e4db96 build by root@techtx-base-dev02, 2019-04-26 17:54:37
0: cs:Connected ro:Secondary/Secondary ds:UpToDate/UpToDate C r-----
    ns:1235836 nr:0 dw:1235836 dr:1409 al:307 bm:0 lo:0 pe:0 ua:0 ap:0 ep:1 wo:f oos:0
```

## 6. DRBD 1 Pimary 11 (slave 11 11 )

```
$> drbdadm primary drbd0
$> cat /proc/drbd
version: 8.4.5 (api:1/proto:86-101)
GIT-hash: 1d360bde0e095d495786eaeb2a1ac76888e4db96 build by root@techtx-base-dev05, 2019-04-26 17:54:29
0: cs:Connected ro:Primary/Secondary ds:UpToDate/UpToDate C r-----
    ns:4 nr:1235836 dw:1235840 dr:1017 al:1 bm:0 lo:0 pe:0 ua:0 ap:0 ep:1 wo:f oos:0
```

## 7. 1111 111 1 master 1111 111 11 11 (slave 11 11 )

```
># mount /dev/drbd0 /mnt/
># ls -l /mnt/
total 1000020
-rw-r--r-- 1 root root          0 Apr 29 13:04 123
```

# DRBD??? ?? ??? ??? ????

1. crm\_mon -1 111 111 11 Active / Standby 11 11
2. Corocync / Pacemaker / DRBD 111 11
3. DRBD 111 11
4. 1111 11 (Standby 1111 11 )

```
$> drbdadm secondary all
$> drbdadm disconnect all
$> drbdadm -- --discard-my-data connect all
```

5. 主 节点 (Active 节点 节点 )

```
$> drbdadm primary all
$> drbdadm disconnect all
$>drbdadm connect all
```

6. 主 节点 节点 (节点 节点 节点 )

```
$> drbdadm invalidate drbd0
```

# Glusterfs rpm ??

## Centos???? GlusterFS?? ? ??

1. 下载 : <http://download.gluster.org/pub/gluster/glusterfs/3.3/3.3.1/CentOS/>

2. 安装

```
$> yum install -y fuse
$> rpm -ivh glusterfs-3.3.1-1.el5.i386.rpm
$> rpm -ivh glusterfs-fuse-3.3.1-1.el5.i386.rpm
$> rpm -ivh glusterfs-devel-3.3.1-1.el5.i386.rpm
$> rpm -ivh glusterfs-server-3.3.1-1.el5.i386.rpm
```

3. gluster 服务 启动 脚本

```
/etc/rc.d/init.d/glusterd start
chkconfig glusterd on
\\* 重启
/etc/glusterfs/, /var/lib/glusterd/
```

4. 检查 fuse 是否 安装 成功

```
modprobe fuse
echo "modprobe fuse" >> /etc/rc.local
dmesg | grep -i fuse
```

fuse init (API version 7.10) <-- 安装 成功 fuse 服务 启动 成功 .

5. Storage Pool 创建

1. 1. Management 节点 上 执行 以下 命令 (在 所有 节点 上 )

```
gluster peer probe 节点IP
ef) gluster peer probe 192.168.150.19
```

6. Pool 创建

```
[root@localhost ]# gluster peer status
Number of Peers: 1
```

```
Hostname: 192.168.150.19
Uuid: 0bb2e93b-1ab1-4420-91c0-e5bcf203586b
State: Peer in Cluster (Connected)
```

## 7. Storage pool 创建

```
gluster peer detach 192.168.150.19
\\ef) gluster peer detach 192.168.150.19
```

## 8. Volume 创建

### 1. Volume 创建 Distributed(分布式, 副本)

- 创建 3 个 Brick 副本
- Distributed 副本 3 个 Brick 副本, 每个 Brick 副本 1 个副本

### 2. replicated(副本, replication 副本数)

- 创建 3 个 Brick 副本 3 个副本, 每个 Brick 副本 3 个副本

### 3. stripe(副本) 副本数

1. 创建 3 个 Brick 副本
2. Stripe 副本 3 个 Brick 副本, 每个 Brick 副本 3 个副本

## 9. volume 创建 (Distributed 副本)

```
gluster volume create data
vmlnx001:/data vmlnx002:/data vmlnx003:/data vmlnx004:/data
\\ef) gluster volume create data 192.168.150.18:/data 192.168.150.19:/data
\\\\gluster> volume status data
Status of volume: data
Gluster process                                Port      Online   Pid
\\-----
\\Brick 192.168.150.18:/data                    24011     Y       3245
Brick 192.168.150.19:/data                    24011     Y       2481
NFS Server on localhost                       38467     Y       3251
NFS Server on 192.168.150.19                  38467     Y       2487
\\gluster> volume info data
Volume Name: data
Type: Distribute
Volume ID: 556d6065-f888-4198-8782-65bc03979a0b
Status: Started
Number of Bricks: 3
```

```

Transport-type: tcp
Bricks:
Brick1: 192.168.150.18:/data
Brick2: 192.168.150.19:/data
Brick3: 192.168.150.16:/data

```

## 10. volume `data` (Replicate `2` )

```

gluster> volume create data replica 2 192.168.150.18:/data 192.168.150.19:/data
\\glusterfs volume data
gluster> volume info all
Volume Name: data
Type: Replicate
Volume ID: c2ecd1b8-708e-47d4-8f15-adcd1b081987
Status: Started
Number of Bricks: 1 x 2 = 2
Transport-type: tcp
Bricks:
Brick1: 192.168.150.19:/data
Brick2: 192.168.150.18:/data

```

## 11. volume `data` (stripe `2` )

```

gluster volume create data stripe 2 transport tcp 192.168.150.18:/data 192.168.150.19:/data

```

## 12. volume `data` `data` `data` volume `data` `data` , `data` `data` `data`

ef) gluster volume create data replica 3 192.168.150.18:/data 192.168.150.19:/data 192.168.150.16:/data

\\data or a prefix of it is already part of a volume

- `data` `data` , `data` `data` `data` `data` , `glusterfs` `data` `data` .

```

root@localhost data]# ls -al
total 44
drwxr-xr-x  4 root root 4096 12月 6 16:00 .
drwxr-xr-x 24 root root 4096 12月 6 16:01 ..
drw-----  7 root root 4096 12月 6 15:17 .glusterfs
drwx-----  2 root root 16384 12月 5 13:48 lost+found

```

```
setfattr -x trusted.glusterfs.volume-id /data/
setfattr -x trusted.gfid /data/
rm -rf .glusterfs
```

```
gluster> volume start data
```

```
gluster volume set <volume_name> auth.allow <[[IP]]>
ef) gluster> volume set data auth.allow 192.168.150.*
```

```
gluster volume stop <volume_name>
gluster volume delete <volume_name>
\\ef)

gluster volume stop data
gluster volume delete data
```

```
mount -t glusterfs <server_name>:<volume_name> <mount point>
mount -t glusterfs 192.168.150.18:/data /test/
\\[root@localhost ]# df -h
```

Filesystem	Size	Used	Avail	Use%	Mounted on
/dev/sda2	4.9G	1.5G	3.2G	31%	/
/dev/sda5	9.6G	151M	9.0G	2%	/data
/dev/sda1	99M	12M	83M	13%	/boot
tmpfs	1014M	0	1014M	0%	/dev/shm
glusterfs#192.168.150.18:/data					
	9.6G	151M	9.0G	2%	/test

1\*. □ □ , netfs □□□ □□□ □□□ □□□ □□□ .



## 18. Volume 配置

```
## 配置副本 (replicate 配置 两个 副本 Brick 配置 两个 副本 数据)
\\gluster volume add-brick dist_vol 172.27.0.9:/data
ef) gluster volume add-brick data 192.168.150.16:/data
Brick 配置 两个 volume 配置
\\ 配置 两个 )
gluster> volume status data
Status of volume: data
Gluster process                                Port      Online   Pid
-----
Brick 192.168.150.18:/data                      24011     Y        3245
Brick 192.168.150.19:/data                      24011     Y        2481
NFS Server on localhost                        38467     Y        3251
NFS Server on 192.168.150.16                  38467     Y        4216
NFS Server on 192.168.150.19                  38467     Y        2487
\\ 配置 两个 )
gluster> volume status data
Status of volume: data
Gluster process                                Port      Online   Pid
-----
Brick 192.168.150.18:/data                      24011     Y        3245
Brick 192.168.150.19:/data                      24011     Y        2481
Brick 192.168.150.16:/data                      24011     Y        4384
NFS Server on localhost                        38467     Y        3268
NFS Server on 192.168.150.16                  38467     Y        4390
NFS Server on 192.168.150.19                  38467     Y        2501
```

## 19. 性能调优

```
gluster> volume set data performance.write-behind-window-size 1024MB
gluster> volume set data performance.cache-size 512MB
```

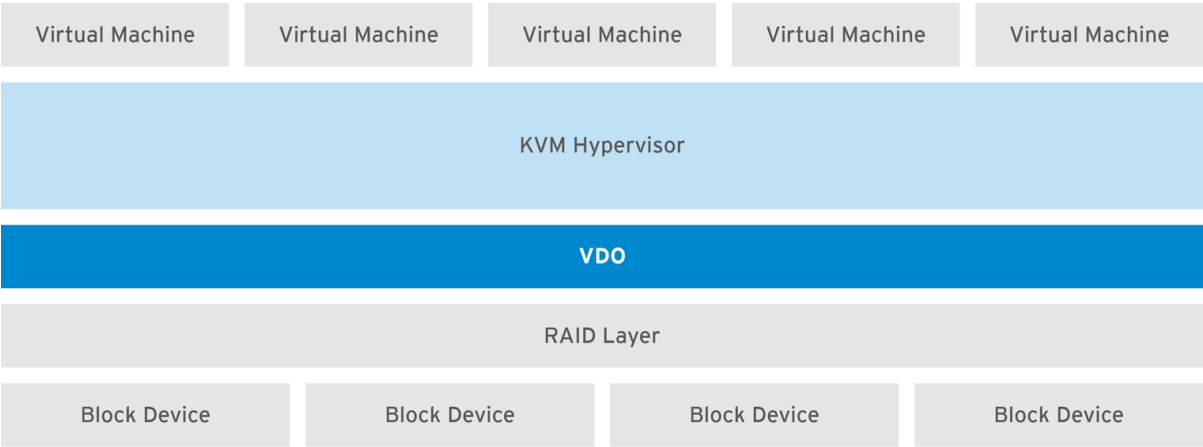
# RHEL???? VDO ?????

## VDO ??

1. VDO; Virtual Data Optimizer 是 一种 软件 技术 , 可 在 虚拟机 中 使用 本地 存储 设备 , 以 提高 性能 和 容量

2. 在 虚拟机 中 使用 VDO

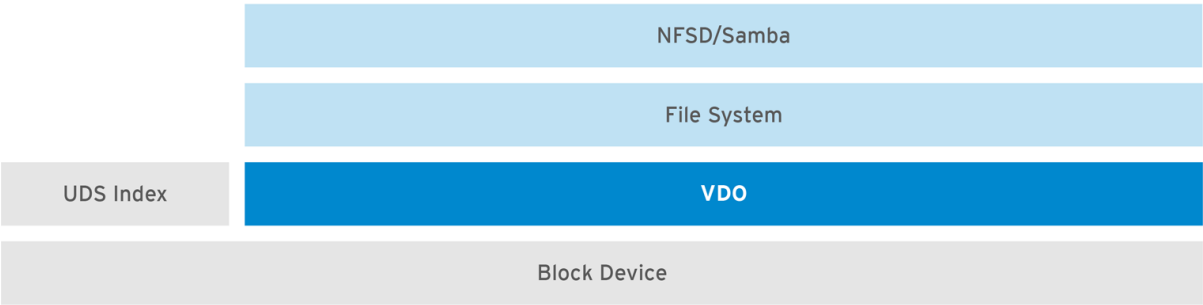
1. VM 使用 本地 存储 设备



RHEL\_462492\_1117

在 虚拟机 中 使用 VDO 时 , 虚拟机 的 磁盘 镜像 文件 将 存储在 本地 存储 设备 上

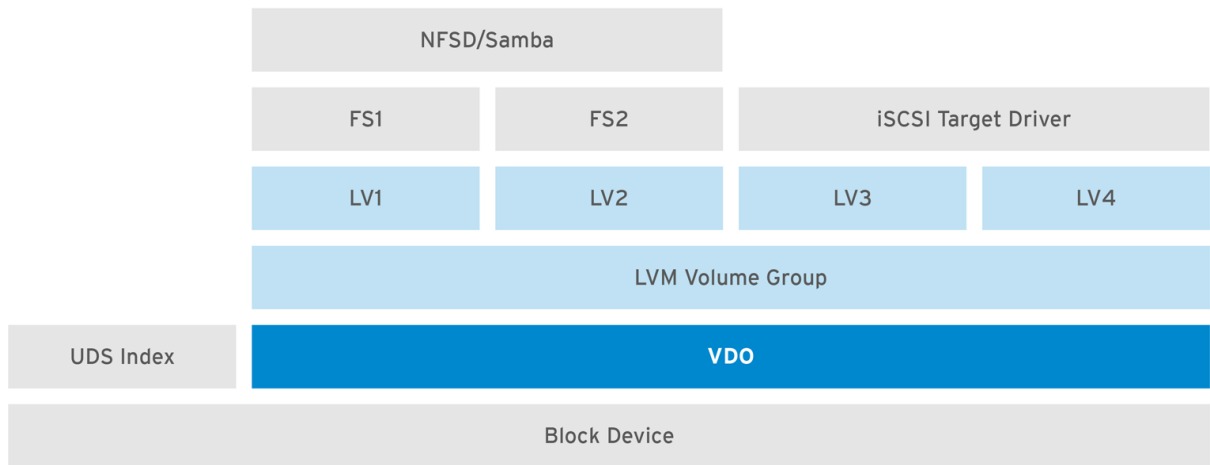
2. NFS 使用 VDO



RHEL\_466924\_0218

VDO 使用 本地 存储 设备 , 在 NFS 和 samba 导出 时

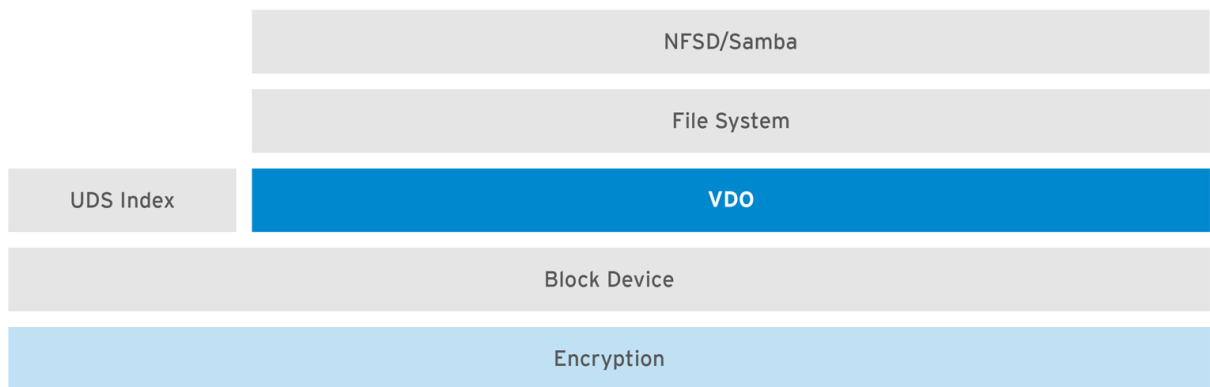
3. VDO 使用 LVM



RHEL\_466924\_0218

VDO 可以 在 LV 上 创建 . 可以 NFS / samba 导出

4. 可以 使用 VDO



RHEL\_466924\_0218

可以 使用 VDO 在 文件 系统 上 创建 快照 . 可以 使用 快照 进行 备份 .

3. 可以 使用 VDO

1. VDO 是 stack 的 component

1. Software RAID / DM Multi path

2. VDO 是 stack

1. LVM cache / snapshot / thin provisioning

3. unsupported stack

1. VDO 是 在 文件 系统 上 , LVM 是 在 VDO,

4. 可以

1. 可以 使用

1. VM 是 在 文件 系统 上 / 在 文件 系统 上 1:10 的 比例

(1TB 的 文件 系统 可以 使用 10TB 的 文件 系统 )

2. Ceph 是 在 1:3 的 比例 使用

5. 

1.  /   I/O

???

1. 

--	--	--

1. VDO ( ) i + ii + iii )

1. 38MB     +   150MB  

2. logical size 1TB ☐ 1.6MB ☐

3. Physical size 1TB □ 268MB □

## 2. UDS Size

1. ☐ 250MB ☐

2. 

<div> <div></div> <div></div> <div></div> <div></div> <div></div> </div>	<div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> </div>	<div> <div></div> <div></div> </div>
--	--	--------------------------------------

D e n s e	1 G B □ □ □ □ 1 T B □ □ □	□ □ □ □ 4 T B □ □ □ □ □ □ 1 G B □ □ □ □ □ □ □
-----------------------	---	---

S	1	□
p	G	□
a	B	□
r	□	□
s	□	,
e	□	□
	□	□
	□	□
	□	□
	1	□
	0	□
	T	4
	B	0
	□	T
	□	B
	□	□
		□
		□
		□
		□
		□
		1
		G
		B
		□
		□
		□
		□
		□
		□
		□
		□

2. 数据

1. physical size 数据 256TB 数据 数据
2. VDO 数据 UDS 数据 数据 数据
3. 数据 数据 数据 4GB 1MB 数据 , slabe 1MB 数据 数据
4. 数据 UDS 数据 数据 dense 1GB 数据 17GB数据 数据 , sparse 170GB 数据 数据

???? ???

1. kvdo : VDO 100GB ~ 99GB
2. uds : VDO 100GB ~ 99GB, physical size 100GB ~ 1TB, logical size 2GB ~ 32GB
3. cli : 100GB ~ 99GB

???? ?? ??

1. physical size : VDO 100GB ~ 99GB, physical size 100GB ~ 1TB
2. logical size : VDO 100GB ~ 99GB, logical size 2GB ~ 32GB, physical size 100GB ~ 1TB, logical size 1:1, logical size 4PB
3. slab size : VDO 100GB ~ 99GB, slab size 2GB ~ 8192GB, Slab size 100GB ~ 99GB, physical size 100GB ~ 99GB, slab size 2GB ~ 8192GB

Physical Size	Slab Size
10 ~ 99GB	1GB
100GB ~ 1TB	2GB
2 ~ 256TB	32GB

slab 100GB vdo 100GB --vdoSlabSize={{ size }} 100GB 100GB

VDO ??

1. VDO 100GB 100GB

```
$ yum install vdo kmod-kvdo
$ lsmod | grep vdo
kvdo                577536  0
uds                 253952  1 kvdo
dm_mod              151552  12 kvdo,dm_thin_pool,dm_bufio
```

## 2. VDO 创建

```
$ vdo create --name= {{ VDO 名称 }} --device={{ 设备名称 }} --
vdoLogicalSize= {{ logical size }}
```

## 3. 格式化

#xfs 格式

```
$ mkfs.xfs -K /dev/mapper/{{ VDO 名称 }}
```

#ext4 格式

```
# mkfs.ext4 -E nodiscard /dev/mapper/{{ VDO 名称 }}
```

## 4. /etc/fstab 配置

# xfs 格式

```
$ /dev/mapper/{{ VDO 名称 }} {{ 设备名称 }} xfs defaults,x-systemd.device-timeout=0,x-
systemd.requires=vdo.service 0 0
```

#ext4 格式

```
$ /dev/mapper/{{ VDO 名称 }} {{ 设备名称 }} ext4 defaults,x-systemd.device-timeout=0,x-
systemd.requires=vdo.service 0 0
```

# VDO ??

## 1. VDO 创建



```
$ vdo start --name= {{ VDO }} or vdo start --all
$ vdo stop --name= {{ VDO }} or vdo stop --all
```

## 2. VDO 启动

#启动

```
$ vdo active --name {{ VDO }} or vdo active --all
```

\\# 查看

```
$ vdo deactivate --name {{ VDO }} or vdo deactivate ~~~all
```

检查 OS 上的 VDO 是否已启动 ( vdo create ~~~-activate=disabled 检查是否已启动 )

## 3. vdo 移除

```
$> vdo remove --name {{ VDO }} or vdo remove --all
```

## 4. VDO 启用 fstrim.timer

```
$ systemctl enable --now fstrim.timer
```

## 5. VDO 状态查看

```
$ vdostats ~~~human-readable
```

\\Device	1K-blocks	Used	Available	Use%	Space saving%
/dev/mapper/nodelosd1	926.5G	21.0G	905.5G	2%	73%
/dev/mapper/nodelosd2	926.5G	28.2G	898.3G	3%	64%

## 6. vdo 扩容

```
$> vdo growLogical --name={{ VDO }} --vdoLogicalSize= {{ 大小 }}
```

## reference

1. [https://access.redhat.com/documentation/en-us/red\\_hat\\_enterprise\\_linux/7/html/storage\\_administration\\_guide/vdo-quick-start](https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/7/html/storage_administration_guide/vdo-quick-start)
2. [https://access.redhat.com/documentation/en-us/red\\_hat\\_enterprise\\_linux/8/html/deduplicating\\_and\\_compressing\\_storage/deploying-vdo\\_deduplicating-and-compressing-storage](https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/8/html/deduplicating_and_compressing_storage/deploying-vdo_deduplicating-and-compressing-storage)