45HomeLab
A Division of 45Drives

HL- 15 Beast - Storage Server

USER MANUAL

REVISION DETAILS

Version	Description
1.0	Initial published version
1.1	Fixed the IPMI credentials.

Contents

GETTING STARTED

Welcome to the 45HomeLab community. We are so glad you chose us.

ABOUT US

Serving our customers well is at the core of everything we do at 45HomeLab. In an industry where technical support is strictly timed, automated and impersonal we are real people solving real storage problems. Our storage solutions are non-proprietary, giving you the freedom to run any software you choose.

 OUR GOAL

Provide you with the best storage solution for your data needs - not the most expensive one. Today we provide the most affordable storage solutions in the industry.

 OUR MISSION

To provide affordable open-source storage solutions while staying true to our community roots by giving back to the open-source community that we rely on.

 OPEN DESIGN

Unlike mainstream data storage providers 45HomeLab maintains an open design and ongoing relationships with the open-source community.

WHY 45HOMELAB HL15

At 45HomeLab we know home labbers have a strong vision of the infrastructure they want, and how to build and configure it. A key pillar of a great home lab is sufficient storage that is customized and configured how you need it.

The problem is home storage offerings today are under powered and have locked down software, while enterprise solutions are just too big and expensive. We understand how frustrating this is, which is why we've created the 45HomeLab product line. It is big, strong, fast, while also being open and flexible so you can easily modify, upgrade, and repair it, all at a price that makes sense for a power home lab user.

Our new HL15 is a 15-bay server that was designed to provide the power and storage needed for a great home lab.

HARDWARE

COMPONENTS AND SPECIFICATIONS

Chassis Dimensions (LxWxH): 27.25"L x 17.125" W x 9.000"H

Below are the components that will be in your HL15 if you have not requested for any modification in the order the below components will be present by default. Your CPU, motherboard, RAM, Boot drives etc. could change based on your customization.

Configuration A – Asrock ROMED8-2T Platform

ID	Component	Model - specs
1	Backplane	15HDD Backplane & 8SSD Backplane
2	Boot Storage	KINGSTON NV2 1TB Gen 4x4 NVMe M.2
3	Cables	miniSAS HD 0.75m
4	Cables	miniSAS HD 0.6m
5	Chassis	HL15 Chassis
6	CPU	AMD EPYC 7252
	CPU	AMD EPYC 7272
	CPU	AMD EPYC 7282
	CPU	AMD EPYC 7452
7	HBA Card	LSI 9400-16i
	HBA Card	LSI 9305-24i
8	Fans	Noctua NF-A14 PWM
9	Heatsink	Dynatron A35
10	Motherboard	Asrock ROMED8-2T
11	Power Supply	Corsair RM1000x (1000W, 80 Plus Gold)
12	Rails	Sliding rails -26.5in -36.4in length (Optional)
13	RAM	16GB DDR4 ECC RDIMM (2x 8GB)
14	RAM	32GB DDR4 ECC RDIMM (2x 16GB)
15	RAM	64GB DDR4 ECC RDIMM (2x 32GB)
16	RAM	128GB DDR4 ECC RDIMM (2x 64B)
17	RAM	64GB DDR4 ECC RDIMM (4x 64GB)
18	RAM	64GB DDR4 ECC RDIMM (8x 64GB)

Configuration B – ASUS ProArt X870E Platform

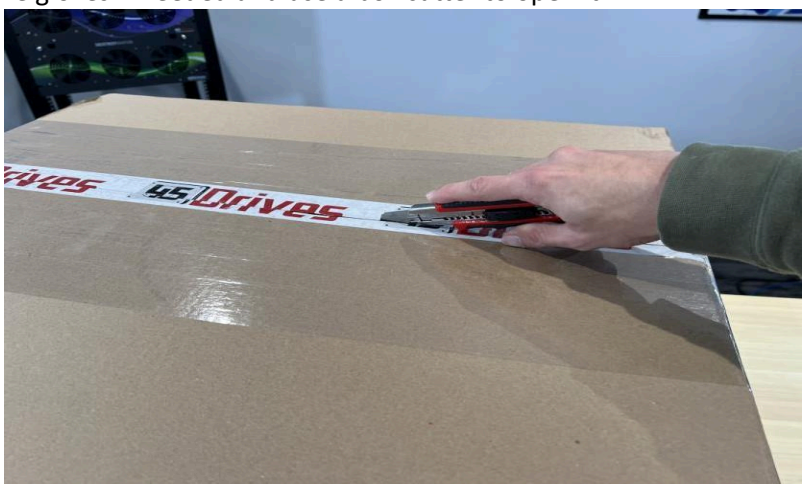
ID	Component	Model - specs
1	Backplane	15HDD Backplane & 8SSD Backplane
2	Boot Storage	KINGSTON NV2 1TB Gen 4x4 NVMe M.2
3	Cables	miniSAS HD 0.75m
4	Chassis	HL15 Chassis
5	CPU	Ryzen9 7950x
6	HBA Card	LSI 9305-24i
7	Fans	Noctua NF-A14 PWM
8	Heatsink	Noctua NH-D15 G2 LBC
9	Motherboard	ASUS ProArt X870E-CREATOR WIFI
10	Power Supply	Corsair RM1000x (1000W, 80 Plus Gold)
11	Rails	Sliding rails -26.5in -36.4in length (Optional)
12	RAM	32B DDR5 EXPO/XMP UDIMM (2x 16GB)
13	RAM	64GB DDR5 EXPO/XMP UDIMM (2x 32GB)
14	RAM	64GB DDR5 EXPO/XMP UDIMM (4x 32GB)

INITIAL SETUP

UNBOXING THE HL15

- Once you receive your new unit from shipping, inspect the box and make sure there isn't any shipping damage.

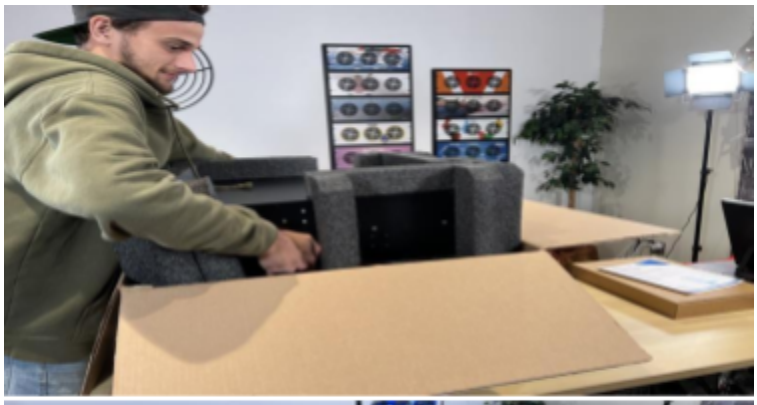
- Put on protective gloves if needed and use a box cutter to open it.



- Once the box is open you can take the unit out of the box.

- Inside we can find another smaller box containing the accessories purchased with the server along with some paper documents.

- Lay the new unit on a flat surface like a table and make one more inspection to make sure everything is in order. Lift the unit out of the box by the left and right side. Having a second person may help.



- Remove foam padding and we are complete!

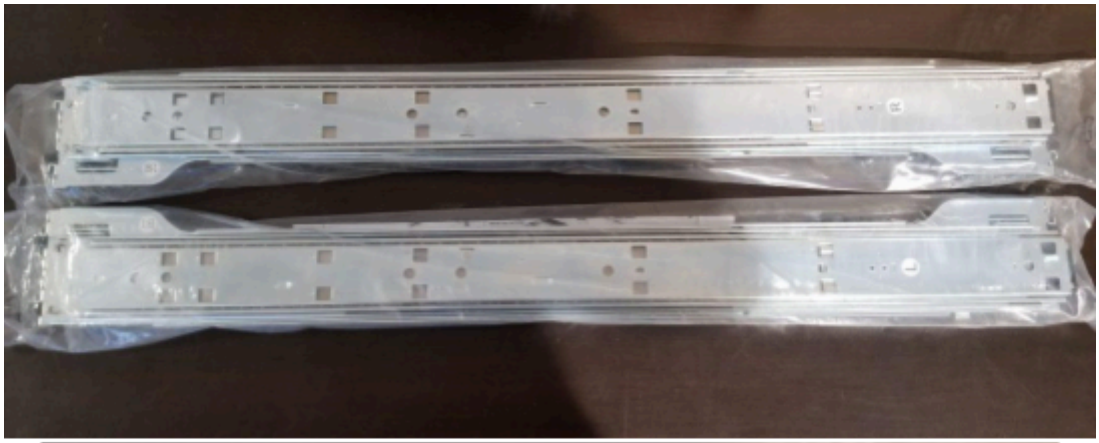
- After the unboxing is complete you can move on to either racking the unit or placing the unit in its final destination with the provided rubber feet.

- The Rubber feet can be screwed into the base of the system for desktop placement.

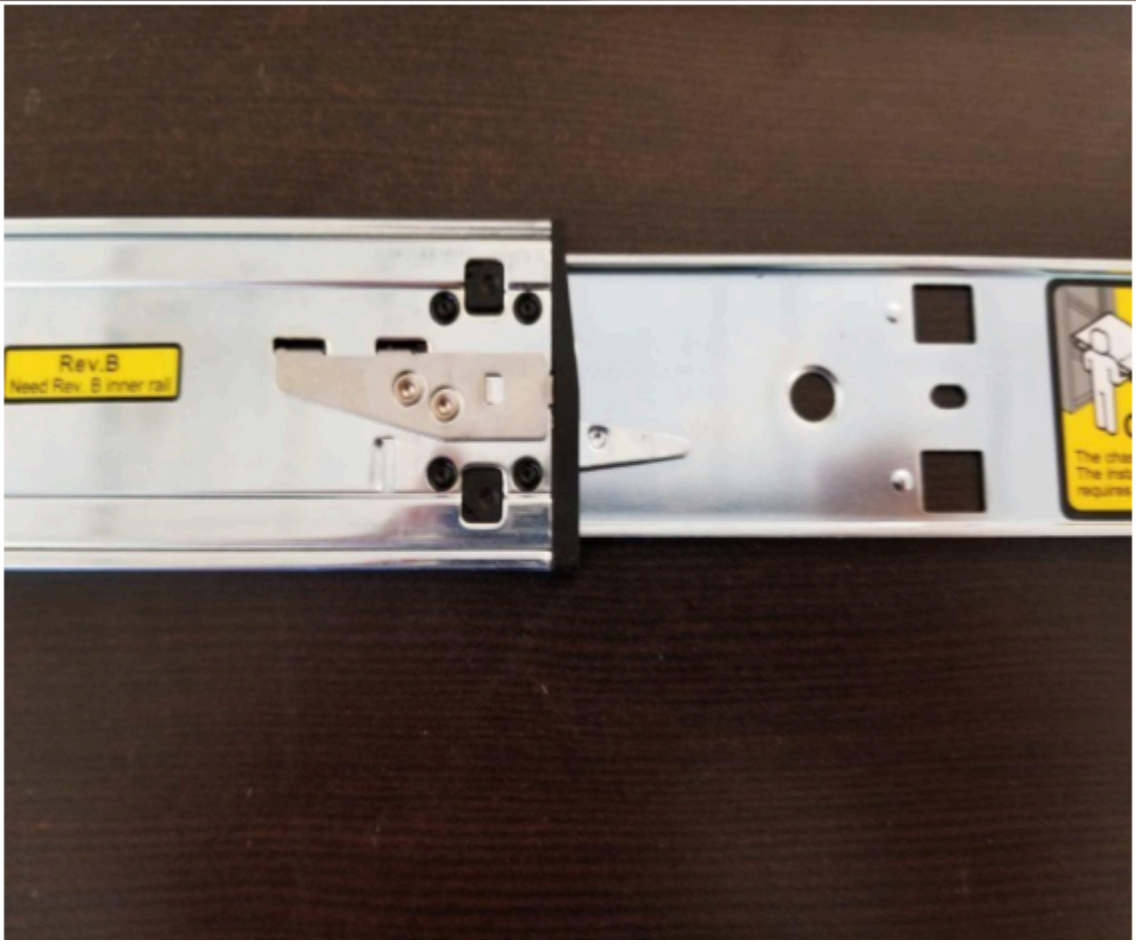
RACKING THE HL15

Note: Do not install the storage hard-drives before racking your new unit.

- If you purchased rails with your new 45HomeLab unit or have supplied your own, you will be able to attach them to the sides of the unit.
- Unbox the rails and disconnect the part that attaches to the unit. There should be a latch on the rail that allows easy disconnection.




R
CAUTION
L


Rev.B
Need Rev. B inner rail

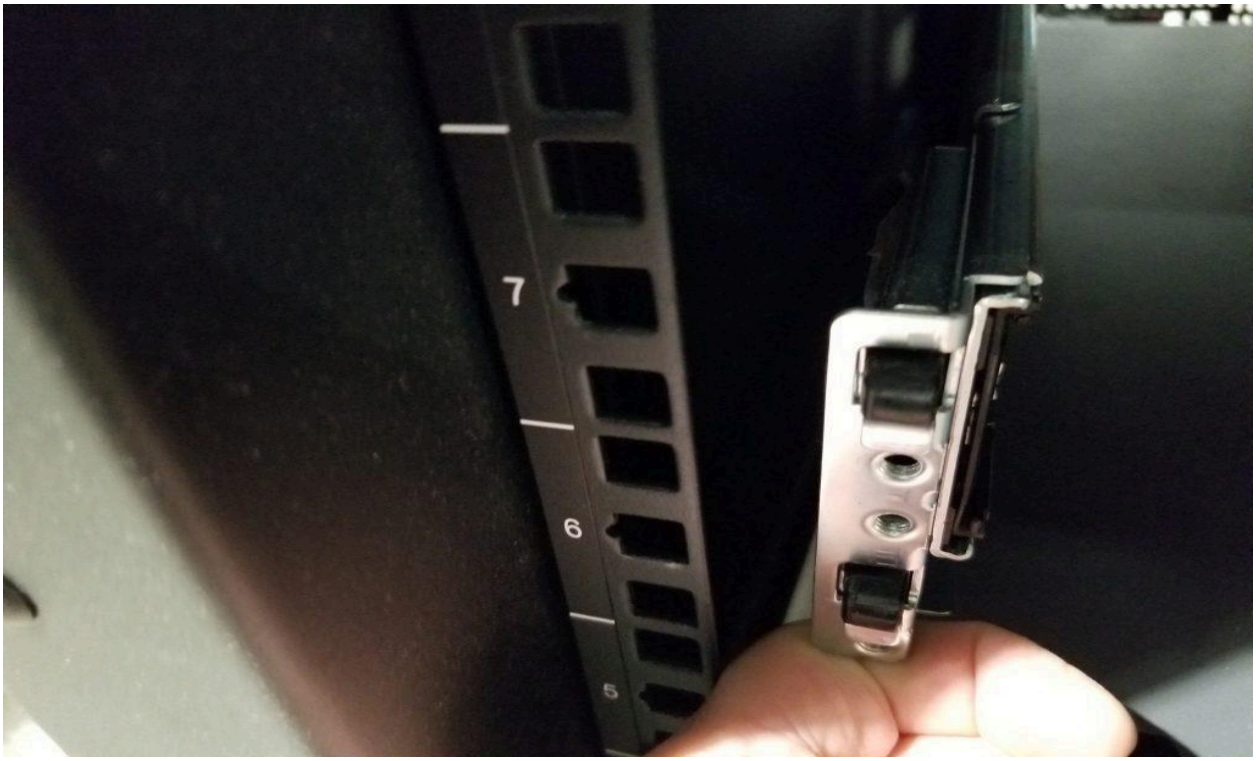
- Locate the screws inside the included rail mount kit and screw the rails onto each side of the unit.

45HomeLab
A Division of 45Drives

- The rails only go one way so make sure you put them in correctly. The rails should have indicators on them to show which side and which side is up.

- Screw the screws back in their respective places, holding the rail as you do this. (It may be easier if one person holds the rail and a second person screws them in.)

- Once the rails are attached to the side of the unit, you can install the other piece of the rail into the rack. Once again there are indicators showing which way needs to be up and which side is which.



45HomeLab
A Division of 45Drives

- After both parts are installed, it is now time to connect them. You have to align the sides perfectly.
- Extend the rails on the rack to full length.

- Pick up the unit with another person and align the sides of the rails on the rack with the sides of the 45HomeLab unit. Make sure that the rails are on securely or the unit may fall.
- Slide the unit onto the rails until you hear a click. You will then need to flip the latch to fully move the unit into the rack.

- Now let’s attach the rack ears with the provided hardware.

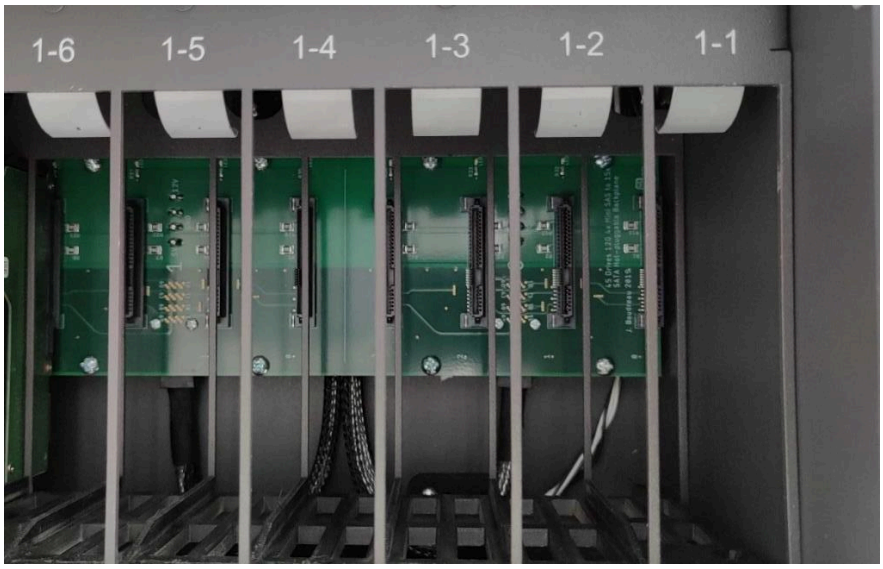
- Your server is now racked!

s

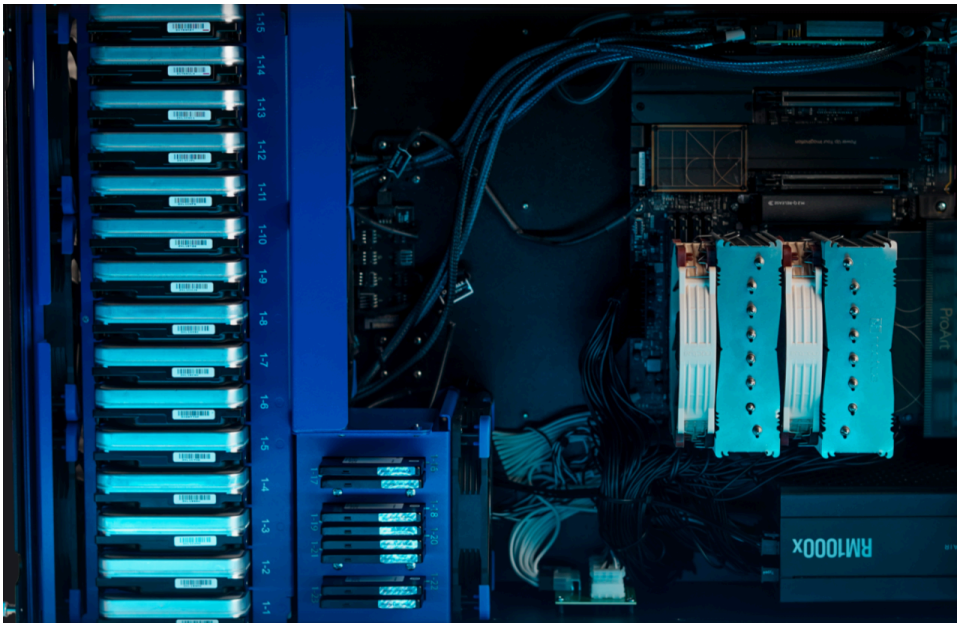
- After the unit is racked the next step is to Insert the Storage Drives

INSERTING THE STORAGE DRIVES

- Once your unit is racked or in its final destination you can now start installing the storage drives.
- It is best practice to install the drives starting from index 1-1 and sequentially incrementing in order.



- The drive should fit snugly into a slot. Make sure the back of the drive is facing the right when placing in the slot



- After all the drives are inserted into the slots, you can close up the unit and move on to hooking up the needed cables to your unit.

CABLE SETUP

At this point, you should have the unit unboxed, racked, and storage drives installed. The next step is to connect the cables needed to connect and configure the unit.

Asrock RomeD8-2T

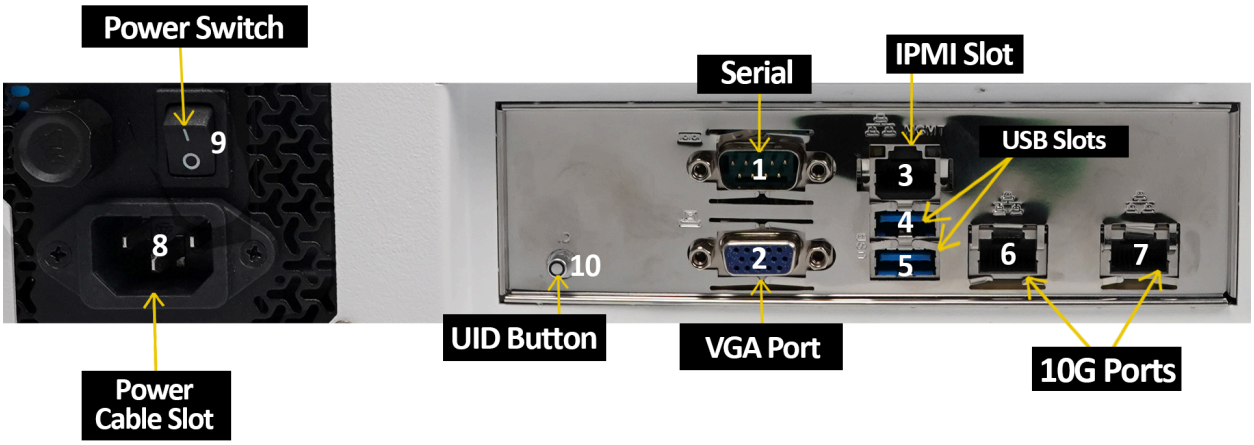


Fig: Back panel slots

No	Component
1	Serial
2	VGA port
3	IPMI slot
4	USB slot
5	USB slot
6	10G port
7	10G port
8	Power cable slot
9	Power switch
10	UID Button

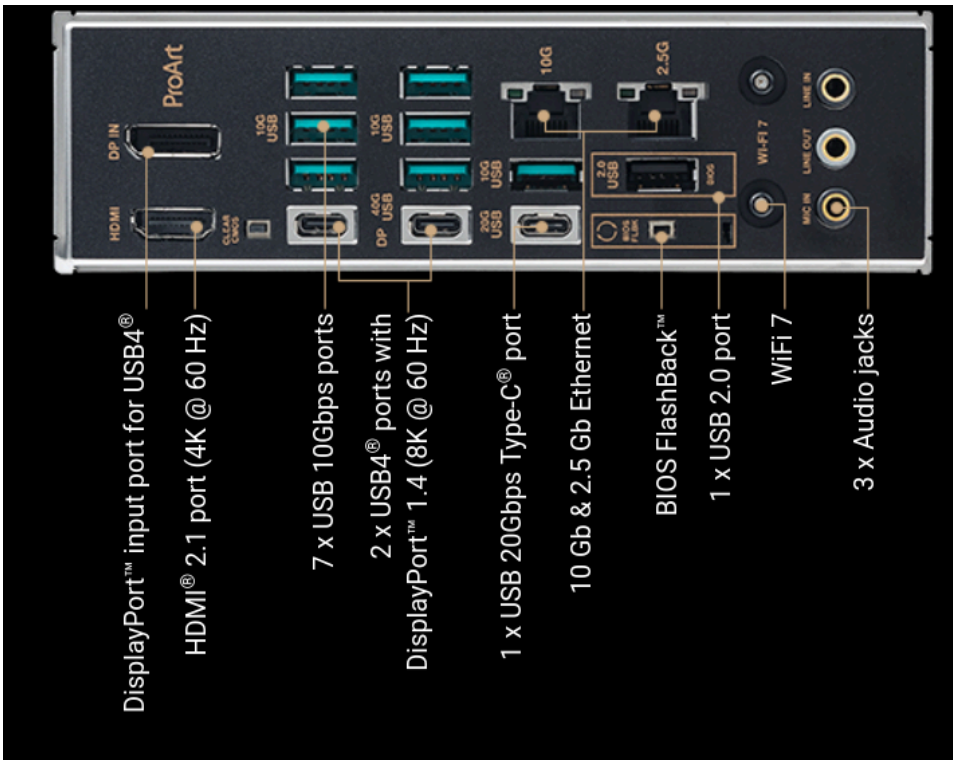
Table: Back panel slot details

- For the initial connection to your unit, you will need to connect an ethernet cable to the IPMI port (3) on the back of the unit. You can connect a 1GB LAN cable from your router or switch as

per your network setup.

- You can plug in another Ethernet/ LAN cable to provide internet to the server and where we will be accessing the data from to either slots labeled 6 or 7. If you want to bond or aggregate them then you can connect LAN cables to both 6 and 7.
- A VGA monitor will also be connected initially so you can see the BMC IP address in the bottom left-hand corner of the screen when the unit first turns on. You can connect the monitor to the VGA port (2).
- If you want to configure the unit locally, a USB keyboard will also be beneficial. You can use any of the USB slots labelled from 4 or 5 in fig above.
- The last step would be to plug in the power cable to the power supply. Slot labelled 8.
- After that you can turn on the power switch and then press the round blue power button to turn on.

Asus ProArt X870E-CREATOR WIFI



- For the initial connection to your unit, you will need to connect an ethernet cable to the either 2.5 Gb or 10 Gb ethernet port on the back of the unit. You can connect a 1Gb LAN cable from your router or switch as per your network setup. If you want to bond or aggregate them then you can connect LAN cables to both 6 and 7.
- Connect a monitor via **HDMI or DP IN**.
- If you want to configure the unit locally, a USB keyboard will also be beneficial. You can use any

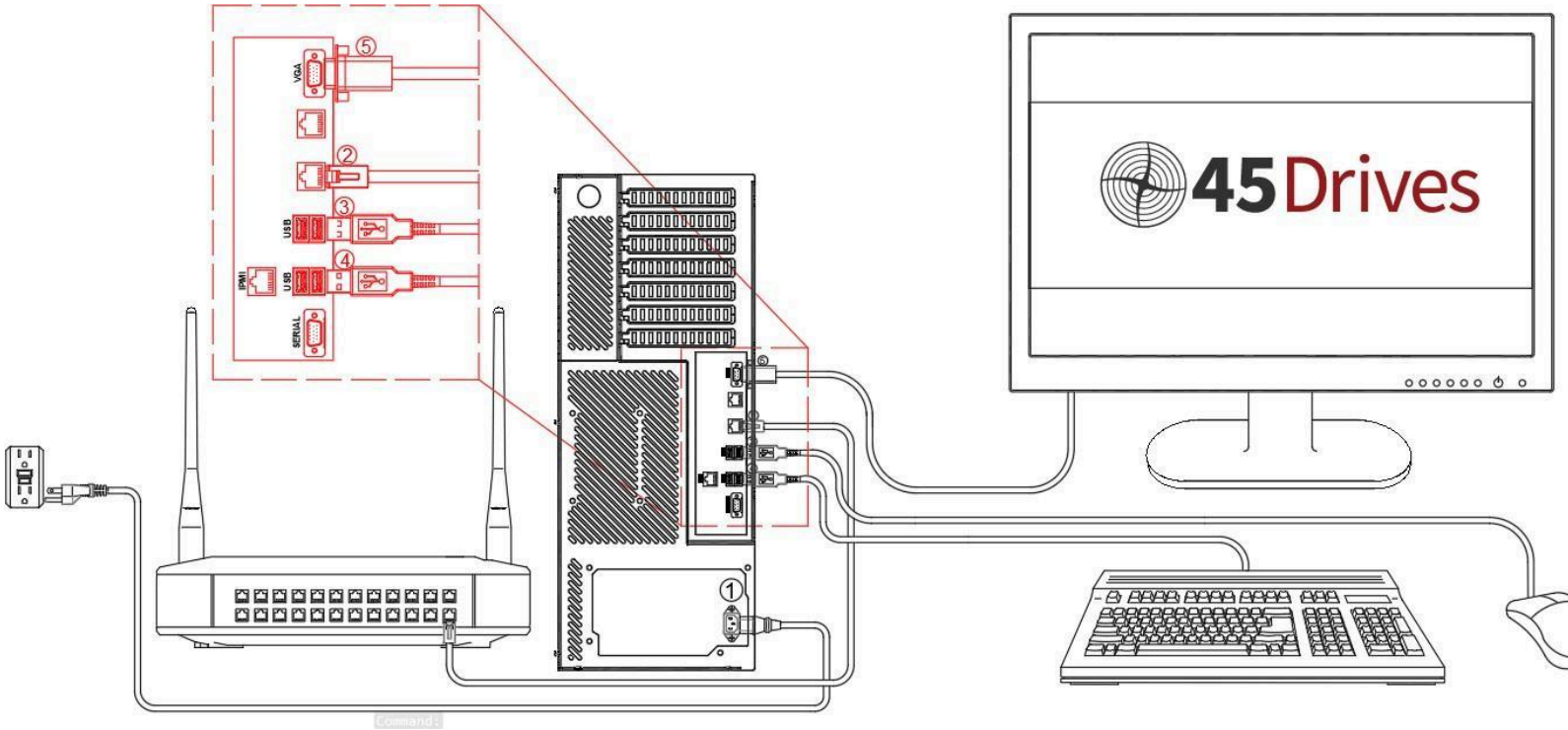
of the USB.

- The last step would be to plug in the power cable to the power supply.
- After that you can turn on the power switch and then press the round blue power button to turn on.

Power requirements

- The PSU we provide with the appropriate options is a Corsair RM1000x
- If you are sourcing your own power supply, you can use this as a minimum guideline. It is especially important to ensure you have at least 20A of 5v power.
- In addition, if you plan on using a graphics card, or anything that will increase power draw, you may require a more powerful supply.
- Corsair Modular ATX Power Supplies are guaranteed to fit

45HOMELAB HL 15 INITIAL CABLING SETUP DIAGRAM



45HomeLab
A Division of 45Drives

INITIAL CONFIGURATION /SETUP

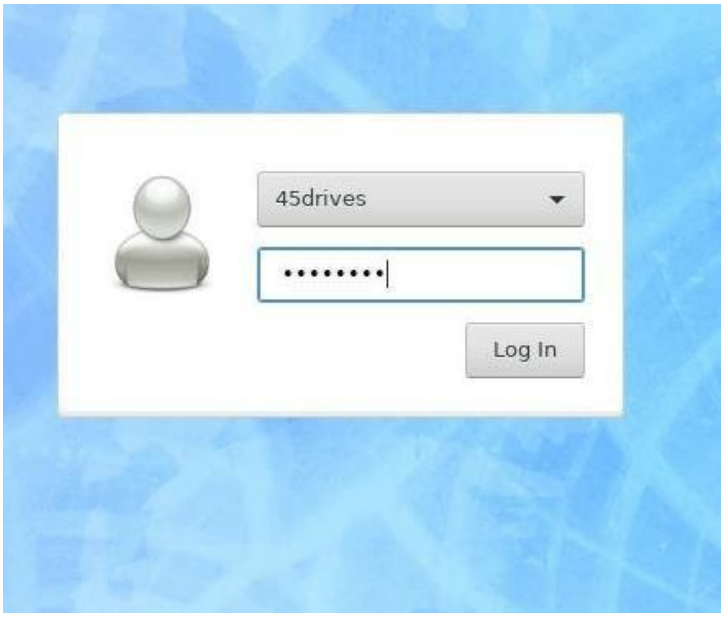
Now you have your HL15 all powered up and ready to be configured. Before we proceed you need to have the below steps completed.

1. The power cables have been connected.
2. The IPMI slot has been plugged in with a LAN cable(for ASRock RomeD8-2T only)
3. At least One of the ethernet slots have a LAN cable plugged in
4. A monitor is connected.

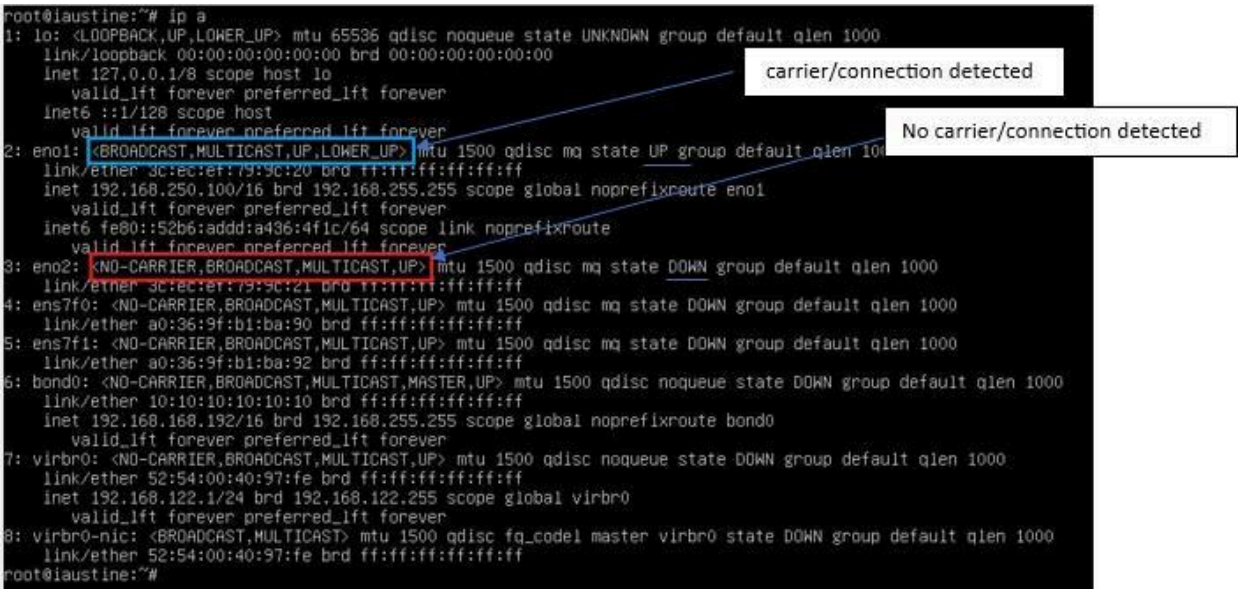
NETWORK SETUP

USING THE DESKTOP UI

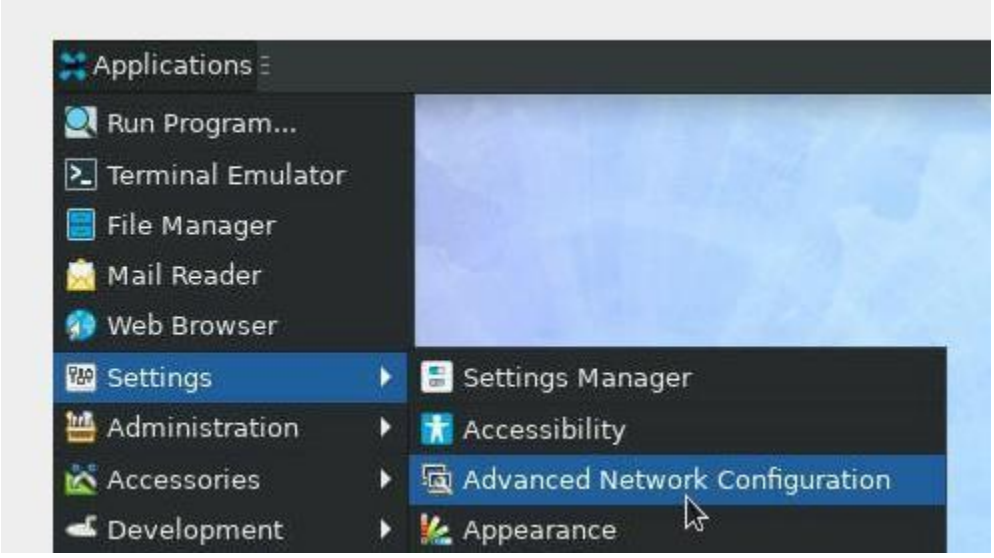
- Login to the desktop UI using the 45drives user and 45Dr!ves as the password.



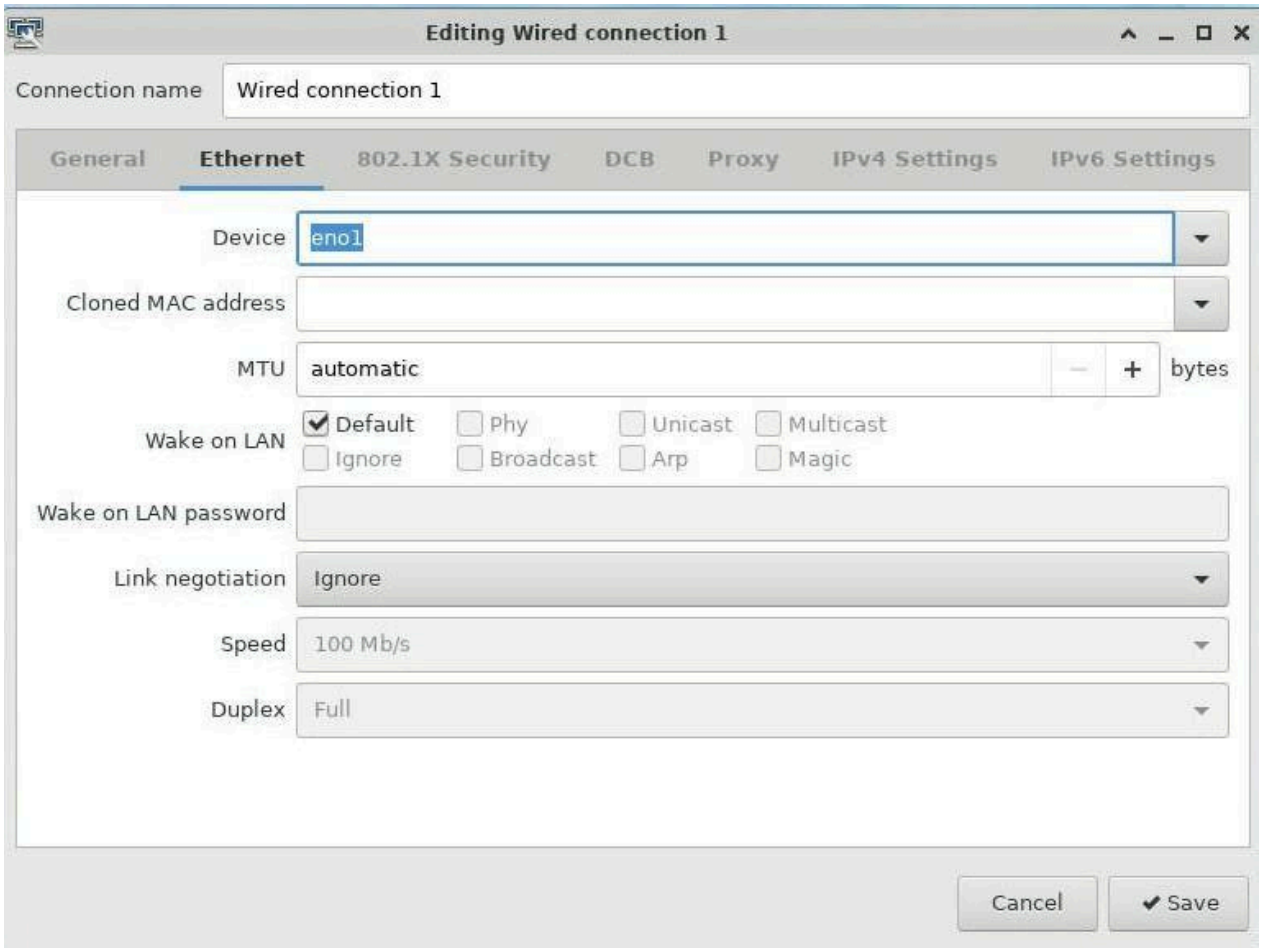
- Once logged in we can check if our interfaces are up
- Open a terminal and type the command **ip a** to check if your interface is up. You should be able to see that a broadcast carrier is available like below.



- Go to the applications in the top left corner. Go to settings -> Advanced network Configurations

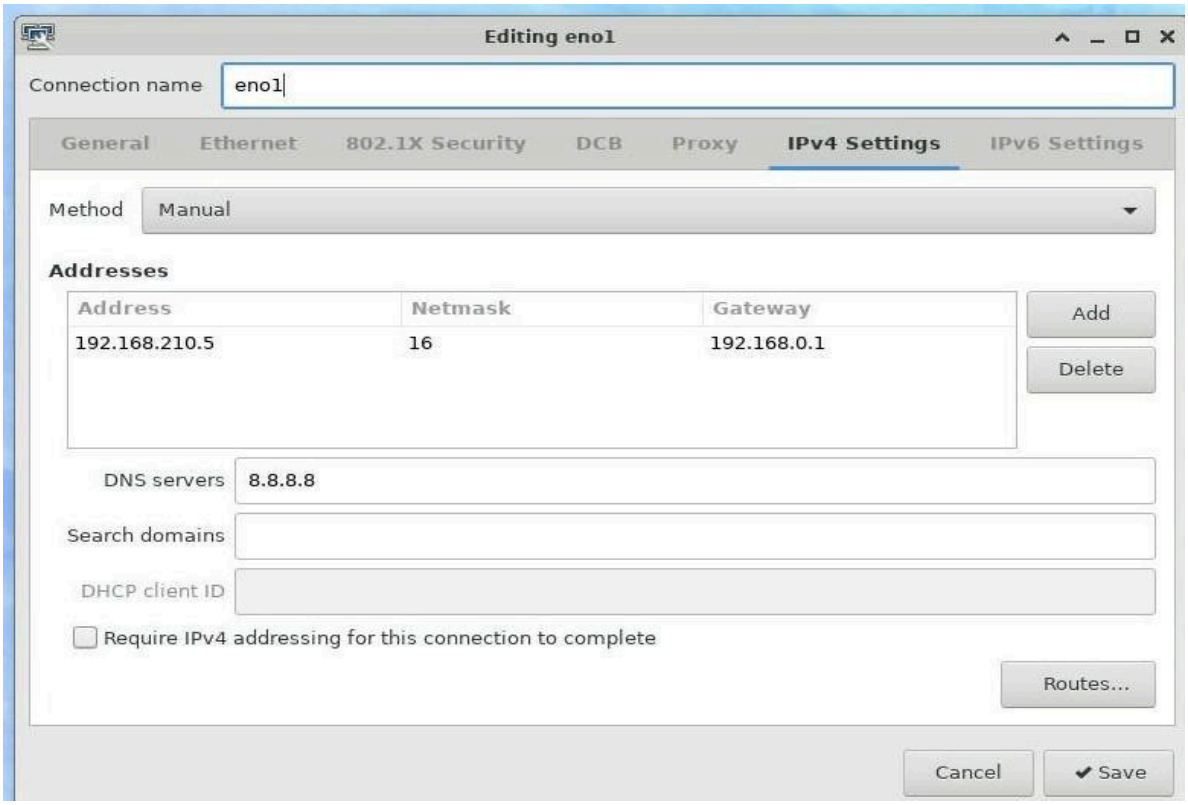


- You should be able to see your network connections/ interfaces there
- It will either show the interface name or might say like wired connection1, 2 etc
- You can double click on it to check the actual interface name
- If the interface is not showing you can click on the plus sign and add a new ethernet connection.

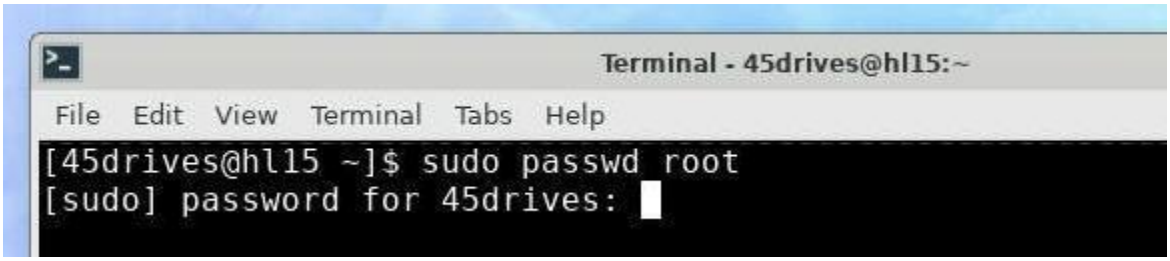


- Edit the interface and go to the IPV4 settings section to set the IP. Change the method to manual for setting the static IP and click on add to enter the IP details.

Once the details have been entered click save



After this you can start setting the password for your root user using below command in the terminal and set the password.



SETTING UP IPMI (Asrock ROMED8-2T Only)

This is just an optional setup, IPMI is not required for the setup. IPMI would help you with remote monitoring. You can skip this step if you prefer.

IPMI or Intelligent Platform Management Interface is an open, industry-standard interface that was designed for the management of server systems over network. It enables you to monitor and control your server platform, as well as to retrieve information about your server platform is a standard. It supports FRU inventory reporting, system monitoring, logging of system events, system recovery (system reset or power off) or alerting.

- Turn on the server by pressing the power button at the back panel.
- Once the server starts booting up the IPMI address will be shown during bootup in the bottom right-hand corner.

Fig – Finding IPMI/BMC IP

- Once you have noted the IPMI IP you can open the same in the browser in your laptop or your desktop as per your preference.
- You need to open the browser and type https://BMCIP to access the IPMI interface

- The default login credentials are :

Username - admin
Password - 45Dr!ves

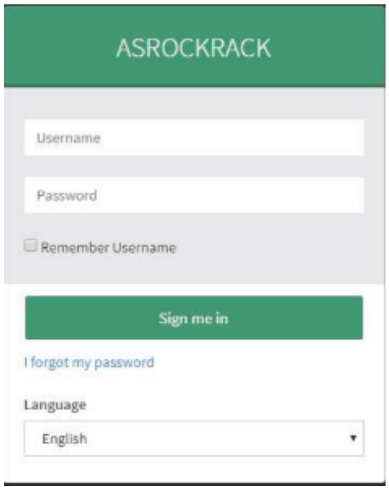
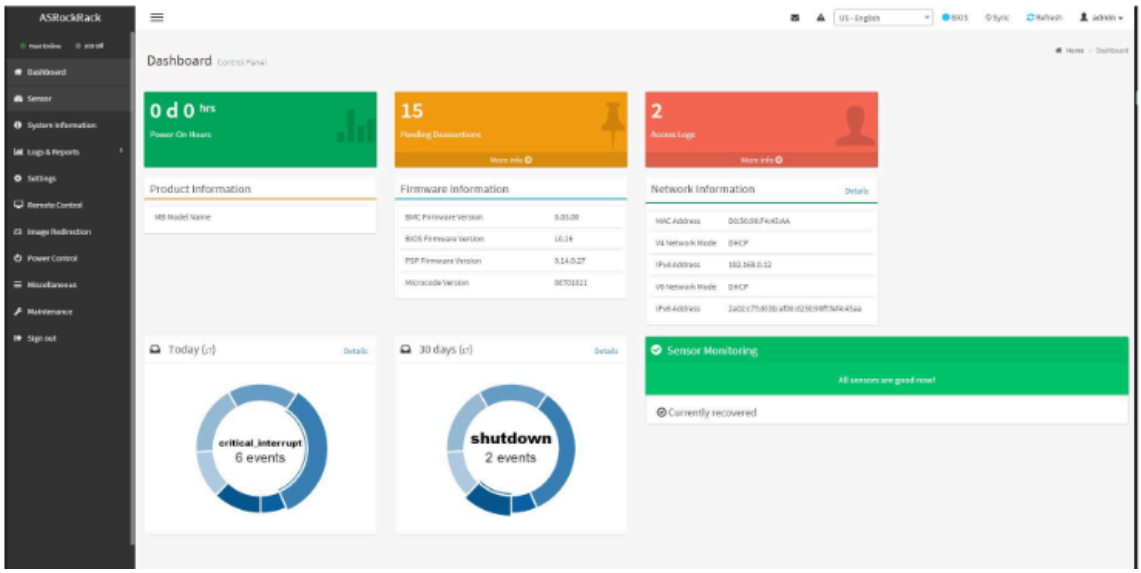


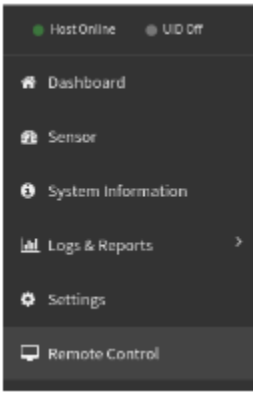
Fig-IPMI log screen

- Pictured below is the IPMI set up screen

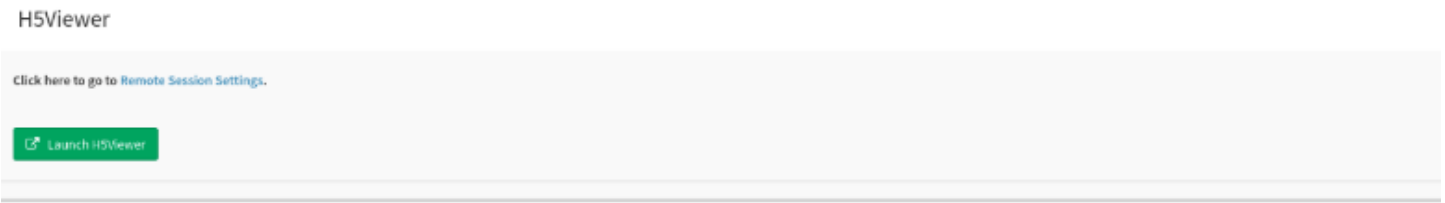


- Once you click on a Launch, a new window will be open. You click “launch H5Viewer” to open the remote console in a

new window using the HTML5 interface. If you prefer the Java interface, click “Launch Jviewer



- Once you click on a Launch, a new window will be open. You click “launch H5Viewer” to open the remote console in a new window using the HTML5 interface. If you prefer the Java interface, click “Launch Jviewer” instead.



- Once you click on a Launch, a new window will be open.
- Here you can login to the desktop using your login user account.

SETTING THE NETWORKING USING NMTUI

This is just an additional option to set up a network other than using the UI. You can ignore this if you have already set the IP using the UI and move to the Houston login and config section.

- You can run “ip -c a” to show your interfaces with colored IP addresses. This helps to distinguish what you’re looking at.

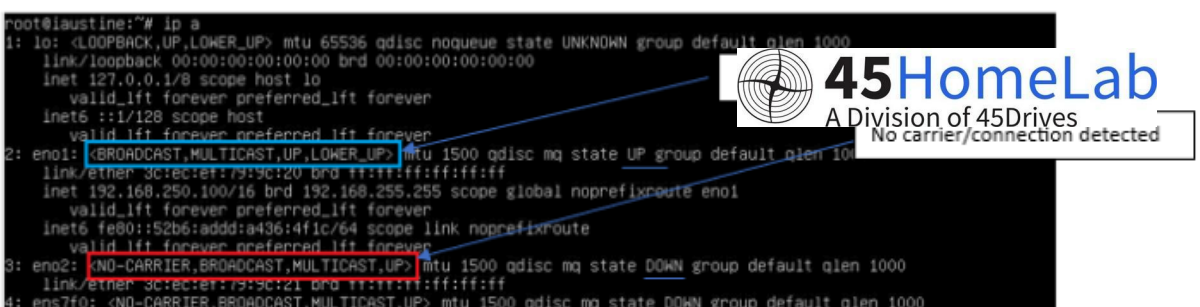


Fig – showing connection detected on the network interfaces

- So, you can see in the above diagram that the interface eno1 has a carrier detected which means there is a LAN cable connected to the NIC slot.
- Similarly, eno2 has no carrier which means it's not connected. If it is connected and still not detected it will need further troubleshooting.
- Once you have your interface connected with the carrier detected, the next step is to set a static ip for your server.
- We will be using this IP going forward to access the server.

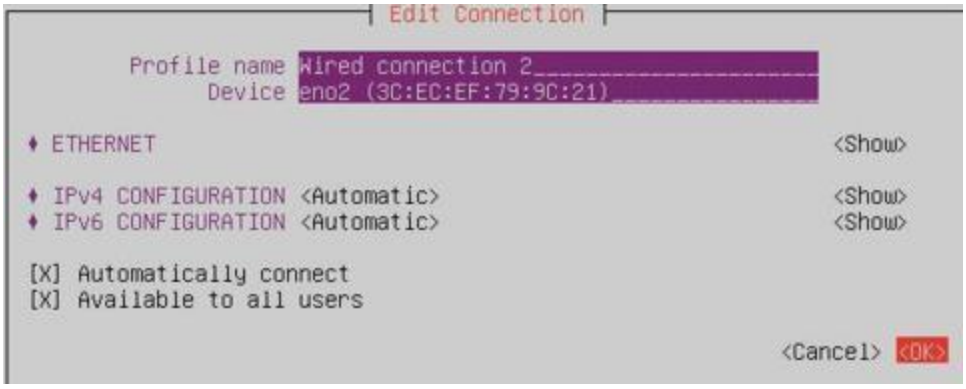
Setting Static IP

- Run the command “nmtui” to access NMTUI, network manager.

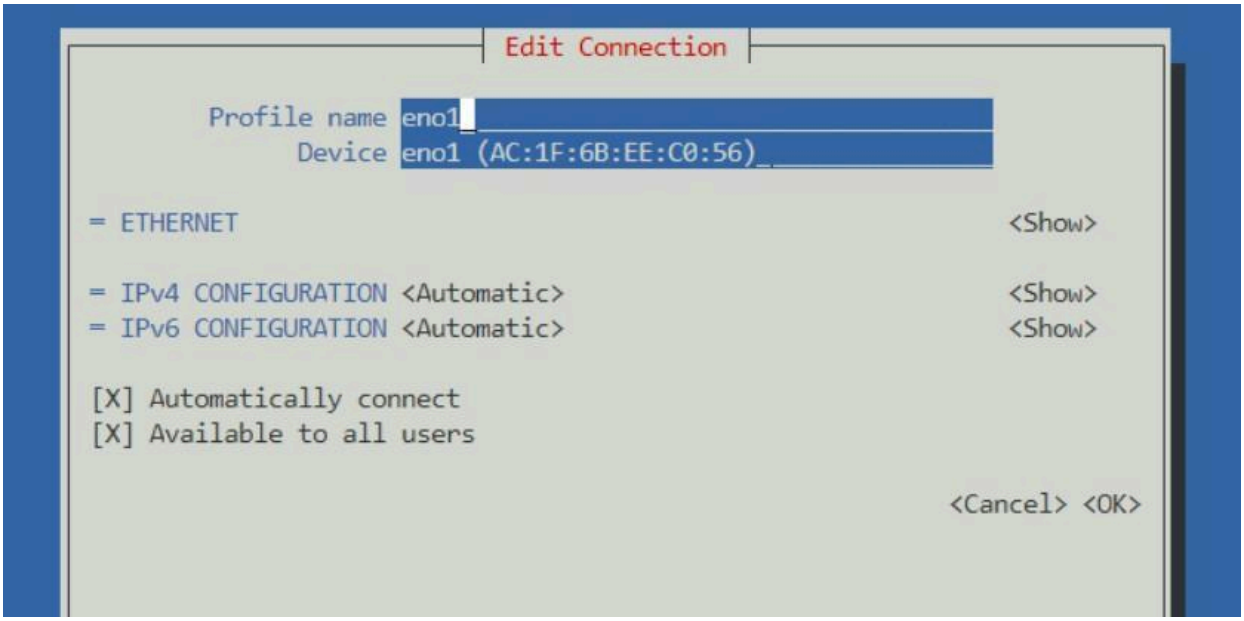
Fig: NMTUI

- First, we will set a static IP. Click Edit a connection.
- Select the interface that we want to set the static IP. This example is eno1. Select that interface and click on edit.

- In case the interface names show up as wired connection 1,2 etc you can select that and confirm what the actual interface name is by checking device name like below screenshot.



- Navigate to IPv4 configuration and change automatic to manual and then click show.



- Here we can enter our static IP information. Be sure to remember to add your subnet after your IP address. (Most common would be /24, which correlates to a subnet mask of 255.255.255.0) If you will be joining a domain, you can set your server's DNS. Click OK. (Even if you aren't joining a domain, if you want to be able to resolve internet addresses via names you will need to give a DNS whether it's a public one or an internal one. Because we're setting a static IP, if you don't include a DNS at all you won't be able to download packages or anything because it needs DNS.)

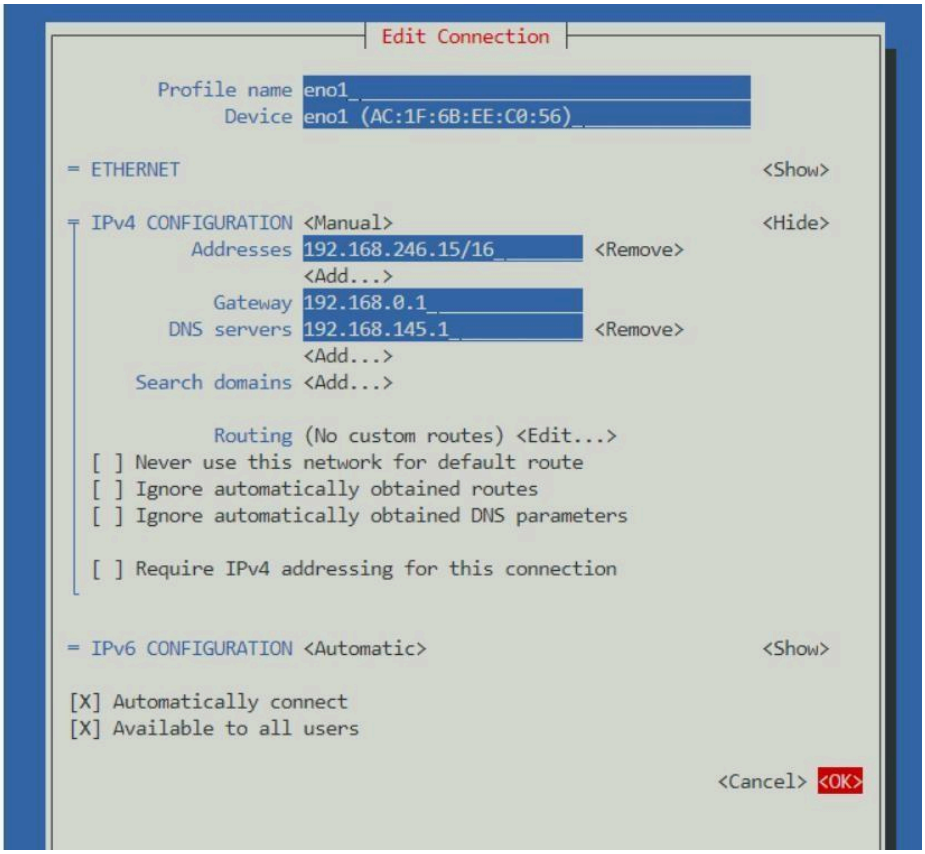
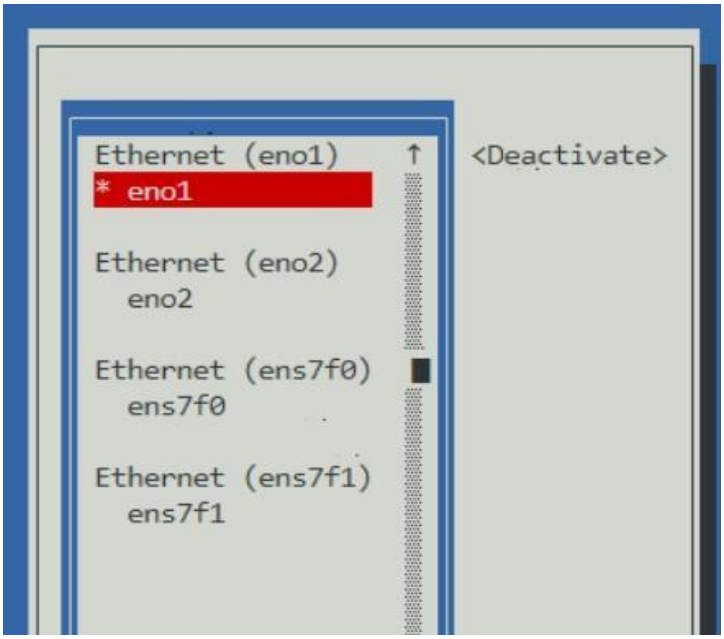


Fig – Setting static IP

- Next, click Back and then click on Activate a connection.
- You will need to activate that interface .



- Deactivate and reactivate the interface. This resets the connection and ensures proper communication. (If you as SSH'd over the interface that you are working on, deactivating will kick you out of your SSH session, so either doing this when you are physically at the server, or over IPMI will allow you to disable and enable the interface.)

Setting a Bond

- Click Edit a connection, and click Add.

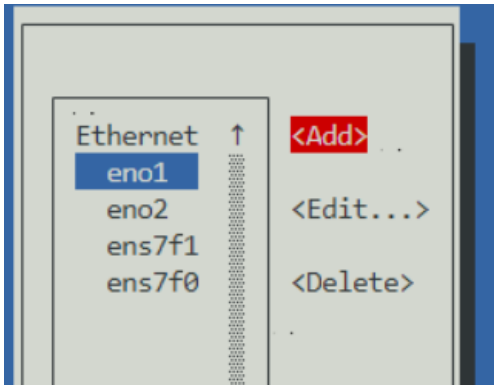
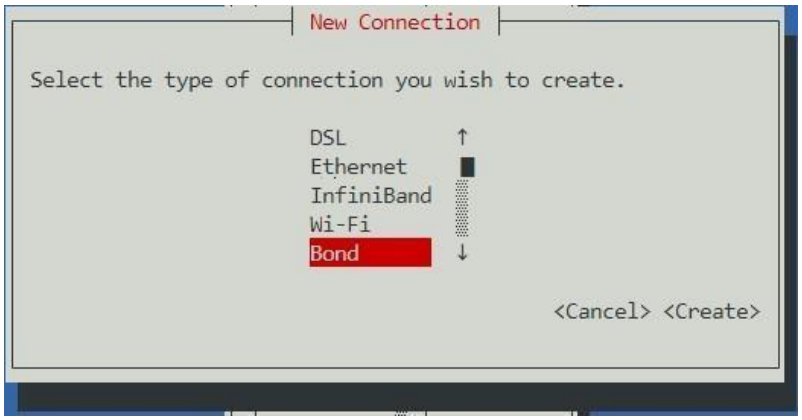
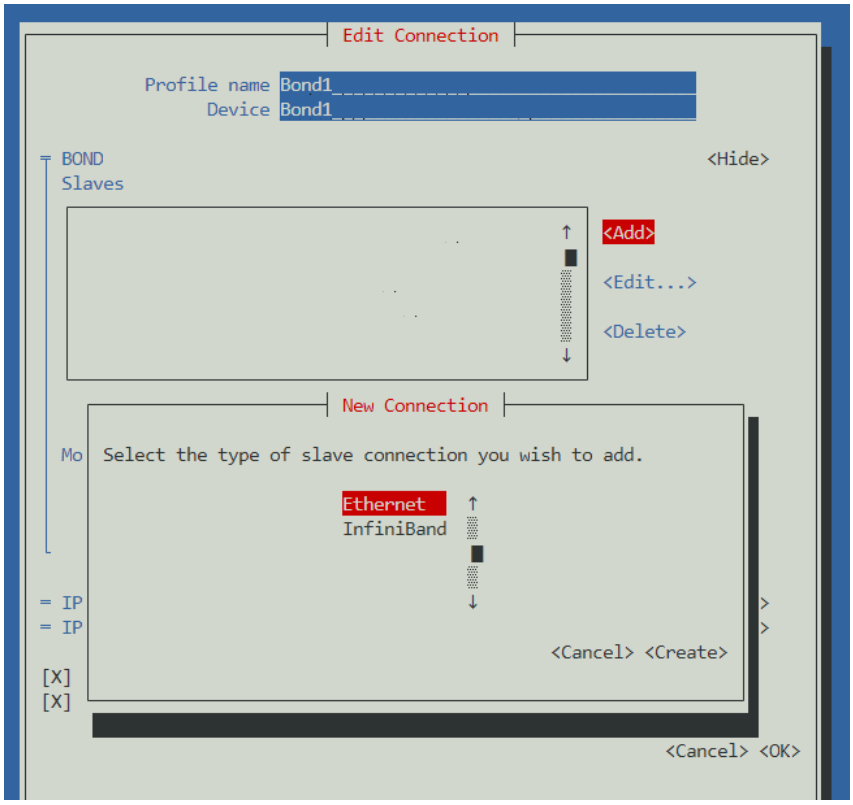


Fig – Add Bond

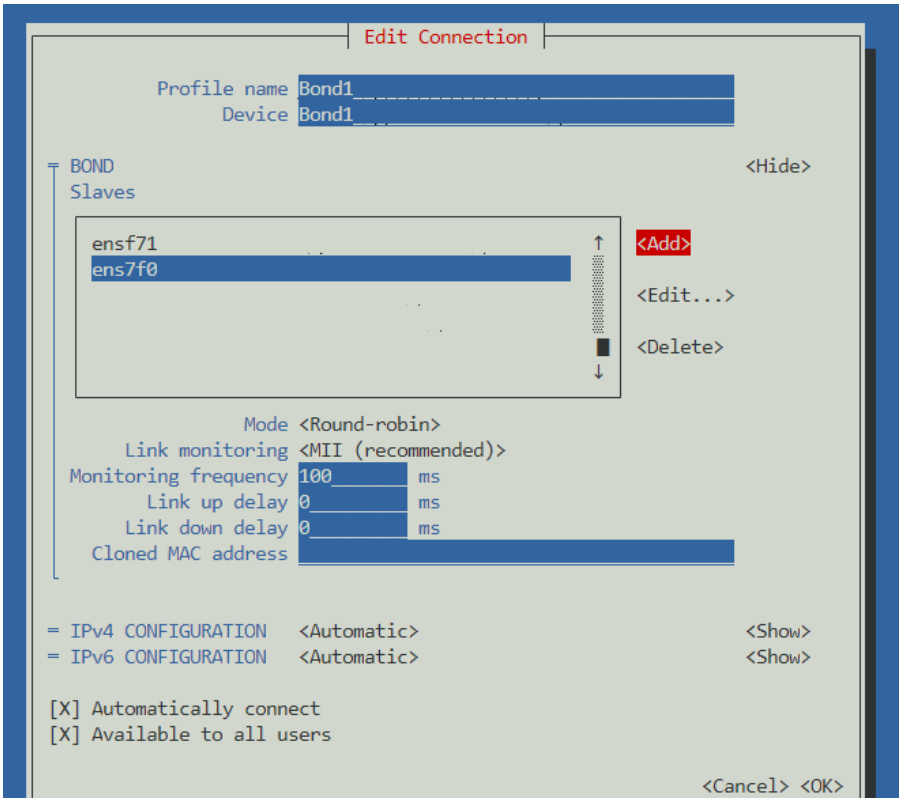
- For the type of connection, select Bond.



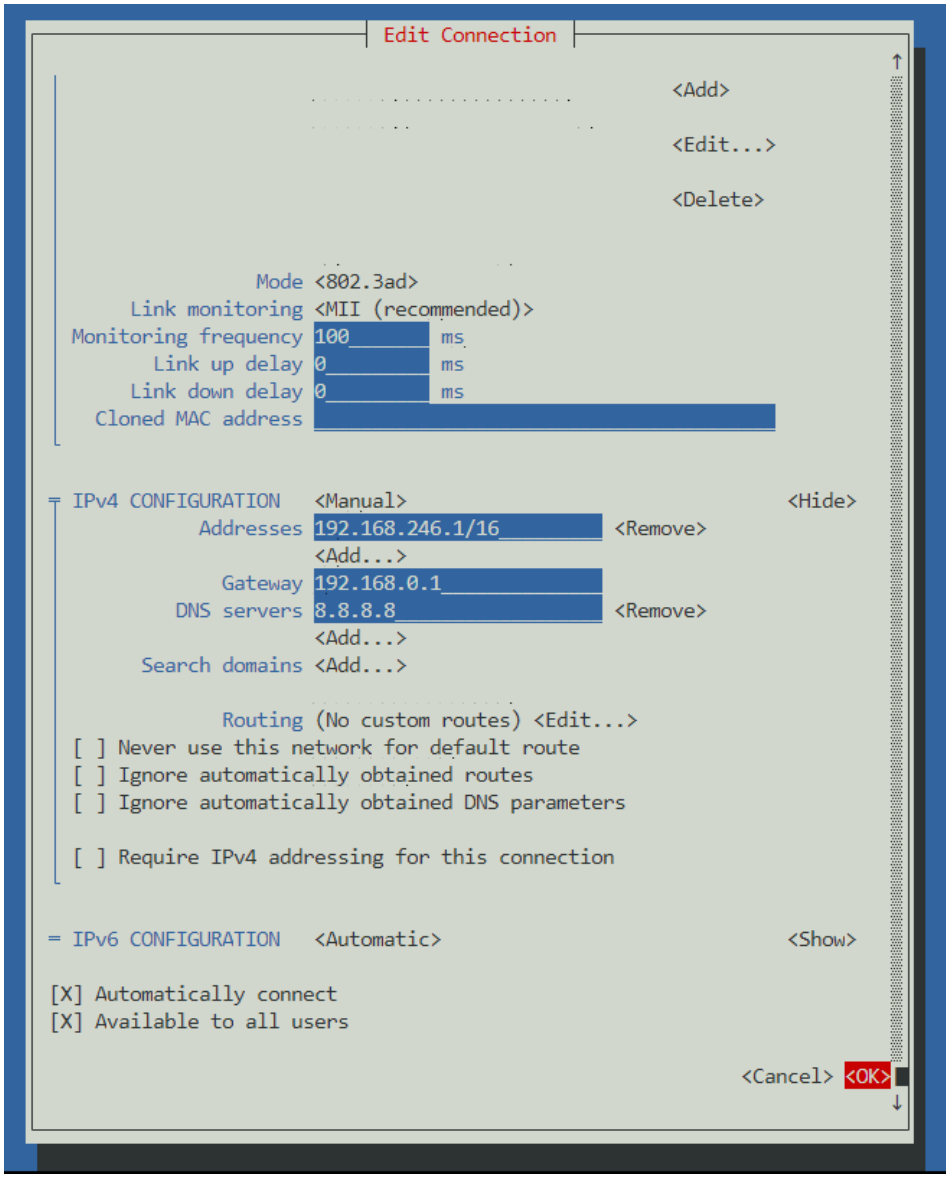
- Enter your Profile name and Device name. For this example, we have used Bond1. Under Slaves, click Add and select Ethernet.



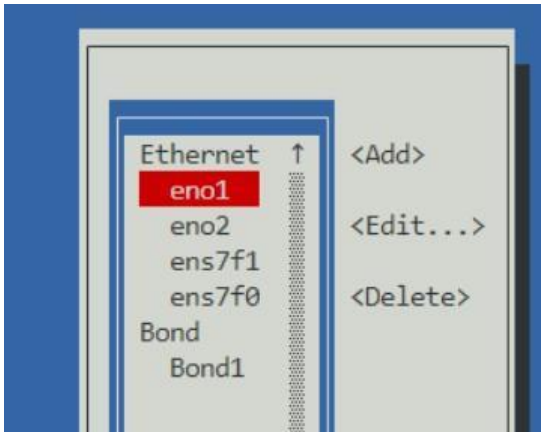
- For this example, we will be using the NIC ports. Enter your interface name under Profile name and Device, and click OK.
- Do the same for ens7f1. Your Slaves table should look like this.



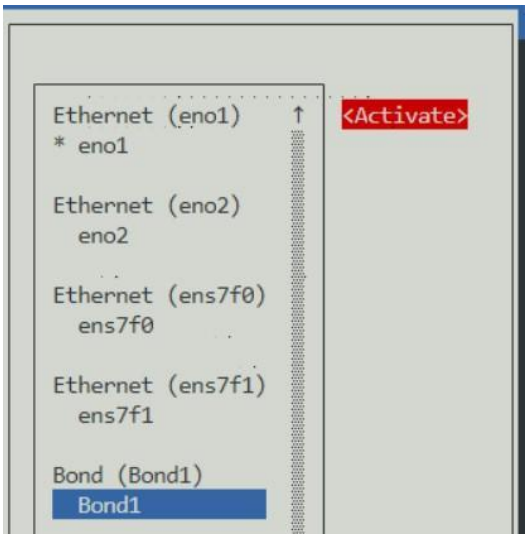
- Add your IP address info for your bond, and click OK.



- Here we can see our bond has been created.



- Click Back, and go to Activate a connection. Highlight Bond1 and Deactivate it.
- Now activate it. Make sure the two slave ports are deactivated.



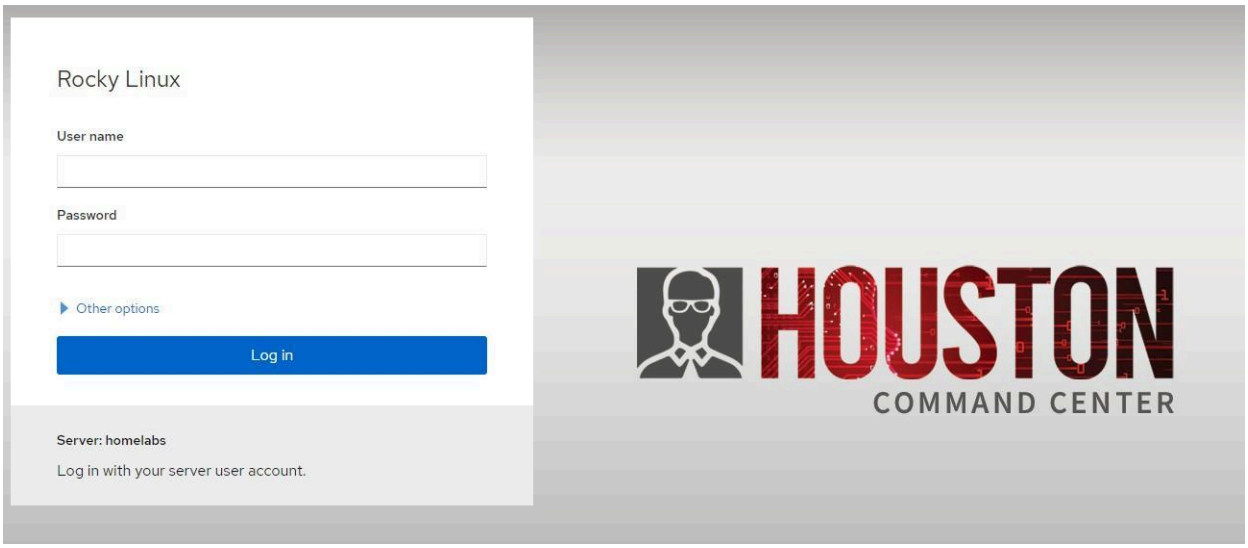
- Now, run another “ip a” command. Your bond is now listed.

```
[root@krocky45d ~]# ip -c a
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
       valid_lft forever preferred_lft forever
2: eno1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
    link/ether ac:1f:6b:ee:c0:56 brd ff:ff:ff:ff:ff:ff
    inet 192.168.246.15/16 brd 192.168.255.255 scope global noprefixroute eno1
       valid_lft forever preferred_lft forever
    inet6 fe80::ae1f:6bff:feee:c056/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
3: eno2: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc mq state DOWN group default qlen 1000
    link/ether ac:1f:6b:ee:c0:57 brd ff:ff:ff:ff:ff:ff
4: ens7f0: <NO-CARRIER,BROADCAST,MULTICAST,SLAVE,UP> mtu 1500 qdisc mq master Bond1 state DOWN group default qlen 1000
    link/ether a0:36:9f:2f:ff:40 brd ff:ff:ff:ff:ff:ff
5: ens7f1: <NO-CARRIER,BROADCAST,MULTICAST,SLAVE,UP> mtu 1500 qdisc mq master Bond1 state DOWN group default qlen 1000
    link/ether a0:36:9f:2f:ff:40 brd ff:ff:ff:ff:ff:ff permaddr a0:36:9f:2f:ff:42
9: Bond1: <NO-CARRIER,BROADCAST,MULTICAST,MASTER,UP> mtu 1500 qdisc noqueue state DOWN group default qlen 1000
    link/ether a0:36:9f:2f:ff:40 brd ff:ff:ff:ff:ff:ff
    inet 192.168.246.1/16 brd 192.168.255.255 scope global noprefixroute Bond1
       valid_lft forever preferred_lft forever
[root@krocky45d ~]#
```

HOUSTON LOGIN AND CONFIG

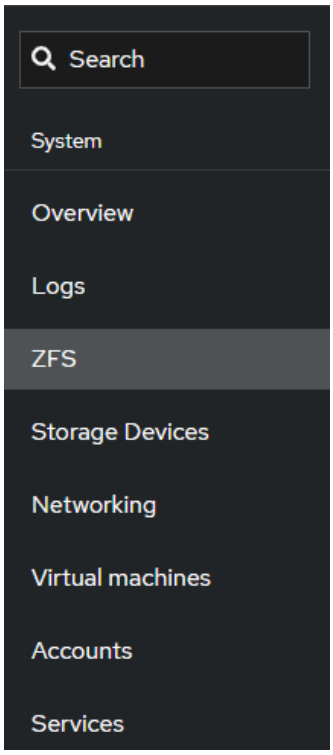
You should already have the Houston modules installed in your server. You can access Houston UI at https://SERVER-IP:9090 .

You can use the root account you created and the password or the 45drives user as well to login. The root account would have the elevated privileges.

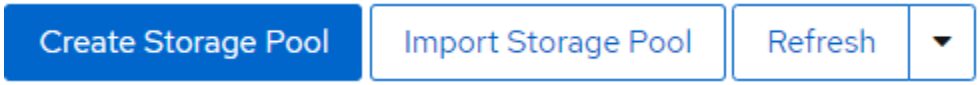


CREATING ZPOOL

- Once we're logged into Houston, you should see a list of modules. You will see the ZFS tab. Click on this to continue.

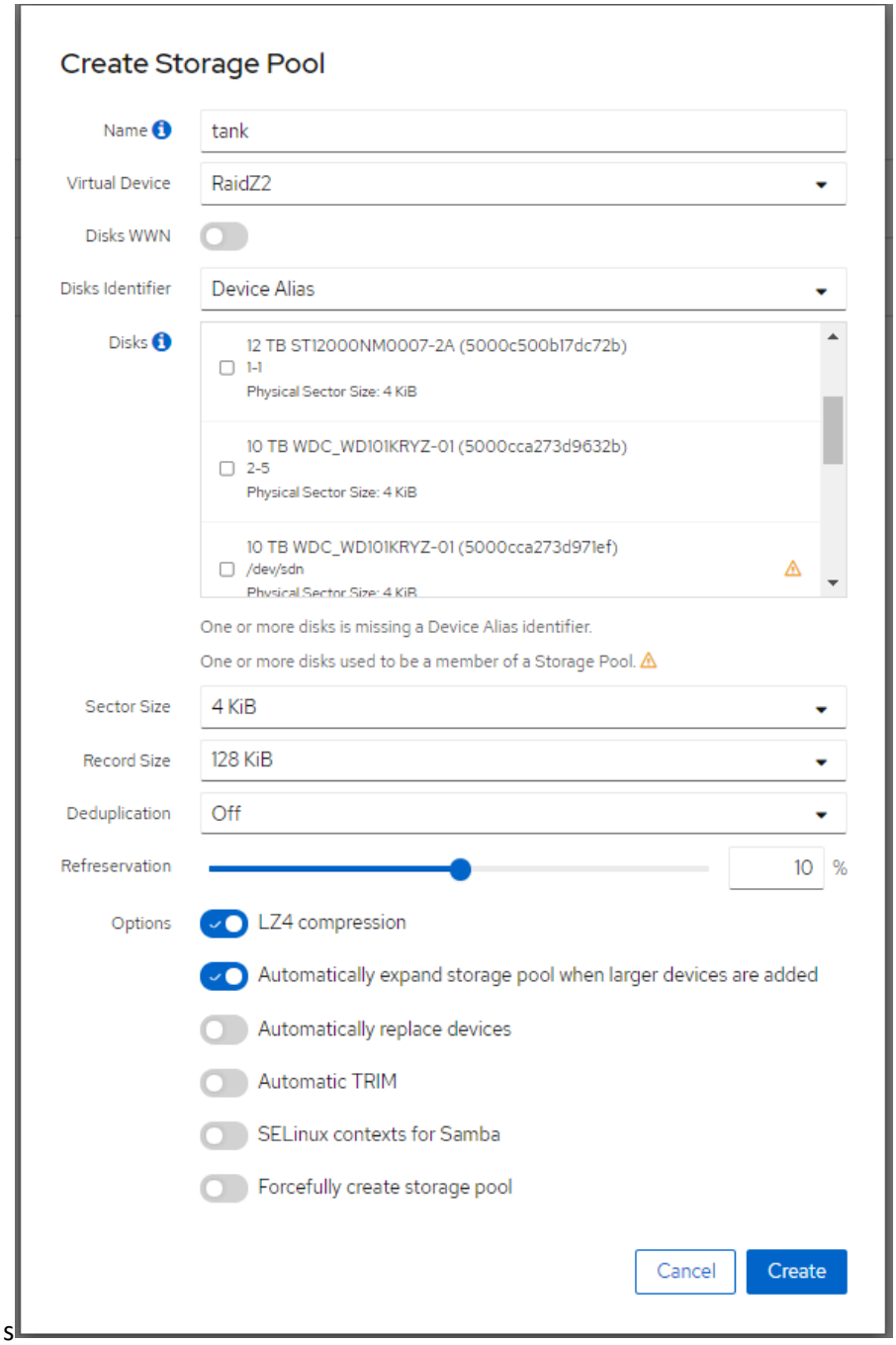


- On the top right of the ZFS page you will notice Storage Pools. Click on this to begin creating your first storage pool. When we create a pool with ZFS, we select the drives we want to be part of the pool. In our case all 15 drives



- Provide a name for your new storage pool, in this test environment we have named the pool tank in this example.
- Select which raid you would like to use under "Virtual Device ". We usually recommend RAIDZ2 which is equivalent to RAID 6 and has two drive redundancy.
- Then select your hard drives and click on "Create" to finish. Ensure you uncheck Disks WWN, and then select "Device Alias" so the pool is created with the Disk Aliasing to easily identify drives (i.e 1-1, 1-2, etc.)

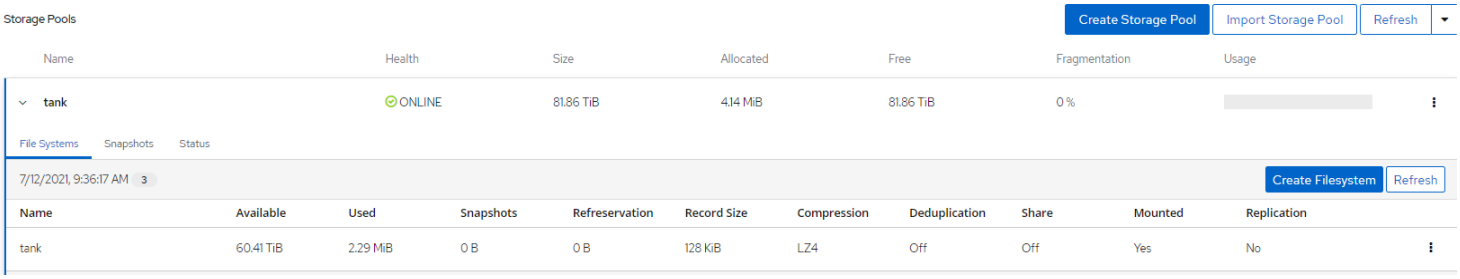
We do not recommend setting “refreservation” below 10%. With this disabled we can write to the ZFS Pool/Dataset and fill it entirely, at which point the pool may be unusable and data could be lost.



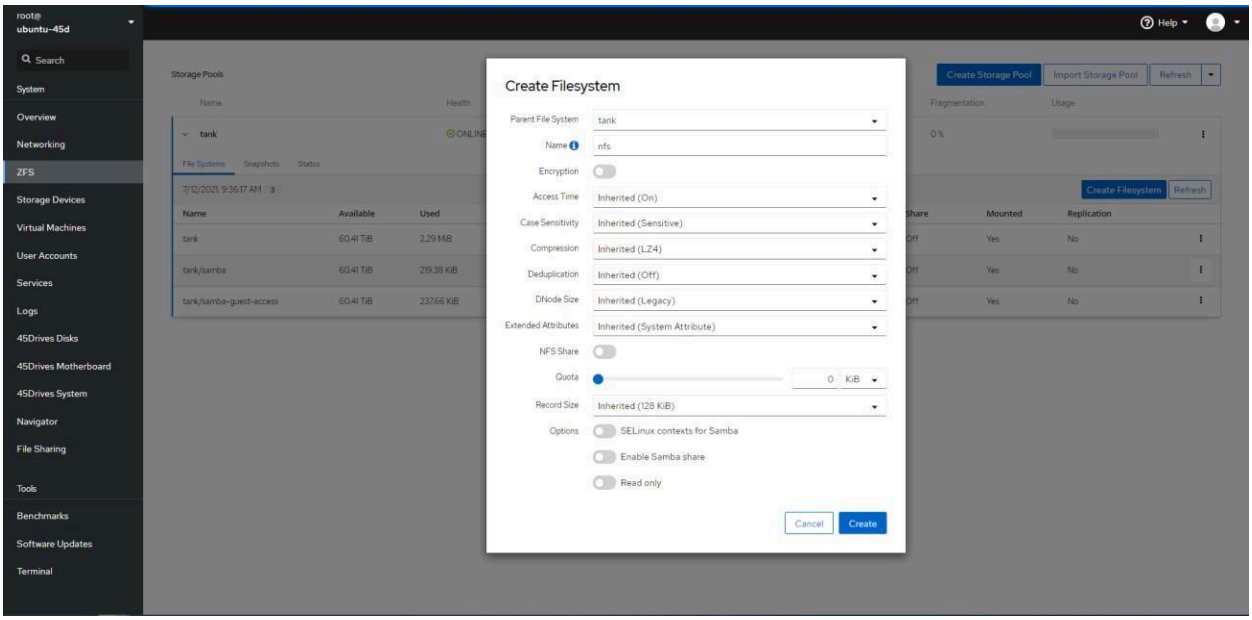
S

CREATING ZFS FILESYSTEM/DATASET

- With your new storage pool created, we can now create some datasets to share out. You will see Create Filesystem. Click on this to continue.



- In the new window that appears, give your new filesystem a name. We can leave all of the options at the default settings. In this example, we have used nfs.



- If you wish you can change those other parameters as per your preference and even set quota as well.
- Once you click on create you will have your dataset created.

45HomeLab
A Division of 45Drives

- Now you will be able to see your ZFS pool, datasets, and drives within the File Systems Tab, and Status Tab within the ZFS module.
- If you are to run zpool status command within the Terminal you should see the output of the ZFS Pool, its VDEVs, and Disks.
- You can create as many datasets you want based on how many shares you are planning to create.

If you are unable to create the pool, ensure the drives you are using are free of any partitions.

Ensure the disks you are using to create the pool are of the same size

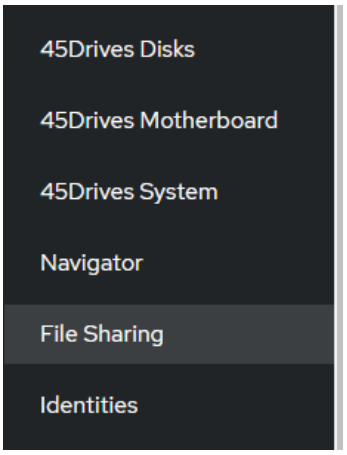
CREATING YOUR SHARES

Once you have your pool and dataset created, the next step is to create network shares. You can either create SAMBA/NFS shares.

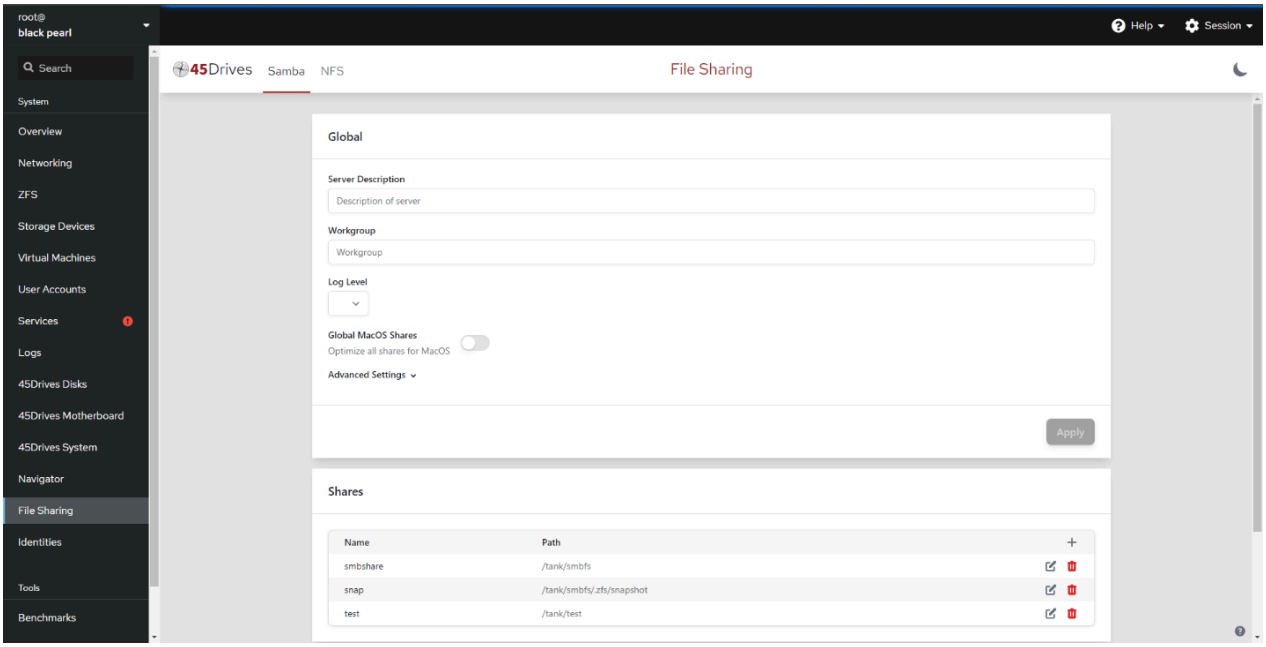
Samba/SMB shares

- Make sure SMB Services Running and Enabled
- SMB Ports Open on Firewall (133/tcp, 445/tcp and 137/udp, 138/udp)

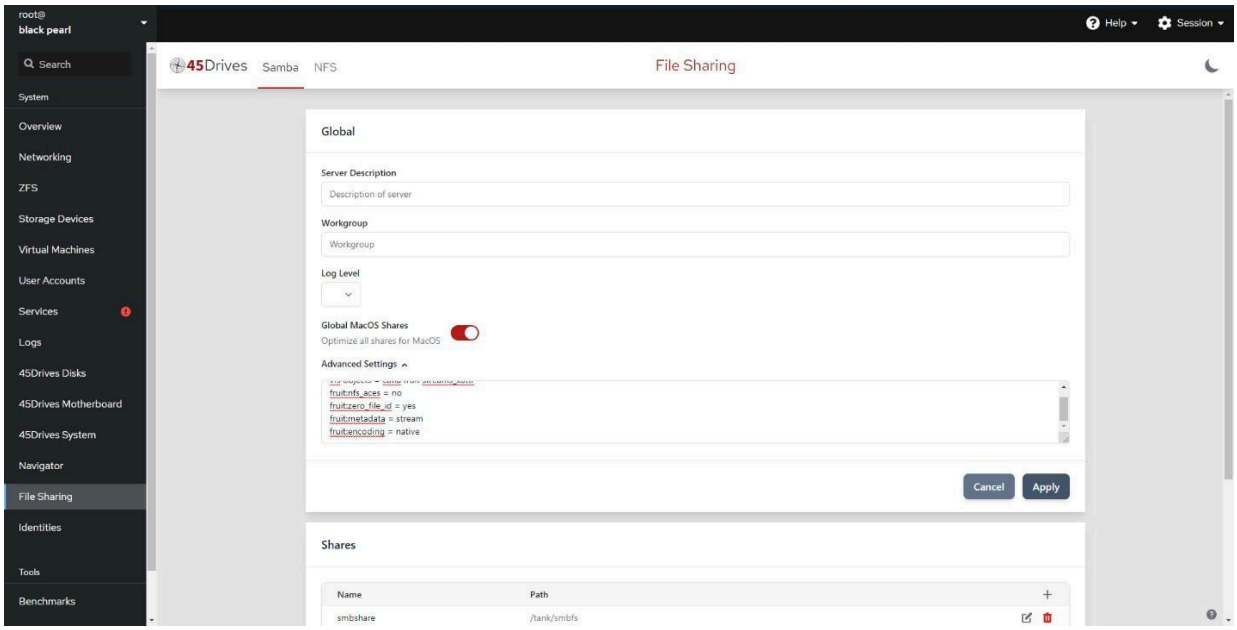
- In Houston UI, navigate to the File Sharing tab. And click on the Samba tab, if not selected.



- Once here, we can begin configuring our SMB Shares



- First, we can add any options to the Global SMB configuration.
- Here we can change the Server Description, the Workgroup, Log Level, and add any additional parameters to the SMB configuration in the Advanced Settings box by clicking the down arrow. For example, here we've added a few parameters to help with MacOS performance on an SMB share.



45HomeLab
A Division of 45Drives

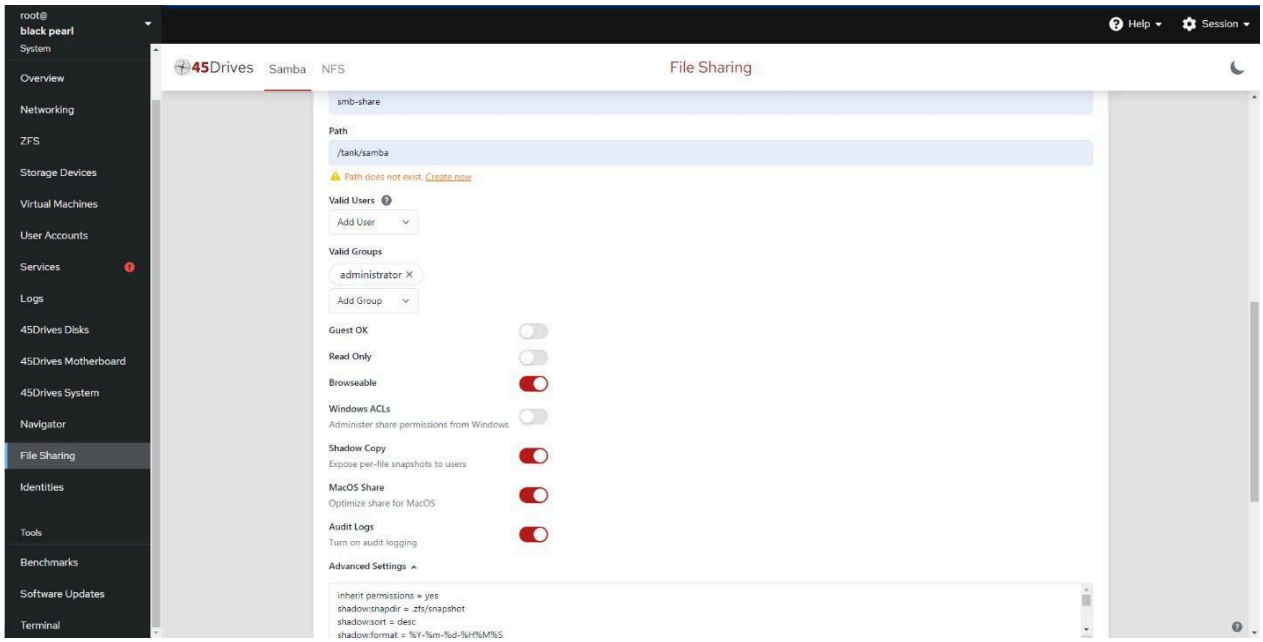
- Next, we can create an SMB share, we can scroll down to Shares and select the Plus button to begin.
- Here we can set the SMB Share Name, a Description, and the Path to be shared out. It will be usually /pool_name/dataset_name (you can now create the path here instead of ZFS)

- if we are domain joined, we can select to use Windows ACLs.

- We can specify any Valid Users and Valid Groups, allow Guest Access, make the SMB share Read Only, make it Browsable, and enable Windows ACLs, as well as some other options.

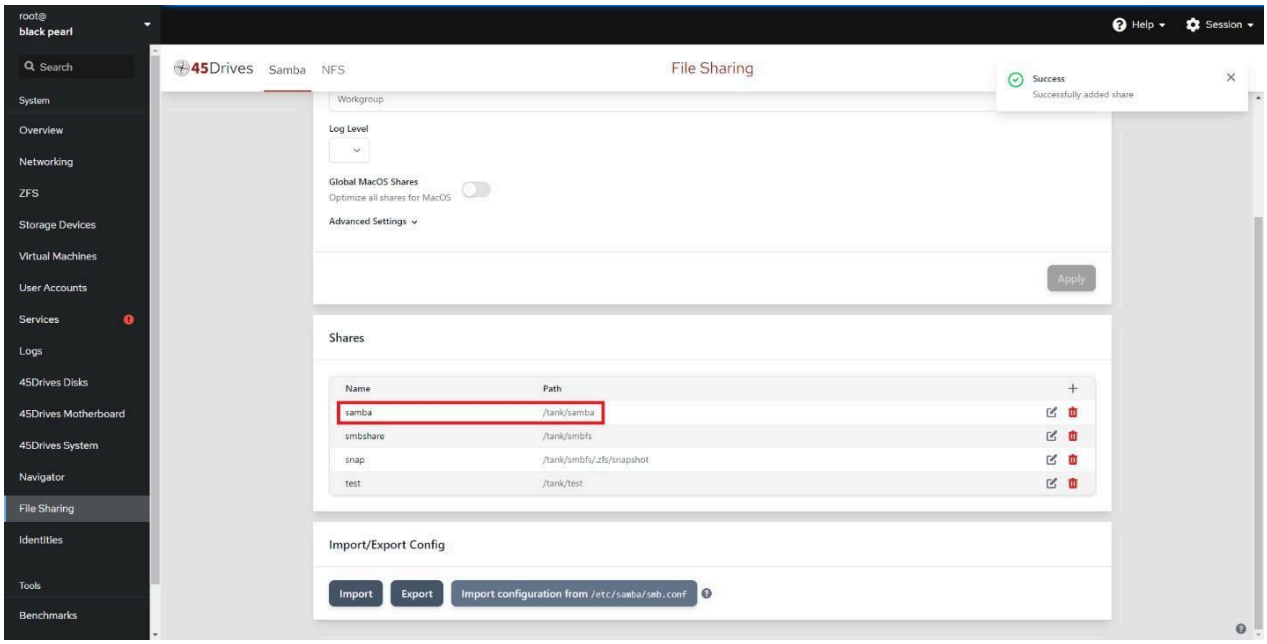
Adding options to the SMB share

- We can also add any additional options in the Advanced Settings box via the drop-down arrow.
- In the Advanced Settings box, we entered “inherit permissions = yes” ourselves. This is to make sure that the permissions are inherited from the parent folder.
- There are a few preselected options we can enter into the Advanced Settings box by selecting one of the buttons below: Shadow Copy, MacOS Share, Audit Logs.
- Here we can see we’ve created a share called “samba” with a description of “smb-share“. It is pathed to our ZFS dataset at “/tank/samba“. We have added the “administrator” to Valid Groups, and left the share Browsable.

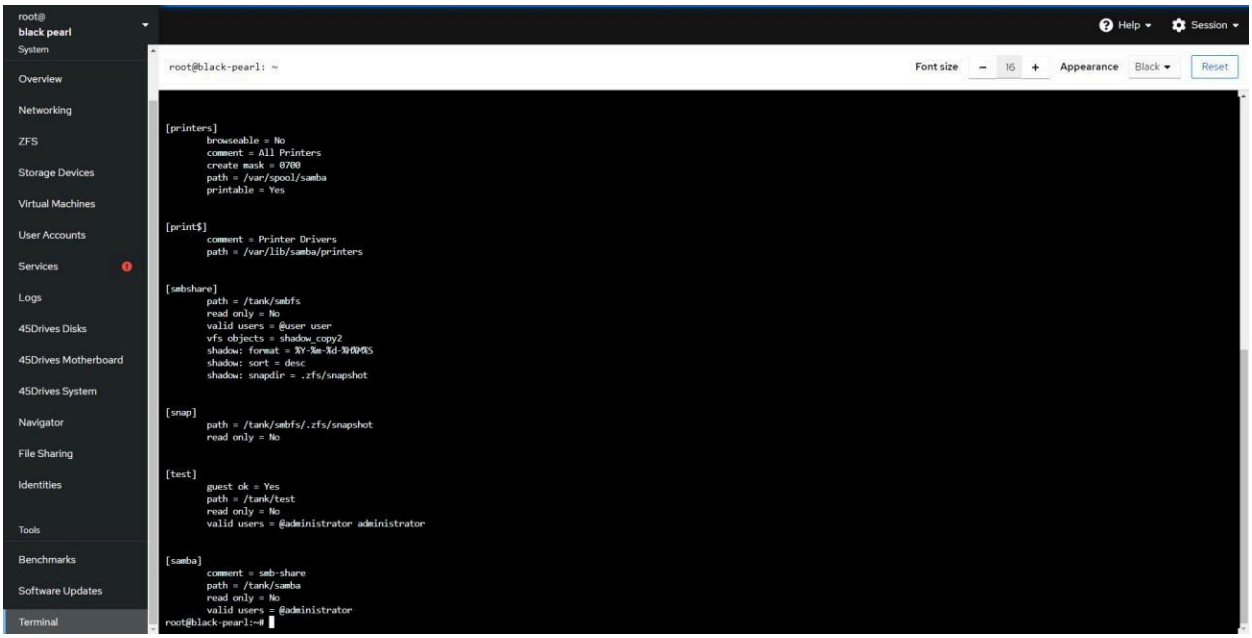


- Here we can see our SMB share is created.

45HomeLab
A Division of 45Drives



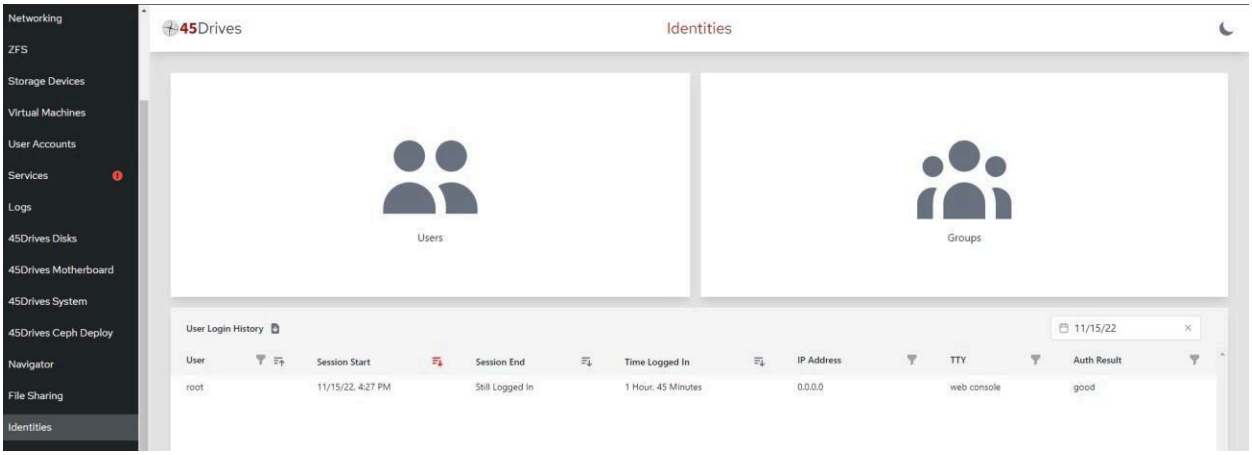
- If you were to run **"testparm -s"** on the command line you will see your samba share has been added and configured properly in its own section.



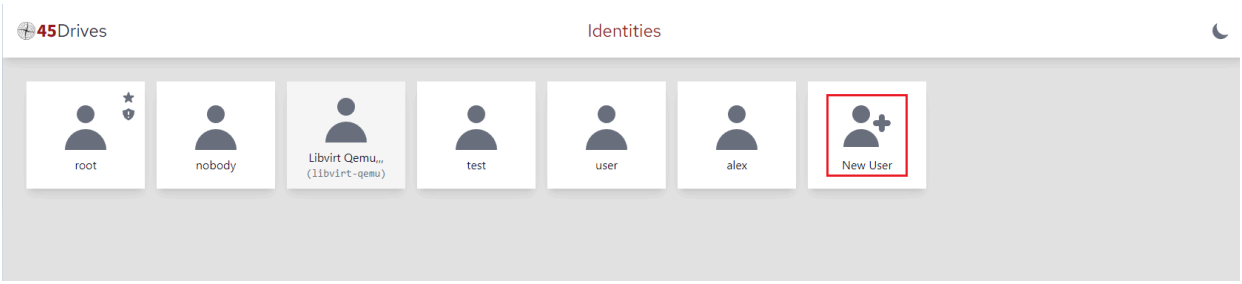
Setting up Local User Access SMB Shares

This is to create local users and groups and add them to provide access to your samba share.

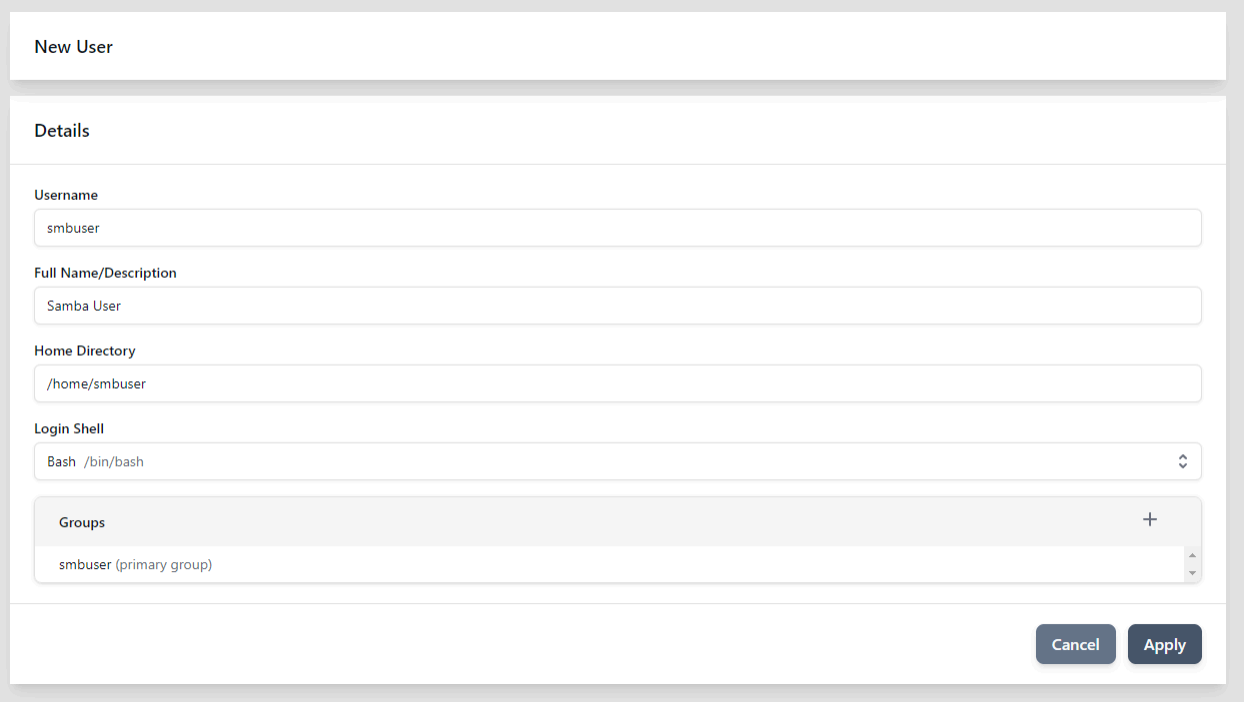
- Go to the identities tab in Houston



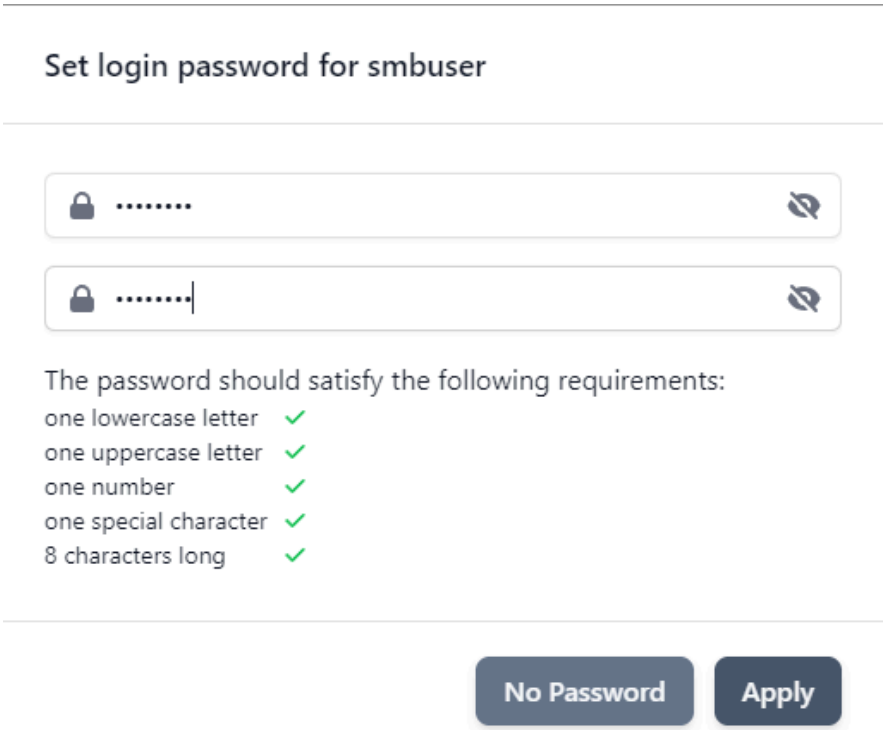
- Click the 'Users' section, this will open a page with a list of users. Select "New User".



- Fill in the Username, Full Name/Description. Then click "Apply". It is also possible to assign this new user to a group at the bottom of the prompt. If applicable it can be done here, or a user can be assigned at a later time.



- A prompt will come up to set the user's password. This is a local Linux password, it can be used to SSH into the machine, or to authenticate to Houston.

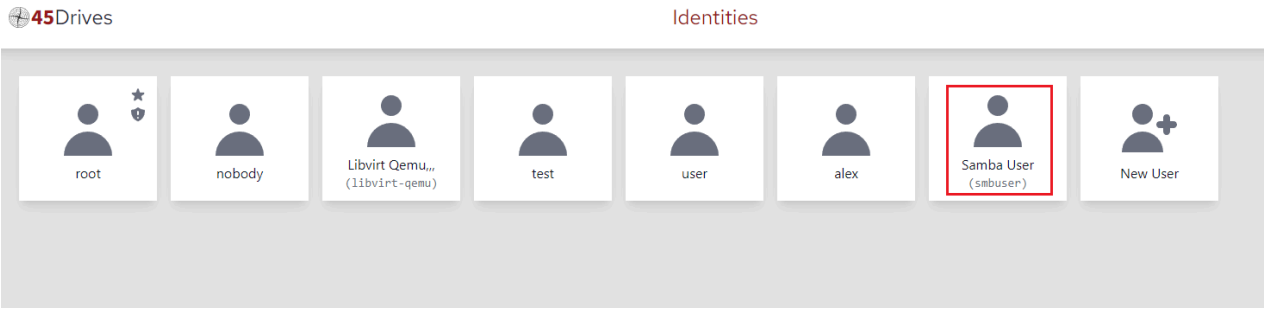


- The user generation will complete successfully. The user should now be able to authenticate to Houston or SSH with that password.

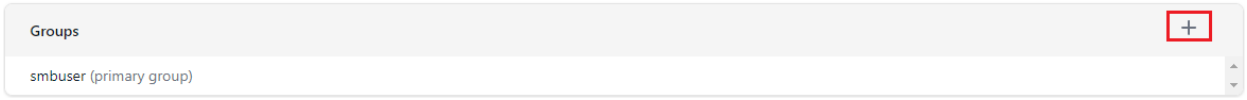
- The below sections will detail assigning a user to a specific group, and setting an SMB password if applicable.

Assigning a Linux Group to a User

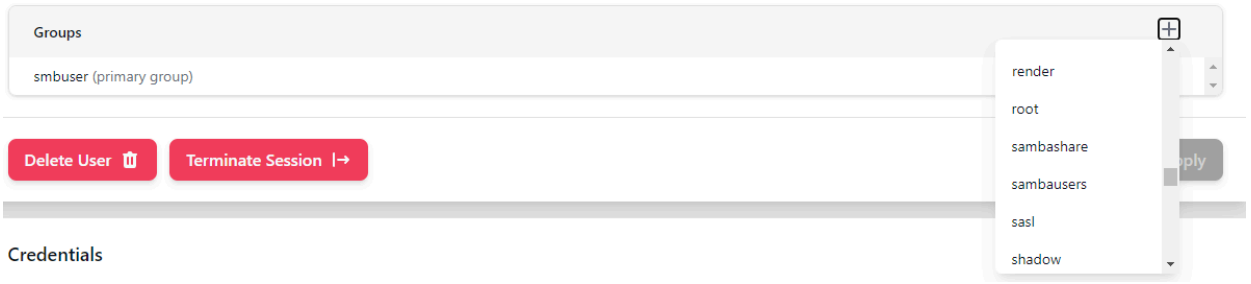
- In the “Users” section the new user should be seen. Click on the user to assign a group.



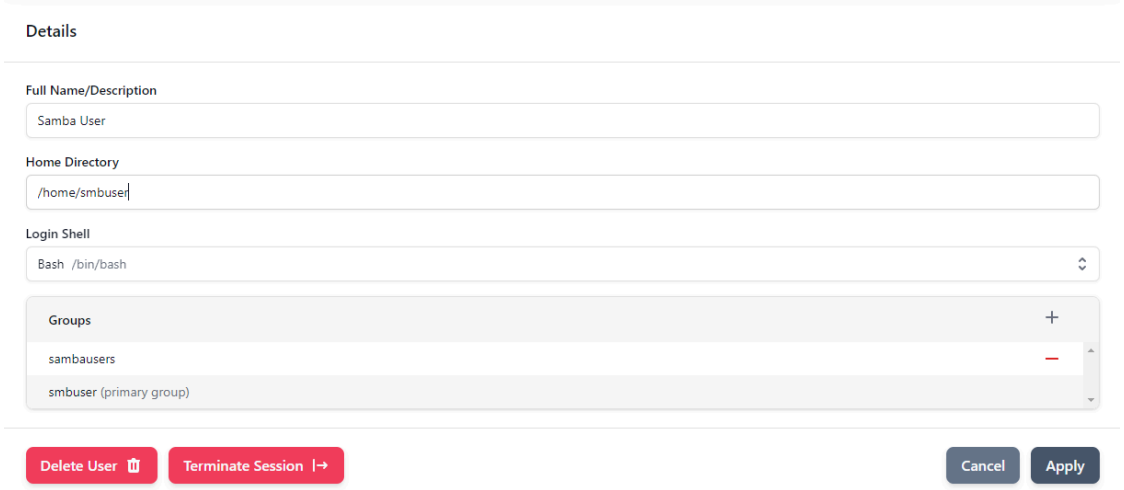
- Click on the “+” under the Groups section.



- A dropdown will pop up and you can now select a group to assign to the user.

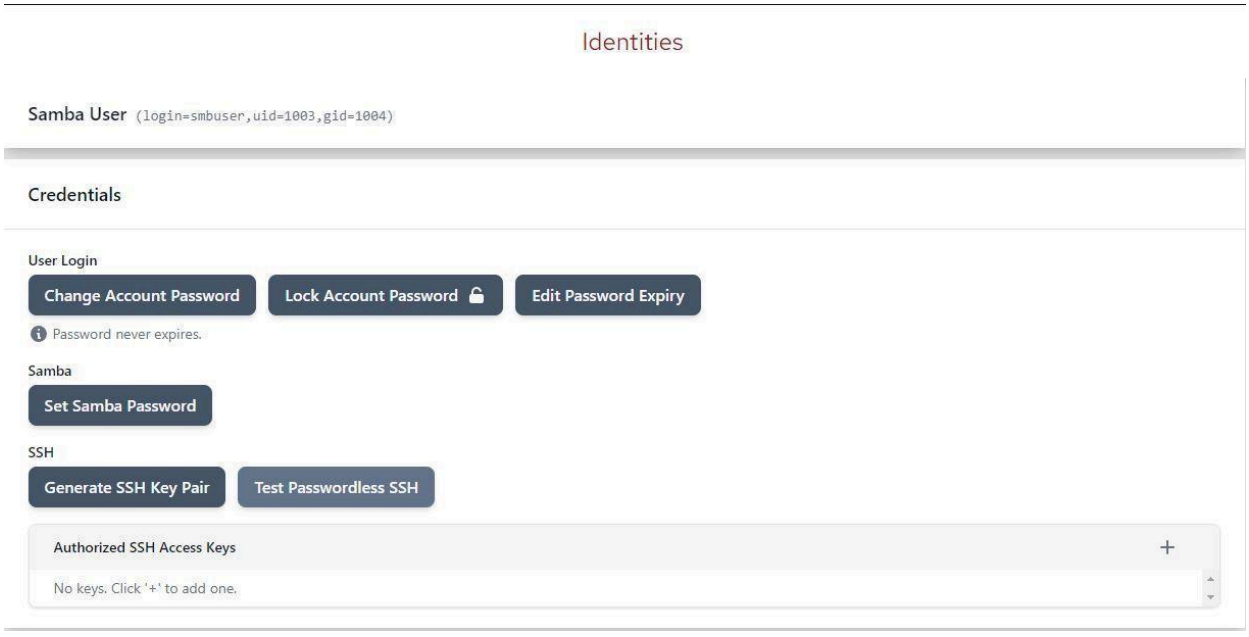


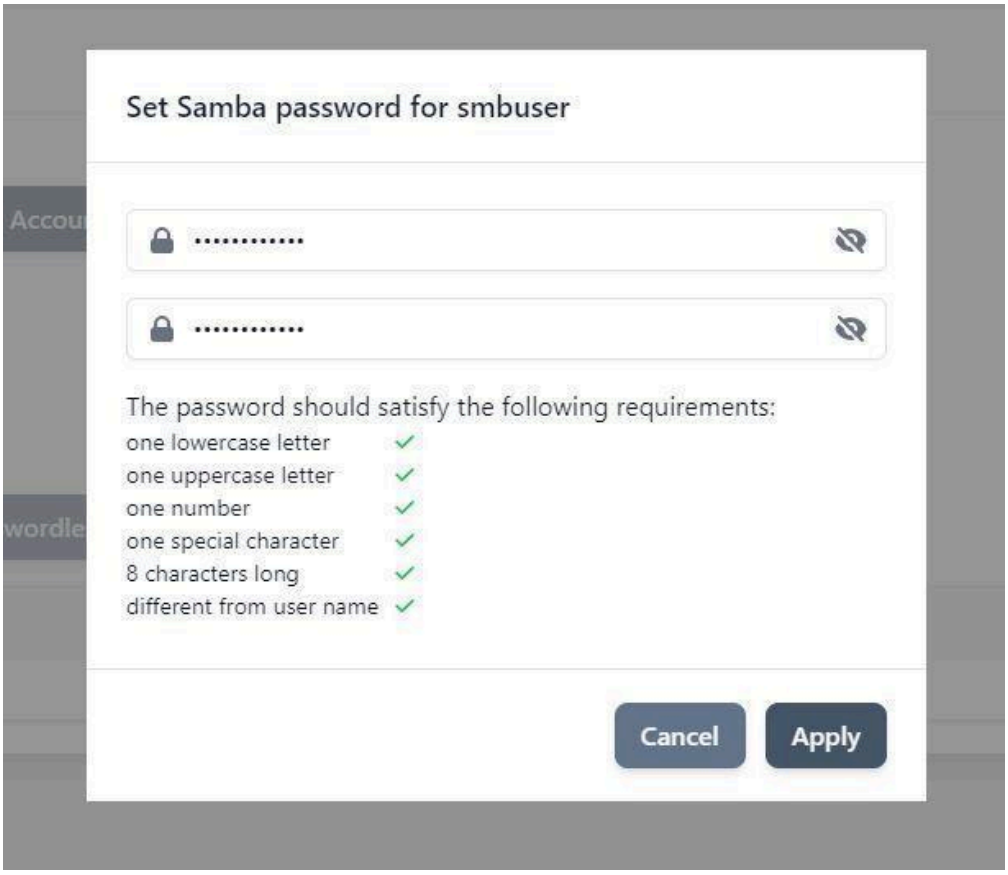
- The newly added group should be displayed under the Groups section. Click “Apply” to save changes.



Configuring a Samba Password for a User

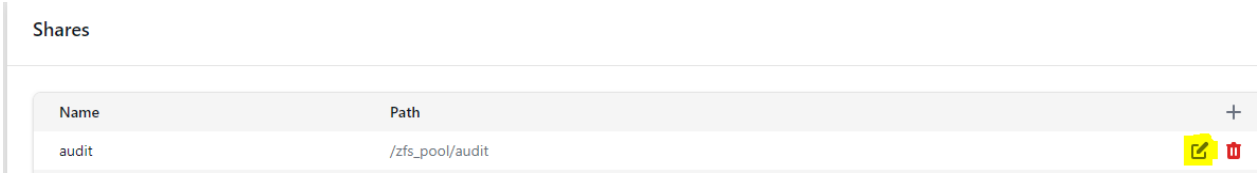
- Select the user you wish to assign a Samba password. The Samba password will need to be set to allow users to authenticate to Samba shares. It can be the same as the Linux password, but does not have to be. Click on “Set Samba Password”. Enter a password and click “Apply”.



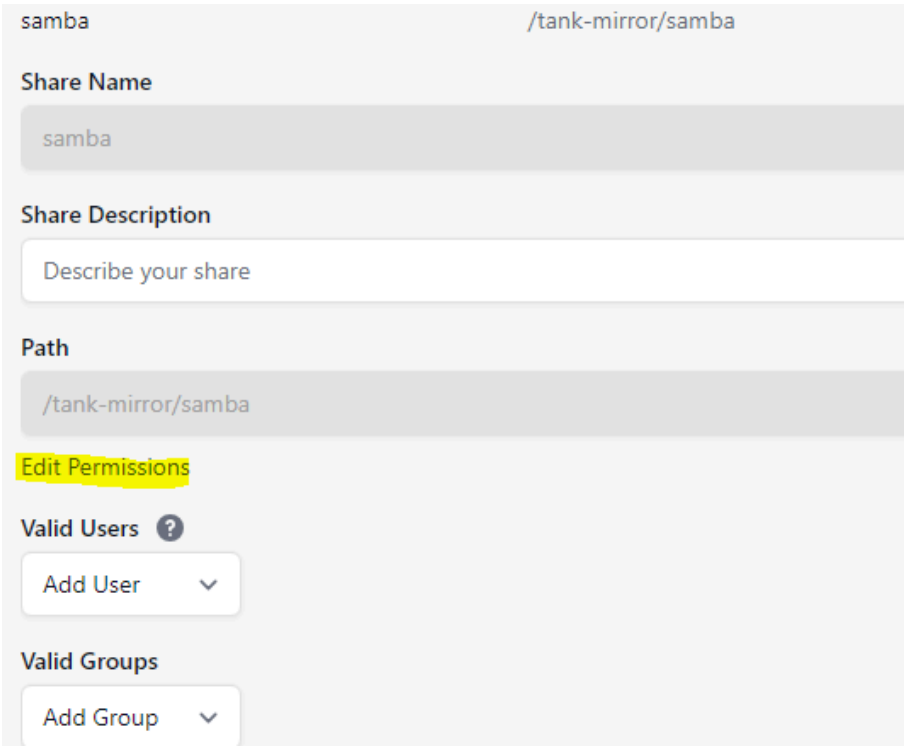


Set SMB permissions using local users and group

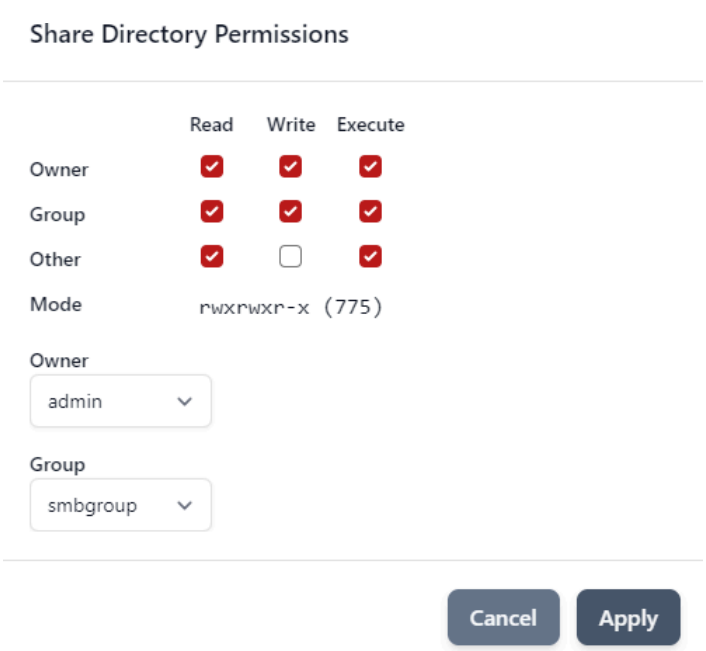
- Click on the edit button of the share you want to add the user to grant the permissions.



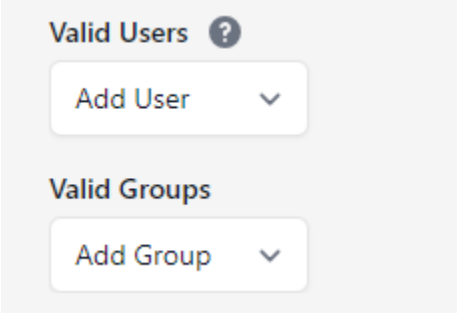
- Click on edit permissions



- Select the owner and the group for the share and set the permissions as well as per your preference.



- You can also select valid users and groups if you want to provide access to multiple of them.

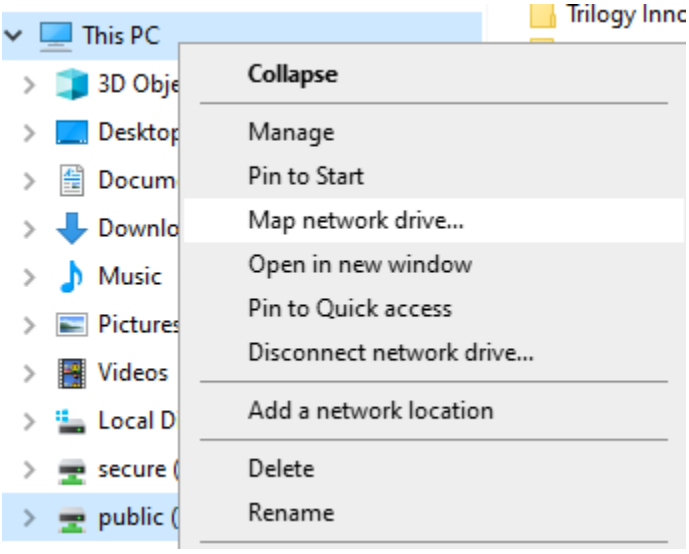


- Now you can verify by connecting the share.

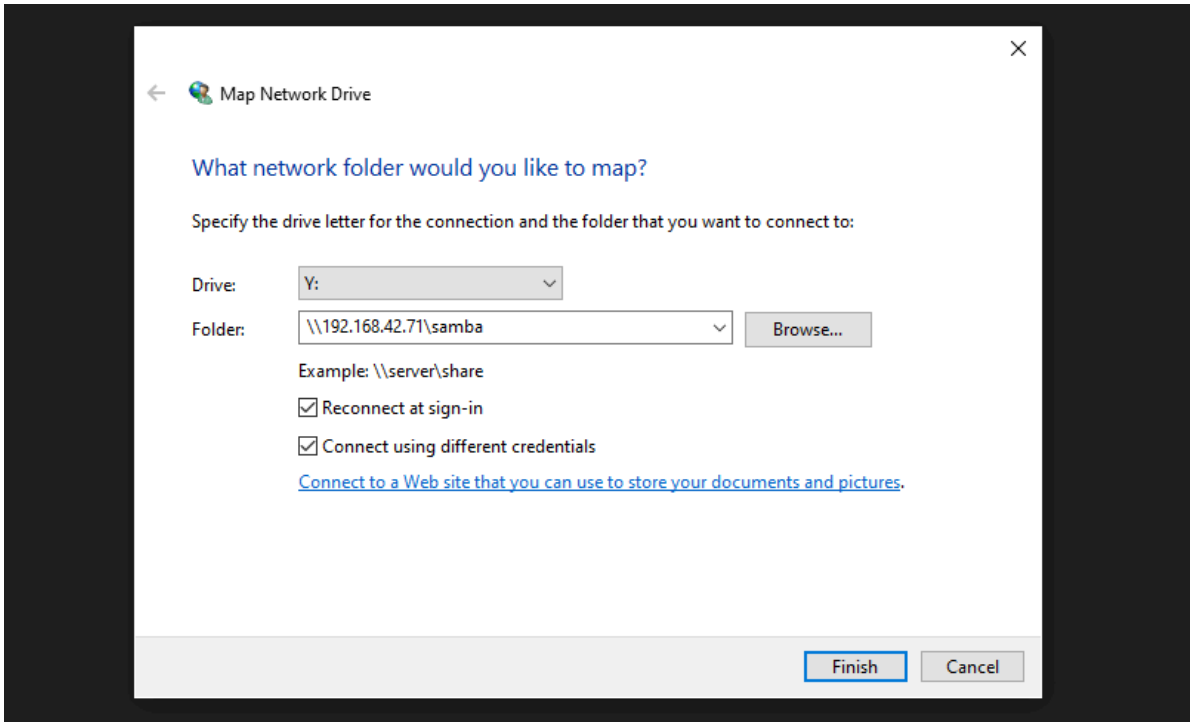
Connecting to SMB Share on Windows and MacOS

Connecting to SMB Share on Windows

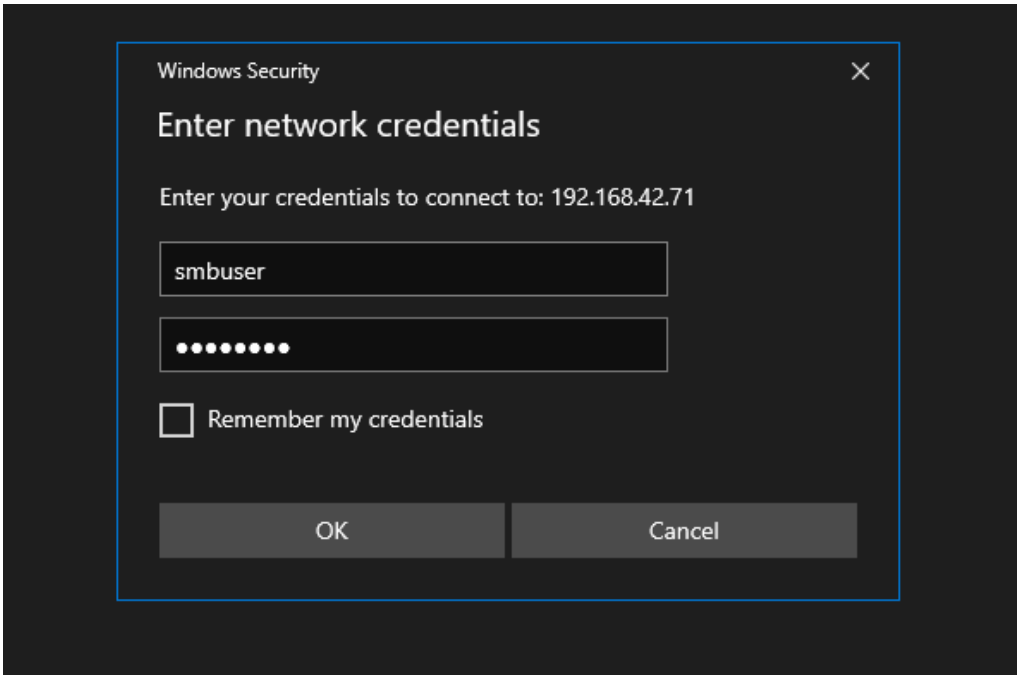
- On a Windows client, go to This PC in File Explorer.
- Right click and select Map network drive.



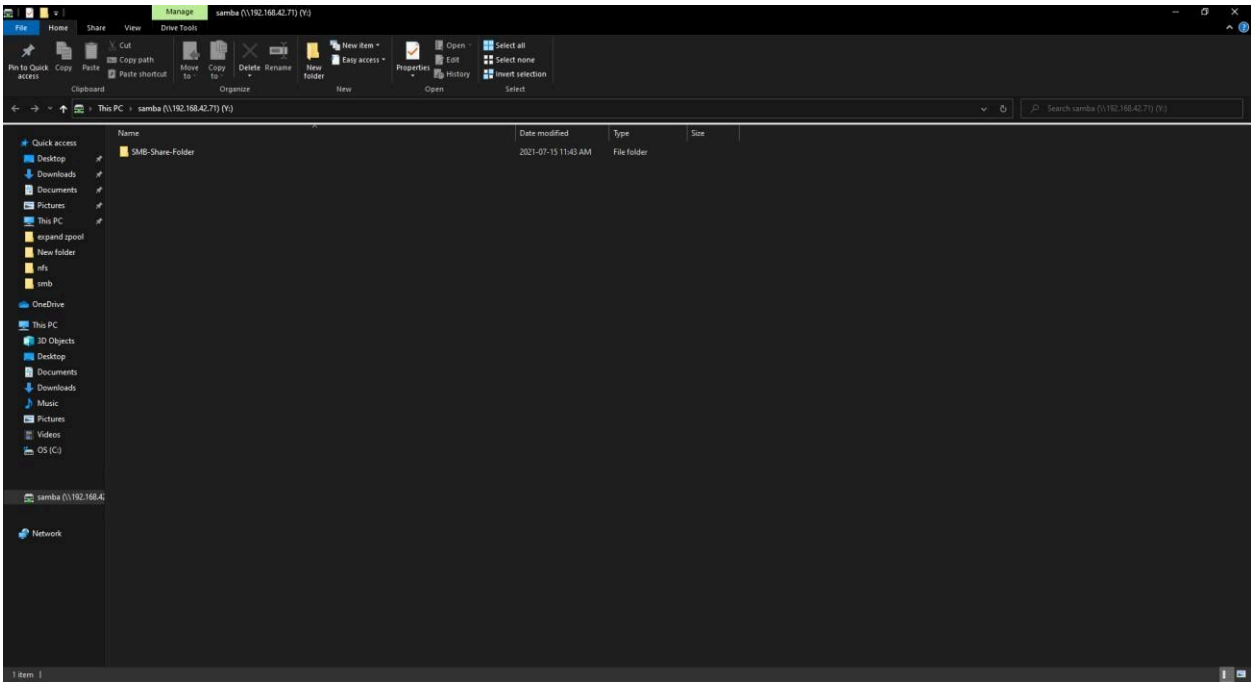
- In this menu, enter the server IP address in the following format **\\SERVER-IP\SHARE-NAME** and select **Connect using different credentials**.



- Enter the login to connect to the SMB share.



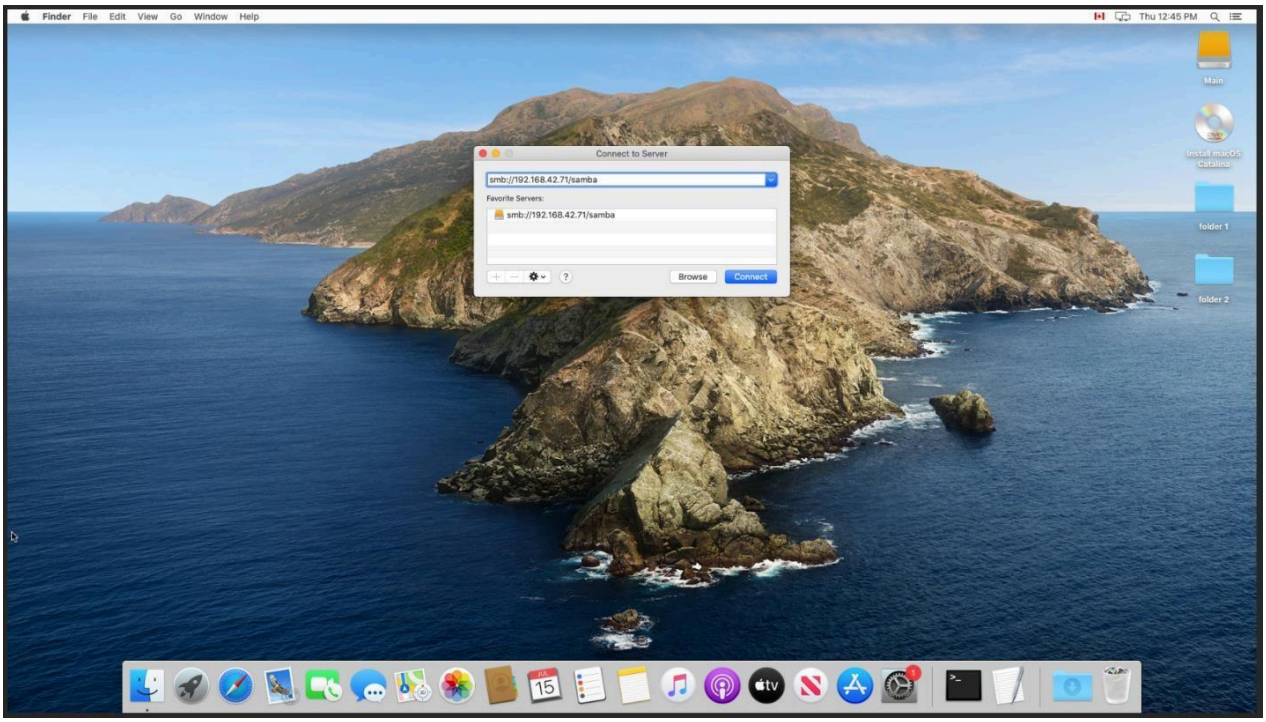
- Here we can see our SMB share is connected, and we can create a folder.



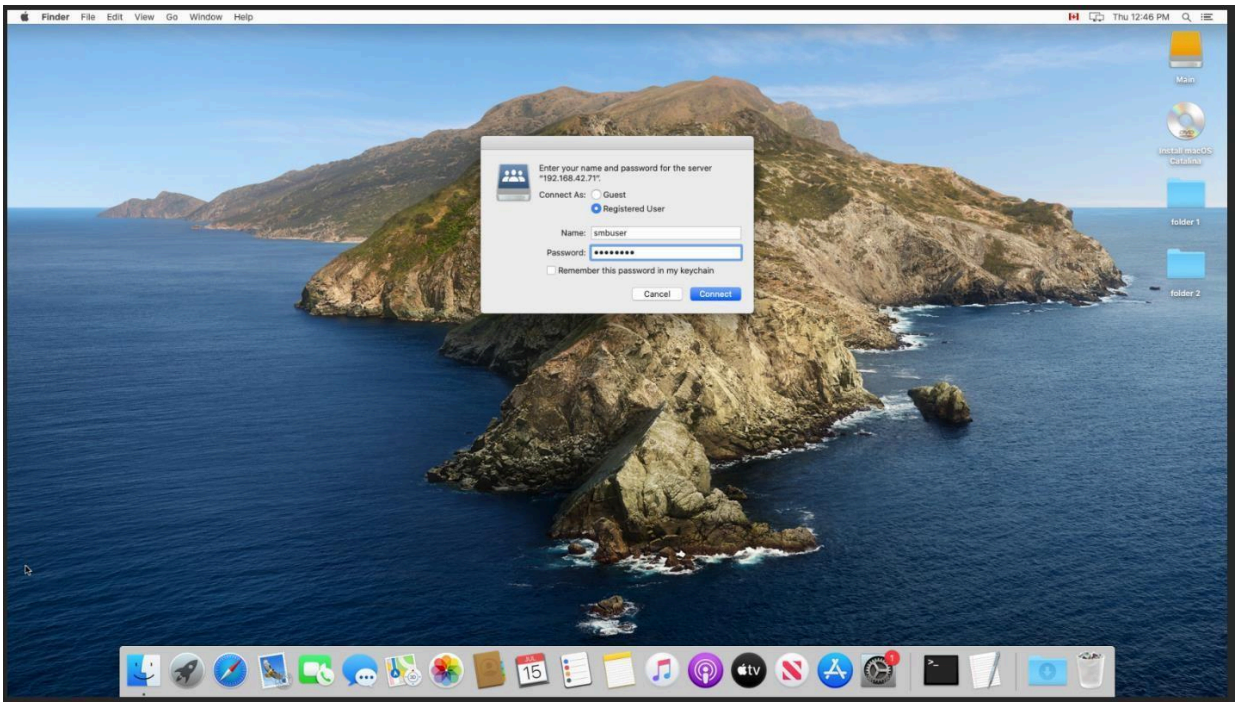
Connecting to SMB Share on MacOS

- At the toolbar on the top of the screen, press Go and then “Connect to Server…“.

- In the address bar, enter **smb://SERVER-IP/SHARE** and click **Connect**. We can also select the plus at the bottom to save this information if we need to reconnect.



- Enter the login to connect to the SMB share.

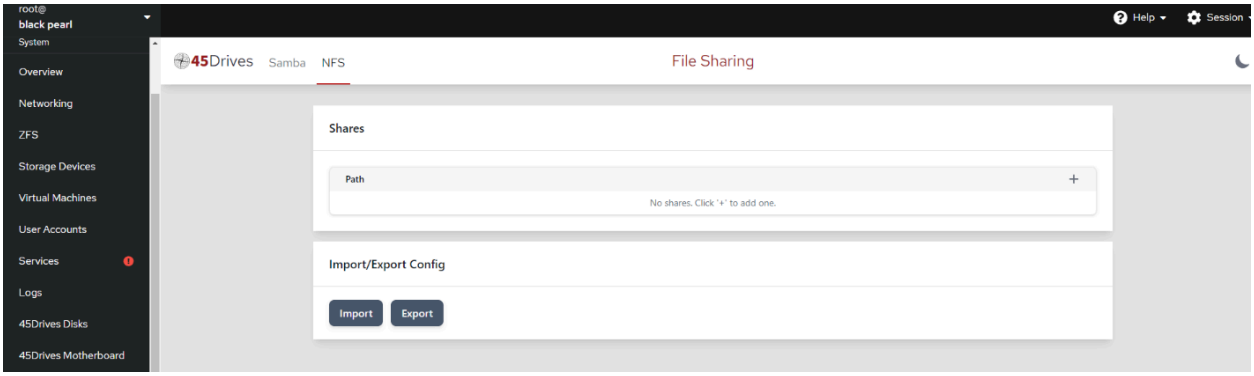


- Here we can see our SMB share is connected, and we can create a folder.

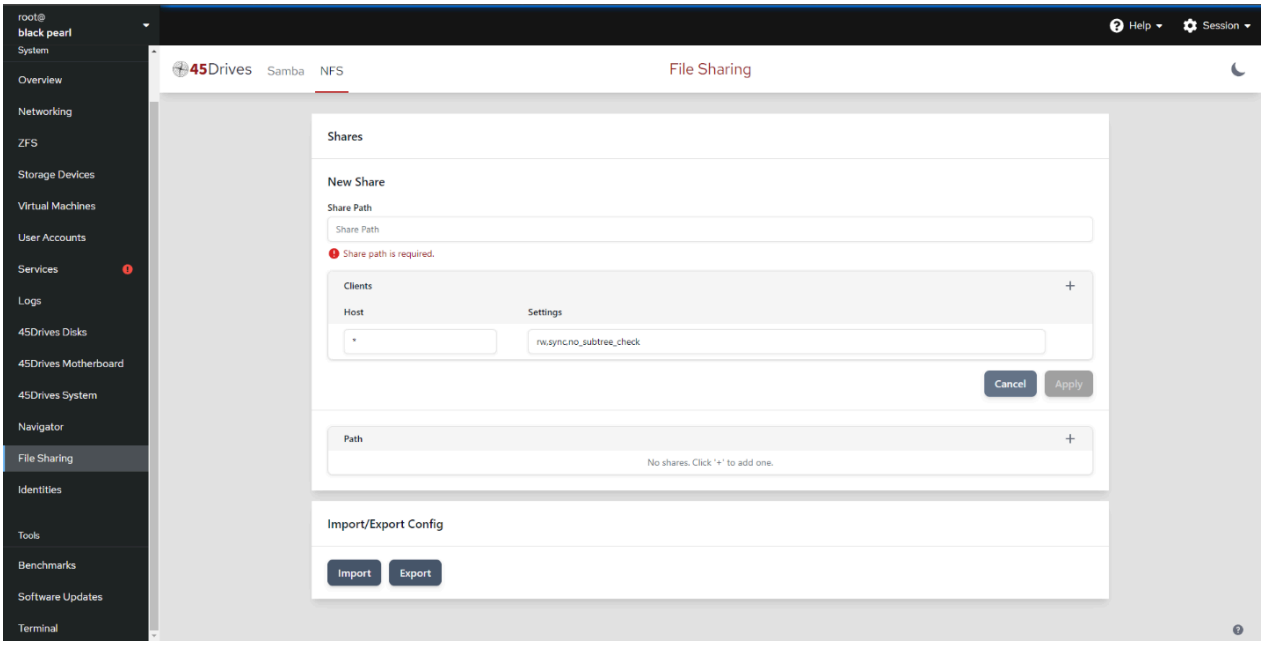
Managing NFS in Houston UI

- NFS Packages Installed
- NFS Services Running and Enabled
- NFS Ports Open on Firewall (2049/tcp, and 2049/udp if NFSv3)

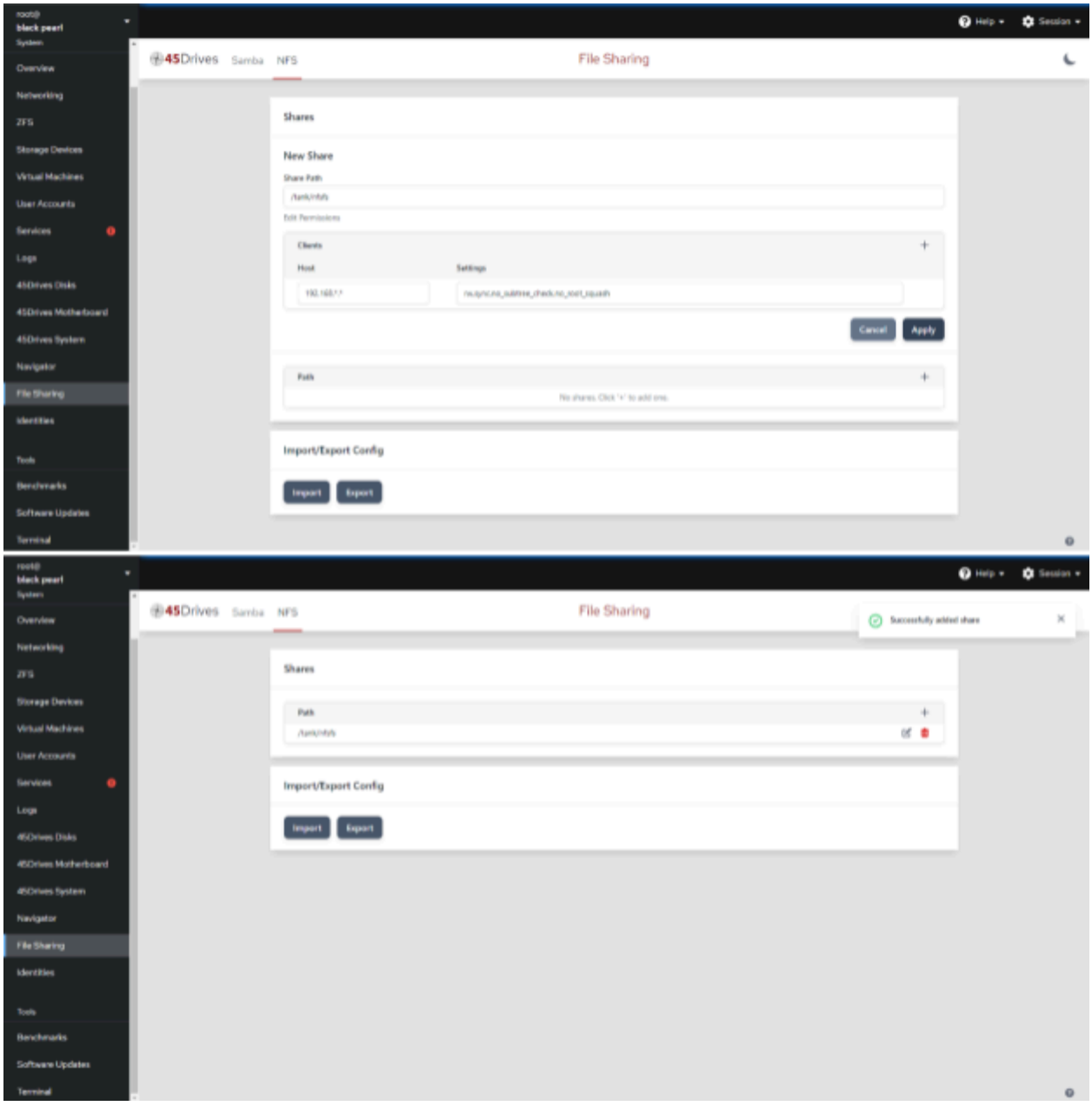
- In Houston UI, navigate to the File Sharing tab.
- Once in that page, we can select the NFS Tab to begin configuring our NFS shares.



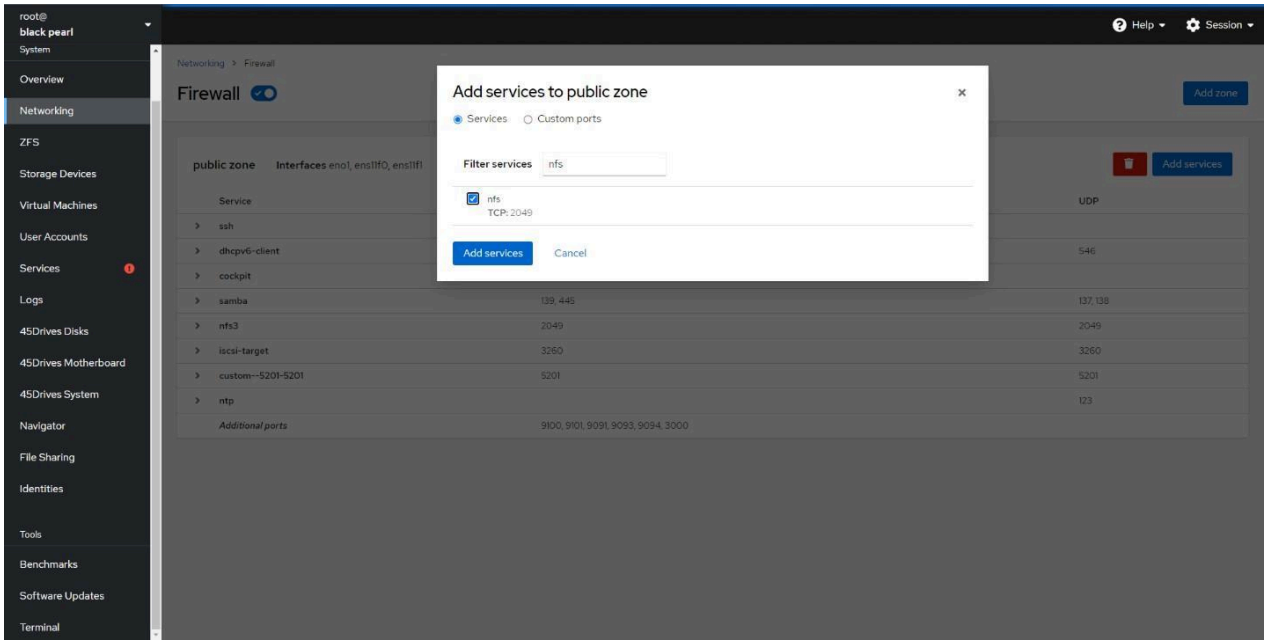
- We first begin to create our NFS export by selecting the Plus Icon on the right.
- We can define the path to be shared out. It would be usually /pool_name/dataset_name
- select the range of IPs for client access, and add any additional NFS options to our share.
- If we leave Client IP empty, it will default to everyone, and if we leave Options empty, it will default to 'rw,sync,no_subtree_check'



- Here we create an NFS share, with a path to our NFS dataset at /tank/nfsfs. We have set the Client IP to be available to **192.168.*.*.** For settings you can give "**rw,sync,no_subtree_check,no_root_squash**"



- If you haven't opened the firewall ports then follow the below steps



Mounting NFS Share to Linux Client

Here we will discuss the process of mounting an NFS share to a Linux client and to mount on reboot.

- Go the terminal and run the command below to install the nfs package

  install nfs-utils

- To mount an NFS share, first create a directory to mount it to.

  mkdir /mnt/(mount_point)

eg:

```
mkdir /mnt/nfs_share
```

- Now use this command to mount the share. Edit the fields for your specific case, i.e. Server IP, Pool Name, and Share name. See example below.

  mount -t nfs {ServerIP}:/(pool_name)/(nfs_share_name) /mnt/(mount_point)

```
mount -t nfs 192.168.35.39:/tank/nfs_sharetest /mnt/nfs_share
```

Add Mount on Reboot

- To allow the share to mount on reboot, you will need to edit the fstab. You can use your preferred text editor, here we have used vim.

  vim /etc/fstab

- Add the mount point in the format see below.

  {ServerIP}:/(pool_name)/(share_name) /mnt/(mount_point) nfs defaults,_netdev 0 0

```
# /etc/fstab: static file system information.
#
# Use 'blkid' to print the universally unique identifier for a
# device; this may be used with UUID= as a more robust way to name devices
# that works even if disks are added and removed. See fstab(5).
#
# <file system> <mount point>   <type>  <options>       <dump>  <pass>
/dev/disk/by-id/md-uuid-5bef0df4:f65beb63:7f835222:41ec8940 none swap sw 0 0
# / was on /dev/md125p1 during curtin installation
/dev/disk/by-id/md-uuid-2a10c4dd:c02a5098:0ee79f38:2078a857-part1 / ext4 defaults 0 0
# /boot was on /dev/md/swap during curtin installation
/dev/disk/by-id/md-uuid-5fbc9b25:25381cd8:2b507526:40ea9196 /boot ext4 defaults 0 0
/swap.img       none    swap    sw      0       0
/dev/RBD-VOLGROUP/RBD-LVM       /mnt/rbd        xfs     defaults,_netdev 0 0
192.168.35.39:/tank/nfs_sharetest /mnt/nfs_share        nfs     defaults        0 0
```

Verify Share Mounted

- To ensure the share has mounted, you can run the command **df**.

```
root@ubuntu-45d:~# df
Filesystem                      1K-blocks     Used   Available Use% Mounted on
udev                             65907972        0    65907972   0% /dev
tmpfs                            13190628     4004    13186624   1% /run
/dev/md126p1                    117313456 16268152    95043028  15% /
tmpfs                            65953124        0    65953124   0% /dev/shm
tmpfs                                5120        0        5120   0% /run/lock
tmpfs                            65953124        0    65953124   0% /sys/fs/cgroup
/dev/md125                        4057792   221160     3610792   6% /boot
tank                          36654107136      256 36654106880   1% /tank
tank/samba-windows-acl        29137914624      256 29137914368   1% /tank/samba-windo
tank/nfs                      29137914624      256 29137914368   1% /tank/nfs
tank/samba                    29137914624      256 29137914368   1% /tank/samba
tank/smb-windows              29137914624      256 29137914368   1% /tank/smb-windo
tmpfs                            13190624        0    13190624   0% /run/user/0
192.168.35.39:/tank/nfs_sharetest  2798961664        0  2798961664   0% /mnt/nfs_share
```


45HomeLab
A Division of 45Drives

Verify Mount on Reboot

- To ensure the share will mount on reboot after editing the /etc/fstab, unmount the share.

```
umount /mnt/nfs_share
```

- Remount the share using the following command, this command will mount all shares in the /etc/fstab file.

```
mount -a
```

- Run the command **df** again to ensure the share was mounted.

SETUP ISCSI STORAGE

Prerequisite:

1. Install Ansible

Rocky Linux

```
dnf install -y epel-release
dnf install -y ansible
```

Ubuntu

```
apt update
apt install -y ansible
```

2. zpool created

Refer to zpool creation section in manual

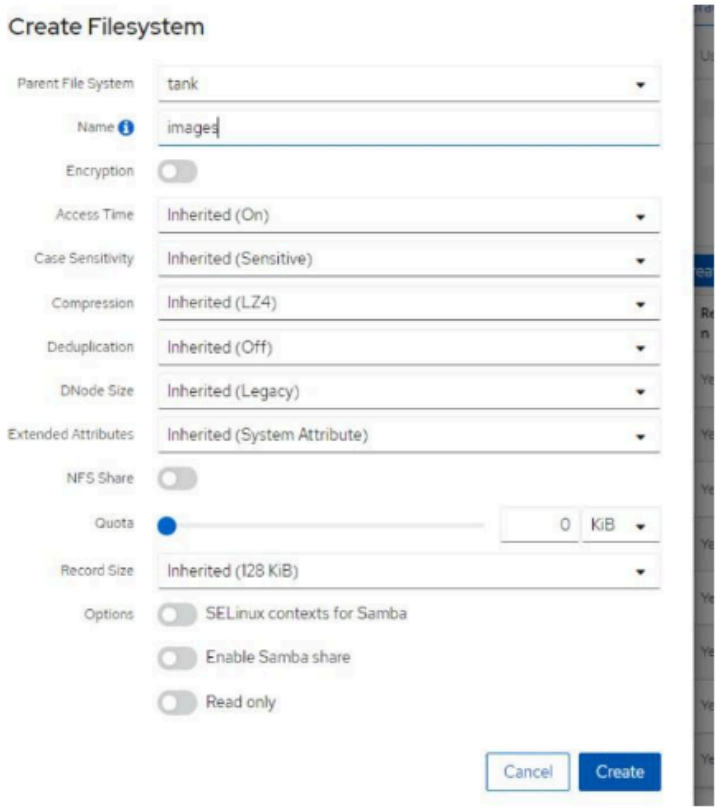
3. ssh key generated and copied to local host in terminal run ssh-keygen then press enter until the command field is blank again then run ssh-copy-id localhost and enter the password of your user

4. image file created in zpool

Create a ZFS dataset with your naming scheme of choice, in this case we'll use "images".

5. /etc/hosts file correct

ensure /etc/hosts file has entry that has hostname and IP along with local 127.0.0.1 local hosts



```
[root@hl ~]# cat /etc/hosts
127.0.0.1   localhost localhost.localdomain localhost4 localhost4.localdomain4
::1         localhost localhost.localdomain localhost6 localhost6.localdomain6
192.168.199.53 hl
[root@hl ~]# ip a
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
       valid_lft forever preferred_lft forever
2: eno1: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc mq state DOWN group default qlen 1000
    link/ether 3c:ec:ef:6b:59:64 brd ff:ff:ff:ff:ff:ff
    altname enp96s0f0
3: eno2: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
    link/ether 3c:ec:ef:6b:59:65 brd ff:ff:ff:ff:ff:ff
    altname enp96s0f1
    inet 192.168.199.53/16 brd 192.168.255.255 scope global noprefixroute eno2
       valid_lft forever preferred_lft forever
```

- Install iscsi-ansible playbook package then navigate go to /usr/share/iscsi-ansible/
- edit the hosts file to point to the host name
- cd groups_vars and run cp all.yml.sample all.yml
- We'll see three options, for a single server deployment, that is not offline and we want the ansible playbook to open the necessary firewall ports we won't change anything

```
dnf install iscsi-ansible
apt install iscsi-ansible # Ubuntu
```

- Edit the all.yml to reflect the below for single server deployment.
- Run the following commands in the terminal:

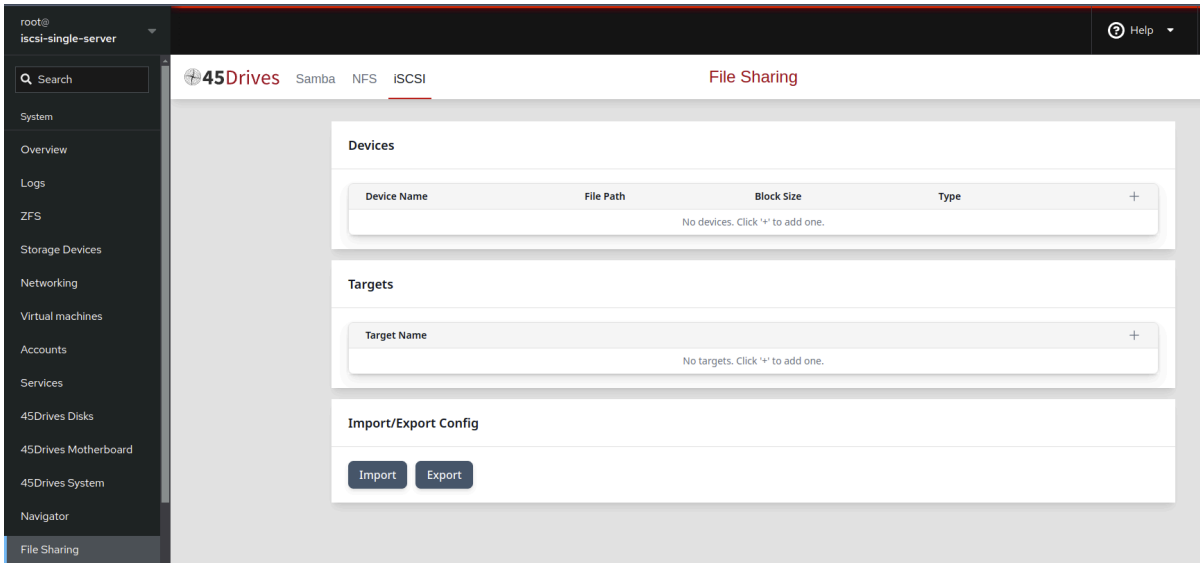
```
cd /usr/share/iscsi-ansible
```

- Edit group_vars/all.yml

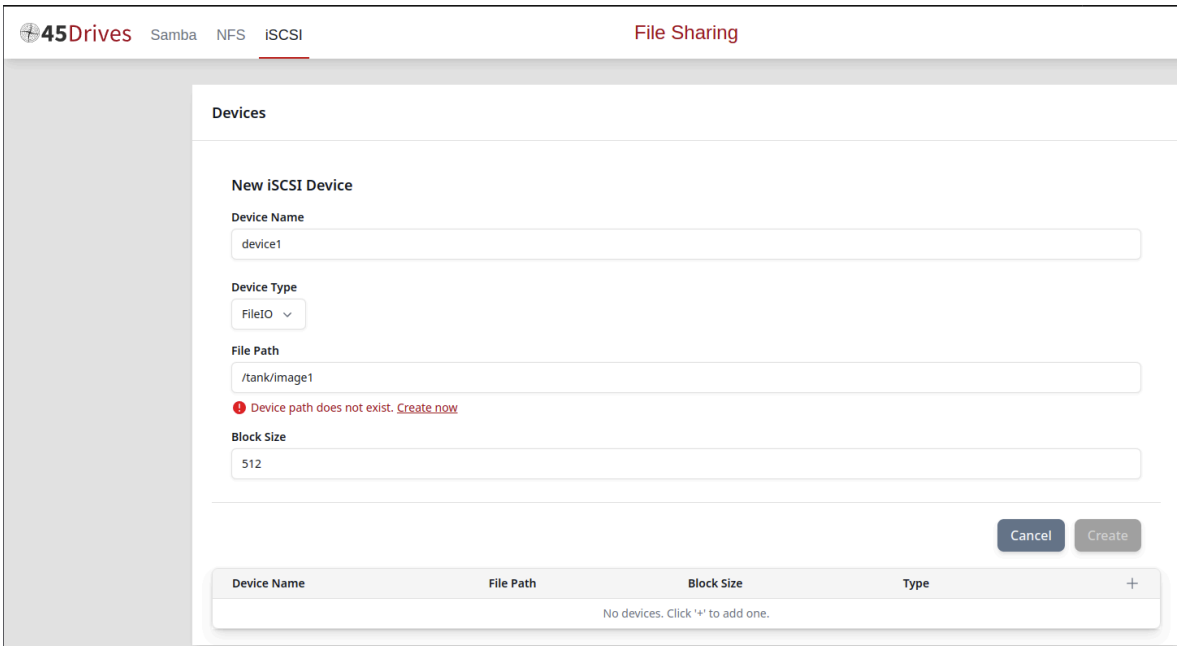
```
---
offline_install: false
manage_firewall: true
clustered: false
```

```
cd ..
ansible-playbook iscsi-deploy.yml -i hosts
```

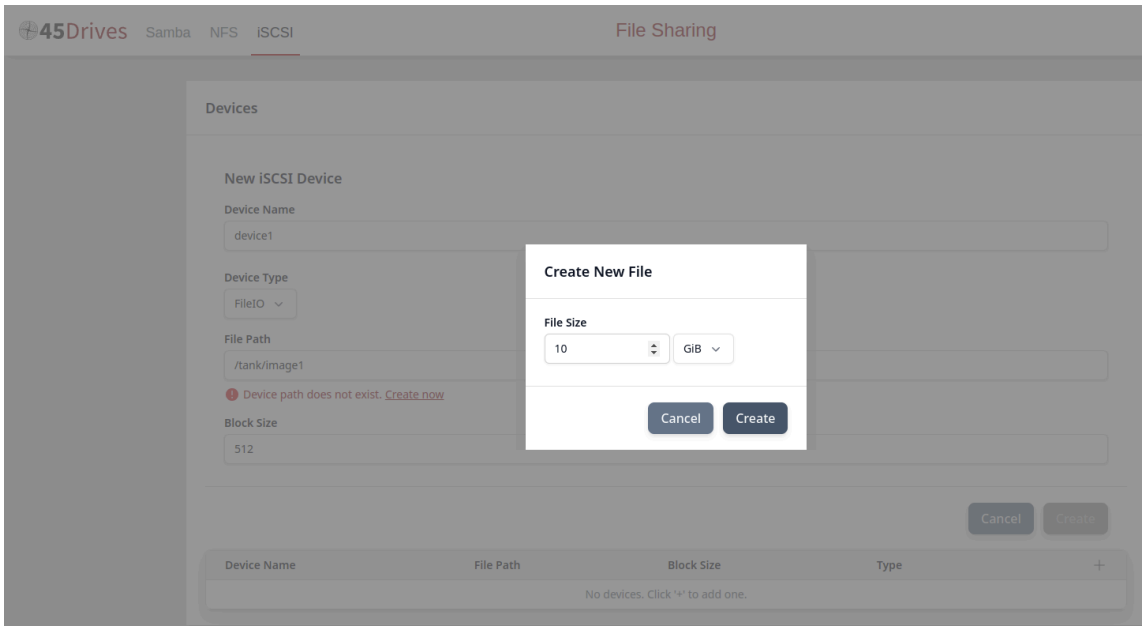
- Once this completes go to filesharing --> iSCSI



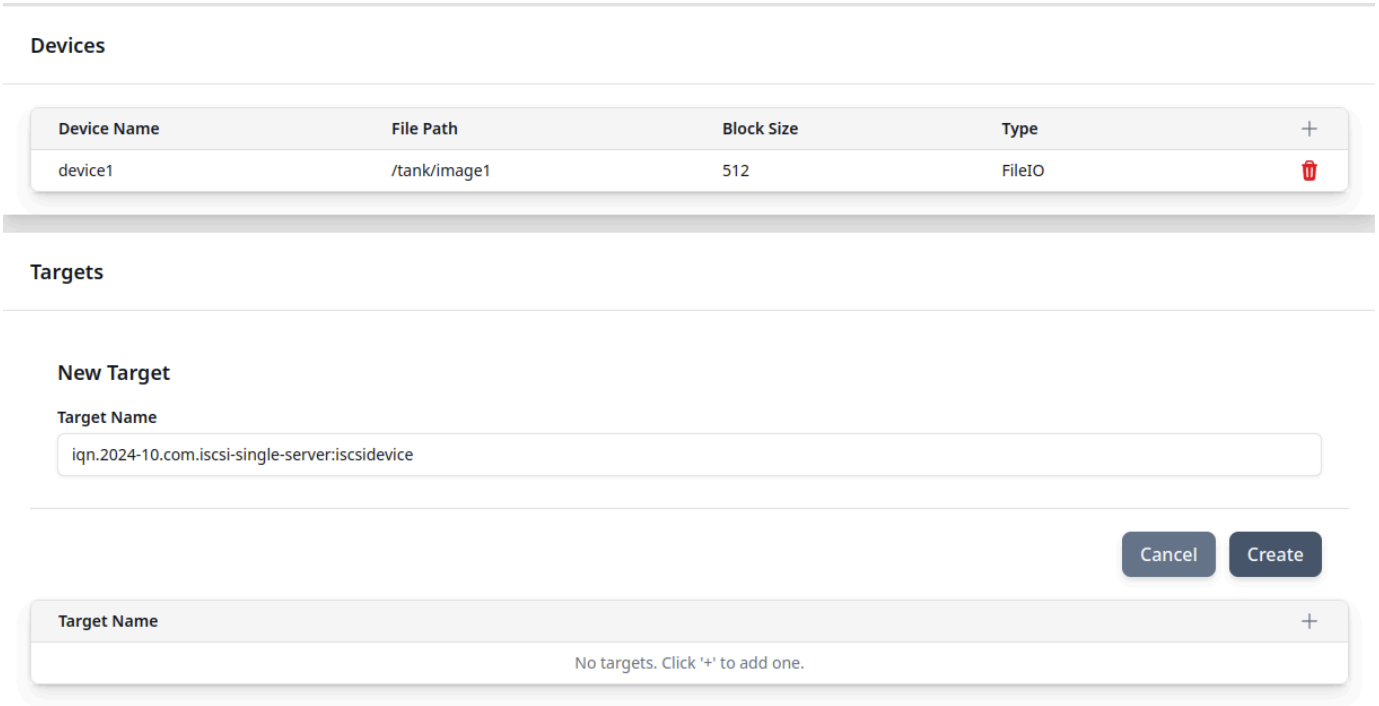
Click the + in devices --> give it a device name --> select your type(fileio) --> file path (zpool name then name for image file you created earlier)



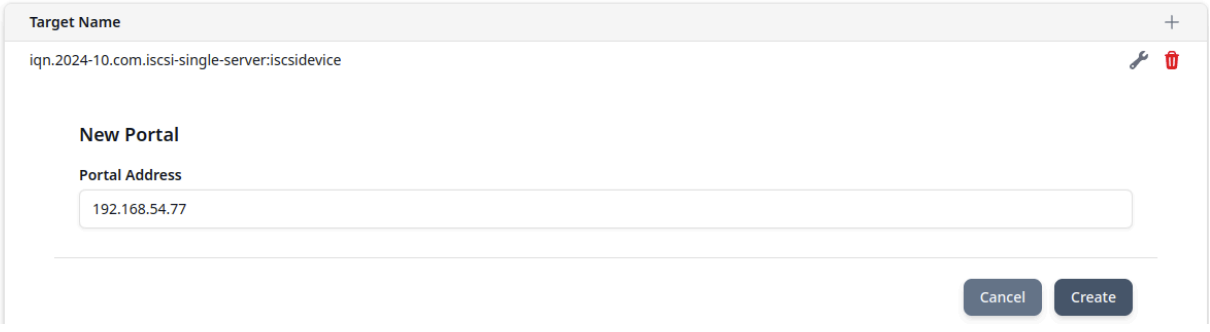
Select size --> click create --> then hit create again



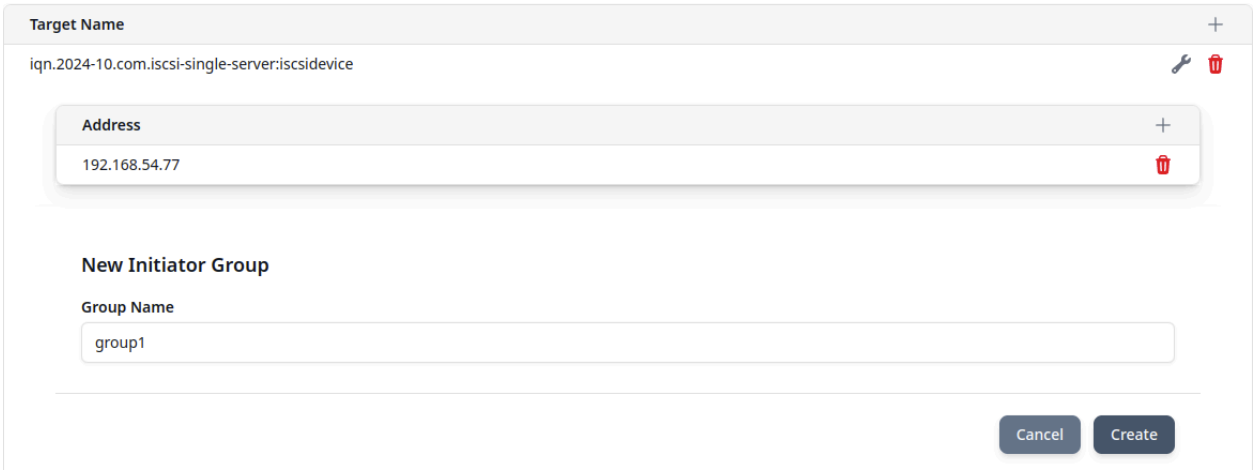
- Then go to targets, click the + and give it a target name



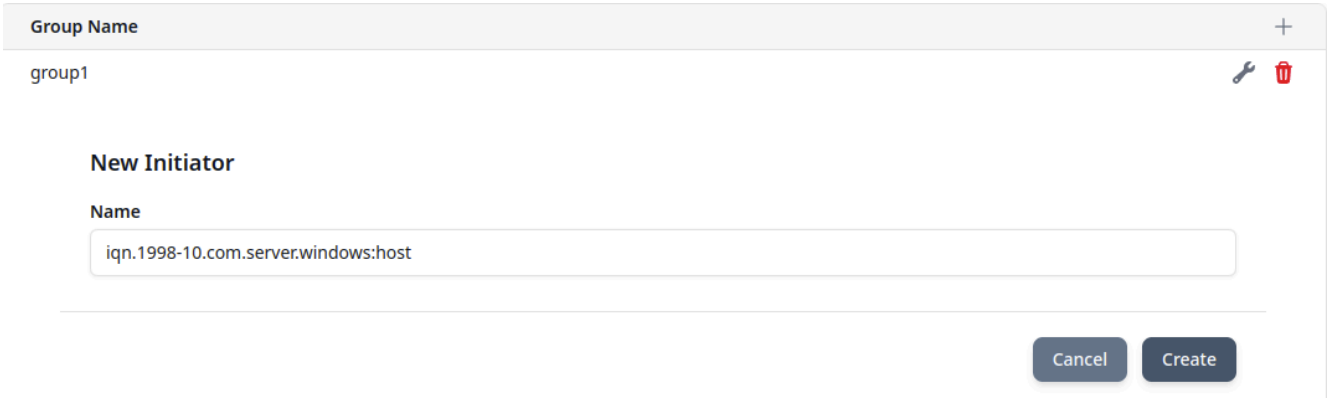
- Click the + on address and add the IP you want your initiator to connect to (Note: port 3260 is used by default so you don't need to input it, if using something different specify IP:8779 for example)



- Next, go to group and give it a name, click create

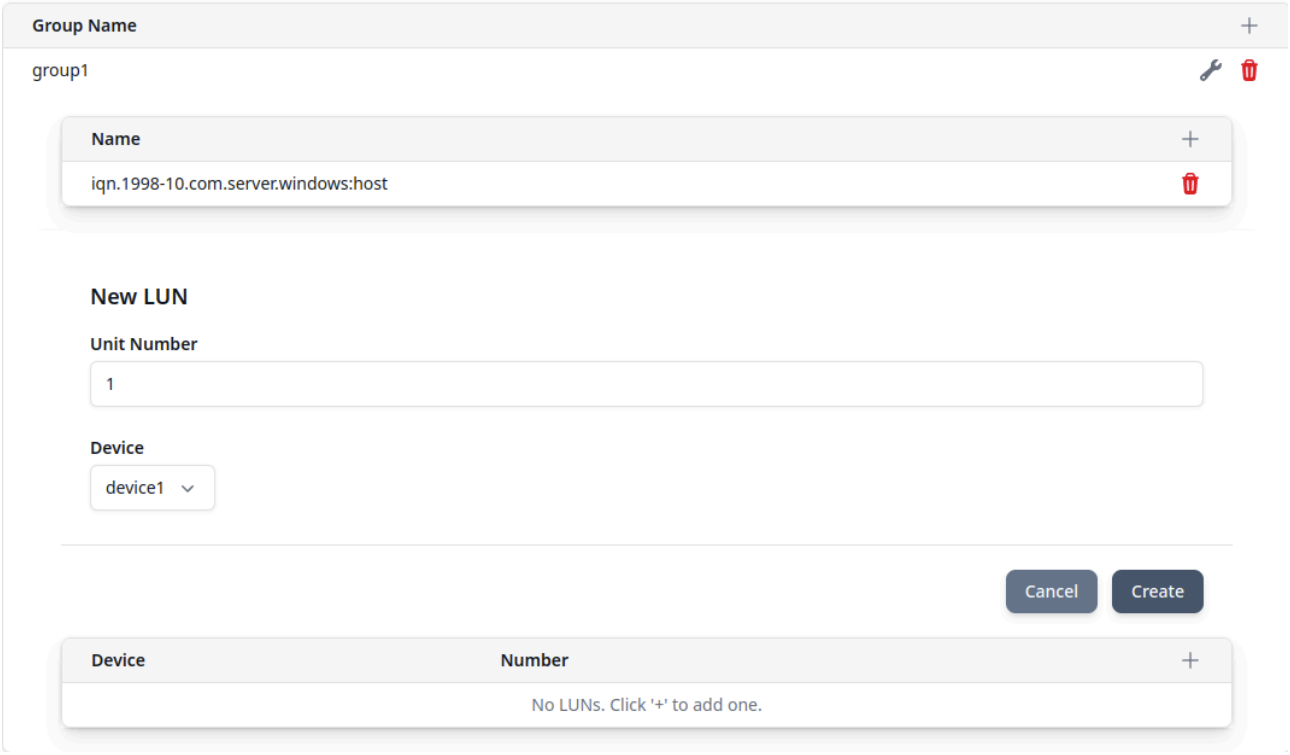


- Click the wrench next to the group name
  * add your initiator IP



45HomeLab
A Division of 45Drives

- add the device, give it a LUN ID and select the device we created

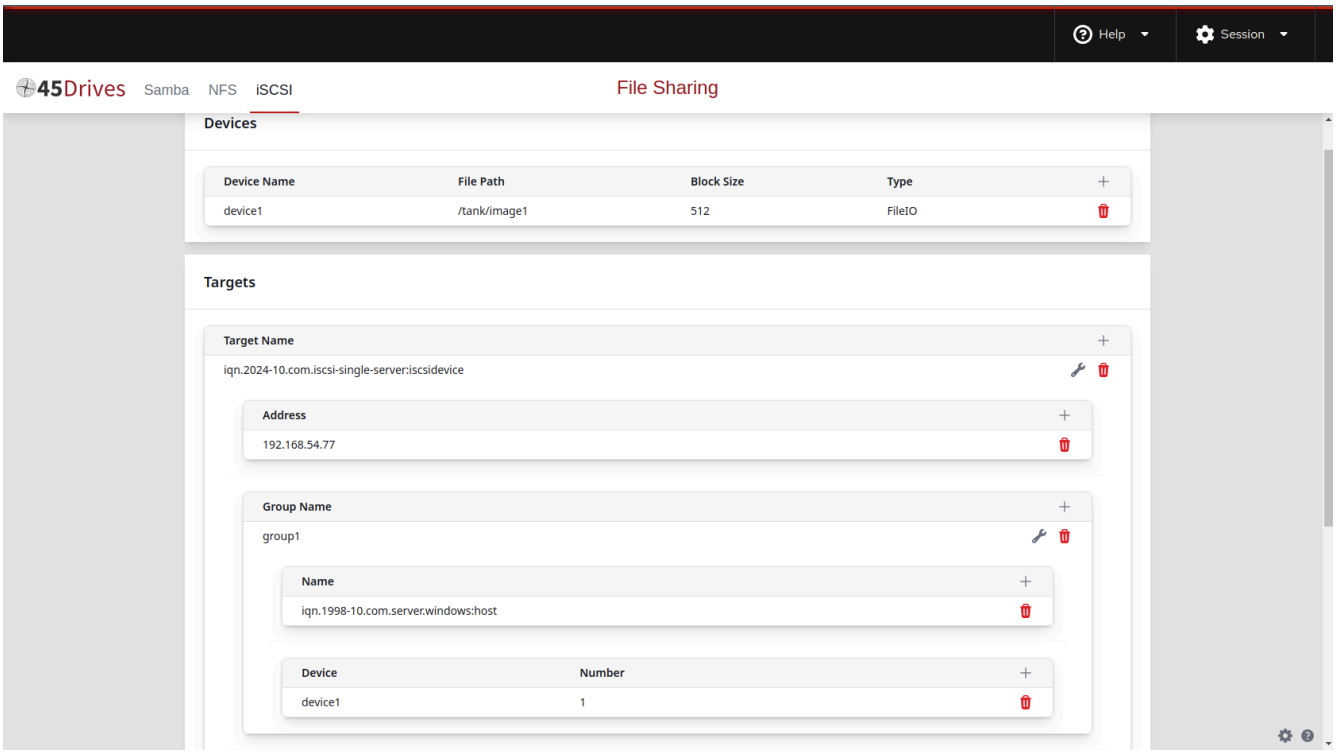


- *This is everything we need to get a basic iSCSI LUN set up * From here you have two ways to connect either quick connect in targets or below

- Go onto windows iscsi initiator –> Discovery –> enter IP –> OK

- Go back to iscsi Target click connect here is where we can connect or set up additional authentication

- For CHAP authentication please see below.

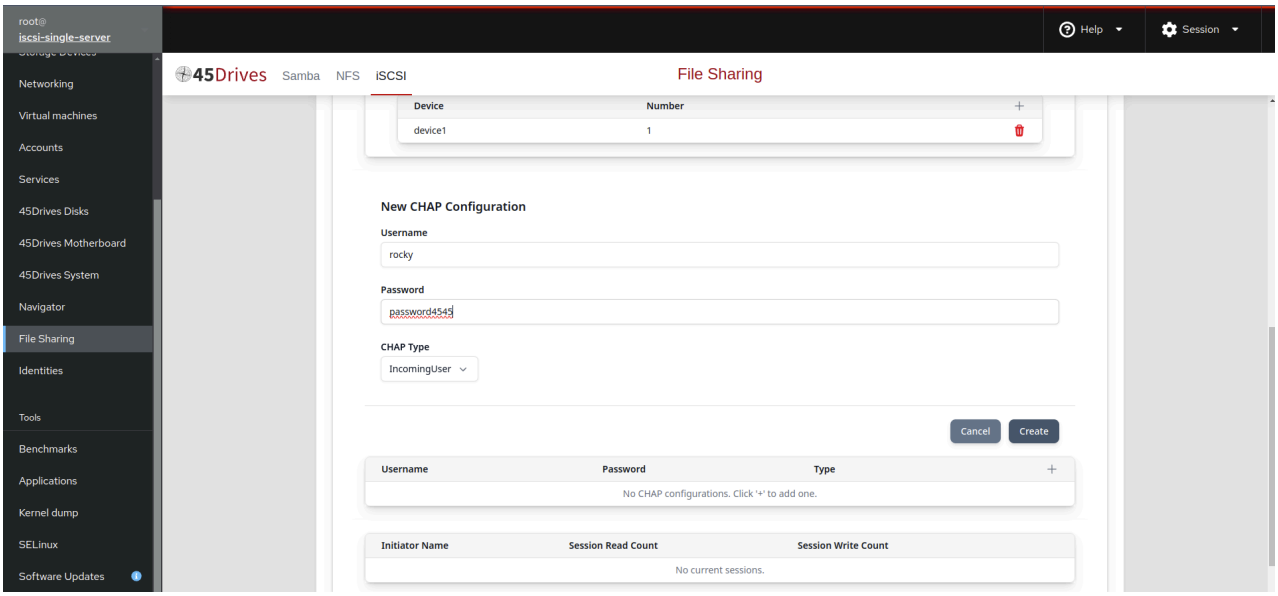


1. From here you have two ways to connect: either quick connect in targets or below.
2. Go onto Windows iSCSI initiator –> Discovery –> enter IP –> OK
3. Go back to iSCSI Target, click connect; here is where we can connect or set up additional authentication

Set-up

IncomingUser

- This method is for the Target to authenticate the connection from the initiators or the IncomingUsers.
- Create username, password, select IncomingUser, and create



- Next, on Windows, go to your iSCSI initiator –> Discovery –> Connect –> Advanced

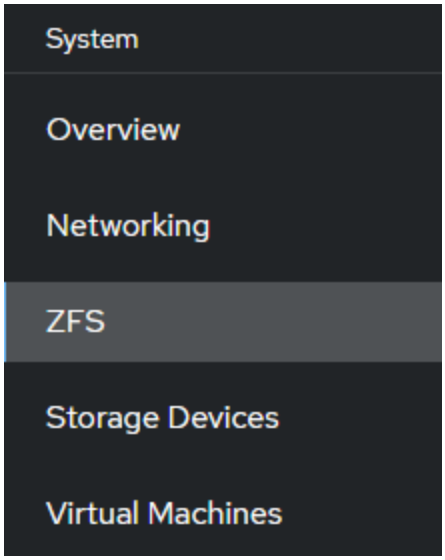
- From here, check off “Enable CHAP login” and enter the user and password we created in Rocky

AUTOMATED ZFS REPLICATION/SNAPSHOTS IN HOUSTON UI

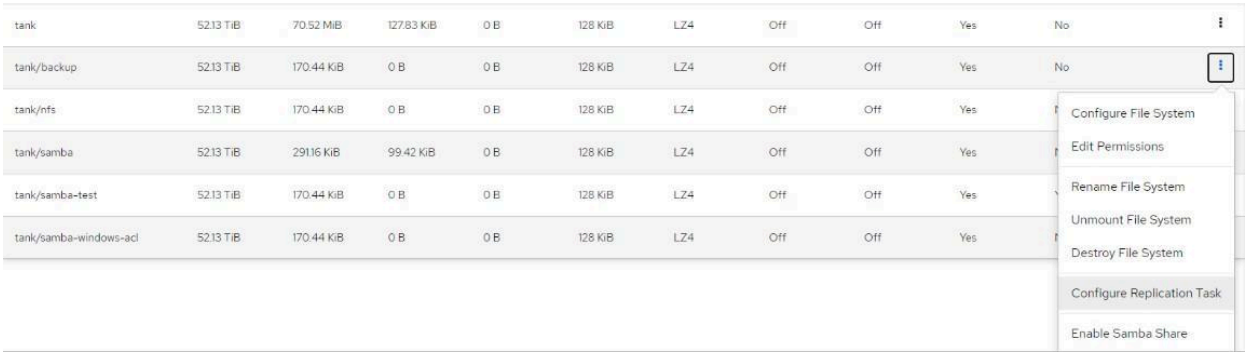
- Make sure to install mbuffer and remove the old auto snapshot service on all Storinators you wish to use snapshots with znapzend.

```
dnf install mbuffer

systemctl restart znapzend
```

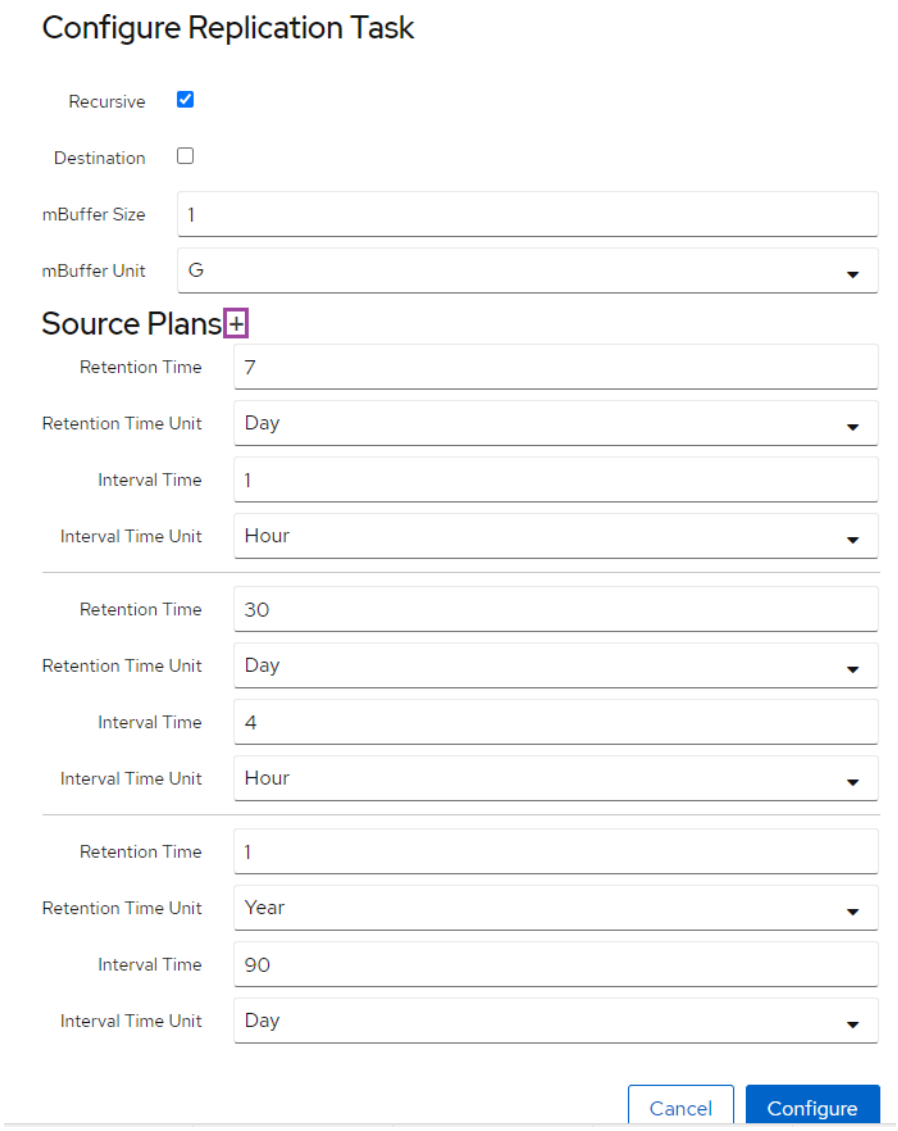
- In the Houston UI, go to the ZFS tab.



- Select the Dataset you wish to backup. Click the drop down on the left and select “Configure Replication Task



- The screenshot below details a task that takes a snapshot once daily, and retains the snapshots for 1 month. This can be customized to your use case.
- Also, you can add multiple rules by clicking the +. For example, the setup below is for every hour for 7 days, every 4 hours for 30 days, and every 90 days for a year.



Make sure to restart the 'znapzend' service after any change has been made to snapshot tasks

- To ensure the snapshots are being created you can go to the Snapshots section of the ZFS to see all snapshots that were created.

SOFTWARE

45HomeLab comes with Rocky as the default OS and Houston UI for the server management. Apart from that you can install other software stacks as well as per your preference. We have included some of them for your reference.

Container Runtime Setup (Docker or Podman)

- Remove conflicting packages (safe to run even if none installed)

```
sudo dnf remove docker \
                docker-client  \
                docker-client-latest \
                docker-common docker-latest \
                docker-latest-logrotate docker-logrotate \
                docker-engine \
                podman \
                runc
```

- Set up the repo:

```
sudo dnf -y install dnf-plugins-core
sudo dnf config-manager –add-repo https://download.docker.com/linux/rhel/docker-ce.repo
```

- Install packages:

```
sudo dnf install docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin
```

- Start docker by running:

```
Systemctl enable –now docker
```

sudo systemctl enable –now docker

- You'll be asked to authenticate a few services

```
[45drives@homelab ~]$ systemctl enable --now docker
==== AUTHENTICATING FOR org.freedesktop.systemd1.manage-unit-files ====
Authentication is required to manage system service or unit files.
Multiple identities can be used for authentication:
 1.  45drives
 2.  testing
Choose identity to authenticate as (1-2): 1
Password:
==== AUTHENTICATION COMPLETE ====
Created symlink /etc/systemd/system/multi-user.target.wants/docker.service → /usr/lib/systemd/system/docker.service.
==== AUTHENTICATING FOR org.freedesktop.systemd1.reload-daemon ====
Authentication is required to reload the systemd state.
Multiple identities can be used for authentication:
 1.  45drives
 2.  testing
Choose identity to authenticate as (1-2): 1
Password:
==== AUTHENTICATION COMPLETE ====
==== AUTHENTICATING FOR org.freedesktop.systemd1.manage-units ====
Authentication is required to start 'docker.service'.
Multiple identities can be used for authentication:
 1.  45drives
 2.  testing
Choose identity to authenticate as (1-2): 1
Password:
==== AUTHENTICATION COMPLETE ====
[45drives@homelab ~]$
```

- Optional: if you want your 45drives user to be able to run docker commands without sudo you can add them to the docker group

```
sudo usermod -aG docker $USER
```

- Log out and log back in so that your group membership is re-evaluated.

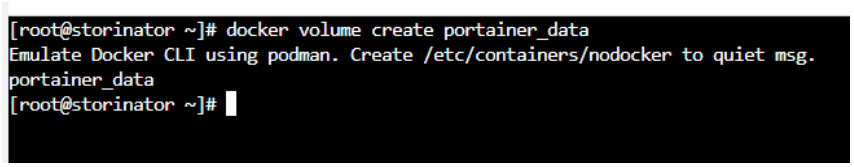
PORTAINER

With the help of Portainer, you can easily interact with containerized programs, monitor your Docker installation, and set up new stacks. To centralize your container administration around a single application, a single Portainer instance may link to numerous Docker hosts.

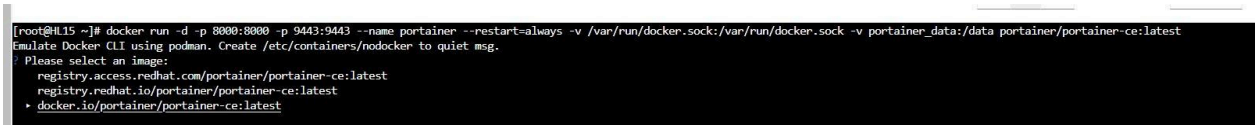
Make sure docker is installed before you proceed.

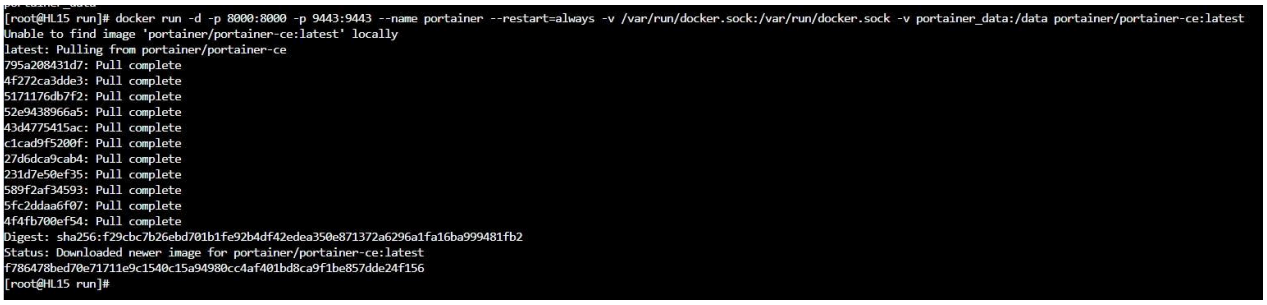
- Create the docker volume

```
docker volume create portainer_data
```

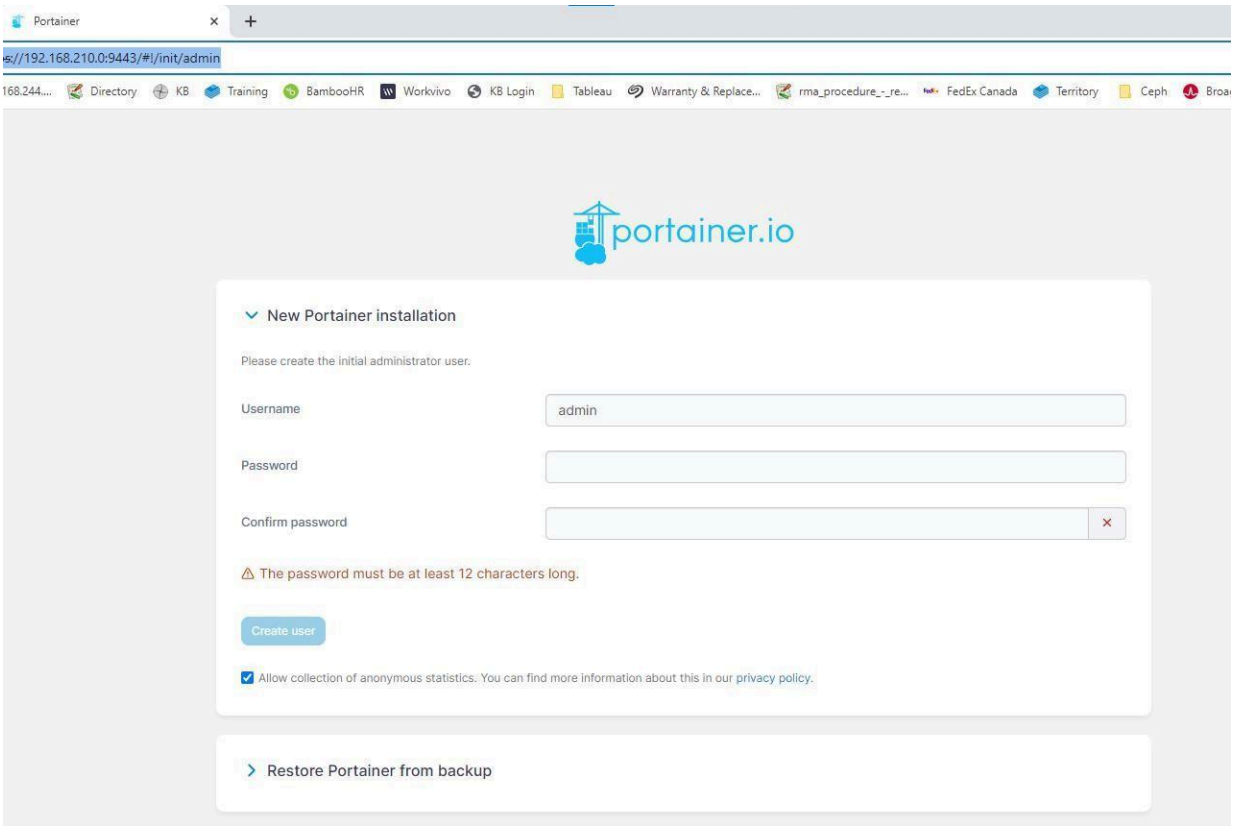


-
```
docker run -d -p 8000:8000 -p 9443:9443 --name portainer --
restart=always -v /var/run/docker.sock:/var/run/docker.sock -v
portainer_data:/data portainer/portainer-ce:lts
```

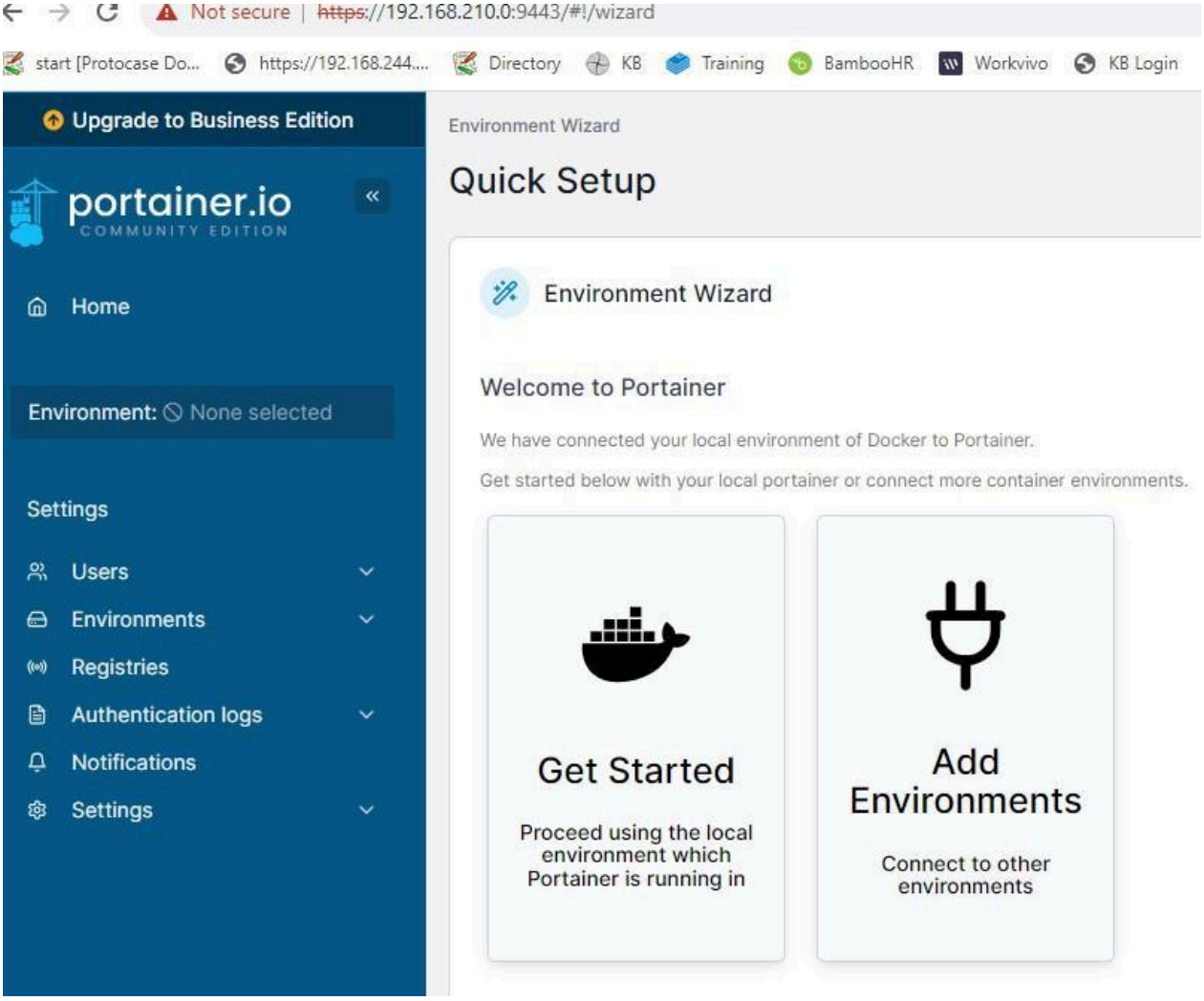




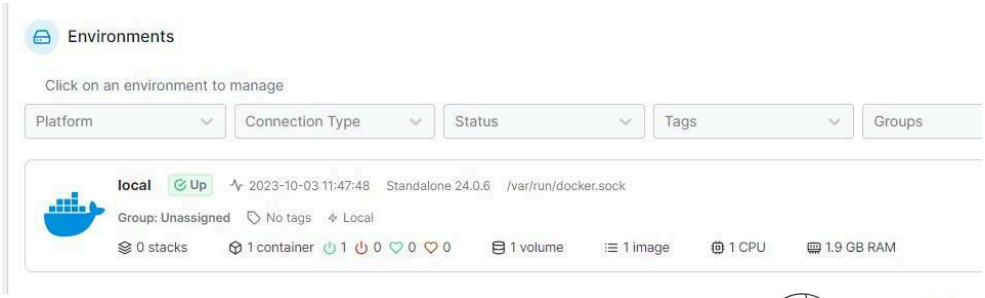
- The portainer will be on port 9443 so access it using https://serverip:9443
- Make sure the port is open in the firewall.
- Create the admin user



- You can click on getting started.
- You can check more on the documentation at https://docs.portainer.io/
- Click Get started to start a local environment.



- You should have portainer local environment like below



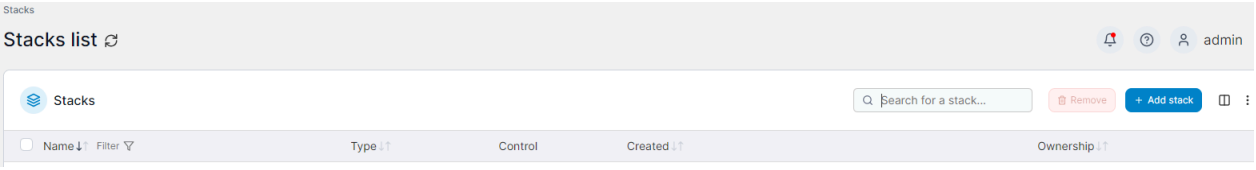
45HomeLab
A Division of 45Drives

Now you have your portainer ready to deploy other software stacks.

INSTALLING NGINX PROXY MANAGER(NPM) ON PORTAINER

Below would be the steps to install NPM to set up proxy for custom ports for various applications.

- Go to your portainer home and select your environment.
- Click on stacks and add a new stack.

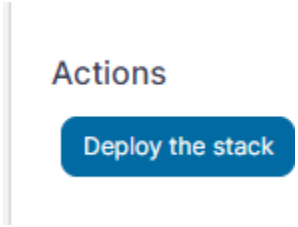


- Give a name for your stack. For example, we can give npm.
- Paste the below contents on the web editor

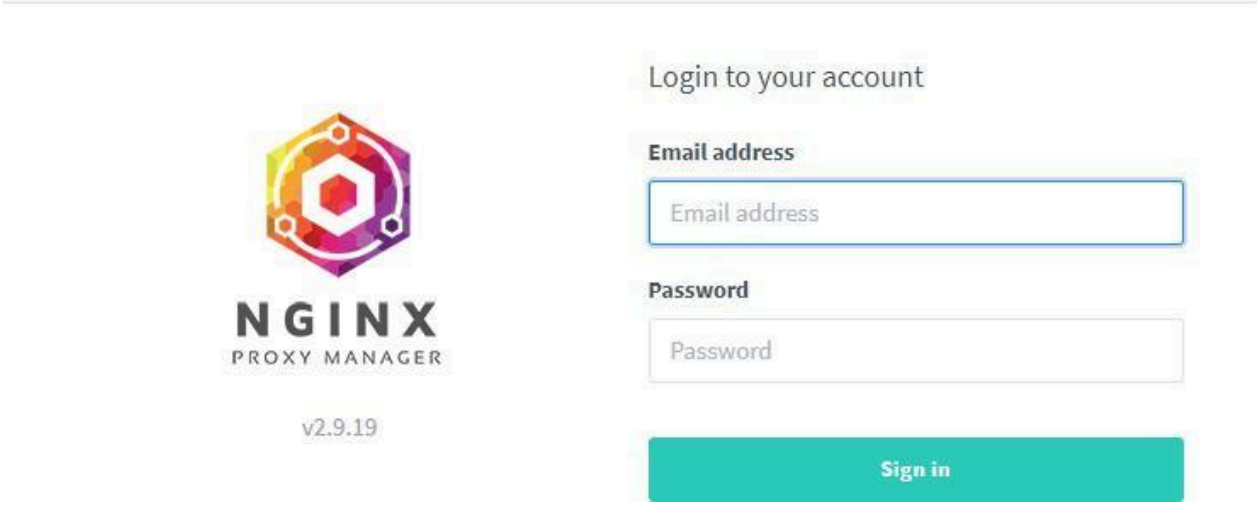
45HomeLab
A Division of 45Drives
45HomeLab
A Division of 45Drives

```
version: '3
'
volumes:
    npm-data:
    npm-ssl:
    npm-db:

networks:
    frontend:
    backend:

services:
    npm-app:
        image: jc21/nginx-proxy-manager:2.9.19
        restart: always
        ports:
        - "80:80"
        - "81:81"
        - "443:443"
    environment:
        - DB_MYSQL_HOST=npm-db
        - DB_MYSQL_PORT=3306
        - DB_MYSQL_USER=npm
        - DB_MYSQL_PASSWORD=thisisjustatest
        - DB_MYSQL_NAME=npm
    volumes:
        - npm-data:/data
        - npm-ssl:/etc/letsencrypt
    networks:
        - frontend
        - backend
npm-db:
    image: jc21/mariadb-aria:latest
    restart: always
    environment:
        - MYSQL_ROOT_PASSWORD=thisisjustatest
        - MYSQL_DATABASE=npm
        - MYSQL_USER=npm
        - MYSQL_PASSWORD=thisisjustatest
    volumes:
        - npm-db:/var/lib/mysql
    networks:

        - backend
```

- After that click on deploy stack and wait for the deployment to be completed.



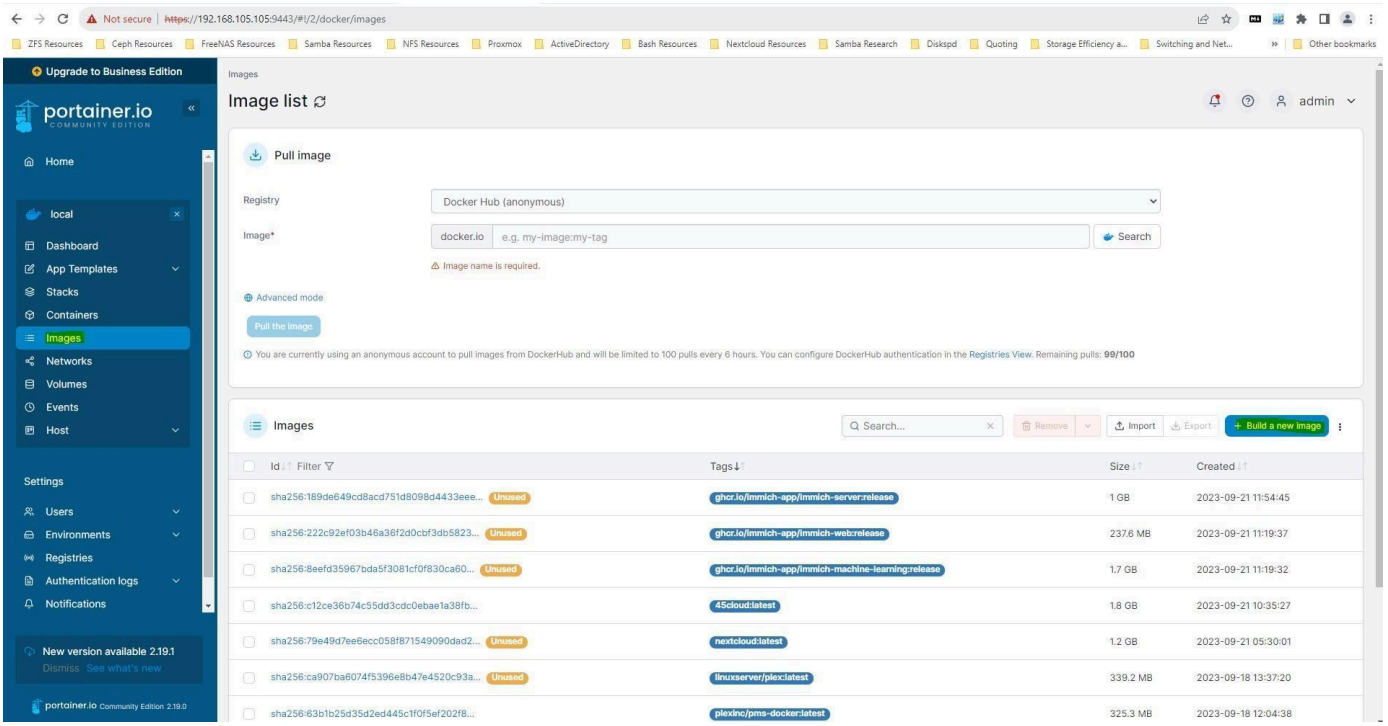
- Nginx Proxy Managers webUI will be available at port :81 http://192.168.210.0:81 with admin@example.com and changeme credentials.



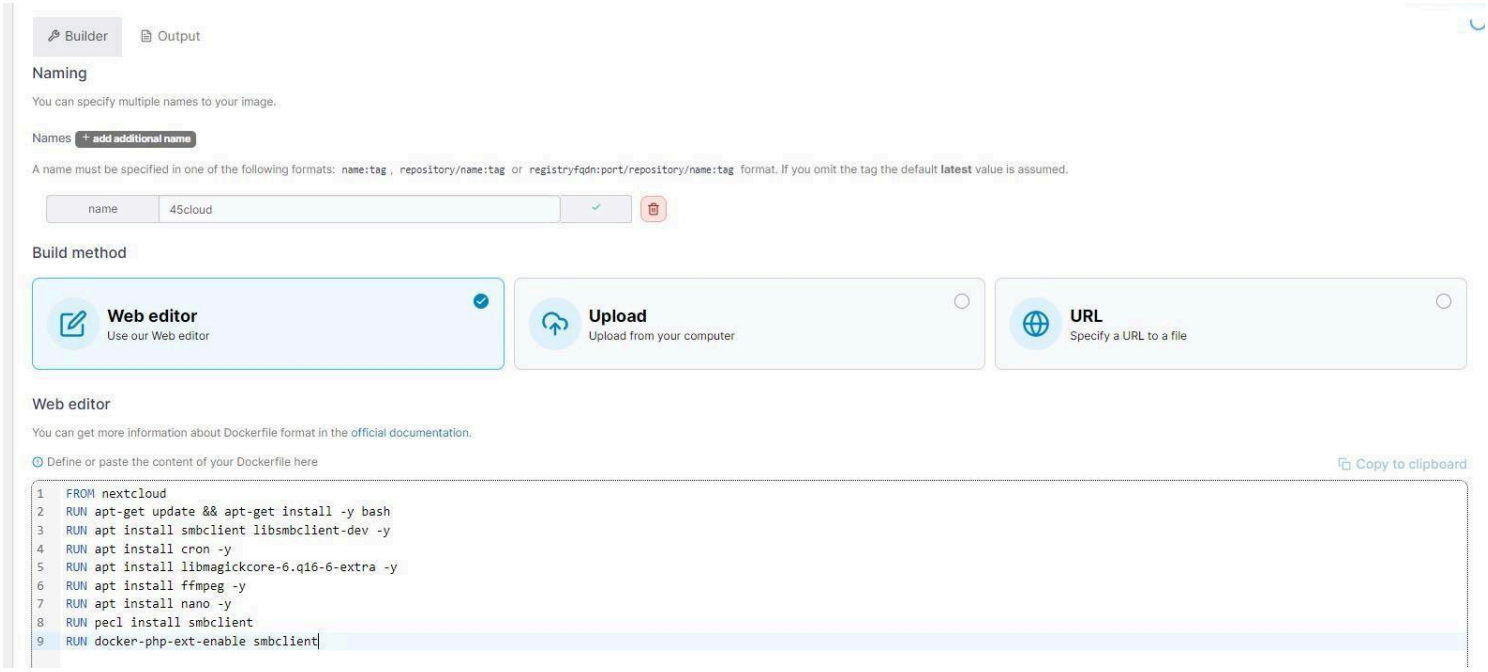
- Once in set the admin account email id and password
- After that you can start creating your proxies.

NEXTCLOUD AND NPM ON PORTAINER

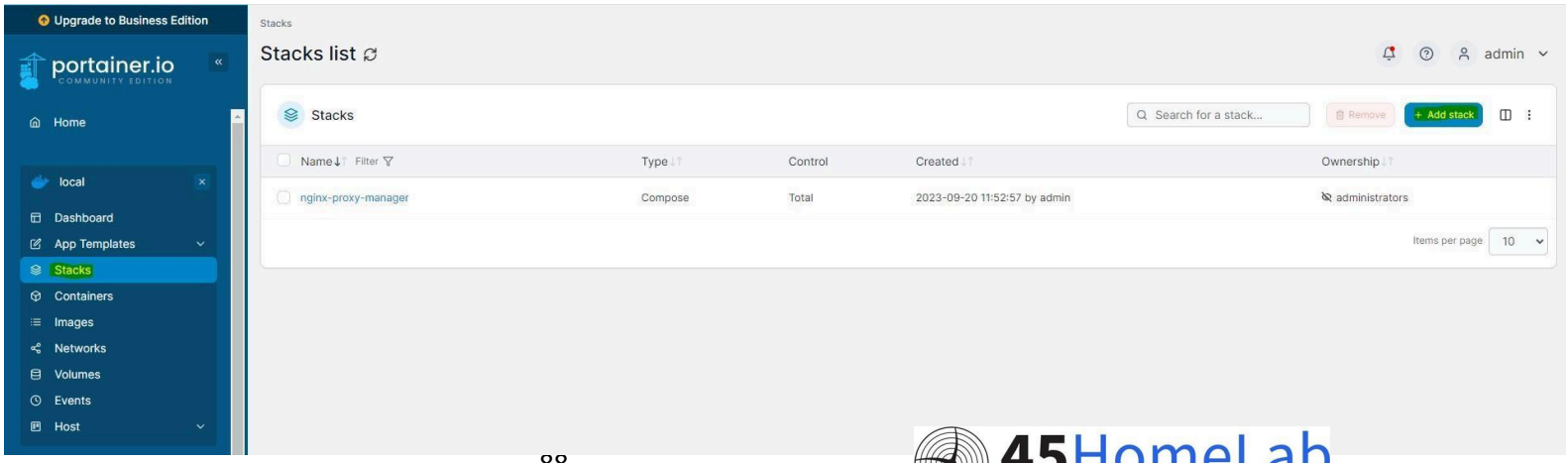
- In portainer navigate to "Images" and select Build a New Image:



- Name the new image "45cloud" and paste the contents of https://github.com/45Drives/scripts/blob/main/nextcloud/Dockerfile



- **Click "Build the image."** This starts the build and pulls base images (about **3–4 GB**).
  Note: This can take several minutes depending on your network and registry speed. Watch the build logs until it completes.
- Then navigate to the "Stacks" tab, and click "Add Stack"



- Name the stack nextcloud-nginx and paste the following into the text box

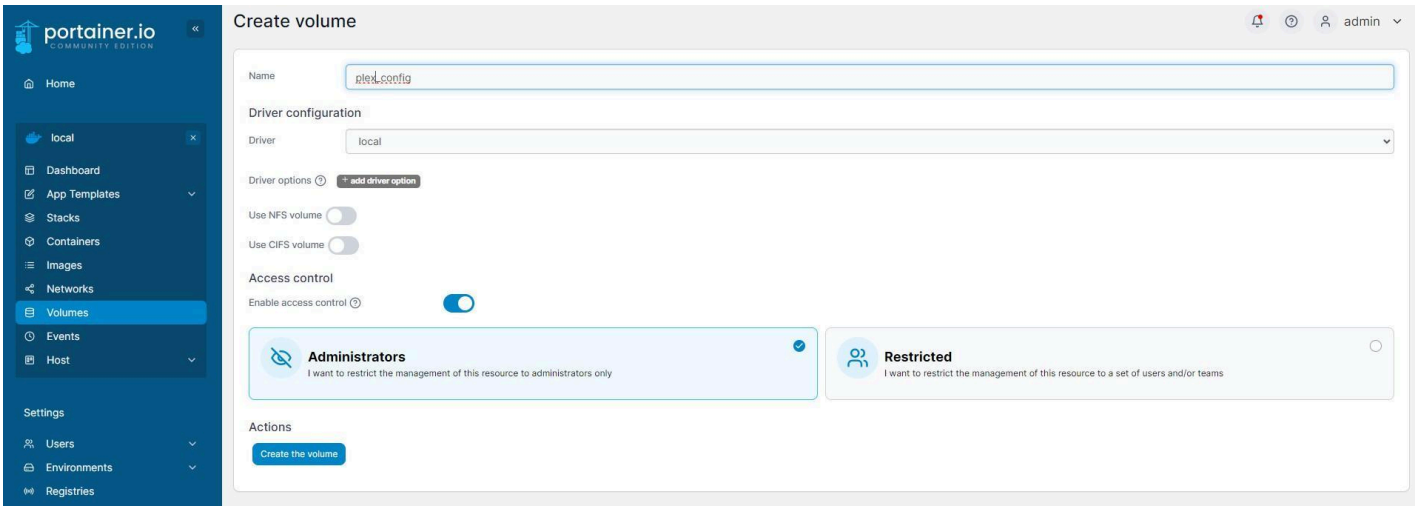
- If you already have NPM installed remove those NPM sections from the text below when pasting into the web editor.

```
version: "3"
volumes:
        nextcloud-data:
        nextcloud-db:
        npm-data:
        npm-ssl:
        npm-db:
networks:192.168.54.104 homelab.local
frontend:
backend:
services:
        nextcloud-app:
                image: 45cloud:latest
                restart: always
                volumes:
                        - nextcloud-data:/var/www/html
        environment:
                - MYSQL_PASSWORD=thisisjustatest
                - MYSQL_DATABASE=nextcloud
                - MYSQL_USER=nextcloud
                - MYSQL_HOST=nextcloud-db
                - PHP_UPLOAD_LIMIT=200G
                - OVERWRITEPROTOCOL=https
        networks:
                - frontend
                - backend
        depends_on:
                - nextcloud-db
nextcloud-db:
        image: mariadb:10.5
        restart: always
        command: --transaction-isolation=READ-COMMITTED --binlog-format=ROW
        volumes:
                - nextcloud-db:/var/lib/mysql
                environment:
                        - MYSQL_ROOT_PASSWORD=thisisjustatest
                        - MYSQL_PASSWORD=thisisjustatest
                        - MYSQL_DATABASE=nextcloud
                        - MYSQL_USER=nextcloud
                networks:
                        - backend
npm-app:
        image: jc21/nginx-proxy-manager:2.9.19
        restart: always
        ports:
                - "80:80"
                - "81:81"
                - "443:443"
                # Enable ONLY if you need to pass these through NPM:
                # - "8900:8900"
                # - "32400:32400"
                # - "2283:2283"
        environment:
                - DB_MYSQL_HOST=npm-db
                - DB_MYSQL_PORT=3306
                - DB_MYSQL_USER=npm
                - DB_MYSQL_PASSWORD=thisisjustatest
                - DB_MYSQL_NAME=npm
        volumes:
                - npm-data:/data
                - npm-ssl:/etc/letsencrypt
        networks:
                - frontend
                - backend
        depends_on:
                - npm-db
npm-db:
        image: jc21/mariadb-aria:latest
        restart: always
        environment:
                - MYSQL_ROOT_PASSWORD=thisisjustatest
                - MYSQL_DATABASE=npm
                - MYSQL_USER=npm
                - MYSQL_PASSWORD=thisisjustatest
        volumes:
                - npm-db:/var/lib/mysql
        networks:
                - backend
```

- Click Deploy Stack and wait for it to finish
- At this point all configuration steps are identical to: https://knowledgebase.45drives.com/kb/kb451402-nextcloud-with-nginx-proxy-manager-on-rocky-linux/
  - Nginx Proxy Managers webUI will be available at port :81 with admin@example.com and changeme credentials.
  - When creating your proxy host use the name of the container found in the containers tab of portainer.

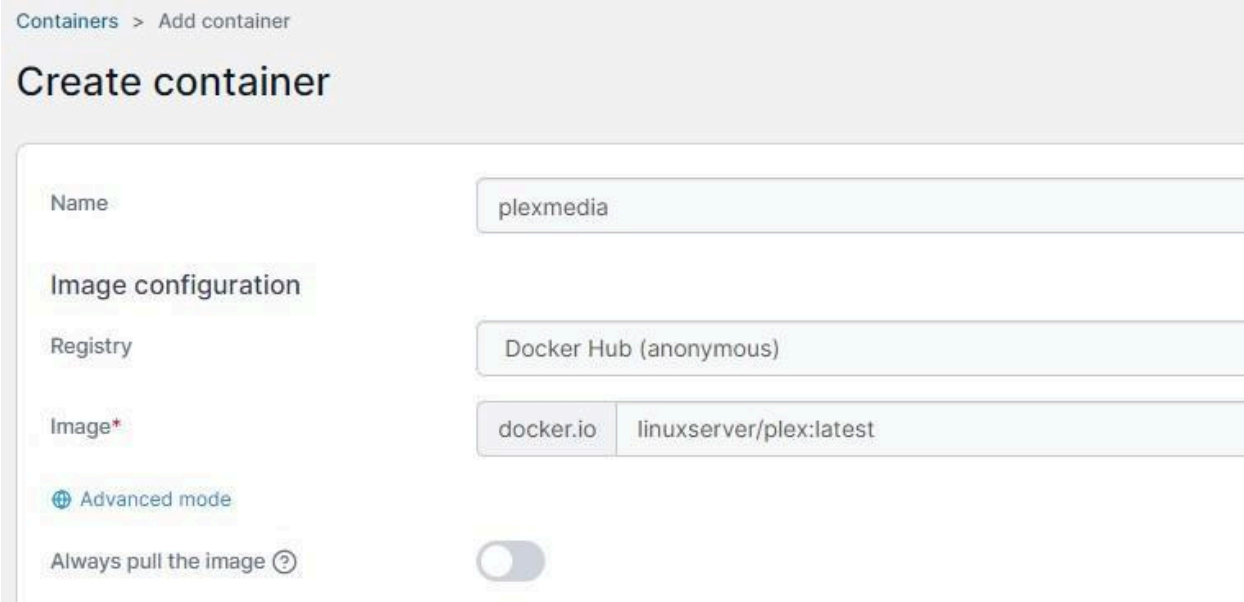
CONFIGURING PLEX PORTAINER

- Go to portainer and create a new volume called plex_config

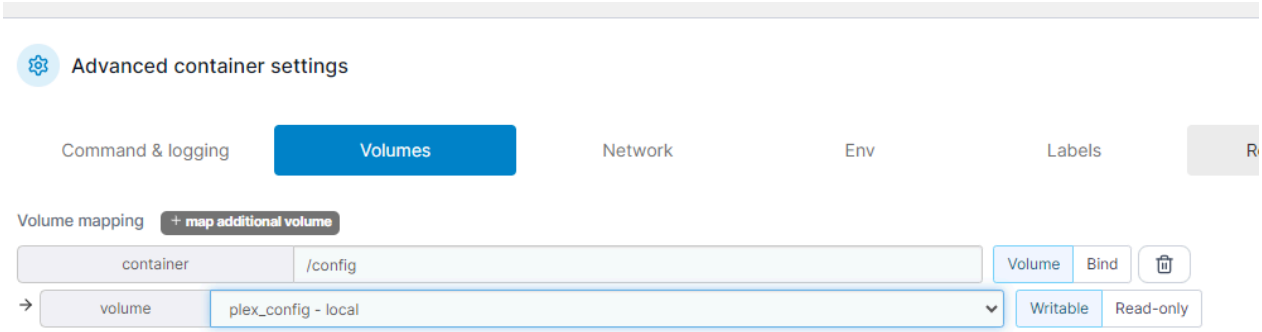


- Then go to create a container name as you want example "plexmedia"

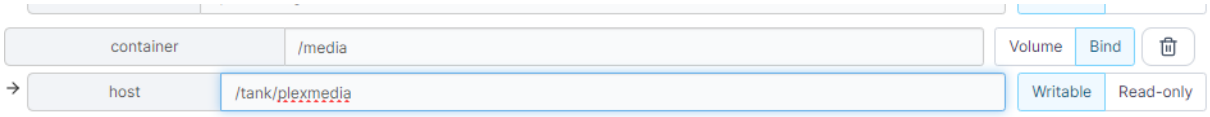
- For the image give **linuxserver/plex:latest**



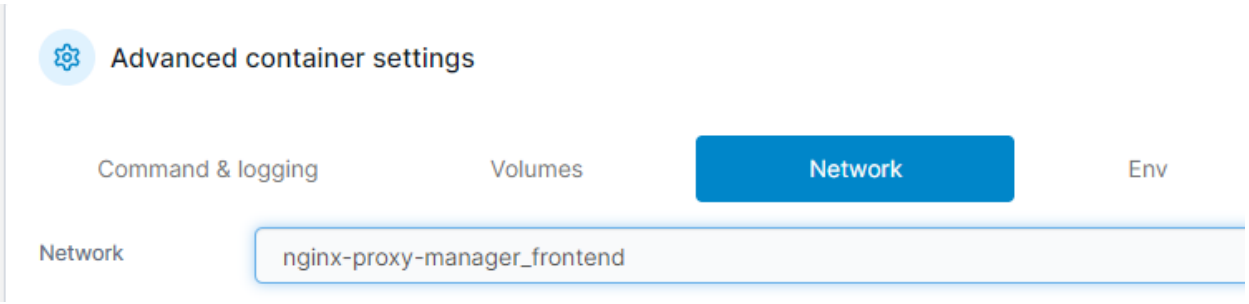
- Under volumes click on map additional volume and create a volume with container path of /config to the volume you created in step 1



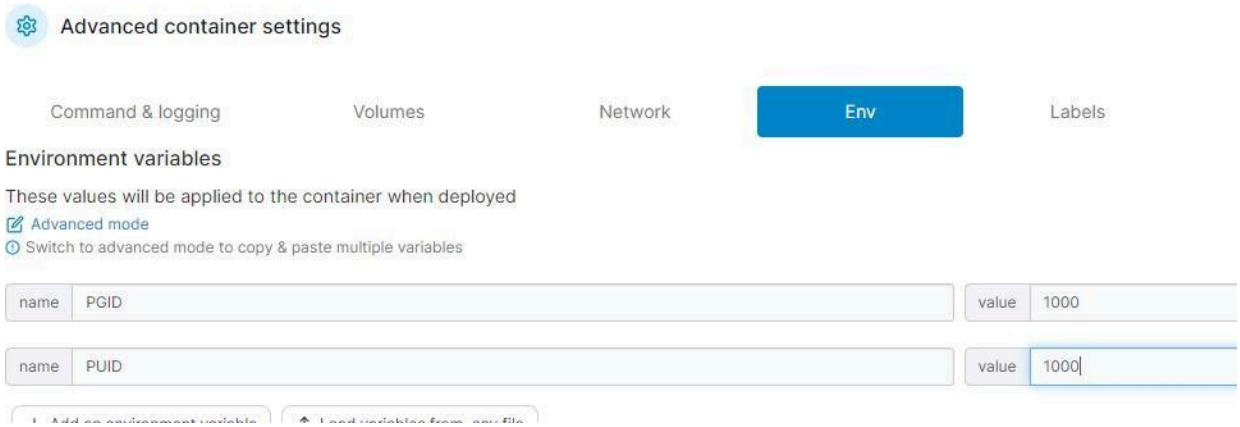
- Click on map additional volume and create a second bind volume with container path to /media. This name can change to whatever and bind it to the host path that they want to store media such as a zpool, etc. (here we can use our path in the zpool where our storage is).



- Under network select the nginx proxy frontend network



- Under ENV click on add environment variable and throw in PGID 1000 and PUID 1000. Here 1000 is the ID of the user that we are given ownership of the content, so add the ID accordingly.



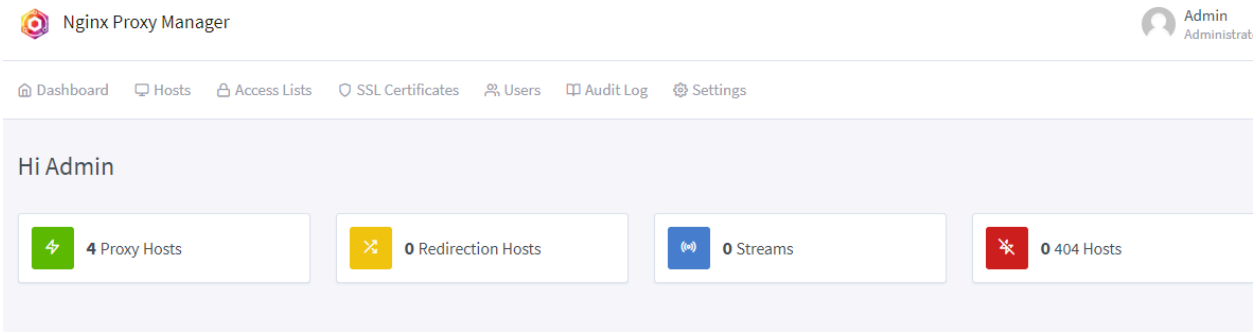
- Select a restart policy of unless stopped

- deploy the container

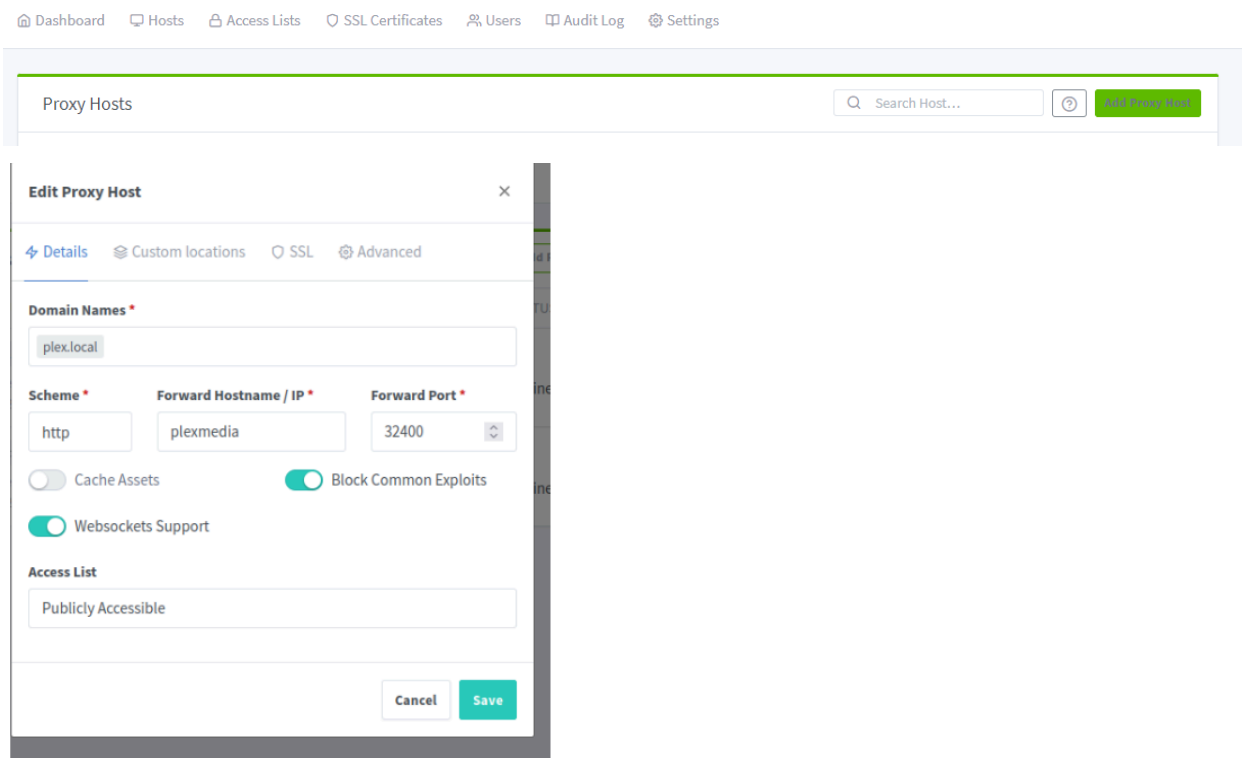
- You should be able to see that the container is deployed and running.

45HomeLab
A Division of 45Drives

- Nginx Proxy Managers web UI will be available at port :81 with admin@example.com and 'change me' credentials.
- When creating your proxy host use the name of the container found in the containers tab of portainer.
- Create a proxy host with domain name IP:PORT or DNS name if they are port forwarding or have a hostname select scheme (http/https), forward hostname is the name of the container and forward port by default would be 32400. Then under advanced custom nginx config put in "listen 32400;"
- Click on proxy hosts

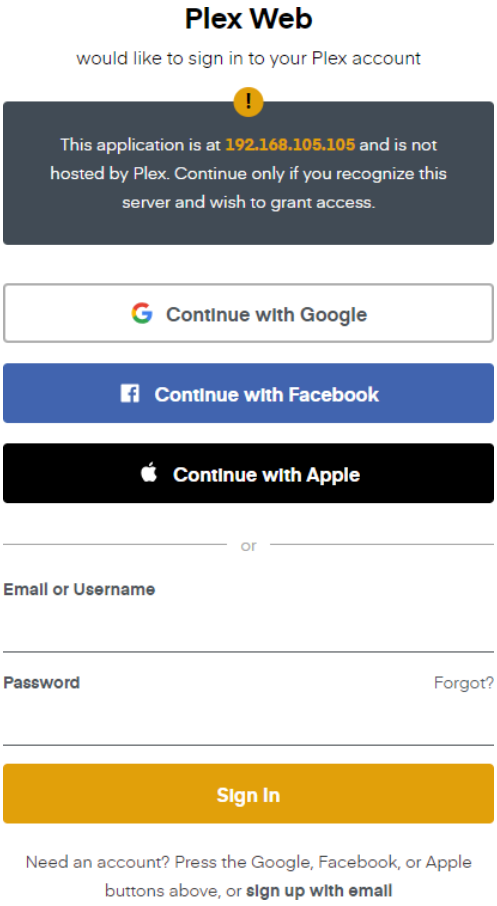


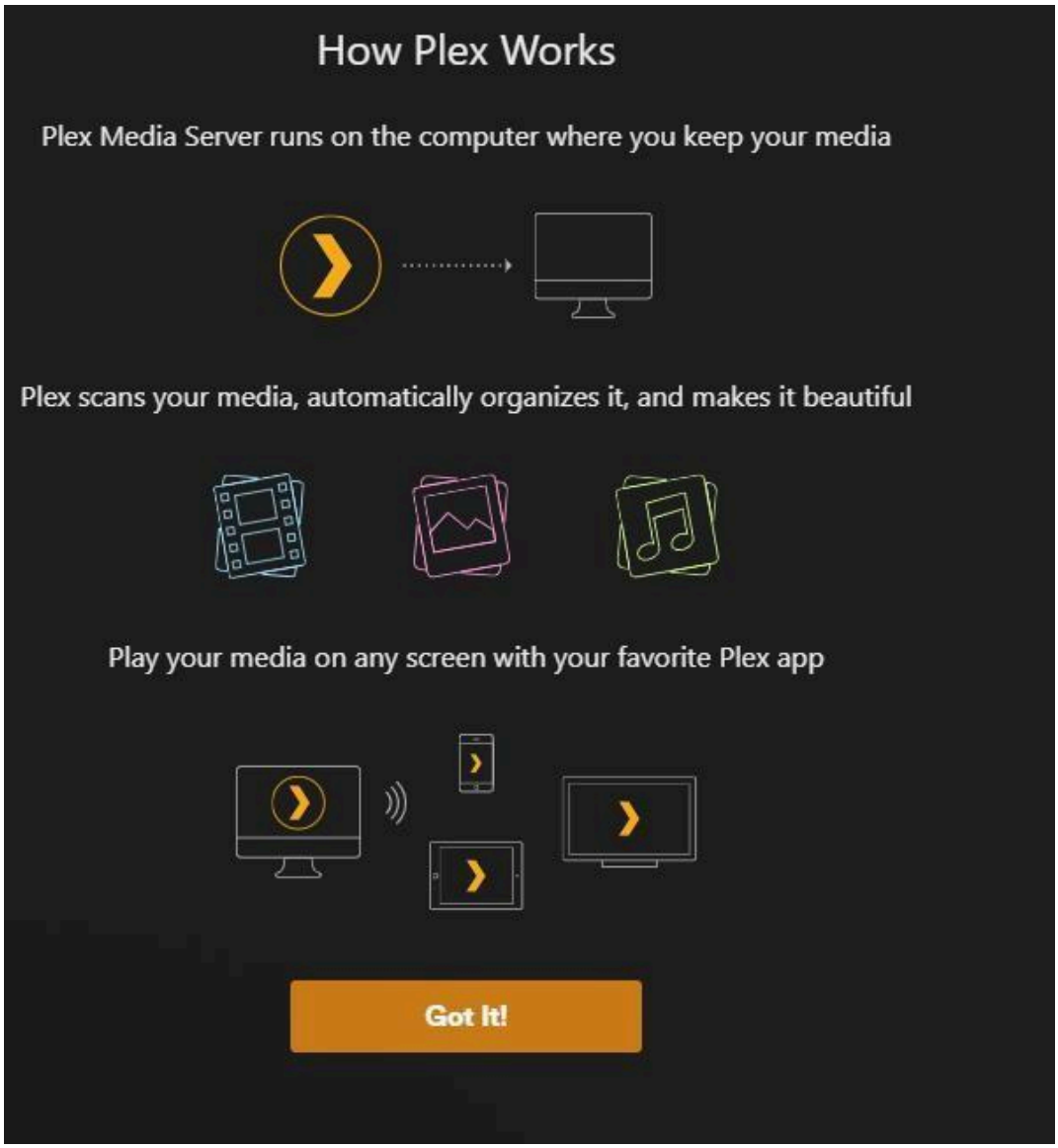
- Add proxy host



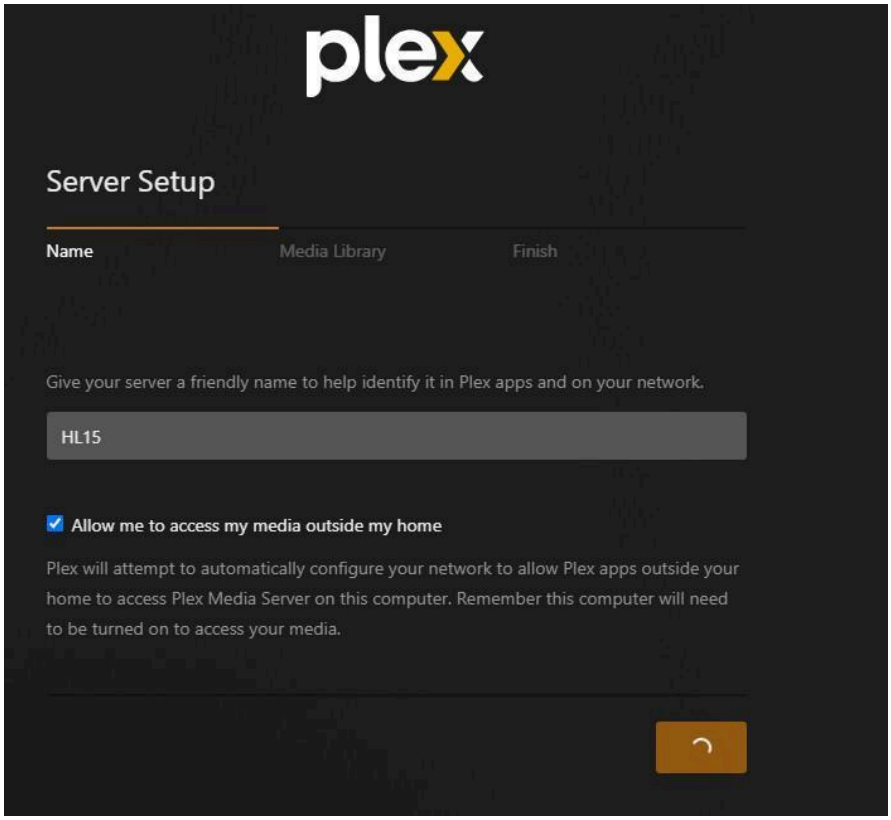
- Now you can access your plex using http://serverip:32400

- Make sure the port is open in the firewall.
- Sign in or create a plex account

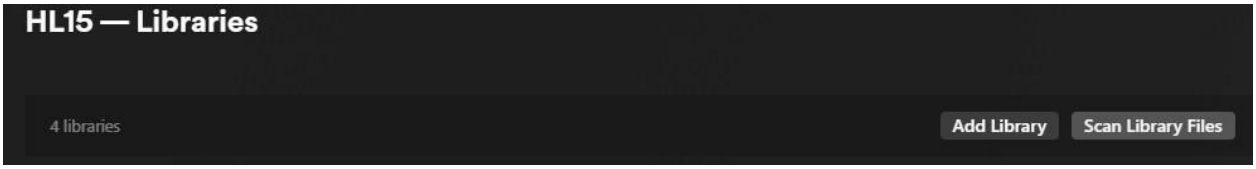




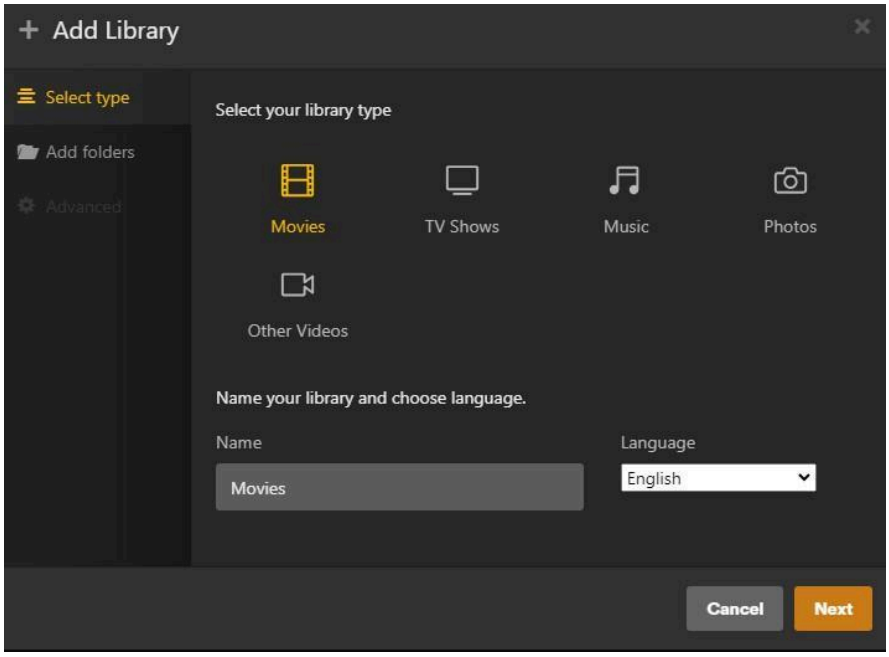
- After that you can set up the library as per your wish



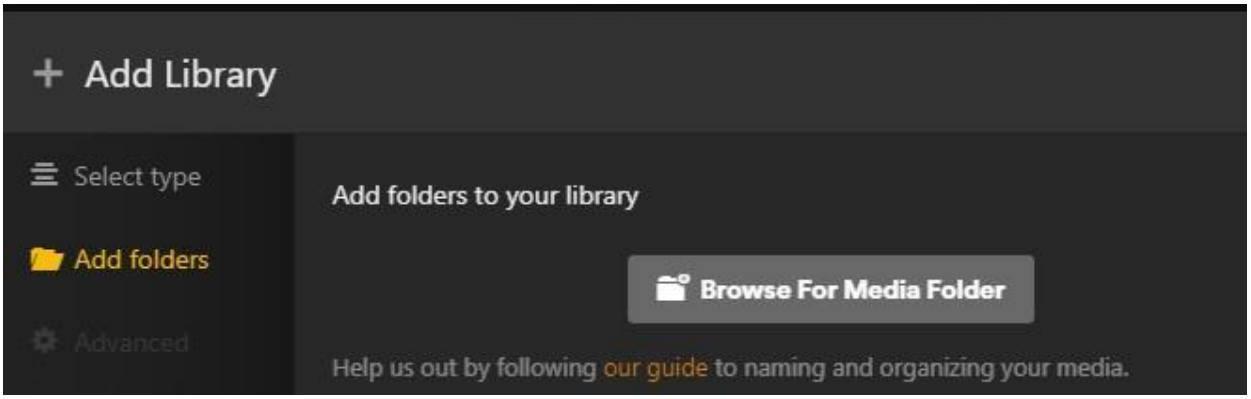
- Click on add libraries

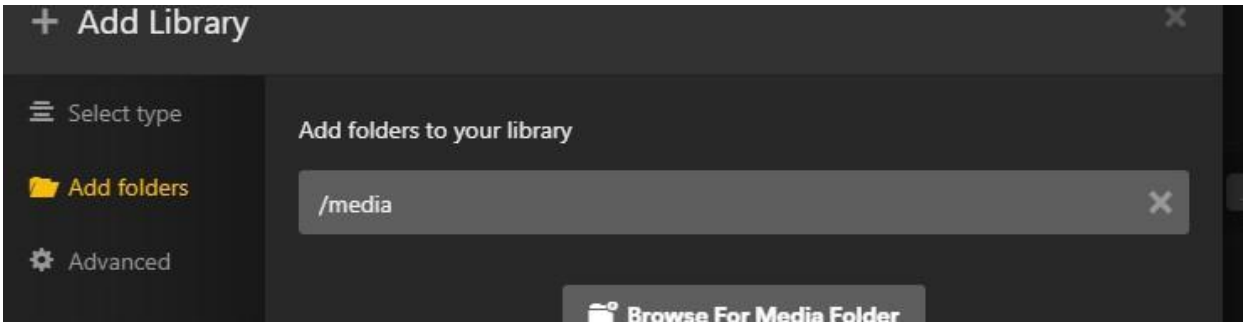


- You can select the library type and then select the media folder that we had linked to our storage in HL15

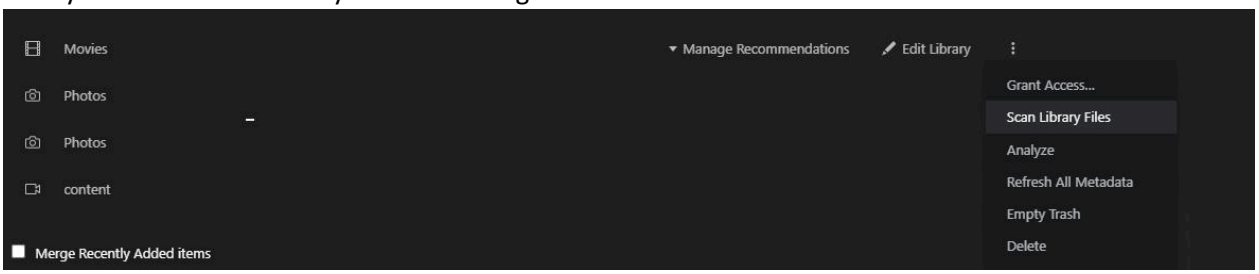


- After that select the folder, we are selecting media as we had given that during our container creation.

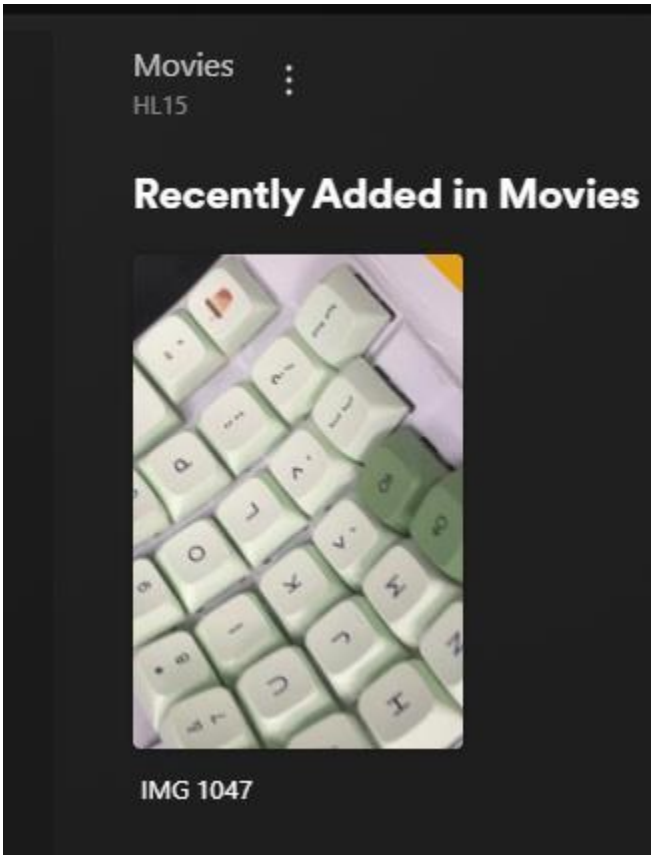




- Then you can scan the library for the existing files



- You will be able to see the videos that are there in your server in the dataset or path you have given.



You can check https://support.plex.tv/articles/200264746-quick-start-step-by-step-guides/ for more guides and info.

IMMICH- self-hosted backup solution for photos and videos

- Go to "Stacks" in the left sidebar.
- Click on "Add stack".
- Give the stack a name (i.e. Immich), and select "Web Editor" as the build method.
- Copy the content of the docker-compose.yml file

```
name: immich
```

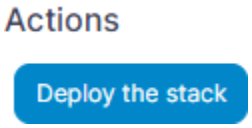
```
services:
  immich-server:
    image: ghcr.io/immich-app/immich-server:${IMMICH_VERSION:-release}
    container_name: immich_server
    environment:
      - TZ=${TZ:-UTC}
    volumes:
      - uploads:/data
      - /etc/localtime:/etc/localtime:ro
    ports:
      - "${IMMICH_PORT:-2283}:2283"
    depends_on:
      - redis
      - database
      - immich-machine-learning
    restart: always

  immich-machine-learning:
    image: ghcr.io/immich-app/immich-machine-learning:${IMMICH_VERSION:-release}
    container_name: immich_machine_learning
    environment:
      - TZ=${TZ:-UTC}
    volumes:
      - model-cache:/cache
    restart: always

  redis:
    image: docker.io/valkey/valkey:8-bookworm
    container_name: immich_redis
    restart: always

  database:
    image: ghcr.io/immich-app/postgres:14-vectorchord0.4.3-pgvectors0.2.0
    container_name: immich_postgres
    environment:
      POSTGRES_PASSWORD: ${DB_PASSWORD:-immich_pass}
      POSTGRES_USER: ${DB_USERNAME:-immich}
      POSTGRES_DB: ${DB_DATABASE_NAME:-immich}
      POSTGRES_INITDB_ARGS: '--data-checksums'
    volumes:
      - dbdata:/var/lib/postgresql/data
    shm_size: 128mb
    restart: always

volumes:
  uploads:
  dbdata:
  model-cache:
```

- Click on "Advanced Mode" in the Environment Variables section.

- Copy the content of the example.env from below and paste into the editor. (github repo - https://github.com/immich-app/immich/releases/latest/download/example.env)

```
IMMICH_VERSION=release
TZ=America/Halifax
IMMICH_PORT=2283
DB_USERNAME=immich
DB_PASSWORD=change_me_strong
DB_DATABASE_NAME=immich
TYPESENSE_API_KEY=supersecret
IMMICH_SERVER_URL=http://immich_server:3001
IMMICH_WEB_URL=http://immich_web:3000
```

- Switch back to "Simple Mode".
- Populate custom database information if necessary.
- Populate UPLOAD_LOCATION with your preferred location for storing backup assets.
- Click on "Deploy the stack".



- Immich will be accessible via http://serverip:2283/
- Make sure the port is open in the firewall.

Post installation you can you can use the guide below to set up
https://documentation.immich.app/docs/install/post-install

Immich user guide

Uploading Pictures

- For First-Time Users
  - On the main home page, you'll see a noticeable box labeled "CLICK TO UPLOAD YOUR FIRST PHOTO." Click on it to begin the upload process.
- For Users with Existing Uploads:
  - If you already have pictures uploaded, simply click the "Upload" option located in the top-right corner of the screen. This will allow you to select images, videos, or other media from your local computer. **YOU MUST BE IN THE PHOTOS TAB FOR THE UPLOAD BUTTON TO APPEAR**

Explore Tab

- Within the Explore tab, you have access to a range of tools that make searching and viewing your photos a breeze. Leveraging the capabilities of ChatGPT, facial recognition, and geolocation technology, finding your pictures has never been more convenient.
- You can also narrow down your search by categories such as Favorites, Recently Added, Videos, Motion Pictures, and Panorama Photos, making it

Map Tab

- In the Map tab, you can harness the geolocation data of your photos to visualize them on a world map. This feature allows you to see exactly where each of your photos was taken, providing a unique and interactive way to explore your memories.
- Inside Map Settings, you have the capability to adjust the map's visual mode (dark or light), decide whether to display only your favorite photos on the map, and specify a date range for the displayed photos. These options enable you to tailor your map experience to your preferences.

Sharing Tab

- Within the Sharing tab, you have the capability to create shared albums with other IMMICH members or share album links with individuals who don't have IMMICH accounts. This feature

simplifies the process of sharing your albums with others, regardless if they use IMMICH or not/

Library

Favorites Tab

- In the Favorites tab, you can conveniently locate all the photos you've marked as your favorites. This tab serves as a dedicated space where your cherished photos are readily accessible.

Albums Tab

- In the Albums tab, you can seamlessly create and organize photos into distinct albums. This feature allows you to categorize your photos and easily access them based on the specific albums they belong to.

Archive Tab

- The Albums tab serves as a storage space for your photos, keeping them separate from the main photo view. This feature is useful for safely tucking away photos that you may not want readily accessible or visible in the main photos tab or to other users.

Mobile App

Settings

The Albums tab serves as a storage space for your photos, keeping them separate from the main photo view. This feature is useful for safely tucking away photos that you may not want readily accessible or visible in the main photos tab

Administration

When logged into IMMICH as an administrative account in the WebUI, you will notice an "Administration" button located in the top-left corner. Clicking this button will navigate you to the administration window, where you can access and configure all the administrative settings for IMMICH.

Users

Within the "User" tab, you have the ability to both view existing users and manage them by adding or removing users as needed. To make changes to a user's settings, simply locate the user and click on the

blue pencil icon situated to the right of their entry. This enables you to edit various aspects of their profile, such as their email address, name, storage label, external path, and even reset their password.

Furthermore, each user account has its dedicated personal settings accessible by clicking on your name or icon located in the top right corner. Within this section, you can perform various actions, including managing your account details, handling API keys, authorizing devices, managing memories, overseeing password settings for your account, and configuring sharing preferences.

Jobs

When in the 'Jobs' tab, you can conveniently monitor all currently active tasks, such as generating thumbnails, extracting metadata, sidecar metadata, tagging objects, encoding clips, recognizing faces, transcoding videos, and handling storage template migration jobs, which can be initiated from the bottom of the page. Furthermore, you have the option to control the concurrency settings by navigating to the top right corner of the page, where you will find a blue box labeled 'Manage Concurrency".

Settings

Inside the settings tab, we have a number of options to choose from. Starting with Job Settings - These settings are the same settings that are found in the jobs tab in the "manage concurrency" button.

- Machine Learning – In this section, you can configure Machine Learning settings, including the option to enable or disable features such as Image Tagging, Smart Search, and Facial Recognition.
- Map Settings – in this section, you have the flexibility to enable or disable the map features, and you can also modify the tile URL if needed.
- OAuth Authentication – In this section, you have the ability to control the login settings using OAuth. You can manage various parameters such as the Issuer URL, Client ID, Client Secret, Scope, Storage Label Claim, Button Text, as well as options like Auto Register, Auto Launch, and Mobile Redirect URL override
- Password Authentication – In this section, you can toggle the option to log in using a username and password, allowing you to either enable or disable this feature.
- Storage Template – In this section, you can customize how your images and videos are saved, as well as configure the desired file structure.
- Thumbnail Settings – Here, you have the option to modify the resolution of both small and large thumbnails, as well as fine-tune the quality percentage. Additionally, there's a setting available for those who prefer the Wide Gamut display.
- Video Transcoding Settings – This tab is for the bit more advanced users. Here we manage the resolutions and encoding information of the video files. We can find things such as:
    - Constant Rate Factor (-crf)
    - Present (-present)
    - Audio Codec
    - Video Codec
    - Target Resolution

- Max Bitrate
- Threads
- Transcode Policy
- Tone-Mappin
- Two-Pass Encoding

This is also two subdirectories inside the Video Transcoding settings being

- Hardware Acceleration (Experimental)
- And Advanced in side advanced are settings most users will not need to change

Server Status

In the Server Status tab, you can access information like the total number of photos and videos stored, as well as the amount of storage used. Additionally, you can view detailed user usage data, which provides insights into the number of photo and video uploads for each individual user.

You can also check the server status, version, and storage usage from any page or tab by looking in the bottom left corner of the WebUI interface.

CLI Commands

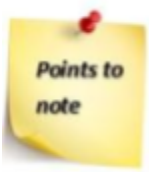
You can find the CLI and bulk upload commands / guides here

https://immich.app/docs/features/bulk-upload

HOME ASSISTANT

An Open-source home automation that puts local control and privacy first. Powered by a worldwide community of tinkerers and DIY enthusiasts.

Method 1- Portainer deployment



This is the bare minimum to get the container up and running, to access the Home Assistant web UI.

1. Go to your portainer UI —> stacks —> “Add Stack” —> Web editor —> Paste your docker-compose.yml

```
version: "3.9"
services:
    homeassistant:
    image: homeassistant/homeassistant:stable
    container_name: homeassistant
    network_mode: host
    environment:
          - TZ=America/Halifax
    volumes:
          - /opt/docker/stack/home_assistant/config:/config
restart: unless-stopped
```

The only things that need to change in this file are the time zone (TZ) and the volume path.

2. Click “Deploy the stack”.
3. Navigate to http://IP:8123
4. Make sure the port is open in the firewall.
5. Follow guided Home Assistant Web UI setup.

Method 2- Deploying in Rocky Terminal

Step 1: Create a Docker Compose file

- Create a directory for your Home Assistant configuration, and inside this directory, create a docker-compose.yml file

```
mkdir -p /opt/docker/stack/home_assistant
cd /opt/docker/stack/home_assistant
vim docker-compose.yml
```

- In the docker-compose.yml file, paste the following content:

```
version: "3.9"
services:
    homeassistant:
        container_name: homeassistant
        image: homeassistant/home-assistant:stable
        network_mode: host
        environment:
            - TZ=America/Halifax
        volumes:
            - /opt/docker/stack/home_assistant/config:/config
            - /etc/localtime:/etc/localtime:ro
        restart: unless-stopped
```

- Save the file and exit the editor.

Step 2: Start the Service

Run Docker Compose from the same directory as your docker-compose.yml file:

```
docker-compose up -d
```

This command will pull the necessary image and start the container in the background.

Step 3: Accessing Home Assistant

Since you're using network_mode: host , you can access the Home Assistant UI by navigating to http://<host-ip>:8123 or http://localhost:8123 if you're on the host machine, replacing <host-ip> with the actual IP address of your host.

Additional Notes:

- Ensure that no other services are using port 8123 on your host.
- If you face any issues accessing Home Assistant, consider checking firewall rules or network policies that might be blocking the port.

WIREGUARD- fast, modern, and secure VPN tunnel

- Go to "Stacks" in the left sidebar.
- Click on "Add stack".
- Give the stack a name (i.e. wireguard), and select "Web Editor" as the build method.
- Copy the content of the docker-compose.yml file
- You will need to set a secure password in the section in the .yml (WGUI_PASSWORD) file if you are planning to do port forwarding.

```
version: “3”
services:
      wireguard:
            image:linuxserver/wireguard:latest
            container_name: wireguard
            cap_add:
                  - NET_ADMIN
            volumes:
                  - ./config:/config ports:
                  - "5000:5000"
                  - "51820:51820/udp"

            wireguard-ui:
                  image: ngoduykhanh/wireguardui:latest
                  container_name: wireguard-ui
                  depends_on:
                        -  wireguard
                  cap_add:
                        -  NET_ADMIN
                  network_mode: service:wireguard
                        environment:
                              -  SENDGRID_API_KEY
                              - EMAIL_FROM_ADDRESS
                              - EMAIL_FROM_NAME
                              - SESSION_SECRET
                              - WGUI_USERNAME=admin
                              - WGUI_PASSWORD=kGMrU6S7(+`Ah93ENLK><8
                              - WG_CONF_TEMPLATE
                              - WGUI_MANAGE_START=true
                              - WGUI_MANAGE_RESTART=true
                  logging:
                        driver: json-file
                        options:
                              max-size: 50m
                  volumes:
                               - ./db:/app/db
                              - ./config:/etc/wireguard
```

- Click on "Deploy the stack".
- If you want to do port forwarding after that you could follow those steps as well.

FRIGATE- open-source NVR built around real-time AI

This is the bare minimum to get the container up and running and will need further configuration based on your specific environment, cameras, needs,storage etc.

Environment:

- rocky 8
- docker files in default location /var/lib/docker
- Example containers location & structure

```
opt
├── docker
│   └── stack
│       ├── frigate
│       │   ├── config
│       │   │   └── config.yml
│       │   └── docker-compose.yml
```

docker-compose.yml

(Note Wireguard and frigate are both using the same ports (5000:5000) note in the doc that if they are deploying both to change one of them to something like 5001:5000)

```
version: "3.9"
services:
      frigate:
            container_name:frigate
            image: ghcr.io/blakeblackshear/frigate:stable
            privileged: true # often needed for VAAPI/USB cams
            restart: unless-stopped
            shm_size: "64mb" # raise if you add more/high-res streams

      environment:
            - FRIGATE_RTSP_PASSWORD=password
      volumes:
            - /opt/docker/stack/frigate/config:/config
            - /etc/localtime:/etc/localtime:ro
            - frigate_media:/media/frigate # or bind a ZFS path instead (see note below)
      Tmpfs:
            - /tmp/cache:size=100m # fast RAM cache for decode frames

      ports:
            - "5000:5000" # Web UI
            - "8554:8554" # RTSP restream
            - "8555:8555/tcp" # WebRTC TCP
            - "8555:8555/udp" # WebRTC UDP

      volumes:
            frigate_media: {}
```

Note: Ensure to use the image in the docker-compose.yml searching will often provide results for the deprecated images.

Explanation of docker-compose.yml

1. **version: "3.9"**

Defines the version of Docker Compose file syntax. 3.9 is a specific version of the

Docker Compose file format.

2. **services:**

Defines the services (containers) to be created.

2.1 **frigate:**

Defines the name of the service (container) to be created.

2.2 **container_name: frigate**

Specifies the name of the container that will be created.

2.3 **image: ghcr.io/blakeblackshear/frigate:stable**

Specifies the Docker image to be used for this service, pointing to Frigate's stable release.

2.4 **privileged: true**

Allows the service to access the host's devices and possibly other privileged functionalities.

2.5 **restart: unless-stopped**

Ensures the container will restart automatically unless explicitly stopped by the user.

2.6 **shm_size: "64mb"**

Allocates shared memory for the container, useful when your application has specific memory requirements.

2.7 **ports:**

Maps the container's ports to the host's ports.

2.8 **environment**

Sets environment variables within the container.

2.9 **volumes:**

Mounts host paths or named volumes to paths inside the container.

/etc/localtime:/etc/localtime:ro binds the host's timezone settings to the container in read-only mode.
/opt/docker/stack/frigate/config:/config binds the configuration directory from the host to the container.
frigate_media:/media/frigate creates a named volume for storing media.
type: tmpfs creates a temporary filesystem in memory, providing high-speed read/write access.

3. **volumes**:

Defines named volumes used by services.

3.1 **frigate_media:**

This is a named volume declaration, which creates persistent storage independent of

the container lifecycle, used here for storing media.

Additional Notes:

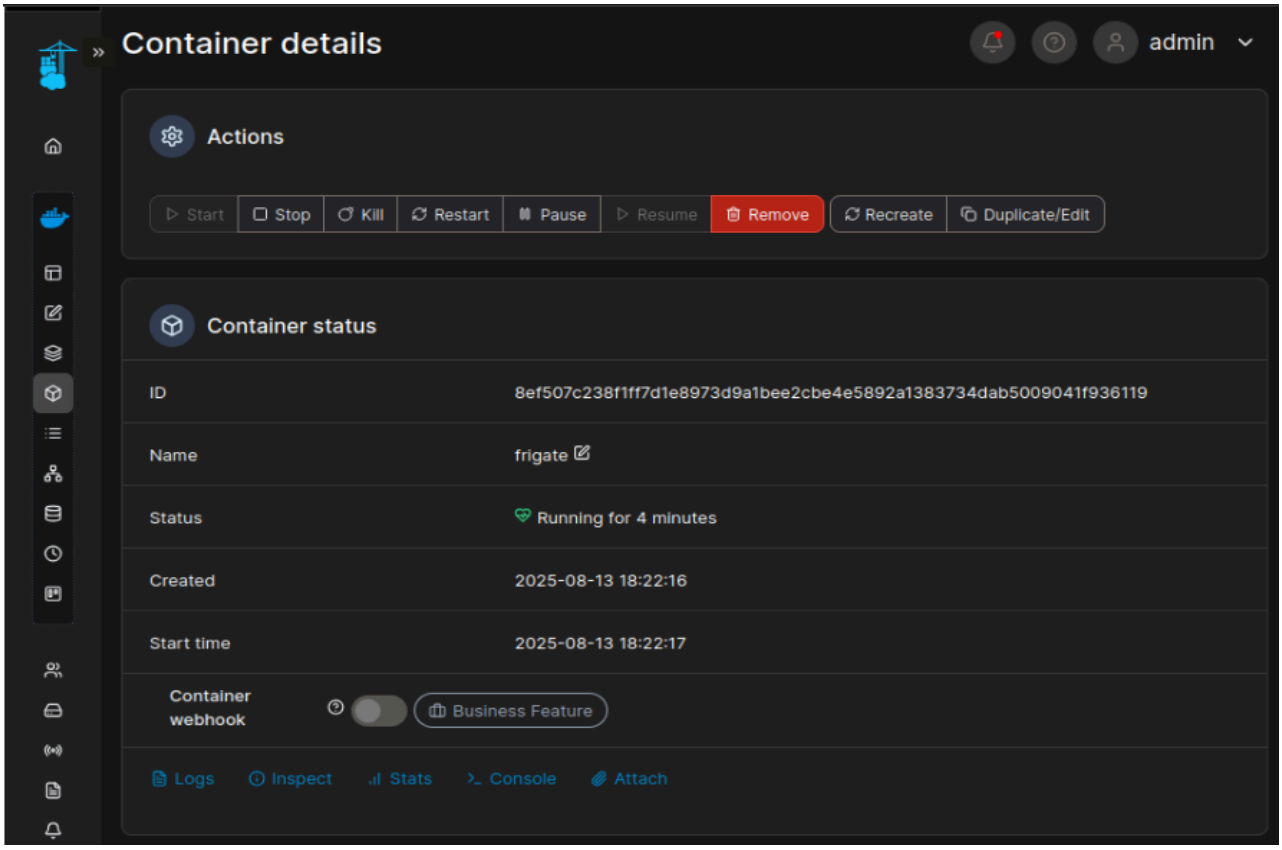
FRIGATE_RTSP_PASSWORD: "password" is setting an environment variable with the password for RTSP access. This will be found in your cameras configuration settings
"5000:5000" makes the web UI of Frigate accessible on port 5000 of the host machine.
"8554:8554" is for RTSP feeds, "8555:8555/tcp" and "8555:8555/udp" are for WebRTC over TCP and UDP, respectively.
Each line in this file serves to configure the behavior, access, or setup of the Docker container or its interaction with the host system or other containers.

Config.yml

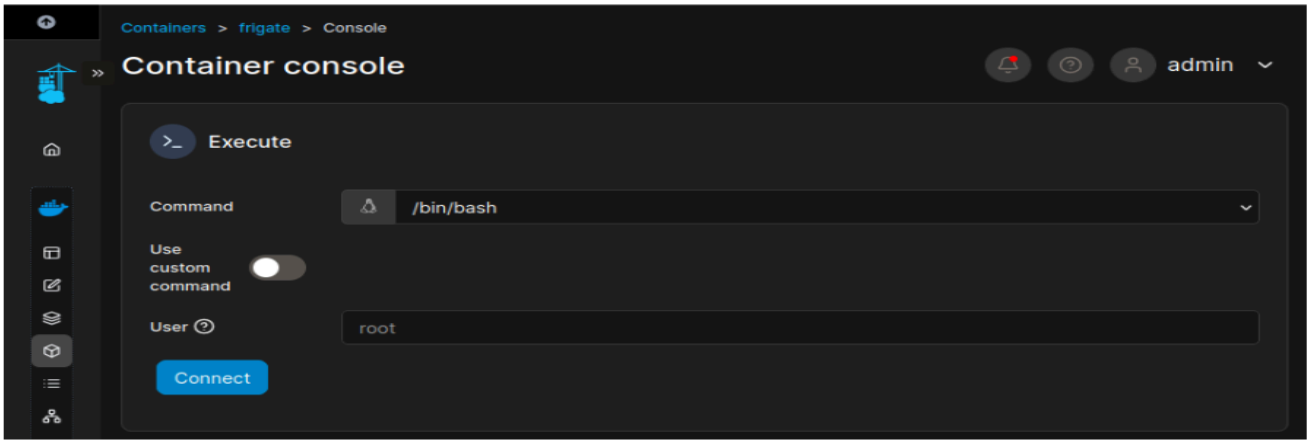
The config.yml file is used to specify configuration settings for Frigate. Frigate needs this file to know how to connect to your cameras, how to process the video streams, how to interact with other services, and many other settings.

Ensure you have this file created & populated with the correct path in your docker-compose.yml before running the container.

- After the container is deployed go to containers --> frigate --> console



- Click connect

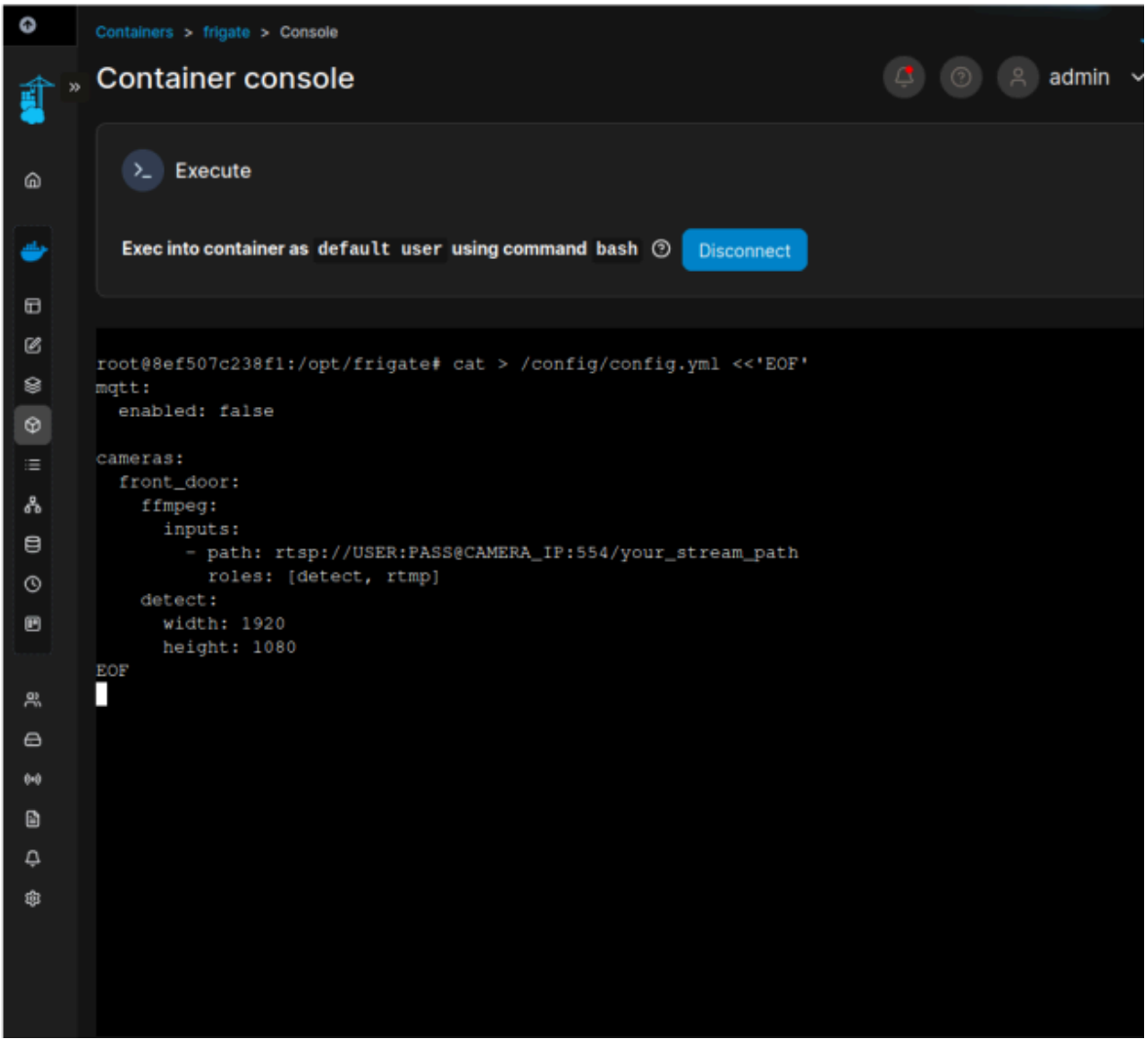


- Paste this press enter then disconnect shell
- Replace USER , PASS , CAMERA_IP , and stream path. If the camera supports a low-res substream, point detect to that and use a hi-res stream just for recording for lower CPU.

```
cat > /config/config.yml <<'EOF'
mqtt:
        enabled: false
cameras:
        front_door:
                ffmpeg:
                        inputs:
                                - path: rtsp://USER:PASS@CAMERA_IP:554/your_stream_path
                                roles: [detect, rtmp]
                        detect:
                                width: 1920
                                height: 1080
EOF
```

- In the future it is recommended to add more cameras or changes to the config.yml via the host and not in the container to do this:

```
# backup first
cp /opt/docker/stack/frigate/config/config.yml \
   /opt/docker/stack/frigate/config/config.yml.bak.$(date +%F-%H%M%S)

# open in vim (you prefer vim)
vim /opt/docker/stack/frigate/config/config.yml
```



- Go back to containers and restart frigate
- Navigate to <IP_of_server>:5000

1. **mqtt**:

This denotes the configuration section related to MQTT, a lightweight messaging protocol typically used in IoT setups.

2. **enabled: False**

This line is configuring whether or not to enable MQTT. Here, it is set to False , meaning MQTT is disabled.

3. **cameras**:

This is the start of the section where you define the configurations for each of your cameras.

4. **name_of_your_camera:**

This is where you name your camera; it is just a placeholder, so you should replace it with a name that makes sense for your setup (e.g., front_door).

5-7. **ffmpeg: ... roles: - detect**

These lines are defining how Frigate should use FFmpeg to interact with the camera. FFmpeg is a multimedia framework used to handle video, audio, and other multimedia files and streams. Here, it is specified to use the camera for detection purposes.

8. **inputs:**

This denotes the start of a list of input streams from the camera that Frigate should use.

9. **path: rtsp://IP_OF_CAMERA:554/rtsp**

This line is specifying the RTSP (Real-Time Streaming Protocol) path to your camera stream. You should replace IP_OF_CAMERA with the actual IP of your camera, and the rest of the path may vary based on the camera model and manufacturer.

10. **roles**:

This specifies the roles that this input stream will be used for.

11. **Detect**

This specifies that the input stream should be used for detection purposes.

12. **detect**:

This is the start of the section where you define the detection settings for this camera.

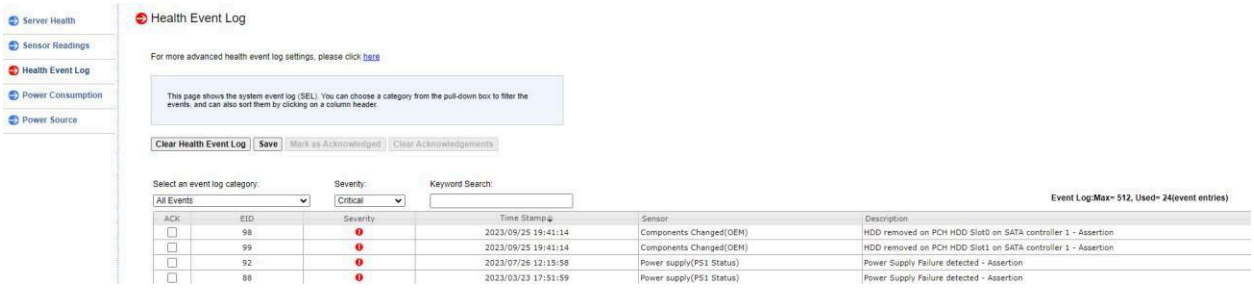
Why is this file needed?

This file is crucial because it allows you to tailor Frigate to your specific needs and hardware. By providing this file, you are informing Frigate about your camera(s), their properties, locations, and how you want Frigate to process their streams, enabling Frigate to function correctly according to your use case. Without this configuration file, Frigate would not know how to interact with your camera(s) or how to process their streams.

SERVER WONT POWER ON

- Try powering the server using the switch
- Check if you are seeing lights on the motherboard and the PSU
- Check if the fans are spinning

- Also check the health event logs.



- If you are not getting any video output it is most likely a motherboard issue or a CPU issue.
- Replace the faulty component and you should be able to power up the server.

HOUSTON UI IS NOT ACCESSIBLE

If your Houston UI is not loading.

- The first thing would be to check if the server IP is accessible.
- Open a terminal or command prompt and try pinging the server IP.
- If you get a response then your IP / network settings are fine.
- Either ssh into the server or open your IPMI web interface.
- Once you have connected to the server via terminal/IPMI check if the cockpit service is running
- **systemctl status cockpit**

```
[root@homelabs ~]# systemctl status cockpit
● cockpit.service - Cockpit Web Service
   Loaded: loaded (/usr/lib/systemd/system/cockpit.service; static; vendor preset: disabled)
   Active: active (running) since Tue 2023-09-26 11:09:05 EDT; 2h 58min ago
     Docs: man:cockpit-ws(8)
  Process: 66010 ExecStartPre=/usr/libexec/cockpit-certificate-ensure (code=exited, status=0/SUCCESS)
 Main PID: 66013 (cockpit-tls)
    Tasks: 2 (limit: 408093)
   Memory: 5.4M
   CGroup: /system.slice/cockpit.service
           └─66013 /usr/libexec/cockpit-tls

Sep 26 11:09:05 homelabs systemd[1]: Starting Cockpit Web Service...
Sep 26 11:09:05 homelabs systemd[1]: Started Cockpit Web Service.
```

- If the status is inactive then restart the service using systemctl restart cockpit
- Your Houston interface should load fine after that.
- If the you were not getting ping response during our troubleshooting steps, then you need to check your networking
- You can use the nmtui interface and check the interface that has the IP set and check if it is up and active.
- If all is good , check if your device is in the correct network.

DRIVES ARE MISSING IN MY ZPOOL

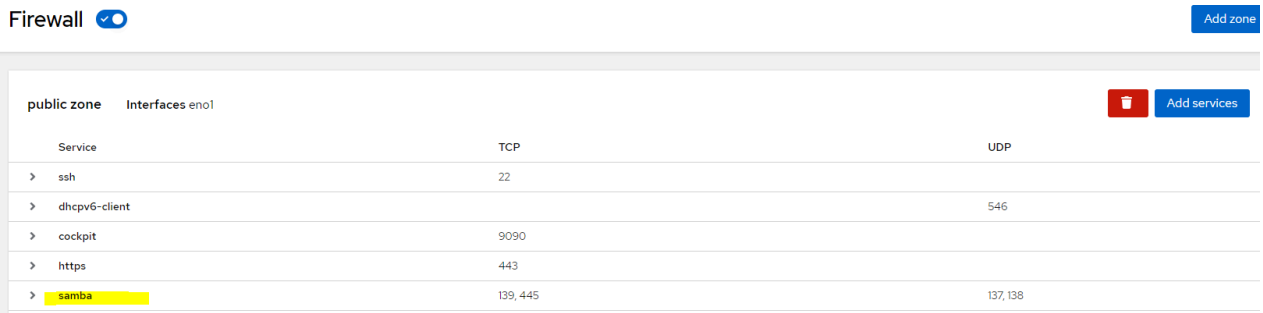
- Try checking which are the drives in your pool that are missing
- You can either use the zpool status or check in the UI using ZFS tab
- Check the server logs using dmesg -T to see if it has any drives drop warning.
- Try re-seating the drive and see if it is getting picked up.
- If the drive is still not detected try swapping it with a drive in another slot and see if it is getting detected.
- If the drives are still not detected it is most likely a failed drive.
- But if the drive gets detected on a different slot. It could be a bad slot.

ZPOOL IS IN A DEGRADED STATE

- Try checking which are the drives in your pool that are missing
- You can either use the zpool status or check in the UI using ZFS tab
- Check the server logs using dmesg -T to see if it has any drives drop warning.
- Check if it is an actual failed drive by running SMART diagnostics on the drive
- Check if you are noticing any uncorrectable or offline sectors if yes then it is most likely a failed drive and needs replacement.
- If the drive is good you would need to troubleshoot if it is the underlying hardware such as backplane/cables etc.

SAMBA SHARES ARE NOT ACCESSIBLE TO MOUNT

- Check if you have samba ports added to the firewall.



- Check if the samba service is running
- Check if you have set a separate samba password for the user
- Make sure you are accessing the share using the right password.

GETTING ACCESS DENIED WHEN ACCESSING THE FILES IN THE SHARE

- Make sure you are having the appropriate permission to access the files.
- Check the permission granted and see if the user getting access denied has access.
- It is most likely that he is not having access or is having access to just the top-level folder and not the child folders and files.
- Modify the permission as per your preference and you should be able to access the files.

HOW DO I UPDATE MY SERVER

- You can update the server by using the software updates tab in the Houston UI

45DRIVES DISK MODULE IS NOT WORKING

- If the 45drives disks module does not show up and does not detect the drives you can perform device mapping manually using **sudo dalias**
- Once the command finishes, reload the 45drives disks module and it should show the drives.

SYSTEM WOULD NOT BOOT INTO THE OS

- If the system is stuck in a boot loop try accessing the server using the IPMI web UI.
- Try to go to the boot menu by pressing F11 during the reboot.
- Once you are in the boot menu try manually selecting the boot drive and boot into it.
- If that works check if the boot order is set properly.
- Press the Delete key to go to BIOS and check the boot order.