

Ekaterina Kaloshina

Proxmox HA virtualization cluster

Bachelor's thesis
Information Technologies

2017



South-Eastern Finland
University of Applied Sciences

Author (authors)	Degree	Time
Ekaterina Kaloshina	Bachelor of Engineering	September 2017
Title		45 pages
Proxmox HA virtualization cluster		
Commissioned by		
Stais		
Supervisor		
Matti Juutilainen		
Abstract		
<p>The objective of the thesis was to create a test virtualization cluster with minimum costs and the best possible quality. The main goal was to create a working system with an opportunity to start a virtual machine on it. The questions of stability, accessibility and downtime were the most important. In case of a site being down, another site is searched by people, in case of a database down, money loss is possible for the employer.</p> <p>For achieving the result, information on clusters, virtualization and storages was studied. In parallel with gathering information for the theory part, the practical part was completed with an installation and configuration of a Proxmox hypervisor and two storage software: Ceph and FreeNAS. Small basic tests were also made in the practical part of this thesis. Possible further development was described and included moving to the newer devices and software versions.</p> <p>As a result of the work, a working cluster was configured, but wasn't tested in the main company's network. During the study, it was also found that every software developed faster than it could be described properly. It wasn't possible to get all the information on the newest software versions that became older during the thesis writing process.</p>		
Keywords		
Virtualization software, IT, storage, cluster, Proxmox, Ceph, FreeNAS		

List of abbreviations

CLI – Command Line Interface

CPU – Central Processing Unit

FLOPS – Floating-point Operations Per Second

GPU – Graphics Processing Unit

GUI – Graphical User Interface

HA – High-Availability

HDD – Hard Disk Drive

HPC – High Performance Computing

IP – Internet Protocol

iSCSI – Internet Small Computer System Interface

KVM – Kernel-based Virtual Machine

LXC – Linux Containers

NAS – Network Attached Storage

NFS – Network File System

NIC – Network Interface Controller

NLB – Network Load Balancing

ODD – Optical Disk Drive

OS – Operating System

OSD – Object Storage Device

RAID – Redundant Array of Independent Disks

RAM – Random Access Memory

RBD – Rados Block Device

SMB – Server Message Block

TCP – Transmission Control Protocol

TCP – Transmission Control Protocol

UPS – Uninterruptable Power Supply

VM – Virtual Machine

ZFS – Zettabyte File System

CONTENTS

LIST OF ABBREVIATIONS	3
1 INTRODUCTION	6
2 CLUSTERING	6
2.1 Cluster history.....	7
2.2 Cluster type	8
2.2.1 HA.....	8
2.2.2 NLB.....	9
2.2.3 HPC	9
3 VIRTUALIZATION	10
3.1 Hypervisors.....	10
3.2 Proxmox	11
4 DATA STORAGE.....	12
4.1 RAID	12
4.2 Software	14
4.2.1 Ceph.....	14
4.2.2 FreeNAS.....	15
5 COMPANY	15
6 INSTALLATION AND CONFIGURATION.....	16
6.1 Hardware	16
6.2 Software	18
6.2.1 Proxmox.....	18
6.2.2 Ceph	24
6.2.3 Proxmox VM: create and test.....	28
6.2.4 FreeNAS.....	32

7	CONCLUSIONS	42
	REFERENCES	44

1 INTRODUCTION

Nowadays the idea of virtualized servers is not strange or new. There are enough positive sides and several big problems associated with that kind of servers that can prevent server owners from choosing virtualization instead of real servers. As soon as the amount of possible virtualization software has grown strongly from the past, the available choice is much greater great today and more available.

The idea for this thesis was given by the company “Stais” in Russia. Their business is not associated with IT, but it was decided to improve their server’s structure due to plans of creating an E-shop. I was asked to find a solution for the possible availability problem. It is known that sometimes servers may shut down, because of hardware issues, and then the service may not be available for hours or even more.

To make the upgrade of the servers and their software possible, I plan to spend two months on the practical part and around two months on studying books, manuals and receiving advice from people working with virtualization systems. The aim of my study is to make a running platform to prove the operability of a cluster on the test servers.

My thesis contains six chapters. Chapter 2 is used to explain the origins of clustering, possible uses of clusters and the storage required. The third chapter takes a closer look into the virtualization software and describes the hypervisor that is planned to be used in a practical part. Chapter 4 includes the description of the storage and storage software. In Chapter 5, I explain why cluster testing is interesting for the Stais company. Chapter 6 includes the hardware description and the installation with the configuration part. The last chapter includes the conclusions and the way of possible development of the practical part.

2 CLUSTERING

What is a cluster? “A cluster is a set of multiple, independent computer systems that project a single system image for access and management to most entities

such a human users and client applications, that are outside the cluster.” (Rajagopal 2000.) The idea of clustering is to use these connected computers for providing the maximum uptime, increase of processing speed and more balanced use of resources. There are three main computer clusters types, which I will describe in a chapter 2.2.

2.1 Cluster history

“Virtually every press release from Digital Equipment Corporation mentioning clusters says ‘DEC, who invented clusters...’. IBM did not invent them either. Customers invented clusters, as soon as they could not fit all their work on one computer, or needed a backup. The date of the first is unknown, but it would be surprising if it was not in the 1960s, or even late 1950s.” (Pfister 1997.) These words can be found in almost any book describing computer clustering history. There are no original dates for cluster appearance. In search of the origins of clusters I found that clusters are closely connected to the internet. The ARPANET project, which was a first network with TCP/IP standard, was developed in 1969. It is a so-called prototype of the Internet. Without that project no clusters as we know them today would have existed. In the early 1970s with the corporation of the TCP/IP developers and Xerox PARC laboratory, the standards of network-based communications were created and after that in 1971, the Hydra operating system was built for a cluster at Carnegie Mellon University.

The first commercial clustering project called ARCnet was announced in 1977 and then stopped because of being non-profitable. With Parallel Virtual Machine software developed in 1989, the possibility of instant creation of a virtual supercomputer was achieved. Version 1.0 of PVM was used only inside of a laboratory and was not planned to be spread for other users. Version 2.0 was developed in 1991.

The Beowulf project was started in 1994. This project was aimed to make a cluster with multiple number of usual cheap personal computers. “The first Beowulf was built with DX4 processors and 10Mbit/s Ethernet. The processors were too fast for a single Ethernet and Ethernet switches were still expensive. To balance the

system, Don Becker rewrote his Ethernet drivers for Linux and built a "channel bonded" Ethernet where the network traffic was striped across two or more Ethernets. As 100Mbit/s Ethernet and 100Mbit/s Ethernet switches have become cost effective, the need for channel bonding has diminished (at least for now)." (Merkey, Beowulf project.) The great explanation of Beowulf may be "Beowulf is a multi computer architecture which can be used for parallel computations. It is a system that usually consists of one server node, and one or more client nodes connected together via Ethernet or some other network. It is a system built using commodity hardware components, like any PC capable of running Linux, standard Ethernet adapters, and switches." (Radajewsky & Eadline 1998.)

2.2 Cluster type

As mentioned in the beginning of chapter two, there are three commonly known cluster types. In this chapter HA, NLB and HPC clusters are introduced.

2.2.1 HA

The HA stands for high-availability which can also be named as a failover. The main purpose of that cluster type is providing higher (higher than the one that can be achieved with the separate servers) availability level, easier scalability and simpler management. (Borzenko 2002.) A HA cluster can migrate a working task to the other node in case of an error or power down. This can reduce downtime because of automatical migration, without constant administrator control. Without the HA cluster in case of physical components or system failure working application will be down until the server is fixed. Usually these clusters are used for databases, business applications or other services sensitive for downtime. Most cluster systems require at least two servers for a proper work of HA. The picture below shows a possible physical structure of a cluster. It includes redundant storage and network and possible clients connected.

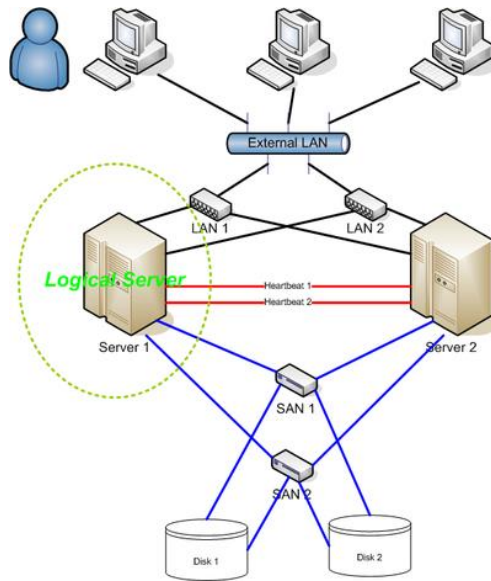


Figure 1. HA cluster structure

2.2.2 NLB

Network load balancing clusters provide better scalability, fault tolerance and high accessibility. For working, these clusters use virtual network interfaces with a new IP address different from the IP addresses of the other nodes. All nodes receive tasks, but only one of servers handles it. In case of heartbeat-message failure or a new node appearance, other nodes start convergence to reallocate load. Even if the main target of that cluster type is performance increase, it also provides increased reliability. The structure of this cluster type is not very different from a HA cluster. Nowadays software for a HA clustering frequently have a balancing function.

2.2.3 HPC

High performance computing clusters are mostly used for faster operations with FLOPS operations. Beowulf, mentioned in the cluster history chapter, is an example of a HPC cluster. The physical structure of that cluster usually requires a low latency network and 1U, 2U, 4U or blade servers. Usual PCs are possible too, but it is recommended to minimize the physical distance between servers. The structure of the HPC cluster includes one master/login node and a multiple number of computing nodes. "The compute nodes do really one thing - compute. That is really about it. Not much else to say about this except that the form factor

(chassis) for the compute node can vary based on the requirements for the compute node.” (Clerk002 2012.) A simple example of that cluster can be a cluster made from five PCs with four cores. A result will have twenty cores and will work as a single system, in which one PC is a master node and each other PC is used as a donor.

3 VIRTUALIZATION

Virtualization is an old technology coming from the 1960s which nowadays is used for better resources allocation and energy, money saving. It can be described as a logical divide of the system resources. Virtualization creates a virtual machine that acts like a real computer.

3.1 Hypervisors

A hypervisor is software, hardware or any other host that creates and manages virtual machines, as defined by Ahmed (2016). The table 1 below introduces four bare-metal hypervisors that are popular nowadays. Hyper-V and ESXi are more adapted for big companies; XenServer and Proxmox are more suitable for small or medium-sized companies.

	Hyper-V 2016	Proxmox	ESXi (VMware vSphere)	XenServer
Hypervisor type	Bare metal	Bare metal	Bare metal	Bare metal
Open source	no	Open-source	no	Open-source
Host memory support	Up to 24Tb per physical server	2Tb	12Tb	5Tb
VM memory support	Up to 12Tb per VM	512Gb	6Tb	1.5Tb

Live Migration	Possible	Possible	Possible	Possible
Price	Standard US\$ 882 for 16 cores for Windows server	Free	Standard US\$ 1,746.23	Free
Support	Microsoft IT academy	Maintenance subscription Standard € 33,17/month Community € 5,83/month per CPU	Maintenance license Basic US\$ 552.90/year	Maintenance subscription US\$ 695.00/year
Failover Minimum	2 server cluster	3 server cluster	2 server cluster	2 server cluster

Table 1. Hypervisors

The main supported systems are the same for these hypervisors and are not included to this table. Proxmox has lack of official specifications on their website. It is possible that the RAM limit is greater nowadays, but as far, there is no confirmation, and only known data is included.

3.2 Proxmox

This section takes a closer look at Proxmox, which is also used in the practical part of this thesis. Proxmox is a cluster-based bare metal hypervisor. It allows building a cluster for the needs of small and even enterprise sized companies without requiring any mandatory payment. To use Proxmox as a failover cluster, it must be installed on three or more nodes. A 3.4 Proxmox version was able to use only two nodes for an HA cluster, but nowadays it is not possible, even if it is included in books, because the www page is deleted. "By any means, a two node cluster is not recommended for a production cluster. However, it is still possible to create using instructions by visiting

https://pve.proxmox.com/wiki/TwoNode_High_Availability_Cluster.” (Ahmed 2015.)

It is possible to download Proxmox on its official site and fully configure it using freely accessed documents. Useful Proxmox features are KVM, LXC, live migration, firewall, almost fully included Ceph(the storage software, described in chapter 4.2.1) support sharing the same GUI with Proxmox, virtual switches, high availability, and support of multiple types of network storages(LVM group, iSCSI target, NFS, Ceph, iSCSI LUN direct connect, GlusterFS), friendly GUI interface through browser, load monitoring, snapshots.

The minimum system requirements are 64bit CPU, 1Gb RAM, HDD and NIC. Proxmox download repositories may vary on paid and free versions that do not affect main functions of a hypervisor, but may be critical if some update is released and the usual update command checks a wrong repository. It is possible to install Proxmox on a running Debian 64bit server.

4 DATA STORAGE

One of the main components of a properly working HA cluster is a data storage that is stable and fault tolerant. For better data safekeeping, RAID is required. The storage, as well as hypervisor must have UPS, redundant network connection and powering; it must be also protected from regular users.

4.1 RAID

Redundant array of independent disks may be used for the read/write speed increment, but the main target of RAID is to save the biggest possible amount of data. There are several popular RAID variations.

RAID-0 uses a sequential recording principle, which means that data is written to one of two drives alternately. That increases work speed, but it may write a single file to different drives and in case of one drive failure the whole data saved on it will be lost.

RAID-1 is a mirroring type. Data is written to both drives at the same time and in case of one drive failure, the second drive will have a full copy of the data. The minus is that in case of two or more drives included in this RAID type, the size of the final drive will be equal to one original drive. Three 500Gb drives will result in 500Gb. Two drives are a minimum for this RAID.

RAID-5 stores the control sum on the same drives where the data is. It has a block-level striping and requires at least three drives. Load is balanced between all drives and only the size of one drive from the array is lost.

“Raid technology has been the fundamental building block for storage systems for many years. It has proven successful for almost every kind of data that has been generated in the last 30 years. However, all eras must come to an end, and this time, it's for RAID. RAID-based storage systems have started to show limitations and are incapable of delivering future storage needs.” (Singh 2015.) ZFS, Zettabyte File System, was originally developed by Sun Microsystems, but it is actively ported to popular platforms. ZFS acts as a volume manager and a file system simultaneously and includes protection against data corruption. ZFS also supports high storage capacities and has its own RAID – RAIDZ.

RAIDZ1 is a RAID configuration that can survive failure of one drive, but in case of a failure of two or more drives data cannot be recovered. The number of drives in this RAID can't be increased after the creation, but it is still possible to replace a dead drive with a new one having a bigger volume, for example. The size of storage should be planned before a RAIDZ creation.

RAIDZ2 uses at least five drives and requires two of them for parity. In case of all drives each have a 500Gb size, the size of the storage will be 1.5Tb. In this RAID any two drives can be lost without data destruction.

The main advantages of any RAIDZ is speed and better data protection. Unless RAID-5, RAIDZ distributes logical blocks among disks. It doesn't aggregate

blocks into fixed-width stripes. The disadvantage is a difficulty with size increment. “With ZFS, you either need to buy all storage upfront or you will lose hard drives to redundancy you don't need, reducing the maximum storage capacity of your NAS.” (Louwrentius 2016)

4.2 Software

This chapter describes two storage solutions that are planned to use in the practical part of the thesis.

4.2.1 Ceph

Ceph is an open source storage solution, which is supported by a Proxmox and included in its GUI. That makes this solution one of the best possible ones to use together with the selected hypervisor.

The philosophy of Ceph is described by Singh (2015) as follows:

- Every component must be scalable.
- There can be no single point of failure.
- The solution must be software-based, open source, and adaptable.
- Ceph software should run on readily available commodity hardware.
- Everything must self-manageable wherever possible.

Ceph installation and configuration can be done directly from Proxmox. All the monitoring can be done from the Proxmox GUI. The recovery process of Ceph is fast, because of all the cluster's drives taking part in it. Ceph can work with the drives of different sizes, it stores data based on a disk size. The disadvantage of Ceph is that more drives should be included inside the servers and a bigger amount of RAM may be used because of the extra size. Not a disadvantage, but important thing is that Ceph requires a separate network between servers for data transmitting.

4.2.2 FreeNAS

FreeNAS is an operating system that can be installed on almost any hardware platform to share data over a network. This is official description that can be easily found on a FreeNAS website. The main and the most interesting features of FreeNAS are ZFS support, file sharing, snapshots and encryption. The most important feature for use of FreeNAS together with Proxmox was possibility to use iSCSI. This OS is free and open-source, that makes it a nice choice for the small company needs.

5 COMPANY

The idea of this thesis was given by a company system administrator who was interested in software testing before the installation of the newly bought servers. The equipment was already bought, but not installed, and as soon as there was no hurry, everything was going to be tested on the old, unconnected to main network, servers. The company is not an IT one, but optimal work is required in any case. The idea is that the work should never stop, if one node fails. Because of that an HA cluster may be one of the possible solutions.

The main problem of server downtime is that the company will lose money because of time waste. Users will not be able to work during downtime and they will still need to receive money for their time wasted. This is not an example of proper work for the employer and may affect partners or clients of the company as well.

The next serious point is the need to save as much data as possible. That is why a separate storage with RAID and a power supply with UPS is required and an example of storage software used is described in Chapter 4.2 of a thesis.

As it was mentioned in the introduction, the main task for the cluster in future will be providing maximum uptime for an E-shop, but as soon as Proxmox supports multiple numbers of VM running, it will not be limited to the shop. Databases and file sharing can also be moved to an HA cluster.

6 INSTALLATION AND CONFIGURATION

This chapter contains information about the equipment used in this case and the software settings. As the next section describes, three servers were used for practical part.

6.1 Hardware

This section includes information on the hardware used in the practical part. These devices are not a minimum requirement. Hardware may vary almost freely.

The all three server used for test are old and not powerful, but they are above the minimum Proxmox requirements. The picture below shows the actual testing cluster. There was no target to make physical reliability for this test.



Figure 2. Devices used for the practical part

The devices used are old servers taken out from a server room, regular PC and two switches. Table 2 describes the internal components of servers. These servers

meet the minimum needed for a Proxmox cluster, they are even above it, but for fast and modern structure, they are too old. There are no original documents left, but the BIOS version date of the oldest server is 2007.

	Server 1	Server 2	Server 3	FreeNAS test PC
CPU	Intel QuadCore Xeon E5430	Intel DualCore Xeon 5050	Intel Xeon x5660	Intel i7 4960x
RAM	16Gb	4Gb	4Gb	8Gb
GPU	ATI Radeon HD 3650 (RV635)	ATI E1000 (RN50)	ATI 3D-Rage XL	
HDD	120Gb 500Gb	140Gb 140Gb	500Gb 500Gb	60Gb 60Gb 60Gb
NIC	Intel 82575EB Intel 82575EB	Intel PRO/1000 EB Intel PRO/1000 EB	Intel PRO/1000 MT Intel PRO/1000 MT	Intel 82579V Realtek 8111GR
CPU Base Frequency	2.66 GHz	3 GHz	2.8GHz	3.6GHz

Table 2. Servers and test PC for storage components

There are two switches TP-Link TL-SG1005D for the networks between three servers. One network is needed to connect servers and the second one is used specially for Ceph, so it is a separate storage network. The 500Gb HDDs were installed in all servers to make it possible to test a Ceph, but a second server wasn't able to support the available 500Gb drives, and therefore a drive of 140Gb was used.

The IP table for the practical part is located below. These IP addresses are the ones, which were used after any editions made during install process. The subnet mask /24 is a 255.255.255.0 subnet mask.

DEVICE	IP AND SUBNET
PX1 NODE	192.168.128.200/24
PX2 NODE	192.168.128.201/24
PX3 NODE	192.168.128.202/24
CEPH PX1	10.10.10.3/24
CEPH PX2	10.10.10.2/24
CEPH PX3	10.10.10.1/24
FREENAS	192.168.128.203/24

Table 3. The IP addresses.

6.2 Software

This part includes installation of Proxmox, Ceph and FreeNAS and configuration for making them work together.

6.2.1 Proxmox

In this section Proxmox installation and configuration is described. It also includes a description of the Ceph and FreeNAS configuration needed to use this storage software together with a Proxmox cluster. There was no specific network, but at some point, I needed to change network configurations on Proxmox machines because of giving them internet access. For Ceph a second ethernet adapter is required at all servers.

The installation process can begin with RAID configuration and a software image prepared. The official link for Proxmox download can be found at <https://www.proxmox.com/en/downloads>

A possible problem for the image may be that some users (who are not familiar with .iso files) miss the idea that some specific burning software may be needed if the image is going to be saved to cd\ dvd discs and used as a booting device. Examples of burning software can be Ashampoo Burning Studio or EasyBoot.

Also, if the image must be written on a flash drive, UltraISO can be used. For this project, Zalman ZM-VE300 was used as virtual ODD (Optical Disc Drive).

The installation process begins on the screen with a list of possible actions in which only “install” is needed now. Next keyboard language, time zone, name and network settings are chosen. At this point all three nodes used for the project had ip addresses from 192.168.1.0/24 network. The nodes were named as px1.pve px2.pve and px3.pve. After installation, which took around thirty minutes, the nodes are rebooted, and it is possible to connect to their web GUI using web browser and an IP address plus port :8006. (example: 192.168.128.200:8006)

Right after booting one node showed error because of a USB attached drive “USB error - usb 1-7: device descriptor read/all, error -71”. Possible solution for this error can be

```
echo -1 >/sys/module/usbcore/parameters/autosuspend
```

written in terminal.

A bit later a problem with the IP addresses occurred, and therefore the settings were updated. A possible way to edit IP settings is described next.

nano /etc/hosts is used to edit hosts file at all nodes. Example of hosts file, copied from px3 server is included below.

```
127.0.0.1 localhost.localdomain localhost
192.168.128.200 px1.pve px1
192.168.128.201 px2.pve px2
192.168.128.202 px3.pve px3 pvelocalhost
```

After file editing the cman and pveproxy processes may be restarted with next commands:

```
/etc/init.d/cman restart
pveproxy restart
```

It may be possible, that DNS is also needed to be changed, if so, resolv.conf must be edited.

/etc/resolv.conf

From now on the IP addresses of nodes are the follows:

192.168.128.200 px1.pve

192.168.128.201 px2.pve

192.168.128.202 px3.pve

After that, the existing nodes must be combined into a cluster. px3 is used as the master node in this thesis. Command below is issued first, to create a cluster. A TClust word stays for a cluster name and can be any.

px3# pvecm create TClust

After that the add command should be issued on two other nodes, to add their connection to a master node:

px2# pvecm add 192.168.128.202

px1# pvecm add 192.168.128.202

After issuing the command it is possible to check if the nodes are connecting to the right master. For a complete confidence password of the master node should be entered at this step and then, after approximately one minute, a node is added. No NTP server was available in testing environment, and time was set on each node with the following command:

root@px1:~# date --set 14:42

Fri Apr 7 14:42:00 MSK 2017

In the production, NTP server should be configured, and time can be received automatically. Picture 2 shows the statistics of a working cluster. The number of working nodes, as well as offline ones, can be seen. Virtual machines, LXC containers and the statistics like a CPU, memory and storage appear at the same screen.

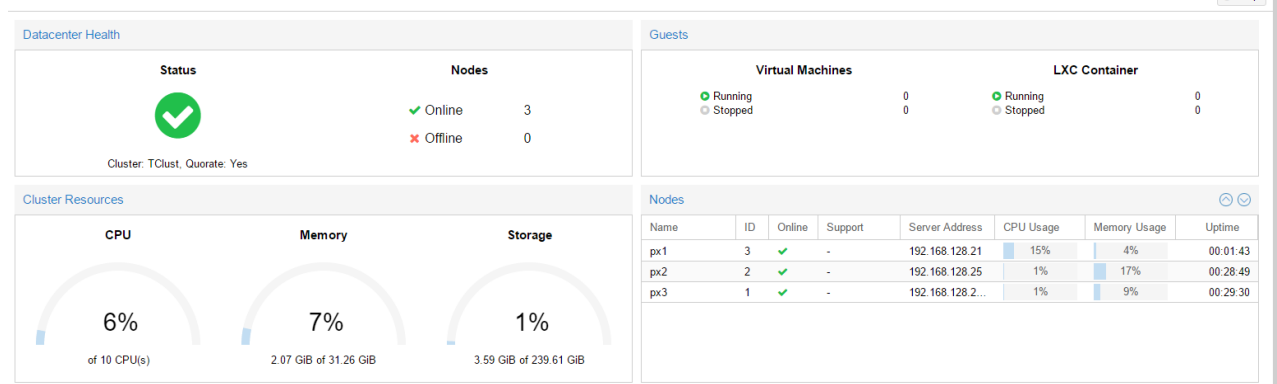


Figure 3. Cluster status screen

Now Proxmox can be updated to support the Ceph version needed. Jewel Ceph is not included in the default Proxmox 4.4. For proper update, all the nodes should be checked for the right source lists. After that, all the nodes should be updated and rebooted.

The next two commands are used to check a source list, output is included:

```
root@px1:~# nano /etc/apt/sources.list.d/pve-enterprise.list
#deb https://enterprise.proxmox.com/debian jessie pve-enterprise
deb http://download.proxmox.com/debian jessie pve-no-subscription
```

```
root@px1:~# nano /etc/apt/sources.list
deb http://ftp.ru.debian.org/debian jessie main contrib
# security updates
deb http://security.debian.org jessie/updates main contrib
deb http://download.proxmox.com/debian jessie pve-no-subscription
```

These commands are used to update repository and packages and get system update.

```
root@px1:~# aptitude update
root@px1:~# apt-get update && apt-get dist-upgrade -y
```

A possible problem occurred after the update on one of the nodes. The following command and its output may be an example of that mistake:

```
root@px3:~# pveversion -v
```

-bash: pveversion: command not found

Usually if the pveversion is found and shows information about the version of Proxmox components. It can be used to check, if the update succeeded. The versions should be the same or higher than in following output:

proxmox-ve: 4.4-86 (running kernel: 4.4.49-1-pve)

pve-manager: 4.4-13 (running version: 4.4-13/7ea56165)

pve-kernel-4.4.49-1-pve: 4.4.49-86

lvm2: 2.02.116-pve3

corosync-pve: 2.4.2-2~pve4+1

libqb0: 1.0.1-1

pve-cluster: 4.0-49

qemu-server: 4.0-110

pve-firmware: 1.1-11

libpve-common-perl: 4.0-94

libpve-access-control: 4.0-23

libpve-storage-perl: 4.0-76

pve-libspice-server1: 0.12.8-2

vncterm: 1.3-2

pve-docs: 4.4-4

pve-qemu-kvm: 2.7.1-4

pve-container: 1.0-97

pve-firewall: 2.0-33

pve-ha-manager: 1.0-40

kvm-control-daemon: 1.2-1

glusterfs-client: 3.5.2-2+deb8u3

lxc-pve: 2.0.7-4

lxcfs: 2.0.6-pve1

criu: 1.6.0-1

novnc-pve: 0.5-9

smartmontools: 6.5+svn4324-1~pve80

zfsutils: 0.6.5.9-pve15~bpo80

The solution that helps in a case of this error with no output (because of this mistake web interface is not working too.) Command installs the missing components back, making a server working again.

```
root@px3:~# apt-get install proxmox-ve
```

After that fix the configuration process can be continued and interfaces are getting ready for Ceph. Eth0 is configured for regular network, that's why eth1 is used for Ceph. The commands bellow shows the configuration that is needed to be added to all servers. There are px3, px2 and px1 servers with a different IP configurations. Picture 3 shows how the configured interfaces look like in a GUI. A GUI can be used for the interface configuring too, but the CLI was opened from the previously used commands and it was faster to continue using it.

```
root@px3:~# nano /etc/network/interfaces
```

```
auto eth1
```

```
iface eth1 inet static
```

```
address 10.10.10.1
```

```
netmask 255.255.255.0
```

```
root@px2:~# nano /etc/network/interfaces
```

```
auto eth1
```

```
iface eth1 inet static
```

```
address 10.10.10.2
```

```
netmask 255.255.255.0
```

```
root@px1:~# nano /etc/network/interfaces
```

```
auto eth1
```

```
iface eth1 inet static
```

```
address 10.10.10.3
```

```
netmask 255.255.255.0
```

Name ↑	Type	Active	Autostart	Ports/Slaves	IP address	Subnet mask	Gateway	Comment
eth0	Network Device	Yes	No					
eth1	Network Device	Yes	Yes		10.10.10.1	255.255.25...		
vbr0	Linux Bridge	Yes	Yes	eth0	192.168.12...	255.255.25...	192.168.12...	

Figure 4. Node 3 network interfaces

After that, all nodes should be rebooted for the network settings update and the Ceph installation can begin.

6.2.2 Ceph

As soon as Ceph is installed as a component of Proxmox and can be controlled from the same GUI (which is accessible from the node IP address with port 8006 added) or CLI, configuration can be continued on the same servers. The next command is issued on all nodes to install software packages. Jewel stands for Ceph version, another version can be selected at this stage.

```
pveceph install -version jewel
```

After that the Ceph network can be initialized and get ready to use.

```
px3# pveceph init --network 10.10.10.0/24
```

Monitors are required for a cluster map copy, and after the activation of at least one monitor, configuration can be also continued in a graphical user interface (Not all required commands are supported in a GUI!). The picture below shows a GUI of Ceph, which is included to a Proxmox. There are no monitors configured and the GUI is showing error status.

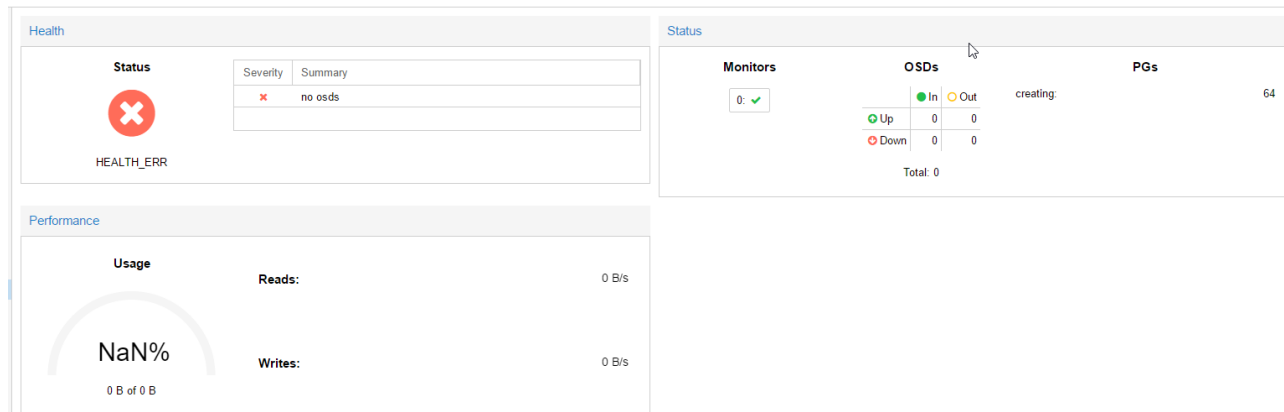


Figure 5. GUI status screen

The next command should be issued at all future monitors. Ceph can work even on one monitor, but in that case the failure of that monitor will destroy whole storage cluster. An example of the successful output is included with the command used, which can be seen below.

```
root@px3:~# pveceph createmon
creating /etc/pve/priv/ceph.client.admin.keyring
monmaptool: monmap file /tmp/monmap
monmaptool: generated fsid 7f9547a6-2d27-4961-871e-b0acf4e59671
epoch 0
fsid 7f9547a6-2d27-4961-871e-b0acf4e59671
last_changed 2017-04-06 10:21:45.530617
created 2017-04-06 10:21:45.530617
0: 10.10.10.1:6789/0 mon.0
monmaptool: writing epoch 0 to /tmp/monmap (1 monitors)
ceph-mon: set fsid to be38de5b-f29c-44cb-8e31-54385b048405
ceph-mon: created monfs at /var/lib/ceph/mon/ceph-0 for mon.0
Created symlink from /etc/systemd/system/ceph-mon.target.wants/ceph-mon@0.service to /lib/systemd/system/ceph-mon@.service.
```

To get a list of all the drives included in a server an lsblk command may be used. In the command output, which is written after this sentence, two drives (sda and sdb) can be seen.

```
root@px1:~# lsblk
```

```

NAME                MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
fd0                  2:0  1    4K 0 disk
sda                  8:0  1 136.8G 0 disk
├─sda1                8:1  1    1M 0 part
├─sda2                8:2  1   256M 0 part
└─sda3                8:3  1 136.6G 0 part
   ├─pve-root          251:0  0   34G 0 lvm /
   ├─pve-swap          251:1  0    8G 0 lvm [SWAP]
   ├─pve-data_tmeta    251:2  0   80M 0 lvm
   │ └─pve-data        251:4  0  78.6G 0 lvm
   └─pve-data_tdata    251:3  0  78.6G 0 lvm
       └─pve-data      251:4  0  78.6G 0 lvm
sdb                  8:16  1  465G 0 disk
├─sdb1                8:17  1  460G 0 part
└─sdb2                8:18  1    5G 0 part

```

After issuing a list of drives, the name of the right drive can be checked. In this project, the drive reserved for Ceph was sdb. As soon as it is not empty, overwrite is required before creating OSD on it with command:

```
# ceph-disk zap /dev/sdb
```

The createOSD command can be used after that. It should be done on all drives, but it would be the right idea to check the drive needed before overwriting or creating OSD to be sure that no useful data will be deleted. The createOSD command itself:

```
# pveceph createosd /dev/sdb
```

After the reboot of nodes, new drives can be seen in GUI as OSD drives. The picture below shows the actual OSD look from GUI.

Node 'px3'

Buttons: Create OSD, Reload, Set noout

Name	Ty...	Status	weight	reweight	Used		Latency (ms)	
					%	Total	Apply	Commit
default	root							
px3	host							
osd.2	osd	up / in	0.449692	1	0.01	460.54 GiB	1568	35
px2	host							
osd.1	osd	up / in	0.128799	1	0.02	131.85 GiB	534	21
px1	host							
osd.0	osd	up / in	0.448899	1	0.01	459.76 GiB	1	0

Figure 6. Installed OSD

An example of a typical RBD storage configuration:

```
# from /etc/pve/storage.cfg
```

```
rbd: cephtest
```

```
monhost 10.10.10.1;10.10.10.2;10.10.10.3
```

```
pool rbd
```

```
content images
```

```
username admin
```

```
krbd 0
```

RBD may also be added using GUI by the button “Add Storage: RBD”. The next commands are issued to create a directory for the Ceph key and to distribute it to all the nodes in a cluster.

```
root@px3:~# cd /etc/pve/priv/
```

```
root@px3:/etc/pve/priv# mkdir ceph
```

```
root@px3:/etc/pve/priv# cp /etc/ceph/ceph.client.admin.keyring
```

```
ceph/cephtest.keyring
```

RBD storage is used to store the disk image files. Ceph RBD shows total size as 1.03TiB (check picture 7), that includes two 500GiB drives and a 128GiB drive. It

may be needed to calculate a size of needed storage before using free space on drive. In case of one drive fail, available space will be needed for data recovering.



Figure 7. RBD storage status

At this point Ceph can be considered as installed and first tests can be started. The next section will describe a VM creation and some simple tests.

6.2.3 Proxmox VM: create and test

Virtual machine can be created using a Create VM button in the upper right corner of GUI. It can be found near the logout button, it can be seen on a picture below.

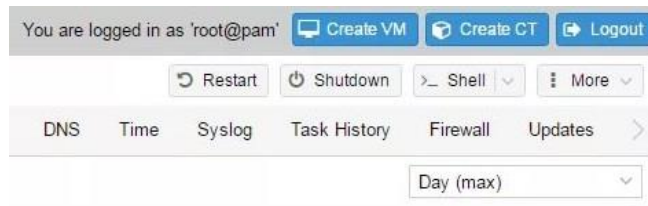


Figure 8. A create VM button.

It is important to select the storage needed when creating a VM, because even if it is possible to migrate/move or copy a machine to another storage, it is not instant. The storage used now is cephtest. On a picture below, it should be selected instead of a local-lvm.



Figure 9. A storage for the VM HDD

The amount of memory can be selected during the creation, or automatic allocation may be used. For this step Windows 7 x32 was used. The installation from ODD has taken at least thirty minutes, later network installation was tested, and it took around ten minutes. A machine may refuse to start in case if kvm is not supported by the hosts processor. A possible solution is disabling kvm in the VM options-edit. If the machine was installed from a physical drive it might be useful to delete the CD/DVD Drive in the settings. Otherwise, the migration or HA may fail. Besides a Windows 7 x32 machine, Windows XP and Ubuntu VMs were created on the other two nodes to give minimum load.

At this point, I want to mention that live migration is possible only if both nodes (the one, which is host now, and the one, which is future host) are alive. In case of an emergency shutdown, VM will be restarted, because RAM can't be transferred from an offline host. Therefore, VM restart is a normal behavior in case of a node fail. It does not depend on the type of storage used.

The target is to make a HA cluster for the virtual machines and a resource group is a thing that should be created to make it possible. To create a group, it is needed to enter the HA settings in a GUI datacenter tab and click create button. For this project a group with ID HAtest was created, and it included all the nodes: px1, px2 and px3. In the resource tab, VM should be associated with the group by the add button, where VM and group are selected by ID. The next picture shows how a machine managed by HA looks like.

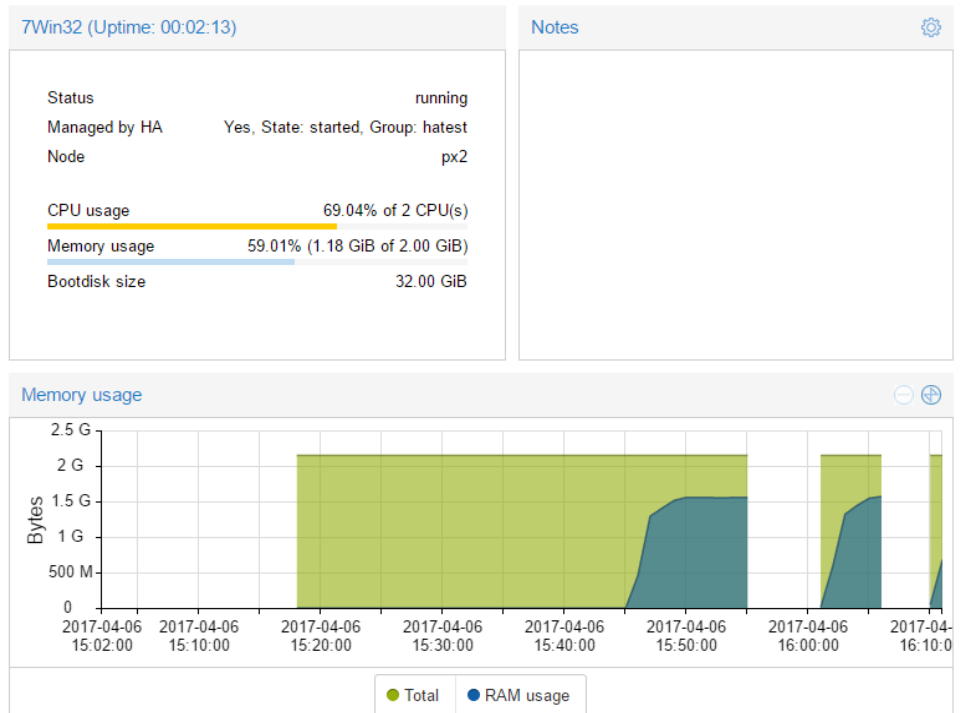


Figure 10. Working virtual machine statistics

The first tests were not successful because of two processors added to the tested VM. Not all the servers used in the practical part of this thesis were powerful enough to run this machine together with other machines, migration and HA didn't work. To make subsequent testing possible without stopping the other VMs, Windows 7 x32 settings were edited to use only one processor.

There are two situation described, which were used to test both Ceph storage recovery and Proxmox VM restart. Situation one: one VM active on node px2 and one VM active on node px3. After the px3 node shutdown, connection to GUI was lost (as soon as it is the main node, and I was connected to it using 192.168.128.202 IP), but it was possible to connect to the cluster using px2 and px1 IP addresses. To restart a VM Proxmox needed around two minutes to make sure the node is dead and then started Windows 7 on a px2 node. In parallel, Ceph was recovering data from the lost drive. VM was back online in less than three minutes and Ceph was fully recovered in approximately forty minutes. Situation two: node back online before Ceph had finished recovering. Ceph immediately checked that data is fully accessible again and showed 100% health.

All the shutdowns were made with detaching a server from the power supply to be sure that nothing may inform other nodes about future connection lost.

In case of two servers dead, the last node decides that the problem is on its side and starts rebooting; in that case, all the working machines will be dead.

Live migration can be started using the button Migrate. The Migrate button is located below the Create VM button. Ping of a migrating machine was stable during a whole migration process, the Windows 7 machine was accessible from a console. Time spent for the operation under given conditions: twenty minutes.

The picture number 11 shows almost whole Proxmox interface and shows how a starting Windows7 machine looks like. At this picture such buttons as Create VM and Migration can be seen in the right upper corner, as it was described earlier.

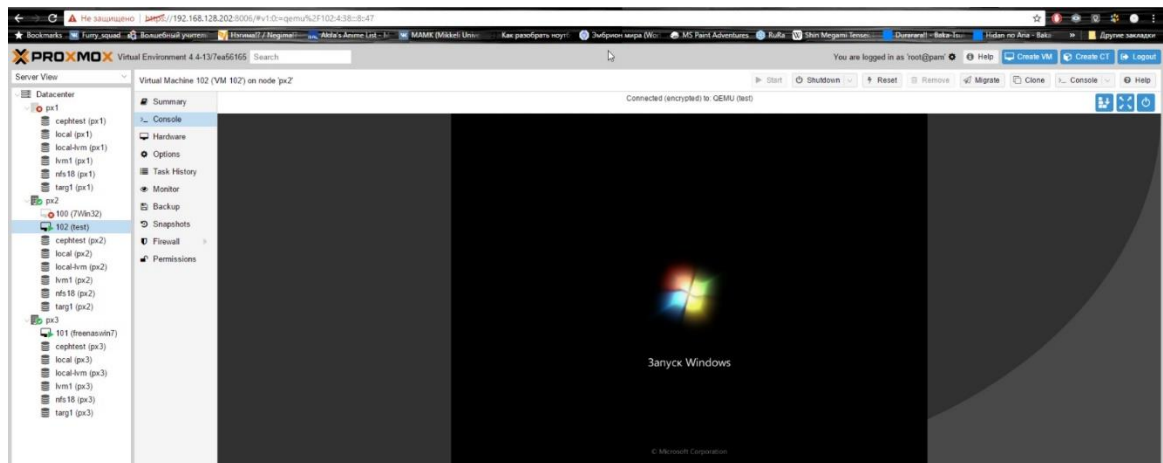


Figure 11. Running Windows 7

The next picture shows how a VM and a node status looks like from a GUI. The Status table shows a status of nodes. A dead node px1 can be seen on it, but quorum status is OK because two other nodes are alive. The Resources table shows the VMs states. It includes the current machines statuses and basic information on the configurations: ID, maximum restart count, maximum relocation count and group.

Status						
Type	Status					
quorum	OK					
master	px3 (active, Thu Apr 20 16:55:02 2017)					
lrm	px1 (old timestamp - dead?, Thu Apr 20 16:53:28 2017)					
lrm	px2 (active, Thu Apr 20 16:55:01 2017)					
lrm	px3 (active, Thu Apr 20 16:54:56 2017)					

Resources						
<input type="button" value="Add"/> <input type="button" value="Edit"/> <input type="button" value="Remove"/>						
ID	State	Node	Max. Restart	Max. Reloc...	Group	Description
vm:100	migrate	px2	1	1	hatest	
vm:101	started	px3	1	1	hatest	
vm:102	fence	px1	2	1	hatest	

Figure 12. Status window for servers and VMs

The same tests with a same result were done with a FreeNAS storage, so the results will not be included again. The next section describes how exactly FreeNAS can be installed.

6.2.4 FreeNAS

As soon as Ceph storage requires additional server drives and uses a part of the server RAM, separate storage may be a better choice in a situation with a drive limit or if servers need all their available resources.

For a separate storage, FreeNAS was installed on a user PC, but in a real case, it should be installed at least on a server. Installation was done the same way as with Proxmox, from a virtual ODD. An 192.168.128.203 IP was reserved for a storage. FreeNAS may be installed with the default settings. After installation, it may be useful to change IP address of a storage, if DHCP is not available.

It is possible to install FreeNAS OS as a VM on Proxmox, but I am not sure if the migration of working machines and storage OS will work correctly simultaneously, so this kind of a storage has vulnerability: in case of FreeNAS OS failure it should be restarted by an administrator (of course if it will not restart on its own).

Otherwise whole structure will be down without storage.

FreeNAS requires at least two drives (or a Flash drive for the operating system), one for OS and at least one for storage. Root password is created during installation, but it may be also edited later with GUI. An example of FreeNAS CLI can be seen on a picture below. On the same picture an instruction of how to connect to a GUI is included. It can be done using browser and machine IP, which is 192.168.128.203.

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

You may try the following URLs to access the web user interface:
http://192.168.128.203
Enter an option from 1-14: █
```

Figure 13. FreeNAS console

To create a storage for the VM drive or image files a volume should be created first. This process will erase an existing data on a drive. To do that, in a volume manager an available disk should be selected and then the volume layout is chosen. With three or more drives, the creation of the RaidZ is possible. In a Volume manager, the Volume name is selected. `zfspxfreenas` in my case, and then the available disks must be selected. The picture below shows how an interface of a volume manager looks like and also shows that a total size of the three selected drives will be 96Gb.

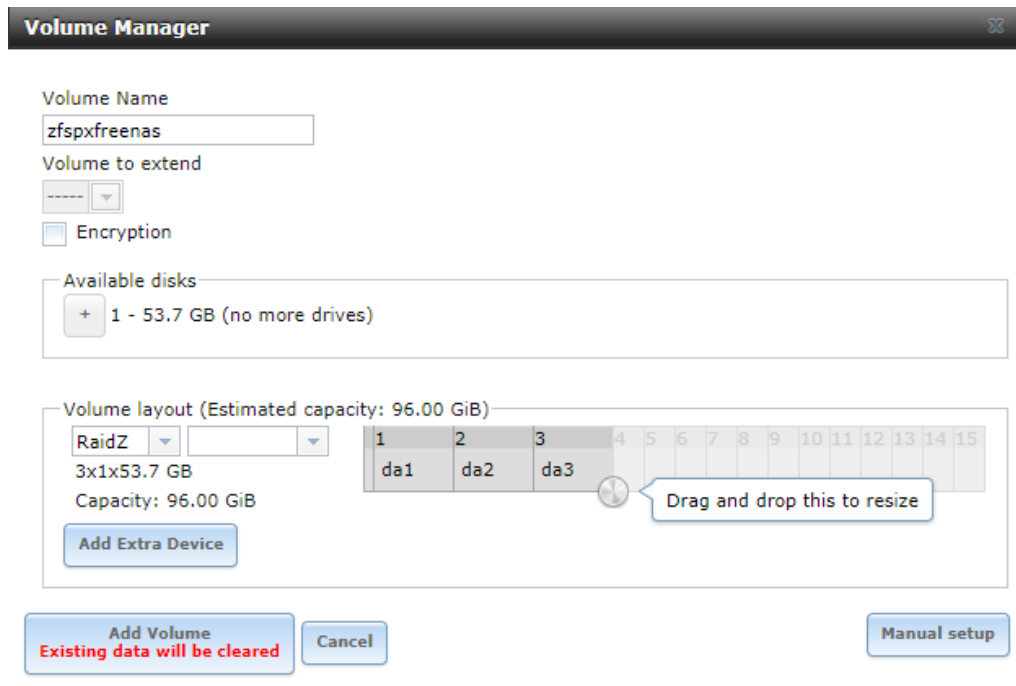


Figure 14. FreeNAS Volume manager

For the next step, it is possible to add a dataset to be able to set permissions, configure deduplication, compression and quotas for an exact dataset.

Permissions are also needed to provide access to the storage. For the dataset creation volume should be selected. Otherwise, the button will not appear. It is not really necessary because a dataset with the name of volume is created automatically, but it is still possible and may be useful in case of need in many different directories with various access levels required. In my project, the dataset `zfspxtest` was created.

As mentioned above, permissions may be edited after creation of a dataset. For test purposes it is possible to make a dataset read/write function available to any user, that is risky in real word, because anyone can edit anything with that configuration. To make it possible, a “write – other” square may be marked. On a picture below this square is not marked, that makes in not possible for any user to write data on a drive.

Change Permissions
✕

Change permission

Change permission on /mnt/zfspxfreenas/zfspxttest to:

Apply Owner (user):	<input checked="" type="checkbox"/>																	
Owner (user):	<input type="text" value="root"/>	i																
Apply Owner (group):	<input checked="" type="checkbox"/>																	
Owner (group):	<input type="text" value="wheel"/>	i																
Apply Mode:	<input checked="" type="checkbox"/>																	
Mode:	<table style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th></th> <th style="text-align: center;">Owner</th> <th style="text-align: center;">Group</th> <th style="text-align: center;">Other</th> </tr> </thead> <tbody> <tr> <td>Read</td> <td style="text-align: center;"><input checked="" type="checkbox"/></td> <td style="text-align: center;"><input checked="" type="checkbox"/></td> <td style="text-align: center;"><input checked="" type="checkbox"/></td> </tr> <tr> <td>Write</td> <td style="text-align: center;"><input checked="" type="checkbox"/></td> <td style="text-align: center;"><input type="checkbox"/></td> <td style="text-align: center;"><input type="checkbox"/></td> </tr> <tr> <td>Execute</td> <td style="text-align: center;"><input checked="" type="checkbox"/></td> <td style="text-align: center;"><input checked="" type="checkbox"/></td> <td style="text-align: center;"><input checked="" type="checkbox"/></td> </tr> </tbody> </table>			Owner	Group	Other	Read	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Write	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Execute	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
	Owner	Group	Other															
Read	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>															
Write	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>															
Execute	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>															
Permission Type:	<ul style="list-style-type: none"> <input checked="" type="radio"/> Unix <input type="radio"/> Mac <input type="radio"/> Windows 																	
Set permission recursively:	<input type="checkbox"/>																	

Change
Cancel

Figure 15. Dataset permission settings

Zvol is created next, size 50Gb and the name zvol1 was selected for it. On the next picture a zvol create interface can be seen.

inner (124) Linux

Create zvol

Create zvol on zfspxfreenas/zfspxtest

zvol name:

Comments:

Size for this zvol: ⓘ

Force size: ⓘ

Compression level: ▾

Sparse volume: ⓘ

Figure 16. Zvol

As far as this storage is not a feature included in Proxmox directly, it should be linked to a cluster using iSCSI. To do that, firstly a portal must be added. The portal is needed to announce targets list (IP addresses and TCP port numbers). To create the portal it is needed to go to the Sharing – Block (iSCSI) sub-menu. 0.0.0.0:3260 is used for a portal address by default.

Configuring initiators is the next thing needed, because it may prevent access by unknown initiators. ALL-ALL at the picture 17 means that any initiator may access storage.

Add Initiator ✕

Initiators:	<input style="width: 95%;" type="text" value="ALL"/>	i
Authorized network:	<input style="width: 95%;" type="text" value="ALL"/>	i
Comment:	<input style="width: 80%;" type="text"/>	i

OK
Cancel

Figure 17. Initiator

After that extents are configured. They may be file or device based. An earlier created zvol1 is used as device for the extent. As it can be seen on a picture 18, extent name selected is ext1.

Extent Name:	<input style="width: 95%;" type="text" value="ext1"/>	i
Extent Type:	<input style="width: 95%;" type="text" value="Device"/>	
Device:	<input style="width: 95%;" type="text" value="zfspxfreenas/zfspxtest/zvol1 (50.0 GiB)"/>	
Serial:	<input style="width: 95%;" type="text" value="000c294f6cc500"/>	i
Logical Block Size:	<input style="width: 95%;" type="text" value="512"/>	i
Disable Physical Block Size Reporting:	<input type="checkbox"/>	i
Available Space Threshold (%):	<input style="width: 95%;" type="text"/>	i
Comment:	<input style="width: 95%;" type="text"/>	i
Enable TPC:	<input checked="" type="checkbox"/>	i
Xen initiator compat mode:	<input type="checkbox"/>	i
LUN RPM:	<input style="width: 95%;" type="text" value="SSD"/>	i
Read-only:	<input type="checkbox"/>	

OK
Cancel
Delete

Figure 18. Extent

The iSCSI target is created next. The portal group ID and initiator group ID is 1 (automatically assigned number for portal group created earlier), name of target is targ1. The button Add Target/Extent in the menu Associated Targets associates the created target with extent, and example of this menu is shown on a picture 19. Now iSCSI can be enabled through Services menu.

The image shows a dialog box titled "Add Target / Extent". It contains three dropdown menus: "Target:" with the value "targ1", "LUN ID:" with the value "0", and "Extent:" with the value "ext1". To the right of the "Target:" dropdown is an information icon. At the bottom of the dialog are two buttons: "OK" and "Cancel".

Figure 19. Associated target

After that on a Proxmox server iSCSI must be added. To do that in a datacenter menu of a Proxmox GUI, storage sub-menu via button add iSCSI must be selected. ID targ1 was used to easier association with FreeNAS, portal is IP of storage (which is 192.168.128.203). The target may be selected after IP is entered. LVM should be created after that. It can be done in the same menu with the add LVM button. Selected ID was lvm1, base storage targ1, volume group name lvmg1pxtest. How it should look can be seen on a picture below. On a background of that picture other created storages can be seen too.

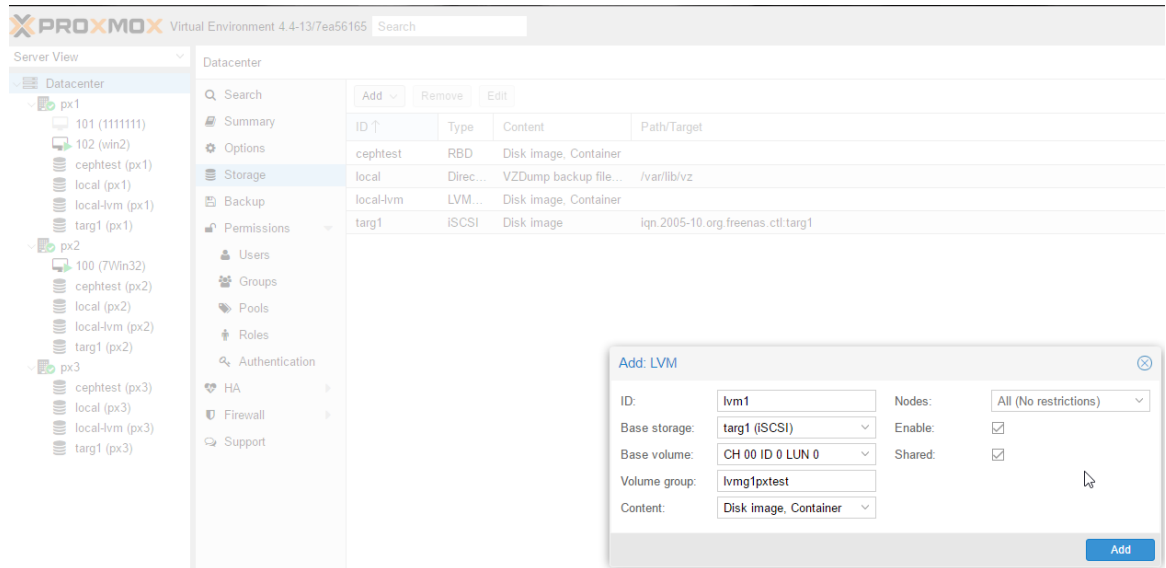


Figure 20. Add: LVM

Now virtual machine can be created with an lvm1 storage. As it was said in a 6.2.2. section right storage should be selected. On a picture 21 the selecting process is shown.

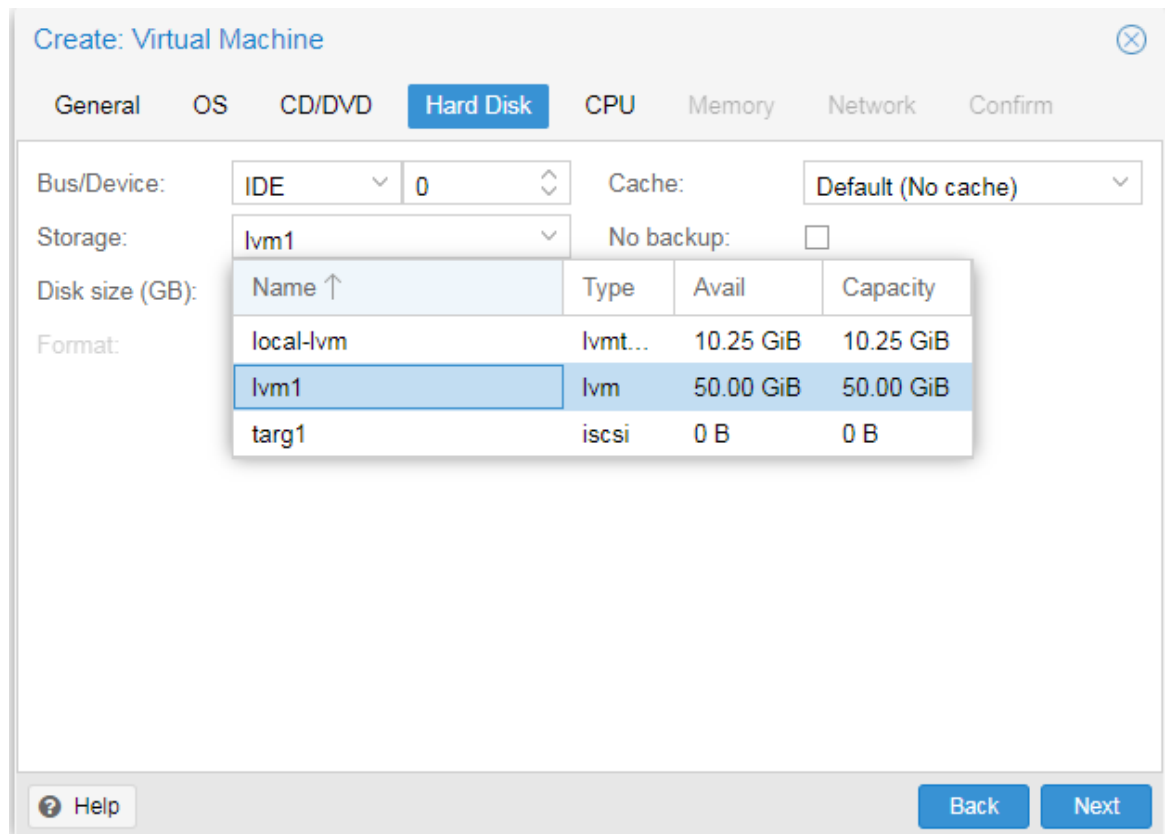


Figure 21. lvm1 as a storage

In case if not the whole space of FreeNAS is going to be used for the Proxmox VMs, it may be useful to create a Windows or Linux share using FreeNAS. It may be used for regular data or for the .iso files for an easier VM installation.

Creating shares is possible with a sharing menu, which includes NFS and SMB shares. With SMB share, it is possible to create folders for regular Windows computers. At the other hand, NFS is easy to use with Proxmox or other Linux based systems. In Proxmox, this kind of storage may be used to store .iso files for fast virtual machine installation without any external disks. In addition, the installation of an image from network storage is often faster than installation with a USB or an optical disk. A picture below shows an add menu for the NFS Share.

Add Unix (NFS) Share

Path

Path: Close

- /
- mnt
 - + vol1

[Add extra Path](#)

Comment:

All Directories: i

Read Only: i

OK Cancel Advanced Mode

Figure 22. NFS share

The next picture is attached to show an example of a working node with all the storages attached. It can be seen on the left side, under any node name. There are also statistics of a selected node, and one VM with ID 100 can be seen on a px3 node.

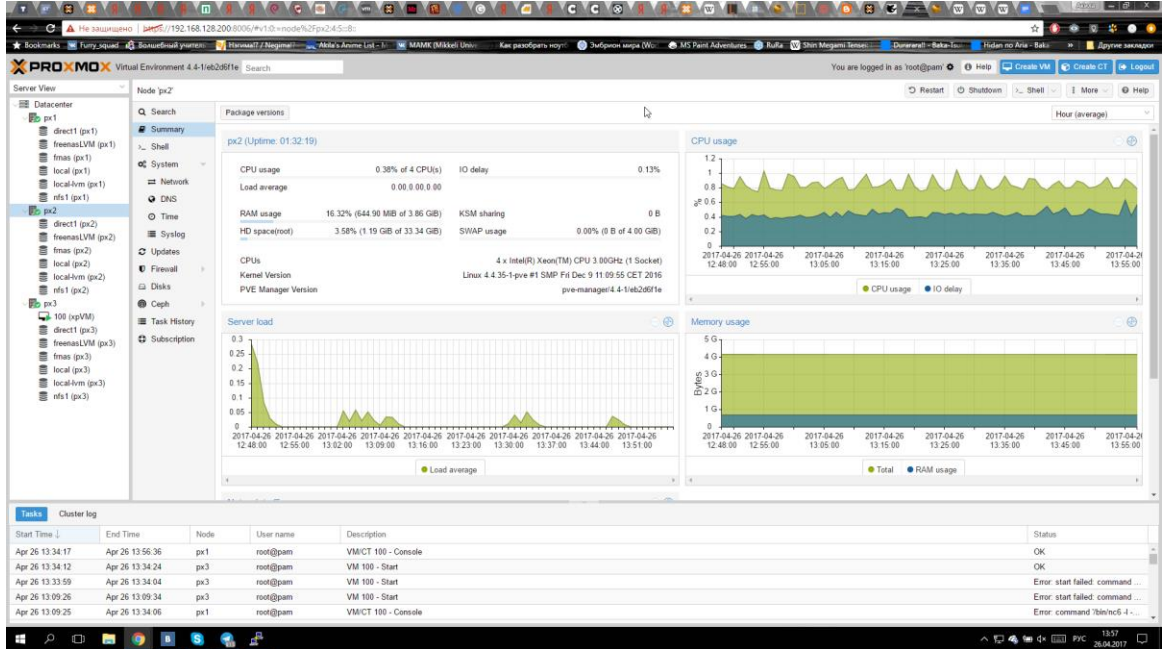


Figure 23. Example of a working cluster with all drives attached

If in some case a drive with the Proxmox system is overwritten, erased, dead or node is not needed anymore, this node can be removed from the cluster using the instructions from the official Proxmox wiki [www page](https://pve.proxmox.com/wiki/Main_Page) https://pve.proxmox.com/wiki/Main_Page.

In case of planning a removal, all the machines should be migrated or deleted from the node before the removal. All the backups, if there are any stored locally, should be moved too. To check the node ID a `pvecm status` command is issued. The node must be offline or turned off before the removal. `pvecm delnode px3` is a command needed to delete the node px3 from a cluster. If the same server is needed to get back to the same cluster for some reason, Proxmox should be installed again (reinstalled) and a node is added just as a new one. It is possible to re-install a node and to get it back to cluster with the same settings, but that was not checked in my project.

7 CONCLUSIONS

The aim of my study was to get a working HA cluster with a possibility to support multiple VMs and an opportunity to work even in case of one server down. In case of need third server may be replaced with a low cost machine needed only for quorum, Proxmox is not much different from other hypervisors mentioned in the theory part of this thesis. This project was really useful for me, because it was the first time configuring not a virtual server, but a real one and without any control from other persons. Even if the devices used were outdated, the process is quite similar to the ones done on newer equipment. I was trying to make a working cluster with a storage and that was complete.

The theoretical part of my work explained some basic information on virtualization, hypervisors and storages. I listed four hypervisors, two of them are free and two are with a payment even for small business. As my aim was to create one cluster, I chose one of free hypervisors looking more user friendly and having more documentation with free access. A company I was doing the thesis for is small, so there was no need to create an industrial-sized datacenter. That's why Proxmox covers all the needs of that company.

The practical part of my thesis described the installation of a Proxmox cluster on the real servers and creating two possible storages for it. The reason to test two storage software was a problem of low amount of RAM on the testing equipment. Both FreeNAS and Ceph are free as well as Proxmox, which simplifies testing in monetary terms and makes this set easily applicable for small or medium sized companies. The security part of the settings was not described in my work, because they must be done together with securing the whole network and user PCs. The settings described include open access to a storage, which is not applicable for a real network. The description of an UPS configuration is not included because the main target of my work was to test the cluster software part, not the storage physical installation.

For a proper future use of the configured cluster, it should be reinstalled on the newest servers with the configuration matching main network of a company. Storage should not be created on a user PC, even if that is enough for testing. During the writing of this thesis newer versions of the software used were released, and the settings described are relevant for Proxmox 4.4 and FreeNAS 9.10. For installation on working devices, it is planned to use newly released Proxmox 5 and FreeNAS 11 versions.

REFERENCES

Ahmed, W. 2015. Proxmox Cookbook. Birmingham. Packt Publishing Ltd

Rajkumar, B. 1999. High Performance Cluster Computing: Architectures and Systems, Vol. 1. New Jersey. Prentice Hall PTR

Singh, K. 2015. Learning Ceph. Birmingham. Packt Publishing Ltd

Merkey, P. Beowulf history. WWW document

<http://www.beowulf.org/overview/history.html> [Accessed 20 September 2017]

Ceph server, 2017. Proxmox wiki. Updated 4 July 2017.

https://pve.proxmox.com/wiki/Ceph_Server [Accessed 20 September 2017]

Borzenko, A. 2002. Clusters. *BYTE № 5(46)*.

Ahmed, W. 2016. Mastering Proxmox 2ed. Ebook. Available at:

<http://smtebooks.com> [Accessed 20 September 2017]

iXsystems, 2017. FreeNAS Documentation project. <http://doc.freenas.org/>

[Accessed 20 September 2017]

Pfister, G. 1997. In search of Clusters (2nd Edition). New Jersey. Prentice Hall

Radajewsky, J. Eadline, D. 1998. Beowulf HOWTO. WWW document

<http://ibiblio.org/pub/Linux/docs/HOWTO/archive/Beowulf-HOWTO.html>

[Accessed 20 September 2017]

Clerk002, 2012. Introduction to HPCC. WWW document

<http://en.community.dell.com/techcenter/high-performance-computing/w/wiki/hpc-cluster-nodes> [Accessed 20 September 2017]

Rajagopal, R. 2000. Introduction to Microsoft Windows NT cluster server: Programming and administration. Boca Raton. CRC Press LLC.

Virtualization: a key to virtualization world. Wikipedia book. Prediapress. Ebook. Available at: <https://books.google.ru/> [Accessed 20 September 2017]

FreeNAS official website, 2017. iXsystems, Inc. WWW document. <http://www.freenas.org/> [Accessed 20 September 2017]

Proxmox wiki page, 2017. WWW document. https://pve.proxmox.com/wiki/Main_Page [Accessed 20 September 2017]

Louwrentius, 2017. WWW document. <http://louwrentius.com/the-hidden-cost-of-using-zfs-for-your-home-nas.html> [Accessed 20 September 2017]