Technical Notes

Implementation of a high availability solution based on Free Libre Open Source Software tools for Zimbra Collaboration System

Daniel H. Gamez V.
daniel.gamez@gmail.com
Venezuela, November 2014

Table of Contents

List of Abbreviations and Symbols

\	Symbol that implies a continuous line in bash commands
CIB	Cluster Information Base
CRM	Cluster Resource Manager
DNS	Domain Name Server
DRBD	Distributed Replicated Block Device
FLOSS	Free Libre Open Source Software
LAN	Local Area Network
NTP	Network Time Protocol
OCF	Open Clustering Framework
OS	Operating System
RHEL	Red Hat Enterprise Linux
RPM	RPM Package Manager
STONITH	Shoot The Other Node In The Head
ZCS	Zimbra Collaboration System

1 Introduction

This document is intended to provide technical documentation in the process of implementing high availability in a Free Libre Open Source Software (FLOSS) Zimbra Collaboration System (ZCS).

The scope of this documentation is limited to the following software components and versions:

- Red Hat Enterprise Linux Server release 6.5 (Santiago)
- GNU/Linux 2.6.32-431.el6.x86_64
- zcs 8.0.7_GA_6021.RHEL6_64 FOSS edition
- drbd 8.4.3-33
- corosync 1.4.5-2.2
- pacemaker 1.1.10-14
- pcs 0.9.90-2
- crmsh 1.2.5-0
- ccs 0.16.2-69
- cman 3.0.12.1-59

The defined cluster consists of two nodes which will be referenced as ***astapor*** and ***braavos*** in the domain got.com (as in Game of Thrones). These nodes are virtual machines hosted on two Proxmox Virtual Environment servers based on KVM virtualization, which are installed on separate physical machines in the same LAN to avoid single point of failure. The proposed scheme is similar to the observed in Figure 1 (took from http://www.sherin.co.in/wp-content/uploads/2010/06/drdb_hearbeat.png).

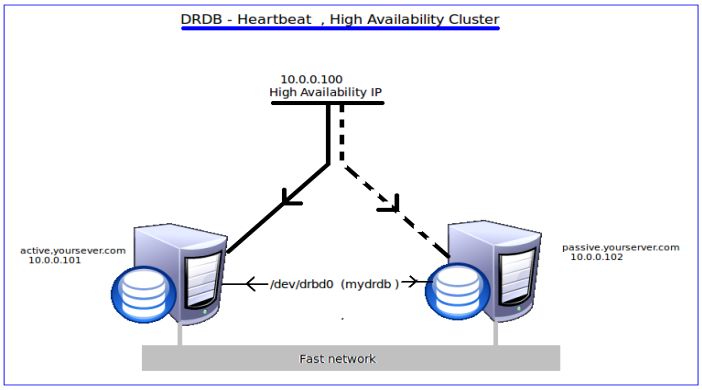

Figure 1. HA Scheme

2 Operating system

The configuration is the same in both nodes:

RHEL 6.5 x86_64

Disk Partitions:

10gb	/	
100mb	/boot	
8gb	/opt/zimbra	(/dev/vdb1) *
150mb	drbd meta-data	(/dev/vdc1) *

CPU: 1

RAM: 2gb

* It is not necessary to format partitions for devices vdb1 or vdc1 during OS install.

2.1 FQDN hostnames and IP addresses

Split DNS IP:	172.17.18.190	zcs-ha.got.com
Astapor:	172.17.18.191	astapor.got.com
Braavos:	172.17.18.192	braavos.got.com

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

```
127.0.0.1       localhost.localdomain   localhost
172.17.18.190   zcs-ha.got.com zcs-ha
172.17.18.191   astapor.got.com astapor
172.17.18.192   braavos.got.com braavos
```

A useful command to handle hostname changes in RHEL:

service hostname restart

2.2 Network

2.2.1 IP

Set the proper network parameters in ifcfg-eth0 file on each server, for instance:

/etc/sysconfig/network-scripts/ifcfg-eth0

Astapor	Braavos
DEVICE=eth0	DEVICE=eth0
HWADDR=26:34:99:65:d7:77	HWADDR=26:34:99:65:d7:78
TYPE=Ethernet	TYPE=Ethernet
ONBOOT=yes	ONBOOT=yes
NM_CONTROLLED=no	NM_CONTROLLED=no
BOOTPROTO=none	BOOTPROTO=none
IPADDR=172.17.18.191	IPADDR=172.17.18.192
NETMASK=255.255.255.0	NETMASK=255.255.255.0
GATEWAY=172.17.18.1	GATEWAY=172.17.18.1
DNS1=127.0.0.1	DNS1=127.0.0.1
IPV6INIT=no	IPV6INIT=no
USERCTL=no	USERCTL=no

Set the correct Netmask and Gateway, so servers are able to reach internet addresses, also disable the firewall or allow the http and ftp outgoing rules on it. The primary DNS server will be configured later to be the localhost, with forwarding to external DNS servers.

Some useful commands to manipulate and consult the network service on RHEL:

```
service network restart
/etc/init.d/network restart
ifconfig eth0 down; ifconfig eth0 up
ifdown eth0; ifup eth0
ifconfig
ip addr show
```

2.2.2 NTP

RPM packages that would be necessary: ntp, ntpdate.

Set the proper NTP parameters in /etc/ntp.conf file on each server, so both nodes share the same date and time, for instance:

```
driftfile /var/lib/ntp/drift
restrict default kod nomodify notrap nopeer noquery
restrict 127.0.0.1
server 172.17.18.1
includefile /etc/ntp/crypto/pw
keys /etc/ntp/keys
```

Some useful commands to manipulate and consult NTP service on RHEL:

```
service ntpd restart
ntpstat
ntpq -pn
date
```

2.2.3 BIND

RPM necessary packages: bind, bind-utils.

A primary DNS server configured on each server is crucial, or alternatively another centralized DNS server on the LAN with the whole configuration. Here is considered the first option.

In /etc/named.conf file is added the following:

```
zone "got.com." IN {
        type master;
        file "got.com.db";
};
```

In astapor node, /var/named/got.com.db file contains:

```
                    IN    1H    NS        zcs-ha.got.com.
                    IN    1H    MX         5   zcs-ha.got.com.
zcs-ha              IN    1H    A        172.17.18.190
astapor             IN    1H    A        172.17.18.191
astapor.got.com IN          CNAME    zcs-ha.got.com.
```

And a similar got.com.db file must be set on braavos node *replacing* the corresponding *hostname* and *IP address*. Leave zcs-ha entries without change in both nodes.

Some useful commands to manipulate and consult BIND service on RHEL:

```
named-checkconf -z
service named restart
service named status
dig -t ANY got.com
nslookup astapor.got.com
```

2.3 ZCS dependencies

As requirement for ZCS, the following RPM packages must be installed in the OS: nc, sudo, libidn, gmp, libaio.

Some other suggested RPM packages are: perl-5.10.1, sysstat, sqlite.

The postfix service must be turned off and excluded from boot start-up:

```
service postfix stop
chkconfig postfix off
```

3 DRBD

The Distributed Replicated Block Device (DRBD) provides a mirrored storage required for the HA environment.

3.1 Initial configuration

The following actions must be performed in parallel on ***both*** nodes, except in those cases where otherwise specified.

- Ensure to adapt hostname to 'astapor' on the primary node and 'braavos' on the secondary node.

- Install RPM packages:

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

  drbd-8.4.3-33.el6.x86_64

- Leave /etc/drbd.conf and /etc/drbd.d/global_common.conf files by default.

- Add /etc/drbd.d/optzimbra.res file with the following content:

```
resource optzimbra {
 protocol C;

 handlers {
  pri-on-incon-degr "halt -f";
 }
 startup {
  degr-wfc-timeout 120; # 2 minutes
 }
```

```
        disk {
         on-io-error detach;
        }
        net {
        }
        syncer {
         rate 10M;
         al-extents 257;
        }
        on astapor.got.com {
         device /dev/drbd0;
         disk /dev/vdb1;
         address 172.17.18.191:7788;
         flexible-meta-disk /dev/vdc1;
        }
        on braavos.got.com {
         device /dev/drbd0;
         disk /dev/vdb1;
         address 172.17.18.192:7788;
         flexible-meta-disk /dev/vdc1;
        }
       }
```

- 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:

  dd if=/dev/zero of=/dev/vdb1 bs=1K count=100

  dd if=/dev/zero of=/dev/vdc1 bs=1K count=100

- Start DRBD module:

  modprobe drbd

- Create resource:

  drbdadm create-md optzimbra

- Execute first DRBD synchronisation on astapor:

  drbdadm up optzimbra

  drbdadm primary --force optzimbra

  drbdadm --discard-my-data connect optzimbra

- It is possible to check synchronisation status with:
  watch cat /proc/drbd

- Final output will show:
  ds:UpToDate/UpToDate

- Verify current roles:
  drbdadm role optzimbra
  It will show 'Primary/Secondary' on astapor
  and 'Secondary/Primary' on braavos node.

- Now make the filesystem on astapor:
  mkfs.ext4 /dev/drbd0

- Then demote node to secondary, by executing only on astapor:
  drbdadm secondary optzimbra

- Promote node to primary, by executing only on braavos:
  drbdadm primary optzimbra

- Make the filesystem on braavos:
  mkfs.ext4 /dev/drbd0

- Now it is necessary to revert the roles back, making braavos the secondary node and astapor the primary one.

3.2 DRBD Split Brain Recovery

Assuming that the primary node is still consistent, and the secondary node has an inconsistent state, it would be necessary to recover data loss.

In Both nodes:

drbdadm disconnect optzimbra

In secondary node:

drbdadm secondary optzimbra

drbdadm connect --discard-my-data optzimbra

In the primary node:

drbdadm connect optzimbra

Finally it is possible to check the sync status, showing a similar message:

cat /proc/drbd

```
cs:Connected ro:Primary/Secondary ds:UpToDate/UpToDate C r-----
```

4 ZCS

Here will be fully installed ZCS on astapor, but just a dummy installation on braavos, since DRBD will replicate the data to the other node. Download and place ZCS installation file in astapor and braavos filesystems. It can be found at http://www.zimbra.com/downloads/os-downloads.html. In order to complete a full install on a single server, the following resource will be useful:
http://files.zimbra.com/website/docs/8.5/Zimbra_OS_Quick_Start_8.5.0.pdf

4.1 ZCS full install on primary node

The following actions must be performed sequentially on ***astapor***.

- Create directory for ZCS:

  mkdir /opt/zimbra
- Mount DRBD device on ZCS mount point:

mount /dev/drbd0 /opt/zimbra

- Check mounted device:

  df | grep zimbra

  mount | grep zimbra

- Set manual virtual link configuration temporally:

  ifconfig eth0:1 inet 172.17.18.190 netmask 255.255.255.0

- Set split DNS hostname temporally:

  hostname zcs-ha.got.com

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

- Unpack ZCS installer and proceed with full installation:

  ./install.sh

- Leave all packages to install by default, and follow the process.
- When prompted for *domain name change*, select “Yes” and then provide: got.com
- On “Main Menu” section, set admin user password by browsing through option 3 and then 4:

  “Password for admin@zcs-ha.got.com (min 6 characters):”

- Apply configuration and advance until ZCS setup process is completed:

  “Configuration complete - press return to exit”

- Check ZCS status:

  service zimbra status

- Stop ZCS:

  service zimbra stop

- Umount DRBD device:

  umount /opt/zimbra

- Set original DNS hostname:

  hostname astapor.got.com

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

- Delete temporal virtual link configuration:

  ifconfig eth0:1 down

- Demote astapor to secondary DRBD, and continue with the next section (4.2):

  drbdadm secondary optzimbra

4.2 ZCS dummy install on secondary node

The following actions must be performed sequentially on ***braavos***.

- Promote braavos to primary DRBD:

  drbdadm primary optzimbra
- Create directory for ZCS:

  mkdir /opt/zimbra
- Mount DRBD device on ZCS mount point:

  mount /dev/drbd0 /opt/zimbra
- Check mounted device:

  df | grep zimbra

  mount | grep zimbra
- Unpack ZCS installer and proceed with dummy installation:

  ./install.sh -s
- Stop ZCS:

  service zimbra stop
- Umount DRBD device:

  umount /opt/zimbra
- Demote braavos back to secondary DRBD:

  drbdadm secondary optzimbra
- Promote astapor back to primary DRBD, executing from astapor node:

  drbdadm primary optzimbra

At this point DRBD has to synchronize data from primary node, so check the status until it is done:

watch cat /proc/drbd

5 OCF

Open Cluster Framework, standard scripts to control services such as ZCS. Following actions must be performed in both nodes.

- Create file /usr/lib/ocf/resource.d/btactic/zimbra:

The following is an Embedded OLE (Object Linking and Embedding) file, double click on it to see the full content.

```
#!/bin/sh
#
# Resource script for Zimbra
```

- Also create the following symbolic link:

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

In section 6 this file will be referenced.

6 Pacemaker

Resource manager, starts and stops services orderly.

- Install RPM packages:

  pacemaker-cluster-libs-1.1.10-14.el6.x86_64
  pacemaker-libs-1.1.10-14.el6.x86_64
  pacemaker-cli-1.1.10-14.el6.x86_64
  pacemaker-1.1.10-14.el6.x86_64
  cman-3.0.12.1-59.el6.x86_64
  crmsh-1.2.5-0.el6.x86_64
  ccs-0.16.2-69.el6.x86_64
  resource-agents-3.9.2-40.el6_5.7.x86_64

Usually it is difficult to obtain the required RPM's for RHEL, so an alternative is to add CentOS repository by editing /etc/yum.repo.d/centos.repo file with:

```
[centos-6-base]
name=CentOS-$releasever - Base
mirrorlist=http://mirrorlist.centos.org/?release=6.5&arch=x86_64&repo=os
enabled=0
gpgcheck=0
baseurl=http://mirror.centos.org/centos/6.5/os/x86_64/
```

- Then update and install:

```
yum install --enablerepo=centos-6-base pacemaker \
        pcs.noarch  cman ccs resource-agents crmsh
```

There are two ways to interact with Pacemaker configuration. The first one is using the crmsh interpreter, starting the crm shell with “crm” command, and then providing configuration sentences. For instance:

```
[root@astapor ~]# crm
crm(live)# help
crm(live)# quit
```

Another way would be through ***pcs*** and ***ccs*** instructions directly from a linux tty in a bash session. Following is going to be used this way to configure de cluster, executing the commands only on the primary node.

- Create the cluster:

```
ccs --file /etc/cluster/cluster.conf --createcluster zcsCluster
```

- Add the nodes:

```
ccs --file /etc/cluster/cluster.conf --addnode astapor.got.com
ccs --file /etc/cluster/cluster.conf --addnode braavos.got.com
```

- Set fencing to defer to Pacemaker:

```
ccs --file /etc/cluster/cluster.conf --addfencedev \
        pcmk agent=fence_pcmk
ccs --file /etc/cluster/cluster.conf --addmethod \
        pcmk-redirect astapor.got.com
ccs --file /etc/cluster/cluster.conf --addmethod \
        pcmk-redirect braavos.got.com
ccs --file /etc/cluster/cluster.conf --addfenceinst pcmk \
        astapor.got.com pcmk-redirect port=astapor.got.com
ccs --file /etc/cluster/cluster.conf --addfenceinst pcmk \
        braavos.got.com pcmk-redirect port=braavos.got.com
```

- Disable CMAN quorum:

This will let the cluster function if only one node is up, and it is necessary to be performed in both nodes.

```
echo "CMAN_QUORUM_TIMEOUT=0" >> /etc/sysconfig/cman
```

- Start Pacemaker Cluster:

```
pcs cluster start --all
```

Also equivalent to execute on each node,

“service pacemaker start” or “pcs cluster start”

- Copy cluster file to secondary node:

```
scp -p /etc/cluster/cluster.conf  braavos:/etc/cluster/
```

- Check Pacemaker cluster status:

```
pcs status
crm_mon -1
```

- Show current cluster config:

```
pcs config
```

```
pcs property
crm configure show
```

- Check configuration validity:

```
crm_verify -L -V
```

- Disable STONITH (a type of fencing):

```
pcs property set stonith-enabled=false
```

- Ignore Quorum Policy:

```
pcs property set no-quorum-policy=ignore
```

- Set reconnect attempt:

```
pcs property set migration-threshold=1 –force
```

- Set stickiness:

```
pcs property set resource-stickiness=100 –force
```

Now, it is going to be used the crmsh interpreter, starting it with the following command:

```
crm configure
```

- Add floating IP address resource (Virtual IP – VIP):

```
pcs resource create VIP1 IPaddr2 ip=172.17.18.190 \
broadcast=172.17.18.255 nic=eth0 cidr_netmask=24 \
iflabel=VIP1 op monitor interval=30s timeout=30s
```

- Define DRBD cluster resource:

```
configure primitive drbd ocf:linbit:drbd params \
drbd_resource=optzimbra \
op monitor role=Master interval=60s \
op monitor role=Slave interval=50s \
```

op start role=Master interval=60s timeout=240s \
op start role=Slave interval=0s timeout=240s \
op stop role=Master interval=60s timeout=100s \
op stop role=Slave interval=0s timeout=100s

- Define DRBD Zimbra data clone:
  configure ms drbd_ms drbd \
  meta master-max=1 master-node-max=1 \
  clone-max=2 clone-node-max=1 notify=true

- Define Zimbra service resource:
  configure primitive zcs_service ocf:btactic:zimbra \
  op monitor interval=2min timeout="40s" \
  op start interval="0" timeout="360s" \
  op stop interval="0" timeout="360s"

- Define Zimbra cluster filesystem resource:
  configure primitive zcs_fs ocf:heartbeat:Filesystem params \
  device="/dev/drbd0" directory="/opt/zimbra" fstype=ext4 \
  op start interval=0 timeout=60s \
  op stop interval="0" timeout="60"

- Group all resources in the same host:
  group zcsgroup zcs_fs zcs_service \
  configure colocation VIP1-with-drbd_ms-Master \
  inf: drbd_ms:Master VIP1
  configure colocation drbd_ms-Master-with-zcs_fs \
  inf: zcs_fs drbd_ms:Master
  configure colocation zcs_fs-with-zcs_service \
  inf: zcs_service zcs_fs

- Order resources:

```
configure order drbd_ms-promote-on-VIP1 \
        inf: VIP1:start drbd_ms:promote
configure order zcs_fs-on-dbrb_ms-promote \
        inf: dbrb_ms:promote zcs_fs:start
configure order zcs_service-on-zcs_fs \
        inf: zcs_fs:start zcs_service:start
```

- Commit configuration changes and quit:

```
commit
quit
```

On both nodes make sure chkconfig is off on every service but DRBD. This means the service will not start up on when the server starts up.

```
chkconfig corosync off
chkconfig cman off
chkconfig ricci off
chkconfig pacemaker off
chkconfig drbd on
```

7 Control and check services

- Check Pacemaker cluster status

```
crm_mon -1
pcs status
```

- Check resources status:

```
crm resource status RESOURCE
```

- Check configuration validity:

```
crm_verify -L -V
```

- Edit values already configured:

  crm configure edit

  After save changes through the preferred text editor, exit and execute:

  cibadmin --replace

- Delete existent resource:

  pcs resource delete *RESOURCE*

- Clean resource history errors (check configuration health):

  crm_resource -P

- List available classes and resources:

  crm ra classes

  crm ra list ocf btactic

  crm ra list lsb

Delete cluster configuration (WARNING):

  pcs cluster destroy

8 Testing failover

- On Primary Node:

  crm node standby

  Or stop Pacemaker:

  service pacemaker stop

- Now “crm_mon” or “pcs status” will show:

```
Node astapor.got.com: standby
Online: [ braavos.got.com ]
```

- It is going to take a while before secondary node takes control. So it is possible to check logs and “crm_mon” status during the process.

  crm_mon

  tail -F /var/log/zimbra.log

  tail -F /var/log/messages

- Also it is possible to check with “crm_standby” command. A value of *true*|*on* indicates that the node is NOT able to host any resources and a value of *false*|*off* indicates that it CAN.

  crm_standby --get-value

- At any moment it will be displayed a message like the following:

```
Master/Slave Set: drbd_ms [drbd]
    Masters: [ braavos.got.com ]
    Slaves: [ astapor.got.com ]
Resource Group: zcsgroup
    zcs_fs     (ocf::heartbeat:Filesystem):    Started braavos.got.com
    zcs_service        (ocf::btactic:zimbra):  Started braavos.got.com
VIP1   (ocf::heartbeat:IPaddr2):       Started braavos.got.com
```

- Now the secondary node has control of the cluster resources, while the primary node is in standby or unreachable state. If primary node is back online, secondary node will keep the control of resources, until an explicit node move is done.

- Set back online the primary node:

  crm node online

  Or start over pacemaker service:

  service pacemaker start

- To give the control back to primary node, execute on secondary node:

  crm node standby

  Then resources will be transferred back to primary node.

- Finally “crm_mon” or “pcs status” on each node will show:

```
Online: [ astapor.got.com braavos.got.com ]
 Master/Slave Set: drbd_ms [drbd]
    Masters: [ astapor.got.com ]
    Slaves: [ braavos.got.com ]
 Resource Group: zcsgroup
    zcs_fs     (ocf::heartbeat:Filesystem):    Started astapor.got.com
    zcs_service        (ocf::btactic:zimbra):  Started astapor.got.com
VIP1    (ocf::heartbeat:IPaddr2):       Started astapor.got.com
```

9 Conclusions

HA schemes can be implemented through FLOSS if the correct tools are chosen and orchestrated in the right way.

In order to implement these schemes, it is required a high-level knowledge regarding the meshing between all elements, as well as low level domain for proper configuration thereof.

The cluster-level and HA technologies are constantly evolving, manufacturers around FLOSS business models are the most interested into developing this field, and communities have played a lead role in this regard. It is important to understand solutions that are been implemented, and be able to replace similar technologies, since eventually they will not be present in future versions of the software used.

Since high availability can be applied to a broad spectrum of services running at the OS level, it is needed adaptation of respective configurations to the specific requirements of each environment, there is no universal configuration in this sense.

10 Bibliography

Beekhof A., Scarazzini R. and Frincu D. (2012): Clusters from Scratch - Pacemaker, URI:
http://clusterlabs.org/doc/en-US/Pacemaker/1.1/html/Clusters_from_Scratch/

Gibanel Lopez, Adrian (2013): Zimbra 8 High Availability on Ubuntu 12.04, Spain, URI:
http://repositori.udl.cat/bitstream/handle/10459.1/46685/agibanell.pdf

Hellman B., Haas F., Reisner P., et al (2012): The DRBD User's Guide, URI:
http://www.drbd.org/users-guide/drbd-users-guide.html

How-to HA with Zimbra 8 OSE (2012), URI: http://forums.zimbra.com/administrators/58113-zimbra-pacemaker-drbd-howto.html

Zimbra Inc. (2014): Collaboration - Open Source Edition – Documentation, URI:
http://www.zimbra.com/documentation/zimbra-collaboration-open-source