

High Availability for Zimbra FOSS Mailserver System

Deployment Guide

Installation HA for Zimbra mailserver system

Version 1.0

Document Approvers

<i>Name</i>	<i>Title</i>	<i>Email Address</i>	<i>Approval Method</i>
			(Email/Signature)

Distribution List

<i>Name</i>	<i>Title</i>	<i>Email Address</i>	<i>Doc Role</i>
			(Copy/Review)

Related Documents

<i>Reference documents</i>	<i>Business Unit Created</i>	<i>Version</i>	<i>Date</i>	<i>Location</i>

Revision History

(*): A – Added; M – Modified; D – Deleted

<i>Date</i>	<i>Version</i>	<i>Change Item(s)</i>	<i>Change Description</i>	<i>(*)</i>	<i>Author(s)</i>
09 Feb 2015	1.0	Initial		A	Hoang Trong Phuc

Mục Lục

1. Mô hình thực hiện	5
2. Cấu hình HA với 2 server độc lập	7
2.1 FQDN hostnames and IP addresses	8
2.2 Cấu hình network	8
3. Cài đặt drbd	11
3.1 Creat ổ đĩa cài đặt và lưu trữ dữ liệu	11
3.2 Cài đặt ssh nokey trên 2 server	15
3.3 Tải và cài đặt drbd trên cả 2 node	15
4. Cài đặt và cấu hình Zimbra trên 2 server	18
4.1 Cài đặt ở node Master (mail1)	18
4.2 Cài đặt ở node Slaves (mail2)	20
5. Thiết lập OCF (Open Cluster Framework)	21
6. Cài đặt điều khiển HA cho các server	22
6.1 Installation of Corosync & Pacemaker	22
6.2 Config Corosync	23
6.3 Config pacemaker	27
7. Automatic launch of Zimbra on the nodes	32
8. Cấu hình HA sử dụng Storage làm chỗ chứa dữ liệu mailbox	35
8.1 Setup FreeNAS 9	36
8.2 Config FreeNAS 9	41
8.3 Cấu hình 2 mail server kết nối tới storage	45
8.4 Kết nối 2 mailserver vào Storage	45
8.5 Tạo nơi thư mục chứa Index và Store mới	47
8.6 Tạo HA kết nối tới Storage	50
9. Open Cluster Framework Script	52
10. Tài liệu tham khảo	58

SETUP Hight Availability Zimbra mail

1. Mô hình thực hiện

Trước hết, phần thực hiện này được chúng tôi tiến hành dựa trên nền tảng ảo hóa để xây dựng hệ thống. Đi cùng đó là các gói phần mềm, gói hỗ trợ và hệ điều hành mã nguồn mở. Tất cả đều FREE.

Trong phần này, chúng tôi sẽ sử dụng hệ điều hành Centos 6.6, bộ phần mềm ZCS 8.5 FOSS (Zimbra Collaboration - Open Source Edition), Chương trình FreeNAS để tạo SAN.

Tính toán dung lượng ổ cứng cho server mail chạy độc lập

STORAGE CALCULATION EXAMPLE

(Based on 'Mailbox Usage of 1GB and 500 users)

+ User Data: 500 users with 1GB/user = 500 GB user data

+ MySQL data: 5% of 500 GB (User Data): 25 GB

+ Zimbra binaries: 10 GB

+ Zimbra logs: 20 GB

+ Zimbra indexes: 25% of 500GB (User Data) = 125 GB

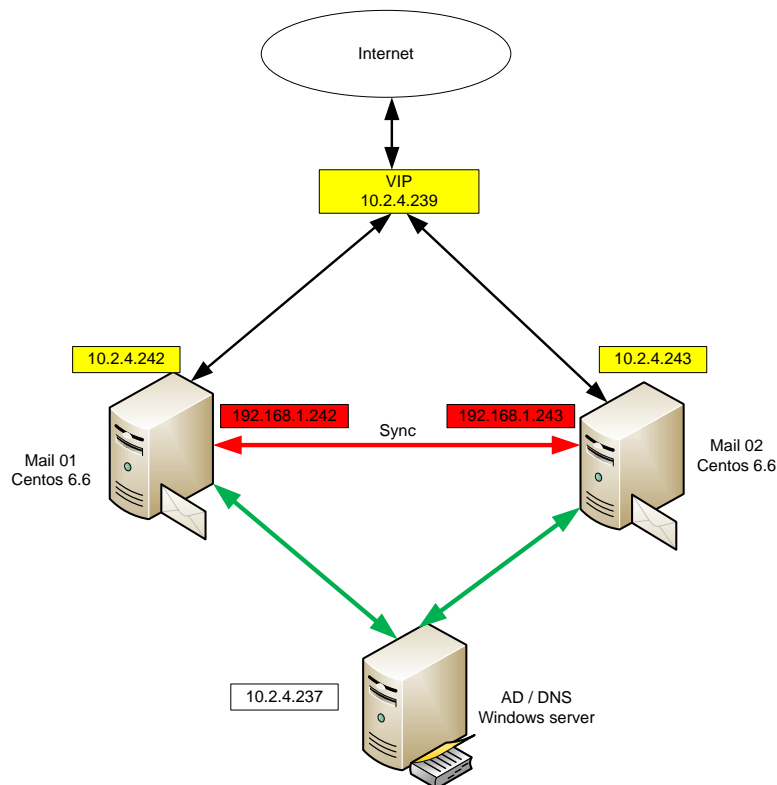
SUBTOTAL: $500 + 25 + 10 + 20 + 125 = 680$ GB

Backups: 160 % of Subtotal: $680 * 160\% = 1088$ GB for backups (= 1,1 Tb)

TOTAL: $680 + 1088 = 1768$ GB (= 1,7 Tb)

Có 2 mô hình thực hiện như dưới đây:

1.1 Phương án 1: Không sử dụng Storage (từ mục 2 tới 7)



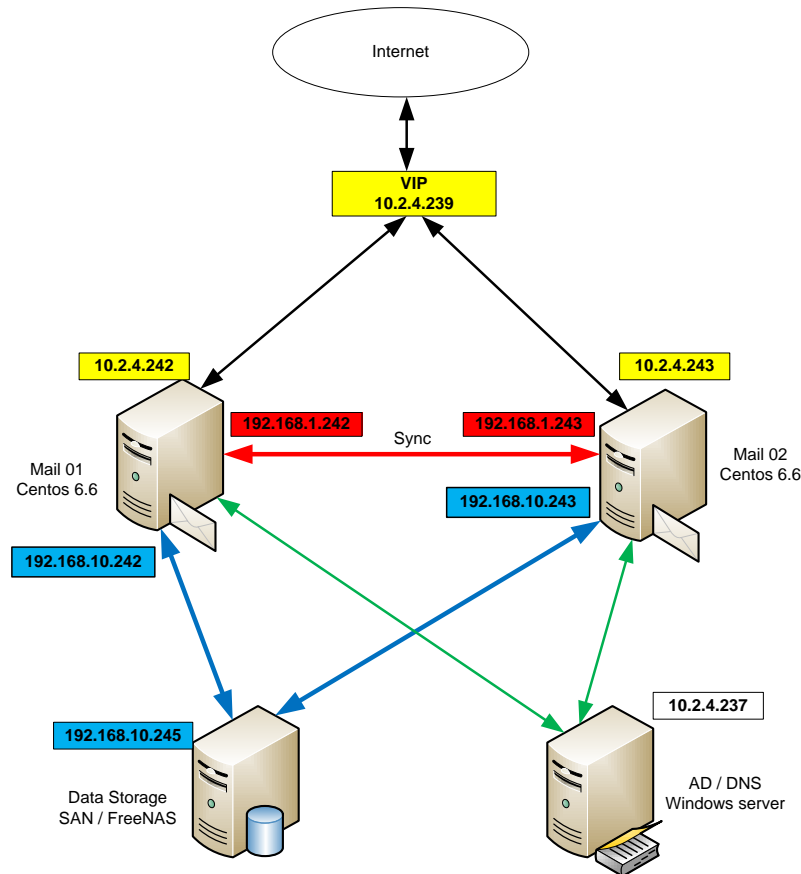
Yêu cầu chuẩn bị:

- 2 server độc lập dùng để cài dịch vụ email chạy hệ điều hành centos 6.6 (free): có ít nhất 2 card mạng 1Gb.

Cấu hình:

- ✓ Intel/AMD 2.0 GHZ+ 64-bit CPU
- ✓ RAM requirements: Minimum of 8GB of RAM is required.
- ✓ 10 GB free disk space for software and logs (SATA or SCSI for performance, and RAID/Mirroring for redundancy)
- ✓ Additional disk space for mail storage, recommend about 1Tb/preserver
- Server AD/DNS cài hệ điều hành Windows Server 2003/2008 (có sẵn).
- Các phần mềm cài đặt: Zimbra 8.5 (FOSS), drbd, Pacemaker, Corosync, Crmsh.

1.2 Phương án 2: Sử dụng Storage (phần này thực hiện tại mục 8)



Yêu cầu chuẩn bị:

- 2 server dùng để cài dịch vụ email chạy hệ điều hành centos 6.6 (free): có ít nhất 3 card mạng 1Gb.

Cấu hình:

- ✓ Intel/AMD 2.0 GHZ+ 64-bit CPU
- ✓ RAM requirements: Minimum of 8GB of RAM is required.
- ✓ 10 GB free disk space for software and logs (SATA or SCSI for performance, and RAID/Mirroring for redundancy)
- ✓ Additional disk space for mail storage
- Server AD/DNS cài hệ điều hành Windows Server 2003/2008 (có sẵn).
- Storage.
- Các phần mềm cài đặt: Zimbra 8.5 (FOSS), drbd, Pacemaker, Corosync, Crmsh.

2. Cấu hình HA với 2 server độc lập

Quy trình cài đặt:

- Setup network trên 2 con server Centos đã dựng sẵn.
- Chuẩn bị Storage (Nếu sử dụng)
- Cài đặt và cấu hình DRBD trên 2 con Server.

- Cài đặt và cấu hình ZCS trên 2 con Server.
- Tạo file OCF (Open Cluster FrameWork).
- Cài đặt HA: dùng Corosync và Pacemaker.
- Cấu hình Corosync và Pacemaker.
- Run and test.

Phần này được cấu hình ở cả 2 node

2.1 FQDN hostnames and IP addresses

	IP	hostname
Virtual IP	10.24.239	Mail
Mail 01	10.2.4.242	Mail1
Mail 02	10.2.4.243	Mail2
AD/DNS	10.2.4.237	Vnpt-tech.org

On both nodes, `/etc/hosts` file should contain the following:

```
127.0.0.1 localhost localhost.localdomain localhost4 localhost4.localdomain4
::1      localhost localhost.localdomain localhost6 localhost6.localdomain6
10.2.4.239 mail.vnpt-tech.org mail
10.2.4.242 mail1.vnpt-tech.org mail1
10.2.4.243 mail2.vnpt-tech.org mail2
```

2.2 Cấu hình network

Trong `/etc/sysconfig/network-scripts:`

Cấu hình đặt ip cho 2 card mạng thật eth0 và eth1.

Creat một card ảo dựa trên card cần tạo ip ảo để public ra ngoài.

Ví dụ tạo card eth0:1 dựa trên card 1 để public ra ngoài. Copy cấu hình `ifcfg-eth0:1` từ `ifcfg-eth0`, thay đổi thông số trong `ifcfg-eth0:1` cho phù hợp.


```
[root@mail2 network-scripts]# ls
ifcfg-eth0    ifdown-bnep  ifdown-post  ifup          ifup-ipv6    ifup-ppp      init.ipv6-global
ifcfg-eth0:1  ifdown-eth   ifdown-ppp   ifup-aliases ifup-isdn    ifup-routes   net.hotplug
ifcfg-eth1    ifdown-ippp  ifdown-routes ifup-bnep     ifup-plip    ifup-sit      network-functions
ifcfg-lo      ifdown-ipv6  ifdown-sit   ifup-eth      ifup-plusb   ifup-tunnel   network-functions-ipv6
ifdown        ifdown-isdn  ifdown-tunnel ifup-ippp     ifup-post    ifup-wireless

[root@mail2 network-scripts]# pwd
/etc/sysconfig/network-scripts
[root@mail2 network-scripts]#
```

Sau khi thực hiện thì tiến hành up cái card mạng ảo mới lên:

```
# ifup ifcfg-eth0:1

# service network restart

# ifconfig
```

```
eth0    Link encap:Ethernet HWaddr 00:50:56:B3:55:1B
        inet addr:10.2.4.242 Bcast:10.2.4.255 Mask:255.255.255.0
        inet6 addr: fe80::250:56ff:feb3:551b/64 Scope:Link
        UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
        RX packets:213515 errors:0 dropped:0 overruns:0 frame:0
        TX packets:51814 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:1000
        RX bytes:21377548 (20.3 MiB) TX bytes:5948433 (5.6 MiB)

eth1    Link encap:Ethernet HWaddr 00:50:56:B3:E3:80
        inet addr:192.168.1.242 Bcast:192.168.1.255 Mask:255.255.255.0
        inet6 addr: fe80::250:56ff:feb3:e380/64 Scope:Link
        UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
        RX packets:364297 errors:0 dropped:0 overruns:0 frame:0
        TX packets:158773 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:1000
        RX bytes:552122368 (526.5 MiB) TX bytes:17916736 (17.0 MiB)

lo      Link encap:Local Loopback
        inet addr:127.0.0.1 Mask:255.0.0.0
        inet6 addr: ::1/128 Scope:Host
        UP LOOPBACK RUNNING MTU:65536 Metric:1
        RX packets:136 errors:0 dropped:0 overruns:0 frame:0
        TX packets:136 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:0
        RX bytes:15988 (15.6 KiB) TX bytes:15988 (15.6 KiB)
```

Cấu hình gateway trong file /etc/sysconfig/network (2 card mạng thì chỉ có 1 gateway public ra, card còn lại chỉ để giao tiếp nội bộ nên ko có gateway)

```
NETWORKING=yes
HOSTNAME=mail2
GATEWAY=10.2.4.1
```

Với hostname: tên của server tương ứng.

Cấu hình file resolv.conf (/etc/resolv.conf)

```
search vnpt-tech.org
nameserver 10.2.4.237
```

Cấu hình xong thì gán thuộc tính readonly cho file resolv này.

```
# sudo chattr +i /etc/resolv.conf
```

Khởi động lại network

```
# service network restart
```

3. Cài đặt drbd

3.1 Creat ổ đĩa cài đặt và lưu trữ dữ liệu.

Trên node 1:

```
[root@mail1 ~]# ls /dev/sd*
/dev/sda /dev/sda1 /dev/sda2 /dev/sdb
[root@mail1 ~]# fdisk /dev/sdb
```

Device contains neither a valid DOS partition table, nor Sun, SGI or OSF disklabel

Building a new DOS disklabel with disk identifier 0xfc03d3bb.

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 (1-5221, default 1):

Using default value 1

Last cylinder, +cylinders or +size{K,M,G} (1-5221, default 5221):

Using default value 5221

Command (m for help): w

The partition table has been altered!

```
Calling ioctl() to re-read partition table.  
Syncing disks.
```

Trên node số 2:

```
[root@mail2 ~]# ls /dev/sd*  
/dev/sda  /dev/sda1  /dev/sda2  /dev/sdb  
[root@mail2 ~]# fdisk /dev/sdb
```

Device contains neither a valid DOS partition table, nor Sun, SGI or OSF disklabel

Building a new DOS disklabel with disk identifier 0xfc03d3bb.

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 (1-5221, default 1):

Using default value 1

Last cylinder, +cylinders or +size{K,M,G} (1-5221, default 5221):

Using default value 5221

Command (m for help): w

The partition table has been altered!

```
Calling ioctl() to re-read partition table.  
Syncing disks.
```

3.2 Cài đặt ssh nokey trên 2 server

Tham khảo tài liệu “[05_TroubleShot SSH no key.docx](#)” trong mục tài liệu

3.3 Tải và cài đặt drbd trên cả 2 node

```
[root@mail1 ~]# rpm -ivh http://elrepo.org/elrepo-release-6-5.el6.elrepo.noarch.rpm
```

```
[root@server1 ~]# chkconfig --level 345 drbd on
```

```
[root@server1 ~]# yum install -y kmod-drbd84
```

Hoặc tải 2 gói dưới về và sử dụng câu lệnh: **rpm --nodeps -ivh “tên gói”** để cài đặt.

drbd-kmdl-2.6.32-431.el6-8.4.3-33.el6.x86_64

drbd-8.4.3-33.el6.x86_64

```
[root@mail1 ~]# vi /etc/drbd.d/global_common.conf
```

Thêm dòng sau vào (hoặc chỉnh sửa)

```
startup {  
    wfc-timeout 60;  
}  
  
syncer {  
    rate 1000M;  
}
```

```
[root@mail1 ~]# scp /etc/drbd.d/global_common.conf  
mail2:/etc/drbd.d/global_common.conf
```

Tạo file **r0.res** trong **/etc/drbd.d/** (r0.res: có thể đổi tên khác. Khi thay đổi phải lưu ý ở các bước đăng sau và bước cấu hình HA)

```
[root@mail1 ~]# nano /etc/drbd.d/r0.res
```

```
resource r0 {
    net {
        protocol C;
    }
    on mail1 {
        device /dev/drbd0;
        disk /dev/sdb;
        address 192.168.1.242:7788;
        meta-disk internal;
    }
    on mail2 {
        device /dev/drbd0;
        disk /dev/sdb;
        address 192.168.1.243:7788;
        meta-disk internal;
    }
}
```

Các giá trị: mail1, mail2: được lấy từ câu lệnh: `uname -n`

Device: phần ổ sẽ thực hiện nhiệm vụ mount và đồng bộ

Disk: chọn ổ đĩa cần đồng bộ sang nhau

Address: ip của 2 máy cần đồng bộ, và đồng bộ qua giao tiếp bằng địa chỉ này.

Remove from `/etc/fstab` file any reference to `/dev/vdb1` or `/dev/vdc1` devices, as drbd is going to handle its mounting.

Initialize data and metadata disks:

```
[root@mail1 ~]# dd if=/dev/zero of=/dev/sdb bs=1K count=100
```

```
[root@mail2 ~]# dd if=/dev/zero of=/dev/sdb bs=1K count=100
```

```
[root@mail1 ~]# scp /etc/drbd.d/r0.res
```



```
mail2:/etc/drbd.d/r0.res
```

```
[root@mail1 ~]# drbdadm create-md r0
```

```
[root@mail1 ~]# ssh mail2
```

```
[root@mail2 ~]# drbdadm create-md r0
```

r0: lấy theo tên của file tạo lúc ban đầu ở trên

```
[root@mail1 ~]# service drbd start
```

```
[root@mail2 ~]# service drbd start
```

Checking.....

```
[root@mail1 ~]# ps ax | grep drbd
```

```
12229 ?          S        0:00 [drbd-reissue]
12234 ?          S        0:00 [drbd_submit]
12242 ?          S        0:00 [drbd_w_r0]
12248 ?          S        0:00 [drbd_r_r0]
12255 ?          S        0:00 [drbd_a_r0]
12259 pts/0      S+       0:00 grep drbd
```

```
[root@mail2 ~]# ps ax | grep drbd
```

```
11871 ?          S        0:00 [drbd-reissue]
11877 ?          S        0:00 [drbd_submit]
11883 ?          S        0:00 [drbd_w_r0]
11890 ?          S        0:00 [drbd_r_r0]
11897 ?          S        0:00 [drbd_a_r0]
```

Đặt mail1 là server primary, mail2 là secondary

```
[root@mail1 ~]# drbdadm -- --overwrite-data-of-peer primary
r0
```

Tạo file system:

```
[root@mail1 ~]# mkfs.ext4 /dev/drbd0
```

Kiểm tra lại ở 2 node

```
[root@mail1 ~]# cat /proc/drbd
```

```
version: 8.4.5 (api:1/proto:86-101)
GIT-hash: 1d360bde0e095d495786eae2a1ac76888e4db96 build by phil@Build64R6,
2014-10-28 10:32:53
0: cs:Connected ro:Primary/Secondary ds:UpToDate/UpToDate C r-----
   ns:647652 nr:385528 dw:1033180 dr:903565 al:111 bm:0 lo:0 pe:0 ua:0 ap:0 ep:1
wo:f oos:0
```

```
[root@mail2 ~]# cat /proc/drbd
```

```
version: 8.4.5 (api:1/proto:86-101)
GIT-hash: 1d360bde0e095d495786eae2a1ac76888e4db96 build by phil@Build64R6,
2014-10-28 10:32:53
0: cs:Connected ro:Secondary/Primary ds:UpToDate/UpToDate C r-----
   ns:0 nr:653696 dw:653696 dr:0 al:0 bm:0 lo:0 pe:0 ua:0 ap:0 ep:1 wo:f oos:0
```

DONE

4. Cài đặt và cấu hình Zimbra trên 2 server

Note:

Nếu lúc cài đặt có lỗi nào xảy ra thì cần uninstall toàn bộ zimbra đi rồi mới cài lại.
Uninstall zimbra: Vào thư mục cài đặt mà giải nén ra lúc đầu. Chạy lệnh:

```
[root@mail2 ~]# ./install.sh -u -platform-override
```

4.1 Cài đặt ở node Master (mail1)

Kiểm tra trạng thái máy 1 ở chế độ DRBD Primary. Nếu chưa ở chế độ Primary thì chuyển nó về chế độ Primary.

```
[root@mail1 ~]# cat /proc/drbd
```

```
version: 8.4.5 (api:1/proto:86-101)
GIT-hash: 1d360bde0e095d495786eae2a1ac76888e4db96 build by phil@Build64R6,
2014-10-28 10:32:53
0: cs:Connected ro:Primary/Secondary ds:UpToDate/UpToDate C r-----
   ns:647652 nr:385528 dw:1033180 dr:903565 al:111 bm:0 lo:0 pe:0 ua:0 ap:0 ep:1
wo:f oos:0
```

Create directory for ZCS:

```
[root@mail1 ~]# mkdir /opt/zimbra
```

Mount DRBD device on ZCS mount point:

```
[root@mail1 ~]# mount /dev/drbd0 /opt/zimbra
```

Check mounted device:

```
[root@mail1 ~]# df | grep zimbra
```

```
[root@mail1 ~]# mount | grep zimbra
```

Set manual virtual link configuration temporally:

```
[root@mail1 ~]# ifconfig eth0:1 inet 10.2.4.239 netmask 255.255.255.0
```

Set split DNS hostname temporally:

```
[root@mail1 ~]# hostname mail
```

It is also recommendable to change on **/etc/sysconfig/network** file.

Unpack ZCS installer and proceed with full installation:

```
[root@mail1 ~]# tar xzf zcs-8...
```

```
[root@mail1 ~]# ./install.sh -platform-override
```

Setup bình thường như trong hướng dẫn cài đặt.

Sau cùng: Apply configuration and advance until ZCS setup process is completed:

“Configuration complete - press return to exit”

Check ZCS status:

```
[root@mail1 ~]# service zimbra status
```

```
[root@mail1 ~]# service postfix stop
```

```
[root@mail1 ~]# chkconfig postfix off
```

```
[root@mail1 ~]# service zimbra restart
```

Stop ZCS service

```
[root@mail1 ~]# service zimbra stop
```

Umount DRBD device:

```
[root@mail1 ~]# umount /opt/zimbra
```

Đặt DNS hostname về ban đầu

```
[root@mail1 ~]# hostname mail1
```

Revert change in /etc/sysconfig/network file as needed.

Delete temporal virtual link configuration:

```
[root@mail1 ~]# ifconfig eth0:1 down
```

Chuyển trạng thái drbd về secondary

```
[root@mail1 ~]# drbdadm secondary r0
```

4.2 Cài đặt ở node Slaves (mail2)

Cài đặt full tương tự như ở trên máy Master.

Chuyển node 2 về chế độ drbd Primary:

```
[root@mail2 ~]# drbdadm primary r0
```

```
[root@mail2 ~]# cat /proc/drbd
```

```
version: 8.4.5 (api:1/proto:86-101)
GIT-hash: 1d360bde0e095d495786eae2a1ac76888e4db96 build by phil@Build64R6,
2014-10-28 10:32:53
0: cs:Connected ro:Primary/Secondary ds:UpToDate/UpToDate C r-----
   ns:647652 nr:385528 dw:1033180 dr:903565 al:111 bm:0 lo:0 pe:0 ua:0 ap:0 ep:1
wo:f oos:0
```

Create directory for ZCS:

```
[root@mail2 ~]# mkdir /opt/zimbra
```

Mount DRBD device on ZCS mount point:

```
[root@mail2 ~]# mount /dev/drbd0 /opt/zimbra
```

Check mounted device:

```
[root@mail2 ~]# df | grep zimbra
```

```
[root@mail2 ~]# mount | grep zimbra
```

Set manual virtual link configuration temporally:

```
[root@mail2 ~]# ifconfig eth0:1 inet 10.2.4.239 netmask
255.255.255.0
```

Set split DNS hostname temporally:

```
[root@mail2 ~]# hostname mail
```

It is also recommendable to change on `/etc/sysconfig/network` file.

Unpack ZCS installer and proceed with full installation:

```
[root@mail2 ~]# tar xzf zcs-8...  
[root@mail2 ~]# ./install.sh -platform-override
```

Setup bình thường như trong hướng dẫn cài đặt.

Sau cùng: Apply configuration and advance until ZCS setup process is completed:

“Configuration complete - press return to exit”

Check ZCS status:

```
[root@mail2 ~]# service zimbra status  
[root@mail2 ~]# service postfix stop  
[root@mail2 ~]# chkconfig postfix off  
[root@mail2 ~]# service zimbra restart
```

Stop ZCS service

```
[root@mail2 ~]# service zimbra stop
```

Umount DRBD device:

```
[root@mail2 ~]# umount /opt/zimbra
```

Đặt DNS hostname về ban đầu

```
[root@mail2 ~]# hostname mail1
```

Revert change in `/etc/sysconfig/network` file as needed.

Delete temporal virtual link configuration:

```
[root@mail2 ~]# ifconfig eth0:1 down
```

Chuyển trạng thái drbd về secondary

```
[root@mail2 ~]# drbdadm secondary r0
```

5. Thiết lập OCF (Open Cluster Framework)

Open Cluster Framework, standard scripts to control services such as ZCS.

Việc thiết lập này được thực hiện trên cả 2 node Master and Slaves.

Tạo thư mục btactic ở trong: /usr/lib/ocf/resource.d/

```
[root@mail2 ~]# mkdir /usr/lib/ocf/resource.d/btactic
```

Tạo file zimbra ở trong thư mục btactic vừa tạo.

```
[root@mail2 ~]# touch /usr/lib/ocf/resource.d/btactic/zimbra
```

Tiến hành thêm vào file zimbra script OCF. (Script ở trang cuối)

```
[root@mail2 ~]# chmod +x /usr/lib/ocf/resource.d/btactic/*
```

Tạo liên kết mềm tới thư mục heartbeat:

```
[root@mail2 ~]# ln -s /usr/lib/ocf/resource.d/btactic/zimbra
/usr/lib/ocf/resource.d/heartbeat/
```

(Cái này dùng ở phần sau)

6. Cài đặt điều khiển HA cho các server

6.1 Installation of Corosync & Pacemaker

Dependencies

```
[root@mail1]# wget
http://download3.fedora.redhat.com/pub/epel/5/x86_64/libesntp
-1.0.4-5.el5.x86_64.rpm
[root@mail1]# rpm -ivh libesntp-1.0.4-5.el5.x86_64.rpm
```

Configure YUM

```
[root@mail1]# wget -O /etc/yum.repos.d/pacemaker.repo
http://clusterlabs.org/rpm/epel-5/clusterlabs.repo
```

Install the software

```
[root@node1]# yum install pacemaker.x86_64 corosync.x86_64
```

Install crmsh

The CRM Shell for Pacemaker cluster management can be installed by fetching the OpenSUSE ha-clustering repository and installing `crmsh` vi yum.

```
[root@mail1]# cd /etc/yum.repos.d/
[root@mail1]# wget
http://download.opensuse.org/repositories/network:/ha-
clustering:/Stable/CentOS_CentOS-6/network:ha-
clustering:Stable.repo
[root@mail1]# yum -y install crmsh
```

6.2 Config Corosync

a. Tạo Corosync Key

Thực hiện tạo **Corosync Key** trên một node. Ví dụ ở node mail1.

```
[root@mail]# corosync-keygen
Corosync Cluster Engine Authentication key generator.
Gathering 1024 bits for key from /dev/random.
Press keys on your keyboard to generate entropy.
Writing corosync key to /etc/corosync/authkey.
```

Copy file authkey đó tới node thứ 2 và set permissions to **400**

```
[root@mail1]# scp /etc/corosync/authkey node2:/etc/corosync/
[root@mail1]# ll /etc/corosync/authkey
-r----- 1 root root 128 May  7 10:26 /etc/corosync/authkey
[root@mail2]# ll /etc/corosync/authkey
-r----- 1 root root 128 May  7 10:27 /etc/corosync/authkey
```

b. Corosync.conf

Now configure the **/etc/corosync/corosync.conf**

Ring number bắt đầu từ 0, máy có bao nhiêu card mạng thì cấu hình bấy nhiêu ring number cho chắc ăn. :d.

```
# Please read the corosync.conf.5 manual page
compatibility: whitetank
aisexec {
```

```
# Run as root - this is necessary to be able to manage resources with Pacemaker
user: root
group: root
}
service {
    # Load the Pacemaker Cluster Resource Manager
    ver:    0
    name:   pacemaker
    use_logd: yes
}
totem {
    version: 2
    secauth: off
    threads: 0
    rrp_mode: passive
    interface {
        ringnumber: 0
        bindnetaddr: 192.168.1.0
        mcastaddr: 239.255.1.177
        mcastport: 5409
    }
    interface {
        ringnumber: 1
        bindnetaddr: 10.2.4.0
        mcastaddr: 239.255.1.178
        mcastport: 5411
    }
}
logging {
```



```
fileline: off
to_stderr: yes
to_logfile: yes
to_syslog: yes
logfile: /var/log/cluster/corosync.log
# debug: off
debug: on
timestamp: on
logger_subsys {
    subsys: AMF
    debug: off
}
}
amf {
    mode: disabled
}
```

c. Activate service

```
[root@mail1]# chkconfig corosync on
[root@mail1]# chkconfig logd on
```

d. Start service trên cả 2 node và check

```
[root@mail1]# service corosync start
[root@mail2]# service corosync start

[root@mail1]# corosync-cfgtool -s
Printing ring status.
Local node ID -1324290657
RING ID 0

    id      = 192.168.1.242
```

status = ring 0 active with no faults

RING ID 1

id = 10.2.4.242

status = ring 1 active with no faults

The state of the cluster

```
[root@mail1]# crm_mon -rf
```

=====

Last updated: Mon May 7 12:36:05 2012

Stack: openais

Current DC: node1 - partition with quorum

Version: 1.0.12-unknown

2 Nodes configured, 2 expected votes

0 Resources configured.

=====

Online: [node1 node2]

Full list of resources:

Migration summary:

* Node node1:

* Node node2:

e. Check Corosync

```
[root@mail1]# crm_verify -L -V
```

error: unpack_resources: Resource start-up disabled since no STONITH resources have been defined

error: unpack_resources: Either configure some or disable STONITH with the stonith-enabled option

error: unpack_resources: NOTE: Clusters with shared data need STONITH to ensure data integrity

Errors found during check: config not valid

-V may provide more details

As you can see, the tool has found some errors.

In order to guarantee the safety of your data ^[8], the default for STONITH ^[9] in Pacemaker is **enabled**. However it also knows when no STONITH configuration has been supplied and reports this as a problem (since the cluster would not be able to make progress if a situation requiring node fencing arose).

For now, we will disable this feature and configure it later in the Configuring STONITH section. It is important to note that the use of STONITH is highly encouraged, turning it off tells the cluster to simply pretend that failed nodes are safely powered off. Some vendors will even refuse to support clusters that have it disabled.

To disable STONITH, we set the *stonith-enabled* cluster option to false.

```
[root@mail1]# crm configure property stonith-enabled=false
[root@mail1]# crm_verify -L
```

With the new cluster option set, the configuration is now valid.

6.3 Config pacemaker

Trước tiên, bước chuẩn bị ban đầu ta cần turn off STONITH và đặt `no-quorum-policy="ignore"`

```
[root@mail1]# crm configure
crm(live)configure# property stonith-enabled="false" no-
quorum-policy="ignore"
```

Sau đó đánh giá và lưu giá trị bằng: **commit**

Mục đích của `no-quorum-policy` là để khi một trong hai máy tắt thì máy kia vẫn hoạt động bình thường. Nếu không chọn thì 2 máy hoạt động một máy bị shutdown, máy kia sẽ die theo.

I tested it both stonith-enabled and no-quorum-policy. As Dejan pointed, this is related to stonith-enabled. With stonith-enabled true (which is default), if I kill the master node, the slave stays as a slave, it seems expecting something from stonith. With stonith-enabled false, the slave was promoted to master.

The no-quorum-policy controls the behavior when the quorum is not enough. With no-quorum-policy stop, when one of two nodes dies, the resources in the other node get stopped. With no-quorum-policy ignore, when one of two nodes dies, nothing happens. If master die, the slave stays as a slave. If slave dies, the master stays as a master.

By the way during the test, no-quorum-policy works only in openais stack. With heartbeat

stack, there is no difference between no-quorum-policy stop and ignore. It seems the heartbeat stack always behavior as no-quorum-policy ignore.

- Check the config with show.
- Push the modifications on all the nodes with the command commit.

To sum up:

```
crm(live)configure# show
property cib-bootstrap-options: \
    dc-version=1.1.11-97629de \
    cluster-infrastructure="classic openais (with plugin)" \
    expected-quorum-votes=2 \
    stonith-enabled=false \
    no-quorum-policy=ignore
rsc_defaults rsc-options: \
    resource-stickiness=INFINITY \
    migration-threshold=1
```

Tiếp theo sẽ đặt virtual-IP:

```
crm(live)configure#           primitive           FAILOVER-ADDR
ocf:heartbeat:IPaddr2 params ip="10.2.4.239" nic="eth0:1"
op monitor interval="10s" meta is-managed="true"
```

Of course there are much more options available like Nic. (eth0:1 set lúc ban đầu)

crm(live)configure#

Chuẩn bị môi trường:

```
property cib-bootstrap-options: \
    dc-version=1.1.11-97629de \
    cluster-infrastructure="classic openais (with plugin)" \
    expected-quorum-votes=2 \
    stonith-enabled=false \
    no-quorum-policy=ignore
rsc_defaults rsc-options: \
    resource-stickiness=INFINITY \
    migration-threshold=1
```

Creat Vituarl IP:

```
primitive FAILOVER-ADDR IPaddr2 \
    params ip=10.2.4.239 nic="eth0:1" \
    op start interval=0s timeout=60s \
    op monitor interval=10s timeout=20s \
    meta is-managed=true
```

Definition of the DRBD resource:

```
primitive drbd0 ocf:linbit:drbd \
    params drbd_resource=r0 drbdconf="/etc/drbd.conf" \
    op monitor role=Master interval=59s timeout=30s \
    op monitor role=Slave interval=60s timeout=30s
```

*có thể thử thay **linbit** bằng **heartbeat** xem được ko (recommand là heartbert, nhưng tôi không change được, phải sử dụng là linbit):*

```
primitive drbd0 ocf:heartbeat:drbd \
    params drbd_resource=r0 drbdconf="/etc/drbd.conf" \
    op monitor role=Master interval=59s timeout=30s \
    op monitor role=Slave interval=60s timeout=30s
```

Role parameters:

```
ms ms_drbd0 drbd0 \
    meta master-max=1 master-node-max=1 clone-max=2 clone-node-max=1
```

```
notify=true globally-unique=false target-role=Started
```

Definition of the mounted file system resource:

```
primitive fs-zimbra Filesystem \
```

```
    params device="/dev/drbd0" directory="/opt/zimbra" fstype=ext4
```

Creation of a group containing the VIP resource, and the FS resource, this group is denied in colocation on ms-zimbra (our main resource).

```
group zimbra-group FAILOVER-ADDR fs-zimbra
```

```
colocation zimbra-group-on-ms-drbd0 inf: zimbra-group ms_drbd0:Master
```

Definition of the starting order of resources, on a unique Master (Mail1 if possible):

```
order ms-zimbra-before-zimbra-group inf: ms_drbd0:promote zimbra-group:start
```

Chữ màu đỏ và xanh có thể thay đổi. Nhưng chú ý lúc nhóm group phải đúng tên.

Drbd resource: theo file *.res để đặt. Nếu đặt là r0 thì phải là r0, nếu là zimbra thì phải là zimbra

```
crm(live)configure# show
```

```

node mail1
node mail2
primitive FAILOVER-ADDR IPaddr2 \
    params ip=10.2.4.239 nic="eth0:1" \
    op start interval=0s timeout=60s \
    op monitor interval=10s timeout=20s \
    meta is-managed=true
primitive drbd0 ocf:linbit:drbd \
    params drbd_resource=r0 drbdconf="/etc/drbd.conf" \
    op monitor role=Master interval=59s timeout=30s \
    op monitor role=Slave interval=60s timeout=30s
primitive fs-zimbra Filesystem \
    params device="/dev/drbd0" directory="/opt/zimbra" fstype=ext4
group zimbra-group FAILOVER-ADDR fs-zimbra zcs_service
ms ms_drbd0 drbd0 \
    meta master-max=1 master-node-max=1 clone-max=2 clone-node-max=1
notify=true globally-unique=false target-role=Started
colocation zimbra-group-on-ms-drbd0 inf: zimbra-group ms_drbd0:Master
order ms-zimbra-before-zimbra-group inf: ms_drbd0:promote zimbra-group:start
property cib-bootstrap-options: \
    dc-version=1.1.11-97629de \
    cluster-infrastructure="classic openais (with plugin)" \
    expected-quorum-votes=2 \
    stonith-enabled=false \
    no-quorum-policy=ignore
rsc_defaults rsc-options: \
    resource-stickiness=INFINITY \
    migration-threshold=1

```

7. Automatic launch of Zimbra on the nodes

Tạo thư mục **btactic**: (đã làm ở mục 5)

Liên kết script đó tới phần heartbeat:

```
ln -s /usr/lib/ocf/resource.d/btactic/zimbra /usr/lib/ocf/resource.d/heartbeat/
```

Sau đó thực hiện cấu hình trong **crm configure**:

Config Zimbra service start with HA

```
primitive zcs_service ocf:btactic:zimbra \  
    op monitor interval=120s timeout=40s \  
    op start interval=0s timeout=360s \  
    op stop interval=0s timeout=360s
```

Gom vào group để khởi động:

```
group zimbra-group FAILOVER-ADDR fs-zimbra zcs_service
```

```
crm(live)configure# show
```



```

node mail1
node mail2
primitive FAILOVER-ADDR IPaddr2 \
    params ip=10.2.4.239 nic="eth0:1" \
    op start interval=0s timeout=60s \
    op monitor interval=10s timeout=20s \
    meta is-managed=true
primitive drbd0 ocf:linbit:drbd \
    params drbd_resource=r0 drbdconf="/etc/drbd.conf" \
    op monitor role=Master interval=59s timeout=30s \
    op monitor role=Slave interval=60s timeout=30s
primitive fs-zimbra Filesystem \
    params device="/dev/drbd0" directory="/opt/zimbra" fstype=ext4
primitive zcs_service ocf:btactic:zimbra \
    op monitor interval=120s timeout=40s \
    op start interval=0s timeout=360s \
    op stop interval=0s timeout=360s
group zimbra-group FAILOVER-ADDR fs-zimbra zcs_service
ms ms_drbd0 drbd0 \
    meta master-max=1 master-node-max=1 clone-max=2 clone-node-max=1
notify=true globally-unique=false target-role=Started
colocation zimbra-group-on-ms-drbd0 inf: zimbra-group ms_drbd0:Master
order ms-zimbra-before-zimbra-group inf: ms_drbd0:promote zimbra-group:start
property cib-bootstrap-options: \
    dc-version=1.1.11-97629de \
    cluster-infrastructure="classic openais (with plugin)" \
    expected-quorum-votes=2 \
    stonith-enabled=false \
    no-quorum-policy=ignore
rsc_defaults rsc-options: \

```

```
resource-stickiness=INFINITY \  
migration-threshold=1
```

Finally, this is the output of a **crm status**:

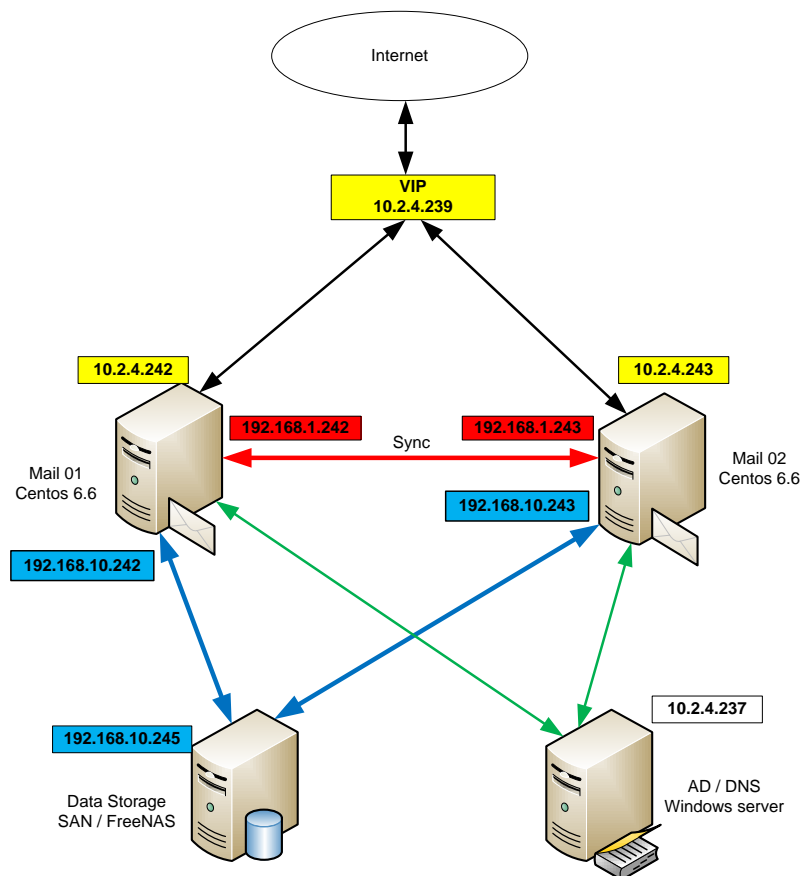
```
Last updated: Mon Feb 2 17:35:52 2015  
Last change: Mon Feb 2 16:45:36 2015  
Stack: classic openais (with plugin)  
Current DC: mail2 - partition with quorum  
Version: 1.1.11-97629de  
2 Nodes configured, 2 expected votes  
5 Resources configured  
  
Online: [ mail1 mail2 ]  
  
Resource Group: zimbra-group  
  FAILOVER-ADDR (ocf::heartbeat:IPaddr2):   Started mail2  
  fs-zimbra (ocf::heartbeat:Filesystem):   Started mail2  
  zcs_service (ocf::btactic:zimbra): Started mail2  
Master/Slave Set: ms_drbd0 [drbd0]  
  Masters: [ mail2 ]  
  Slaves: [ mail1 ]
```

8. Cấu hình HA sử dụng Storage làm chỗ chứa dữ liệu mailbox

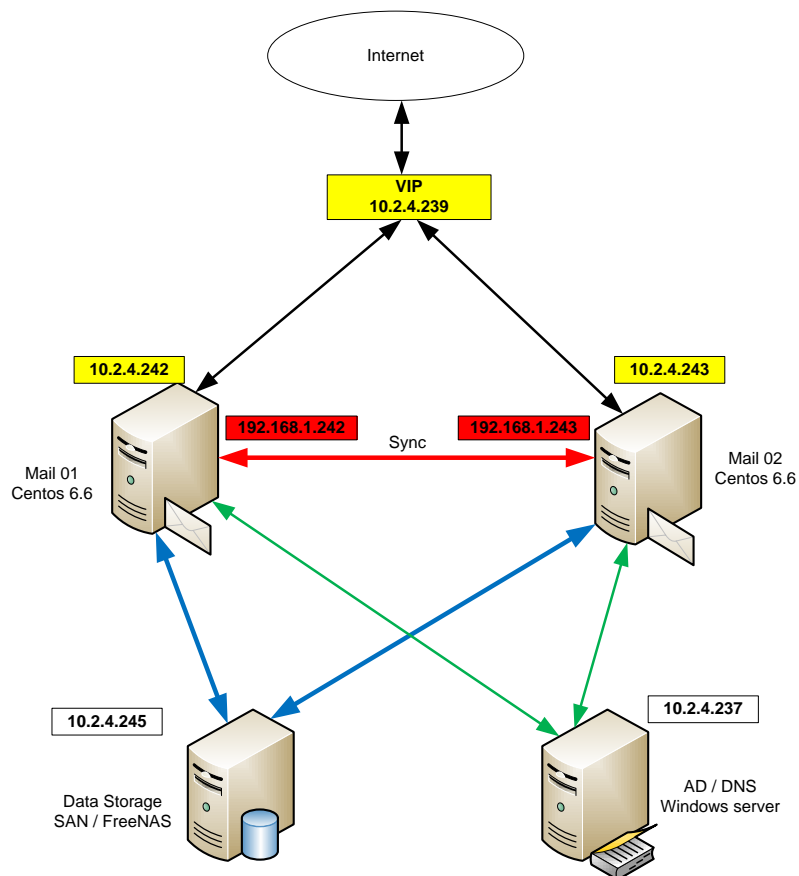
Trong phần này tôi sử dụng FreeNAS 9 để tạo Storage ảo phục vụ cho việc cấu hình thử nghiệm HA. Và sử dụng chung dải mạng local **10.2.4.x/25** chứ không thêm 1 card mạng thứ 3.

Thực hiện các bước từ 1 tới 7 như phần trên.

Mô hình:



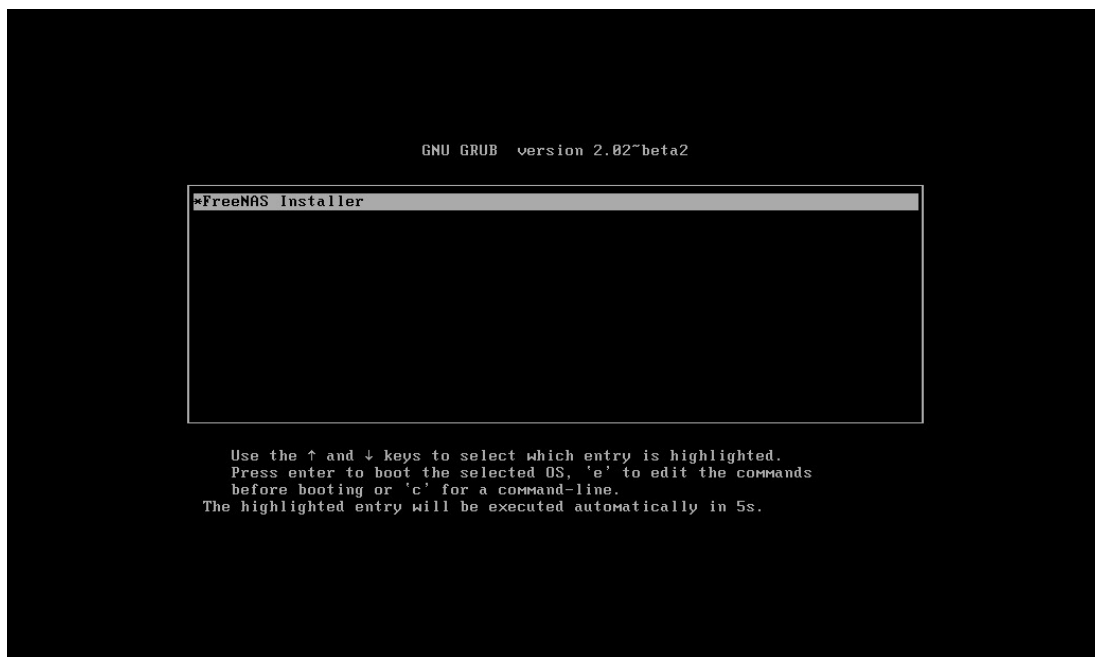
Thực tế trong bài mô phỏng này tôi thực hiện theo mô hình sau:

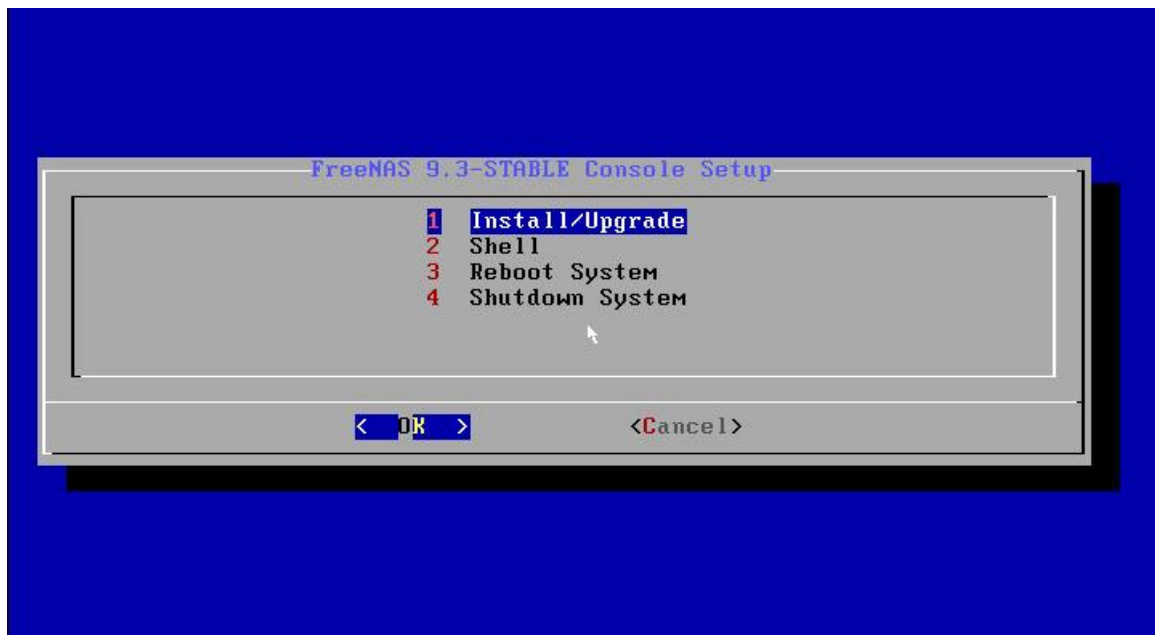


8.1 Setup FreeNAS 9

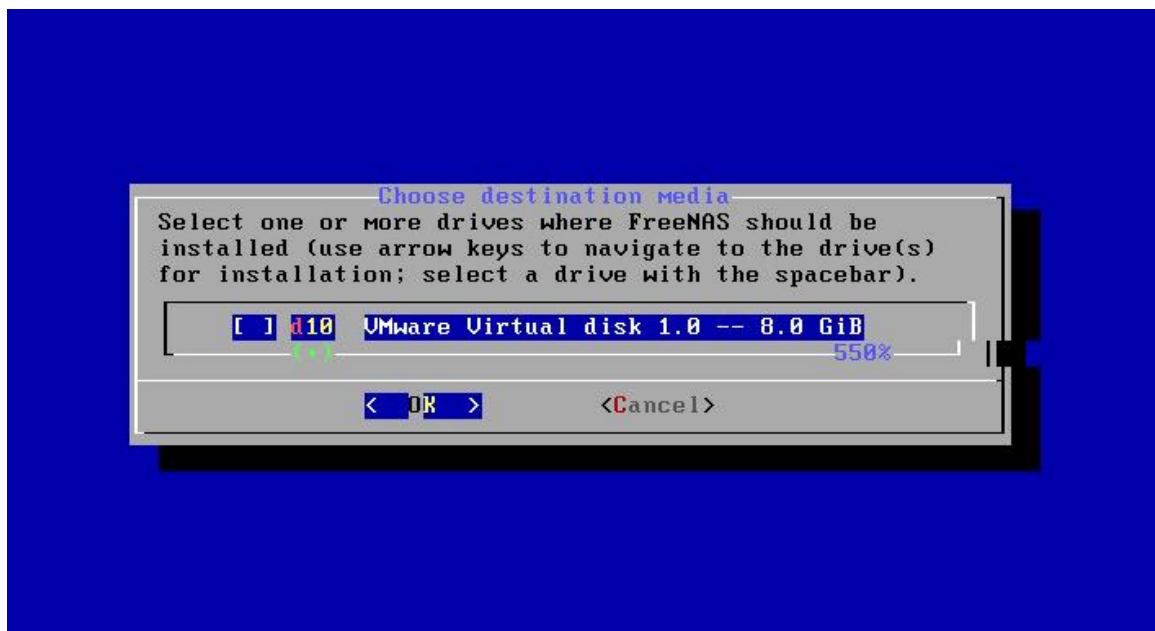
Tải bộ cài FreeNas 9 tại: <http://www.freenas.org/download-freenas-release.html> và burn ra USB để cài đặt

Sau khi boot bằng USB, hiển thị giao diện cài đặt

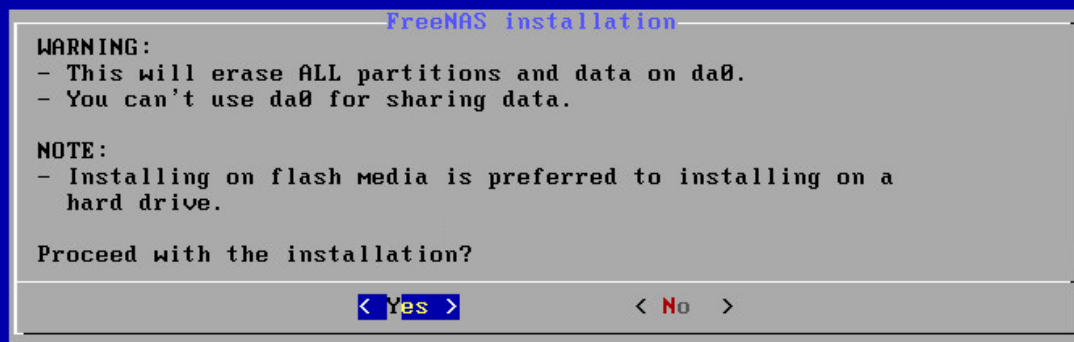




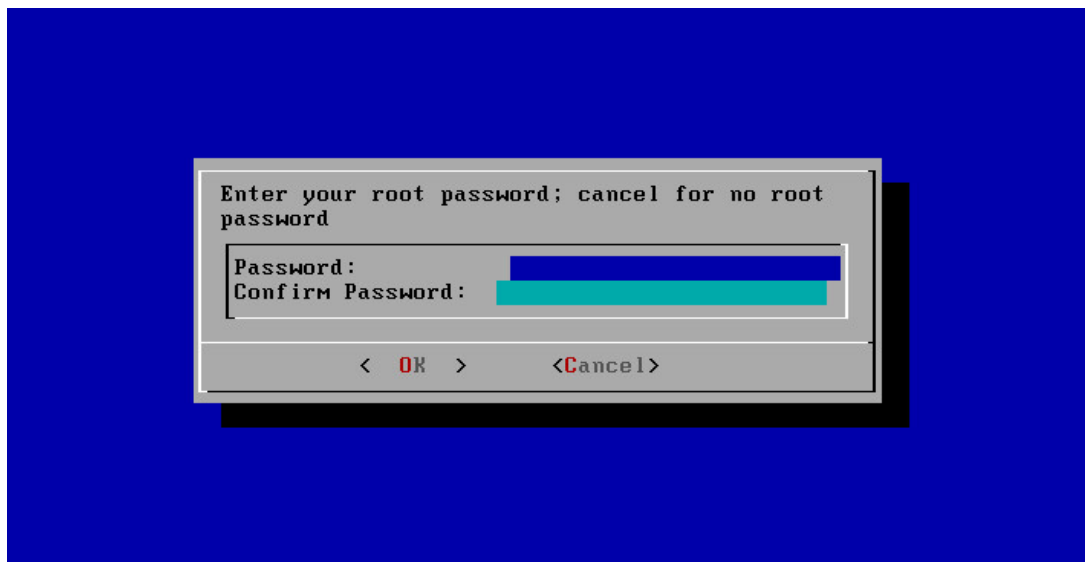
Chọn 1 để cài đặt FreeNas



Lựa chọn ổ để cài đặt FreeNas, nhấn Space để lựa chọn ổ muốn cài lên và OK



Chọn yes để tiếp tục



Đặt mật khẩu đăng nhập, và nhấn OK để tiếp tục cài đặt

A screenshot of a FreeNAS installation completion message box. The text inside the box reads: "The FreeNAS installation on da0 succeeded! Please remove the CDROM and reboot." Below the text is a button labeled "< OK >".

The FreeNAS installation on da0 succeeded!
Please remove the CDROM and reboot.

< OK >

Quá trình cài đặt xong, reboot lại

A screenshot of the FreeNAS 9.3-STABLE Console Setup menu. The title bar of the window says "FreeNAS 9.3-STABLE Console Setup". Inside the window, there is a list of four options: "1 Install/Upgrade", "2 Shell", "3 Reboot System", and "4 Shutdown System". The option "3 Reboot System" is highlighted with a blue background. At the bottom of the window, there are two buttons: "< OK >" and "<Cancel>".

FreeNAS 9.3-STABLE Console Setup

- 1 Install/Upgrade
- 2 Shell
- 3 Reboot System
- 4 Shutdown System

< OK >

<Cancel>

Chọn 3 để Reboot System

```
psm0: [GIANT-LOCKED]
psm0: model IntelliMouse, device ID 3
uart0: <16550 or compatible> port 0x3f8-0x3ff irq 4 flags 0x10 on acpi0
uart1: <16550 or compatible> port 0x2f8-0x2ff irq 3 on acpi0
fdc0: <floppy drive controller> port 0x3f0-0x3f5,0x3f7 irq 6 drq 2 on acpi0
fd0: <1440-KB 3.5" drive> on fdc0 drive 0
qpi0: <QPI system bus> on motherboard
orm0: <ISA Option ROMs> at iomem 0xc0000-0xc7fff,0xca000-0xcacff,0xdc000-0xdffff
,0xe0000-0xe7fff on isa0
sc0: <System console> at flags 0x100 on isa0
sc0: UGA <16 virtual consoles, flags=0x300>
vga0: <Generic ISA UGA> at port 0x3c0-0x3df iomem 0xa0000-0xbffff on isa0
wbwd0: HEFRAS and EFER do not align: EFER 0x2e DevID 0x0f DevRev 0x00 CR26 0x4f
acpi_throttle0: <ACPI CPU Throttling> on cpu0
coretemp0: <CPU On-Die Thermal Sensors> on cpu0
coretemp0: Tj(target) value 0 does not seem right.
ZFS NOTICE: Prefetch is disabled by default if less than 4GB of RAM is present;
to enable, add "vfs.zfs.prefetch_disable=0" to /boot/loader.conf.
ZFS filesystem version: 5
ZFS storage pool version: features support (5000)
Timecounters tick every 1.000 msec
ipfw2 (+ip6) initialized, divert enabled, nat enabled, default to accept, logging disabled
em0: link state changed to UP
█
```

Sau khi reboot, FreeNas sẽ tự động khởi động lại ở chế độ Normal, chờ khoảng 5 – 10 phút

```
Console setup
-----

1) Configure Network Interfaces
2) Configure Link Aggregation
3) Configure VLAN Interface
4) Configure Default Route
5) Configure Static Routes
6) Configure DNS
7) Reset Root Password
8) Reset to factory defaults
9) Shell
10) System Update (requires networking)
11) Create backup
12) Restore from a backup
13) Reboot
14) Shutdown

You may try the following URLs to access the web user interface:

http://10.2.4.157

Enter an option from 1-14: █
```

Giao diện Console sau khi cài đặt xong. Chọn 1 để đặt IP tĩnh


```

9) Shell
10) System Update (requires networking)
11) Create backup
12) Restore from a backup
13) Reboot
14) Shutdown

You may try the following URLs to access the web user interface:

http://172.24.104.52

Enter an option from 1-14: 1
1) em0
Select an interface (q to quit): 1
Reset network configuration? (y/n) n
Configure interface for DHCP? (y/n) n
Configure IPv4? (y/n) y
Interface name: eth0
Several input formats are supported
Example 1 CIDR Notation:
  192.168.1.1/24
Example 2 IP and Netmask separate:
  IP: 192.168.1.1
  Netmask: 255.255.255.0, /24 or 24
IPv4 Address: 10.2.4.250

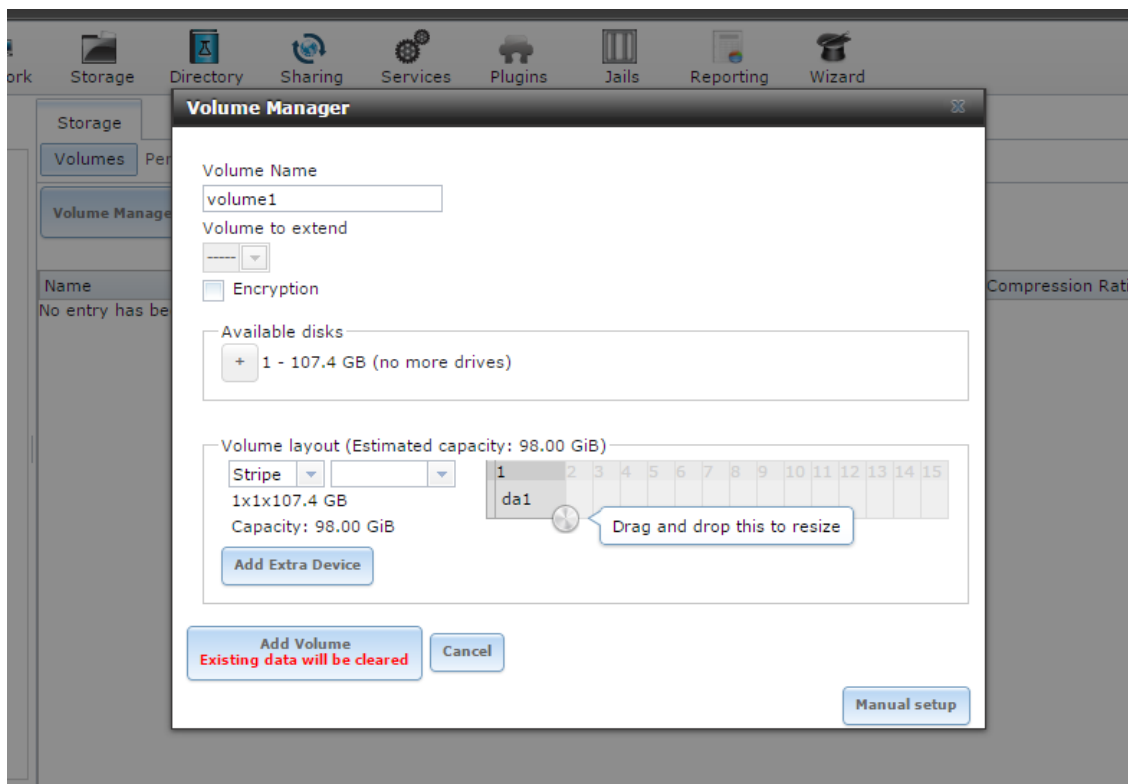
```

Sau khi đặt IP, có thể đăng nhập qua giao diện web để cấu hình cho FreeNas

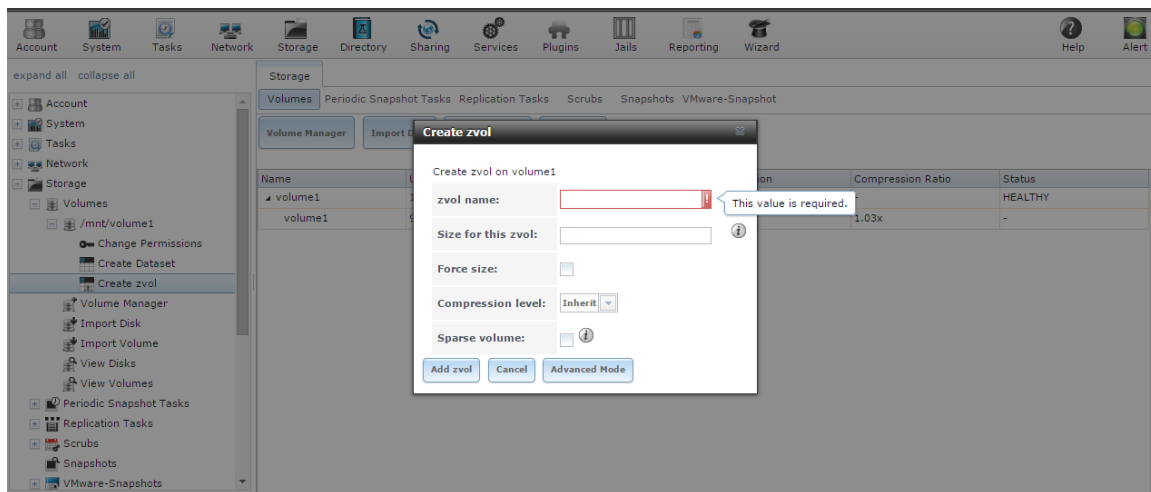
8.2 Config FreeNAS 9

Đăng nhập vào bằng username = root. Lần đầu đăng nhập sẽ có phần cài đặt wizard, thực hiện các bước theo thứ tự:

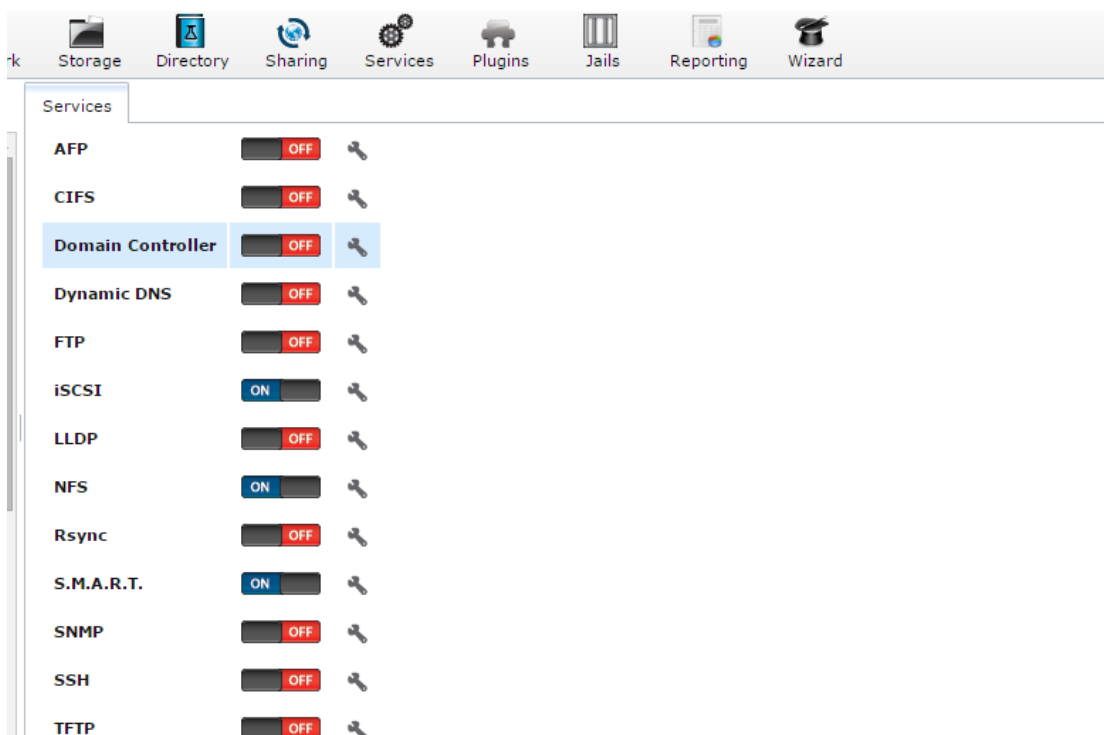
Để tạo Storage, vào phần Storage, chọn Volume manager



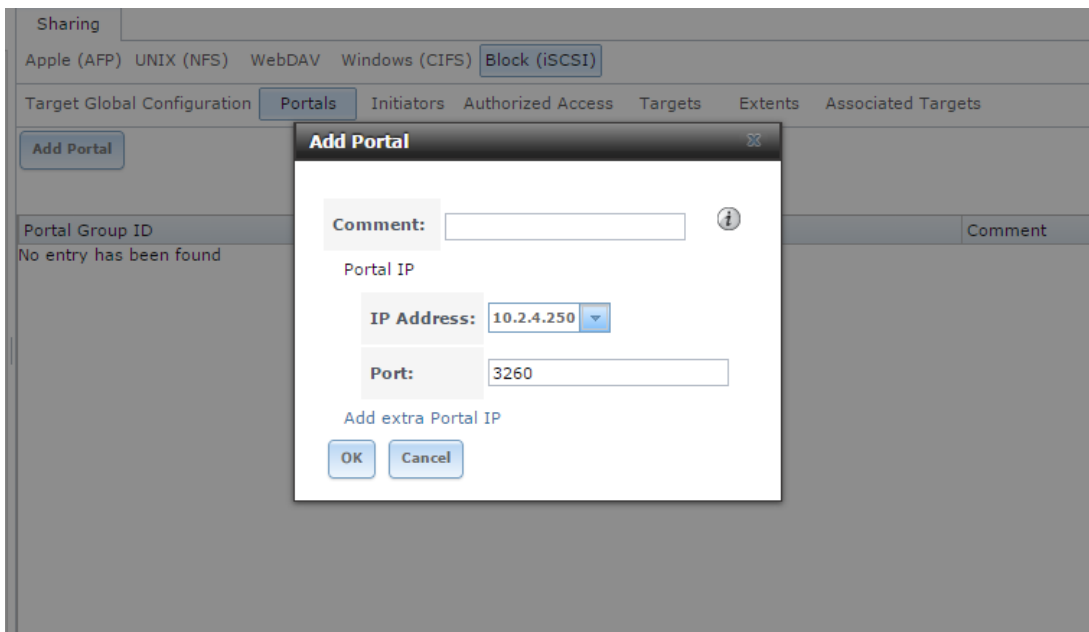
Tạo zvol



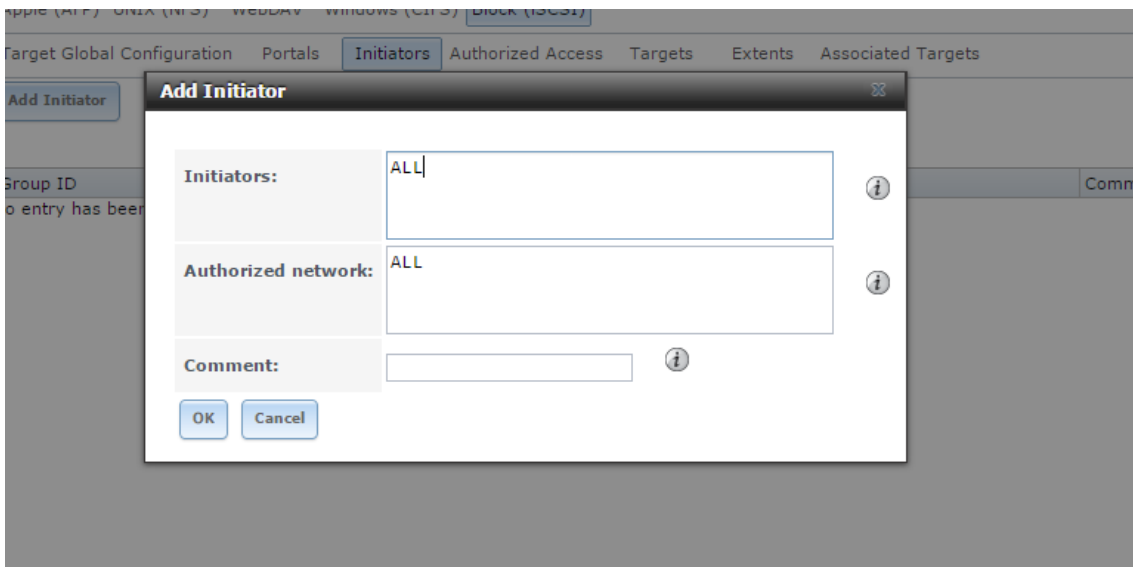
Vào Services để turn on iSCSI và NFS



Vào Sharing để tiếp tục cấu hình, chọn Block (iSCSI) -> portal -> Add Portal



Add Initiators



Add Target

Add Target

Target Name: data

Target Alias:

Serial: 005056b31ef800

Portal Group ID: 1

Initiator Group ID: 1

Auth Method: None

Authentication Group number: None

OK Cancel

Add Extent

Add Extent

Extent Name: data

Extent Type: Device

Device: volume1/data (90.0 GiB)

Logical Block Size: 512

Disable Physical Block Size Reporting:

Available Space Threshold (%):

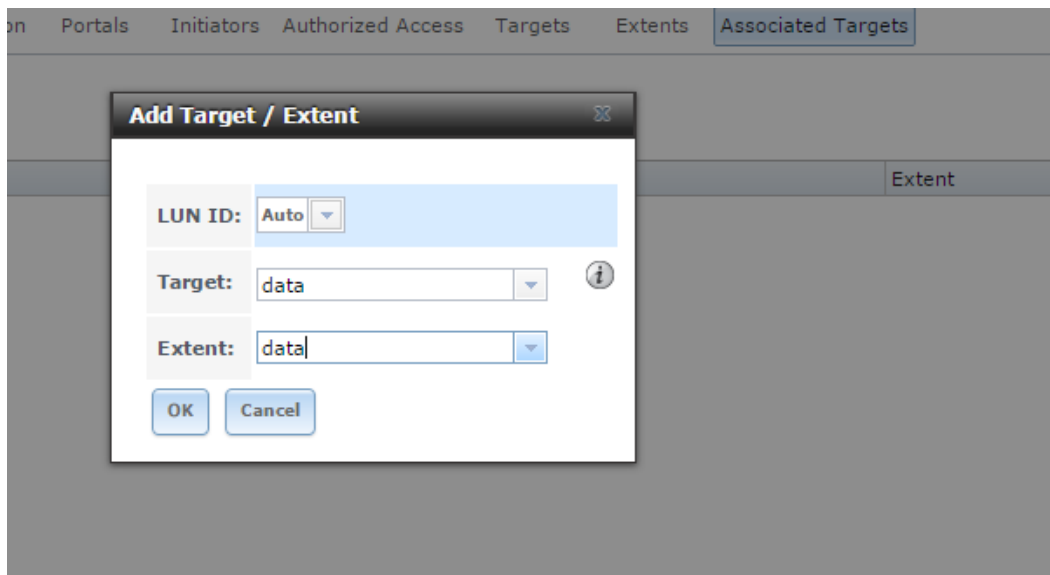
Comment:

Enable TPC:

Xen initiator compat mode:

OK Cancel

Add Target/Extent



8.3 Cấu hình 2 mail server kết nối tới storage

Ở trên 2 máy tiến hành cài đặt gói **iscsi initiators**

```
[root@mail1 ~]# yum install iscsi-initiator-utils
```

8.4 Kết nối 2 mailserver vào Storage

Trên 2 máy thực hiện Discovery và login vào trong Storage:

```
[root@mail1 ~]# iscsiadm -m discovery -t sendtargets -p
10.2.4.245
10.2.4.245:3260,1 iqn.2015-02:data
```

Sau khi tìm được sẽ trả ra kết quả các ổ storage tìm được.

Tiến hành kết nối storage vào mailserver (tiến hành trên cả 2 server mail)

```
[root@mail1 ~]# iscsiadm -m node -T iqn.2015-02:data -p
10.2.4.245 --login
Logging in to [iface: default, target: iqn.2015-02:data,
portal: 10.2.4.245,3260]
Login to [iface: default, target: iqn.2015-02:data, portal:
10.2.4.245,3260]: successful
```

Kiểm tra lại trên 2 máy sẽ xuất hiện thêm phân vùng ổ đĩa mới là **sdb** hoặc **sdc**

```
[root@mail1 ~]# ls /dev/sd*
```

```
/dev/sda /dev/sda1 /dev/sda2 /dev/sdb /dev/sdc
```

Tiến hành định dạng phân vùng mới **sdc**

```
[root@mail1 ~]# fdisk /dev/sdc
```

```
Command (m for help): n
```

```
Command action
```

```
  e   extended
```

```
  p   primary partition (1-4)
```

```
p
```

```
Partition number (1-4): 1
```

```
First cylinder (1-36864, default 1): 1
```

```
Last cylinder or +size or +sizeM or +sizeK (1-36864, default 36864): 36864
```

```
Command (m for help): p
```

```
Disk /dev/iscsi/linux3-data-1/part: 38.6 GB, 38654705664 bytes
```

```
64 heads, 32 sectors/track, 36864 cylinders
```

```
Units = cylinders of 2048 * 512 = 1048576 bytes
```

	Device	Boot	Start	End	Blocks	Id
System						
/dev/iscsi/linux3-data-1/part1			1	36864	37748720	83
Linux						

```
Command (m for help): w
```

```
The partition table has been altered!
```

```
Calling ioctl() to re-read partition table.
```

```
Syncing disks.
```

Format Device (chỉ làm trên máy active có thể làm trên cả 2 cũng được, nhưng làm trên máy active sau, máy standby trước)

```
[root@mail1 ~]# mkfs.ext4 /dev/sdc1
```

```
mke2fs 1.39 (29-May-2006)
```

```
Filesystem label=
```

```
OS type: Linux
```

```
Block size=4096 (log=2)
```

```
Fragment size=4096 (log=2)
```

```
4718592 inodes, 9437180 blocks
```

```
471859 blocks (5.00%) reserved for the super user
```

```
First data block=0
```

```
Maximum filesystem blocks=0
```

```
288 block groups
```

```
32768 blocks per group, 32768 fragments per group
```

```
16384 inodes per group
```

```
Superblock backups stored on blocks:
```

```
    32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632, 2654208,
    4096000, 7962624
```

```
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.
```

Trên 2 máy tạo directory để mount vào storage (tùy theo tên đặt)

Ở đây tôi đặt là /mail/data

```
[root@mail1 ~]# mkdir -p /mail/data
[root@mail1 ~]# chown zimbra:zimbra /mail/data
[root@mail1 ~]# chmod 775 /mail/data
[root@mail1 ~]# mkdir /mail/data/index
[root@mail1 ~]# mkdir /mail/data/store
[root@mail1 ~]# chown zimbra:zimbra /mail/data/index
/mail/data/store
[root@mail1 ~]# chmod 775 /mail/data/index /mail/data/store
```

Sau khi tạo 2 thư mục ở trên 2 máy xong. Tiến hành tắt 1 máy standby đi, thực hiện ở trên máy chủ động.

8.5 Tạo nơi thư mục chứa Index và Store mới

Máy 2 standby được tắt hẳn đi.

```
[root@mail1 ~]# service zimbra stop
[root@mail1 ~]# mv /opt/zimbra/index/* /mail/data/index/
[root@mail1 ~]# mv /opt/zimbra/store/* /mail/data/store/
```

Xóa thư mục Store và index trong mailserver đi.

```
[root@mail1 ~]# rmdir /opt/zimbra/store
[root@mail1 ~]# rmdir /opt/zimbra/index
```

Tạo liên kết mềm (shortcut) của thư mục chứa data mới trong thư mục mặc định của zimbra

```
[root@mail1 ~]# ln -s /mail/data/index /opt/zimbra/index
[root@mail1 ~]# ln -s /mail/data/store /opt/zimbra/store
```

Khởi động lại zimbra

```
[root@mail1 ~]# service zimbra start
```

Truy cập vào admin web console , Configuration > Servers > server-name > Volumes Tab.

Select index volume > Click edit và chỉnh sửa đường dẫn theo đường dẫn mới.

Select store volume > Click edit và chỉnh sửa đường dẫn theo đường dẫn mới.

Click on Save in the upper left hand section.

Hoặc thực hiện sửa đường dẫn bằng câu lệnh trong CLI:

```
[root@mail1 ~]# zmlocalconfig -e
zimbra_store_directory=/mail/data/store

[root@mail1 ~]# zmlocalconfig -e
zimbra_index_directory=/mail/data/index
```

To Modify Volume From CLI After Data Move

See what the details are of your volumes:

```
[root@mail1 ~]# zmvolume -l /opt/zimbra/index
```

và

```
[root@mail1 ~]# zmvolume -l /opt/zimbra/store
```

Volume id: 2

name: index1

type: index

path: /opt/zimbra/index

compressed: false

current: true

Volume id: 1

name: message1

type: primaryMessage

path: /opt/zimbra/store

compressed: false

current: true

Có thể thấy kết quả trả về đường dẫn mail store và index cũ là:

/opt/zimbra/index /opt/zimbra/store

Và directory paths mới của chúng ta là:

/mail/data/index /mail/data/store

Cần phải chỉnh sửa nó về đúng đường dẫn mới.

```
[root@mail1 ~]# zmvolume -e -id 2 -p /mail/data/index
```

```
[root@mail1 ~]# zmvolume -e -id 1 -p /mail/data/store
```

Check lại:

```
[root@mail1 ~]# zmvolume -l /opt/zimbra/index
```

```
[root@mail1 ~]# zmvolume -l /opt/zimbra/store
```

Khởi động lại dịch vụ. Bật lại máy 2.

Check dịch vụ drbd:

```
[root@mail1 ~]# service drbd status
```

Nếu thấy lỗi drbd thì chỉnh sửa.

version: 8.3.0 (api:88/proto:86-89)

GIT-hash: 9ba8b93e24d842f0dd3fb1f9b90e8348ddb95829 build by

argus@docidtxt04, 2009-03-04 16:23:58

0: cs:WfConnection ro:Secondary/Unknown ds:UpToDate/DUnknown A r---

ns:0 nr:0 dw:0 dr:0 al:0 bm:14 lo:0 pe:0 ua:0 ap:0 ep:1 wo:b oos:55544

Check lại dịch vụ drbd trên cả 2 máy.

```
[root@mail1 ~]# service drbd status
```

version: 8.3.0 (api:88/proto:86-89)

GIT-hash: 9ba8b93e24d842f0dd3fb1f9b90e8348ddb95829 build by

argus@docidtxt04, 2009-03-04 16:23:58

0: cs:WfConnection ro:Secondary/Unknown ds:UpToDate/DUnknown A r---

ns:0 nr:0 dw:0 dr:0 al:0 bm:14 lo:0 pe:0 ua:0 ap:0 ep:1 wo:b oos:55544

Chọn máy 1 là máy chính (vì đang cấu hình từ đầu).

On both node:

```
[root@mail1 ~]# service corosync stop
```

```
[root@mail1 ~]# service pacemaker stop
```

```
[root@mail1 ~]# service zimbra stop
[root@mail1 ~]# service drbd stop
[root@mail1 ~]# service drbd start
```

Repair:

chọn primary ở node “good” data, ở đây là máy 1

```
[root@mail1 ~]# drbdadm primary r0
```

On node Secondary /Unknown: (on node with "bad" data)

```
[root@mail2 ~]# drbdadm -- --discard-my-data connect all
```

On node Primary/Unknown: (on node with "good" data)

```
[root@mail1 ~]# drbdadm connect all
```

8.6 Tạo HA kết nối tới Storage

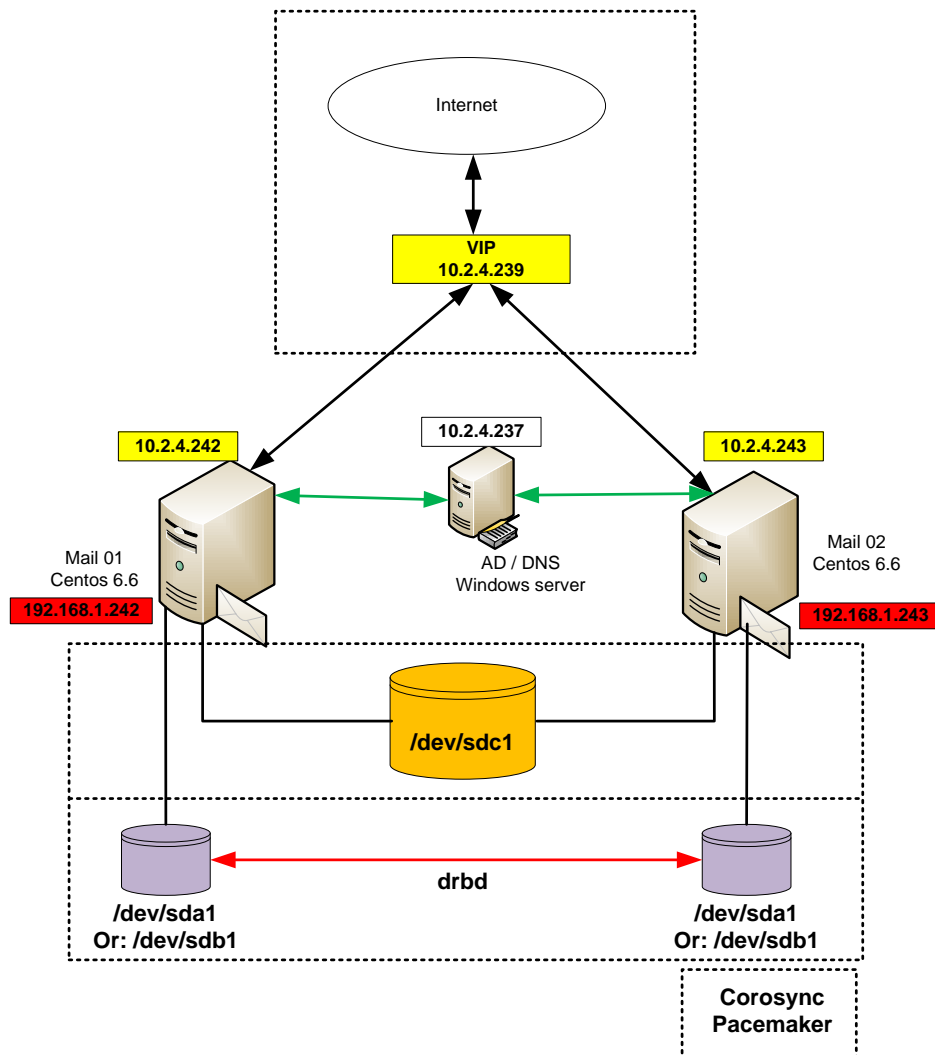
Then, in **crm configure**, inject the following configuration:

```
primitive fs-zmdata ocf:heartbeat:Filesystem \
params device="/dev/sdb1" directory="/mail/data" fstype="ext4"
```

Therefore, edit with the “**edit**” command the config file to add our new resource to the group, before validating with “**commit**”:

```
group zimbra-group FAILOVER-ADDR fs-zimbra zcs_service fs-zmdata
```

Lúc này sơ đồ đồng bộ dữ liệu sẽ như sau:



Phần data sẽ được lưu trong storage **/dev/sdc1**

Phần System hệ thống nằm trên **/dev/sda** hoặc **/dev/sdb** và được đồng bộ với nhau bằng drbd. Việc này không chiếm quá nhiều dung lượng như việc đồng bộ trực tiếp data của 2 phương án 1 trên.

9. Open Cluster Framework Script

```
#!/bin/sh
#
# Resource script for Zimbra
#
# Description: Manages Zimbra as an OCF resource in
# an high-availability setup.
#
# Author: Adrian Gibanel
# <adrian.gibanel@btactic.com> : Original Author
#
# License: GNU General Public License (GPL)
# Note: Aimed at an active/passive cluster originally
# Inspired from postfix OCF script
# Inspired from Ubuntu LSB script.
# Not sure it will work for other distros without modifying
#
#
# usage: $0 {start|stop|reload|status|monitor|validate-all|meta-data}
#
# The "start" arg starts Zimbra
#
# The "stop" arg stops it.
#
# OCF parameters:
# OCF_RESKEY_binary
# OCF_RESKEY_config_dir
# OCF_RESKEY_parameters
#
#####
# Initialization:
: ${OCF_FUNCTIONS_DIR:=${OCF_ROOT}/lib/heartbeat}
. ${OCF_FUNCTIONS_DIR}/ocf-shellfuncs
: ${OCF_RESKEY_binary="zmcontrol"}
: ${OCF_RESKEY_zimbra_dir="/opt/zimbra"}
: ${OCF_RESKEY_zimbra_user="zimbra"}
: ${OCF_RESKEY_zimbra_group="zimbra"}
USAGE="Usage: $0 {start|stop|restart|status|monitor|validate-all|meta-data}";
#####
usage() {
echo $USAGE >&2
}
meta_data() {
cat <<END
<?xml version="1.0"?>
<!DOCTYPE resource-agent SYSTEM "ra-api-1.dtd">
<resource-agent name="zimbra">
```

```

<version>0.1</version>
<longdesc lang="en">
This script manages Zimbra as an OCF resource in a high-availability setup.
</longdesc>
<shortdesc lang="en">
Manages a highly available Zimbra mail server instance
</shortdesc>
<parameters>
<parameter name="binary" unique="0" required="0">
<longdesc lang="en">
Short name to the Zimbra control script.
For example, "zmcontrol".
</longdesc>
<shortdesc lang="en">
Short name to the Zimbra control script</shortdesc>
<content type="string" default="zmcontrol" />
</parameter>
<parameter name="zimbra_dir" unique="1" required="0">
<longdesc lang="en">
Full path to Zimbra directory.
For example, "/opt/zimbra".
</longdesc>
<shortdesc lang="en">
Full path to Zimbra directory</shortdesc>
<content type="string" default="/opt/zimbra" />
</parameter>
<parameter name="zimbra_user" unique="1" required="0">
<longdesc lang="en">
Zimbra username.
For example, "zimbra".
</longdesc>
<shortdesc lang="en">Zimbra username</shortdesc>
<content type="string" default="zimbra" />
</parameter>
<parameter name="zimbra_group"
unique="1" required="0">
<longdesc lang="en">
Zimbra group.
For example, "zimbra".
</longdesc>
<shortdesc lang="en">Zimbra group</shortdesc>
<content type="string" default="zimbra" />
</parameter>
</parameters>
<actions>
<action name="start" timeout="360s" />
<action name="stop" timeout="360s" />
<action name="restart" timeout="360s" />

```

```

<action name="monitor" depth="0" timeout="40s"
interval="60s" />
<action name="validate-all" timeout="360s" />
<action name="meta-data" timeout="5s" />
</actions>
</resource-agent>
END
}
command()
{
if [ -f ${zimbra_dir}/redolog/redo.log ]; then
chown -f ${zimbra_user}:${zimbra_group} \
${zimbra_dir}/redolog/redo.log
fi
su - ${zimbra_user} -c "${binary} $1 </dev/null"
}
running() {
# run Zimbra status
command status
}
zimbra_status()
{
running
}
zimbra_start()
{
# if Zimbra is running return success
if zimbra_status; then
ocf_log info "Zimbra is already running."
return $OCF_SUCCESS
fi
# start Zimbra
command start
ret=$?
if [ -d /var/lock/subsys -a $ret -eq 0 ]; then
touch /var/lock/subsys/zimbra
fi
if [ $ret -ne 0 ]; then
ocf_log err "Zimbra returned an error." $ret
return $OCF_ERR_GENERIC
fi
# grant some time for
# startup/forking the sub processes
sleep 2
# initial monitoring action
running
ret=$?
if [ $ret -ne $OCF_SUCCESS ]; then

```

```

ocf_log err "Zimbra failed initial monitor action." $ret
return $OCF_ERR_GENERIC
fi
ocf_log info "Zimbra started."
return $OCF_SUCCESS
}
zimbra_stop()
{
# if Zimbra is not running return success
if ! zimbra_status; then
ocf_log info "Zimbra already stopped."
return $OCF_SUCCESS
fi
# stop Zimbra
command stop
ret=$?
if [ -d /var/lock/subsys -a $ret -eq 0 ]; then
rm -f /var/lock/subsys/zimbra
fi
if [ $ret -ne 0 ]; then
ocf_log err "Zimbra returned an error while stopping." $ret
return $OCF_ERR_GENERIC
fi
# grant some time for shutdown and recheck 5 times
for i in 1 2 3 4 5; do
if zimbra_status; then
sleep 1
fi
done
# escalate to abort if we did not stop by now
# @TODO shall we loop here too?
if zimbra_status; then
ocf_log err "Zimbra failed to stop. Escalating to 'abort'."
ORPHANED=`ps -u ${zimbra_user} -o "pid="` && kill -9 $ORPHANED 2>&1
ret=$?
sleep 10
# zimbra abort did not succeed
if zimbra_status; then
ocf_log err "Zimbra failed to abort."
return $OCF_ERR_GENERIC
fi
fi
ocf_log info "Zimbra stopped."
return $OCF_SUCCESS
}
zimbra_restart()
{
if zimbra_status; then

```

```

ocf_log info "Reloading Zimbra."
command restart
fi
}
zimbra_monitor()
{
if zimbra_status; then
return $OCF_SUCCESS
fi
return $OCF_NOT_RUNNING
}
zimbra_validate_all()
{
# check zimbra_dir parameter
if [ ! -d "$zimbra_dir" ]; then
ocf_log err "Zimbra directory '$config_dir' does not exist." $ret
return $OCF_ERR_INSTALLED
fi
# check that the Zimbra binaries
# exist and can be executed
if ! have_binary "${zimbra_dir}/bin/${binary}"; then
return $OCF_ERR_INSTALLED
fi
# check permissions
user=${zimbra_user}
zimbra_writable_dirs="${zimbra_dir}/conf"
for dir in "$zimbra_writable_dirs"; do
if ! su -s /bin/sh - $user -c "test -w $dir"; then
ocf_log err "Directory '$dir' is not writable by user '$user'."
exit $OCF_ERR_PERM;
fi
done
return $OCF_SUCCESS
}
#
# Main
#
if [ $# -ne 1 ]; then
usage
exit $OCF_ERR_ARGS
fi
binary=$OCF_RESKEY_binary
zimbra_dir=$OCF_RESKEY_zimbra_dir
zimbra_user=$OCF_RESKEY_zimbra_user
zimbra_group=$OCF_RESKEY_zimbra_group
parameters=$OCF_RESKEY_parameters
# debugging stuff
#echo OCF_RESKEY_binary=$OCF_RESKEY_binary \

```



```
# >> /tmp/prox_conf_${OCF_RESOURCE_INSTANCE}
#echo OCF_RESKEY_binary=${OCF_RESKEY_zimbra_dir} \
#>> /tmp/prox_conf_${OCF_RESKEY_zimbra_dir}
#echo OCF_RESKEY_binary=${OCF_RESKEY_zimbra_user} \
#>> /tmp/prox_conf_${OCF_RESKEY_zimbra_user}
#echo OCF_RESKEY_binary=${OCF_RESKEY_zimbra_group} \
#>> /tmp/prox_conf_${OCF_RESKEY_zimbra_group}
#echo OCF_RESKEY_binary=${OCF_RESKEY_parameters} \
#>> /tmp/prox_conf_${OCF_RESKEY_parameters}
# build Zimbra options string
# *outside* to access from each method
OPTIONS="
OPTION_CONFIG_DIR="
# check if the Zimbra config_dir exist
if [ "x$config_dir" != "x" ]; then
# check for postconf binary
#check_binary "${zimbra_dir}/bin/${binary}"
# remove all trailing slashes
zimbra_dir=`echo $zimbra_dir | sed 's/\/*$//'^
fi
case $1 in
meta-data) meta_data
exit ${OCF_SUCCESS}
;;
usage|help) usage
exit ${OCF_SUCCESS}
;;
esac
zimbra_validate_all
ret=$?
#echo "debug[$1:$ret]"
LSB_STATUS_STOPPED=3
if [ $ret -ne ${OCF_SUCCESS} ]; then
case $1 in
stop) exit ${OCF_SUCCESS} ;;
monitor) exit ${OCF_NOT_RUNNING};;
status) exit ${LSB_STATUS_STOPPED};;
*) exit $ret;;
esac
fi
case $1 in
monitor) zimbra_monitor
exit $?
;;
start) zimbra_start
exit $?
;;
stop) zimbra_stop
```

```
exit $?
;;
restart) zimbra_restart
exit $?
;;
status) if zimbra_status; then
ocf_log info "Zimbra is running."
exit $OCF_SUCCESS
else
ocf_log info "Zimbra is stopped."
exit $OCF_NOT_RUNNING
fi
;;
validate-all) exit $OCF_SUCCESS
;;
*) usage
exit $OCF_ERR_UNIMPLEMENTED
;;
esac
```

10. Tài liệu tham khảo

<http://www.tokiwinter.com/clustering-with-drbd-corosync-and-pacemaker/>

<http://zeldor.biz/2010/12/activepassive-cluster-with-pacemaker-corosync/>

http://www.server-world.info/en/note?os=CentOS_6&p=pacemaker

<http://geekpeek.net/linux-cluster-resources/>

<http://albertolarripa.com/2012/05/13/install-configure-mysql-cluster/>

http://docs.openstack.org/high-availability-guide/content/_install_packages.html

<http://www.slideshare.net/pawanitrooman/drbd-corosyncpacemakercluster>

pdf file: How to HA zimbra 8

High Availability ZCS FOSS Implementation.docx