# Installation der VMS

Ich habe Ubuntu und Fedora gewählt und diese mit Virtual Box installiert. Einfach das ISO File mounten und dann anschließend starten und die Installation durchführen.

# Open SSH

Habe ich auf Ubuntu installiert. Dies wurde mir gleich bei der Installation des Systems angezeigt. Ich habe dies dort auch gleich ausgewählt. Ansonsten kann man es mit sudo apt install openssh-server installieren.

## Konfiguration

Erlauben von SSH auf Port 22 würde gehen mit -> sudo ufw allow ssh, sudo ufw enable, sudo ufw status, da wir aber auf einen anderen Port setzen brauchen wir das nicht.

In dem File „sshd\_config“ muss man **PermitRootLogin** auf **no** setzen um root Login zu deaktivieren. **PasswordAuthentication** **yes** muss man auf **false** setzen.

Dasselbe Spiel wie bei Punkt 1 wir setzen jetzt den Port auf einen nicht Standard Port. Im File sshd\_config setzen wird den Port jetzt auf 22000. Den Port mit sudo ufw allow 22000/ssh erlauben. Dann den SSH Server neu starten.

OpenSSH beim Start automatisch starten geht mit sudo systemctl enable ssh

# Apache Server

Diesen installieren wir mit dem Befehl sudo apt install apache2

Auf Fedora installieren mit dem Befehl sudo yum -y install httpd

Den Port 80 erlauben mit sudo ufw allow 'Apache'

Auf Fedora: sudo firewall-cmd --permanent --add-port=80/tcp, sudo firewall-cmd --permanent --add-port=443/tcp, sudo firewall-cmd –reload.

Der Apache Server startet im Normalfall automatisch, so war es auch bei mir.

Auf Fedora: Apache starten mit dem Befehl: sudo systemctl start httpd und damit es beim Booten startet sudo systemctl enable httpd.

# Wireguard auf Ubuntu als Server

Installation von Wireguard mit sudo apt install software-properties-common, sudo add-apt-repository ppa:wireguard/wireguard und sudo apt install wireguard.

Public und Private Keys kann man mit wg genkey | sudo tee /etc/wireguard/privatekey | wg pubkey | sudo tee /etc/wireguard/publickey erstellen lassen. Diese befinden sich dann in /etc/wireguard. Private Key auslesen (siehe Screenshot).

Konfigurationsfile mit nano /etc/wireguard/wg0.conf erstellen. Hier fügen wir das ein:

[Interface]  
Address = 192.168.0.1/24  
SaveConfig = true  
ListenPort = 51820  
PrivateKey = Hier Private Key einfügen  
PostUp = iptables -A FORWARD -i %i -j ACCEPT; iptables -t nat -A POSTROUTING -o enp0s3 -j MASQUERADE  
PostDown = iptables -D FORWARD -i %i -j ACCEPT; iptables -t nat -D POSTROUTING -o enp0s3 -j MASQUERADE

Private Key

Bei PostUp und PostDown **eigenes** public network Interface eingeben. Siehe enp0s3. Dies kann man finden mit dem Befehl ip -o -4 route show to default | awk '{print $5}'.

Das wg0 Interface jetzt einschalten mit sudo wg-quick up wg0.

Beim Booten automatisch starten mit sudo systemctl enable wg-quick@wg0.

Jetzt noch die Firewall freigeben. Dafür öffnen wir das File „sysctl.conf“ mit einem Editor z.B. Nano.

Befehl: sudo nano /etc/sysctl.conf. Wir fügen dort net.ipv4.ip\_forward=1 ein. Falls notwendig setzen wir den Port 51820/udp noch frei. Befehl: sudo ufw allow 51820/udp.

Unsere Ubuntu Vm sollte nun als Server dienen.

## Wireguard auf Arch als Client

Installation auf Fedora mit sudo dnf install wireguard-tools. Anschließen holen wir uns wieder Private und Public Key. Befehl: wg genkey | sudo tee /etc/wireguard/privatekey | wg pubkey | sudo tee /etc/wireguard/publickey

Das File wg0.conf wieder erstellen mit sudo nano /etc/wireguard/wg0.conf und folgendes einfügen.

[Interface]  
PrivateKey = CLIENT\_PRIVATE\_KEY  
Address = 192.168.0.2/24  
[Peer]  
PublicKey = Hier Server Key einfügen  
Endpoint = 192.168.0.151820  
AllowedIPs = 0.0.0.0/0

Public Key Server

Client Private Key mit dem Befehl: sudo cat /etc/wireguard/privatekey bekommen.