

Ali Anafcheh

Private Cloud Storage Home Server

Project Work
Advanced Server Development Project
Information Technology

CONTENTS

1	INTRODUCTION	1
2	VIRTUALIZATION	2
2.1	Consolidation	4
2.2	Redundancy and Recovery.....	4
2.3	Isolated Environment	5
3	PROXMOX VE.....	5
4	OWNCLOUD	6
5	PRACTICAL PART	7
5.1	Installing Proxmox on an External HDD	8
5.2	Proxmox Configuration.....	9
5.2.1	Network Configuration	9
5.2.2	Storage Configuration.....	10
5.3	Virtual Machines.....	10
5.4	Virtual Machine 100 Debian 8 – Installing ownCloud 9	11
5.4.1	Configuring OpenSSL	13
5.4.2	Configuring MariaDB.....	16
5.5	Virtual Machine 101 PfSense	16
5.5.1	Assigning Interfaces	17
5.5.2	Set Interfaces IP addresses.....	18
5.6	Virtual Machine 102 Lightweight Linux Distribution	19
5.7	ownCloud Web Interface	19
5.8	PfSense Web Interface	20
5.8.1	OpenVPN Server	21
5.9	Port Forwarding	24
6	AUTOMATION	25
7	CONCLUSIONS	25
	BIBLIOGRAPHY	27

1 INTRODUCTION

During the past semesters, I had a few courses about servers and virtualization in which we worked with software like Microsoft Hyper-V and VMware vSphere and server operating systems such as Microsoft Windows Server and Ubuntu Server. Due to my deep interest in this field, I have decided to research more about free open source alternatives of server virtualization and experience how to work with at least one of them. The aim of this study is to familiarize the reader with the basics of virtualization and demonstrate Server Virtualization with Proxmox VE which is free and open source.

2 VIRTUALIZATION

Virtualization is the creation of an instance of something in the virtual world. Virtualization creates the possibility to have multiple virtual instances on a single physical device. Networks, storage, operating systems etc. can be virtualized to simply save money, time, and increase efficiency and productivity. Virtualization can be used to consolidate a group of hardware and used them as if they are one single physical device. This is used in networking such as combining NICs, in storage such as combining a group of hard disk drives and in other areas. Most of us have used virtualization in our life, for example, when partitioning a hard disk drive.

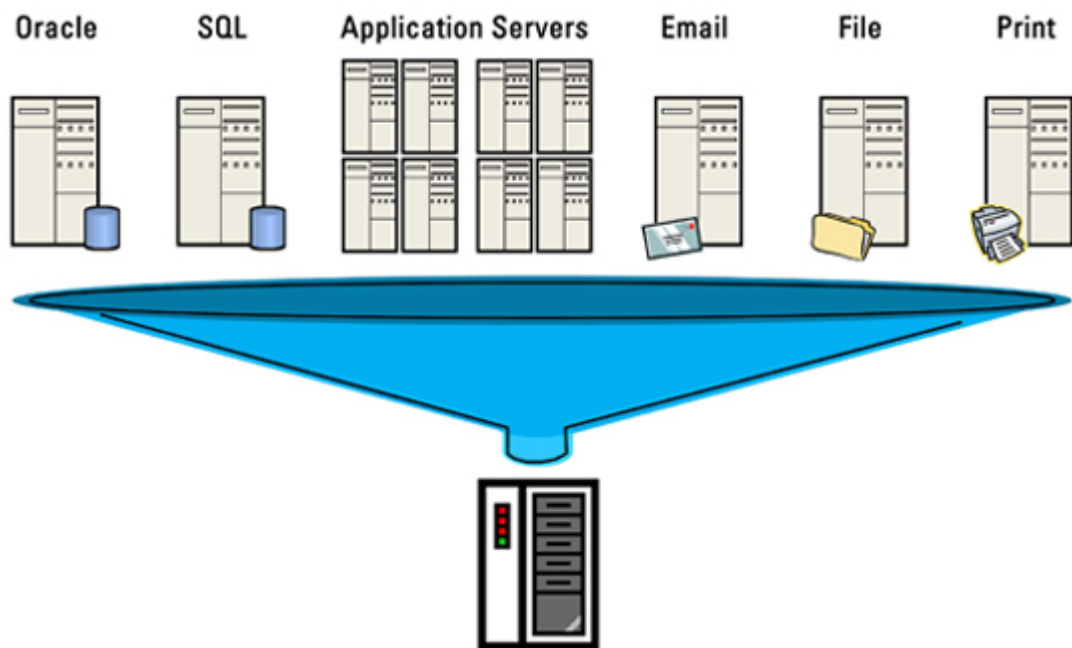


Figure 1. Virtualization

Hypervisors or Virtual Machine Monitor (VMM) make virtualization possible. The hypervisor manages the connection between the virtual machine and the hardware. It separates the hardware from the software which this allows to define the accessibility level

of a virtual machine to the hardware such as the use of RAM, CPU, storage etc. Therefore, multiple virtual machines with different operating systems and different access levels can be created in a single physical device.

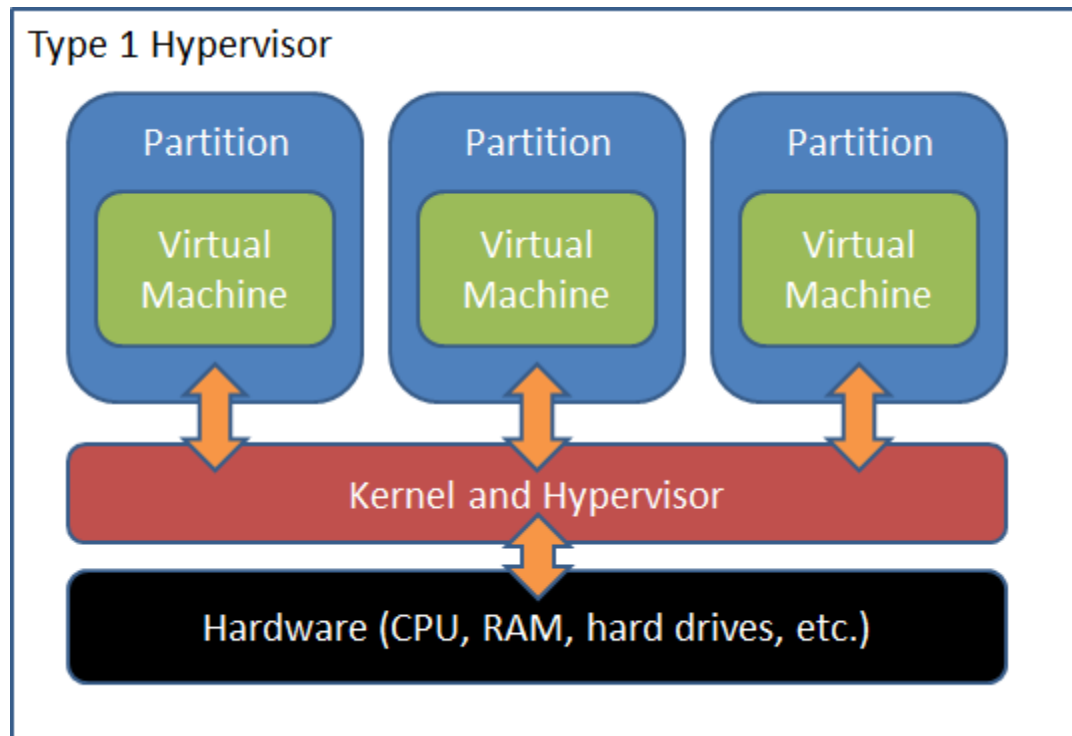


Figure 2. Hypervisor Type 1

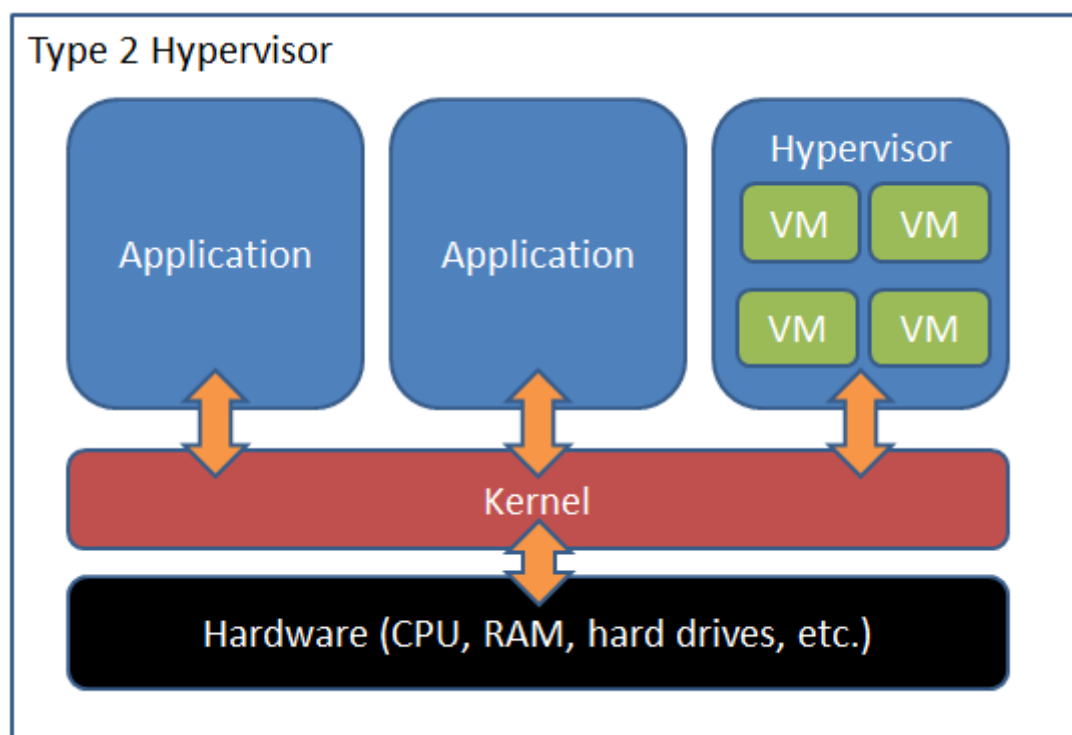


Figure 3. Hypervisor Type 2

A decade ago, without virtualization, one of the biggest problems was that the x86 servers are capable of running one operating system at a time. So, it was not possible to use the full capacity of a server and different machines had to be purchased to do one task. Virtualization is a very useful and important technology even for small businesses. With other different types of virtualization such as network virtualization, application virtualization and storage virtualization, the advantages of this technology are far more than some minor disadvantages and issues such as storage and I/O. So, the question is why do we need virtualization, especially server virtualization?

2.1 Consolidation

By running multiple virtual machines on a single server and utilizing the full capacity, a data center would need much less servers, racks etc. For big data centers, this brings a significant decrease in the operating costs. Saving energy and space makes virtualization an environment friendly technology.

2.2 Redundancy and Recovery

Many businesses use different servers for each application as they do not want to risk application interference as a crash of a single application might cause other applications to crash as well which this will cause more server downtime. With virtualization, running the same application on different servers can help to minimize server downtime and recover instantly in case a server crashes. Backup of virtual machines makes recovering even from unwanted accidents such as natural disasters very fast. However, the security of the virtual machines must be taken into account as they can be easily deleted

by a simple human error or even stolen on a USB flash disk. These kind of threats are significantly lower for physical servers.

2.3 Isolated Environment

By having an isolated virtual environment, one can test and experience different applications or operating systems without the need to buy new hardware. This also eliminates the dependency on specific vendors that require their hardware to work with specific kind of software only. Hence, when the hardware of a server is out of date and some application still depend on it, using virtualization the legacy hardware can be easily virtualized. This improves business continuity considerably as well. In addition, using virtual machines, security and crucial updates can be tested first before being implemented on an actual system which this makes them great to use as a lab environment.

3 PROXMOX VE

Proxmox VE is a server virtualization application which is completely open source and free. Proxmox VE is build based on Kernel-based Virtual Machine (KVM) which is a virtualization solution for x86 hardware and Container-based virtualization which is for running Linux servers. The application development is supported through premium subscriptions which are to support the users with updates that are necessary if you are using this software in a production environment. However, compared to their competitors, the prices are very cheap starting from € 4, 99. This gives you access to Enterprise Repository, stable software updates and support via community forum and that is all you need. During my project research about open-source virtualization solutions, Proxmox VE attracted me the most.

Here's a comparison carried out by Proxmox which compares its virtualization application with their biggest competitors in this sector whom are VMware, Microsoft and Citrix.

	Proxmox VE	VMware vSphere	Windows Hyper-V	Citrix XenServer
Guest operating system support	Windows and Linux (KVM) Other operating systems are known to work and are community supported (OpenVZ supports Linux only)	Windows, Linux, UNIX	Modern Windows OS, Linux support is limited	Most Windows OS, Linux support is limited
Open Source	yes	no	no	yes
OpenVZ container (known as OS Virtualization)	yes	no	no	no
Single-view for Management (centralized control)	yes	Yes, but requires dedicated management server (or VM)	Yes, but requires dedicated management server (or VM)	yes
Simple Subscription Structure	Yes, one subscription pricing; all features enabled	no	no	no
High Availability	yes	yes	Requires Microsoft Failover clustering; limited guest OS support	Proxmox VE compared to other virtualizat
Live VM snapshots: Backup a running VM	yes	yes	limited	yes
Bare metal hypervisor	yes	yes	yes	yes
Virtual machine live migration	yes	yes	yes	yes
Max. RAM and CPU per Host	160 CPU/2 TB Ram	160 CPU/2 TB Ram	64 CPU/1 TB Ram	?

Note: This list is not complete and work in progress. Facts could change from release to release. It provides a very rough overview about some differences and major features.

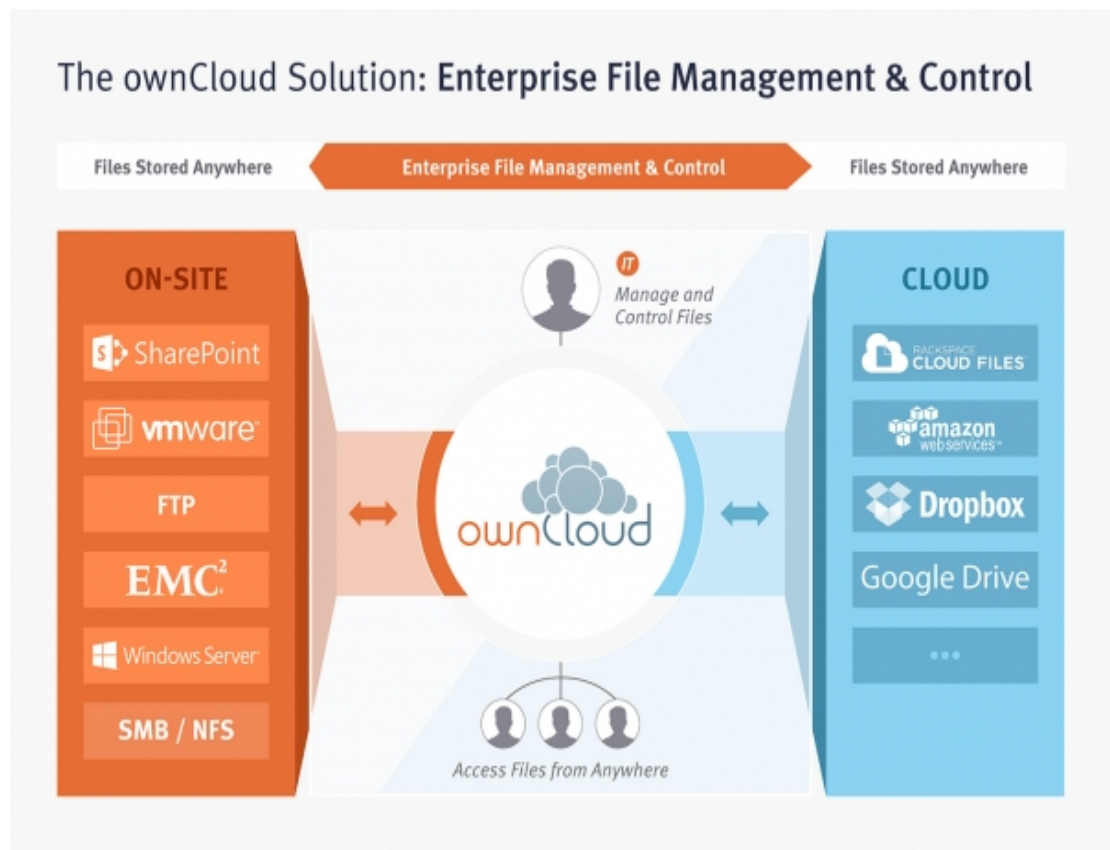
Figure 4. Virtualization Applications Comparison

It shown here that Proxmox VE supports the most important and necessary features such as High Availability, Live VM Snapshots, VM live migration, and the support of 160 CPU/2 TB of RAM which is equal to its biggest competitor VMware. In addition, the software challenges the other competitors for offering a simple subscription plan with a simple monthly price, support for a wide range of guest operating systems and most importantly supporting OpenVZ container known as Operating System Virtualization which is not supported by the rest of the competitors.

4 OWN CLOUD

ownCloud is an open-source solution to share and sync files. It is a great alternative to big cloud services which you have to trust with your data. ownCloud is highly suitable for people who want to have control on their data. It is very simple to use for a home user and is fully featured and simple to set up for an IT administrator. ownCloud can be deployed on a datacenter or it may be used to centralize all files from different services to provide a simple interface to the users where all their data is easily accessible. As an

open-source solution, ownCloud can be personalized to meet the user's needs and it is very easy to integrate with administrative or security services such as event logging, monitoring, Active Directory, SharePoint etc. ownCloud Server is the base of ownCloud which is a community driven editions and it comes in two versions suitable for standard and enterprise users. The base solution is perfectly suitable for a home user as the other versions are better in terms of support and come with a few enterprise advantages which are not necessary for a home user.



5 PRACTICAL PART

In this part, I explored and learned almost all parts of Proxmox VE and I explained some of the parts in theory shortly. The two important things done here are creating a virtual

machine using local storage, and also creating a container using Proxmox templates which download the operating systems directly to the interface.

5.1 Installing Proxmox on an External HDD

Download the Proxmox ISO file from:

<http://www.proxmox.com/en/downloads/category/iso-images-pve>

Use a virtualization software like VirtualBox or VMware and create a virtual machine that uses Proxmox ISO file as a live CD/DVD. Do not add any storage to your virtual machine. Instead, go to USB settings and add your portable storage so that your virtual machine is able to recognize it. Now, run the virtual machine and the installation should recognize your added portable device. Finish the installation and now you have Proxmox installed on your portable HDD. Remember to configure Proxmox IP address to be inside your local area network when installing Proxmox.

Network Configuration

Please verify the displayed network configuration. You will need a valid network configuration to access the configuration interface after installation.

Afterwards press the Next button to continue installation. The installer will then partition your hard disk and start copying packages.

- **IP address:** Set the IP address for the Proxmox Virtual Environment.
- **Netmask:** Set the netmask of your network.
- **Gateway:** IP address of your gateway or firewall.
- **DNS Server:** IP address of your DNS server.

Hostname (FQDN):	ali.anafcheh											
IP Address:	192	.	168	.	1	.	124					
Netmask:	255	.	255	.	255	.	0					
Gateway:	192	.	168	.	1	.	1					
DNS Server:	82	.	197	.	20	.	5					

When you boot from your external HDD, Proxmox might fail to find your root partition as your HDD might have a slow response so the OS will not boot. You can fix this by adding `rootdelay=5` the kernel parameters when booting the system. So, when you see the grub boot menu, press “e” to edit the boot commands. Find the line below and add `rootdelay=5` to the end. Then press “Ctrl+X” or “F10” to continue the boot:

```
linux      /boot/vmlinuz-4.2.6-1-pve root=/dev/mapper/pve-root ro quiet rootdelay=5
```

If you have done everything correctly and your drivers are supported by Proxmox, then it will boot successfully. You need to make the above changes in the grub file as well to make them permanent. So, edit /etc/default/grub using your favorite command line editor like nano:

```
# nano /etc/default/grub
```

Add rootdelay=5 as below:

```
GRUB_CMDLIN_LINUX_DEFAULT="quiet rootdelay=5"
```

5.2 Proxmox Configuration

In this section, we configure the basic settings to be able to setup our network. Navigate to the IP address of your Proxmox web configuration that you have assigned during the installation. It should be <https://youripaddress:8006>. Login with your credentials.

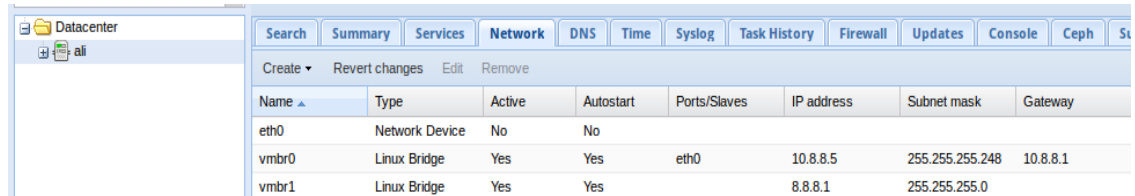
To run the newly created virtual machine, right click on it and choose “Run”. If you would look to view your virtual machine interface, you must right click again and choose “Console”. A pop up ran by HTML5 will appear viewing your virtual machine.

5.2.1 Network Configuration

Choose your node from the left side. Go to the “Network” tab. In this tab you can see your physical NIC at the top. In most cases it is “eth0”. Proxmox does not use your physical NIC directly instead it uses a bridge called “vmbr0” which is like a switch connected to your physical interface. So, if you have one NIC on your PC, this should be your default network settings. However, as we need to create an internal network for our virtual machines, we need to create a virtual NIC which is connected to no other port. So, choose “Create” from the menu bar and select “Linux Bridge”. Add an IP

address and a subnet mask and click “Create” . In other words, we just created a switch which is connected to nothing.

Proxmox NICs	Vmbr0(external network)	Internet	10.8.8.6/29
	Vmbr1(internal network)	No Internet	8.8.8.1/29



Name	Type	Active	Autostart	Ports/Slaves	IP address	Subnet mask	Gateway
eth0	Network Device	No	No				
vmbr0	Linux Bridge	Yes	Yes	eth0	10.8.8.5	255.255.255.248	10.8.8.1
vmbr1	Linux Bridge	Yes	Yes		8.8.8.1	255.255.255.0	

Vmbr0 IP address has to be in the same network as your physical local area network. Vmbr1 is the virtual NIC and may have any address you wish.

5.2.2 Storage Configuration

Download all the needed iso files from:

- <https://www.debian.org/distrib/>
- <https://www.pfsense.org/download/>
- <http://lubuntu.net/> or choose any other lightweight distro

Click on the “+” next to your node to expand it. Click on your storage “local”. Go to the “Content” tab. Choose upload from the menu bar and upload all the ISO files one by one.

5.3 Virtual Machines

Here are the settings that you should use when creating the virtual machines:

VMs	RAM	CPU	Storage	OS Type	NIC
1.Debian (ownCloud)	2GB	2 Cores	+50GB	Linux(I26)	1 vmbr1 Bridge - 1 NAT
2.PfSense	512GB	1 Core	5GB	Linux(I26)	1 vmbr0 Bridge – 1 vmbr1 Bridge
3.Lubuntu	2GB	2 Cores	5GB	Other	1 NAT

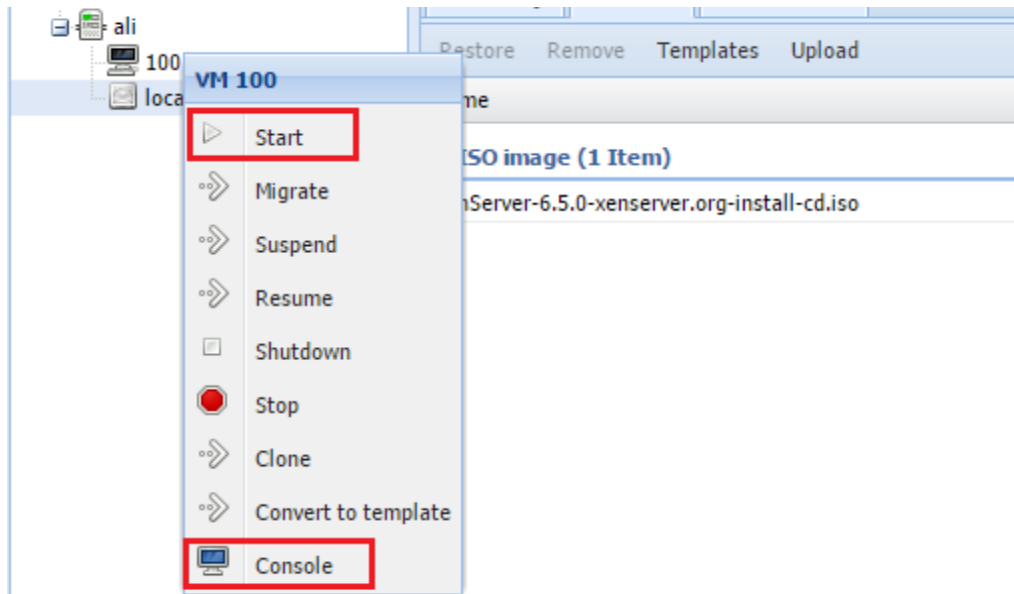
1. Assign as much HDD space as you have available to Debian, as this will be your ownCloud free space as well.
2. PfSense is very lightweight.
3. Lubuntu or any other distro of your choice is just used to access hosts on our internal network through a web browser. This virtual machine is just needed at the beginning and can be deleted or kept turned off when we create our OpenVPN server.

The NAT interfaces are just used to provide a quick Internet connection because we need to install some packages at the beginning. However, they may be deleted or disconnected after we have installed our packages. It is a good idea to keep NAT on Debian as you would need to keep it updated.

Start creating your virtual machines by clicking on “Create VM” on the top right corner of the page. Use the ISO file that you download in the CD/DVD tab. Fill in the rest of the configurations as above and keep the rest as default. Note that during the virtual machine creation, you can only create one NIC. You can add the second NIC by selecting your virtual machine listed under your node, then choose the “Hardware” tab. Choose “Add” from the menu bar and select “Network Device”. Also, always remember to go back to the hardware tab of each virtual machine and remove the ISO file from the CD/DVD so you don’t boot back into it after installation. You should have all your 3 virtual machines ready by now. Let’s start configuring them.

5.4 Virtual Machine 100 Debian 8 – Installing ownCloud 9

Run your Debian virtual machine by right clicking on it and selecting “Start” then “Console”. Follow the installation process of Debian.



When you run the virtual machine you might receive the error “No accelerator found” in the “Tasks” logs. This is caused if your CPU does not support KVM virtualization which in this case must be disabled. If you face this error, continue to “**Troubleshooting**”, otherwise skip it.

Troubleshooting: First, you can go to your Bios settings and check if there is an option to enable virtualization technology on your CPU. Otherwise, choose your virtual machine, then click on “Options” tab. In the list shown, double click on “KVM hardware virtualization”. In the pop up, untick the “Enabled” checkbox.

After installing Debian, first configure your bridge NIC to get a static IP address. You can recognize that NIC by running a command such as “ip link show” or “ifconfig” and comparing the MAC addresses. The NIC nickname in my case is “eth0”. Run the following command and make the changes as the picture. The IP address you specify here will be the IP address of ownCloud web configurator:

```
# nano /etc/network/interfaces
```

```
# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto eth0
allow-hotplug eth0
iface eth0 inet static
    address 8.8.8.2
    netmask 255.255.255.248
```

If you have added a secondary NAT interface you should have internet connection in your Debian virtual machine. Run the following commands to add the repository and install ownCloud, MariaDB and openSSL

```
# wget -nv https://download.owncloud.org/download/repositories/stable/Debian_8.0/Release.key -O Release.key apt-key add - < Release.key
```

```
# sh -c "echo 'deb http://download.owncloud.org/download/repositories/stable/Debian_8.0/ /' >> /etc/apt/sources.list.d/owncloud.list"
```

```
# apt-get update
```

```
# apt-get install owncloud mariadb-server openssl
```

5.4.1 Configuring OpenSSL

OwnCloud does not support HTTPS by default. That is why we have to configure it ourselves. First enable SSL by running these commands:

```
# a2enmod ssl
```

```
# a2enmod rewrite
```

Next, let's create a self-signed certificate using these commands. Fill in the certificate questions as you want:

```
# mkdir -p /etc/apache2/ssl
```

```
# openssl req -new -x509 -days 365 -nodes -out /etc/apache2/ssl/owncloud.pem -keyout
/etc/apache2/ssl/owncloud.key
```

```
root@debian-AliAnafcheh:~# openssl req -new -x509 -days 365 -nodes -out /etc/ap
che2/ssl/owncloud.pem -keyout /etc/apache2/ssl/owncloud.key
Generating a 2048 bit RSA private key
.....+++
.....+++
writing new private key to '/etc/apache2/ssl/owncloud.key'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:FI
State or Province Name (full name) [Some-State]:Eastern Finland
Locality Name (eg, city) []:Mikkeli
Organization Name (eg, company) [Internet Widgits Pty Ltd]:MAMK
Organizational Unit Name (eg, section) []:
Common Name (e.g. server FQDN or YOUR name) []:Ali Anafcheh
Email Address []:ali.anafcheh@edu.mamk.fi
```

Now, we have to configure ownCloud to use HTTPS. Run the following command:

```
# nano /etc/apache2/sites-enabled/ownCloud.conf
```

Type the code below in the new file above. Remember to replace the bold parts with your own information.

```
<VirtualHost 8.8.8.2:80>
```

```
##### Redirect to port 443 ###
```

```
RewriteEngine on
```

```
ReWriteCond %{SERVER_PORT} !^443$
```

```
RewriteRule ^/(.*) https://%{HTTP_HOST}/$1 [NC,R,L]
```

```
##### End of Redirection configuration #####
```

```
DocumentRoot /var/www/html/owncloud/
```

```
<Directory /var/www/html/owncloud>
```



```
Options Indexes FollowSymLinks MultiViews
```

```
AllowOverride All
```

```
Require all granted
```

```
</Directory>
```

```
</VirtualHost>
```

```
<VirtualHost 8.8.8.2:443>
```

```
#####Configuration for SSL#####
```

```
SSLEngine on
```

```
SSLCertificateFile /etc/apache2/ssl/owncloud.pem
```

```
SSLCertificateKeyFile /etc/apache2/ssl/owncloud.key
```

```
##### End of SSL Configuration #####
```

```
DocumentRoot /var/www/html/owncloud/
```

```
<Directory /var/www/html/owncloud>
```

```
Options Indexes FollowSymLinks MultiViews
```

```
AllowOverride All
```

```
Require all granted
```

```
</Directory>
```

```
</VirtualHost>
```

Restart Apache server and that is it for OpenSSL. If you faced an error when running the command below, most probably you mistyped something in the code above, so re-check it and everything should work fine:

```
# service apache2 restart
```

5.4.2 Configuring MariaDB

During the installation of ownCloud, you were asked to assign a root password to MySQL. Use it to login to MariaDB server when entering the command below:

```
# mysql -u root -p
```

Now, as a root user, create a user with all privileges specifically for ownCloud.

```
# CREATE DATABASE yourDatabaseName;
```

```
# CREATE USER 'databaseUsername'@'localhost' IDENTIFIED BY 'databaseUsernamePassword';
```

```
# GRANT ALL PRIVILEGES ON yourDatabaseName.* TO 'databaseUsername'@'localhost';
```

```
# exit
```

So, remember your database name and credentials as we will use them later when we set up ownCloud through the web interface.

5.5 Virtual Machine 101 PfSense

Run the virtual machine, wait for PfSense to boot in live mode:

Type “99” and press “Enter”. The installation will begin. Choose “Quick/Easy Install” and go with the defaults.

```

WAN (wan)      -> em0      ->
LAN (lan)      -> em1      -> v4: 192.168.1.1/24
0) Logout (SSH only)      9) pfTop
1) Assign Interfaces      10) Filter Logs
2) Set interface(s) IP address 11) Restart webConfigurator
3) Reset webConfigurator password 12) pfSense Developer Shell
4) Reset to factory defaults 13) Upgrade from console
5) Reboot system          14) Enable Secure Shell (sshd)
6) Halt system            15) Restore recent configuration
7) Ping host              16) Restart PHP-FPM
8) Shell

99) Install pfSense to a hard drive, etc.

Enter an option: 99

```

After installing it, make sure you remove the iso from the CD/DVD and restart pfsense by typing “7” and pressing “Enter”. You need to make sure you boot in the installed version, then continue to make the following configurations.

5.5.1 Assigning Interfaces

Note down the mac addresses of the two bridge NICs by clicking on the “Hardware” tab of PfSense virtual machine. Now, go back to the consol and start assigning the interfaces by pressing “1” then “Enter”. Configure the interfaces as below:

Vmbr0 Bridge

WAN

Vmbr1 Bridge

LAN

Do not set up VLANs or any optional interfaces. At the end type “y” to proceed and save the changes.

```

The interfaces will be assigned as follows:

WAN -> em0
LAN -> em1

Do you want to proceed [y/n]?

```

5.5.2 Set Interfaces IP addresses

Type in “2” then “Enter”. Choose WAN and assign a static IP address. This IP address must be an IP address which is in your physical local area network. So, this is the interface which is a bridge of you physical NIC somehow, and it is connected to the Internet. Here is my addressing scheme:

Physical Network	10.8.8.0/29(Internet)
-------------------------	-----------------------

Proxmox	10.8.8.6/29(Internet)
----------------	-----------------------

PfSense WAN	10.8.8.3/29(Internet)
--------------------	-----------------------

PfSense LAN	8.8.8.3/29(No Internet)
--------------------	-------------------------

```

WAN (wan)      -> em0      -> v4: 10.8.8.3/29
LAN (lan)      -> em1      -> v4: 8.8.8.3/29
0) Logout (SSH only)      9) pfTop
1) Assign Interfaces      10) Filter Logs
2) Set interface(s) IP address  11) Restart webConfigurator
3) Reset webConfigurator password  12) pfSense Developer Shell
4) Reset to factory defaults  13) Upgrade from console
5) Reboot system          14) Enable Secure Shell (sshd)
6) Halt system            15) Restore recent configuration
7) Ping host              16) Restart PHP-FPM
8) Shell

```

So, configure both interfaces, add IP address, Subnets and Gateways. The gateway for the WAN Interface is the main router of you physical LAN. The gateway for the LAN interface is the IP Address of vmbr1 (the virtual bridge).

The LAN interface IP address is the address to PfSense web configurator.

We are done with the backend configuration. Let's do the rest of the configuration through the web interfaces.

5.6 Virtual Machine 102 Lightweight Linux Distribution

As I mentioned before, I chose to use Lubuntu. Running any distribution in live mode is enough for this setup. One NAT will give us access both to the internet and the internal network as well. As you login, you should get an IP address automatically. It's time to test what you have done so far. Make sure the other two virtual machines are running. Run the browser and type in the vmbr1 bridge IP address of each virtual machine like the structure below:

ownCloud

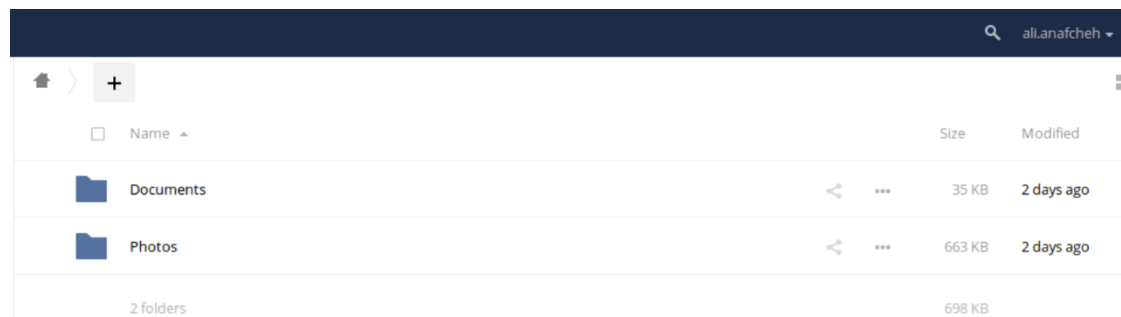
<https://8.8.8.2/owncloud>

PfSense

<https://8.8.8.3>

5.7 ownCloud Web Interface

The first thing you see is the final step of setting up ownCloud. Write a “Username” and a “Password”. This will be as your admin account. Make sure you choose a strong password, otherwise you will receive a database error. Choose MySQL/MariaDB as your database server and fill in the info as we have set them before when [Configuring MariaDB](#). Click on “Finish” and there is ownCloud. Full of features and simple to use. Make sure to check its settings and preferences to personalize it.



Now, the purpose of this project is to make this available from the internet through an OpenVPN connection. Let's move on to that.

5.8 PfSense Web Interface

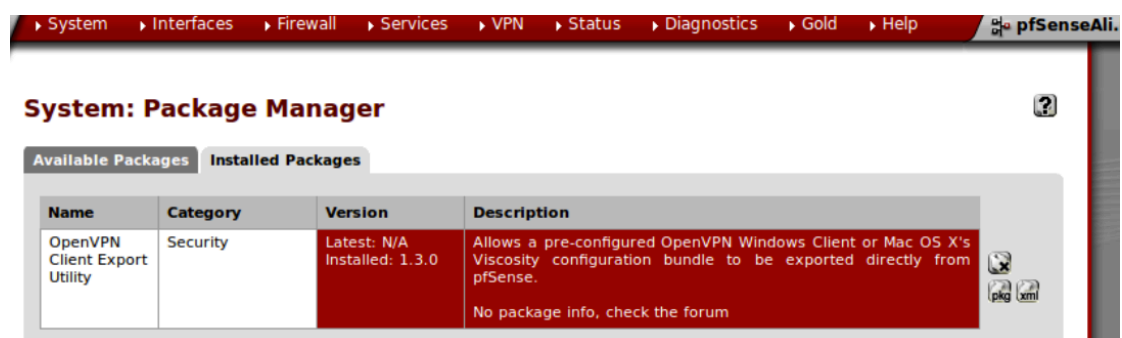
The default credentials to access PfSense are as below, remember to change them:

username	admin
----------	-------

password	pfsense
----------	---------

Complete the initial setup. When configuring your LAN and WAN interface, make sure to uncheck the options to “Block private networks” and “Block bogon networks”. This is because our firewall is placed in a network in which the WAN IP address has a private address. Also, we might use bogon addresses in our setup as I will use one for my OpenVPN tunnel.

Now, we need a package to export our OpenVPN configuration files for different devices. Navigate to System>Packages. Make sure you are on the “Available Packages” tab. Search for “OpenVPN Client Export Utility”. Click on the “+” button next to the package then click on “Confirm” in the next page. Remember you need internet connection for this step.



5.8.1 OpenVPN Server

Navigate to VPN>OpenVPN>Wizards. Choose “Local User Access” for the type of server and click “Next”.

Create your new “Certificate Authority” which will be used to authorize other certificates such as user certificates. Fill in the information as you wish and click “Create new Certificate”.

Next create the OpenVPN “Server Certificate”. Fill in the information as you wish and click “Create New Certificate”

We need to fill in the information for our OpenVPN server now. Fill in all that is stated below and leave the rest to defaults. Remember that these are my preferences and you are free to choose different settings based on your own needs:

General OpenVPN Server Information	
Interface:	<div>WAN</div> <div>The interface where OpenVPN will listen for incoming connections (typically WAN.)</div>
Protocol:	<div>UDP</div> <div>Protocol to use for OpenVPN connections. If you are unsure, leave this set to UDP.</div>
Local Port:	<div>1194</div> <div>Local port upon which OpenVPN will listen for connections. The default port is 1194. This can be left at its default unless you need to use a different port.</div>
Description:	<div>Ali-OpenVPN-ownCloud</div> <div>A name for this OpenVPN instance, for your reference. It can be set however you like, but is often used to distinguish the purpose of the service (e.g. "Remote Technical Staff"). It is also used by OpenVPN Client Export to identify this VPN on clients.</div>

Cryptographic Settings	
TLS Authentication:	<input checked="" type="checkbox"/> Enable authentication of TLS packets.
Generate TLS Key:	<input checked="" type="checkbox"/> Automatically generate a shared TLS authentication key.
TLS Shared Key:	<div style="border: 1px solid #ccc; height: 40px; width: 100%;"></div> <p>Paste in a shared TLS key if one has already been generated.</p>
DH Parameters Length:	<div>2048 bit</div> <p>Length of Diffie-Hellman (DH) key exchange parameters, used for establishing a secure communications channel. As with other such settings, the larger values are more secure, but may be slower in operation.</p>
Encryption Algorithm:	<div>AES-256-CBC (256-bit)</div> <p>The algorithm used to encrypt traffic between endpoints. This setting must match on the client and server side, but is otherwise set however you like. Certain algorithms will perform better on different hardware, depending on the availability of supported VPN accelerator chips.</p>
Auth Digest Algorithm:	<div>SHA1 (160-bit)</div> <p>The method used to authenticate traffic between endpoints. This setting must match on the client and server side, but is otherwise set however you like.</p>
Hardware Crypto:	<div>No Hardware Crypto Acceleration</div> <p>The hardware cryptographic accelerator to use for this VPN connection, if any.</p>

The “Tunnel Network” option below can be any network of your choice. When connected to this OpenVPN Server you will receive an IP address in this range. Also we need to add our virtual machines internal network IP address in the “Local Network” text box as shown below.

Tunnel Settings	
Tunnel Network:	<div>4.4.4.0/25</div> <p>This is the virtual network used for private communications between this server and client hosts expressed using CIDR notation (eg. 10.0.8.0/24). The first network address will be assigned to the server virtual interface. The remaining network addresses can optionally be assigned to connecting clients. (see Address Pool)</p>
Redirect Gateway:	<input checked="" type="checkbox"/> Force all client generated traffic through the tunnel.
Local Network:	<div>8.8.8.0/29</div> <p>This is the network that will be accessible from the remote endpoint, expressed as a CIDR range. You may leave this blank if you don't want to add a route to the local network through this tunnel on the remote machine. This is generally set to your LAN network.</p>
Concurrent Connections:	<div>5</div> <p>Specify the maximum number of clients allowed to concurrently connect to this server.</p>
Compression:	<div>No Preference</div> <p>Compress tunnel packets using the LZO algorithm. Adaptive compression will dynamically disable compression for a period of time if OpenVPN detects that the data in the packets is not being compressed efficiently.</p>
Type-of-Service:	<input type="checkbox"/> Set the TOS IP header value of tunnel packets to match the encapsulated packet's TOS value.
Inter-Client Communication:	<input type="checkbox"/> Allow communication between clients connected to this server.
Duplicate Connections:	<input type="checkbox"/> Allow multiple concurrent connections from clients using the same Common Name. NOTE: This is not generally recommended, but may be needed for some scenarios.

Client Settings	
Dynamic IP:	<input checked="" type="checkbox"/> Allow connected clients to retain their connections if their IP address changes.
Address Pool:	<input checked="" type="checkbox"/> Provide a virtual adapter IP address to clients (see Tunnel Network).

Click next and check both the firewall and OpenVPN rules in the next page. Finish the setup.

Let's export our configurations. Go to VPN>OpenVPN>Client Export.

In "Host Name Resolution", choose "Other" and then enter your public IP address that was given to you by your ISP. You can check it by checking your main router's interface.

Server **Client** **Client Specific Overrides** **Wizards** **Client Export** **Shared Key Export**

Remote Access Server Ali-OpenVPN-ownCloud UDP:1194

Host Name Resolution Other
 your public IP Address Enter the hostname or IP address the client will use to connect to this server.

Verify Server CN Automatic - Use verify-x509-name (OpenVPN 2.3+) where possible
 Optionally verify the server certificate Common Name (CN) when the client connects. Current clients, including the most recent versions of Windows, Viscosity, Tunnelblick, OpenVPN on iOS and Android and so on should all work at the default automatic setting.
 Only use tls-remote if you must use an older client that you cannot control. The option has been deprecated by OpenVPN and will be removed in the next major version.
 With tls-remote the server CN may optionally be enclosed in quotes. This can help if the server CN contains spaces and certain clients cannot parse the server CN. Some clients have problems parsing the CN with quotes. Use only as needed.

Use Random Local Port ☒ Use a random local source port (lport) for traffic from the client. Without this set, two clients may not run concurrently.
 NOTE: Not supported on older clients. Automatically disabled for Yealink and Snom configurations.

Certificate Export Options
☐ Use Microsoft Certificate Storage instead of local files.
☒ Use a password to protect the pkcs12 file contents or key in Viscosity bundle.
 Password :
 Confirm :

After you have done the configurations above, scroll down and choose the configuration suitable for you:

Client Install Packages		
User	Certificate Name	Export
ali.ovpn	ali=ovpn-user	- Standard Configurations: Archive Config Only - Inline Configurations: Android OpenVPN Connect (iOS/Android) Others - Windows Installers (2.3.8-lx01): x86-xp x64-xp x86-win6 x64-win6 - Viscosity (Mac OS X and Windows): Viscosity Bundle Viscosity Inline Config

5.9 Port Forwarding

We have to forward our traffic when an OpenVPN connection is requested from our public IP address. As there are two interfaces in between our firewall's LAN interface then we need to set two port forwarding rules. One will be in the main home router to redirect traffic to Pfsense's WAN Interface. The other will be in Pfsense to redirect the OpenVPN traffic from any source to our LAN interface which is 8.8.8.3.


On Pfsense, navigate to Firewall>NAT>Port Forwarding. Click on the “+” button to add a new rule. As you can see below, my “redirect target IP” is my LAN interface's IP address.

Edit Redirect entry	
Disabled	<input type="checkbox"/> Disable this rule Set this option to disable this rule without removing it from the list.
No RDR (NOT)	<input type="checkbox"/> Enabling this option will disable redirection for traffic matching this rule. Hint: this option is rarely needed, don't use this unless you know what you're doing.
Interface	WAN Choose which interface this rule applies to. Hint: in most cases, you'll want to use WAN here.
Protocol	UDP Choose which IP protocol this rule should match. Hint: in most cases, you should specify TCP here.
Source	Advanced - Show source address and port range
Destination	<input type="checkbox"/> not Use this option to invert the sense of the match. Type: WAN address Address: / 31
Destination port range	from: OpenVPN to: OpenVPN Specify the port or port range for the destination of the packet for this mapping. Hint: you can leave the 'to' field empty if you only want to map a single port
Redirect target IP	8.8.8.3 Enter the internal IP address of the server on which you want to map the ports. e.g. 192.168.1.12
Redirect target port	OpenVPN

Click on Save. We are all set-up in pfsense.

Login to your main home router now. Find the port forwarding tab and make the changes as below. Again as you can see I assigned my WAN IP address as the internal IP address.

New port forward:						
Name	Protocol	External zone	External port	Internal zone	Internal IP address	Internal port
All-OpenVPN	UDP	wan	1194	lan	10.8.8.3	1194

 Add

Add the rule and now you are all ready to connect through any OpenVPN client and enjoy ownCloud or any other server that you have in your internal Network.

6 AUTOMATION

Last but not least is to delete or disable any other thing that you do not need on Proxmox such as the third virtual machine. Now, let's make our necessary virtual machines auto boot when our Proxmox server restarts. Click on each virtual machine, go to the "Options" tab. Click on the "Start at boot option" and check it. Now you have a server that you would rarely need to check. Everything auto starts and you can just keep your physical server somewhere out of sight as you can control everything through Proxmox. It is also possible to connect your server to WIFI through the command line so you would not even need to connect it to a cable.

7 CONCLUSIONS

I have learned a lot during researching and implementing this project. It was very interesting to explore different features each software specially PfSense. The aim of my study was to combine open-source user-friendly solutions to create a private secure cloud storage home server.

BIBLIOGRAPHY

- [1] Promox, Promox VE Wiki, 2011, web page,
https://pve.proxmox.com/wiki/Main_Page Referred 13.02.2015.

- [2] RedHat, What is Virtualization? PDF Document. http://www.redhat.com/f/pdf/virtualization/gunner_virtual_paper2.pdf. Referred 06.05.2015.

- [3] VMware, Virtualization Basics. Web Page. <http://www.vmware.com/virtualization/virtualization-basics/what-is-virtualization>. Referred 06.05.2015.

- [4] CIO, Six Reasons Small Businesses Need Virtualization, 2012. Web Page. <http://www.cio.com/article/2400612/virtualization/six-reasons-small-businesses-need-virtualization.html> Referred 06.05.2015.

- [5] InfoWorld, Top 10 benefits of server virtualization, 2011. Web Page. <http://www.infoworld.com/article/2621446/server-virtualization/server-virtualization-top-10-benefits-of-server-virtualization.html> Referred 06.05.2015.

- [6] Small Business Computing, What is Virtualization, and Why Should You Care? 2009. Web Page. <http://www.smallbusinesscomputing.com/testdrive/article.php/3819231/What-is-Virtualization-and-Why-Should-You-Care.htm> Referred 06.05.2015.

- [7] How Stuff Works, How Server Virtualization Works. Web Page. <http://computer.howstuffworks.com/server-virtualization1.htm> Referred 06.05.2015.

- [8] Techtarget, Virtualization, 2010. Web Page. <http://searchservvirtualization.techtarget.com/definition/virtualization>. Referred 06.05.2015.

- [9] Altaro, Hyper-V Terminology – Host Operating System or Parent Partition? 2015. Web Page. <http://www.altaro.com/hyper-v/hyper-v-terminology-host-operating-system-or-parent-partition/>. Referred 06.05.2015.

- [10] Spanningtree, Remote desktop control services to access your office from half way across the world. Web Page. <http://www.spanningtree.ca/virtualization-and-consolidation.php>.
- [11] Chubbable, Step-by-step Guide On How To Setup OpenVPN From pfSense's Web-GUI. <https://chubbable.com/setup-openvpn-pfsense>.
- [12] NoobsLAB, Install ownCloud Version 8.1.13 In Ubuntu/Linux Mint Via PPA And Easily Configure ownCloud (SSL Encryption). Web Page. <http://www.noobslab.com/2015/01/install-owncloud-in-ubuntulinux-mint.html>.
- [13] Xmodulo, How to install and configure ownCloud on Debian. Web Page. <http://xmodulo.com/install-configure-owncloud-debian.html>.
- [14] PfSense, PfSenseDocs. Web Page. https://doc.pfsense.org/index.php/Main_Page.
- [15] ownCloud, ownCloud. Web Page. <https://owncloud.org/>