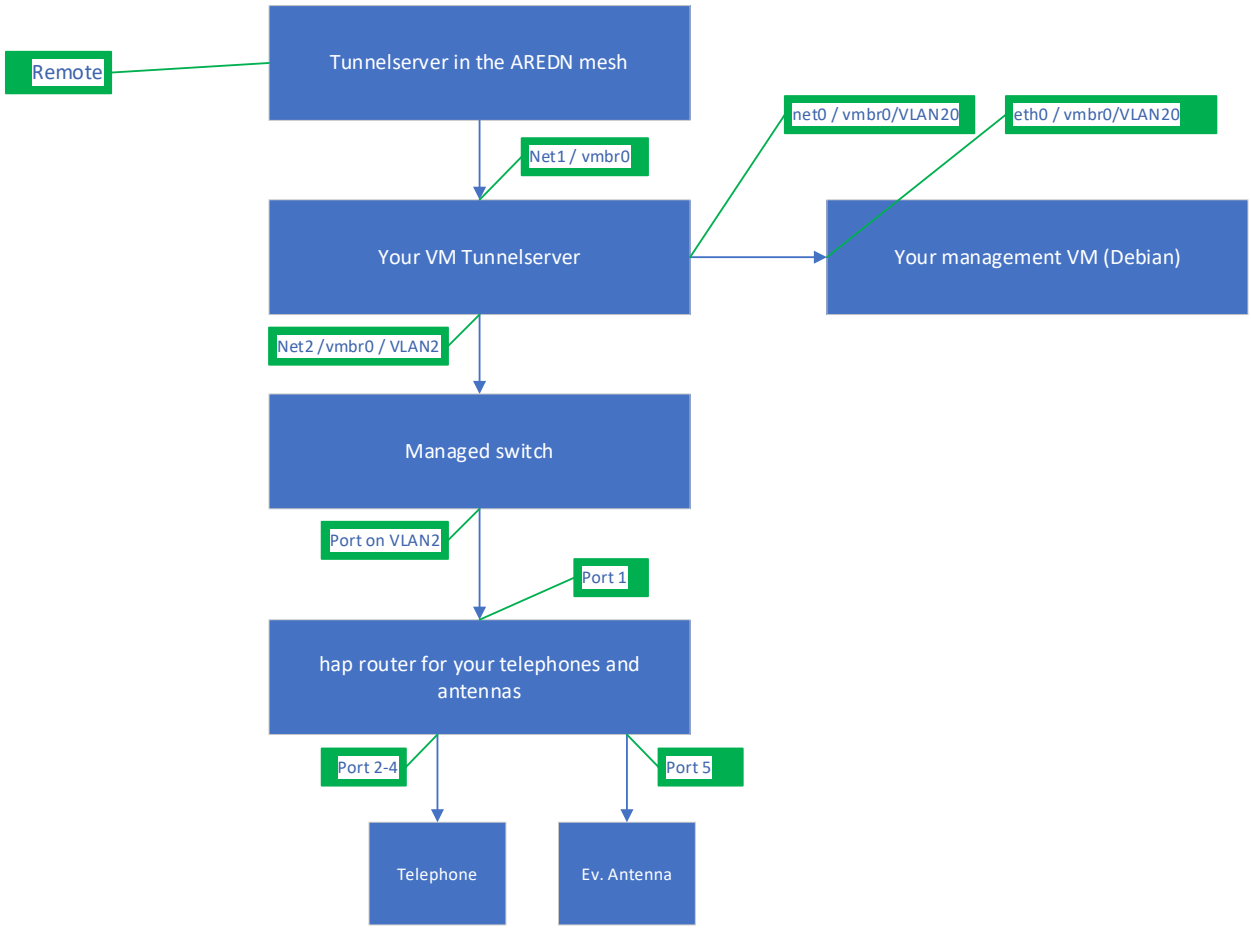
AREDN on Proxmox

Inhalt

System Overview

This is the setup we want to achieve:



Needed Steps

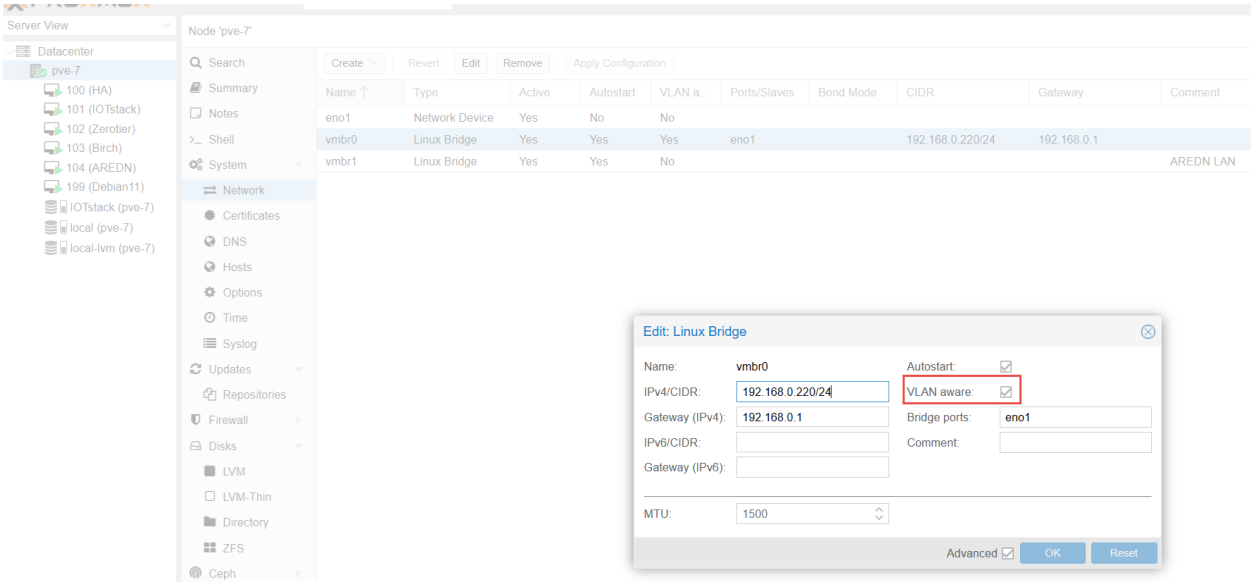
1- Install Proxmox on an X.86 machine (PC)
2- Adjust network of the node
3- Create an AREDN VM
4- Create a Debian VM (or any other distribution that can run a terminal and a browser)
5- Connect to AREDN from Debian to setup the network configuration as well as the AREDN node

Install Proxmox

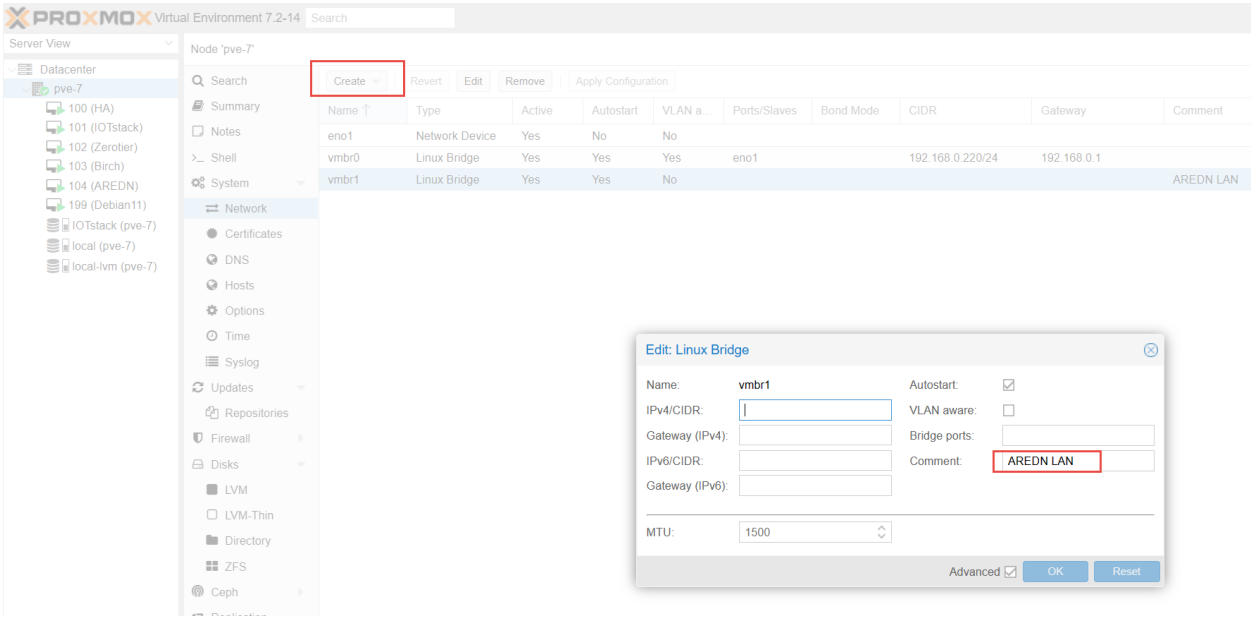
You find many tutorials to install Proxmox on an X86 machine.

Adjust the network of the node

Make vmbr0 VLAN aware:



Create a second bridge for the management of the AREDN machine. This bridge is not connected to your home network. It only exists in Proxmox and will connect the AREDN machine to the Debian machine. Like that we insulate the internal DHCP server from your home network where it can hurt.



Create AREDN VM

Based on: https://i12bretro.github.io/tutorials/0405.html

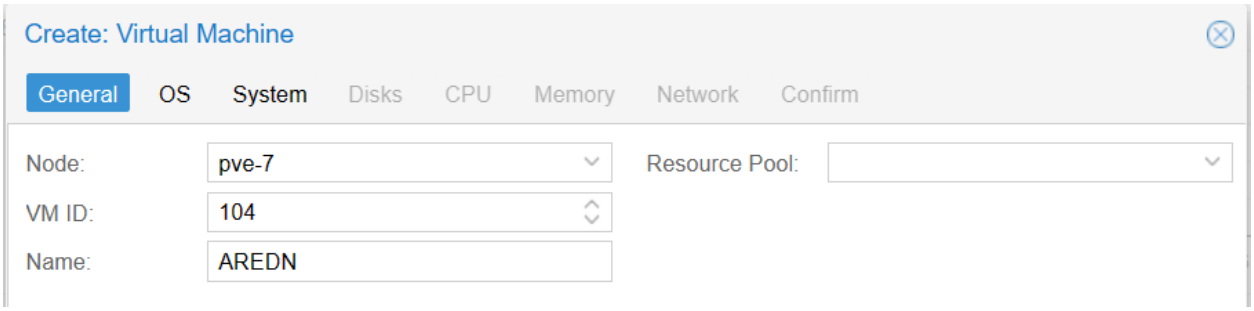
and

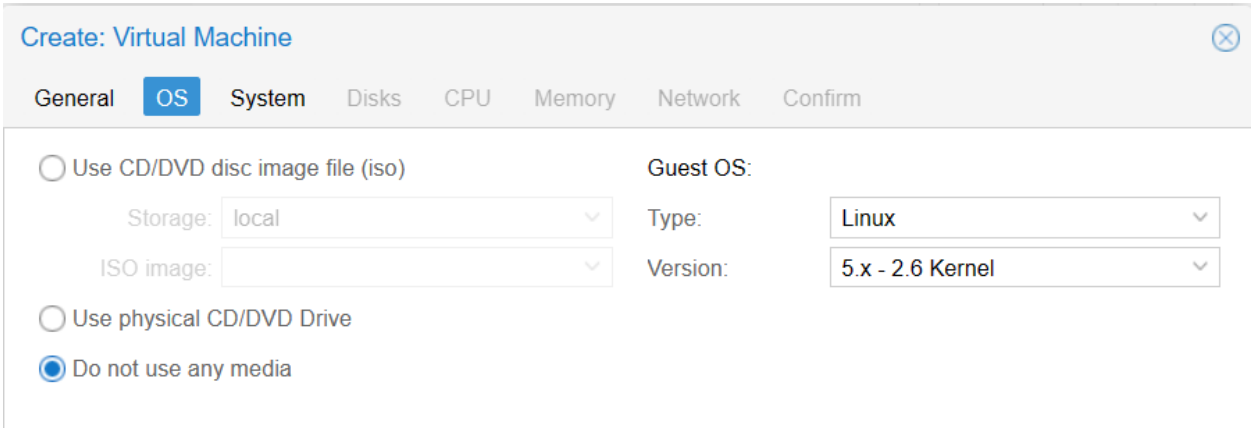
https://youtu.be/8nsdrWeeg8o

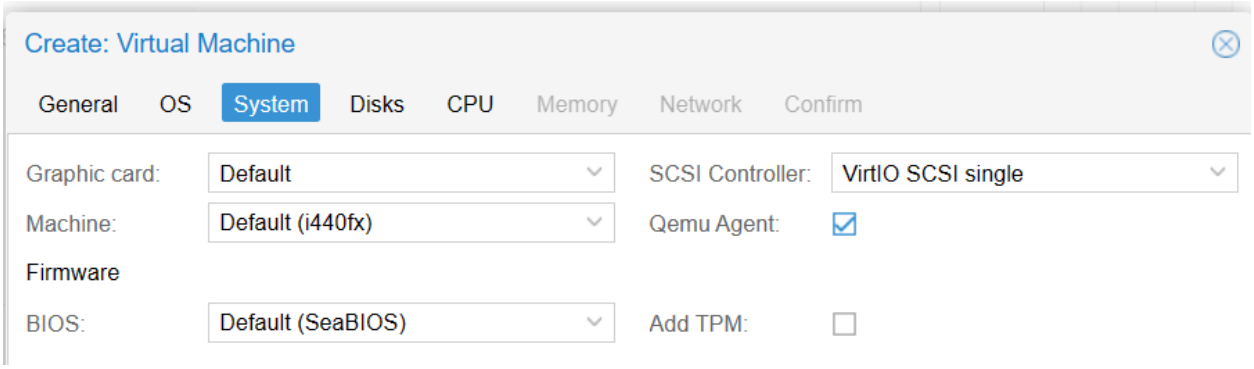
Creating the VM

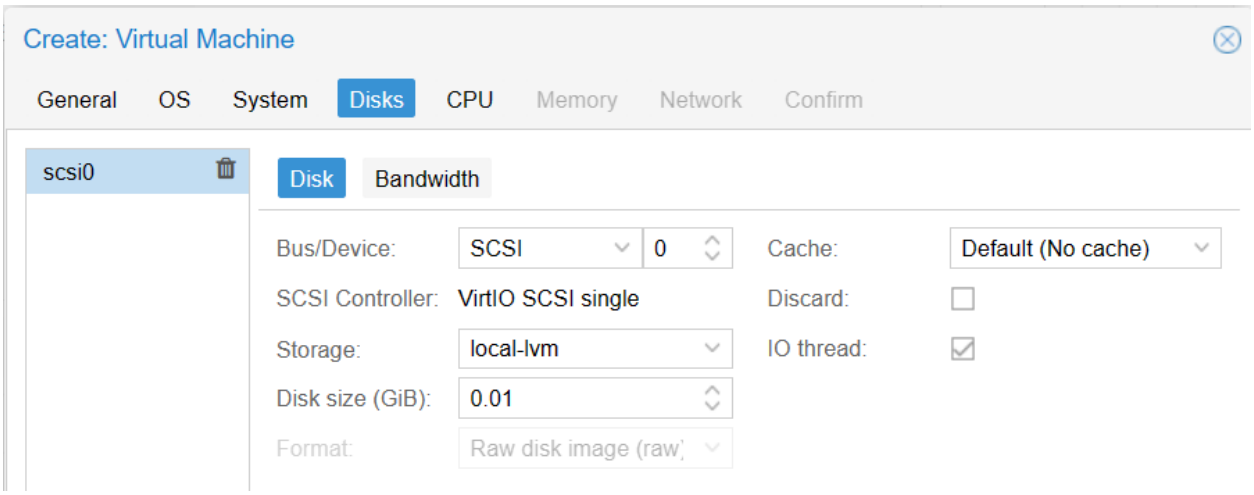
Open a web browser and navigate to the Proxmox web UI https://ProxMoxDNSorIP:8006/

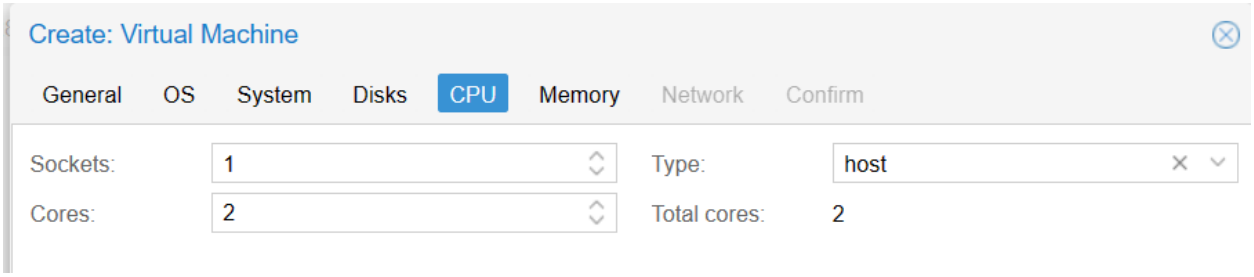
Click the Create VM button at the top right

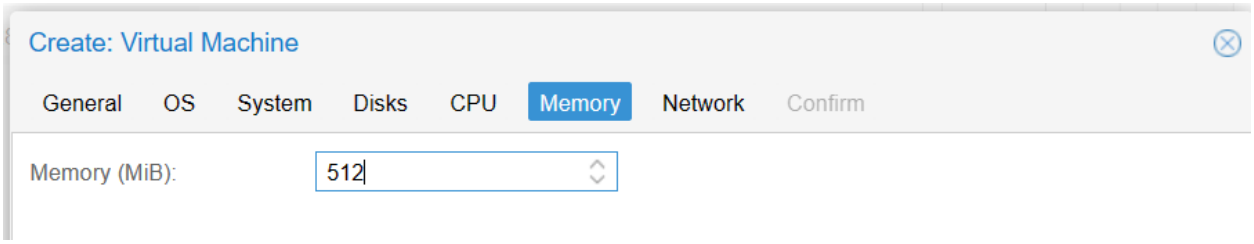


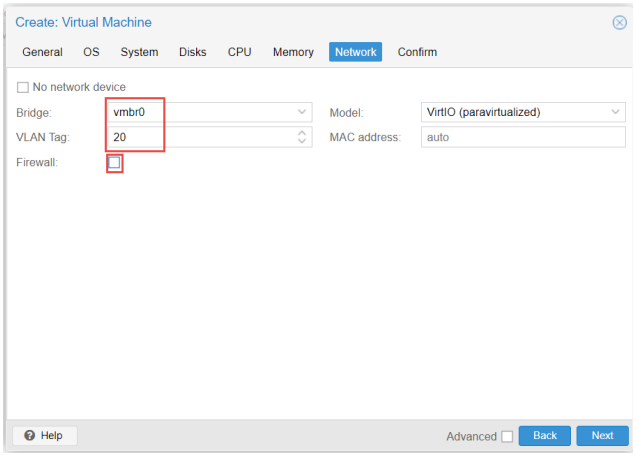




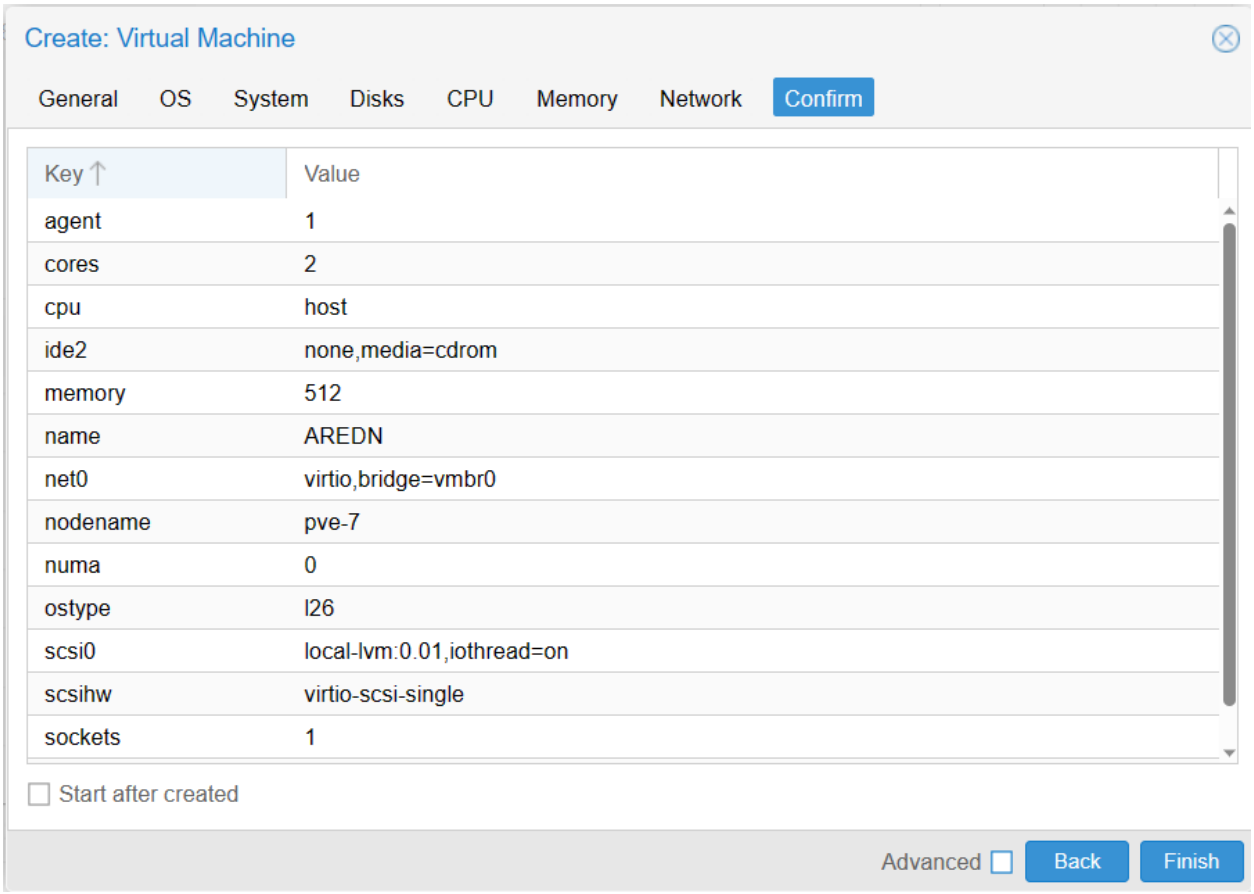






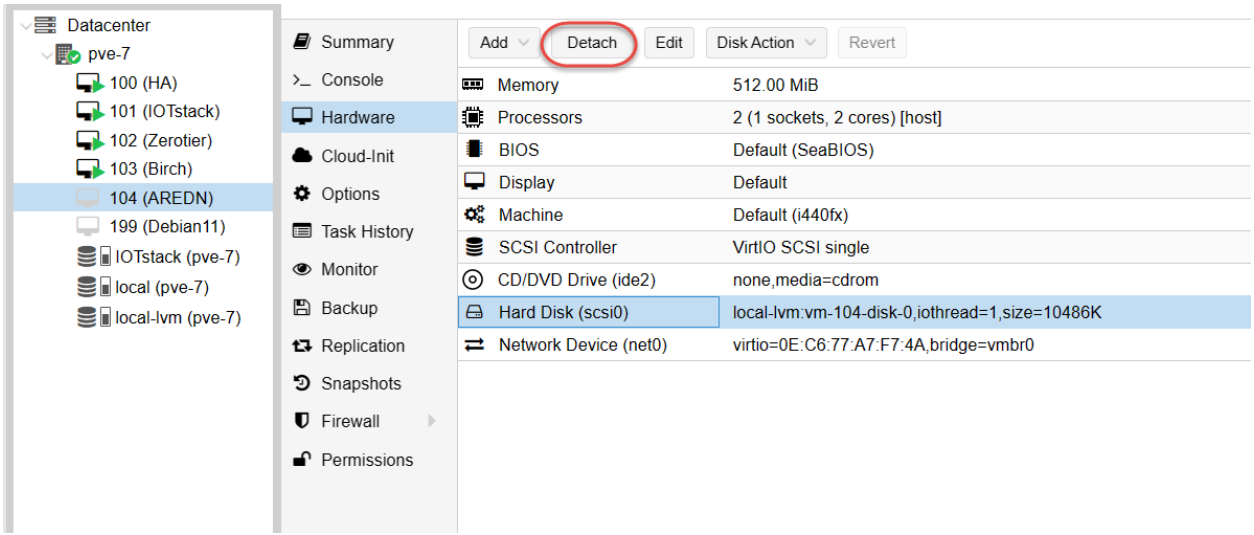


Add VLAN 20 (to prevent DHCP server interacting with your home network (and untick firewall):

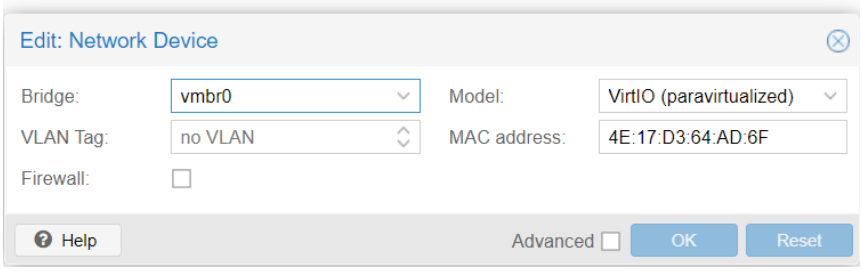


Confirm the summary

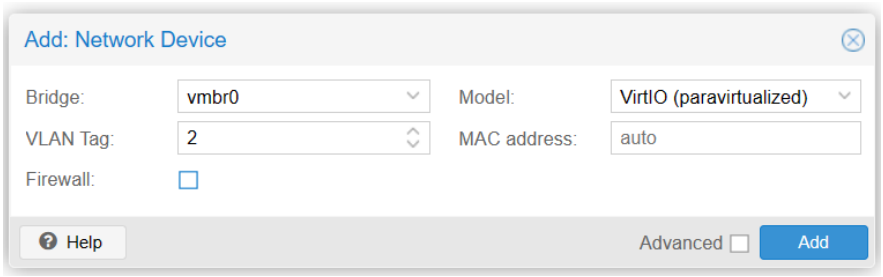
Select the newly created OpenWRT VM from the left navigation panel and detach and remove the disk



Click the Add button > Network Device. This will become your LAN connection



Add a third network with VLAN20 for DtD (untick Firewall)

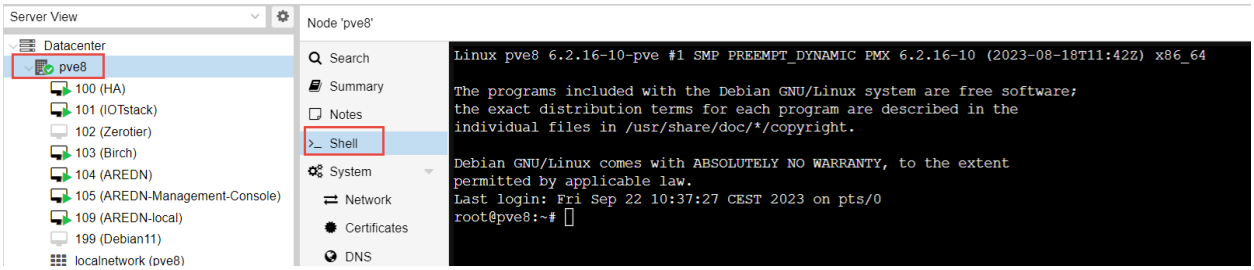


Result:

Network Device (net0)	virtio=92:E6:52:0D:E8:AA,bridge=vmbr0,tag=20
Network Device (net1)	virtio=4E:17:D3:64:AD:6F,bridge=vmbr0
Network Device (net2)	virtio=2A:D4:97:EB:34:CC,bridge=vmbr0,tag=2

Setting Up the OpenWRT Disk

Go to the shell of the server and download the image




```
wget -O aredn.img.gz
http://downloads.arednmesh.org/releases/3.23.8.0/targets/x86/64/aredn-3.23.8.0-
x86-64-generic-ext4-combined.img.gz
```

extract the openwrt img

```
gunzip ./aredn.img.gz
```

rename the extracted img

```
mv ./aredn*.img ./aredn.raw
```

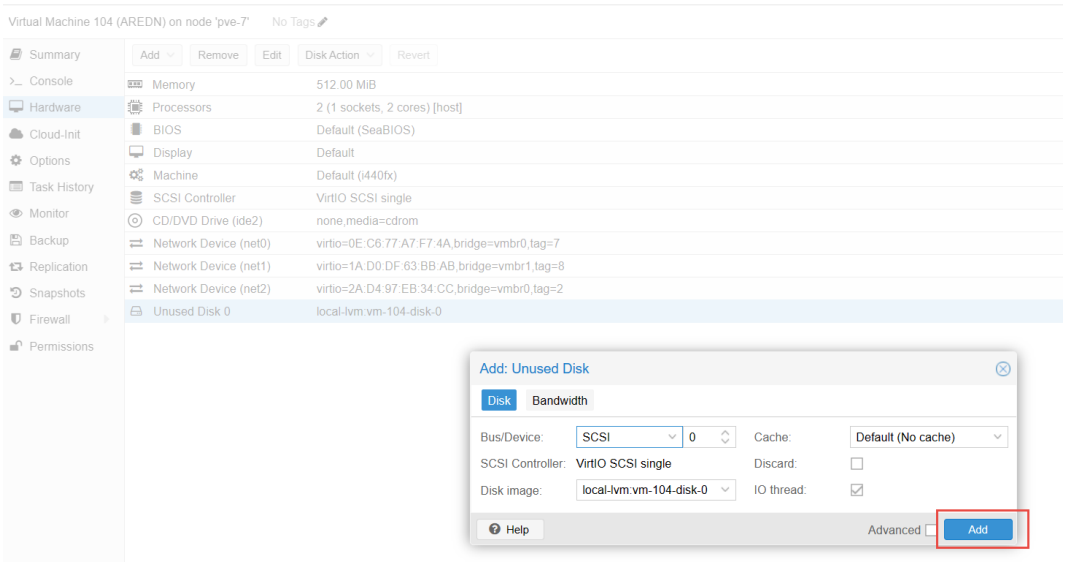
increase the raw disk to 512 MB

```
qemu-img resize -f raw ./aredn.raw 512M
```

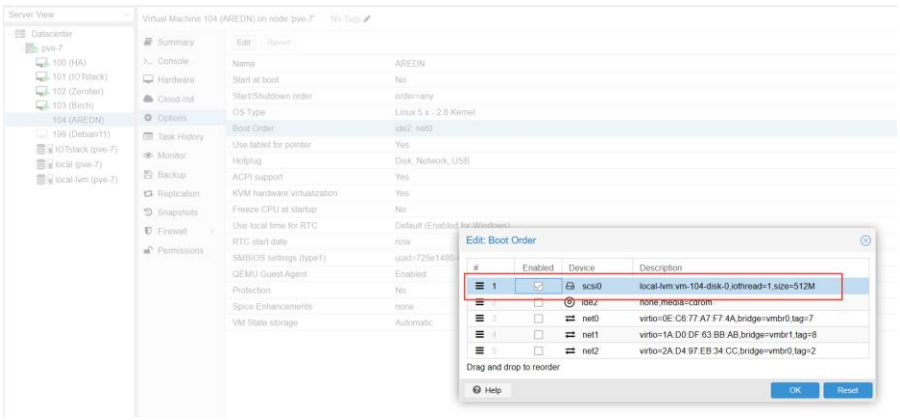
import the disk to the openwrt VM (replace 104 with your VM number)

```
qm importdisk 104 /root/aredn.raw local-lvm
```

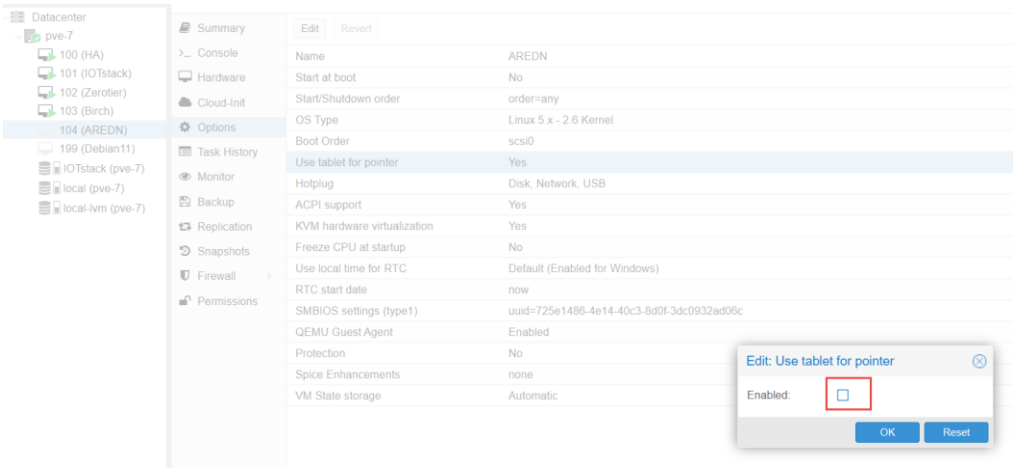
The disk is now visible as “Unused Disk 0”. Add the disk (double click on it) to your VM:



Change boot order:



Double click Use tablet pointer > Uncheck the Enabled box > Click OK



Click the Start button in the top right of the Proxmox screen

Click the Console link to watch the boot process

Wait for the text to stop scrolling and press “Enter”

Run the following command to change/set the root password

passwd

Type a new root password twice

For the moment we are done with the AREDN node. Because we had to separate the LAN from our home network, we only can manage the node in the Proxmox terminal (it does not offer a browser). To solve the problem, we add a VM with plain vanilla Debian and connect it to VLAN20, too. Like that we have a private connection to manage our AREDN node.

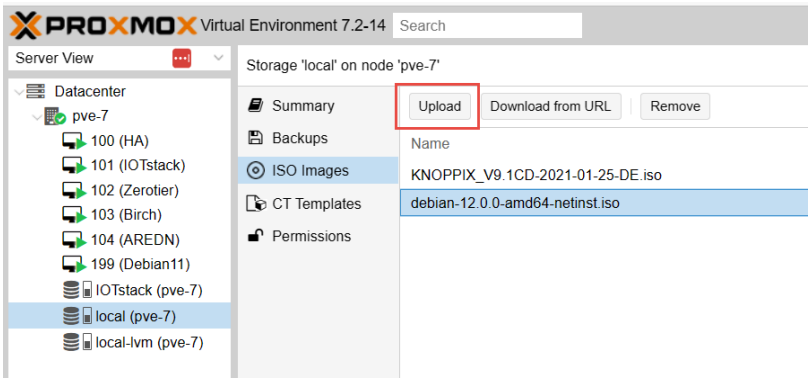
Of course you can use any other VM that offers a browser and is connected to vmbr0/20.

Create a Debian VM as a management console

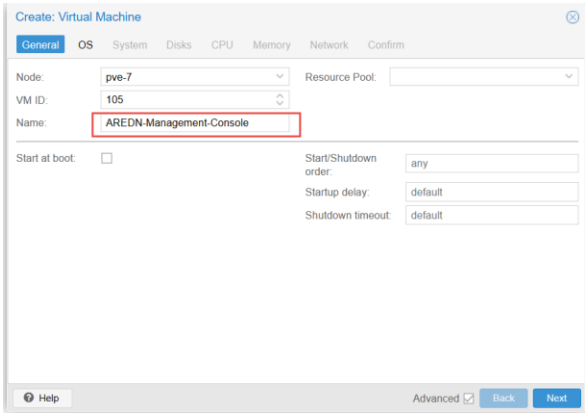
https://youtu.be/OUC7DMBfR3Y?si=y4r3edcwzzpd2JHJ

Download the Debian 11 iso Download

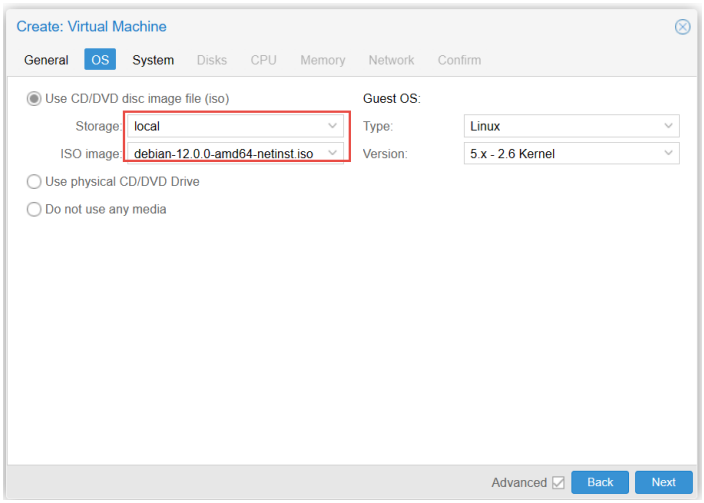
https://cdimage.debian.org/debian-cd/current/amd64/iso-cd/ (debian-12.1.0-amd64-netinst.iso)



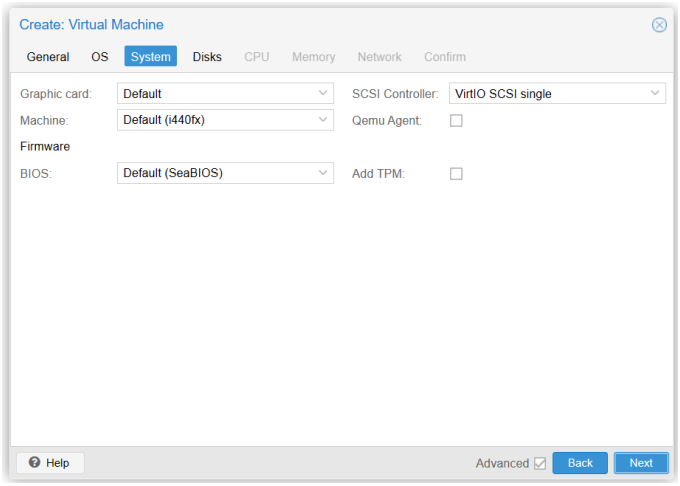
Create VM



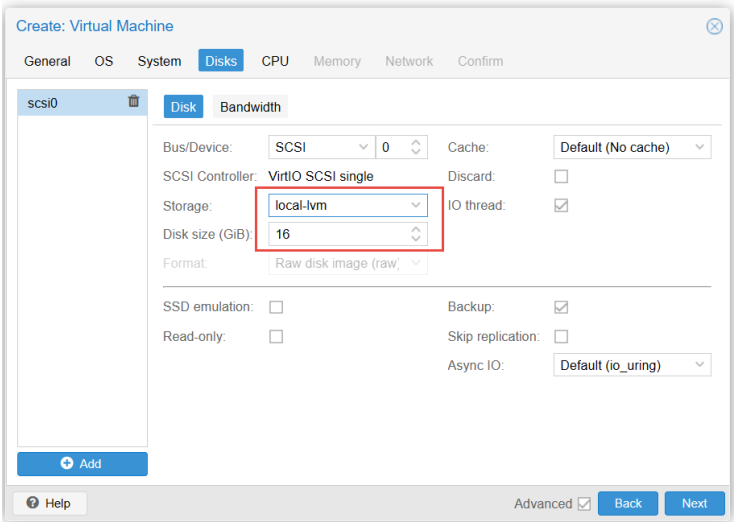
On the OS tab,



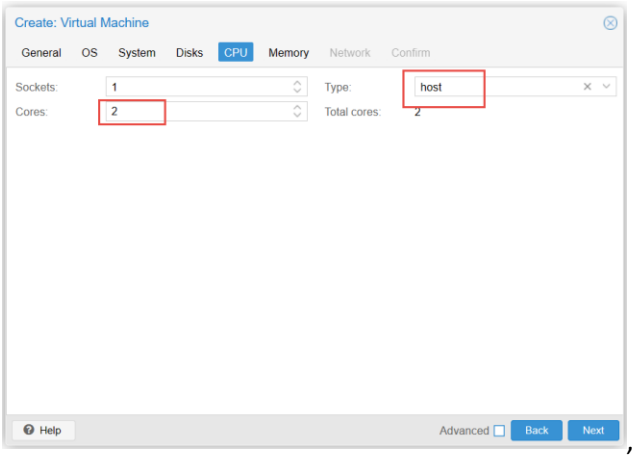
Leave the defaults on the System tab



On the Hard Disk tab, set the Disk size to 16 GiB or more

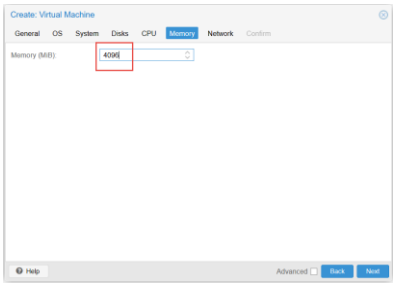


On the CPU tab

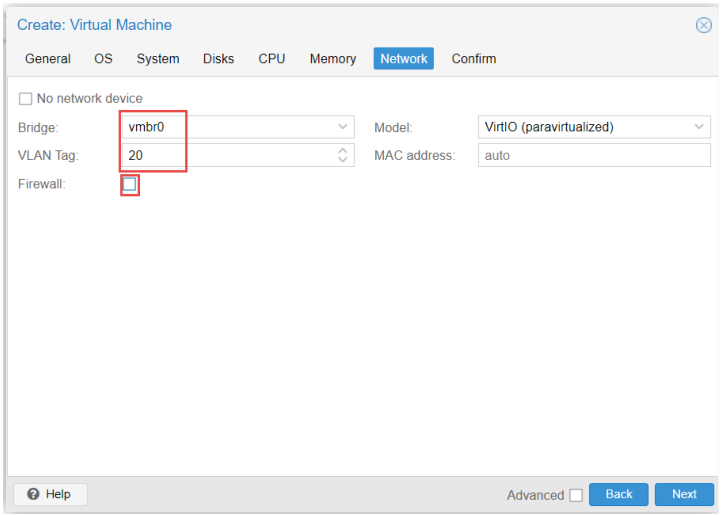


,

On the Memory tab

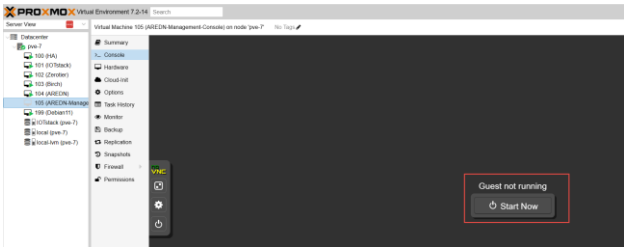


Network tab (untick Firewall)

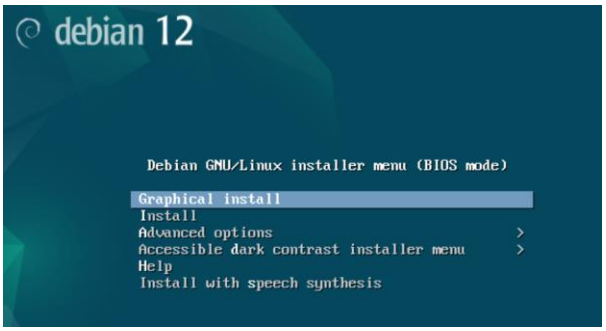


Verify the summary and click Finish

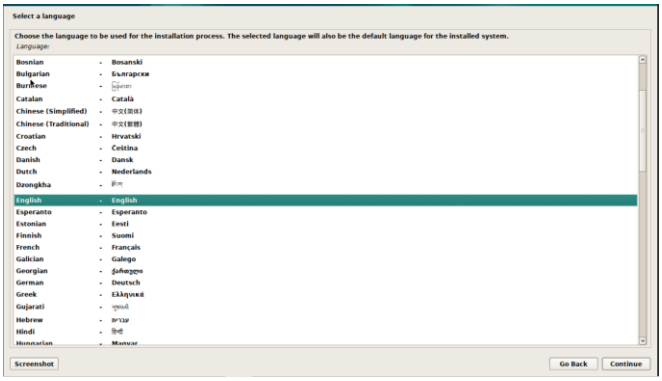
Start the VM



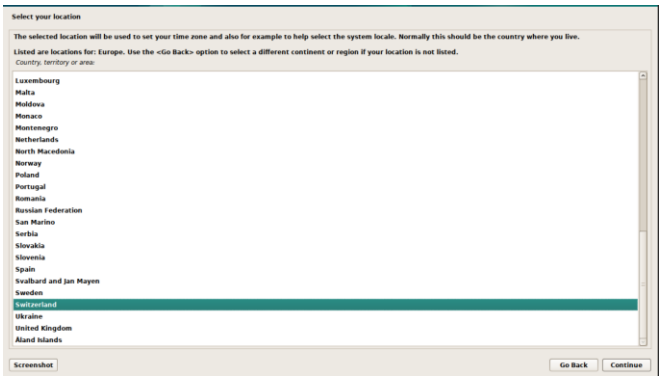
Graphical Install



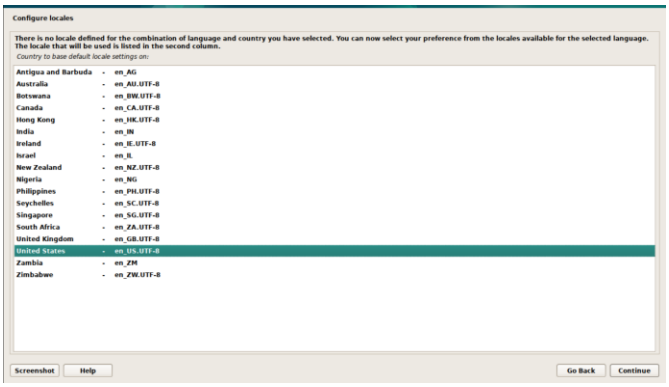
Select a language



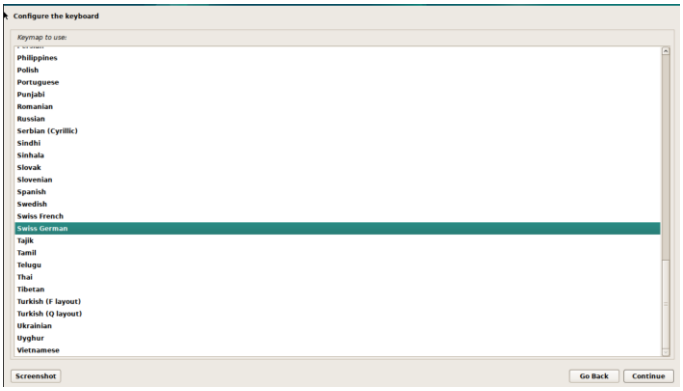
Select a Location (other →Europe→Switzerland)



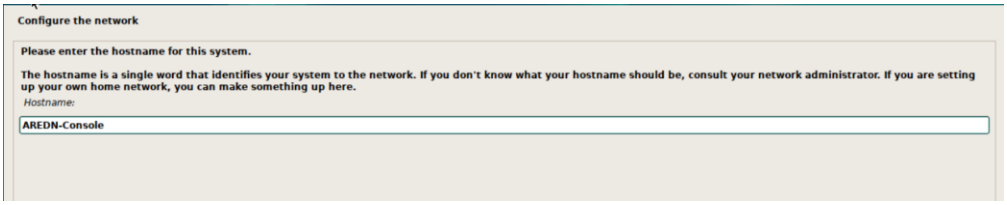
Select locales (can be adapted later)



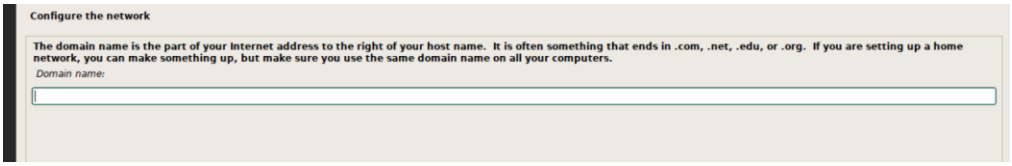
Select a keyboard layout (can be changed later)



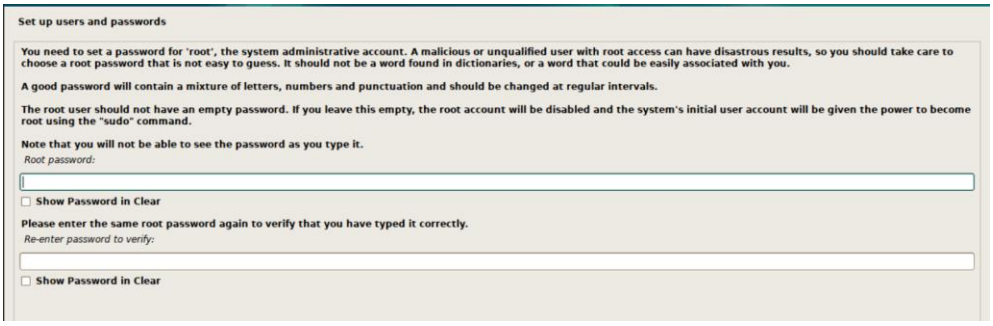
Enter a hostname for the VM



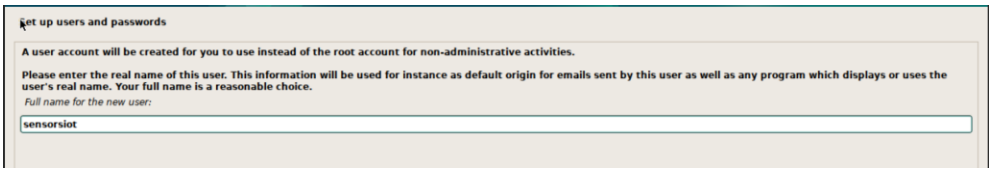
Leave domain name empty



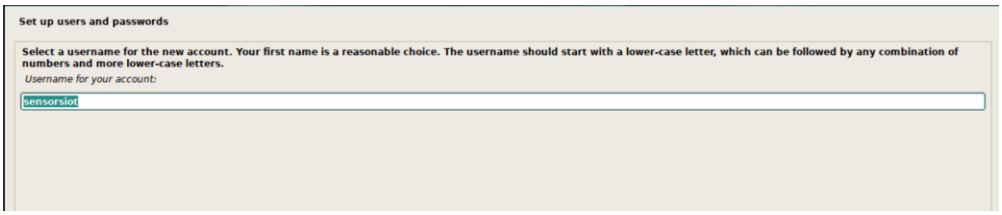
Leave root password empty



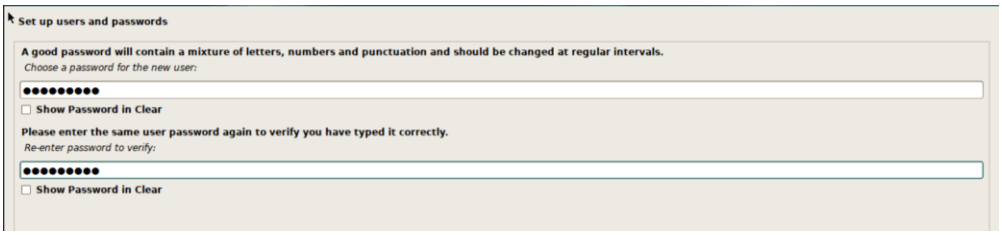
Enter the full name for the new user



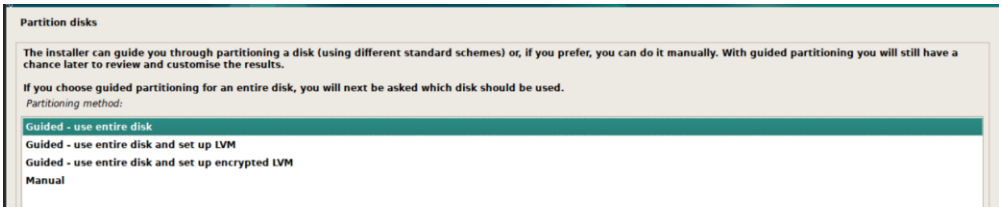
Enter the username for the new user

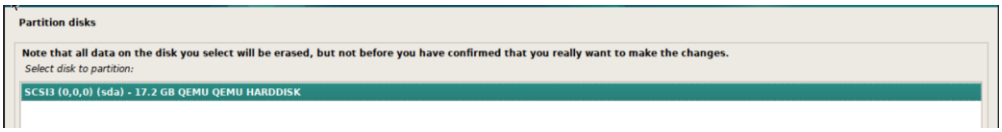


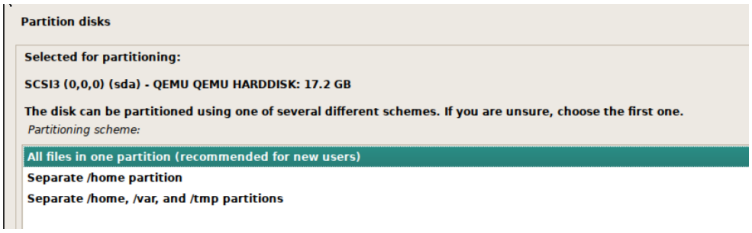
Enter and confirm a password for the new user



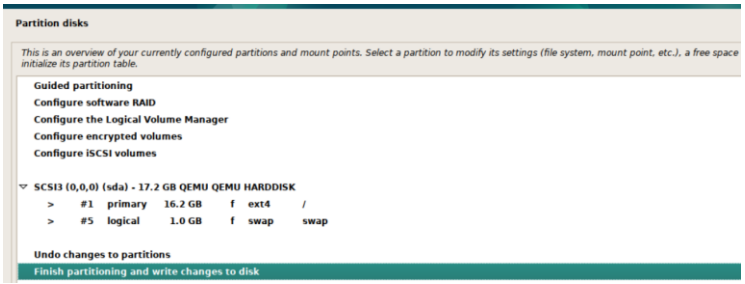
Select Disk



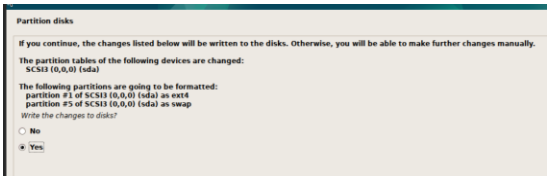




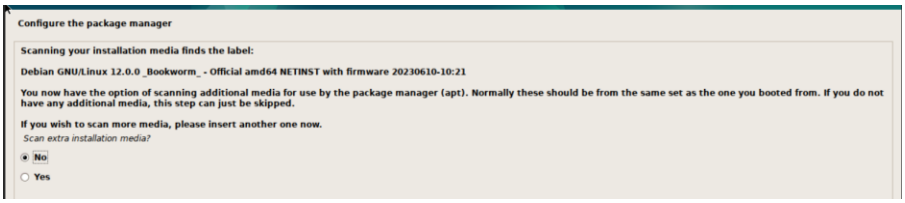
Check and click Continue

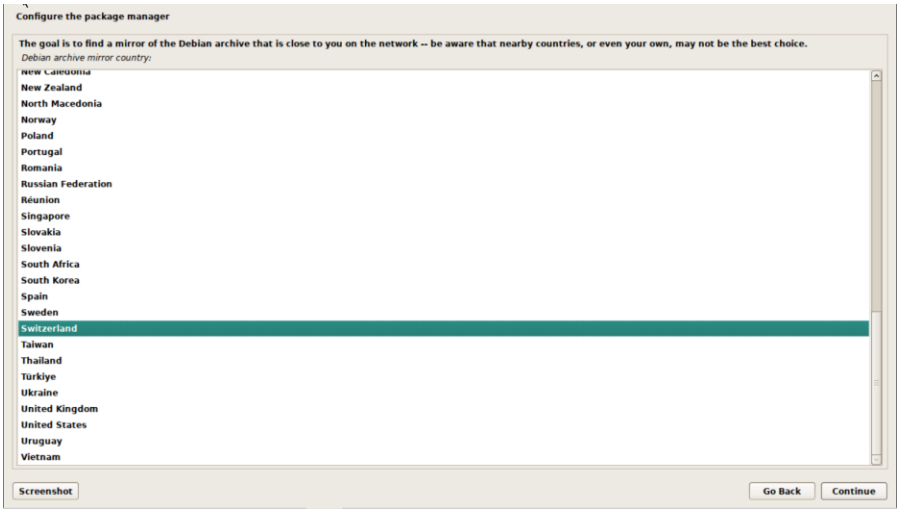


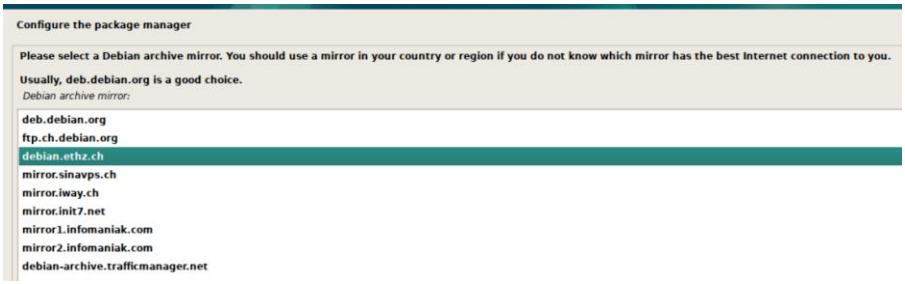
Select Yes to confirm writing the changes



Wait for Debian to copy and install files

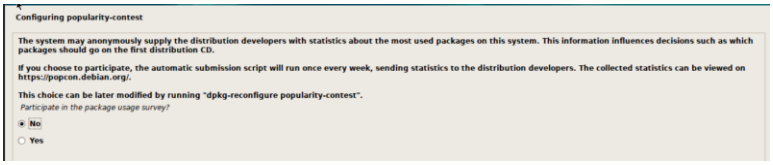




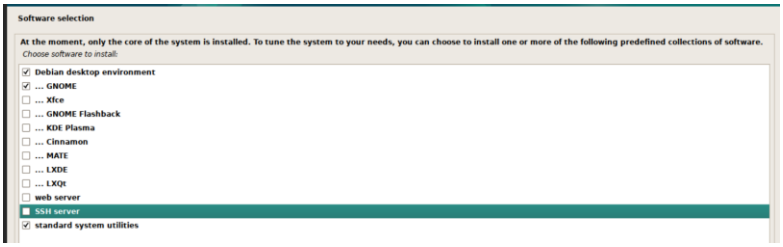


Wait

Participating in package survey



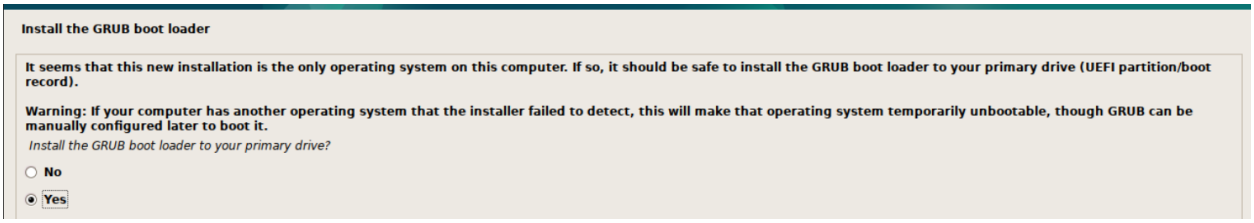
Select the software to install



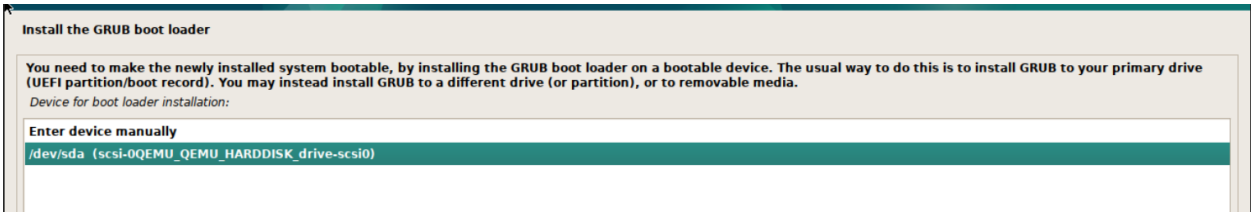
Select (tick) “SSH server”

Wait

Select Yes to install GRUB

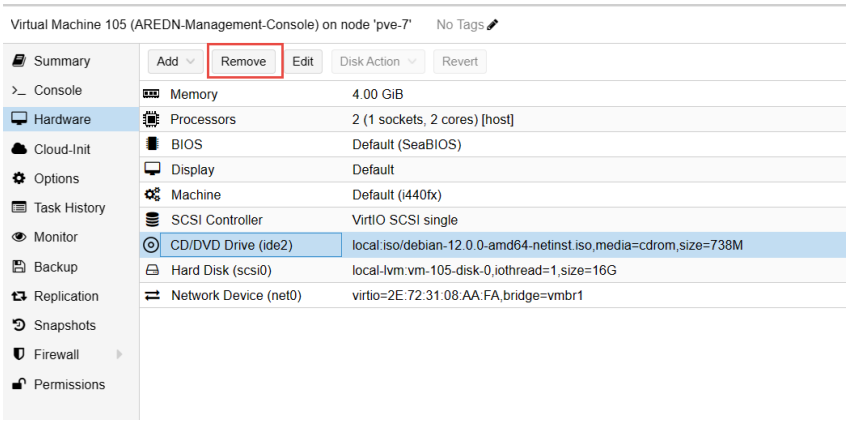


Select drive for the boot loader

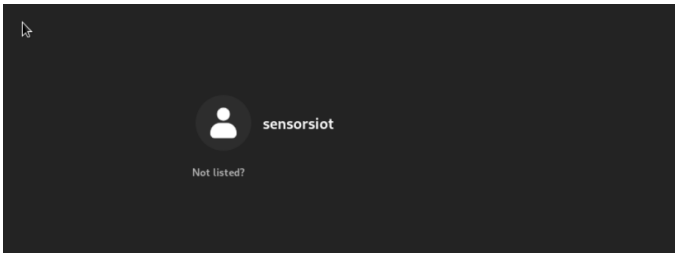


Reboot

Remove CD drive (it will only be removed after reboot)



Go to the Console and login



Start Firefox. On http://localnode.local.mesh you should see your AREDN server because your management console is connected to the AREDN node via the vmbr0/20 interface.

Setup AREDN server networks

Open terminal and ssh into the AREDN server:

```
ssh root@localnode.local.mesh -p 2222
```

Confirm fingerprint with “yes”

Desired mapping

In the end we would like the following mapping:

⇄ Network Device (net0)	virtio=92:E6:52:0D:E8:AA,bridge=vmbr0,tag=20	**eth0**	**LAN port (ports 2-4 on hap)**
⇄ Network Device (net1)	virtio=4E:17:D3:64:AD:6F,bridge=vmbr0	**eth1**	**WAN port (port 1 on hap)**
⇄ Network Device (net2)	virtio=2A:D4:97:EB:34:CC,bridge=vmbr0,tag=2	**eth2**	**DtD port (port 5 on hap)**

The mapping of the nets has to be done now in the terminal of the Management console. Connect first to your AREDN server:

Press yes, and enter the password

```
ssh-keygen -l root -p 2222 localnode
```

Now you are in the console

Type:

```
vi /etc/config/network
```

and hit the “insert” key.

Leave the Wi-Fi configuration as it is:

```
### Loopback configuration
config interface loopback
        option device   "lo"
        option proto    static
        option ipaddr   127.0.0.1
        option netmask  255.0.0.0

#### WIFI configuration
config device
        option name 'br-nomesh'
        option type 'bridge'
        option bridge_empty '1'

config interface wifi
        option device 'br-nomesh'
        option proto 'static'
        option ipaddr '10.124.142.47'
        option netmask '255.255.255.255'


config interface wifi_mon
        option proto    none
```

Add two ports to the bridge configuration:

```
### Bridge configuration
config device
        option name 'br0'
        option type 'bridge'
        list ports 'eth0'
        list ports 'eth1'
        list ports 'eth2'
```

Adjust the WAN configuration:

```
#### LAN configuration

config bridge-vlan
        option device 'br0'
        option vlan '3'
        list ports 'eth0:u'

config device
        option name 'br-lan'
        option type 'bridge'
        option macaddr 02:40:B0:C3:7B:A5
        list ports 'br0.3'

config interface lan
        option device 'br-lan'
        option proto 'static'
        option ipaddr '10.61.86.209'
        option netmask '255.255.255.240'
        option dns '8.8.8.8 8.8.4.4'
```

The LAN is connected to eth0 that is net0 in Proxmox on VLAN 20. Only our management console is connected to this network. So the DHCP server inside the AREDN server is insulated and does not harm your home network.

Connect the WAN port to the internet. eth1 is connected to the RJ45 connector of your server using vmbr0. It uses the general purpose VLAN 1 and will get a DHCP address from your home network.

```
#### WAN configuration

config bridge-vlan
        option device 'br0'
        option vlan '1'
        list ports 'eth1:u'

config device
        option name 'br-wan'
        option type 'bridge'
        list ports 'br0.1'

config interface wan
        option device 'br-wan'
        option proto 'dhcp'
```

The DtD (device-to-device) port is connected to the RJ45 of your Proxmox server via vmbr0, but this time using VLAN 2. To avoid problems with the DHCP server of other AREDN devices like the hap routers, we have to add a managed switch where we only transfer VLAN 2 to the ports where we connect port 5 of the hap routers or any “antennas”.

```
#### DtD configuration

config bridge-vlan
        option device 'br0'
        option vlan '2'
        list ports 'eth2:u'

config device
        option name 'br-dtdlink'
        option type 'bridge'
        list ports 'br0.2'

config interface dtdlink
        option device 'br-dtdlink'
        option proto 'static'
        option ipaddr '10.60.118.41'
        option netmask '255.0.0.0'
```

All networks have to marked “untagged” (e.g. “eth2:u”). They are tagged by Proxmox. And Proxmox does not like tagged stuff from the container.

The rest of the configuration file is only comments and is not changed for the moment.

Press the “escape” button and “:wq” to save your changes.

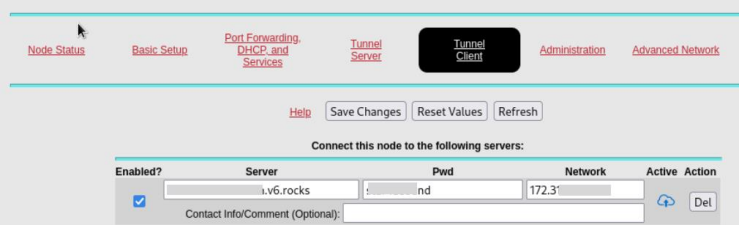
Reboot.

Now you can connect to localnode.local.mesh via Firefox and start to customize your server.

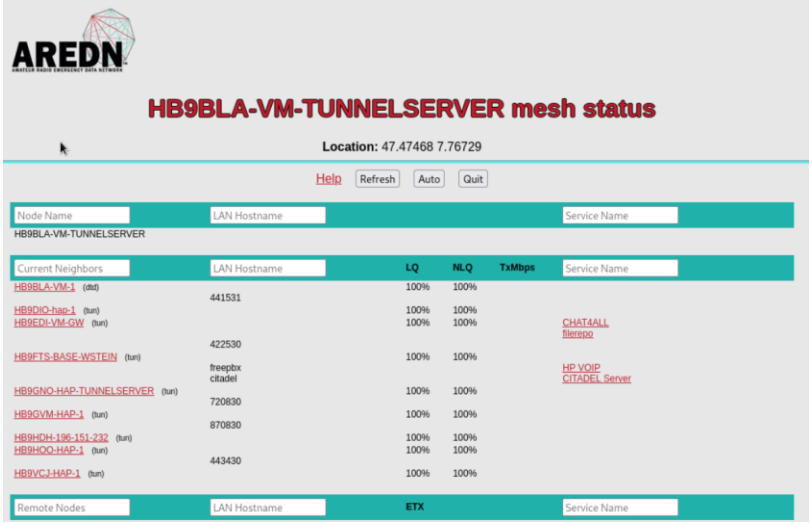
Give it the name: Callsign-VM-TUNNELSERVER

Connect it as a client to your tunnel server in the AREDN network. Make sure the administrator of your tunnel adjusts the name and the address of your new server. Maybe you want to run in parallel for a few days. After connecting to your tunnel server, you should see the other nodes in the net.

Now, you can add the tunnels you serve. Do not forget to forward port 5525 to the address of your tunnel server.

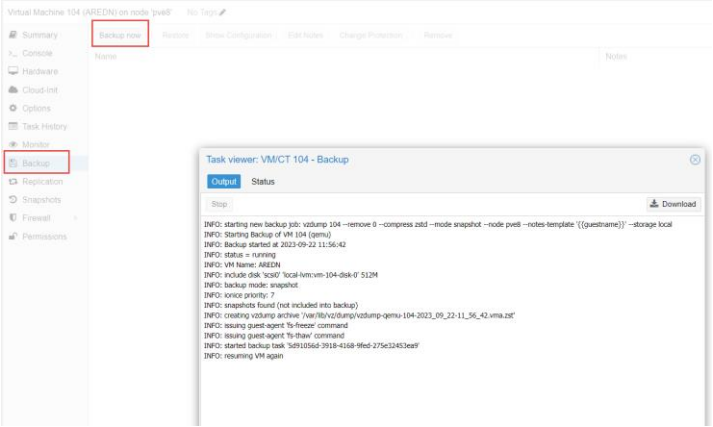


Now your Server should be connected to the SwissDigitalNet. Congratulations!



Backup

Backup machine to the local directory



You find this directory

```
cd /var/lib/vz/dump/
```

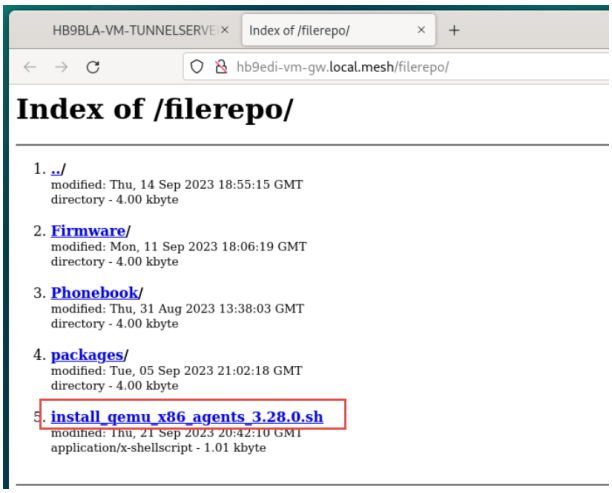
on the Proxmox server. It can be saved to the local disk using WinSCP

Install QUEMU Agent

Necessary to control (e.g. shut down) the AREDN VM from Proxmox

In your management console, you select the filerepo:

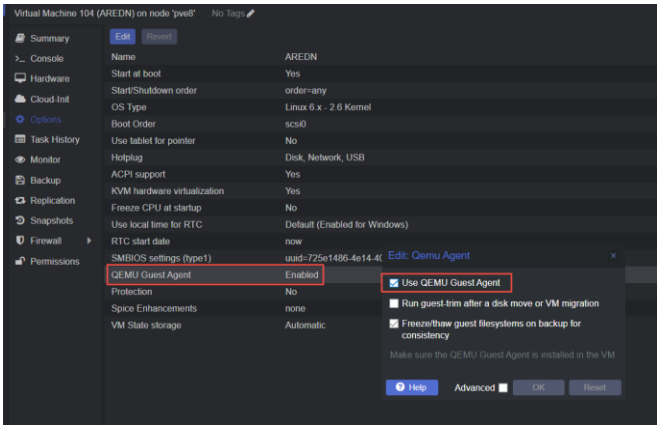
And copy this link:



Now you paste the link into your terminal and add the yellow command

```
curl http://hb9edi-vm-gw.local.mesh/filerepo/install_qemu_x86_agents_3.28.0.sh |ash
```

Finally, you enable Guest Agent



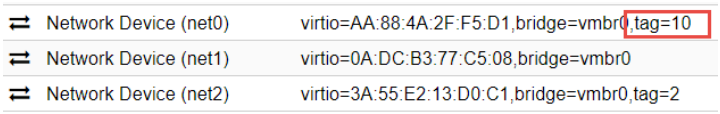
Now your tunnel server on a Gateway is ready to be connected to the world.

AREDN Virtual Machine as a telephone server

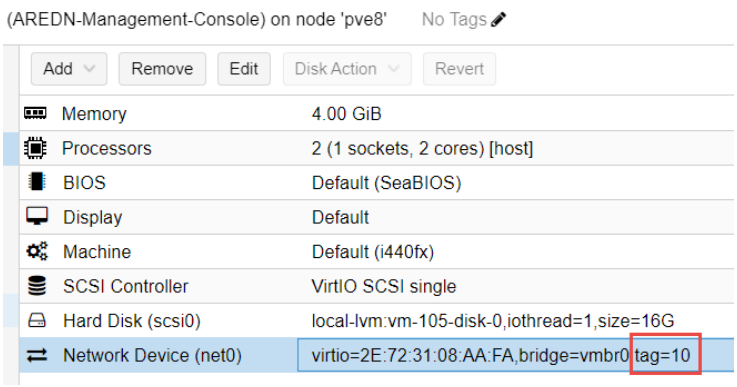
Setup

I add a new VM in addition to the tunnel server. If you do not operate a tunnel server, you just create a VM for your telephone server

The basis setup has to be done the same way as before. One difference is that we chose VLAN10 as the management connection:



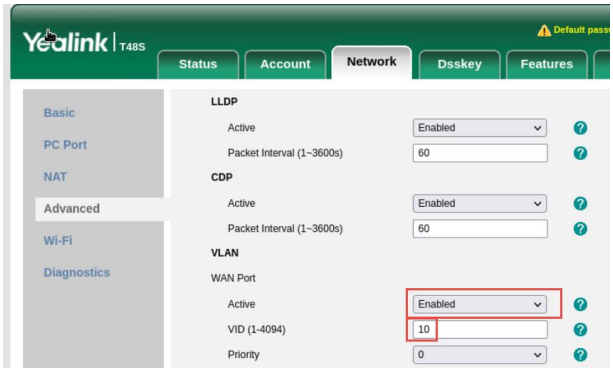
And we change the network of the management console, too:



Now we should be able to manage the telephone server from the management console.

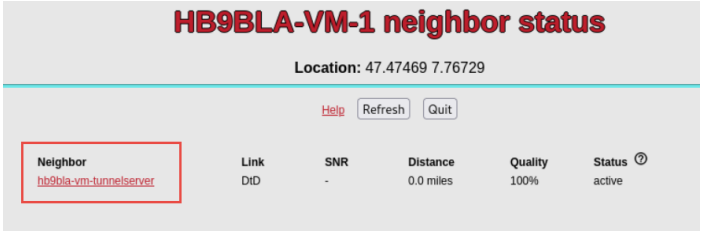
We can connect our phones to our home network instead of the hap router and they should get a normal address in the 192.168.xxx.xxx range.

Because our LAN of the VM is on VLAN 10, we must change the VLAN of the phone to 10, too. We login and go to Network→ Advanced and enable VLAN 10.



Now your phone should get a 10.xx.xx.xx address (instead of a 192.168.xxx.xxx) and you can install the phonebook and the SIP server as in every hap router and make the first test call.

The tunneelserver VM and the Telephone VM should be connected by a DtD link. You see this in the neighbor status:



Be aware that you can only manage your phones from the management console. They are not connected to your home network.

You can shut the management console VM down if you do not need it if you do not have too much resources on your server.

With this setup, the management console is only connected to the tunnel server (by VLAN 20). The tunnel server is connected to the telephone server by DtD. So you can reach all your devices from just one Management console.