Proxmox

- Proxmox-Exposed-Host
- Proxmox Fixes and Workarounds
- The special case with a VPS
- Monitoring
- LXC set mountpoint

Proxmox-Exposed-Host

In This Post I'm showing you How to create a Proxmox host which is reachable trough internet. It presupposes you have Debian already installed on your server:

Access and Update the Server

Add User

```
adduser yourusername
```

install sudo

```
apt-get install sudo
```

Add new user to sudo Group

```
sudo adduser mynewuser sudo
```

Create and copy your SSH Key

Creating SSH-key

Connect with SSH Key

```
ssh yourusername@ip-address
```

Upgrade Server

```
apt-get update && apt-get dist-upgrade -y
```

Harden SSH

Install UFW

```
apt-get install ufw
```

Allow Port 22 (SSH Port) with Protocol TCP

```
ufw allow 22/tcp
```

activate UFW

```
ufw enable
```

edit SSH Config File

```
nano /etc/ssh/sshd_config

    Now edit / instert the following

PermitRootLogin no
MaxAuthTries 6
AllowUsers yourusername
PasswordAuthentication no
PermitEmptyPasswords no
PubkeyAuthentication yes
```

Reload SSH

```
systemctl restart sshd
```

Convert your Debian 10 Server to Proxmox 6

Add an `/etc/hosts` entry for your IP address

- Note: Make sure that no IPv6 address for your hostname is specified in /etc/hosts
- For instance, if your IP address is 192.168.15.77, and your hostname prox4m1, then your /etc/hosts file should look like:

`nano /etc/hosts`

```
127.0.0.1       localhost.localdomain localhost
  192.168.15.77   prox4m1.proxmox.com prox4m1

 # The following lines are desirable for IPv6 capable hosts
 ::1     localhost ip6-localhost ip6-loopback
 ff02::1 ip6-allnodes
 ff02::2 ip6-allrouters
```

```
You can test if your setup is ok using the hostname command:
```

```
hostname --ip-address
```

```
192.168.15.77 # should return your IP address here
```

Adapt your sources.list

```
Add the Proxmox VE repository:
```

```
echo "deb http://download.proxmox.com/debian/pve buster pve-no-subscription" >
/etc/apt/sources.list.d/pve-install-repo.list
```

Add the Proxmox VE repository key

```
wget http://download.proxmox.com/debian/proxmox-ve-release-6.x.gpg -O
/etc/apt/trusted.gpg.d/proxmox-ve-release-6.x.gpg
chmod +r /etc/apt/trusted.gpg.d/proxmox-ve-release-6.x.gpg  # optional, if you have a non-
default umask
```

Update your repository and system by running

```
apt update && apt full-upgrade
```

Install the Proxmox VE packages

```
apt install proxmox-ve postfix open-iscsi
```

Recommended: remove the os-prober package

- The os-prober package scans all the partitions of your host, including those assigned to guests VMs, to create dual-boot GRUB entries. If you didn't install Proxmox VE as dual boot beside another Operating System, you can safely remove the os-prober package.

```
apt remove os-prober
```

Update and check grub2 config by running:

```
update-grub
```

Now Reboot

```
reboot
```

Enter Proxmox Management UI

Allow the Proxmox management Port (8006) to be open

```
ufw allow 8006/tcp
```

Reload UFW

```
ufw reload
```

After that your Management Web Interface should be reachable in your Browser under https://your-ip-address:8006/

Note: we won't expose the Control Interface for very long

Configure Proxmox

Edit the file /etc/network/interfaces

Paste the following (if your Main Interface is eth0)

```
auto vmbr1
iface vmbr1 inet static
        address  10.10.10.254
        netmask  255.255.255.0
        bridge-ports none
        bridge-stp off
        bridge-fd 0

        # OpenDNS - Nameservers
        dns-nameservers 208.67.222.222 208.67.220.220

        post-up echo 1 > /proc/sys/net/ipv4/ip_forward

        post-up iptables -t nat -A POSTROUTING -s '10.10.10.0/24' -o eth0 -j MASQUERADE
        post-down iptables -t nat -D POSTROUTING -s '10.10.10.0/24' -o eth0 -j MASQUERADE

       # Like this, you can Portforward external Ports to internal TCP / UDP Ports from LXC
Container
```

```
       iptables -t nat -A PREROUTING -p tcp -i vmbr0 --dport 8080 -j DNAT --to-destination
10.10.10.9:8080
```

Note: that I moved the Part *post-up echo 1 > /proc/sys/net/ipv4/ip_forward* now from the Hardware Interface to the newly created Linux Bridge (vmbr1) Note: repace eth0 for your real ethernet Interface Now Reboot

```
reboot
```

(Optional but recommendet) Make Admin Portal accessable only via VPN Connection or your Static IP:

Use / download Openvpn script: https://github.com/angristan/openvpn-install

```
git clone https://github.com/angristan/openvpn-install
```

Change directory to Openvpn script

```
cd openvpn-install/
```

Make script executable

```
chmod +x openvpn-install.sh
```

run Openvpn script

```
./openvpn-install.sh
```

Allow SSH traffic from your OpenVPN connection

```
ufw allow from  10.8.0.0/24  to any port 22
```

Allow SSH traffic from your Static IP Address (if you have one at home or use another VPS)

```
ufw allow from  *staticip*  to any port 22
```

Change loglevel of your UFW so that the logfiles don't get gigantic

```
ufw logging low
```

Edit /etc/default/ufw

```
nano /etc/default/ufw
```

Allow troughput trough your VPN Connection and avoid getting no internet connection when you are connected with your VPN by pasting the following

```
DEFAULT_FORWARD_POLICY="ACCEPT"
```

Allow Traffic to OpenVPN Port 1194

```
ufw allow 1194
```

Note: Depending if you choose UDP or TCP while installing the Openvpn Script you may want to use: 'ufw allow 1194/udp' or 'ufw allow 1194/tcp' reload ufw

```
ufw reload
```

test Admin Portal Connection via https://10.10.10.254:8006

```
sudo openvpn /path/to/openvpn.file
```

and then simply point your Browser to: https://10.10.10.254:8006 if >>EVERYTHING<< works, continue with 13. remove firewall rule to allow connection to port 8006/tcp

```
ufw delete allow 8006/tcp
```

reload ufw

```
ufw reload
```

The Only way to connect now to your servers Admin Panel is either via your (if you have one) static IP or trough your VPN connection.

Fix Locales Error

Copy paste the Commands, I also just googled them, and I'm not exactly sure what the Commands are exactly doing, besides, fixing the locales...

```
export LANGUAGE=en_US.UTF-8
export LANG=en_US.UTF-8
export LC_ALL=en_US.UTF-8
locale-gen en_US.UTF-8
dpkg-reconfigure locales
```

No Subscription Repo

Now we are pasting the right (no-subscription) Proxmox Apt-Repository. Since we don't have a Subscription and we don't want one (most of the time...) First we remove the file `/etc/apt/sources.list.d/pve-enterprise.list`

```
rm /etc/apt/sources.list.d/pve-enterprise.list
```

Create a new file named pve-no-subscription.list via nano:

```
nano /etc/apt/sources.list.d/pve-no-subscription.list
```

there we paste simply the following, which has no deeper meaning, besides, it's the Proxmox no subscription Repository

```
deb http://download.proxmox.com/debian/pve buster pve-no-subscription
```

test if your repositories are correctly set up with updating your Server:

```
apt-get update
apt-get dist-upgrade
```

if there are no error messages, your repositories are correctly setup

Create a Template

The special case with a VPS

Container

in most cases a VPS has only one virtual drive attached, what makes it impossible (if the VPS uses LVM) for Proxmox to create a template, since the template needs to be on another Storage (correct me, if it changed in meantime). So what you do instead is download a LXC Template from the GUI, assign it the last possible IP you have and costumize it. This has several advantages:

the first Container has the id 0, if it's your template, the first Container can be assigned with your IP X.X.X.1 you can simply clone your fist Container via GUI even tough it's no "real" Template

Note: This is more or less a workaround, since if you have f.e. ZFS as storage, you CAN create templates. Netherless, it is good practice to use your first created container / VM as template, since it's easier, to assign your IP addresses in order.

Create a reverse Proxy

Install a webserver

in this case we are using a Nginx webserver

```
apt-get install nginx
```

Configure nginx

for Nginx configuration I am linking a sample Nginx configuration creator:

https://nginxconfig.io/

test Nginx configuration for mistakes

```
nginx -t
```

restart Nginx

```
systemctl restart nginx
```

... enjoy your nginx reverse proxy

Proxmox Fixes and Workarounds

XMPP Letsencrypt Container

Create Certificate Folder

```
mkdir /var/www/ssl/xmpp
```

get Letsencrypt Certificate

```
certbot certonly --webroot -w /var/www/ssl/xmpp --email mail@domain.tld -d xmpp.domain.tld -d
conference.domain.tld -d pubsub.domain.tld -d upload.domain.tld -d domain.tld
```

Set Mountpoint from Host to Container

Assuming your XMPP Container ID is 100:

```
pct set 100 -mp0 /etc/letsencrypt/live,mp=/etc/letsencrypt/live
pct set 100 -mp1 /etc/letsencrypt/archive,mp=/etc/letsencrypt/archive
```

Fix Locales Debian 10 LXC

```
Enter the following Commands:



export LANGUAGE=en_US.UTF-8
export LANG=en_US.UTF-8
```

```
export LC_ALL=en_US.UTF-8
locale-gen en_US.UTF-8
dpkg-reconfigure locales
```

Docker under LXC

Use unprivileged container in Options set keyctl=1, nested=1

Execute on Host:

```
systemctl edit containerd.service
```

Paste the following:

```
[Service] ExecStartPre=
```

.img to proxmox image (.qcow2)

move disk as qcow2 to external Storage sshfs to storage

```
qemu-img convert -f raw -O qcow2 image.img vm-104-disk.qcow2
```

- remove old qcow2 image
- remane new qcow2 to old imagename
- move disk to local storage

snmpd Monitoring fix

```
# This file controls the activity of snmpd and snmptrapd

# MIB directories. /usr/share/snmp/mibs is the default, but
# including it here avoids some strange problems.
export MIBDIRS=/usr/share/snmp/mibs

# snmpd control (yes means start daemon).
SNMPDRUN=yes
```

```
# snmpd options (use syslog, close stdin/out/err).
SNMPDOPTS='-Lsd -Lf /dev/null -u snmp -g snmp -I -smux -p /var/run/snmpd.pid'

# snmptrapd control (yes means start daemon). As of net-snmp version
# 5.0, master agentx support must be enabled in snmpd before snmptrapd
# can be run. See snmpd.conf(5) for how to do this.
TRAPDRUN=no

# snmptrapd options (use syslog).
TRAPDOPTS='-Lsd -p /var/run/snmptrapd.pid'

# create symlink on Debian legacy location to official RFC path
SNMPDCOMPAT=yes
```

The special case with a VPS Container

in most cases a VPS has only one virtual drive attached, what makes it impossible (if the VPS uses LVM) for Proxmox to create a template, since the template needs to be on another Storage (correct me, if it changed in meantime). So what you do instead is download a LXC Template from the GUI, assign it the last possible IP you have and costumize it. This has several advantages:

the first Container has the id 0, if it's your template, the first Container can be assigned with your IP X.X.X.1 you can simply clone your fist Container via GUI even tough it's no "real" Template

Note: This is more or less a workaround, since if you have f.e. ZFS as storage, you CAN create templates. Netherless, it is good practice to use your first created container / VM as template, since it's easier, to assign your IP addresses in order.

Monitoring

Remember to do Backups!!!

install smpd

```
apt-get install snmpd libsnmp-dev
```

add v3 user

```
net-snmp-config --create-snmpv3-user -ro -A authpass -X privpass -a SHA -x AES username
```

add clientserver to observium

```
cd /opt/observium
./add_device.php <ipaddress> ap v3 username authpass privpass sha aes 161 udp
```

remember to open ufw

Proxmox fix

```
# This file controls the activity of snmpd and snmptrapd

# MIB directories. /usr/share/snmp/mibs is the default, but
# including it here avoids some strange problems.
export MIBDIRS=/usr/share/snmp/mibs

# snmpd control (yes means start daemon).
SNMPDRUN=yes

# snmpd options (use syslog, close stdin/out/err).
SNMPDOPTS='-Lsd -Lf /dev/null -u snmp -g snmp -I -smux -p /var/run/snmpd.pid'
```

```
# snmptrapd control (yes means start daemon). As of net-snmp version
# 5.0, master agentx support must be enabled in snmpd before snmptrapd
# can be run. See snmpd.conf(5) for how to do this.
TRAPDRUN=no

# snmptrapd options (use syslog).
TRAPDOPTS='-Lsd -p /var/run/snmptrapd.pid'

# create symlink on Debian legacy location to official RFC path
SNMPDCOMPAT=yes
```

LXC set mountpoint

pct set 103 -mp0 /host/dir,mp=/container/mount/point