

# Unifi Protect im Docker Container in einer amd64 Debian VM installieren mit qemu arm64

## Beschreibung:

Unifi UNVR im Docker Container auf einem x86/x864 Host. Wir können einmal das Image aus dem Docker Hub Pullen oder uns auch ein eigenes Image bauen. Erstmal die Variante mit dem Pullen.

## Vorraussetzung:

Einmal ein x86/x64 Host in dem Docker installiert ist.

Eine zweite HDD mindesten 128 GB groß. würde aber eine Größe wählen die größer ist. Unbedingt eine Partioion und ext4 so kann sie on the fly vergrößert werden.

## Grundinstallation Docker und Einbungung Festplatte

Installation Docker, als root per ssh einloggen

```
apt install docker.io gdisk curl
#Installieren der quemu erweiterungen für arm64
apt-get install qemu qemu-system-arm binfmt-support qemu-user-static # Install the qemu packages
docker run --rm --privileged multiarch/qemu-user-static --reset -p yes # This step will execute the registering
scripts
```

Nun eine weitere Festplatte hinzufügen und diese Partitionieren wenn nicht schon geschehen  
Mit lsblk die neue Festplatte ermitteln

```
oot@unnvr:~# lsblk
NAME MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sda   8:0  0  32G  0 disk
├─sda1 8:1  0  512M  0 part /boot/efi
├─sda2 8:2  0 30.5G  0 part /
└─sda3 8:3  0  976M  0 part [SWAP]
sdb   8:16  0 256G  0 disk
sr0   11:0  1 329.3M  0 rom
```

sdb ist die gewünschte Festplatte

nun partitionieren

```
gdisk /dev/sdb

dann o eingeben für neues layout
dann n eingeben für eine neue partition

alles durchentern

nun w für schreiben
und exit
```

Nun Dateisystem ext4 erzeugen auf sdb1

```
mkfs.ext4 /dev/sdb1
```

Nun ein verzeichnis für den mount point erstellen

```
mkdir /unvr
```

Nun die part UUID auslesen

```
blkid /dev/sdb1
/dev/sdb1: UUID="a396f93a-305e-4412-9040-5c51c9203f78" BLOCK_SIZE="4096" TYPE="ext4"
PARTLABEL="Linux filesystem"
PARTUUID="c9c72c32-250f-4b08-b74f-0aadc682d313"
```

Nun haben wir die UUID. Mit dieser erstellen wir einen automount in der fstab.

```
nano /etc/fstab
```

Dort tragen wir unsere UUID ein dann das Mountverzeichnis das Dateisystem

```
...
UUID=a396f93a-305e-4412-9040-5c51c9203f78 /unvr ext4 defaults,errors=remount-ro 0 1
...
```

testen des mount points.

Wenns geklappt hat können wir mit

```
mount /unvr
```

die Festplatte mounten.

Das Ergebnis ob gemountet sehen wir durch lsblk

```
lsblk
NAME MAJ:MIN RM SIZE RO TYPE MOUNTPOINT
sda   8:0  0  32G  0 disk
├─sda1 8:1  0  512M  0 part /boot/efi
├─sda2 8:2  0 30.5G  0 part /
└─sda3 8:3  0  976M  0 part [SWAP]
sdb   8:16  0 256G  0 disk
└─sdb1 8:17  0 256G  0 part /unvr
sr0   11:0  1 329.3M  0 rom
```

Die Grub config um diesen Eintrag erweitern.

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

Inhalt:

```
GRUB_CMDLINE_LINUX_DEFAULT="quiet systemd.unified_cgroup_hierarchy=0"
```

Sollten schon andere parameter drin stehen diese logischerweise hinzufügen und nicht ersetzen...

nun update grub ausführen

```
update-grub
```

Neustarten. Achtung bei einem neustart könnte die sda / sdb reihenfolge anders sein.

```
reboot
```

Netzwerkkarten Namen anpassen sonst kann man sich nicht mit einem Cloud Knto von Unifi anmelden. Der Anme muss enp0s2 lauten

```
nano /etc/systemd/network/99-enp0s2.link
```

Die MACAddress mit der MAC Adresse unserer Netzwerkkarte austauschen.  
Diese bekommt man mit

```
ip a
```

raus

Inhalt:

```
[Match]
MACAddress=xx:xx:xx:xx:xx:xx

[Link]
Name=enp0s2

[Network]
DHCP=yes
```

nun

```
update-initramfs -u
```

ausführen und neustarten.

## Erstellen der Docker instanz durch pullen von dockerhub

Folgendes script laufen lassen, die Pfade sind angepasst auf /unvr  
Soll der conateiner automatisch gestartet werden an den Befehl noch

```
--restart always hinter -d  
dranhängen
```

Hiernochmals darauf achten das bei STORAGE\_DISK=/dev/sdb1 oder sdb2 die richtige partion angegeben wird. Denn nach dem neustart können diese sich ändern

```
docker run -d --name unifi-protect \  
  --privileged \  
  --tmpfs /run \  
  --tmpfs /run/lock \  
  --tmpfs /tmp \  
  -v /sys/fs/cgroup:/sys/fs/cgroup:ro \  
  -v /unvr/srv:/srv \  
  -v /unvr/data:/data \  
  -v /unvr/persistent:/persistent \  
  --network host \  
  -e STORAGE_DISK=/dev/sda1 \  
  -p 443:443 \  
  -p 80:80 \  
  -p 7446:7446 \  
  -p 7443:7443 \  
  -p 8443:8443 \  
  snowsnot/unifi-unvr:latest
```

## Mit Autostart Flag

```
docker run -d --restart always --name unifi-protect \  
  --privileged \  
  --tmpfs /run \  
  --tmpfs /run/lock \  
  --tmpfs /tmp \  
  -v /sys/fs/cgroup:/sys/fs/cgroup:ro \  
  -v /unvr/srv:/srv \  
  -v /unvr/data:/data \  
  -v /unvr/persistent:/persistent \  
  --network host \  
  -e STORAGE_DISK=/dev/sda1 \  
  -p 443:443 \  
  -p 80:80 \  
  -p 7446:7446 \  
  snowsnot/unifi-unvr:latest
```

```
-p 7443:7443 \  
-p 8443:8443 \  
snowsnot/unifi-unvr:latest
```

Nun 9-15 Minuten warten

mit folgendem Befehl kann man sich in den Container einloggen und zum beispiel den service status sich anzuschauen.

```
docker exec -it unifi-protect bash
```

Nun

```
service unifi-protect status
```

oder

```
journalctl -u unifi-protect
```

Nach 9-15 Minuten sollte die Website aufrufbar sein

Dann unter <https://ip:443> aufrufbar



## UI is committed to protecting your privacy and security

Our goal is to keep as much of your data off-cloud. The diagnostics data we do collect is used to improve performance and stability of the products we deliver to you - keeping your personal information private from everyone - including us. You have the option to opt out of sharing diagnostics data from your settings page after setup.



Setup UNVR

Nun sind wir fertig.

Sollte Unifi Protect noch ewig beim Status starten stehen.  
Dann einloggen mittels

```
docker exec -it unifi-protect /bin/bash
```

Nun den unifi-core neustarten

```
systemctl restart unifi-core
```

# Fehler:

## Failed to create /init.scope control group:

Wenn wir folgende Fehler bekommenr (or any systemd error):

```
Failed to create /init.scope control group: Read-only file system
Failed to allocate manager object: Read-only file system
[!!!!!!] Failed to allocate manager object.
Exiting PID 1...
```

Die Grub config um diesen Eintrag erweitern.

```
/etc/default/grub
```

Inhalt:

```
GRUB_CMDLINE_LINUX_DEFAULT="quiet systemd.unified_cgroup_hierarchy=0"
```

Sollten schon andere parameter drin stehen diese logischerweise hinzufügen und nicht ersetzen...

nun update grub ausführen

```
update-grub
```

## Certiface not Found

```
journalctl -u unifi-protect
```

Ausgabe:

```
[Error: ENOENT: no such file or directory, open '/data/unifi-core/config/unifi-core.crt'] {
Jul 07 10:18:36 unvr node[220735]:  errno: -2,
Jul 07 10:18:36 unvr node[220735]:  code: 'ENOENT',
Jul 07 10:18:36 unvr node[220735]:  syscall: 'open',
Jul 07 10:18:36 unvr node[220735]:  path: '/data/unifi-core/config/unifi-core.crt'
Jul 07 10:18:36 unvr node[220735]: }
Jul 07 10:18:36 unvr node[220735]: Unifi core certificate not found
```

---

Version #8

Erstellt: 7 Juli 2024 09:28:19 von Admin

Zuletzt aktualisiert: 7 Juli 2024 10:51:43 von Admin