



Master in Free Libre Open Source Software

Academic Course 2014/2015

Master Thesis

Implementation of a high availability solution based
on Free Libre Open Source Software tools for
Netnovation's Email and Collaboration System

Author: DANIEL H. GÁMEZ V.

Tutor: DR. GREGORIO ROBLES

(c) 2015, Daniel H. Gámez V.
daniel.gamez@gmail.com

This work is licensed under a
Creative Commons Attributions 3.0 License



<http://creativecommons.org/licenses/by-sa/3.0/legalcode>

Abstract

This is the Abstract...

Key words: Cluster, Corosync, CRM, DRBD, FOSS, FLOSS, High Availability, OCF, Pace-maker, Zimbra

Acknowledgements

These are the Acknowledgements..

Contents

Abstract	3
Acknowledgements	5
1 Introduction	13
2 Problem statement	15
2.1 Justification / Motivation	16
2.2 Objectives	16
2.3 Scope	16
3 Related technologies	17
3.1 Commercial Enterprise Cluster Software	17
3.1.1 HP Serviceguard	17
3.1.2 Red Hat Cluster Suite	19
3.1.3 SUSE Linux Enterprise High Availability Extension	20
3.2 High availability FLOSS based tools	21
4 Methodology	23
4.1 OpenBRR	23
5 Architecture	25
5.1 Company Infrastructure	25
5.2 Existent Hardware	25
5.3 Company Network Scheme	26
5.4 Software Supporting the Infrastructure	26
6 Technological background	29
7 Implementation	31
7.1 Operating system considerations	31
7.1.1 FQDN hostnames and IP addresses	32
7.1.2 Network	33
7.1.3 ZCS dependencies	35
7.2 DRBD	35
7.2.1 Initial configuration	35
7.2.2 DRBD Split Brain Recovery	36

7.3	ZCS	37
7.3.1	ZCS full install on primary node	37
7.3.2	ZCS dummy install on secondary node	38
7.4	OCF	39
7.5	Pacemaker	39
7.6	Control and check services	42
7.7	Testing failover	42
8	Results and discussion	45
9	Conclusions and future work	47
	Bibliography	49
A	Title of appendix 1	51

List of Figures

3.1	High-availability race in enterprise environments	18
3.2	Basic HP Serviceguard for Linux cluster	19
5.1	Network interconnection scheme	26
6.1	CCS overview	30
7.1	Two nodes HA cluster	32

List of Tables

5.1	Software Supporting the Infrastructure	27
7.1	Operating system configuration	32
7.2	FQDN hostnames and IP addresses	32
7.3	/etc/hosts file	32
7.4	/etc/sysconfig/network-scripts/ifcfg-eth0 file	33
7.5	/etc/ntp.conf file	34
7.6	/etc/named.conf file	34
7.7	/etc/named.conf file	34
7.8	/etc/drbd.d/optzimbra.res file	35
7.9	/etc/yum.repo.d/centos.repo file	39
7.10	DRBD failover test	43
7.11	DRBD synced status	43

Chapter 1

Introduction

With the growing use of cloud-oriented systems and the need for this information to be always available, online systems play an increasingly important role in our society nowadays. The technologies that support such schemes have evolved fleetingly and today there are numerous ways to get this kind of solutions, from proprietary software implementations, tools oriented to corporate environments that can be even based on FLOSS products and a variety of business models, to FLOSS standalone tools representing a robust solution to meet today's demands.

Since 2004 Netnovation™ is a Venezuelan SME formed by a team of professionals in the areas of IT and telecommunications who adopted a business model based on consulting around FLOSS, providing system integration, timely development and Software as a Service (SaaS) cloud services.

Due to a number of reasons that will be discussed throughout this dissertation, cloud services require a set of components to ensure the security, availability and reliability of data that is stored in data center facilities remotely accessible from internet. The study conducted here is focused specifically on the availability of the data that must be accessible to their applicants at all times.

This study was undertaken in the one hand to test some aspects of business practices in the current technology market and the use of FLOSS as a key factor, and in the other one, to fulfill the demands of a business organization showing that business models around FLOSS are a fact.

The way this dissertation is going to be organized is the following:

Chapter 2 with the problem statement, setting why there is an issue in the current situation, what are the justification and motivation of this dissertation, also defining the objectives and the proper scope.

Chapter 3 describing the most relevant related technologies around the possible solutions.

Chapter 4 establishing the used methodology, supported by the OpenBRR and the Lazy User models.

Chapter 5 showing the architecture of the company and its infrastructure, available hardware for operations, the network scheme, and an overview of the software supporting the current situation, all of this in order to understand how to adapt an actual solution.

Chapter 6 describing the implemented technologies to achieve the solution for the problem stated.

Chapter 7 will provide a detailed description of the actual implementation for the proposed solution.

Chapter 8 with the results and discussion around the subject.

Finally in Chapter 9 the conclusions and future work will be raised.

Chapter 2

Problem statement

Business continuity in the field of information technology is supported in a large extent by the uninterrupted operation of the systems used in productivity tasks [3]. These systems must be fault tolerant, so that operations have the least possible impact in the event that an unexpected incident occurs.

Nowadays there are increasingly more people and organizations using centralized remote systems that allow online access to resources and everyday services, this scheme is called cloud computing [6]. Through this type of services, end users whether individuals or corporations, are abstracted to support the infrastructure that this entails, giving responsibility to intermediary companies providing cloud services. So these intermediaries are the ones who must ensure the proper availability of the services, as well as factors such as communications security and redundancy of stored data, among many others.

In particular Netnovation is a SME in the field of information technologies, which provides private cloud services from data storage to hosting virtual private servers (VPS), including e-mail and collaboration servers. The latter is precisely one of the mainstays for the operations of the company, which employs mainly FLOSS to its internal systems, specifically using the FLOSS e-mail and collaboration suite Zimbra™¹. One of the main problems that Netnovation faces is to ensure the communication and workflow continuity that is carried through this collaboration tool, as well as meet the SLAs offered to its customers over this software.

There are various software solutions offering high availability of services such as those provided by Zimbra, each one with its legal implications, associated costs and implementation difficulty². A valid alternative is the integration of multiple tools in the field of FLOSS providing a framework to ensure continuity of systems operation or business continuity. By doing it this way it is possible to adapt the different requirements and use different technologies to provide the most consistent solution to what is desired.

On previous occasions, Netnovation has managed to successfully consolidate most of its operations infrastructure adapting FLOSS, making it a wonderful idea to keep this scheme working. To achieve this, it is necessary to evaluate the state of the art in the field of systems that provide high availability, with the aim of offering an effective solution, all of this in accordance to the guidelines that have been proposed by the company.

¹ <http://www.zimbra.com>

² Some of these solutions will be addressed in Chapter 3

2.1 Justification / Motivation

The factors that motivates this work are on one hand, give proper credit to business models based on FLOSS as those used by technology companies nowadays [2], and on the other hand, show that private enterprise can be benefited by FLOSS, through a set of toolsets and mechanisms who allow obtaining robust solutions in accordance to technology needs.

2.2 Objectives

The overall objectives are:

- To frame the FLOSS business model used by Netnovation
- To show various current alternatives provided by FLOSS at the corporate level
- To adapt the proposed solution to the guidelines established by Netnovation
- To establish an initial point reference for implementing high availability private cloud services offered by Netnovation

The specific objectives are:

- To implement a high availability solution based on FLOSS for the e-mail and collaboration system Zimbra used by Netnovation
- To describe the methodology used for the selection of the solution to be implemented
- To describe the process undertaken to implement the selected solution
- To perform tests in a controlled laboratory environment and validate the correct operation of the solution in order to promote it to a production environment

2.3 Scope

The solution to be implemented consists of FLOSS tools that allow its adaptation to the current infrastructure of Netnovation, they are not intended to replace the elements of the existing operations platform.

The methodology used for the selection of FLOSS tools that make the proposed solution is not intended to provide an exhaustive process that considers all possibilities in the area, but a flexible way that allows classify them qualitatively, justifying their choice through concrete metrics.

Having successfully implemented a high availability solution on the e-mail and collaboration system used by Netnovation, this will serve as a reference for providing high availability to other enterprise systems, but these other configurations are not covered in this exercise.

Chapter 3

Related technologies

These high-availability technologies could be categorized as Enterprise Solutions and FLOSS based tools, considering that usually the first ones have associated a complete support service over the whole provided solution, whereas the second ones provide support over its own tools, but not necessarily over the whole cluster implementation.

3.1 Commercial Enterprise Cluster Software

Since the beginning many of these high-availability implementations have been proprietary, either way at some point commercial companies realized that these technologies could be integrated into open systems thanks to licenses such as GNU/LGPL. Since then, as a business model strategy, companies such as Hewlett Packard, Red Hat Inc. and SUSE (a Novell company) provide assistance services considering their entire solution, charging for business support licenses, generally on annual basis. Examples of this situation, are in the following sub-sections.

3.1.1 HP Serviceguard

Hewlett Packard (HP) claims the credit for the development of the first high availability solution for UNIX since 1990¹. MC/ServiceGuard, is a high-availability cluster software released for HP-UX and later for GNU/Linux systems. Since the first development of the software, HP has competed with companies such as Oracle or SAP to deploy high-availability in enterprise environments, as Figure 3.1 (took from <http://www.hpintelco.net>) shows.

With the appearance of Linux in 1994 and its increasing popularity together with GNU, in 1999 HP released a Linux port called SG/LX, allowing high-availability features on it.

Since 2001, companies Intel, Red Hat and HP joined efforts to produce the Red Hat Open Source Solutions Initiative² (OSSI), with the aim to reduce partner's sale cycle by delivering enterprise reliable solutions to their customers. To some extent, realizing that their technologies could be strengthened by FLOSS.

Some technical specifications of this product are the following:

¹<http://www.hpintelco.net/sglx/service.html>

²<http://www.hpintelco.net/hp-intel-redhat.htm>

1990	First high availability (HA) solution for UNIX
1992	New major release of High Availability solution
1994	First Oracle RAC HA solution
1997	SAP R/3 Zero downtime joint initiative kick off
1998	First automated disaster recovery solution
1999	<ul style="list-style-type: none"> • HP introduced ContinentalClusters • HP launched SGLX extension for SAP : SGeSAP
2000	First graphical cluster management with Serviceguard Manager
2005	SGLX integration with workload management & utility pricing
2006	SGLX integration with virtual machines, capacity planning and GiCAP
2007	First HA for SAP liveCache
2009	Fast Failover;Online everything initiative
2010	Simplified availability; new solutions for Oracle EBS and Data Guard
2012	SG/LX introduced on Red Hat Enterprise Linux

Figure 3.1: High-availability race in enterprise environments

- Proprietary Licensing Model³
- Active/active, active/standby, and rotating standby cluster types
- Quorum Server support
- 32 Fibre Channel nodes, 2 single-path SCSI nodes, and 32 multipath SCSI nodes
- Operating Systems RHEL and SLES
- Hardware HP ProLiant ML, DL, and BL G7, Gen8 and Gen9 servers
- HP 3PAR, EVA, StoreSure, EMC VMAX, VNX storage
- ext3, ext4, NFS, XFS, VxFS, VxVM and btrfs filesystems
- Logical Volume Manager

The general overview of an HP Serviceguard architecture is shown in Figure 3.2 (took from data sheet HP Serviceguard Solutions for Linux⁴).

³http://h20564.www2.hp.com/hpsc/doc/public/display?docId=emr_na-c02199685

⁴http://h20195.www2.hp.com/v2/GetDocument.aspx?docname=4AA4-1792ENW&doctype=data%20sheet&doclang=EN_US

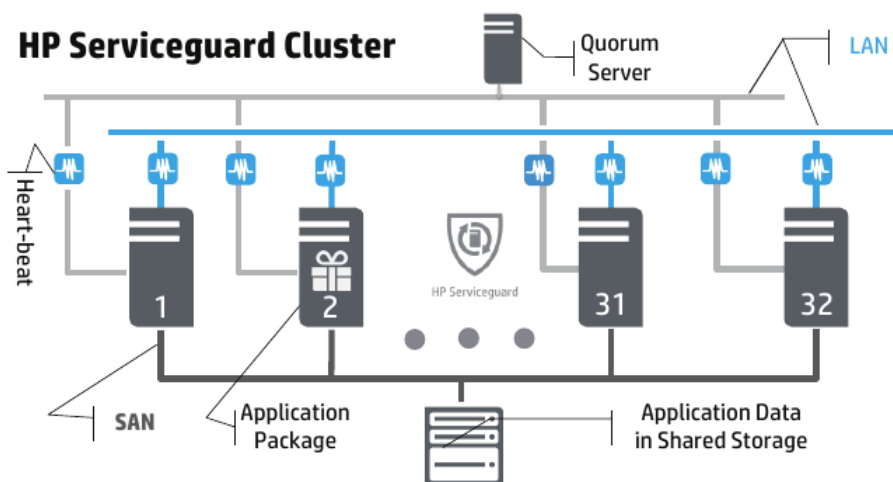


Figure 3.2: Basic HP Serviceguard for Linux cluster

3.1.2 Red Hat Cluster Suite

At the end of the 90's Red Hat Inc. introduced its Red Hat Enterprise Linux Advanced Server product, designed specifically for use in enterprise environments to deliver superior application support, performance, availability and scalability. It included a high-availability clustering feature as part of the base product. Since then, the product has evolved to the current Red Hat Cluster Suite (RHCS), provided as a separately licensed product, also on top of Red Hat's base Linux Enterprise Server license.

The RHCS has two major features, one is the Cluster Manager that provides high availability, and the other is the IP Load Balancing (originally called Piranha). The first is Red Hat's implementation of Linux-HA, which will be discussed in Section 3.2.

On the other hand Piranha was originally developed by researchers at Oak Ridge National Laboratory⁵, basically a text mining technology that Red Hat adapted in order to allow transparent load balancing and failover between servers. The software Piranha is under copyright and count with its own patents, but the organization allow strategic partnerships, evaluating deals based on the ability and commitment to successfully commercialize the technology⁶.

The Cluster Manager and IP load balancing are complementary high-availability technologies that can be used separately or in combination, depending on application requirements, and both of these technologies are integrated in RHCS.

Some technical details of this product are the following:

- Support for up to 128 nodes (16 nodes on Red Hat Enterprise Linux 3, 4, 5, and 6)
- NFS, CIFS, GFS share and cluster filesystem managers

⁵<http://www.ornl.gov/>

⁶<http://www.ornl.gov/connect-with-ornl/for-industry/partnerships/technology-licensing/licensing-opportunity-announcements/piranha>

- File system and services failover support
- Fully shared storage subsystem
- Comprehensive data integrity
- SCSI and fiber channel support
- OCF and LSB resource agents

3.1.3 SUSE Linux Enterprise High Availability Extension

This is an integrated suite of clustering technologies that enables the implementation of high availability over physical and virtual Linux clusters [7]. It allows monitoring, messaging, and cluster resource management, handling failover and load balancing of resources.

As Red Hat did, other companies such as Novell relies on SG/LX Linux port to provide high availability solutions. This product is available as a paid add-on to SUSE Linux Enterprise Server GNU/Linux distribution, although in OpenSUSE many of these tools are included into the base system for free (without charge), with available repositories on OBS to provide newer versions of the packages for various GNU/Linux distributions⁷.

Among the main product features of this product are the following:

- Multiple clustering scenarios, as active/active and active/passive configurations, as well as hybrid physical and virtual clusters
- Supports mixed clustering, physical and virtual Linux servers, based on Xen and KVM hypervisors
- Corosync messaging and membership layer, also Pacemaker cluster resource manager
- Storage and data replication supporting Fibre Channel or iSCSI SAN
- Cluster-aware file systems with GFS and OCFS, and cLVM as volume manager
- Supports replication through DRBD
- Samba clustering with CTDB
- Provides resource agent manager OCF
- GUI and CLI administration tools, such as YaST, HAWK and CRM

⁷https://en.opensuse.org/openSUSE:High_Availability

3.2 High availability FLOSS based tools

These tools are possibly those that conform some of the enterprise solutions mentioned before, but are not supported as a complete enterprise cluster solution. Either way, most of these FLOSS technologies are backed by professional support over the standalone software provided. Examples of this tools, are mentioned below.

Linux-HA has been released since 1999 under GNU GPL and GNU LGPL license, maintains a set of building blocks for high availability cluster systems, including a cluster messaging layer, a bunch of resource agents for a variety of applications, and a plumbing library and error reporting toolkit. Around 2007 this project evolved to Pacemaker, integrating or allowing interaction with multiple FLOSS cluster stacks such as Heartbeat, Corosync and OCF.

Nowadays Pacemaker is a resource manager responsible for starting and stopping cluster services in a proper way. Combined with other tools is able to detect service-level failures and move resources between cluster nodes as needed, to ensure the smooth operation of the services.

Corosync is responsible for cluster membership, message passing and quorum, using the totem protocol for *heartbeat*, monitoring other node's health.

DRBD works on top of block devices, such as hard disk partitions or LVM logical volumes, mirroring each data block that it is written to disk to the peer node.

Chapter 4

Methodology

The followed roadmap to achieve the objectives outlined is a set of guidelines and suggestions for the adoption of FLOSS within SMEs [1], in the sense that a methodology is not an exact formula but a set of practices. At using this model, companies find a supporting guide from the initial selection and adoption of FLOSS within the IT infrastructure and even to the consolidation of business models around open source.

On the one hand, the guidelines proposed by Daffara suggest a research method by collecting and read as much information related to the project is available and select the appropriate solution from a matching set that fulfill the requirements. On the other hand this dissertation also contemplates an open and standard framework to select the technologies to be used, this is the light weight model OpenBRR [8], which performs a quick and flexible assessment over software tools considered.

Additionally, and following a practical formal approach, applying a model of technology acceptance such as the Lazy User Model (LUM) [4] is possible to frame the process by which are chosen the technological tools that will make up the solution that meets user requirements, which in this case is being represented by the company Netnovation. This model focuses on user needs and the demanded effort when selecting a solution to a problem from a set of possible solutions, “According to the lazy user model, a user is likely to choose the solution that requires the least effort. The user examines this cost in terms of time, energy and money when considering how to use a new solution.” [5] The LUM proposes that technology acceptance is impacted by this principle.

4.1 OpenBRR

Chapter 5

Architecture

In order to provide IT services to customers, considering software services hosted in an on-line remote location, Netnovation requires a proper software and hardware infrastructure to operate. Some of these services are from VPS, Data Storage Systems, Customer Relationship Management Systems (CRM), Email and Collaboration Systems, to Voice over IP PBX systems. Both parties, software infrastructure and product services offered are based on FLOSS.

In particular the Zimbra server to which a high availability schema is been configured, resides into this architecture, and it is consistent with the company's principles and business model, which is why it is useful to understand the environment to which it belongs.

5.1 Company Infrastructure

Currently the services are offered from two DataCenters (DC1 and DC2) geographically distributed with the aim of guarantee data redundancy. Assuming that communication with the main DC is lost, it has been defined a procedure that allows the restoration of services in the other DC, with the disadvantage that it is a manual procedure that requires administrators intervention.

5.2 Existent Hardware

Each DC has an average of seven Dell PowerEdge™ Racked Servers with different capacities, interconnected via communication devices that provide various services analogously.

There is a Dell PowerEdge 2850/2950 server serving as firewall and main router on each DC. On the one hand it has a WAN 1000Mbps interface, which is connected through UTP Cat-6 wired to 24 PoE ports Switches Netgear FS728TPv1 Gigabit. Physical servers are installed in 20U rack cabinets. These servers range from models Dell PowerEdge 1950, R510 to R710, have Intel Xeon CPUs within 24 and 64 cores, count with 8 to 64GB of RAM, and also have SCSI HDD with capacities between 100GB and 2,5TB.

5.3 Company Network Scheme

The housing services leased by the providers offer a pool of public IPv4 addresses that are handled by the main router on each DC facility. DC1 and DC2 are interconnected by a VPN through WAN, each of them associated to a different private Class B network internally. To the LAN Ethernet ports of the switches are connected the physical servers of the private network with transfer speed rates of 100/1000Mbps. The overall interconnection scheme can be appreciated in Figure 5.1.

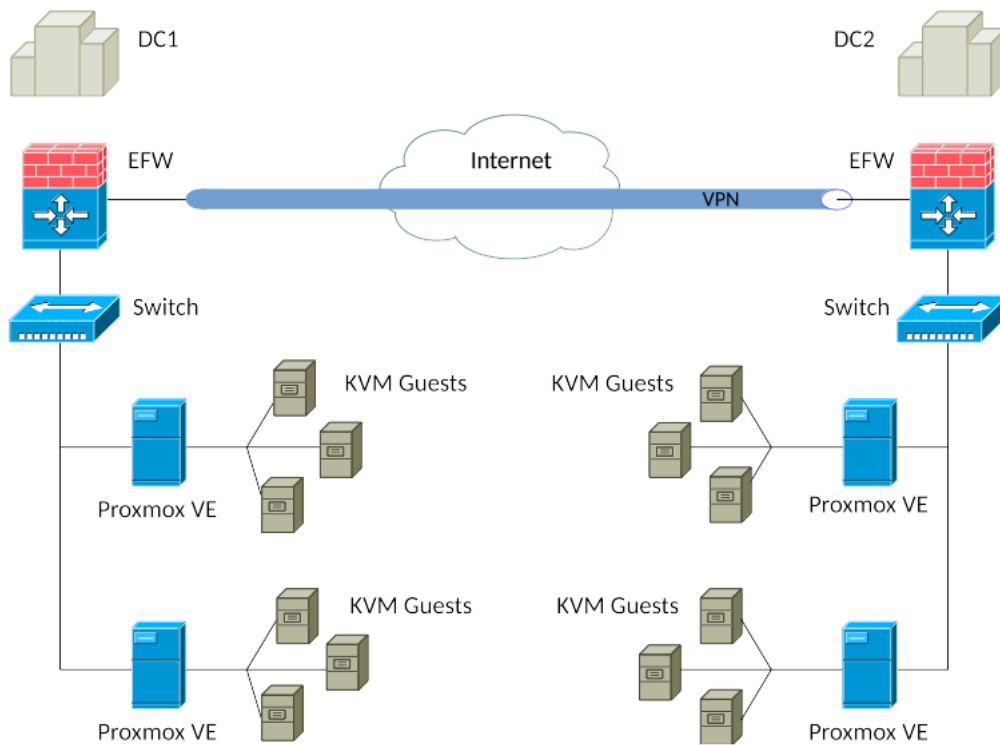


Figure 5.1: Network interconnection scheme

5.4 Software Supporting the Infrastructure

In Table 5.1 there are some software solutions currently used by Netnovation that are related with the required architecture to provide IT cloud-oriented services, with a brief description and legal licensing information for each one of them.

UTM Endian Firewall	
Company:	Endian S.r.l.
Industry:	Unified Threat Management
License:	GNU GPL
Website:	endian.com
Description:	A linux security distribution with full featured Unified Threat Management functionality. Include a stateful packet inspection firewall, application-level proxies for various protocols, antivirus support, virus and spam-filtering for email traffic, content filtering of Web traffic, also an Open-VPN solution. Distribution based on Red Hat.
Supported Platforms:	GNU/Linux
Commercial support:	annual subscription
Proxmox VE	
Company:	Proxmox Server Solutions GmbH
Industry:	Server Virtualization
License:	GNU Affero and GPLv3
Website:	pve.proxmox.com
Description:	Virtualization management solution for servers, based on KVM and containers Server Virtualization Platform, provides KVM and OpenVZ hypervisors. Distribution based on Debian.
Supported Platforms:	GNU/Linux
Commercial support:	annual subscription
FreeNAS	
Company:	iXsystems, Inc.
Industry:	Computer Storage
License:	BSD 2-Clause
Website:	freenas.org
Description:	Network-attached storage server, supporting many network and storage protocols such as Samba and NFS. Also supports ZFS. Distribution based on FreeBSD.
Supported Platforms:	BSD Unix
Commercial support:	custom quotes and support tickets
Zabbix	
Company:	Zabbix SIA
Industry:	IT Monitoring
License:	GNU GPLv2
Website:	zabbix.com
Description:	Solution for monitoring of networks, applications and databases.
Supported Platforms:	GNU/Linux
Commercial support:	custom quotes and support tickets

Table 5.1: Software Supporting the Infrastructure

Chapter 6

Technological background

The following software tools represent the key elements on which it has been possible to implement a comprehensive high availability solution, some of them mentioned in Section 3.2.

- Red Hat Enterprise Linux Server

GNU/Linux enterprise-oriented distribution providing a very stable base system, vast documentation and proper support from manufacturer, released as FLOSS mainly under the terms of the GNU Lesser General Public License 2.1, except for some optional components. In order to be specific in this exercise, the Linux kernel 2.6.32-431.el6.x86_64 that is included by the RHEL version 6.5 was used.

- Zimbra Collaboration System (ZCS)

Server and client collaboration software, supporting e-mail, contacts, calendar, documents, push synchronization, and many other enterprise features related to groupware. The software is FLOSS released under the terms of the Common Public Attribution License version 1 and the GNU General Public License version 2 (GPLv2). The exact version implemented was ZCS FOSS 8.0.7_GA_6021.RHEL6_64.

- Distributed Replicated Block Device (DRBD)

A distributed replicated storage system for Linux, implemented as several userspace management applications and shell scripts, used to provide data redundancy. DRBD software is FLOSS released under the terms of the GNU GPLv2.

- Corosync

It is released as FLOSS under the 3-clause BSD License. This software provides features based on C programming language implementing high availability within applications, through virtual synchrony for replicated state machines, simple availability handling responsible for applications restart when fail, it keeps configuration and statistics in a memory database providing the ability to set, retrieve, and receive change notifications of information, and a quorum system that notifies applications when it is achieved or lost.

- Pacemaker

A FLOSS high availability resource manager software released under GNU GPLv2. This software was part of the Linux-HA project until 2007, then was split out to be its own project. It implements APIs for resources control, including the Open Cluster Framework (OCF). It is used on computer clusters since 2004.

- Cluster Resource Manager Shell (CRMsh¹) and Pacemaker Configuration System (PCS²)

Initially, the CRMsh was distributed as part of the Pacemaker project, but it was split into its own separate project in 2011. Also as CRMsh, PCS is a command-line interface to the Pacemaker cluster resource management stack.

- Cluster Configuration System (CCS)

Manages the cluster configuration and provides information to other cluster components. Runs in each cluster node and makes sure that the cluster configuration file in each cluster node is up to date. In Figure 6.1 (took from https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/4/html/Cluster_Suite_Overview/images/) is represented a CCS overview.

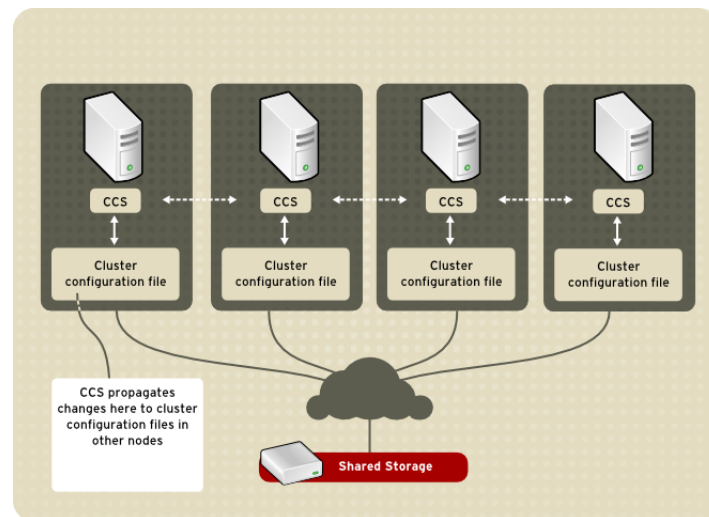


Figure 6.1: CCS overview

- Cluster Manager (CMAN³)

A set of kernel patches and a userspace program, formed by a Connection Manager (cnxman) and a Service Manager (sm). The first one handles membership, messaging, quorum, event notification and transitions, and the second one is responsible for instances of external systems. It combines some functionalities provided by CRMsh, PCS and CCS.

¹<http://crmsh.github.io>

²<https://github.com/feist/pcs>

³<https://www.sourceware.org/cluster/cman/>

Chapter 7

Implementation

This section is intended to provide technical documentation in the process of implementing high availability in a FLOSS Zimbra Collaboration System (ZCS). The scope of this implementation is framed by the following software components and versions:

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

The defined cluster consists of two nodes which will be referenced as Astapor and Braavos in the domain got.com (as in the novel Game of Thrones). These nodes are virtual machines hosted on two Proxmox Virtual Environment servers based on KVM virtualization, which are installed on separate physical machines in the same LAN to avoid single point of failure. The proposed scheme is conceptually similar to the observed in Figure 7.1.

7.1 Operating system considerations

The configuration must be similar in both nodes. In Table 7.1 is shown the configuration selected for the current implementation. It should be take into consideration that it is not necessary to format partitions for devices vdb1 or vdc1 during OS install.

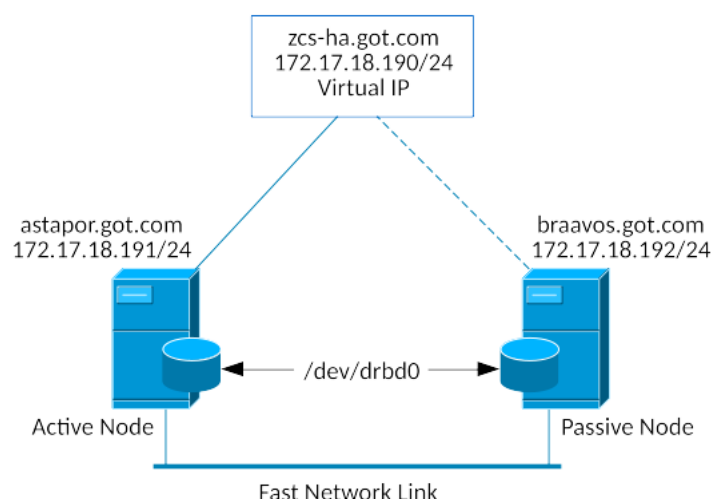


Figure 7.1: Two nodes HA cluster

RHEL 6.5 x86_64				
Disk Partitions:	/	10 Gb		
	/boot	100 Mb		
	/opt/zimbra	8 Gb	(/dev/vdb1)	
	drbd meta-data	150 Mb	(/dev/vdc1)	
CPU:	1			
RAM:	2 Gb			

Table 7.1: Operating system configuration

7.1.1 FQDN hostnames and IP addresses

Table 7.2 shows the current configuration for the virtual IP address shared by the two nodes, and for the primary IP address on each node.

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

Table 7.2: FQDN hostnames and IP addresses

On both nodes, /etc/hosts file should contain at least the entries described in Table 7.3 :

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

Table 7.3: /etc/hosts file

A useful command to handle hostname changes in RHEL:
service hostname restart

7.1.2 Network

- Internet Protocol version 4 (IPv4)

Set the proper network parameters in `/etc/sysconfig/network-scripts/ifcfg-eth0` file on each server, as described in Table 7.4.

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

Table 7.4: `/etc/sysconfig/network-scripts/ifcfg-eth0` file

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

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

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

- NTP

Required RPM packages to sincronize cluster nodes through network time protocol: ntp, ntpdate.

Set the proper NTP parameters in `/etc/ntp.conf` file on each server, so both nodes share the same date and time, as shown in Table 7.5.

driftfile	DEVICE=eth0/var/lib/ntp/drift
restrict	default kod nomodify notrap nopeer noquery
restrict	127.0.0.1
server	172.17.18.1
includefile	/etc/ntp/crypto/pw
keys	/etc/ntp/keys

Table 7.5: /etc/ntp.conf file

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

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

- BIND

Required RPM packages for domain name resolution: bind, bind-utils.

A primary DNS server configured on each server is crucial, or alternatively a remote centralized DNS server on the LAN with the whole configuration. Here is considered the first option. Table 7.6 shows the content of /etc/named.conf file.

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

Table 7.6: /etc/named.conf file

Astapor node holds /var/named/got.com.db file, with the content described in Table 7.7. Dot characters at the end of hostnames are not a typo, they should be included so that the configuration is correct, and must be absent in the case of IP addresses.

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

Table 7.7: /etc/named.conf file

A similar got.com.db file must be set on braavos node replacing the corresponding hostname and IP address. Leaving zcs-ha entries without changes in both nodes.

Some useful commands to handle and request BIND service on RHEL are:

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

7.1.3 ZCS dependencies

As requirement for ZCS, the following RPM packages must be installed in the OS:

- nc
- sudo
- libidn
- gmp
- libaio

Some other suggested RPM packages are:

- perl-5.10.1
- sysstat
- sqlite

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

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

7.2 DRBD

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

7.2.1 Initial configuration

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

- Ensure to adapt hostname to ‘astapor’ on the primary node and ‘braavos’ on the secondary node.
- Install RPM packages:
drbd-kmdl-2.6.32-431.el6-8.4.3-33.el6.x86_64
drbd-8.4.3-33.el6.x86_64
- Leave /etc/drbd.conf and /etc/drbd.d/global_common.conf files by default.
- Add /etc/drbd.d/optzimbra.res file with the content described in Table 7.8:

zcs-ha	IN	1H	A	172.17.18.190
astapor	IN	1H	A	172.17.18.191

Table 7.8: /etc/drbd.d/optzimbra.res file

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

- Initialize data and metadata disks:
dd if=/dev/zero of=/dev/vdb1 bs=1K count=100
dd if=/dev/zero of=/dev/vdc1 bs=1K count=100
- Start DRBD module:
modprobe drbd
- Create resource:
drbdadm create-md optzimbra
- Execute first DRBD synchronisation on astapor:
drbdadm up optzimbra
drbdadm primary --force optzimbra
drbdadm --discard-my-data connect optzimbra
- It is possible to check synchronisation status with:
watch cat /proc/drbd
- Final output will show:
ds:UpToDate/UpToDate
- Verify current roles:
drbdadm role optzimbra
It will show 'Primary/Secondary' on astapor
and 'Secondary/Primary' on braavos node.
- Now make the filesystem on astapor:
mkfs.ext4 /dev/drbd0
- Then demote node to secondary, by executing only on astapor:
drbdadm secondary optzimbra
- Promote node to primary, by executing only on braavos:
drbdadm primary optzimbra
- Make the filesystem on braavos:
mkfs.ext4 /dev/drbd0

Now it is necessary to revert the roles back, making braavos the secondary node and astapor the primary one, with the procedure described in section 7.2.1.

7.2.2 DRBD Split Brain Recovery

Assuming that the primary node is still consistent, and the secondary node has an inconsistent state, it would be necessary to recover data loss. The following actions will allow to recover the data corrupted in secondary node.

- In both nodes:
drbdadm disconnect optzimbra
- In the secondary node:
drbdadm secondary optzimbra
drbdadm connect - --discard-my-data optzimbra

- In the primary node:
drbdadm connect optzimbra
- Finally it is possible to check the sync status, running the command “cat /proc/drbd”, which is going to show a message similar to this:
cs:Connected ro:Primary/Secondary ds:UpToDate/UpToDate C r- - - -

7.3 ZCS

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

7.3.1 ZCS full install on primary node

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

- Create directory for ZCS:
mkdir /opt/zimbra
- Mount DRBD device on ZCS mount point:
mount /dev/drbd0 /opt/zimbra
- Check mounted device:
df | grep zimbra
mount | grep zimbra
- Set manual virtual link configuration temporally:
ifconfig eth0:1 inet 172.17.18.190 netmask 255.255.255.0
- Set split DNS hostname temporally:
hostname zcs-ha.got.com
It is also recommendable to change /etc/sysconfig/network file.
- Unpack ZCS installer and proceed with full installation:
./install.sh
- Leave all packages to install by default, and follow the process.
- When prompted for domain name change, select “Yes” and then provide: got.com
- On “Main Menu” section, set admin user password by browsing through option 3 and then 4:
“Password for admin@zcs-ha.got.com (min 6 characters)”
- Apply configuration and advance until ZCS setup process is completed:
“Configuration complete - press return to exit”
- Check ZCS status:
service zimbra status

- Stop ZCS:
service zimbra stop
- Umount DRBD device:
umount /opt/zimbra
- Set original DNS hostname:
hostname astapor.got.com
Revert change in /etc/sysconfig/network file if needed.
- Delete temporal virtual link configuration:
ifconfig eth0:1 down
- Demote astapor to secondary DRBD, and continue with section 7.3.2:
drbdadm secondary optzimbra

7.3.2 ZCS dummy install on secondary node

The following actions must be performed sequentially on braavos.

- Promote braavos to primary DRBD:
drbdadm primary optzimbra
- Create directory for ZCS:
mkdir /opt/zimbra
- Mount DRBD device on ZCS mount point:
mount /dev/drbd0 /opt/zimbra
- Check mounted device:
df | grep zimbra
mount | grep zimbra
- Unpack ZCS installer and proceed with a dummy installation:
./install.sh -s
- Stop ZCS:
service zimbra stop
- Umount DRBD device:
umount /opt/zimbra
- Demote braavos back to secondary DRBD:
drbdadm secondary optzimbra
- Promote astapor back to primary DRBD, executing from astapor node:
drbdadm primary optzimbra

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

```
watch cat /proc/drbd
```

7.4 OCF

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

- Create file `/usr/lib/ocf/resource.d/btactic/zimbra`:
- Also create the following symbolic link:
`ln -s /usr/lib/ocf/resource.d/btactic/zimbra /usr/lib/ocf/resource.d/heartbeat/`

In section 7.5 this file will be referenced.

7.5 Pacemaker

Resource manager, starts and stops services orderly.

- Install the required RPM packages:
`pacemaker-cluster-libs-1.1.10-14.el6.x86_64`
`pacemaker-libs-1.1.10-14.el6.x86_64`
`pacemaker-cli-1.1.10-14.el6.x86_64`
`pacemaker-1.1.10-14.el6.x86_64`
`cman-3.0.12.1-59.el6.x86_64`
`crmsh-1.2.5-0.el6.x86_64`
`ccs-0.16.2-69.el6.x86_64`
`resource-agents-3.9.2-40.el6_5.7.x86_64`

Usually it is difficult to obtain the required RPM's for RHEL, so an alternative is to add CentOS repository by editing `/etc/yum.repo.d/centos.repo` file with the content described in Table 7.9.

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

Table 7.9: `/etc/yum.repo.d/centos.repo` file

- Then update and install the packages:
`yum install --enablerepo=centos-6-base pacemaker pcs.noarch cman \`
`ccs resource-agents crmsh`

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

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

```
crm(live)# quit
```

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

- Create the cluster:
`ccs --file /etc/cluster/cluster.conf --createcluster zcsCluster`
- Add the nodes:
`ccs --file /etc/cluster/cluster.conf --addnode astapor.got.com`
`ccs --file /etc/cluster/cluster.conf --addnode astapor.got.com`
- Set fencing to defer to Pacemaker:
`ccs --file /etc/cluster/cluster.conf --addfencedev pcmk agent=fence_pcmk`
`ccs --file /etc/cluster/cluster.conf --addmethod pcmk-redirect astapor.got.com`
`ccs --file /etc/cluster/cluster.conf --addmethod pcmk-redirect braavos.got.com`
`ccs --file /etc/cluster/cluster.conf --addfenceinst pcmk astapor.got.com \`
`pcmk-redirect port=astapor.got.com`
`ccs --file /etc/cluster/cluster.conf --addfenceinst pcmk braavos.got.com \`
`pcmk-redirect port=braavos.got.com`
- Disable CMAN quorum:
This will let the cluster function if only one node is up, and it is necessary to be performed in both nodes.
`echo "CMAN_QUORUM_TIMEOUT=0" >> /etc/sysconfig/cman`
- Start Pacemaker Cluster:
`pcs cluster start --all`
Also equivalent to execute on each node,
"service pacemaker start" or "pcs cluster start"
- Copy cluster file to secondary node:
`scp -p /etc/cluster/cluster.conf braavos:/etc/cluster/`
- Check Pacemaker cluster status:
`pcs status`
`crm_mon -l`
- Show current cluster config:
`pcs config`
`pcs property`
`crm configure show`
- Check configuration validity:
`crm_verify -L -V`
- Disable STONITH (a type of fencing):
`pcs property set stonith-enabled=false`
- Ignore Quorum Policy:
`pcs property set no-quorum-policy=ignore`

- Set reconnect attempt:
pcs property set migration-threshold=1 --force
- Set stickiness:
pcs property set resource-stickiness=100 --force

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

- Add floating IP address resource (Virtual IP - VIP):
pcs resource create VIP1 IPaddr2 ip=172.17.18.190 broadcast=172.17.18.255 \
nic=eth0 cidr_netmask=24 iflabel=VIP1 op monitor interval=30s timeout=30s
- Define DRBD cluster resource:
configure primitive drbd ocf:linbit:drbd params \
drbd_resource=optzimbra \
op monitor role=Master interval=60s \
op monitor role=Slave interval=50s \
op start role=Master interval=60s timeout=240s \
op start role=Slave interval=0s timeout=240s \
op stop role=Master interval=60s timeout=100s \
op stop role=Slave interval=0s timeout=100s
- Define DRBD Zimbra data clone:
configure ms drbd_ms drbd \
meta master-max=1 master-node-max=1 \
clone-max=2 clone-node-max=1 notify=true
- Define Zimbra service resource:
configure primitive zcs_service ocf:btactic:zimbra \
op monitor interval=2min timeout="40s" \
op start interval="0" timeout="360s" \
op stop interval="0" timeout="360s"
- Define Zimbra cluster filesystem resource:
configure primitive zcs_fs ocf:heartbeat:Filesystem params \
device="/dev/drbd0" directory="/opt/zimbra" fstype=ext4 \
op start interval=0 timeout=60s \
op stop interval="0" timeout="60"
- Group all resources in the same host:
group zcsgroup zcs_fs zcs_service \
configure colocation VIP1-with-drbd_ms-Master inf: drbd_ms:Master VIP1
configure colocation drbd_ms-Master-with-zcs_fs inf: zcs_fs drbd_ms:Master
configure colocation zcs_fs-with-zcs_service inf: zcs_service zcs_fs
- Order resources:
configure order drbd_ms-promote-on-VIP1 inf: VIP1:start drbd_ms:promote
configure order zcs_fs-on-dbrb_ms-promote inf: dbrb_ms:promote zcs_fs:start
configure order zcs_service-on-zcs_fs inf: zcs_fs:start zcs_service:start

- Commit configuration changes and quit:
commit
quit

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

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

7.6 Control and check services

- Check Pacemaker cluster status:
crm_mon -l
pcs status
- Check resources status:
crm resource status RESOURCE
- Check configuration validity:
crm_verify -L -V
- Edit values already configured:
crm configure edit
After save changes through the preferred text editor, exit and execute:
cibadmin --replace
- Delete existent resource:
pcs resource delete RESOURCE
- Clean resource history errors (check configuration health):
crm_resource -P
- List available classes and resources:
crm ra classes
crm ra list ocf btactic
crm ra list lsb
- Delete cluster configuration (WARNING):
pcs cluster destroy

7.7 Testing failover

- On primary node:
crm node standby
Or stop pacemaker:
service pacemaker stop

- Now “crm_mon” or “pcs status” will show:
Node astapor.got.com: standby
Online: [braavos.got.com]
- It is going to take a while before secondary node takes control. So it is possible to check logs and “crm_mon” status during the process.
crm_mon
tail -F /var/log/zimbra.log
tail -F /var/log/messages
- Also it is possible to check with “crm_standby” command. A value of true|on indicates that the node is not able to host any resources and a value of false|off indicates it does.
crm_standby --get-value
- At any moment it will be displayed a message like the depicted in Table 7.10.

Master/Slave Set:	drbd_ms [drbd]
Masters:	[braavos.got.com]
Slaves:	[astapor.got.com]
Resource Group:	zcsgroup
zcs_fs (ocf::heartbeat:Filesystem):	Started braavos.got.com
zcs_service (ocf::btactic:zimbra):	Started braavos.got.com
VIP1 (ocf::heartbeat:IPAddr2):	Started braavos.got.com

Table 7.10: DRBD failover test

- Now the secondary node has control of the cluster resources, while the primary node is in standby or unreachable state. If primary node is back online, secondary node will keep the control of resources, until an explicit node move is done.
- Set back online the primary node:
crm node online
Or start over pacemaker service:
service pacemaker start
- To give the control back to primary node, execute on secondary node:
crm node standby
Then resources will be transferred back to primary node.
- Finally “crm_mon” or “pcs status” on each node will display a similar to the one showed in Table 7.11

Online:	[astapor.got.com braavos.got.com]
Master/Slave Set:	drbd_ms [drbd]
Masters:	[astapor.got.com]
Slaves:	[braavos.got.com]
Resource Group:	zcsgroup
zcs_fs (ocf::heartbeat:Filesystem):	Started astapor.got.com
zcs_service (ocf::btactic:zimbra):	Started astapor.got.com
VIP1 (ocf::heartbeat:IPAddr2):	Started astapor.got.com

Table 7.11: DRBD synced status

Chapter 8

Results and discussion

This chapter presents the evidence and/or results of primary research which you have undertaken Presented in the form of:

- hypothesis testing to analysis suggested by guidelines for the adoption of FLOSS within SMEs (Daf-fara), search method by collecting and read as much information related to the project is available and select the appropriate solution from a matching set that fulfill the requirements
- Detailed quantitative model OpenBRR, quick and flexible assessment over software tools considered
- Lazy User Model, frame the process by which are chosen the technological tools that will make up the solution that meets user requirements. This model focuses on user needs and the demanded effort when selecting a solution to a problem from a set of possible solutions

Chapter 9

Conclusions and future work

Text..

Bibliography

- [1] Carlo Daffara. The sme guide to open source software. FLOSSMETRICS EU, 2009.
- [2] Carlo Daffara. The economic value of open source software. Conecta Research. Italy, 2012.
- [3] International Organization for Standardization Technical Committee 223. Iso/pas 22399:2007. International Organization for Standardization, 2007.
- [4] Collan M. and Tetard F. Lazy user theory of solution selection. Proceedings of the CELDA 2007 Conference. Portugal, 2007.
- [5] Collan M. and Tetard F. Lazy user theory: A dynamic model to understand user selection of products and services. 42nd Hawaii International Conference on System Sciences, 2009.
- [6] Peter Mell and Timothy Grance. The nist definition of cloud computing. National Institute of Standards and Technology, 2011.
- [7] Tanja R. and Schraitle T. High availability guide. SUSE LLC and contributors, 2014.
- [8] Pal M. Wasserman A. and Chan C. The business readiness rating model: an evaluation framework for open source. Proceedings of the EFOSS: OpenBRR Workshop. Italy, 2006.

Appendix A

Title of appendix 1