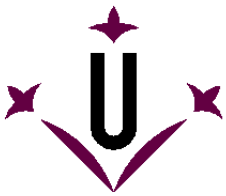

Universitat de Lleida

Escola Politècnica Superior

Màster en Enginyeria de Programari Lliure

Treball de final de màster

Zimbra 8 High Availability on Ubuntu 12.04

Autor/a: Adrián Gibanel López

Director/s: Josep Maria Ribó Balust

Setembre 2013

To the bTactic crew

ACKNOWLEDGMENTS

I acknowledge Richard M. Stallman for his personal commitment to the free software movement.

ABSTRACT

ZIMBRA 8 HIGH AVAILABILITY ON UBUNTU 12.04

Adrián Gibanel López

The purpose of this master thesis is to design and test the setup of a Zimbra 8 Open Source Edition (OSE) High Availability System (HA) in Ubuntu 12.04.

A HA system has been proposed and tested in a laboratory environment. Its setup has been documented in its all length.

The master thesis shows that thanks to some minor modifications to Zimbra OSE core and thanks to freely available Open Source HA software one can achieve a HA Zimbra OSE system.

The proposed HA Zimbra OSE system can be improved in many ways and the author suggests several ways of doing so.

Contents

List of Figures

Chapter 1

Introducing Zimbra High Availability

This thesis was written to reflect the state of the art in High Availability methods for Zimbra Open Source Edition.

1.1 High Availability

High availability is a system design approach and associated service implementation that ensures that a prearranged level of operational performance will be met during a contractual measurement period.

One of the most common high availability examples are web servers. Two web servers share the same information thanks to a shared storage. If one of the web servers fails to serve pages the other server can reclaim its primary role and shot the other node in the head (stonith) so that it can serve pages instead of the original primary node. The amount of time since the detection of the first server failure to its reestablishment is denoted as downtime. A contract for High Availability might stipulate that in a month time web servers service might be in downtime status for no more than five minutes.

High Availability, or HA as it is abbreviated, refers to the availability of resources in a computer system, in the event of component failures in the system. This can be achieved in a variety of ways, either with custom and redundant hardware to ensure availability or with software solutions using off-the-shelf hardware components.

The former class of solutions provide a higher degree of availability, but are significantly more expensive than the latter. This has led to the popularity of the latter class, with almost all vendors of computer systems offering various HA products. Typically, these products endure single points of failure in the system.

([ATFC])

As an example for more expensive systems we can mention OVH.co.uk web hosting service which uses more than 1,000 servers not only for ensuring high availability but also for dealing with high loads of visitor's queries. SQL servers seem to achieve high availability by using several RAID-1 (mirror) hard disks.

These HA systems usually ensure that a prearranged level of operational performance will be met during a contractual measurement period. ([WIHA])

High availability systems typically operate 24x7 and usually require built-in redundancy to minimize the risk of downtime due to hardware and/or telecommunication failures.

Availability can be measured relative to "100% operational" or "never failing." A widely-held but difficult-to-achieve standard of availability for a system or product is known as "five 9s" (99.999 percent) availability. ([BCHA])

From now on high availability will be refered as HA.

1.2 Vmware Zimbra OSE

VMware Zimbra is a complete email, address book, calendar and tasks solution that can be accessed from the Zimbra Web Client, Zimbra Desktop offline client, Outlook and a variety of other standards-based email clients and mobile devices. It can be deployed as a traditional binary install on Linux, or as a software virtual appliance, commonly referred to as Zimbra appliance.

Among the Zimbra Collaboration Server (ZCS) versions this thesis will approach the ZCS Open Source Edition also known as Zimbra OSE.

Vmware Zimbra OSE will be referred most of the times as Zimbra.

For more information about Vmware Zimbra you can visit: [ZLEA].

1.3 Vmware Zimbra OSE High Availability

Zimbra OSE High Availability is a project which attempts to attain HA to each one of the Zimbra Collaboration Server components so that the risk of downtime due to hardware and/or telecommunication failures is minimized. High availability is usually implemented using High Availability software aimed at Gnu/Linux originally which is adapted to the Zimbra OSE setup.

1.4 History

Prior documentation about Zimbra High Availability was written with Zimbra 6 version in mind which is devoted to work (among others) in Ubuntu 8.04 64 bit.

That documentation was based on Gnu/Linux High Availability software (heartbeat) which is no longer used for High Availability purposes nowadays.

To the best of the author's knowledge there is no updated documentation on how to setup this system.

On September 13th, 2012 Vmware announced Zimbra 8 which could be run in Ubuntu 12.04 ([VWZ8]).

1.5 Main thesis topic

The main topic of this thesis is:
Design and test the setup of a Zimbra 8 High Availability System in Ubuntu 12.04 and write a detailed report of this setup procedure.

1.6 Chosen technologies and solution

We have decided to use the following proved open source HA technologies: Corosync, Distributed Replicated Block Device (DRBD), OCF, and Pacemaker.

Corosync is a software meant to synchronize configuration files. Synchronize configuration files in a cluster is essential because all the cluster members have to share the same information and knowledge of the other cluster members. DRBD allows us to share a common storage between hosts as if a network RAID-1 hard disk it was. Finally, in order to setup and manage the whole cluster we will use Pacemaker cluster which uses OCF scripts as Cluster control scripts.

The reason for using these technologies is because they are the best well known and most used open source HA technologies. They have supersesed old technologies like heartbeat which we mentioned in **section 1.4 History**.

A set of virtual machines will be used to setup Zimbra 8 and Ubuntu 12.04 HA system. The neccessary steps to carry out this setup are: Setup network in both hosts, setup DRBD, initial Pacemaker setup, corosync installation, pacemaker installation, corosync setup, startup script disabling, DRBD script boot disabling, Pacemaker final setup, Pacemaker case of use tests.

This report describes in detail the setup procedure of each of these steps.

1.7 Notation remarks

Zimbra will be run in two *servers* in our solution. These servers will be refered to as *virtual servers* when we take the point of view of Virtualbox or virtualization point of view as in section **2.5 Virtualbox implementation**.

When the HA system will be described these same servers will be refered to as *nodes* as this is the way most HA software refer to their own cluster machines.

1.8 Structure of the document

Chapter 2 explains the high availability system that will be tested through the thesis.

Chapters 3 to 12 explain the detailed setup procedure of each element that constitutes the final system: *2 - High availability schema*, *3 - Operating System installation*, *4 - Network setup*, *5 - Zimbra installation*, *6 - DRBD Setup*, *7 - Zimbra and DRBD Startup Script Disabling*, *8 - Corosync Setup*, *9 - Zimbra OCF Resource Agent development*, and *10 - Pacemaker Setup*. Final system is a Zimbra OSE High Availability system with two servers.

We learn how to manage our new High Availability system thanks to the *11 - High Availability System Management* chapter.

Finally both conclusions and some of the ways this thesis can be improved are described in *12* chapter.

Chapter 2

High availability schema

This chapter explains the high availability system that will be tested through the Thesis.

2.1 Purpose

The proposed HA system is an Active/passive configuration. An Active/passive cluster provides a fully redundant instance of each node, which is only brought online when its associated primary node fails ([HANC]).

In our case, the primary node will act as the active server and it will provide Zimbra services such as web server, smtp, imap, etc. The secondary node will be idle just waiting for the primary node to fail and bring Zimbra services online when that event happens. In addition the secondary node also mirrors Zimbra data partition in the background thanks to DRBD.

2.2 Main schema

There are two servers which we will be named as the primary one and the secondary one. They are linked by means of two connections: The service and the communication link.

The service link is the main network interface which is connected via a normal switch. It will serve content to the final users. The communication link, which is used for the cluster management and synchronization is done via a crossover cable.

We can see the main schema, where we have added two final clients at figure 2.1.

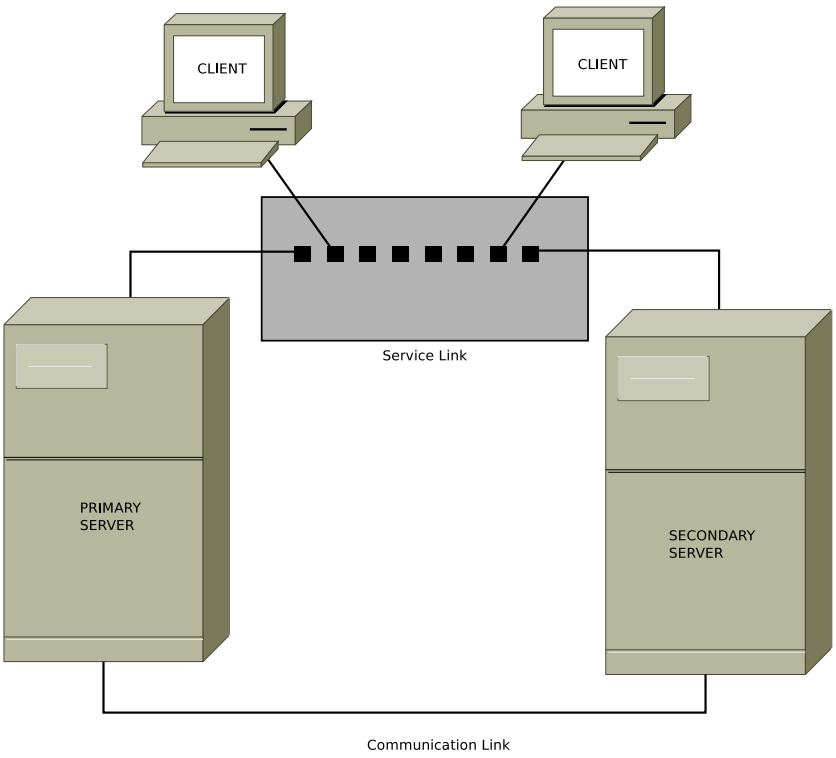

Figure 2.1: High Availability main schema

2.3 Primary server

2.3.1 Specifications

The primary server specifications are as follow:

- RAM: 4 GB
- Hard disk: 100 GB
- Processor: 2 x 2,40 Ghz

2.4 Secondary server

2.4.1 Specifications

The primary server specifications are as follow:

- RAM: 4 GB
- Hard disk: 100 GB
- Processor: 2 x 2,40 Ghz

2.5 Virtualbox implementation

2.5.1 Introduction

Although in production environments High Availability systems are implemented in Physical servers or highly optimized virtualized servers, we are going to use Oracle VM Virtualbox software to emulate the described system. This section summarizes how to create both virtual machines and link them.

2.5.2 Primary Virtual Machine creation

We click on *Machine* menu and select *New* option. The Create Virtual Machine wizard will appear.

Name and operating system

- Name: PrimaryZimbraHA
- Type: Linux
- Ubuntu (64 bit)

Memory size

Zimbra needs: 2048 MB as a minimum.

Hard drive

We select *Create a virtual hard drive now*, *Virtualbox Disk Image* as the hard drive file type, Dynamically allocated (so that the hard drive file only uses space as it fills up).

We leave the default File location and select hard disk size as 110 GB which is quite bigger than the strictly needed for our HA system.

2.5.3 Service link network on Primary Virtual Machine

We select *PrimaryZimbraHA* virtual machine and click on *Machine* menu and then in *Settings* option. We will make sure we are in *Network* section.

We will use default *Adapter 1* for service link. We are going to summarize its setup:

- Attached to: *Internal Network*
- Name: ZimbraHAService

Finally we click on OK for saving changes.

Secondary Virtual Machine creation

In order to create secondary virtual machine we can either repeat the same steps as in **2.5.2 Primary Virtual Machine creation**. Or we can make a linked clonation of the original machine. We will describe the latter option.

We select PrimaryZimbraHA virtual machine and then in *Machine* menu we select *Clone* option.

New machine name

- New machine name: SecondaryZimbraHA

- Reinitialize the MAC address of all network cards: Checked

We select *Linked clone* as Clone type.
Finally we click on *Clone* button so that cloning is performed.

2.5.4 Service link network on Secondary Virtual Machine

As we did in subsection **2.5.3 Service link network on Primary Virtual Machine** we select *SecondaryZimbraHA* virtual machine and click on *Machine* menu and then in *Settings* option. We will make sure we are in *Network* section.

We will use default *Adapter 1* for service link. We are going to summarize its setup:

- Attached to: *Internal Network*
- Name: *ZimbraHAService*

If we have cloned the virtual machine settings might be correct by default.
Finally we click on OK for saving changes.

2.5.5 Communication link

For both PrimaryZimbraHA and SecondaryZimbraHA virtual machines we will perform a very similar operation than the one done in subsection **2.5.3 Service link network on Primary Virtual Machine**.

But now we make sure that we *Adapter 2* is enabled as an *Internal Network* which name is *ZimbraHACommunication*.

2.5.6 NAT link

In order to make installation easier we will enable *Adapter 3* in both virtual machines so that it can use the host Internet in order to fetch packages and perform post installation setup.

Similarly to subsection **2.5.3 Service link network on Primary Virtual Machine** we make sure that *Adapter 3* is enabled and that it is attached to NAT.

2.5.7 Email client Virtual Machine

A Virtual Machine whose only purpose is to test HA from a service link point of view might be added if needed. We are not to cover the installation and its setup here. We will just mention its network setup is similar to PrimaryZimbraHA and SecondaryZimbraHA but removing the second interface which serves for communication link and that, of course, does not make sense in an Email client VM.

Chapter 3

Operating System installation

This chapter explains the Operating System installation.

3.1 Introduction

Ubuntu is a complete desktop Linux operating system, freely available with both community and professional support. Ubuntu is suitable for both desktop and server use and includes more than one thousand pieces of software ([UWHA]).

We will use Ubuntu 12.04 in its 64 bit mode because it is one of the official supported Operating System for Zimbra 8 versions. We denote an external DRBD metadata as DRBD-Meta-Disk. We can understand it is an special partition that helps DRBD system to track changes between synchronized partitions between both primary and secondary nodes. We can find a more accurate definition at Linbit site: [LDIN].

These instructions are valid for both primary and secondary nodes. The only difference is that each one of them will have a different host name.

An Ubuntu Gnu/Linux installation might be as complex as of having Logical Volumes Group (LVM) for the ease of hard disk space management. In this installation we will avoid that kind of partitions so that final Pacemaker setup (Section 10.4 Pacemaker final setup) is easier.

3.2 Ubuntu 12.04 64 bit minimal

In order to track all the requisites and just install what the high availability system needs we will use an Ubuntu minimal disk for installation. These disks can be downloaded from [UMIN].

The used download was: *Ubuntu 12.04 "Precise Pangolin" Minimal CD* from the *64-bit PC (amd64, x86_64)* section.

3.2.1 Installer boot menu

We just press Return key to select default boot option: *Install*.

3.2.2 Select a language

We choose the desired language: *English*

3.2.3 Select your location

We choose our location: *United States*

3.2.4 Configure the keyboard

We choose *no* for semi automated keyboard detection. That let us choose *English (US* as country of origin for the keyboard. Finally we select default *English (US)* keyboard.

3.2.5 Network

The network detection will be automatically sorted out if one our of interfaces is bridge in a network provided of a DHCP server. Once we boot into the operating system we will setup network manually. We are asked our host name. We choose: *primary* in case we are installing primary node and *secondary* in case we are installing secondary node.

3.2.6 Ubuntu archive mirror country

We select default *United States* and its associated mirror: *us.archive.ubuntu.com*. When asked for HTTP proxy information we just press Return key to continue.

3.2.7 Checking Ubuntu mirror

Ubuntu installation will perform some background checking and downloads without the user being noticed. Then the installation suddenly continues by retrieving and installing additional packages and components.

3.2.8 Set up users and passwords

We first are asked to write *Full name for the new user* and then *Username for the account*. We will provide both of them. We are also prompted twice the user password. When asked we choose not to encrypt the home directory.

3.2.9 Configure the clock

Given our detected physical location we just press Return key to validate it.

3.2.10 Partition disks

Introduction

Assuming a 1.8 Terabyte hard disk in order to setup DRBD-Meta-Disk there is enough with 59 megabytes. We will be on the safe side and setup it with a 150 megabytes size. In order to safe calculate other DRBD meta disk partitions we can check [LDIN].

We will not be using a SWAP partition because Zimbra works better without it according to Zimbra Performance Tunning Guidelines ([ZPTG]). If a SWAP partition would be needed to be used we encourage to add it as a fixed size secondary hard disk.

Manual partitioning

We select *Manual* partitioning method. We then select our hard disk in our case: *SCSI1 (0,0,0) (sda) - 107.4 GB ATA VBOX HARD DISK*. When asked to create a new empty partition table on the device we reply *Yes*. Now we are back at Partition disk screen. For each one of the partitions to be created we should select *FREE SPACE* under our hard disk. Then select create a new partition. Once a partition has been defined we will confirm its settings by selecting *Done setting up the partition*. We will describe in an schematic manner how the partitions should be created.

Root partition

- New partition size: *10 GB*
- Type for the new partition: *Primary*
- Location for the new partition: *Beginning*

Partition settings for Root partition will be left as default except for the Bootable flag which should be set to *on*. These were:

- Use as: *Ext4 Journaling file system*
- Mount point: */*
- Mount options: *defaults*
- Label: *none*
- Reserved blocks: *5%*
- Typical usage: *standard*
- Bootable flag: ***on***

DRBD-Meta-Disk partition

- New partition size: *150 MB*
- Type for the new partition: *Primary*
- Location for the new partition: *Beginning*

Partition settings for DRBD-Meta-Disk partition were finally:

- Use as: *do not use*
- Bootable flag: *off*

ZimbraData partition

- New partition size: *Rest of disk space*
- Type for the new partition: *Primary*
- Location for the new partition: *Beginning*

Partition settings for ZimbraData partition were finally:

- Use as: *Ext4 Journaling file system*
- Mount point: ***/opt***
- Mount options: ***defaults***
- Label: *none*
- Reserved blocks: *0%*
- Typical usage: *standard*

- Bootable flag: *off*

We are using */opt* partition for Zimbra Data because Zimbra programs and its data is stored in **/opt/zimbra** by default.

Now we are going to confirm all our created partitions by selecting *Finish partitioning and write changes to disk*. When asked to return to the partitioning menu because a swap partition is needed we will just skip it by saying: *No*. Finally we confirm that we want to write changes to disk by selecting: *Yes*.

3.2.11 Configuring x11-common

We select: *No automatic updates* when asked how to manage upgrades just to make things easier. In a production environment we must select *Install security updates automatically*.

3.2.12 Software selection

When asked which software to install we will only check *OpenSSH server* for installation in order to keep the installation as minimal and functional as possible.

3.2.13 Install the GRUB

When asked to Install the GRUB boot loader to master boot record we select Yes.

3.2.14 Finish the installation

When we are asked if the system clock is set to UTC we select Yes. Once clock it is been set we need to do a change in Virtualbox. In the running Virtual machine we select *Devices* menu, CD/DVD devices submenu and then we uncheck the minimal Ubuntu 12.04 64 bit iso.

On request we can force unmount in Virtualbox.

Finally we click on *Continue* to finish the installation.

Chapter 4

Network setup

This chapter explains the network setup.

4.1 Network schema

We can just check the High Availability main schema (figure 2.1) where network has been already described. There are three networks. The service link is the main network interface for serving content to the final users. The communication link, which is used for the cluster management and synchronization is done via a crossover cable. Finally NAT link gives the machines access to Internet.

One of the reasons why there are two links is because we use communication link for DRBD synchronization and that prevents DRBD traffic from interfering with service traffic. The other reason is because we do not want cluster communication to be interfered by DRBD traffic. This way we minimize false off line node detections.

For more detailed explanations on how it is implemented in Virtualbox check **2.5 Virtualbox implementation** chapter.

4.2 Network setup

4.2.1 High Availability service

FQDN

High Availability server FQDN will be:

```
public.zimbraha.lan
```

Service link

Service link network setup consists of a Class C configuration where the network card address is 192.168.77.203, as per being a Class C its net mask is 255.255.255.0 and thus its broadcast is 192.168.77.255.

This interface will not be configured in */etc/network/interfaces* but by pacemaker itself thanks to its cluster definition.

That ip will be the one that email client will connect to our service. Our cluster will make sure that it is only configured in only one server, the active one.

4.2.2 Primary server

Communication link

Primary server's Communication link network setup consists of a Class C configuration where the network card address is 10.0.66.201, as per being a Class C its net mask is 255.255.255.0 and thus its broadcast is 10.0.66.255. As a gateway it will be using the first network address in the network range which is 10.0.66.1.

The correspondent configuration code for */etc/network/interfaces* file is:

```
auto eth0
iface eth0 inet static
        address 10.0.66.201
        netmask 255.255.255.0
        broadcast 10.0.66.255
```

4.2.3 FQDN

Fully Qualified Domain Name for primary server will be:

```
primary.zimbraha.lan
```

4.2.4 Additional links

In order to have Internet connectivity an additional link set as a NAT interface in Virtualbox will be added.

4.2.5 Secondary server

Communication link

Secondary server's Communication link network setup consists of a Class C configuration where the network card address is 10.0.66.202, as per being a Class C

its net mask is 255.255.255.0 and thus its broadcast is 10.0.66.255. As a gateway it will be using the first network address in the network range which is 10.0.66.1.

The correspondent configuration code for */etc/network/interfaces* file is:

```
auto eth1
iface eth1 inet static
        address 10.0.66.202
        netmask 255.255.255.0
        broadcast 10.0.66.255
```

4.2.6 FQDN

Fully Qualified Domain Name for secondary server will be:

```
secondary.zimbraha.lan
```

4.2.7 Additional links

In order to have Internet connectivity an additional link set as a NAT interface in Virtualbox will be added.

4.3 Firewall

Our default Ubuntu minimal installation does not have

4.3.1 Zimbra ports

These are the Zimbra ports that would need to be open in a production environment ([ZWPO]):

- 25 smtp [mta] - incoming mail to postfix
- 80 http [mailbox] - web mail client
- 110 pop3 [mailbox]
- 143 imap [mailbox]
- 443 https [mailbox] - web mail client over ssl
- 465 smtps [mta] - incoming mail to postfix over ssl (Outlook only)
- 587 smtp [mta] - Mail submission over tls

- 993 imaps [mailbox] - imap over ssl
- 995 pops [mailbox] - pop over ssl
- 7071 https [mailbox] - admin console

And these are the Zimbra ports typically only used by the zimbra system itself ([ZWPO]).

- 389 ldap [ldap]
- 636 ldaps [ldaps, if enabled]
- 7025 lmtp [mailbox] - local mail delivery
- 7047 conversion server
- 7306 mysql [mailbox]
- 7307 mysql [logger] - logger
- 7780 http [mailbox] - spell check
- 10024 smtp [mta] - to amavis from postfix
- 10025 smtp [mta] - back to postfix from amavis

System access ports ([ZWPO]) are:

- 22 ssh
- 514 syslogd [logger] (udp)

4.3.2 High Availability ports

In the service link interfaces we need to make sure these ports are accessible from one node to another:

- Corosync communication port: 5405 (udp)
- Corosync communication port: 5404 (udp)
- DRBD: 7788

Chapter 5

Zimbra installation

This chapter explains the Zimbra OSE installation.

5.1 Introduction

Zimbra is an enterprise-class email, calendar and collaboration solution, built for the cloud, both public and private. With a redesigned browser-based interface, Zimbra offers the most innovative messaging experience available today, connecting end users to the information and activity in their personal clouds ([ZWWW]).

One of the main Zimbra services is a web server where the final user can check its own email, calendar and contacts among others. This is the service that we will check when testing HA.

5.2 Operating system checks

5.2.1 /etc/hosts

On both nodes We will make sure in */etc/hosts* file that we have:

```
192.168.77.203 public.zimbraha.lan public
```

5.2.2 /etc/hostname

On primary node we will make sure that */etc/hostname* contains:

```
primary
```

. In order to apply changes we will run:

```
service hostname restart
```

Same operation but using *secondary* will need to be performed **on secondary node**.

5.3 Package requirements

On both nodes we will make sure we meet the Zimbra system package requirements.

```
sudo apt-get install libgmp3c2 libexpat1 \
libstdc++6 sysstat libpcre3 libperl5.14 \
sqlite3 libidn11 pax
```

5.4 Zimbra 8.0.4 for Ubuntu 12.04

On both nodes Zimbra 8.0.4 for Ubuntu 12.04 in form of a tar.gz file was downloaded from [VWZD]. Once downloaded is advised to check its md5sum. Finally we untar it and cd into the untarred directory.

First we download tgz file and its md5sum file:

```
wget "http://files2.zimbra.com\
/downloads/8.0.4_GA/zcs-8.0.4\
_GA_5737.UBUNTU12_64\
.20130524120036.tgz.md5"

wget "http://files2.zimbra.com\
/downloads/8.0.4_GA/zcs-8.0.4\
_GA_5737.UBUNTU12_64\
.20130524120036.tgz"
```

Then we check if the download is correct by calculating its md5sum:

```
md5sum -c zcs-8.0.4\
_GA_5737.UBUNTU12_64\
.20130524120036.tgz.md5
```

If md5sum is correct you will see:

```
zcs-8.0.4_GA_5737.UBUNTU12_64.20130524120036.tgz: OK
```

Finally we untar it:

```
tar xzf zcs-8.0.4\
_GA_5737.UBUNTU12_64\
.20130524120036.tgz
```

5.5 Complete Install script on Primary

5.5.1 Service link manual configuration

We need to configure service link manually with:

```
ifconfig eth0 192.168.77.203 \
netmask 255.255.255.0
```

5.5.2 Installation start

Only on primary node we will change directory into Zimbra installation directory and run installation script:

```
cd zcs-8.0.4_GA_5737.UBUNTU12_64.20130524120036
./install.sh
```

5.5.3 License agreement

We will have to agree with the terms of the software license agreement by typing *Yes* and pressing Return key.

5.5.4 Zimbra packages install

Now we are requested which Zimbra packages we want to install. We will choose the default ones.

```
Select the packages to install

Install zimbra-ldap [Y]

Install zimbra-logger [Y]

Install zimbra-mta [Y]

Install zimbra-snmp [Y]

Install zimbra-store [Y]

Install zimbra-apache [Y]

Install zimbra-spell [Y]
```

```
Install zimbra-memcached [N]

Install zimbra-proxy [N]
```

We are told that the system will be modified and if we want to continue. We reply *Yes*.

Zimbra packages installation is being performed.

5.5.5 Change hostname

When asked:

```
DNS ERROR resolving primary.zimbraha.lan
It is suggested that the hostname be resolvable via DNS
Change hostname [Yes]
```

we will change it (*Yes*) so that it says: *public.zimbraha.lan*.

As DNS is not setup the same question will be asked. We will answer that *No* we do not want to re-enter hostname.

In a production environment we will need to make sure an A DNS field so that public.zimbraha.lan resolves to its public ip or internal ip.

5.5.6 Change domain name

When asked:

```
DNS ERROR resolving MX for public.zimbraha.lan
It is suggested that the domain name have
 an MX record configured in DNS
Change domain name? [Yes]
```

we will change the domain name and specify *zimbraha.lan*. As DNS is not setup the same question will be asked. We will answer that *No* we do not want to re-enter domain name.

5.5.7 Set password and apply

We are shown the zmsetup script:

```
Main menu

   1) Common Configuration:
```

```
   2) zimbra-ldap:                                 Enabled
   3) zimbra-store:                                Enabled
    +Create Admin User:                           yes
    +Admin user to create:                        admin@zimbraha.lan
*** +Admin Password                               UNSET
    +Anti-virus quarantine user:                  virus-quarantine.
       uthvrv6amf@zimbraha.lan
    +Enable automated spam training:              yes
    +Spam training user:                 spam.aw9h_i7ms@zimbraha.lan
    +Non-spam(Ham) training user:         ham.djdehqsd@zimbraha.lan
    +SMTP host:                                   public.zimbraha.lan
    +Web server HTTP port:                        80
    +Web server HTTPS port:                       443
    +Web server mode:                             https
    +IMAP server port:                            143
    +IMAP server SSL port:                        993
    +POP server port:                             110
    +POP server SSL port:                         995
    +Use spell check server:                      yes
    +Spell server URL:                            http://public.
      zimbraha.lan:7780/aspell.php
    +Configure for use with mail proxy:           FALSE
    +Configure for use with web proxy:            FALSE
    +Enable version update checks:                TRUE
    +Enable version update notifications:         TRUE
    +Version update notification email:           admin@zimbraha.lan
    +Version update source email:                 admin@zimbraha.lan

   4) zimbra-mta:                                  Enabled
   5) zimbra-snmp:                                 Enabled
   6) zimbra-logger:                               Enabled
   7) zimbra-spell:                                Enabled
   8) Default Class of Service Configuration:
   r) Start servers after configuration            yes
   s) Save config to file
   x) Expand menu
   q) Quit
```

we will type *3* in order to select *zimbra-store* configuration.

We are shown Store configuration:

```
Store configuration
```

```
   1) Status:                            Enabled
   2) Create Admin User:                 yes
   3) Admin user to create:              admin@zimbraha.lan
** 4) Admin Password                     UNSET
   5) Anti-virus quarantine user:        virus-quarantine.
      uthvrv6amf@zimbraha.lan
   6) Enable automated spam training:  yes
   7) Spam training user:   spam.aw9h_i7ms@zimbraha.lan
   8) Non-spam(Ham) training user: ham.djdehqsd@zimbraha.lan
   9) SMTP host:                         public.zimbraha.lan
  10) Web server HTTP port:              80
  11) Web server HTTPS port:             443
  12) Web server mode:                   https
  13) IMAP server port:                  143
  14) IMAP server SSL port:              993
  15) POP server port:                   110
  16) POP server SSL port:               995
  17) Use spell check server:            yes
  18) Spell server URL:                  http://public.
      zimbraha.lan:7780/aspell.php
  19) Configure for use with mail proxy:  FALSE
  20) Configure for use with web proxy:   FALSE
  21) Enable version update checks:       TRUE
  22) Enable version update notifications:TRUE
  23) Version update notification email: admin@zimbraha.lan
  24) Version update source email:       admin@zimbraha.lan

Select, or 'r' for previous menu [r]
```

we will select *4* for *Admin password*.

We are shown:

```
Password for admin@zimbraha.lan (min 6 characters): [31I2YlBw3]
```

We will just press enter to accept suggested password.

Now setup is complete:

```
Main menu

   1) Common Configuration:
   2) zimbra-ldap:                                Enabled
```

```
   3) zimbra-store:                          Enabled
   4) zimbra-mta:                            Enabled
   5) zimbra-snmp:                           Enabled
   6) zimbra-logger:                         Enabled
   7) zimbra-spell:                          Enabled
   8) Default Class of Service Configuration:
   r) Start servers after configuration      yes
   s) Save config to file
   x) Expand menu
   q) Quit

*** CONFIGURATION COMPLETE - press 'a' to apply
Select from menu, or press 'a' to apply config (? - help)
```

We press *r* to return to previous menu and finally we apply the configuration by pressing *a*.

We are asked to save configuration data to a file and we are offered a file name for it. We will just accept defaults:

```
Save configuration data to a file? [Yes]
Save config in file: [/opt/zimbra/config.21624]
Saving config in /opt/zimbra/config.21624...done.
```

When informed that the system will be modified we will just said that *Yes* we want to continue.

```
The system will be modified - continue? [No]
```

5.5.8 Zimbra notification

We are asked to notify Zimbra of our installation. We will reply *No*.

```
You have the option of notifying Zimbra of your installation.
This helps us to track the uptake of the
   Zimbra Collaboration Server.
The only information that will be transmitted is:
  The VERSION of zcs installed (8.0.4_GA_5737_UBUNTU12_64)
  The ADMIN EMAIL ADDRESS created (admin@zimbraha.lan)

Notify Zimbra of your installation? [Yes]
```

5.5.9 End of installation

Installation ends with this message where we can just press Return.

```
Configuration complete - press return to exit
```

5.5.10 Service link disable

In order to perform dummy installation on Secondary we need service link to be configured manually on secondary node. That conflicts with service link being configured on primary node. We will also need to stop Zimbra services.

We are going to disable service link on primary with:

```
service zimbra stop
ifconfig eth0 down
```

5.6 Dummy installation on Secondary

We will perform the same exact installation steps as the ones described in **5.5 Complete Install script on Primary** even the **5.5.10 Service link disable** subsection which we need because the Cluster will be the only one to configure the service link.

Now we are going to delete our unused Zimbra installation **only on secondary node** with:

```
rm -rf /opt/zimbra
```

As this is a dummy installation we do not need to write down Zimbra Administration password.

Chapter 6

DRBD Setup

This chapter explains the DRBD Setup.

6.1 Introduction

DRBD is a system that let us mirror a block device via an assigned device. DRBD can be understood as network based raid-1 and it is used in HA clusters ([LDWI]). We are using DRBD to mirror the block device where Zimbra files will be stored in.

6.2 Requirements

We will install DRBD packages for the Ubuntu 12.04 system thanks to the following command:

```
sudo apt-get install drbd8-utils
```

in both nodes.

6.3 Communication hosts

For the purpose of both DRBD and Corosync communication we need to define our hosts from the communication link point of view. We need to edit */etc/hosts* and make sure we have **in both nodes**:

```
10.0.66.201 primary.zimbraha.lan primary
10.0.66.202 secondary.zimbraha.lan secondary
```

6.4 Disable Zimbra

As Zimbra is using the DRBD backing device which is */dev/sda3* we need to unmount it and that means that we need to stop Zimbra services before doing so.

So we will first stop zimbra on **primary node** by doing:

```
service zimbra stop
```

And then in **both nodes** we will umount the partition with:

```
sudo umount /dev/sda3
```

and make sure that the partition is not mounted automatically by editing:

```
/etc/fstab
```

and commenting its line:

```
#UUID="f23efeab" /opt ext4 errors=remount-ro 0 1
```

6.5 DRBD Resource config

To be performed in both nodes.

We will backup main DRBD configuration file:

```
cp /etc/drbd.conf /etc/drbd.conf.orig
```

.

We then need to edit:

```
/etc/drbd.conf
```

so that it has:

```
include "drbd.d/global_common.conf";
include "drbd.d/*.res";

resource zimbradata {
  protocol C;
  handlers {
    pri-on-incon-degr "halt -f";
  }
  startup {
    degr-wfc-timeout 120; # 2 min
  }
```

```
  disk {
    on-io-error detach;
  }
  net {
  }
  syncer {
    rate 10M;
    al-extents 257;
  }
  on primary {
    device /dev/drbd0;
    disk /dev/sda3;
    address 10.0.66.201:7788;
    meta-disk /dev/sda2[0];
  }
  on secondary {
    device /dev/drbd0;
    disk /dev/sda3;
    address 10.0.66.202:7788;
    meta-disk /dev/sda2[0];
  }
}
```

Among other issues DRBD configuration settings establish a *10 megabits synchronisation* rate using one of the DRBD available protocols (*C*). It also defines that on the node named `primary` we define a new DRBD *device* named *drbd0* which is going to track */dev/sda3* partition thanks to its meta-disk partition found in */dev/sda2*. The same setting is also specified for the node named `secondary`. The only difference is IP *address* where each one of the nodes have DRBD server listening to.

If we want to modify our drbd.conf to increase synchronisation rate, synchronised devices or any other settings we can check documentation at [LDDC].

6.6 Start DRBD module

To be performed in both servers.

```
modprobe drbd
```

.

6.7 Metadata disk initialisation

To be performed in both servers. We make sure the metadata partition does not have any prior metadata signature.

```
dd if=/dev/zero of=/dev/sda2 bs=1K count=100
```

And we create the zimbradata metadata partition:

```
drbdadm create-md zimbradata
```

.

The output for both servers will similar to:

```
Writing meta data...
initializing activity log
NOT initialized bitmap
New drbd meta data block successfully created.
success
```

if sucedeed.

6.8 First DRBD synchronisation

To be performed in both servers.

```
drbdadm up all
```

If everything goes ok we should return to the prompt.

Were we will asked about usage, we just reply that we do not want to participate in the survey by saying 'no'.

To be performed in Primary server only.

```
drbdadm -- --overwrite-data-of-peer primary all
drbdadm -- connect all
```

.

If we happen to find a *net-config disconnect first* error we can safely ignore it.

We are goint to check DRBD first synchronisation status.

```
cat /proc/drbd
```

which will output something like:

```
version: 0.7.20 (api:77/proto:74)
SVN Revision: 1743 build by <a href="mailto:phil@mescal">\
phil@mescal</a>, 2005-01-31 12:22:07
0: cs:SyncSource st:Primary/Secondary ld:Consistent
ns:13441632 nr:0 dw:0 dr:13467108 al:0 \
bm:2369 lo:0 pe:23 ua:226 ap:0
[==>..............] sync'ed: 3.1% (7000/7168)M
finish: 1:14:16 speed: 2,644 (2,204) K/sec
1: cs:Unconfigured
```

.

We must wait for the first synchronisation to end so that we can safely complete the rest of the instructions.

Final end output will similar to:

```
version: 8.3.11 (api:88/proto:86-96)
srcversion: 93CE421BB73A731BDC72D8E
 0: cs:Connected ro:Primary/Secondary
          ds:UpToDate/UpToDate C r-----
    ns:5691029 nr:0 dw:0 dr:6002290
  al:0 bm:367 lo:0 pe:0 ua:0 ap:0
  ep:1 wo:f oos:0
```

where we can see that both primary and secondary are updated (*UpToDate*).

Chapter 7

Zimbra and DRBD startup scripts disabling

This chapter explains how to disable both Zimbra and DRBD startup scripts.

7.1 Introduction

We need to disable both default Zimbra and DRBD startup scripts because Pacemaker (see subsection **10.1 About Pacemaker**) will be the responsible for starting and stopping both Zimbra and DRBD thanks to OCF scripts.

The explanation is that it is not safe to start Zimbra at boot because Zimbra needs its filesystem to be mounted. Filesystem needs DRBD to be loaded so that ZimbraData partition exists. All of these requirements are handled by Pacemaker which has been setup for the task. The same reasoning applies to DRBD.

7.2 Disable Zimbra startup scripts

We just have to run **on both nodes**:

```
update-rc.d -f zimbra remove
```

7.3 Disable DRBD startup scripts

We just have to run **on both nodes**:

```
update-rc.d -f drbd remove
```

Chapter 8

Corosync setup

This chapter explains the Corosync setup.

8.1 About Corosync

The Corosync Cluster Engine is a group communication system for implementing HA within applications. Among its features we can find:

- Create replicated state machines thanks to a closed process group communication model
- Restart application process if it fails thanks to simple availability manager
- A configuration and statistics in-memory database
- A quorum system that notifies applications when quorum is achieved or lost.

The software is designed to operate on UDP/IP and InfiniBand networks natively.

We can find more information about Corosync. Either at its Wikipedia article ([WCSA]) or at its web page ([CORW]).

Corosync enables a group communication system so that Pacemaker can talk to all the cluster nodes without having to implement communication capabilities.

8.2 Corosync installation

In both nodes we just need to install Corosync packages for Ubuntu 12.04.

We need to issue this command:

```
apt-get install corosync
```

8.3 Corosync.conf

The corosync.conf file on both computers it will be modified to use upnp so that we can use in non multicast environments. In order to use upnp we need to use a corosync version higher than 1.3 but that is not a problem because current version is higher than that.

8.3.1 Primary server Corosync.conf

We create the file:

`/etc/corosync/corosync.conf`

which its contents will be:

```
# Please read the openais.conf.5 manual page

totem {
version: 2

# How long before declaring a token lost (ms)
token: 5000

# How many token retransmits before
# forming a new configuration
token_retransmits_before_loss_const: 20

# How long to wait for join messages
# in the membership protocol (ms)
join: 1000

# How long to wait for consensus to be achieved
# before starting a new round of
# membership configuration (ms)
consensus: 7500

# Turn off the virtual synchrony filter
vsftype: none

# Number of messages that may be sent by one
# processor on receipt of the token
max_messages: 20
```

```
# Limit generated nodeids to 31-bits
# (positive signed integers)
clear_node_high_bit: yes

# Disable encryption
  secauth: off

# How many threads to use for encryption/decryption
  threads: 0

# Optionally assign a fixed node id (integer)
 #nodeid: 1234


        rrp_mode: passive


interface {
member {
memberaddr: 10.0.66.201
}
member {
memberaddr: 10.0.66.202
}
        ringnumber: 0
        bindnetaddr: 10.0.66.201
        mcastport: 5405
        }
transport: udpu
}
amf {
mode: disabled
}

service {
  # Load the Pacemaker Cluster Resource Manager
  ver:       0
  name:      pacemaker
}
```

```
aisexec {
        user:   root
        group:  root
}

logging {
        fileline: off
        to_stderr: yes
        to_logfile: no
        to_syslog: yes
syslog_facility: daemon
        debug: off
        timestamp: on
   logger_subsys {
           subsys: AMF
           debug: off
tags: enter|leave|trace1|trace2|trace3|trace4|trace6
   }
}
```

Among other settings the configuration defines that there are two node members which their ips are: 10.0.66.201 and 10.0.66.202. The transport must use unicast protocol (udpu) through 5405 port (and 5404 too). Logging is set to output into syslog and stderr but without debug output. As we are using unicast we need to define the current node ip as the multicast one: 10.0.66.202.

8.3.2 Secondary server Corosync.conf

We create the file:

```
/etc/corosync/corosync.conf
```

which its contents will be:

```
# Please read the openais.conf.5 manual page

totem {
version: 2

# How long before declaring a token lost (ms)
token: 5000
```

```
# How many token retransmits before
# forming a new configuration
token_retransmits_before_loss_const: 20

# How long to wait for join messages
# in the membership protocol (ms)
join: 1000

# How long to wait for consensus to be achieved
# before starting a new round of
# membership configuration (ms)
consensus: 7500

# Turn off the virtual synchrony filter
vsftype: none

# Number of messages that may be sent by one
# processor on receipt of the token
max_messages: 20

# Limit generated nodeids to 31-bits
# (positive signed integers)
clear_node_high_bit: yes

# Disable encryption
  secauth: off

# How many threads to use for encryption/decryption
  threads: 0

# Optionally assign a fixed node id (integer)
 #nodeid: 1234


        rrp_mode: passive


interface {
member {
memberaddr: 10.0.66.201
}
```

```
member {
memberaddr: 10.0.66.202
}
        ringnumber: 0
        bindnetaddr: 10.0.66.202
        mcastport: 5405
        }
transport: udpu
}
amf {
mode: disabled
}

service {
  # Load the Pacemaker Cluster Resource Manager
  ver:       0
  name:      pacemaker
}

aisexec {
        user:   root
        group:  root
}

logging {
        fileline: off
        to_stderr: yes
        to_logfile: no
        to_syslog: yes
syslog_facility: daemon
        debug: off
        timestamp: on
   logger_subsys {
            subsys: AMF
            debug: off
tags: enter|leave|trace1|trace2|trace3|trace4|trace6
   }
}
```

You can check subsection **8.3.1 Primary server Corosync.conf** for configuration file settings explanation.

8.4 Corosync's Authkey

In **primary node** we will create the file:

`/etc/corosync/authkey`

thanks to running:

`corosync-keygen`

We might be requested to press keys on our keyboard to generate entropy.

Once the file created we will copy the very same file to the **secondary node** in the same path as in primary server.

In order to secure it in secondary node we will need to run:

```
chown root:root /etc/corosync/authkey
chmod 400 /etc/corosync/authkey
```

8.5 Cfgtool

In **both nodes** we need to edit:

`/etc/rc.local`

in order to add the line:

`/usr/sbin/corosync-cfgtool -r`

just before the:

`exit 0`

line.

This way we make sure that redudant ring state is reset cluster wide after a fault to re-enable redundant ring.

8.6 Corosync startup enabling

In **both nodes** in order to enable Corosync at boot we need to edit:

`/etc/default/corosync`

file so that:

`START=no`

line becomes:

`START=yes`

.

8.7 Corosync reboot and check

In **both nodes** in order to check Corosync startup we need to reboot the machine thanks to:

```
sync
shutdown -r now
```

.

Once the machine has rebooted we can cluster status thanks to:

```
crm_mon
```

In order to check that everything is ok we need to make sure that the output shown in both nodes is the same one. If both nodes are shown inside Online line that means that both nodes are detected to be online from the Pacemaker point of view.

Here there is a crm_mon output where both nodes are online:

```
============
Last updated: Sun Sep  8 13:06:11 2013
Last change: Sun Sep  8 13:04:19 2013 via crmd on primary
Stack: openais
Current DC: primary - partition with quorum
Version: 1.1.6-9971ebba4494012a93c03b40a2c58ec0eb60f50c
2 Nodes configured, 2 expected votes
0 Resources configured.
============

Online: [ secondary primary ]
```

Chapter 9

Zimbra OCF Resource Agent development

This chapter explains the development of a Zimbra OCF Resource Agent.

9.1 Introduction

A resource agent is a standardized interface for a cluster resource. It translates a standard set of operations into steps specific to the resource or application, and interprets their results as success or failure ([LHRA]). An OCF resource agent is based on the Open Clustering Framework Resource Agent API specifications ([LORA]).

In order to use the latest Pacemaker version to manage Zimbra HA an OCF Resource Agent script was needed. So a Zimbra OCF Resource Agent was developed.

9.2 Development log

- In order to develop a resource agent it is advised to have cluster management stopped with:

  ```
  service corosync stop
  ```

  and to handle manually the resources which current script is dependant on.

- It is advised to use *ocf-tester* script which will be found in *cluster-agents* package in order to debug if the OCF script actions comply with the required ones in [LORA].

- The minimal required actions were implemented because Zimbra server does not promote or demote itself.
- In order to validate the tests with ocf-tester you can avoid generating valid meta-data XML output. However when using it in production XML output has to be a valid meta-data XML.
- Default times for waiting to Zimbra service to start or stop were modified to satisfy large (more than 4 minutes) Zimbra start or stop actual times.

9.3 Zimbra OCF source code

You can find Zimbra OCF source code at appendix B - Zimbra OCF Source Code.

Chapter 10

Pacemaker setup

This chapter explains the Pacemaker installation and setup.

10.1 About Pacemaker

Pacemaker is an Open Source, High Availability resource manager suitable for both small and large clusters ([CLPW]) which features:

- Detection and recovery of machine and application-level failures
- Supports practically any redundancy configuration
- Supports both quorate and resource-driven clusters
- Configurable strategies for dealing with quorum loss (when multiple machines fail)
- Supports application startup/shutdown ordering, regardless machine(s) the applications are on
- Supports applications that must/must-not run on the same machine
- Supports applications which need to be active on multiple machines
- Supports applications with multiple modes (eg. master/slave)
- Provably correct response to any failure or cluster state. The cluster's response to any stimuli can be tested off line before the condition exists

Pacemaker let us manage the HA cluster as a single system from anyone of the cluster nodes. In order to interact with each one of the nodes it needs Corosync communication capabilities.

10.2 Pacemaker installation

In both nodes we just need to install Pacemaker packages for Ubuntu 12.04.

We need to issue this command:

```
apt-get install pacemaker
```

We can safely ignore this warning:

```
Warning: The home dir /var/lib/heartbeat
 you specified already exists.
Adding system user ‘hacluster’ (UID 105) ...
Adding new user ‘hacluster’ (UID 105) with group ‘haclient’ ...
The home directory ‘/var/lib/heartbeat’ already exists.
 Not copying from ‘/etc/skel’.
adduser: Warning: The home directory ‘/var/lib/heartbeat’
 does not belong to the user you are currently creating.
Processing triggers for libc-bin ...
ldconfig deferred processing now taking place
```

10.3 bTactic Zimbra OCF installation

In both nodes we need to obtain the Zimbra OCF script. Zimbra OCF script is found inside BtacticOCF tar.gz file ([BTOC]) which can be downloaded from BtacticOCF tar.gz file ([BTAO]).

From the tar.gz we will use the zimbra script which we will copy into:

```
/usr/lib/ocf/resource.d/btactic
```

We will use a temporary directory in order to use it:

```
mkdir /tmp/temp
cd /tmp/temp
```

We download and extract it:

```
wget "http://www.btactic.org/btactic_ocf_0.0.2.tar.gz"
tar xzf btactic_ocf_0.0.2.tar.gz
```

Make the btactic resource directory and copy zimbra file in there:

```
mkdir --parents /usr/lib/ocf/resource.d/btactic
cp zimbra /usr/lib/ocf/resource.d/btactic
```

We finally make sure to give the script executable permissions:

```
chmod +x /usr/lib/ocf/resource.d/btactic/*
```

10.4 Pacemaker final setup

As explained in *section 10.1 About Pacemaker* Pacemaker is High Availability resource manager, here we will explain how our setup pretends to manage our two servers resources. These instructions should only be performed on **primary** node.

We need to introduce resource stickiness concept. Resource stickiness controls how much a service prefers to stay running where it is. You may like to think of it as the *cost* of any downtime. By default, Pacemaker assumes there is zero cost associated with moving resources and will do so to achieve *optimal* resource placement ([PMCS]).

These are the main settings we define in our configuration:

- Setup deletes prior configuration.
- DRBD, Filesystem mount and Zimbra Server are setup to work in the same server as they work as in a team.
- System stickiness is changed so that Zimbra Server resource is not moved from where it is running to avoid Zimbra unnecessary downtimes.
- System are forced to be started in the right order. The right order is: DRBD, Filesystem mount, and Zimbra Server.
- We disable stonith (definition on subsection 12.2.2 Fencing) in order to simplify the setup.
- Primary server will be the preferred server where resources need to be run.

We make sure that */tmp/zimbrapacemaker.config* file contents are:

```
configure
erase
node primary
node secondary
# Activate failover
property no-quorum-policy=ignore
# Disable stonith
property stonith-enabled=false
# Resource stickiness
rsc_defaults resource-stickiness=100
# Public ip fail over check
primitive ClusterIP ocf:heartbeat:IPaddr2 \
params nic=eth0 ip=192.168.77.203 \
```

```
cidr_netmask=24 \
broadcast=192.168.77.255 \
op monitor interval=30s
# Configure zimbra resource
primitive ZimbraServer ocf:btactic:zimbra op \
monitor interval=2min timeout="40s" \
op start interval="0" timeout="360s" \
op stop interval="0" timeout="360s"
# Prefered location: primary node
location prefer-primary-node \
ZimbraServer 50: primary
# Define DRBD ZimbraData
primitive ZimbraData ocf:linbit:drbd params \
drbd_resource=zimbradata op monitor \
role=Master interval=60s op monitor \
role=Slave interval=50s \
op start role=Master interval="0" timeout="240" \
op start role=Slave interval="0" timeout="240" \
op stop role=Master interval="0" timeout="100" \
op stop role=Slave interval="0" timeout="100"
# Define DRBD ZimbraData Clone
ms ZimbraDataClone ZimbraData meta \
master-max=1 master-node-max=1 \
clone-max=2 clone-node-max=1 notify=true
# Define ZimbraFS so that zimbra can use it
primitive ZimbraFS ocf:heartbeat:Filesystem \
params device="/dev/drbd/by-res/zimbradata" \
directory="/opt" fstype="ext4" \
op start interval="0" timeout="60s" \
op stop interval="0" timeout="60s"
group MyZimbra ZimbraFS ZimbraServer


# Everything in the same host
colocation everything-together \
inf: MyZimbra \
ZimbraDataClone:Master ClusterIP

# Everything ordered
order everything-ordered \
inf: \
```

```
ClusterIP \
ZimbraDataClone:promote MyZimbra
commit
```

In order to apply the configuration we will run:

```
crm < /tmp/zimbrapacemaker.config
```

Pacemaker configuration is not trivial. The main document from which the configuration file was adapted and written was Clusters from Scratch ([PMCS]).

Chapter 11

High Availability System Management

This chapter describes some common management tasks that can be used in High Availability systems like ours.

11.1 Introduction

Most of these examples have been adapted from Clusters From Scratch document ([PMCS]).

At Zimbra Forums Zimbra on Pacemaker + DRBD howto thread ([ZFTA]) we can find a Zimbra High Availability Howto ([HAZ8]) where more every-day uses of Pacemaker examples are shown.

The reason why we need these examples is that, instead of managing a single server installation Zimbra server system, now we need to manage a high availability system which has been initially setup by Pacemaker. Some of the most useful management tasks will be described.

11.2 DRBD Split Brain recovery

Split brain is a situation where, due to temporary failure of all network links between cluster nodes, and possibly due to intervention by a cluster management software or human error, both nodes switched to the primary role while disconnected. This is a potentially harmful state, as it implies that modifications to the data might have been made on either node, without having been replicated to the peer. Thus, it is likely in this situation that two diverging sets of data have been created, which cannot be trivially merged ([DSBN]).

This is an example of *cat /proc/drbd* output:

```
version: 8.3.11 (api:88/proto:86-96)
srcversion: 93CE421BB73A731BDC72D8E
 0: cs:WFConnection ro:Primary/Unknown
    ds:UpToDate/DUnknown C r-----
    ns:0 nr:0 dw:0 dr:1977 al:0
      bm:0 lo:0 pe:0 ua:0 ap:0
        ep:1 wo:f oos:153220
```

where there is an split brain.

In order to fix it we can decide that primary node contents will prevail and that secondary node contents will be discarded. First of all we will need to stop the cluster on **both nodes** thanks to:

```
service corosync stop
```

.

Then we need to start drbd service manually in **both nodes** thanks to:

```
service drbd start
```

First of all **in secondary node** we will discard its data with:

```
drbdadm secondary zimbradata
drbdadm -- --discard-my-data connect zimbradata
```

Then we will need to run **in primary node**:

```
drbdadm connect zimbradata
```

Once we check that *cat /proc/drbd* has a non split brain state like:

```
version: 8.3.11 (api:88/proto:86-96)
srcversion: 93CE421BB73A731BDC72D8E
 0: cs:Connected ro:Primary/Secondary
   ds:UpToDate/UpToDate C r-----
    ns:5691029 nr:0 dw:0 dr:6002290
      al:0 bm:367 lo:0 pe:0 ua:0
        ap:0 ep:1 wo:f oos:0
```

we can restart the cluster with running (**in both nodes**):

```
service drbd stop
service corosync start
```

so that drbd is not handled manually and the cluster takes care of it.

11.3 Host down simulation

These commands simulate that a host is down by stopping both pacemaker and corosync services. We will run them only in the primary server. The secondary server is supposed to take control of the cluster system and start Zimbra.

```
service pacemaker stop
service corosync stop
```

11.4 Node recover

In order to simulate a node recover we will start both pacemaker and corosync services in primary server:

```
service corosync start
service pacemaker start
```

As per our our cluster system stickiness Zimbra service will stay in secondary server.

11.5 Resources check

In order to check the overall Cluster status we just run the cluster management monitor command:

```
crm_mon
```

One example of crm_mon output where the cluster has not problems at all is:

```
============
Last updated: Sun Sep  8 23:28:38 2013
Last change: Sun Sep  8 23:24:33 2013 via cibadmin on primary
Stack: openais
Current DC: secondary - partition with quorum
Version: 1.1.6-9971ebba4494012a93c03b40a2c58ec0eb60f50c
2 Nodes configured, 2 expected votes
5 Resources configured.
============

Online: [ secondary primary ]
```

```
Master/Slave Set: ZimbraDataClone [ZimbraData]
    Masters: [ primary ]
    Slaves: [ secondary ]
Resource Group: MySystem
    ClusterIP  (ocf::heartbeat:IPaddr2):       Started primary
Resource Group: MyZimbra
    ZimbraFS   (ocf::heartbeat:Filesystem):    Started primary
    ZimbraServer       (ocf::btactic:zimbra):  Started primary
```

11.6 Move cluster resources temporarily

In the case that we want to move *temporarily* resources to primary server we should run:

```
crm resource move ZimbraServer primary
```

11.7 Revert cluster resources movement

If we want to return Resource control to the cluster we can run:

```
crm resource unmove primary
```

In our setup, thanks to our defined stickiness the cluster will not perform any resource movement.

11.8 Migration testing

We can simulate the migration by declaring a node in standby. This way the standby node hardware can be fixed. We just have to run:

```
crm node standby
```

in the affected node.

In order to check the migration status we can run:

```
crm_mon
```

.

This is a crm_mon output while node is migrating from primary node to secondary node:

```
============
Last updated: Sun Sep  8 23:33:41 2013
Last change: Sun Sep  8 23:31:50 2013
            via crm_attribute on primary
Stack: openais
Current DC: secondary - partition with quorum
Version: 1.1.6-9971ebba4494012
              a93c03b40a2c58ec0eb60f50c
2 Nodes configured, 2 expected votes
5 Resources configured.
============

Node primary: standby
Online: [ secondary ]

 Master/Slave Set: ZimbraDataClone [ZimbraData]
     Masters: [ secondary ]
     Stopped: [ ZimbraData:1 ]
 Resource Group: MySystem
     ClusterIP  (ocf::heartbeat:IPaddr2):
                           Started secondary
 Resource Group: MyZimbra
     ZimbraFS   (ocf::heartbeat:Filesystem):
                           Started secondary
     ZimbraServer       (ocf::btactic:zimbra):
                           Stopped
```

as we can see both ClusterIP and ZimbraFS have already started on secondary node and ZimbraServer is not started in secondary node yet.

Finally once we have fixed the hardware we can declare the node online again thanks to:

```
crm node online
```

.

Once again in our setup, thanks to our defined stickiness the cluster will not perform any resource movement.

11.9 Starting and stopping resources

Sometimes, mainly for debugging purposes, is needed to start or stop cluster resources manually.

E.g. in order to start ZimbraFS resource we will issue:

```
crm resource start ZimbraFS
```

In order to stop the same resource we will issue:

```
crm resource stop ZimbraFS
```

Chapter 12

Conclusions and future work

This chapter draws some conclusions and describes some of the improvements that can be applied to our High Availability system. These improvements can be used in further research.

12.1 Conclusions

We have shown, thanks to a reproducible example, that Zimbra OSE could be enhanced to be high available. This system has been based on state of the art open source high availability software such as Pacemaker and Corosync.

The master thesis writer has learnt how to develop OCF Resource Agents which can be useful for other HA systems. How to mirror a partition between two servers thanks to DRBD was also covered. More over, Zimbra, one of the easiest email server solutions in its open source edition, can now be used as a HA system with standard HA software.

12.2 Future work

12.2.1 OVH Datacentre network handling

There have been some efforts from the bTactic team to handle OVH Datacentre networking thanks to three OCF scripts named: ClusterOVHFailover, ClusterHostRoute and ClusterDefaultRoute. These scripts make sense in setup that does not use Virtual Rack + RIPE but just normal servers connected via Internet only.

ClusterOVHFailover makes sure an OVH ip-fail-over is failed over from one machine to the another one. This way Zimbra Server is being served by the correct host. ClusterHostRoute and ClusterDefaultRoute are meant to help to setup OVH

networking for ip-fail-over which is not covered by standard network setup found in default Pacemaker package.

These scripts can be found inside BtacticOCF tar.gz file ([BTOC]) which can be downloaded from BtacticOCF tar.gz file ([BTAO]).

12.2.2 Fencing

Fencing is the process of locking resources away from a node whose status is uncertain ([LHFE]). The default method of fencing in Pacemaker is using stonith. Stonith is a technique for node fencing, that means that the node (either primary or secondary server in our example) that it is supposed to have failed is *shot in head* ([LHST]). That makes sure that the node is actually dead.

In our described example we have disabled stonith. We can improve it by enabling it and using one of the available methods.

Once again bTactic team has developed an stonith script (or fence agent as known per Pacemaker) to make sure an OVH server does not use a shared resource. The failing node is rebooted into a Rescue mode which is quite similar to booting a computer with a live CD that does not make any change to local hard disks. Fence agent name is: fence_ovh.

This script is available as a Red Hat's fence-agents package since fence-agents package 4.0.2 release version ([LCML]).

12.2.3 Mysql HA

One of the Zimbra components is a Mysql database. This Mysql database can be excluded from DRBD syncing and can be setup as an active / active cluster.

This setup will offer a better handling of node failing because you would only loose last mysql queries. In the DRBD case you loose all the file changes that have not been stored into the files that compose actual Mysql database.

The only drawback for this improvement is that Zimbra OSE upgrades tend to be much more difficult when we want to preserve HA.

12.2.4 Project Always ON

Always ON is a project from Zimbra ([ZPAO]) (September 2013) with a very simple goal: Email and collaboration should be *always on* for end users. Its design goals are:

- Inherently resilient to failure
- Scaling should be elastic based on workload demands

- The software can be enhanced without service disruption
- Efficient usage of commodity hardware resources

. That design goals imply:

- No single points of failure in the application components
- Separating the application code from the data
- Distributing state information across commodity storage
- Automatic failover of application and data storage components
- Automatic load balancing of client requests across the application and data layers

Project Always ON has just started and might be implemented as early as in Zimbra 10 version. It is an improvement over Mysql HA future work described on subsection 12.2.3 Mysql HA because not only enforces HA on each of the Zimbra components but it is also focused on scaling resources. That means, that the more service is needed the more Zimbra (virtual) machines will be deployed to meet that service requirements.

12.2.5 Data loss

Additional tests can be performed to measure how much data is lost is when one of the nodes is offline while having an active role. We can find some of these tests at [HAZ8].

Appendix A

GNU Free Documentation License

GNU Free Documentation License

Version 1.3, 3 November 2008

Copyright (C) 2000, 2001, 2002, 2007, 2008 Free Software Foundation, Inc. ¡http://fsf.org/¿

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful document ”free” in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondarily, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of ”copyleft”, which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The ”Document”, below, refers to any such manual or work. Any member of the pub-

lic is a licensee, and is addressed as ”you”. You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A ”Modified Version” of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A ”Secondary Section” is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document’s overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The ”Invariant Sections” are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The ”Cover Texts” are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A ”Transparent” copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not ”Transparent” is called ”Opaque”.

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-

generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The ”Title Page” means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, ”Title Page” means the text near the most prominent appearance of the work’s title, preceding the beginning of the body of the text.

The ”publisher” means any person or entity that distributes copies of the Document to the public.

A section ”Entitled XYZ” means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as ”Acknowledgements”, ”Dedications”, ”Endorsements”, or ”History”.) To ”Preserve the Title” of such a section when you modify the Document means that it remains a section ”Entitled XYZ” according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document’s license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as

long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.

B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.

C. State on the Title page the name of the publisher of the Modified Version, as the publisher.

D. Preserve all the copyright notices of the Document.

E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.

F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.

G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.

H. Include an unaltered copy of this License.

I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.

J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.

K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.

L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.

M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.

N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.

O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties–for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of,

you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled ”History” in the various original documents, forming one section Entitled ”History”; likewise combine any sections Entitled ”Acknowledgements”, and any sections Entitled ”Dedications”. You must delete all sections Entitled ”Endorsements”.

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an ”aggregate” if the copyright resulting from the compilation is not used to limit the legal rights of the compilation’s users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, or distribute it is void, and will automatically terminate your rights under this License.

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, receipt of a copy of some or all of the same material does not give you any rights to use it.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU

Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See http://www.gnu.org/copyleft/.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation. If the Document specifies that a proxy can decide which future versions of this License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Document.

11. RELICENSING

"Massive Multiauthor Collaboration Site" (or "MMC Site") means any World Wide Web server that publishes copyrightable works and also provides prominent facilities for anybody to edit those works. A public wiki that anybody can edit is an example of such a server. A "Massive Multiauthor Collaboration" (or "MMC") contained in the site means any set of copyrightable works thus published on the MMC site.

"CC-BY-SA" means the Creative Commons Attribution-Share Alike 3.0 license published by Creative Commons Corporation, a not-for-profit corporation with a principal place of business in San Francisco, California, as well as future copyleft versions of that license published by that same organization.

"Incorporate" means to publish or republish a Document, in whole or in part, as part of another Document.

An MMC is "eligible for relicensing" if it is licensed under this License, and if all works that were first published under this License somewhere other than this MMC, and subsequently incorporated in whole or in part into the MMC, (1) had no cover texts or invariant sections, and (2) were thus incorporated prior to November 1, 2008.

The operator of an MMC Site may republish an MMC contained in the site under CC-BY-SA on the same site at any time before August 1, 2009, provided the MMC is eligible for relicensing.

ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

Copyright (c) YEAR YOUR NAME.

Permission is granted to copy, distribute and/or modify this document
under the terms of the GNU Free Documentation License, Version 1.3

Appendix B

Zimbra OCF source code

```
#!/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|reload\
|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="reload"  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 already running."
        return $OCF_SUCCESS
    fi

    # start Zimbra
    command startup
    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 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 shutdown
    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_reload()
{
    if zimbra_status; then
        ocf_log info "Reloading Zimbra."
```

```
        command reload
    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 $?
                ;;

    reload)     zimbra_reload
```

```
                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
```

Bibliography

[HANC] High Availability Cluster - Node configurations - Wikipedia Article http://en.wikipedia.org/wiki/High-availability_cluster#Node_configurations. Last access 2013-09-12.

[WIHA] High Availability Wikipedia Article http://en.wikipedia.org/wiki/High_availability. Last access 2013-08-20.

[ATFC] IEEE Task Force on Cluster Computing. archive.org capture from February 2011. http://web.archive.org/web/20110216113021/http://www.ieeetfcc.org/high-availability.html. Last access 2013-08-20.

[BCHA] High Availability Bcmpedia definition http://www.bcmpedia.org/wiki/High_Availability. Last access 2013-08-20.

[ZLEA] Zimbra - Learn http://www.zimbra.com/learn/. Last access 2013-08-20.

[VWZ8] Zimbra Blog - Announcing The General Availability of Zimbra 8 http://blog.zimbra.com/blog/archives/2012/09/announcing-the-general-availability-of-zimbra-8.html. Last access 2013-08-20.

[UWHA] Ubuntu Official Documentation - What is Ubuntu? https://help.ubuntu.com/lts/installation-guide/amd64/what-is-ubuntu.html. Last access 2013-09-11.

[UMIN] Ubuntu Community Help Wiki - Installation - MinimalCD https://help.ubuntu.com/community/Installation/MinimalCD. Last access 2013-08-20.

[LDIN] LinBit DRBD version 8.4 Guide - Chapter 17 . DRBD Internals http://www.drbd.org/users-guide/ch-internals.html. Last access 2013-08-20.

[ZWPO] Zimbra Wiki - Ports http://wiki.zimbra.com/wiki/Ports. Last access 2013-09-06.

[ZWWW] Zimbra - Official web page http://www.zimbra.com. Last access 2013-09-11.

[VWZD] Zimbra - Open Source Edition Downloads http://www.zimbra.com/downloads/os-downloads.html. Last access 2013-08-20.

[ZPTG] Zimbra Wiki - Performance Tuning Guidelines for Large Deployments http://wiki.zimbra.com/wiki/Performance_Tuning_Guidelines_for_Large_Deployments. Last access 2013-09-12.

[LDWI] Linbit DRBD - What is DRBD http://www.drbd.org/home/what-is-drbd/. Last access 2013-08-20.

[LDDC] drbd.conf Configuration file for DRBDś devices http://www.drbd.org/users-guide/re-drbdconf.html. Last access 2013-08-20.

[WCSA] Corosync project Wikipedia Article http://en.wikipedia.org/wiki/Corosync_(project). Last access 2013-08-20.

[CORW] Corosync web page http://corosync.github.io/corosync/. Last access 2013-08-20.

[CLPW] Cluster Labs - The Home of Pacemaker http://clusterlabs.org/. Last access 2013-08-20.

[LHRA] Linux High Availability Wiki - Resource Agents http://linux-ha.org/wiki/Resource_Agents. Last access 2013-08-20.

[LORA] Linux High Availability Wiki - OCF Resource Agents http://linux-ha.org/wiki/OCF_Resource_Agents. Last access 2013-08-20.

[BTAO] Btactic Org web page http://www.btactic.org. Last access 2013-08-20.

[BTOC] Btactic OCF Resource Agents for High Availability Version 0.02 http://www.btactic.org/btactic_ocf_0.0.2.tar.gz. Last access 2013-08-20.

[PMCS] Pacemaker 1.1 for Corosync 2.x and crmsh - Cluster from Scratch http://clusterlabs.org/doc/en-US/Pacemaker/1.1-crmsh/html/Clusters_from_Scratch/index.html. Last access 2013-08-20.

[ZFTA] Zimbra Forums - Zimbra on Pacemaker + DRBD howto - Page 2 http://www.zimbra.com/forums/administrators/58113-zimbra-pacemaker-drbd-howto-2.html#post251023. Last access 2013-08-20.

[HAZ8] How-to HA with Zimbra OSE https://www.dropbox.com/s/8w4f7koz9ym3cle/Howto_HA_Zimbra8.pdf. Last access 2013-08-20.

[DSBN] Split brain notification and automatic recovery http://www.drbd.org/users-guide-8.3/s-split-brain-notification-and-recovery.html. Last access 2013-09-08.

[LHFE] Linux High Availability Wiki - Fencing Article http://linux-ha.org/wiki/Fencing. Last access 2013-08-20.

[LHST] Linux High Availability Wiki - Stonith Article http://linux-ha.org/wiki/STONITH. Last access 2013-08-20.

[LCML] [Linux-cluster] fence-agents-4.0.2 stable release http://www.redhat.com/archives/linux-cluster/2013-July/msg00028.html. Last access 2013-08-20.

[ZPAO] Zimbra Blog - Project Always On http://blog.zimbra.com/blog/archives/2013/09/project-always-on.html. Last access 2013-09-12.