AREDN on Proxmox

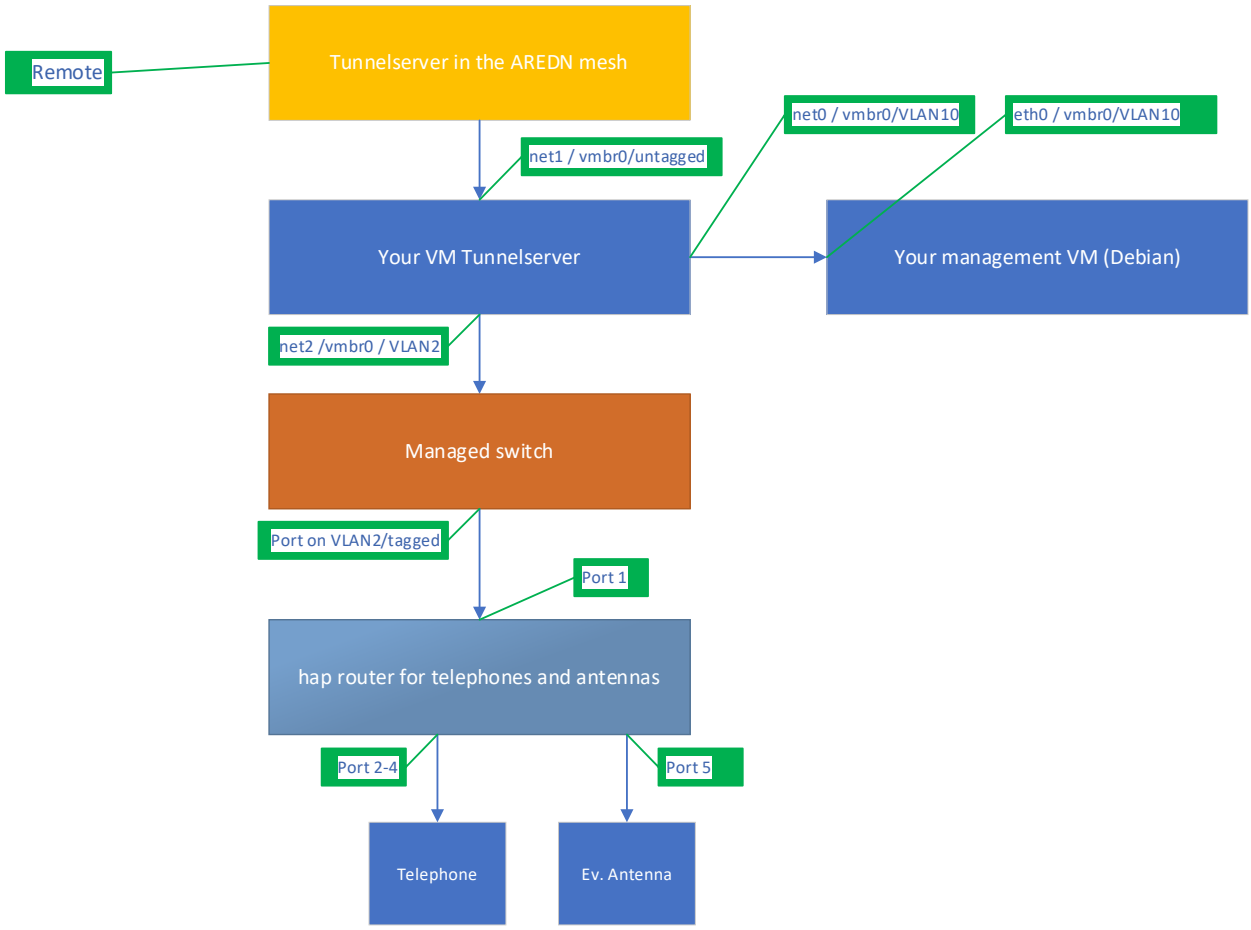
December, 18th 2023

German Video of Webinar on this channel: https://www.youtube.com/@uskapr113

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

Proxmox installation

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

Find the IP address of your Proxmox server (using Advanced IP Scanner)

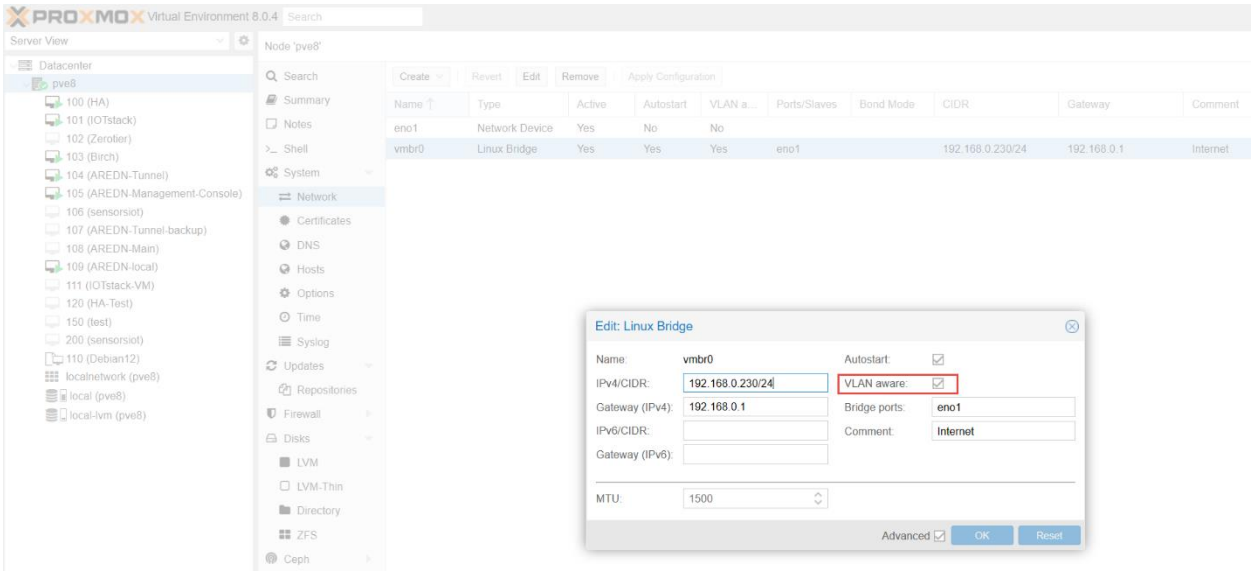
Open a web browser and navigate to the Proxmox web UI (e.g. https://192.168.0.230:8006/

VLAN enabling of the Proxmox network

Navigate to pve8→Network

vmbr0 is the same as port 1 on the hap router and is connected to the internet (for tunnels) It is default on Proxmox.

Make vmbr0 VLAN aware:



VLAN definition for this document

VLAN1: Used as “untagged” VLAN by AREDN

VLAN2: Used for Device-to-Device (DtD) communication by AREDN

VLAN10: Used as LAN network for the Tunnel Server (by the author, can be changed)

VLAN20: Used as LAN network for the Telephone Server (by the author, can be changed)

Create an AREDN VM (Tunnel or Telephone Server)

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

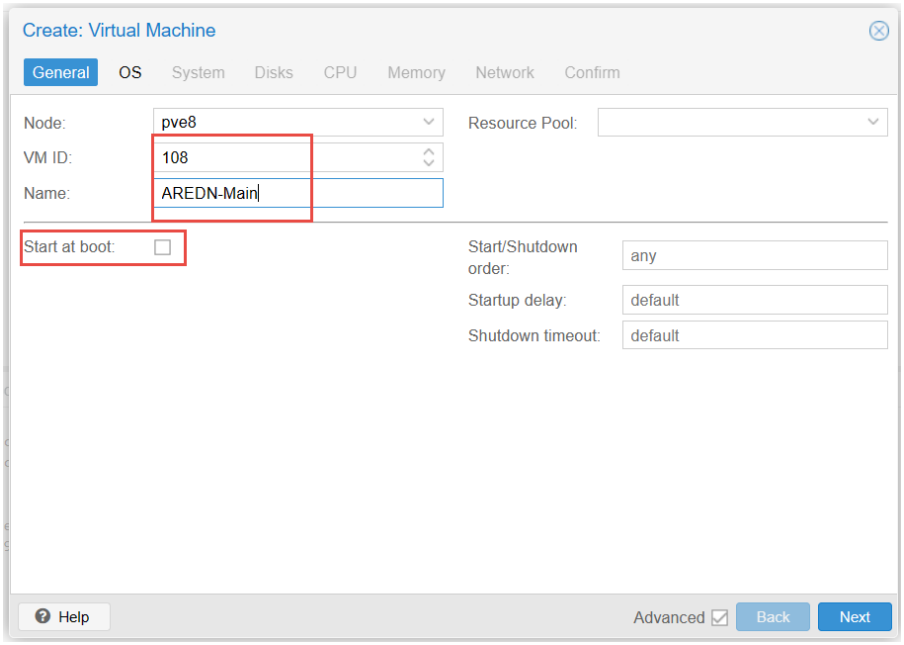
and

https://youtu.be/8nsdrWeeg8o

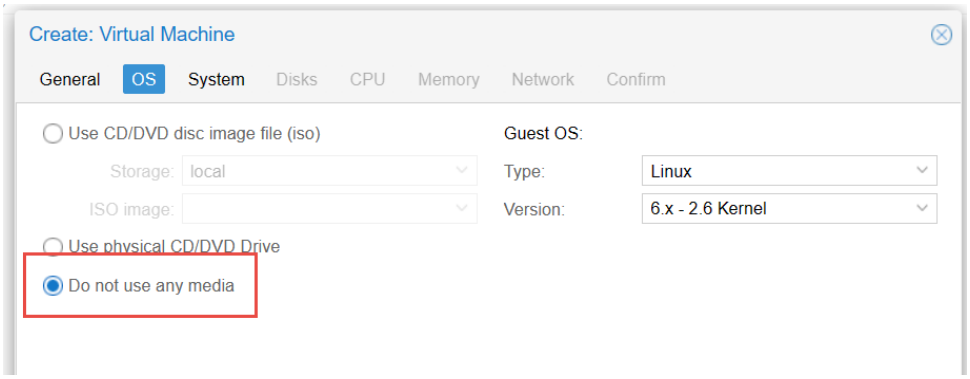
Creating the VM

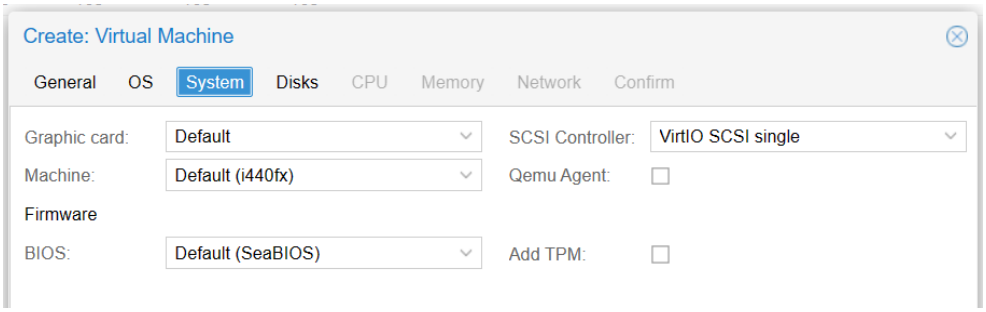
Click the Create VM button at the top right:

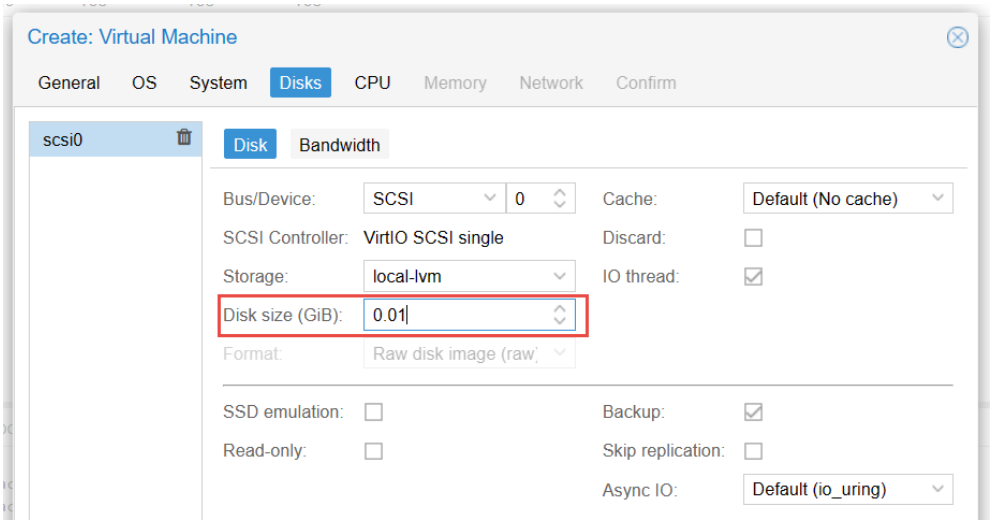
And fill in a name and an ID for your VM (start at boot not ticked):

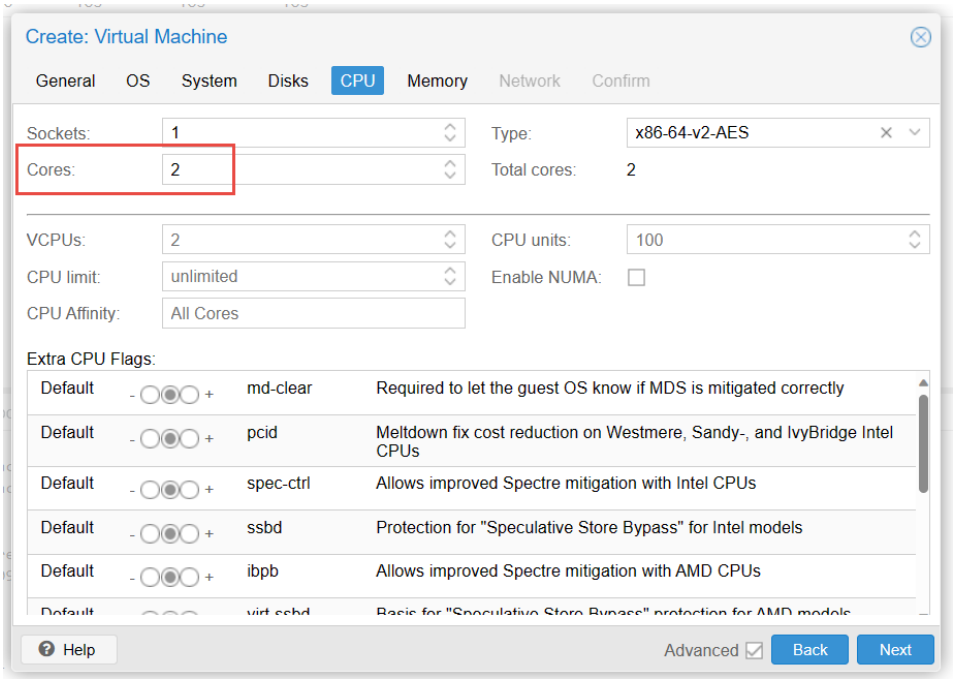


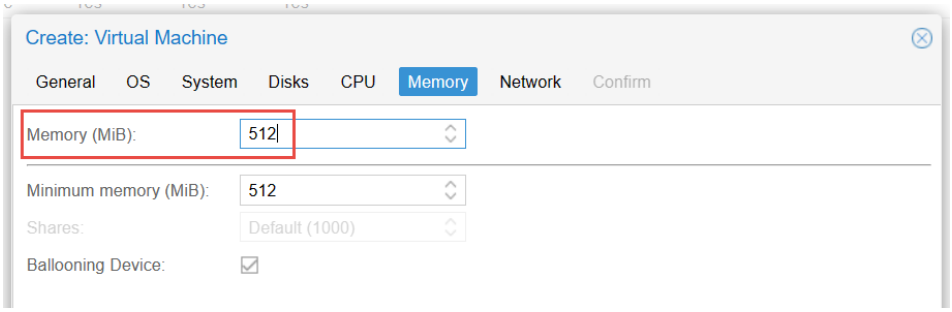
Start at boot can be ticked when everything works

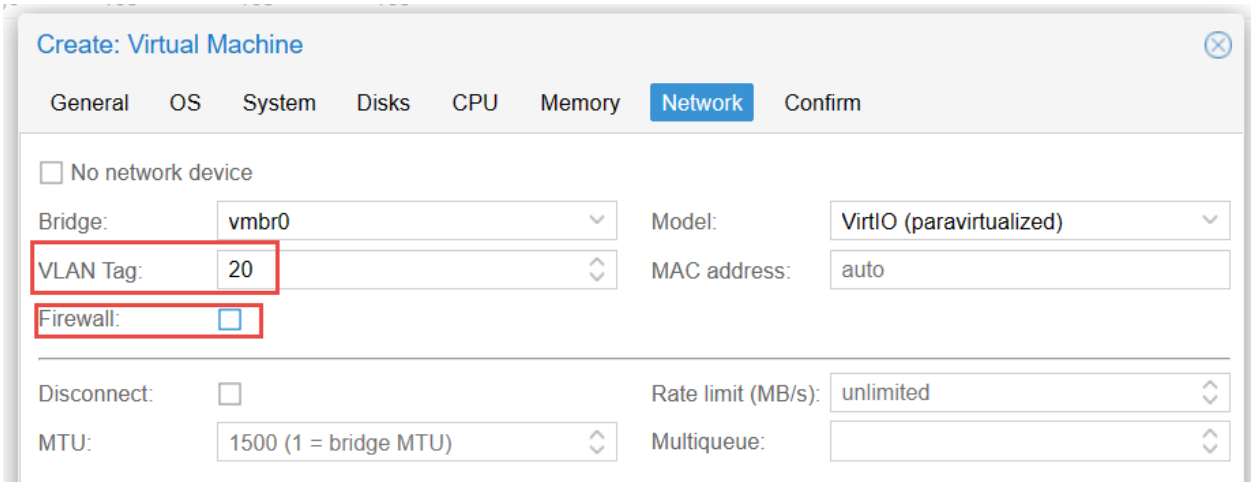



Create: Virtual Machine
General
OS
System
Disks
CPU
Memory
Network
Confirm
Graphic card:
Default
SCSI Controller:
VirtIO SCSI single
Machine:
Default (i440fx)
Qemu Agent:
Firmware
BIOS:
Default (SeaBIOS)
Add TPM:

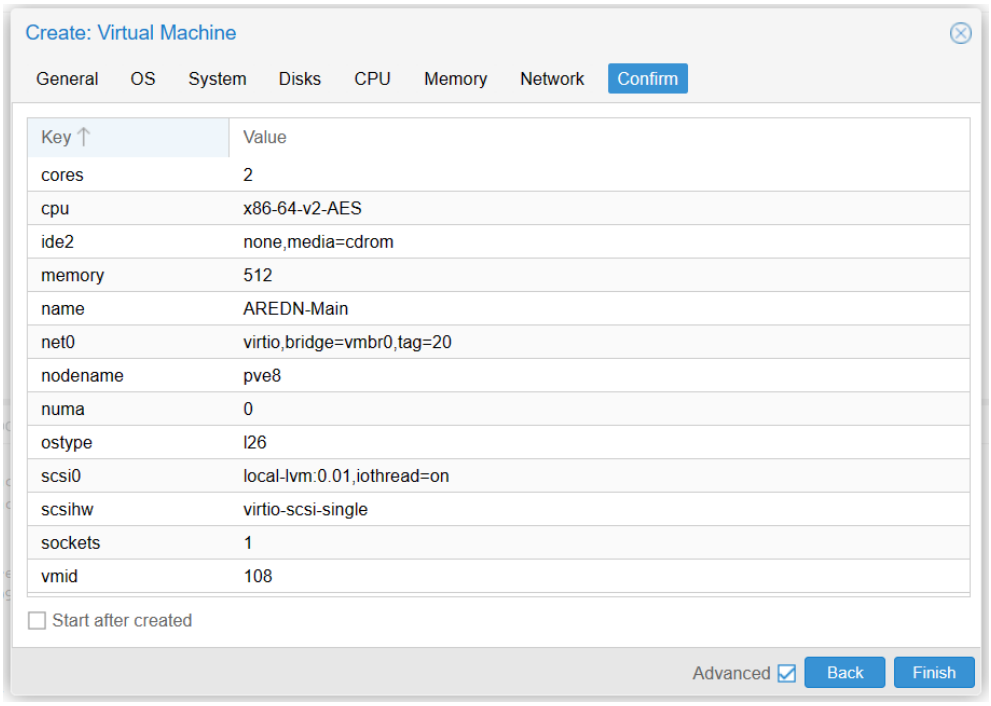

Create: Virtual Machine
General
OS
System
Disks
CPU
Memory
Network
Confirm
scsi0
Disk
Bandwidth
Bus/Device:
SCSI
0
Cache:
Default (No cache)
SCSI Controller:
VirtIO SCSI single
Discard:
Storage:
local-lvm
IO thread:
Disk size (GiB):
0.01
Format:
Raw disk image (raw)
SSD emulation:
Backup:
Read-only:
Skip replication:
Async IO:
Default (io_uring)


Create: Virtual Machine
General
OS
System
Disks
CPU
Memory
Network
Confirm
Sockets:
1
Type:
x86-64-v2-AES
Cores:
2
Total cores:
2
VCPUs:
2
CPU units:
100
CPU limit:
unlimited
Enable NUMA:
CPU Affinity:
All Cores
Extra CPU Flags:
Default
md-clear
Required to let the guest OS know if MDS is mitigated correctly
Default
pcid
Meltdown fix cost reduction on Westmere, Sandy-, and IvyBridge Intel CPUs
Default
spec-ctrl
Allows improved Spectre mitigation with Intel CPUs
Default
ssbd
Protection for "Speculative Store Bypass" for Intel models
Default
ibpb
Allows improved Spectre mitigation with AMD CPUs
Help
Advanced
Back
Next





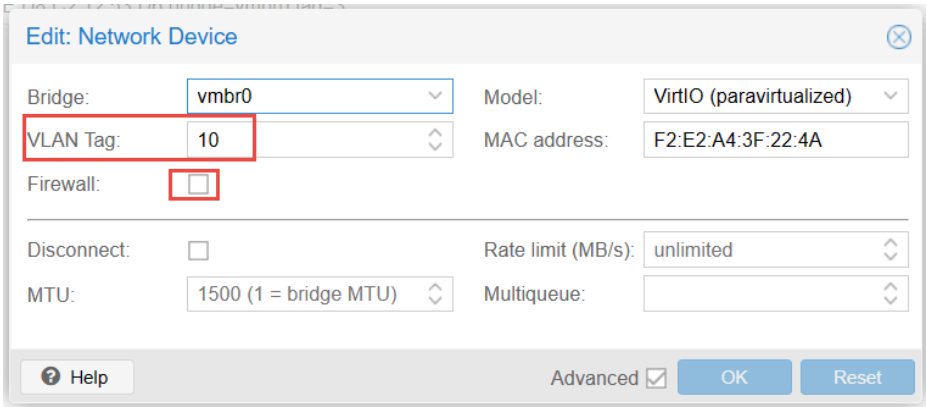
Add VLAN 20 (to prevent DHCP server interacting with your home network (and untick firewall):



Confirm the summary and wait till the VM is created

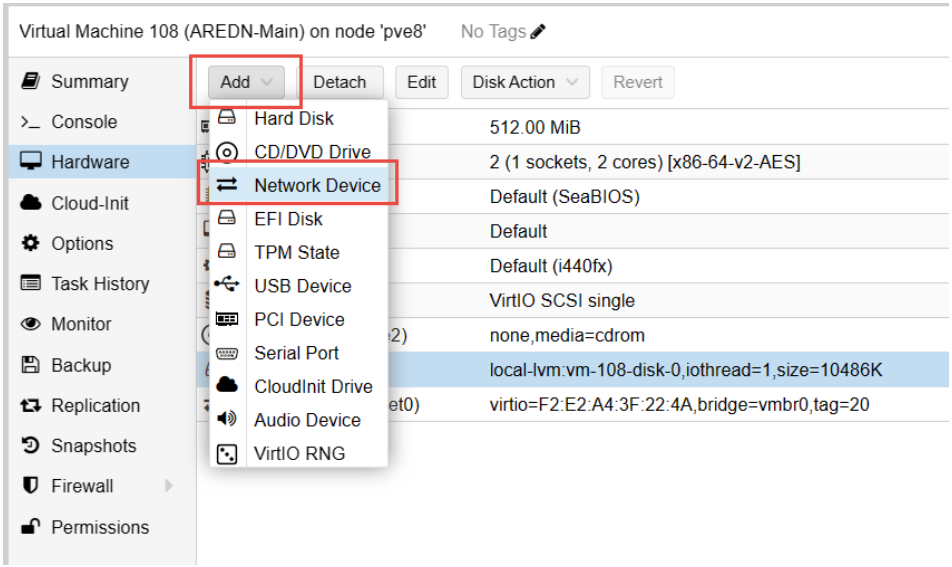
Configuring the AREDN VM

Change VLAN tag and remove firewall of Network device (net0) to create LAN port

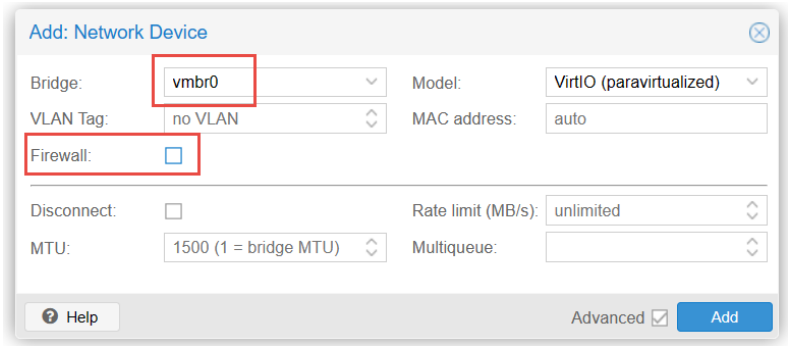


VLAN 10 (or any other above 10) is used to insulate the DHCP server of the AREDN node. If you forget it, you can shut down your whole home network when you start the server!

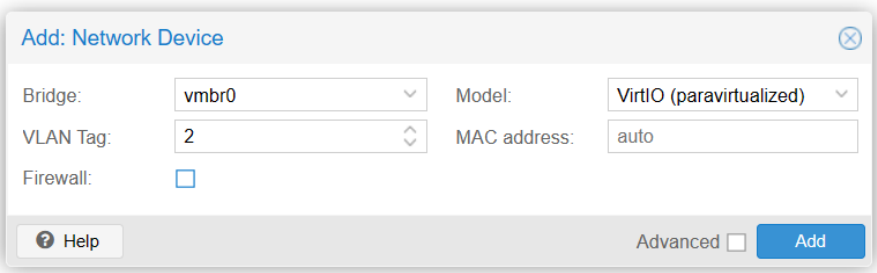
Click the Add button > Network Device



to create the WAN port



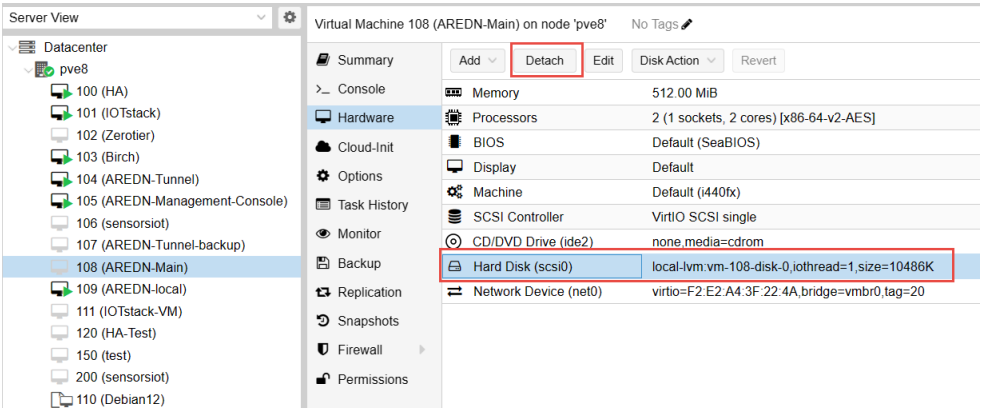
Add the DtD interface in VLAN2



The result should look like that:

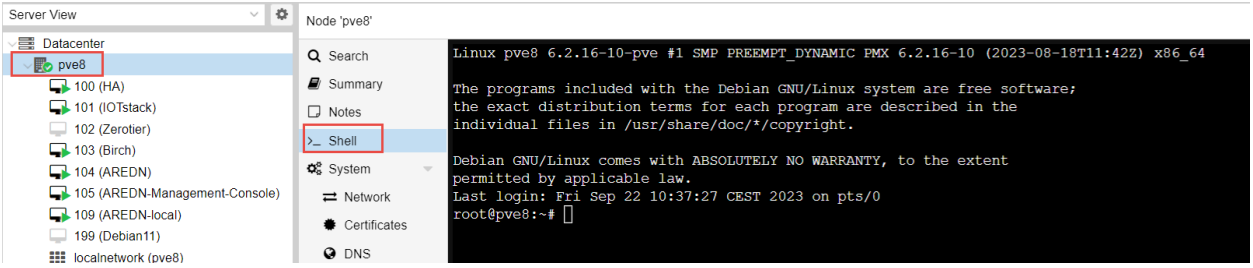
Network Device (net0)	virtio=BA:F4:D5:53:E9:96,bridge=vmbr0,tag=10
Network Device (net1)	virtio=36:16:35:EC:E9:85,bridge=vmbr0
Network Device (net2)	virtio=2E:81:B8:7A:FF:66,bridge=vmbr0,tag=2

And finally, detach and remove harddisk (scsi0)



Setting Up the AREDN Disk

Go to the shell of the server and download the image




```
wget -O aredn.img.gz
http://downloads.arednmesh.org/releases/3/23/3.23.12.0/targets/x86/64/aredn-
3.23.12.0-x86-64-generic-ext4-combined-efi.img.gz
```

Adjust the release if you want. Or upgrade your VM later using the AREDN VM

extract the AREDN img

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

rename the extracted img

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

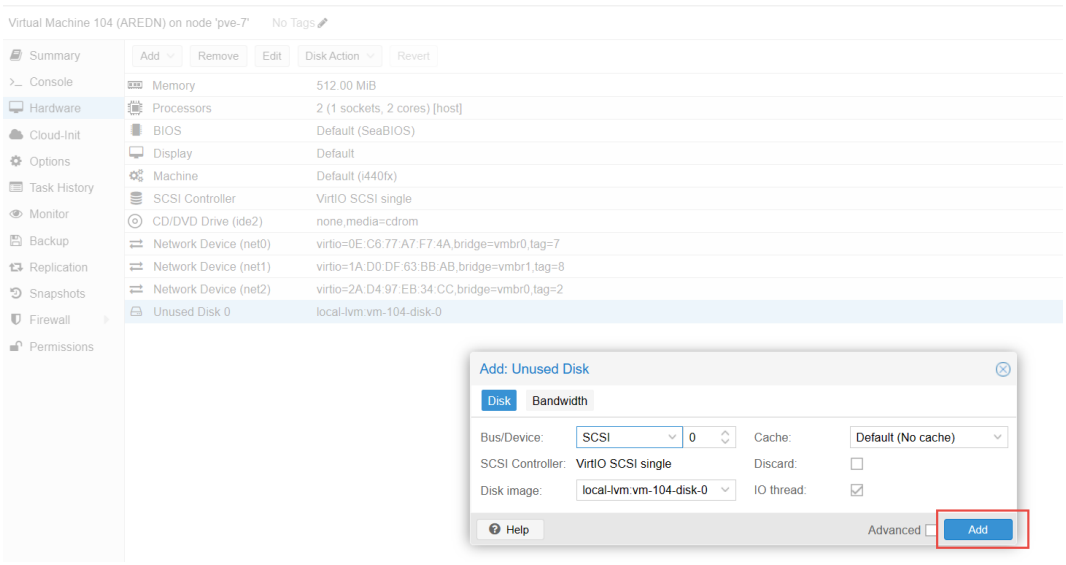
increase the raw disk to 512 MB

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

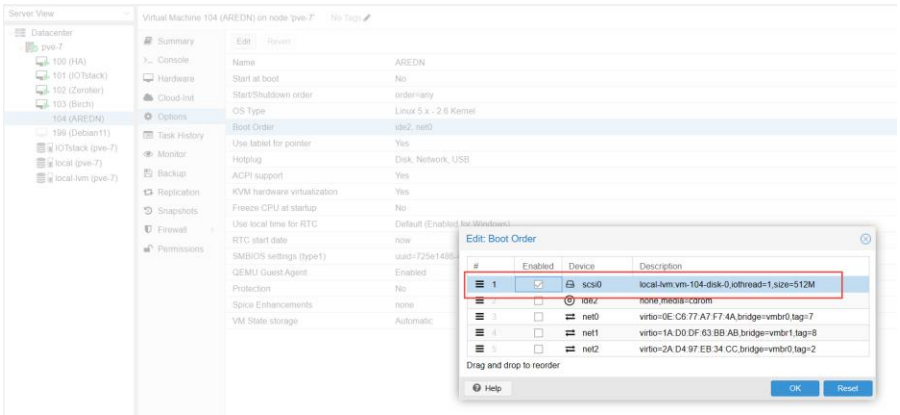
import the disk to the openwrt VM (replace 104 with your VM number)

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

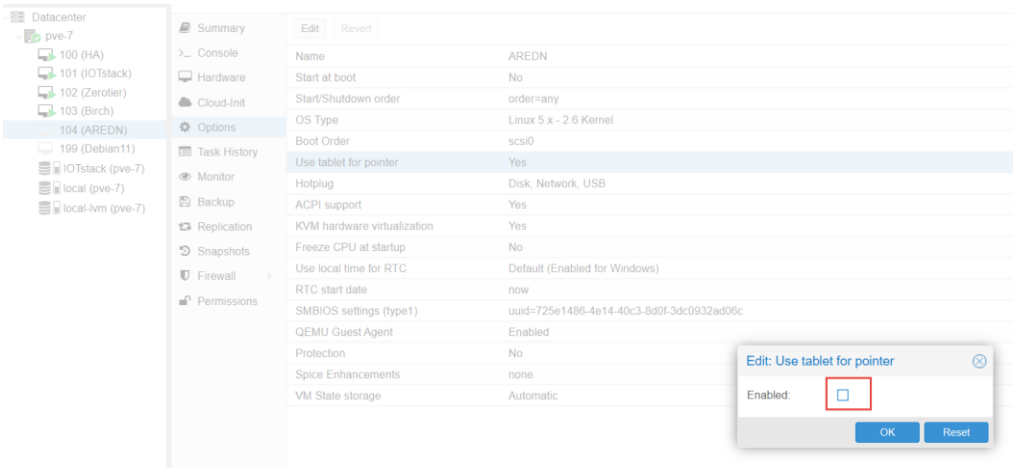
The disk is now visible as “Unused Disk 0”. Add the disk (double click on it) to your VM:



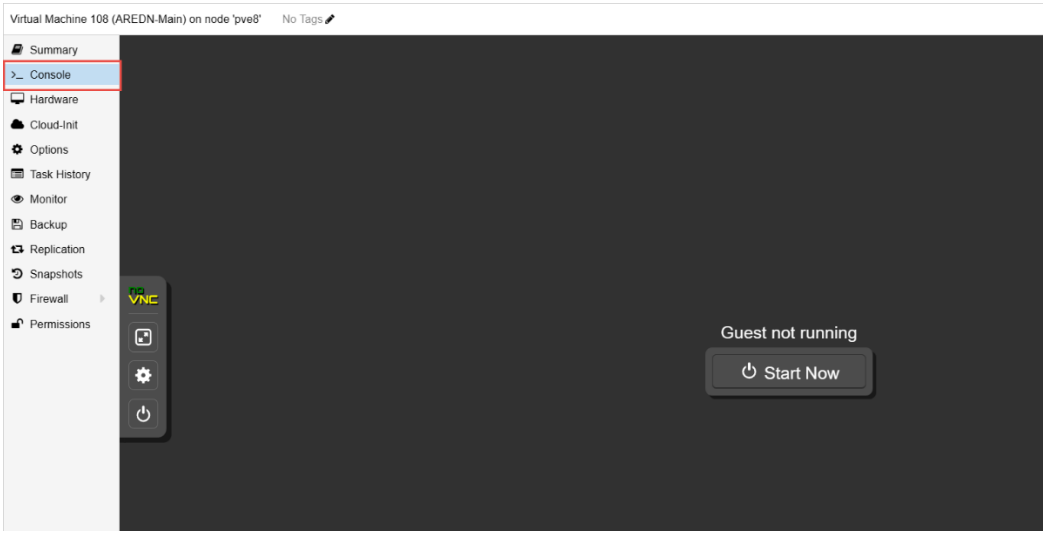
Change boot order:



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



Go to Console and press “Start Now”



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. Attention: It is a US keyboard (Y/Z)

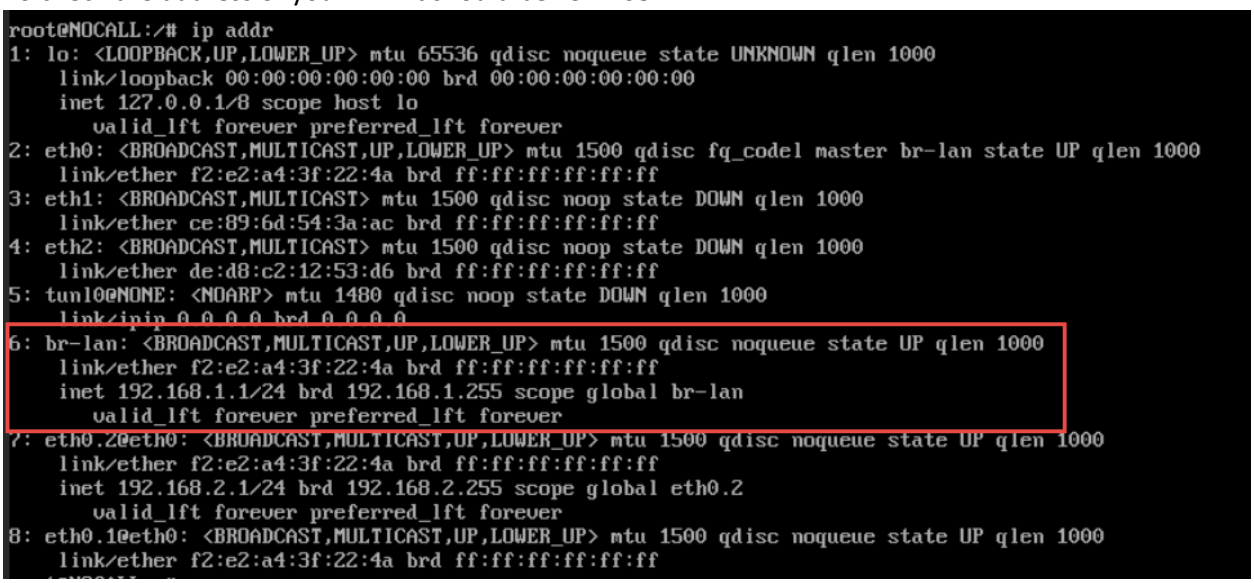
For the moment we are done with the AREDN node. Because we had to separate the LAN from our home network, we only can manage the node in the Proxmox terminal (it does not offer a browser). To solve the problem, we add a VM with plain vanilla Debian and connect it to VLAN20, too. Like that we have a private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

Of course, you can use any other VM that offers a browser and is connected to vmbr0/VLAN10.

Type

```
ip addr
```

To check the address of your VM. It should be 192.168.1.1

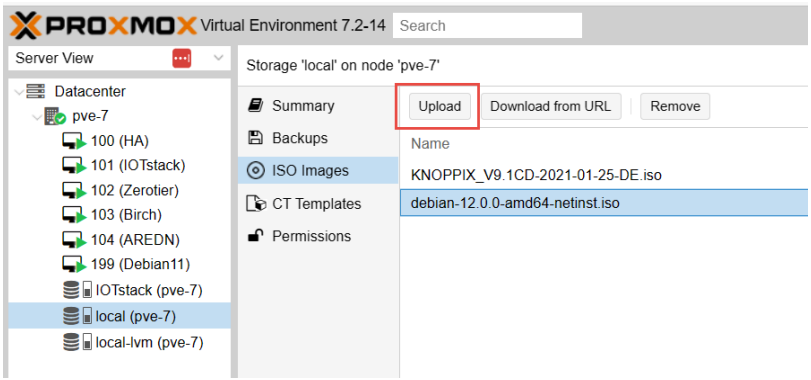


Create a Debian VM as a Management Console

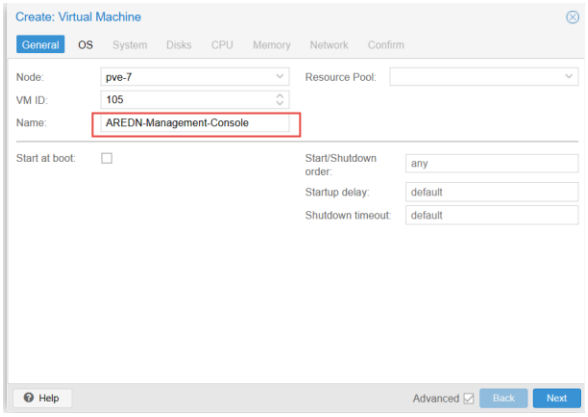
https://youtu.be/OUC7DMBfR3Y?si=y4r3edcwzzpd2JHJ

Download the Debian 11 iso Download

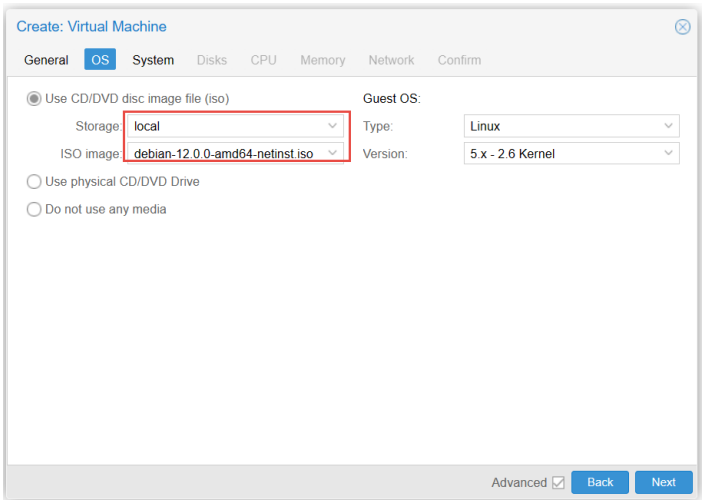
https://cdimage.debian.org/debian-cd/current/amd64/iso-cd/ (debian-12.1.0-amd64-netinst.iso)



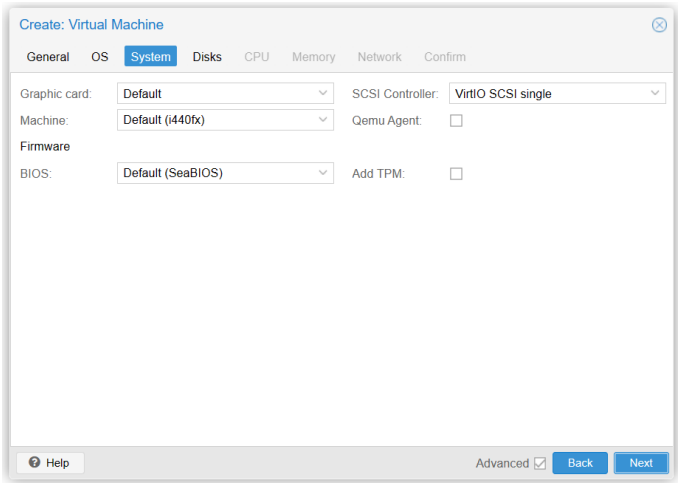
Create VM



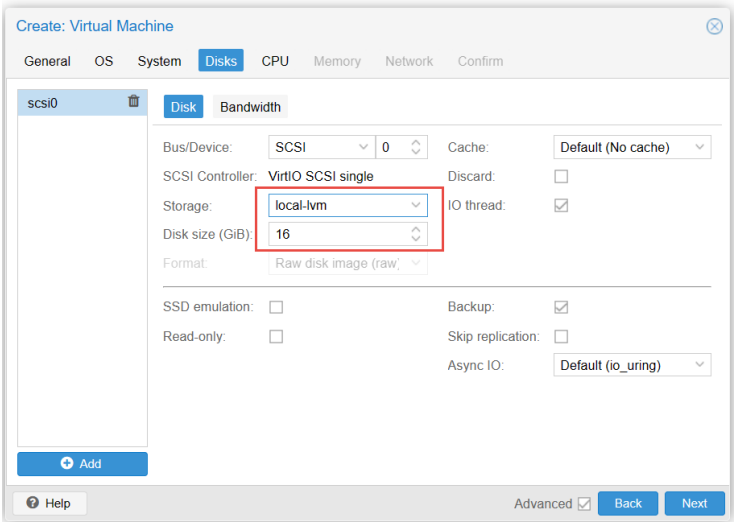
On the OS tab,



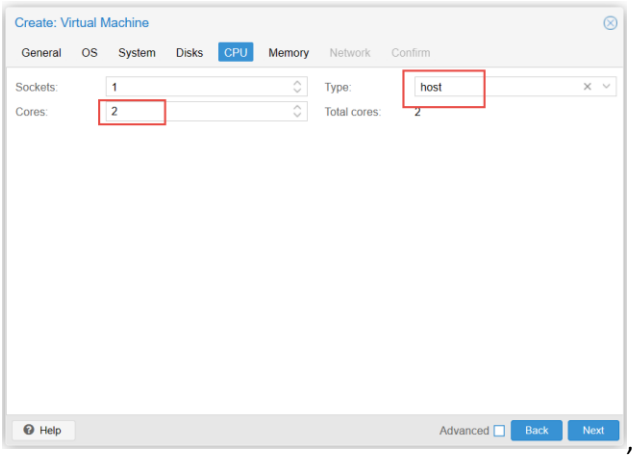
Leave the defaults on the System tab



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

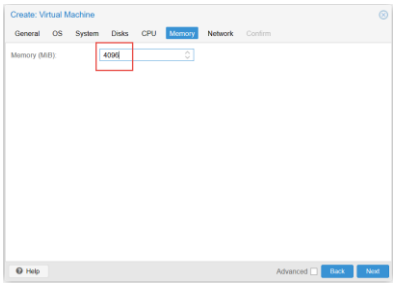


On the CPU tab



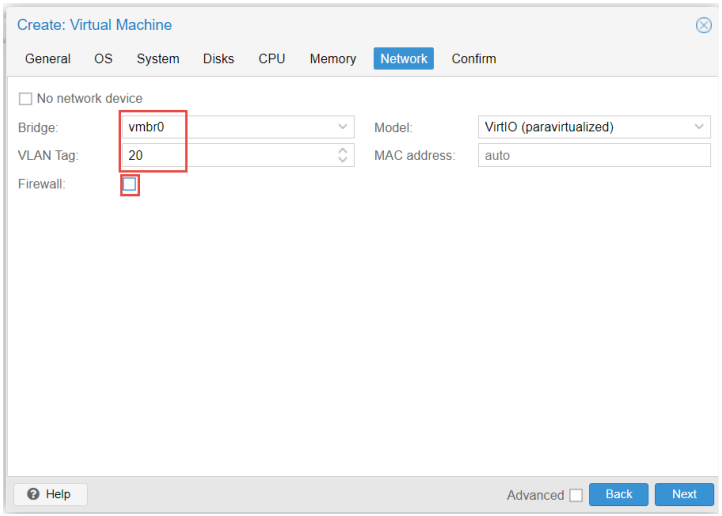
,

On the Memory tab



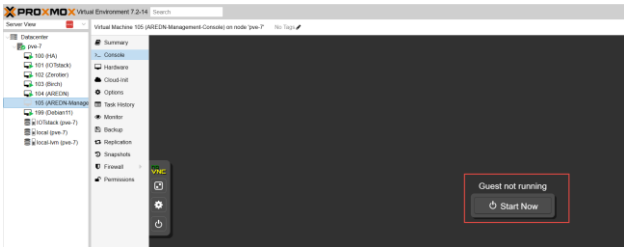
Network tab: Add the VLAN number of the LAN port of the AREDN server you want to connect to and untick the firewall.

Edit: Do not add the VLAN right now. Add it when you are finished with the installation and want to connect to an AREDN node. Otherwise you will not be able to continue.

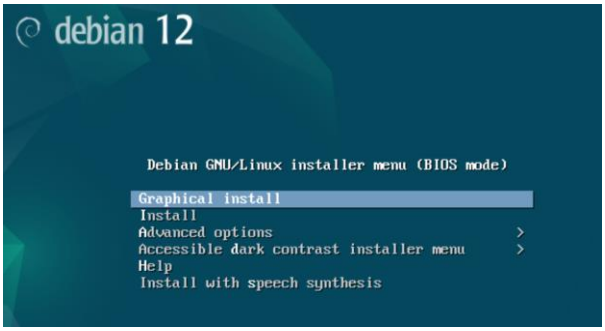


Verify the summary and click Finish

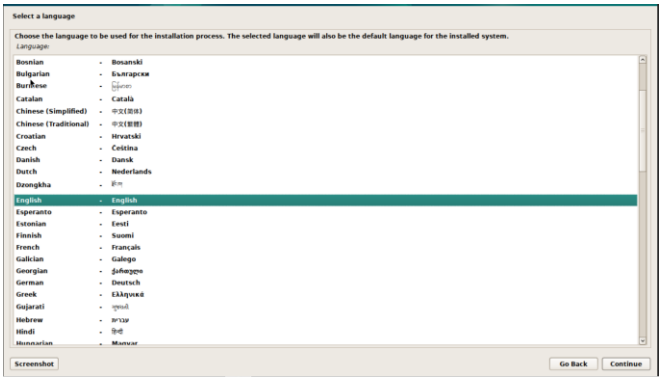
Start the VM



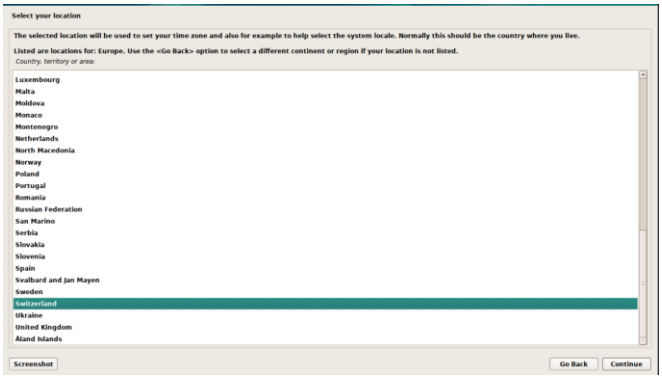
Graphical Install



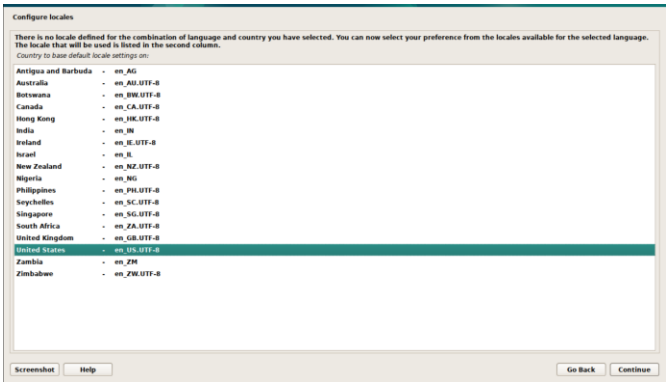
Select a language



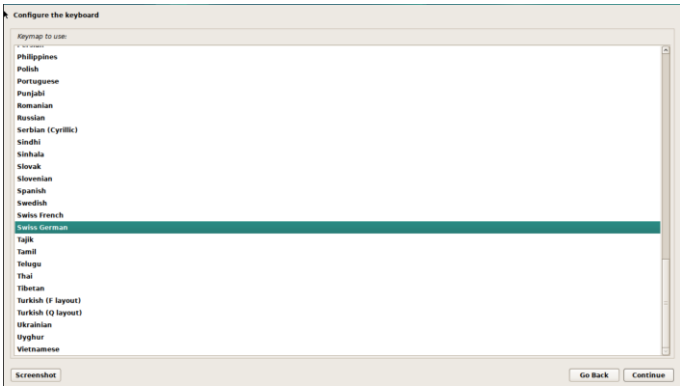
Select a Location (other →Europe→Switzerland)



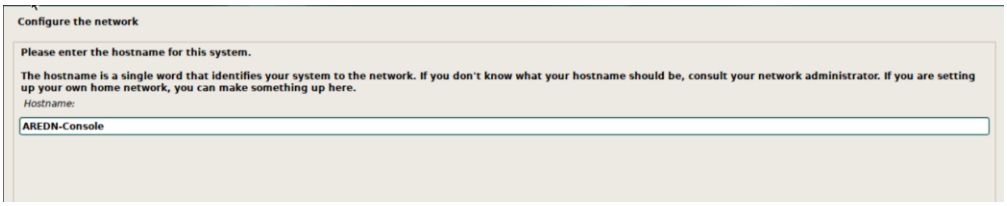
Select locales (can be adapted later)



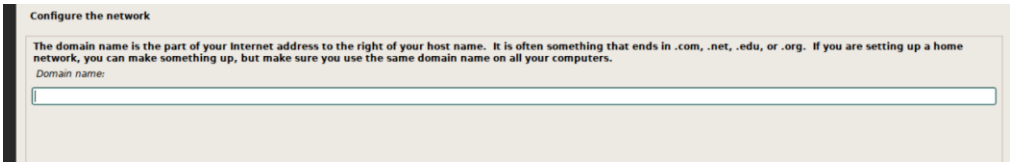
Select a keyboard layout (can be changed later)



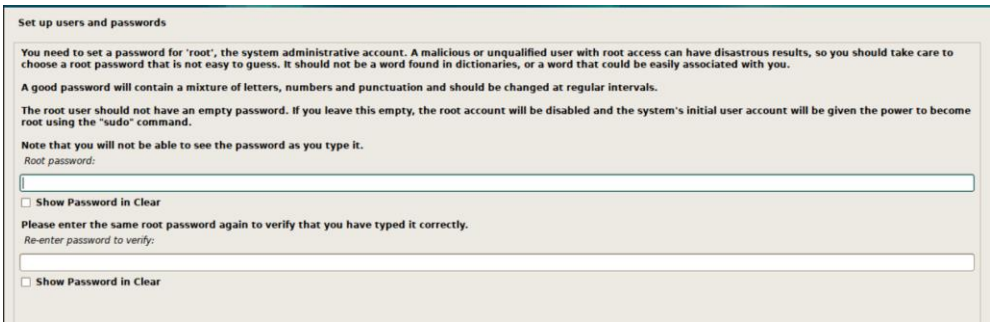
Enter a hostname for the VM



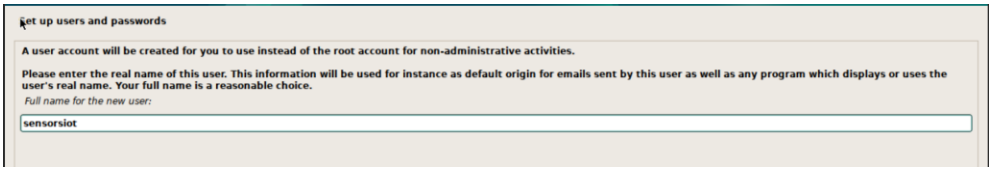
Leave domain name empty



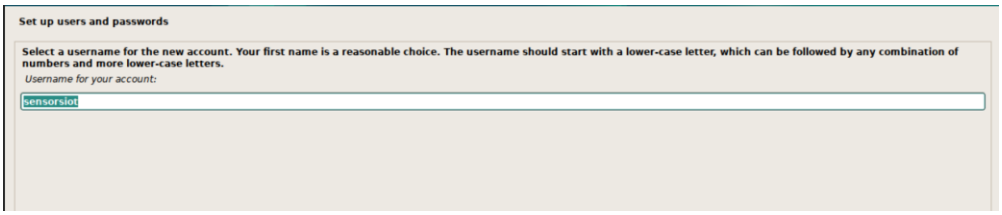
Leave root password empty



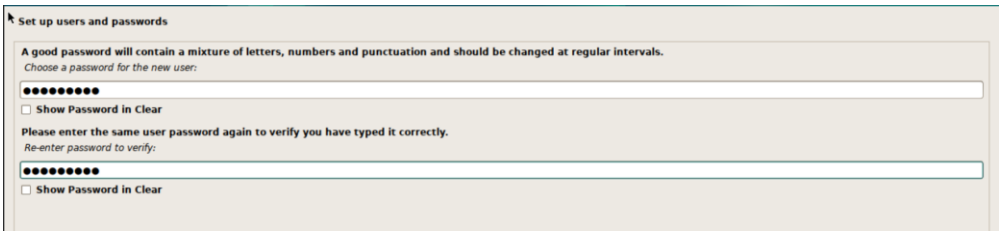
Enter the full name for the new user



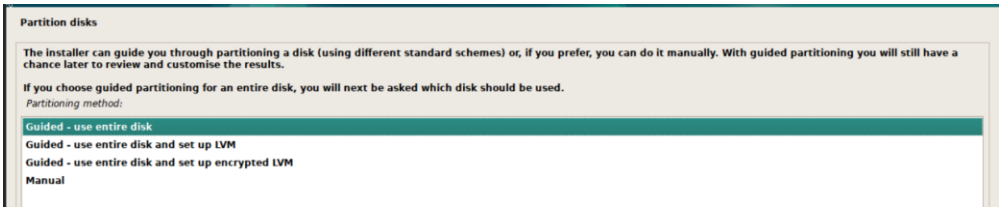
Enter the username for the new user

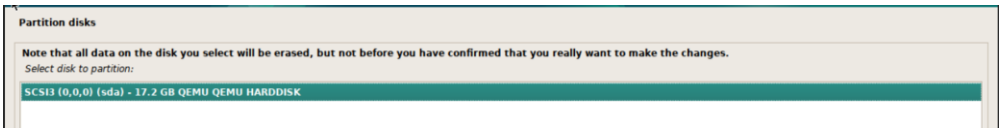


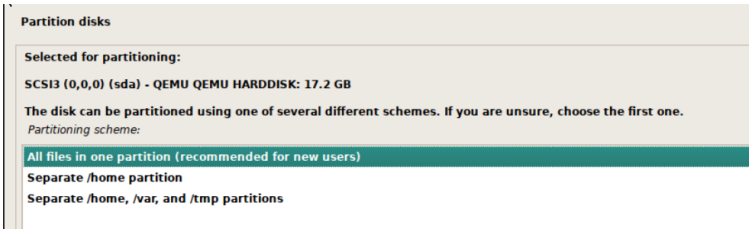
Enter and confirm a password for the new user



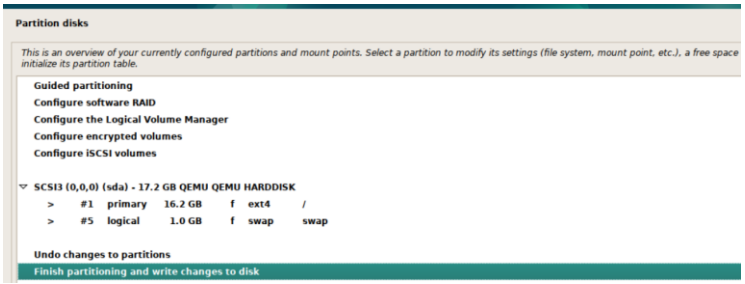
Select Disk



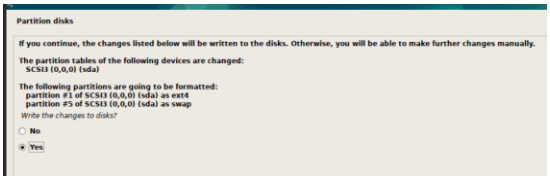




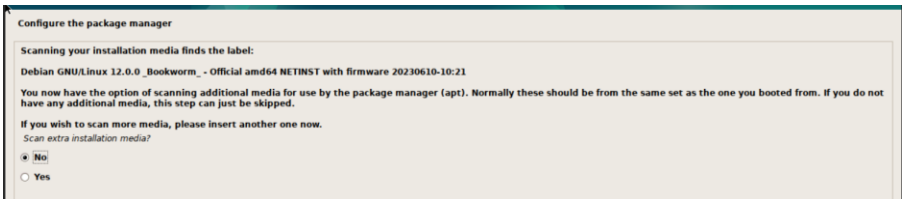
Check and click Continue

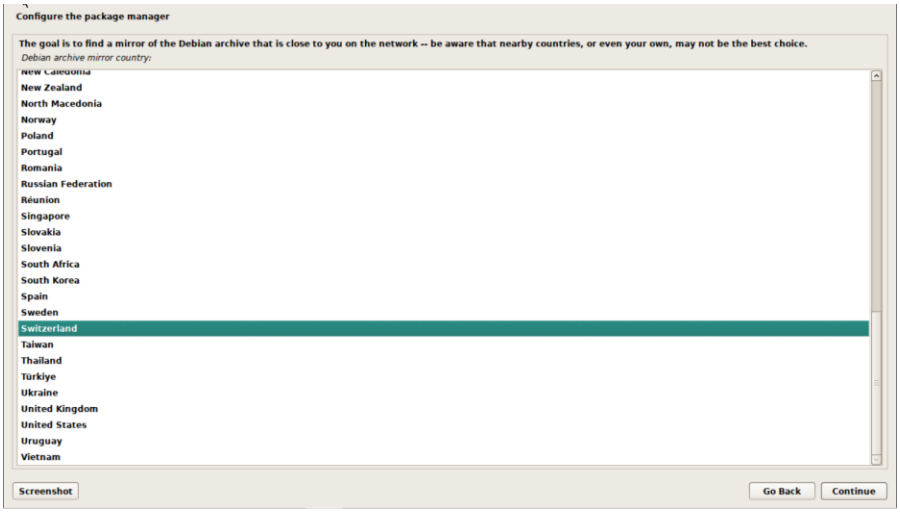


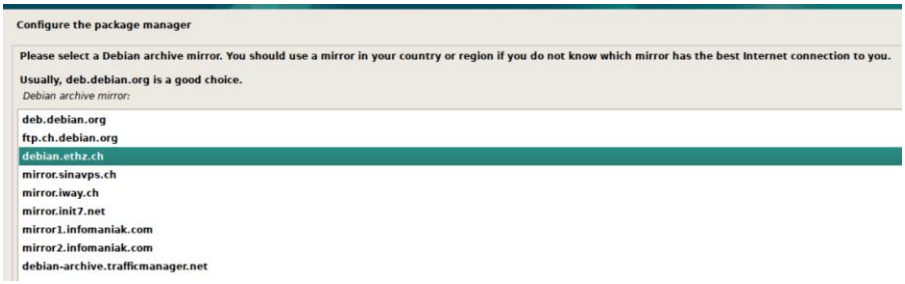
Select Yes to confirm writing the changes



Wait for Debian to copy and install files

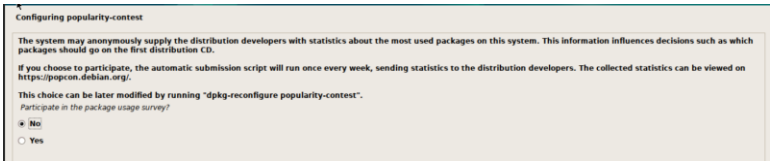




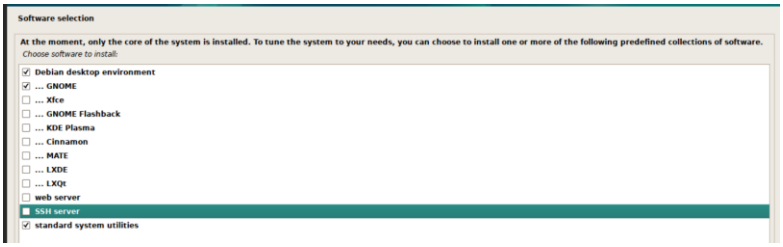


Wait

Participating in package survey



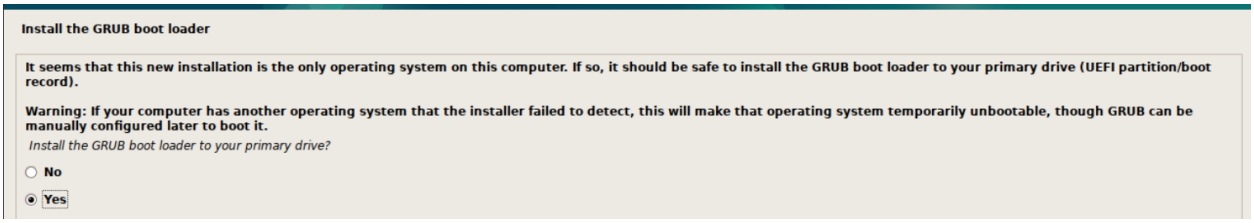
Select the software to install



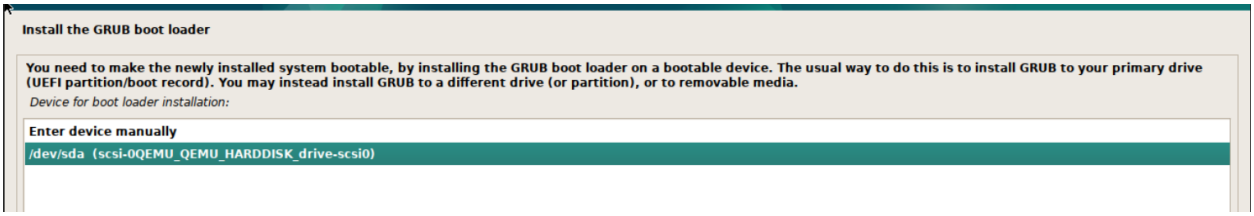
Select (tick) “SSH server”

Wait

Select Yes to install GRUB

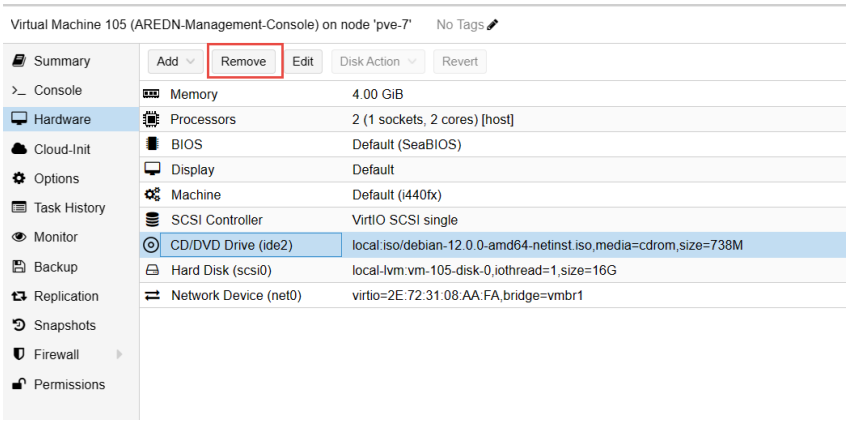


Select drive for the boot loader

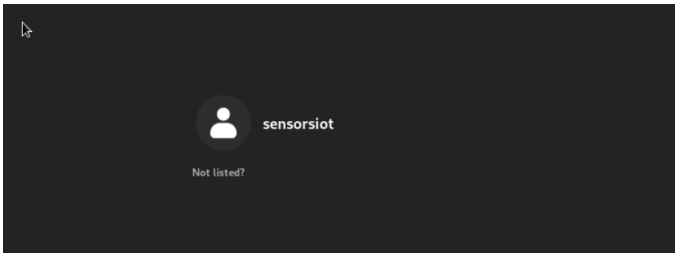


Reboot

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 on the same VLAN because your Management Console is connected via the vmbr0/10 interface.

Create a Management Console in an LXC Container (work in progress, not finished)

Create a Container using a Debian image

Create a new user

```
apt install sudo curl -y
```

```
adduser aredn
```

```
usermod -aG sudo aredn
exit
```

Logout and login with new user

Check if it worked:

```
sudo apt update
```

```
sudo apt install xrdp -y
```

```
sudo echo xfce4-session >~/.xsession


sudo nano /etc/xrdp/startwm.sh
```

add

```
startxfce4
```

at the end of the file

Configure the AREDN Tunnel Server

Because our Management Console VM is connected to our AREDN VM via vmbr0/VLAN10 it should get an IP.

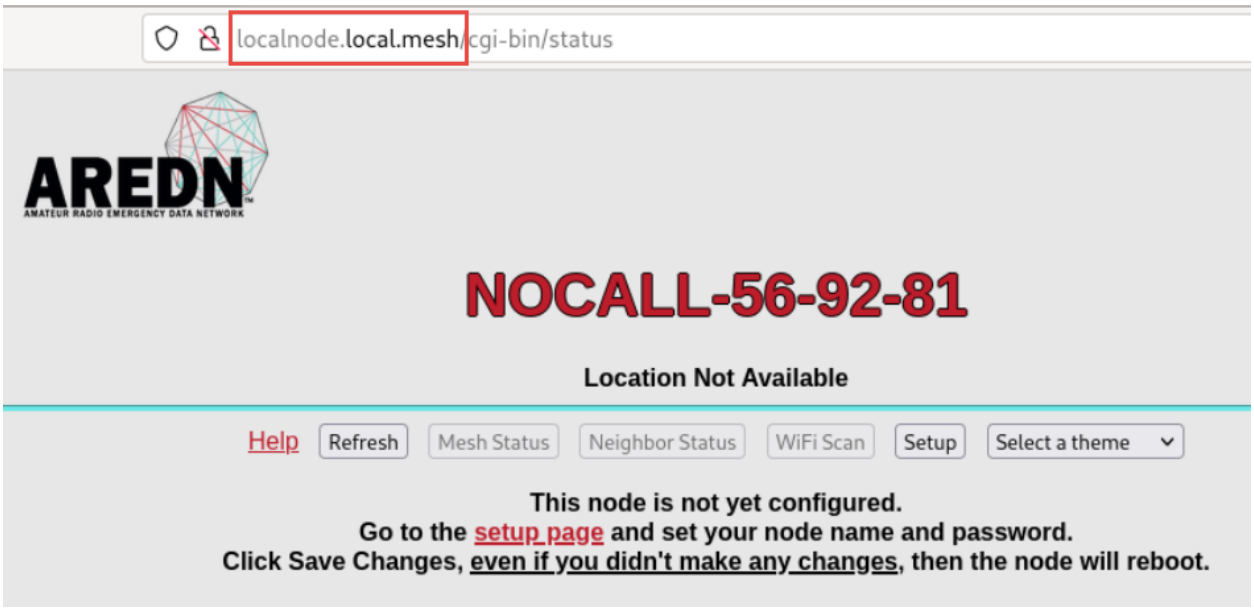
```
sensorsiot@AREDN-Console:~$ ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
       valid_lft forever preferred_lft forever
2: ens18: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 2e:72:31:08:aa:fa brd ff:ff:ff:ff:ff:ff
    altname enp0s18
    inet 192.168.1.20/24 brd 192.168.1.255 scope global dynamic ens18
       valid_lft 41302sec preferred_lft 41302sec
    inet6 fe80::2c72:31ff:fe08:aafa/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
```

To check the connection, you can ping the AREDN VM:

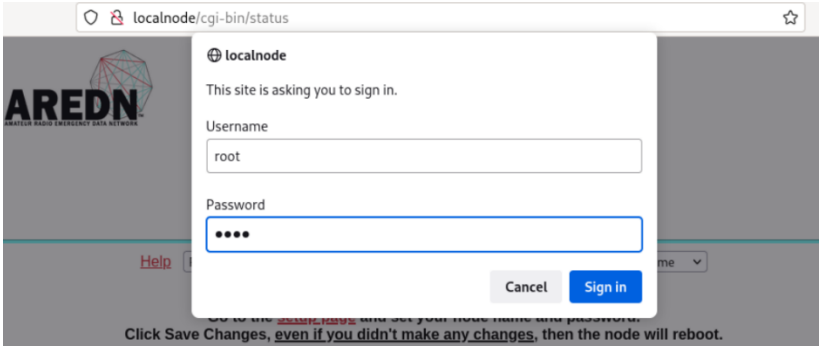
```
ping localnode.local.mesh
```

Initial setup of the AREDN router

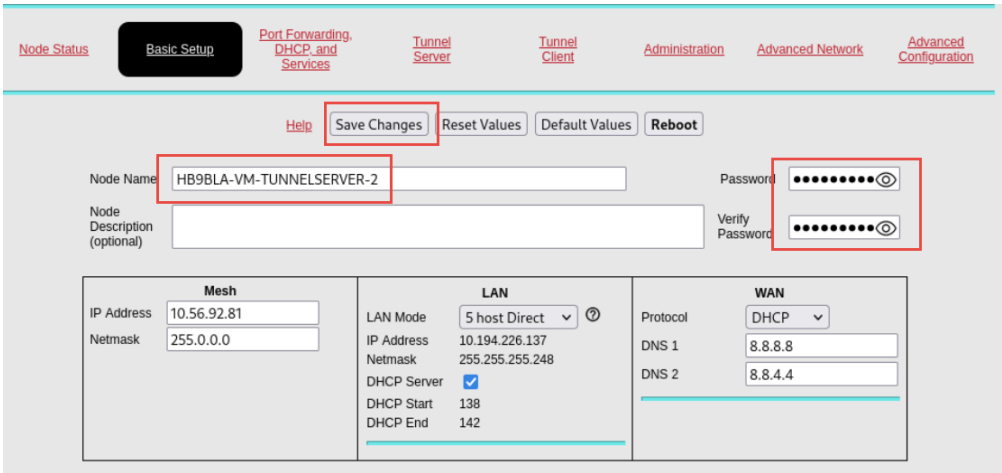
Open the browser and call localnode.local.mesh



Press setup (password: “hsmm” or your password given above)



Give it a name and a password, hit “Save changes” and “reboot”



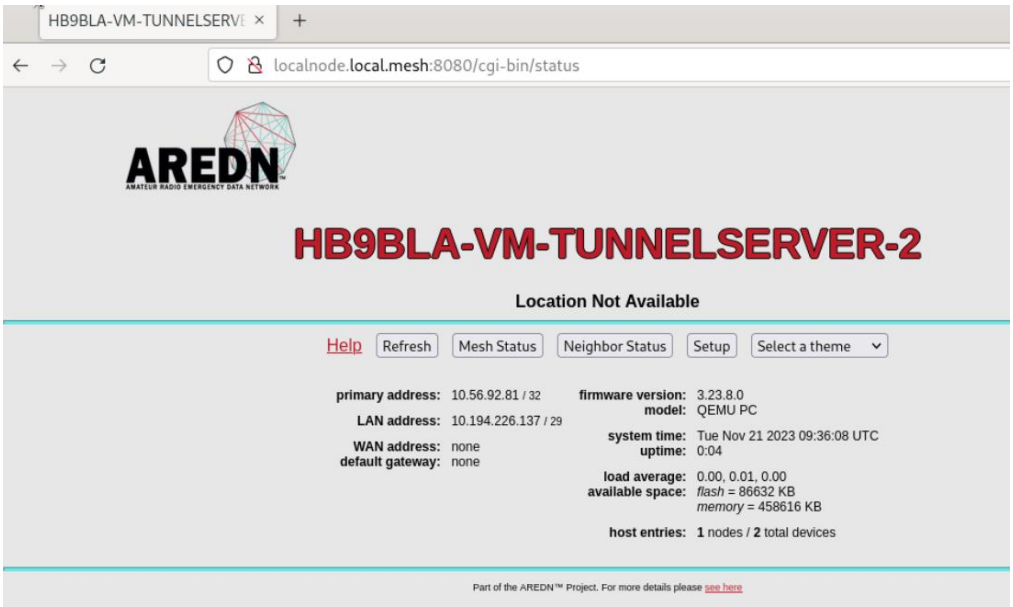
After reboot, your AREDN VM should have an 10.x.x.x address

If the Management Console does not connect to the AREDN VM, it still has its old address.

So type these two commands to get a new address from the AREDN VM:

```
sudo dhclient -r
sudo dhclient
```

From now on, you can manage your AREDN node from the Management Console



Connection to the configured Tunnel Server (incl. ssh)

Connect again to your AREDN server (unfortunately, Proxmox does not allow copy-paste):

sensorsiot@AREDN-Console: ~

```
sensorsiot@AREDN-Console:~$ ssh -l root -p 2222 localnode
The authenticity of host '[localnode]:2222 ([10.228.113.121]:2222)' can't be est
ablished.
ECDSA key fingerprint is SHA256:GzuLrrLzL805914bcGbMPe9KsDg0+hJnXiwd3ZWpYy0.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '[localnode]:2222' (ECDSA) to the list of known hosts
.
root@localnode's password: █
```

Press yes, and enter the password **hsmm**

If you get a “man in the middle” warning, key in the suggested command:

```
sensorsiot@AREDN-Console:~$ ssh -l root -p 2222 localnode
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@    WARNING: REMOTE HOST IDENTIFICATION HAS CHANGED!     @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
IT IS POSSIBLE THAT SOMEONE IS DOING SOMETHING NASTY!
Someone could be eavesdropping on you right now (man-in-the-middle attack)!
It is also possible that a host key has just been changed.
The fingerprint for the ECDSA key sent by the remote host is
SHA256:in9jB/hOPyHlR26zc3jBTufKKWf3jFfWxIgxPdzV2qs.
Please contact your system administrator.
Add correct host key in /home/sensorsiot/.ssh/known_hosts to get rid of this message.
Offending ECDSA key in /home/sensorsiot/.ssh/known_hosts:3
  remove with:
  ssh-keygen -f "/home/sensorsiot/.ssh/known_hosts" -R "[localnode]:2222"
Password authentication is disabled to avoid man-in-the-middle attacks.
Keyboard-interactive authentication is disabled to avoid man-in-the-middle attacks.
UpdateHostkeys is disabled because the host key is not trusted.
root@localnode: Permission denied (publickey,password).
sensorsiot@AREDN-Console:~$
```

This is the correct answer:

```
sensorsiot@AREDN-Console:~$ ssh-keygen -f "/home/sensorsiot/.ssh/known_hosts" -R "[localnode]:2222"
# Host [localnode]:2222 found: line 3
/home/sensorsiot/.ssh/known_hosts updated.
Original contents retained as /home/sensorsiot/.ssh/known_hosts.old
sensorsiot@AREDN-Console:~$ █
```

Now you can try to ssh into your server as before

root@HB9BLA-VM-TUNNELSERVER: ~

```
Warning: Permanently added '[localnode]:2222' (ECDSA) to the list of known hosts
.
root@localnode's password:
Permission denied, please try again.
root@localnode's password:


BusyBox v1.35.0 (2023-04-27 20:28:15 UTC) built-in shell (ash)

          _____  ______ _____  _   _
     /\   |  __ \|  ____|  __ \| \ | |TM
    /  \  | |__) | |__  | |  | |  \| |
   / /\ \ |  _  /|  __| | |  | | . ` |
  / ____ \| | \ \| |____| |__| | |\  |
 /_/    \_\_|  \_\______|_____/|_| \_|
 AMATEUR  RADIO EMERGENCY DATA NETWORK
--------------------------------------------
 1) Research AREDN and choose a supported device
 2) Download and install AREDN firmware
 3) Deploy and enjoy the mesh
--------------------------------------------
  3.23.8.0, r20134-5f15225c1e
 -------------------------------------------
root@HB9BLA-VM-TUNNELSERVER:~# █
```

Network mapping inside the AREDN VM

In the end we would like the following mapping:

⇄ Network Device (net0)	virtio=F2:E2:A4:3F:22:4A,bridge=vmbr0,tag=10	eth0	LAN port (ports 2-4 on hap)
⇄ Network Device (net1)	virtio=CE:89:6D:54:3A:AC,bridge=vmbr0	eth1	LAN port (port 1 on hap
⇄ Network Device (net2)	virtio=DE:D8:C2:12:53:D6,bridge=vmbr0,tag=3	eth2	LAN port (port 5 on hap

The mapping of the nets has to be done now in the terminal of the Management 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 LAN 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.

The WAN port has to be connected to the internet. Vmbr0 is connected to the RJ45 connector of your server. 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 '10.60.118.4'
        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.

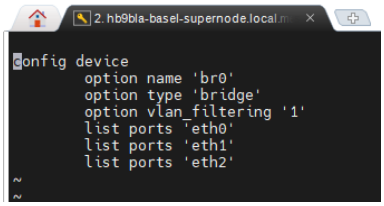
Press the “escape” button and “:wq” to save your changes.

Reboot.

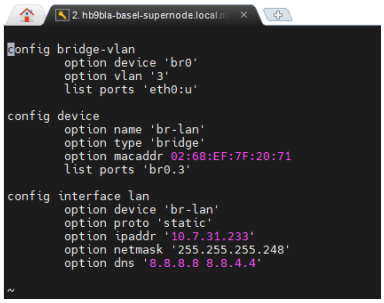
Make the configuration surviving changes and reboots

Create 4 new files and copy the respective part of the configuration files into these files:

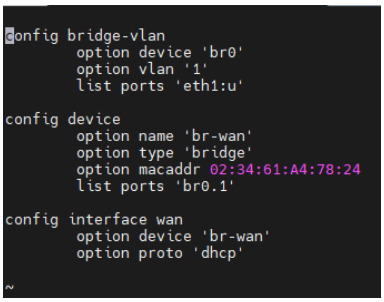
```
vi /etc/aredn_include/bridge.network.user
```



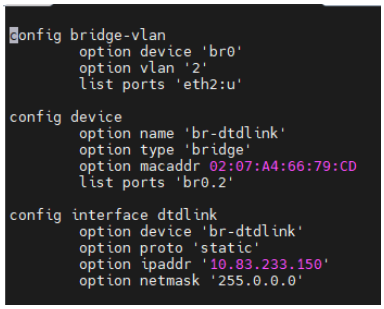

```
vi /etc/aredn_include/lan.network.user
```




```
vi /etc/aredn_include/wan.network.user
```




```
vi /etc/aredn_include/dtdlink.network.user
```



For a test you can type:

```
/usr/local/bin/node-setup -a mesh
```

And you should see the same config file as before. Also all tunnels should be up again.

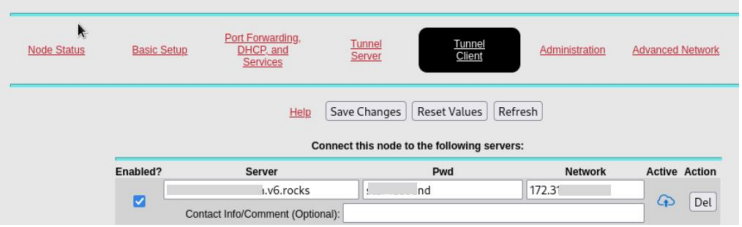
Keep in mind that, if you change the network connection of your tunnel server in Proxmox, you have to delete all four files, reboot to create the right config, and recreate the four files with the copied content.

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

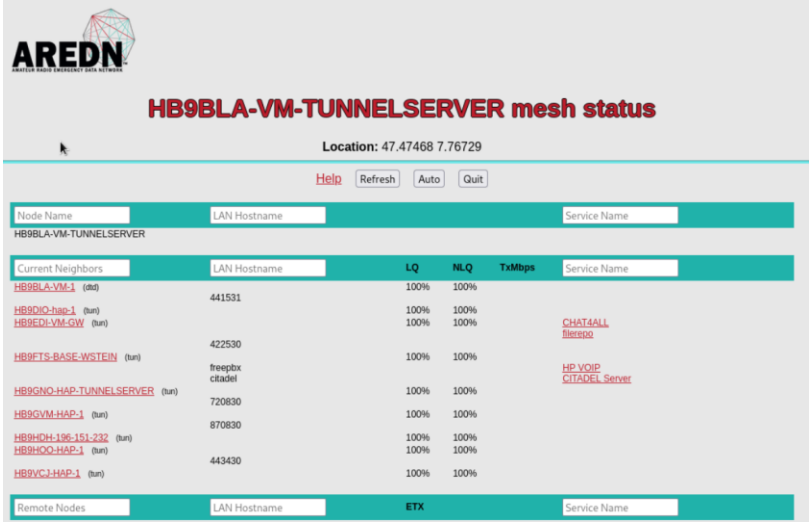
Give it the name: Callsign-VM-TUNNELSERVER (if not done before).

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.

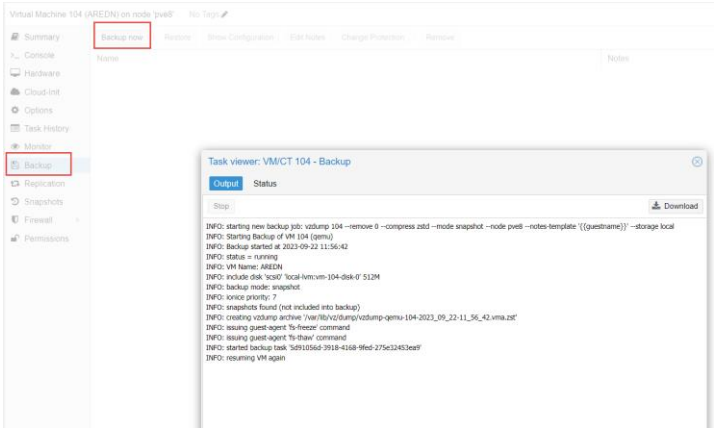


Finally, your Server should be connected to the SwissDigitalNet. Congratulations!



Backup of the VM

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

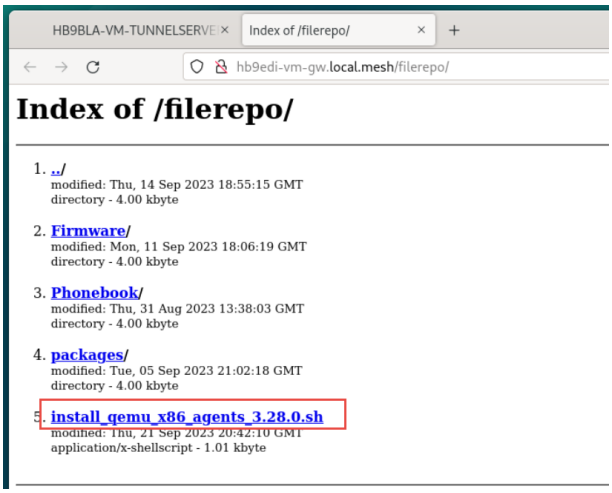
Install QUEMU Agent

This step is needed to control (e.g. shut down) the AREDN VM from Proxmox

You have to be connected to the SwissDigitalNet for this task

In your Management Console, you go to mesh status and select HB9EDI's filerepo:

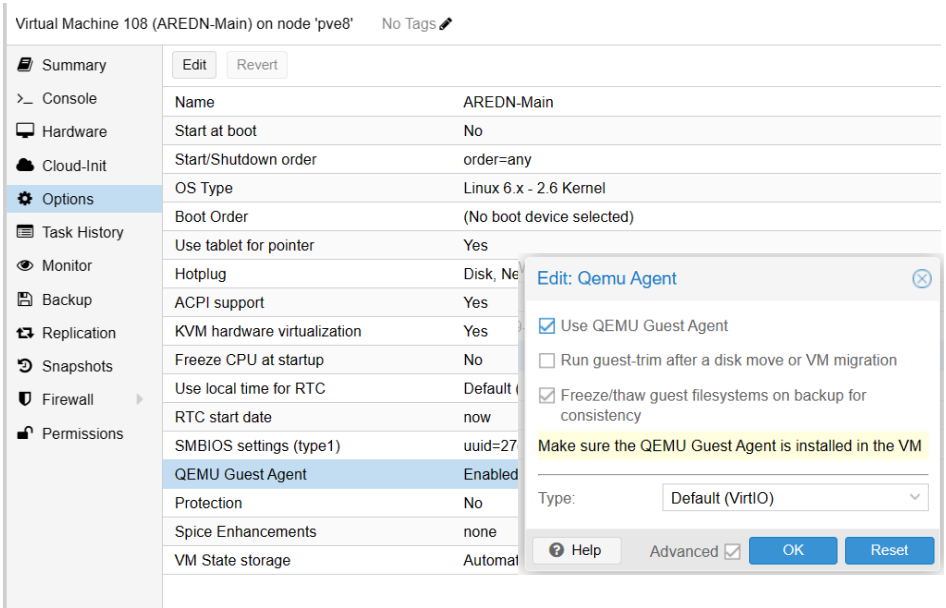
And copy this link (right mouse click):



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 and reboot the machine



Now your Tunnel Server can be controlled by Proxmox.

Without connection to the SwissDigitalNet

Download and install these packets from

```
https://downloads.openwrt.org/releases/23.05.0/packages/x86_64/
```

```
/22.03.5/packages/x86_64/packages/libattr_2.5.1-1_x86_64.ipk
/22.03.5/packages/x86_64/packages/libffi_3.4.2-2_x86_64.ipk
/22.03.5/packages/x86_64/base/libpcre_8.45-3_x86_64.ipk
/22.03.5/packages/x86_64/packages/glib2_2.70.5-4_x86_64.ipk
/22.03.5/packages/x86_64/packages/virtio-console-helper_6.2.0-2_x86_64.ipk
22.03.5/packages/x86_64/packages/libevdev_1.13.0-1_x86_64.ipk
/22.03.5/packages/x86_64/packages/libudev-zero_1.0.1-1_x86_64.ipk
/22.03.5/packages/x86_64/packages/libstdcpp6_11.2.0-4_x86_64.ipk
/22.03.5/packages/x86_64/packages/qemu-ga_6.2.0-2_x86_64.ipk
```

If you do not find libstdcpp6_11.2.0-4_x86_64.ipk, download and install it from

```
https://github.com/dhamstack/AREDNstack/blob/main/Firmware%20Repo/Proxmox/libstdcpp6_11.2.0-4_x86_64.ipk
```

You can integrate all files into one file with the name install_qemu_x86_agents_3.28.0.sh.

```
#!/bin/ash
opkg install http://YourServer/packages/x86_64/packages/libattr_2.5.1-1_x86_64.ipk
opkg install http://YourServer/packages/x86_64/packages/libffi_3.4.2-2_x86_64.ipk
opkg install http://YourServer/packages/x86_64/base/libpcre_8.45-3_x86_64.ipk
opkg install http://YourServer/packages/x86_64/packages/glib2_2.70.5-4_x86_64.ipk

opkg install http://YourServer/packages/x86_64/packages/virtio-console-helper_6.2.0-
2_x86_64.ipk

opkg install http://YourServer/packages/x86_64/packages/libevdev_1.13.0-1_x86_64.ipk
opkg install http://YourServer/packages/x86_64/packages/libudev-zero_1.0.1-1_x86_64.ipk
opkg install http://YourServer/packages/x86_64/libstdcpp6_11.2.0-4_x86_64.ipk
opkg install http://YourServer/packages/x86_64/packages/qemu-ga_6.2.0-2_x86_64.ipk
```

Upgrade Tunnelserver

After upgrade, you will lose the network configuration. So it has to be added.

Ssh into the node:

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

Go on with editing the network config:

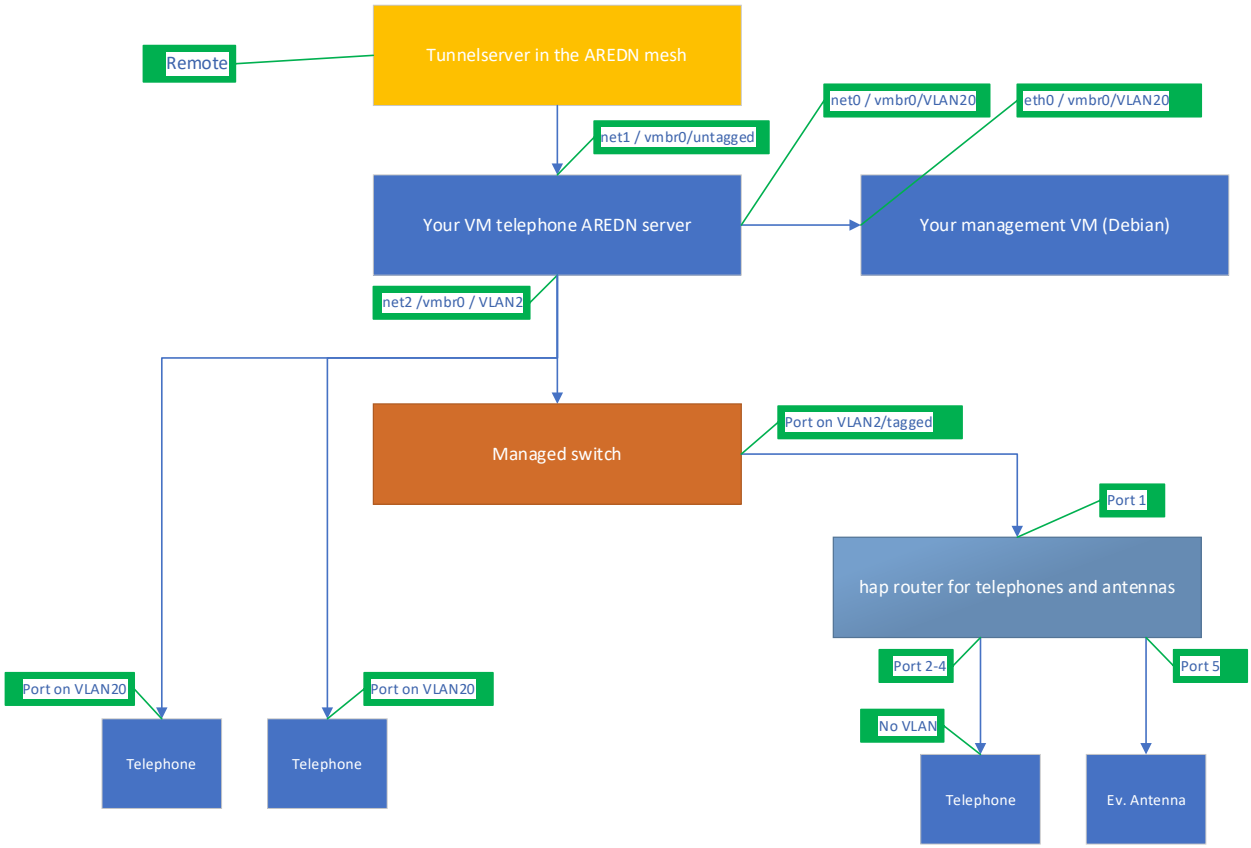
```
vi /etc/config/network
```

Instructions here: Configure the AREDN Tunnel Server

Make sure you still have the same IP address assigned (because of port forewarding)

AREDN Virtual Machine as a Telephone Server

Overview



Setup

We create 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.

Do not copy the VM of the Tunnel Server.

The basis setup has to be done the same way as for the Tunnel Server. The main difference is that we chose VLAN10 as the management connection:

⇄ Network Device (net0)	virtio=AA:88:4A:2F:F5:D1,bridge=vmbr0,tag=20
⇄ Network Device (net1)	virtio=0A:DC:B3:77:C5:08,bridge=vmbr0
⇄ Network Device (net2)	virtio=3A:55:E2:13:D0:C1,bridge=vmbr0,tag=2

And we have to change the network of the Management Console, too:

Memory	2.00 GiB
Processors	2 (1 sockets, 2 cores) [host]
BIOS	Default (SeaBIOS)
Display	Default
Machine	Default (i440fx)
SCSI Controller	VirtIO SCSI single
Hard Disk (scsi0)	local-lvm:vm-105-disk-0,iothread=1,size=16G
Network Device (net0)	virtio=2E:72:31:08:AA:FA,bridge=vmbr0,tag=20

Do not assign the same VLAN to two AREDN management connections (you will get two concurrent DHCP servers)!

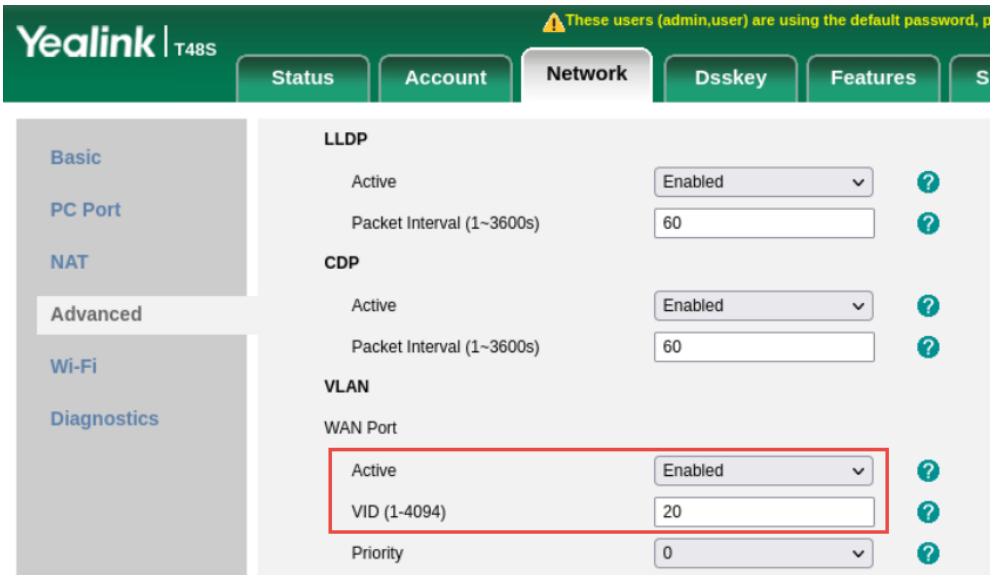
Now we should be able to manage the Telephone Server from the Management Console.

Maybe you have to reassign a new IP to the Management Console:

```
sudo dhclient -r
sudo dhclient
```

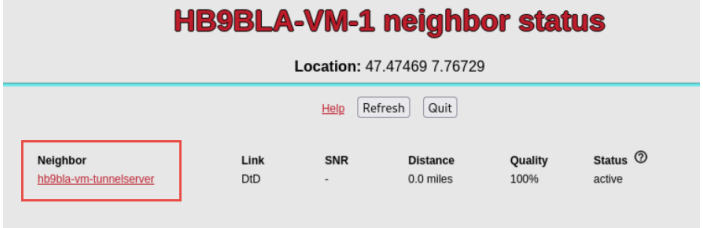
We can connect our phones to our home network instead of the hap router and they should get a normal address in the 192.168.xxx.xxx range.

Because our LAN of the VM is on VLAN 20, we must change the VLAN of the phone, too. We login and go to Network→ Advanced and enable VLAN 20.



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 Tunnel Server VM and the telephone VM should be connected by a DtD link. You see this in the neighbour status:



Be aware that you can only manage your VMs (other than the Tunnel Server) and 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.

Final tip:

You can connect your Management Console to whatever AREDN VM you want (select VLAN 10 for the Tunnel and VLAN20 for the Telephone Server). From there, you can reach all your devices without changing the VLAN of the Management Console.