Getting Started with Virtualization on TrueNAS SCALE

Community Guidelines and Best Practices
Revision 1.1 – NickF

Why TrueNAS SCALE?

TrueNAS SCALE was released for general use in February of 2022, after several years of development. As of April of 2023, SCALE is now on it's second major release (Bluefin) and with 22.12.2 is now available to Enterprise customers. TrueNAS SCALE has become a mature offering in the Hyper Converged Infrastructure market, and is an excellent place to start for Small to Medium businesses and HomeLabbers.

SCALE uses KVM, or Kernel-based Virtual Machine, as it's hypervisor. KVM has an excellent market adoption rate and is an extremely mature and stable solution. In a recent review by the website Tech Addressed TrueNAS SCALE stacks up well against it's closest competitor as a Virtual Machine host, Proxmox. If you, like me, are interested in both storing all of your things and hosting your virtual machines and application on the same hardware, SCALE was a winner:

> *Choose TrueNAS Scale, as it's the superior storage platform and your virtual machines on top of it should work well enough. While TrueNAS Scale imposes some amount of additional overhead on a host compared to Proxmox, I don't think the performance difference is significant enough that it would be a problem…*

So we've established that you want a single piece of hardware that can do it all. That's what the SCALE moto is all about

> ***S**caled-Out ZFS*
>
> ***C**onverged*
>
> ***A**ctive-Active*
>
> ***L**inux Containers*
>
> ***E**asy to Manage*

For the sake of this article, let's focus our letter C, converged. This is derived from the term Hyper-Converged, which generally refers to software-defined IT infrastructure consisting of a hypervisor, software-defined storage, and virtualized networking. Let's take a look at what SCALE has to offer.

Getting Started: The Use Case

Before we can get on with the digital fun, we have to get our analog hands dirty. We need to build ourselves a server! It's difficult to make broad recommendations on exactly what to buy without knowing what your underlying goals might be.

Let's start with some general rules of thumb for some minimum requirements for a homelab installation. I'll assume that your goal is to store some files, host your own private DNS with something like PiHole, maybe do some home automation, and have a media server at your disposal. IXSystems says that the minimum hardware requirements for SCALE, with virtually no services except file sharing as:

Minimum Hardware Requirements

Our recommended system requirements to install TrueNAS:

Processor	*Memory*	*Boot Device*	*Storage*
2-Core Intel 64-Bit or AMD x86_64 processor	*8 GB Memory*	*16 GB SSD boot device*	*Two identically-sized devices for a single storage pool*

Since we will be running other services, and hosting our own VMs that is simply not enough and more specific information is going to be required. Here are my minimum recommendations based on the workload described above:

Processor	*Memory*	*Boot Device*	*Storage*
4-Core Intel 64-Bit or AMD x86_64 processor ***Intel VT-D/AMD-Vi for IOMMU/ SR-IOV Support***	***32** GB Memory* ***ECC Preferred***	*16 GB SSD boot device*	*Two or more identically-sized* ***SSDs** for a single storage pool*

There are many options that can be found inexpensively in this performance footprint. Let's do this as a "Good", "Better", "Best" configuration.

"Good"

An older corporate desktop like the HP 800 G4 SFF ticks a lot of the boxes, but doesn't include ECC memory. Depending on your risk-tolerance and your budget, that may be an enticing deal, as of the writing of this article they can be purchased for under $200. It comes equipped with Intel I219LM Gigabit Network standard, has 4 DIMM slots, supports 3 2.5" drives, 2 M.2 3.0 x4 slots, and some additional PCI-E connectivity for a graphics card. It's small and quiet and is an excellent platform for a first server.

“Better”

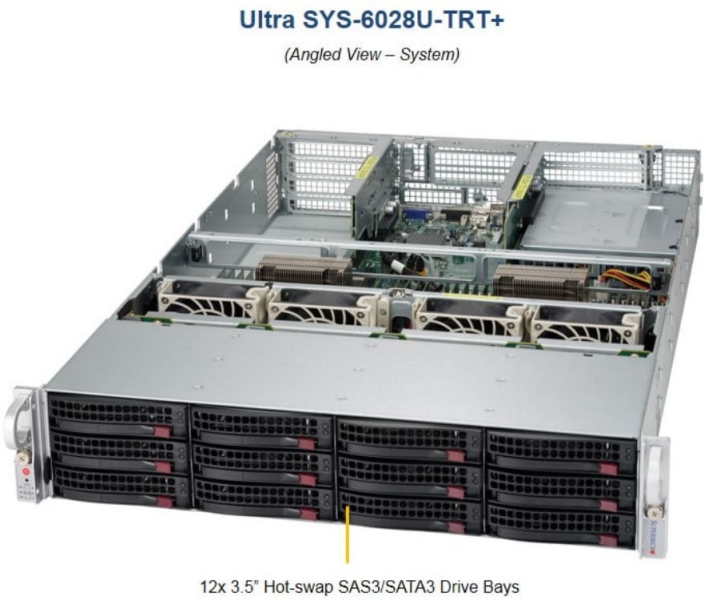
If you don’t want to color outside of the lines and are looking for something that you can buy that’s supported, a TrueNAS Mini is a great choice for around $1,000. You get out-of-band management through IPMI, ECC RAM, a 1 year manufacturers warranty, and you get the warm and fuzzies for supporting this community.

> *Certainly, like Papa Johns is famous for saying, Better Ingredients, Better NAS, IXSystems.*

“Best”

If the cheap corporate desktop doesn’t sound very good to you, and you don’t see enough value in buying first party hardware, you can certainly step it up a notch. This last option can fall in line between the first two options in price, but what you will sacrifice is space, increased power consumption, and noise. It’s picking up a used enterprise server. In other words, folks out there who do this for a living have solved the problem of building a computer that acts a server. Let’s not reinvent the wheel. The biggest thing to look out for is finding one that has an HBA, or host bus adapter, and not a RAID card.

You can find deals on /r/homelabsales, on eBay or even on Craigslist all of the time. Something like a Supermicro Storage Server with an X10 generation board can be found for under $700 without having to go hunting or buy from less reputable sellers.



Those certainly your only options, and you will find many, many more viable options and opinions on the TrueNAS community forums, Reddit, and other "techie" outlets like ServeTheHome, LevelOneTechs, etc. This guide is **not meant to be** a conclusive hardware recommendations guide, but I wanted to start the conversation and throw out some options I would personally consider.

A Bit On Networking

Before proceeding, I would advise that you familiarize yourself with the OSI Model. We are primarily going to be focusing on Layer 2 of the network stack, which can be summarized as the communication of frames (not packets!) between MAC addresses (not IP addresses!).

For my lab setup, I will be using a mini enterprise switch called the Brocade/Ruckus ICX 7150. For some more information on why I chose this particular switch, there is a fantastic resource over at the ServeTheHome community forum.

The primary goal here was to have a switch that supports L3 functionality, VLANs and Link Aggregations. For most homelab type deployments, L3 isn't necessary but I will be following this resource up with a "Datacenter-in-a-box" so more on that in another piece.

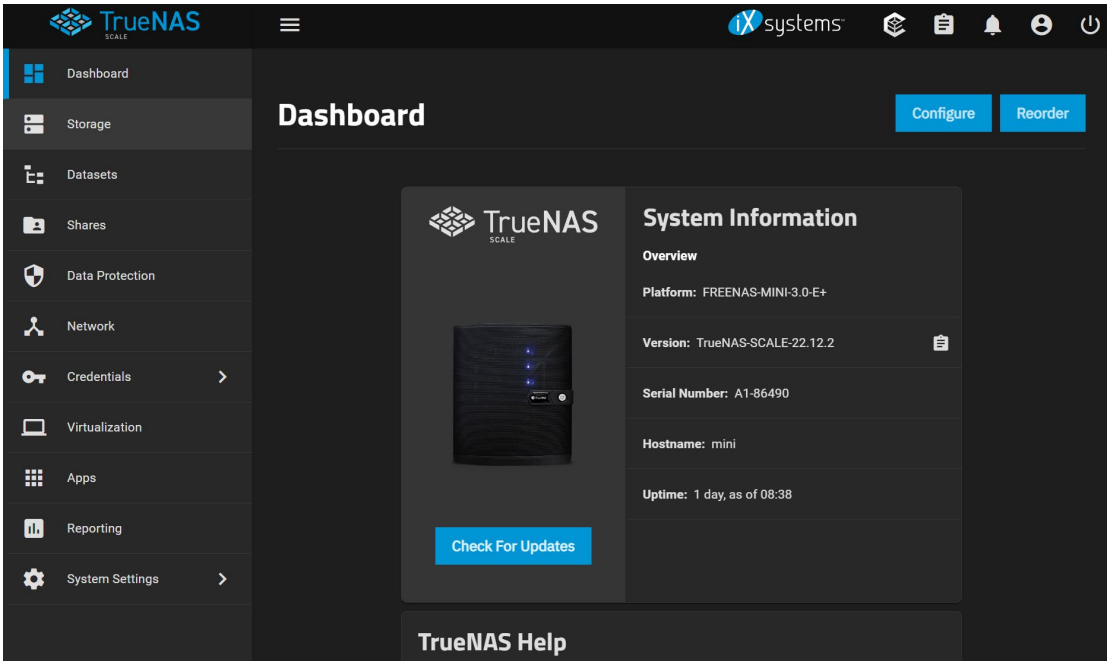
So why are VLANs necessary? Segmentation of services. In alignment with best-practices, splitting off your network into logical pieces is one of the primary layers of the security onion. For my deployment, I have logically separated my network into the following chunks:

VLAN Number	Network	Name	Description
10	10.69.10.0/24	PCs	Where all of my computers and laptops live.
20	10.69.20.0/24	IOT	Where all of my cellphones, smart speakers, sensors, etc live
30	10.69.30.0/24	Guest	Guest Wireless Services
40	10.69.40.0/24	Management	Where my IPMI, TrueNAS management, Switch management, live
50	10.69.50.0/24	RESERVED	This IP Space is reserved for future use.
60	10.69.60.0/24	Servers	Where all of my virtual machines and applications run.

In order to route all of these VLANs, you could use the L3 features of a switch, or you could rely on your firewall to do the routing for you. Routing is out of scope of this article, and we will be focusing in on L2 technologies here. If you want to read more, pfSense has a getting started guide on routing VLANs here, and I have written a resource on ServeTheHome on using VLANs for a device with a single interface.

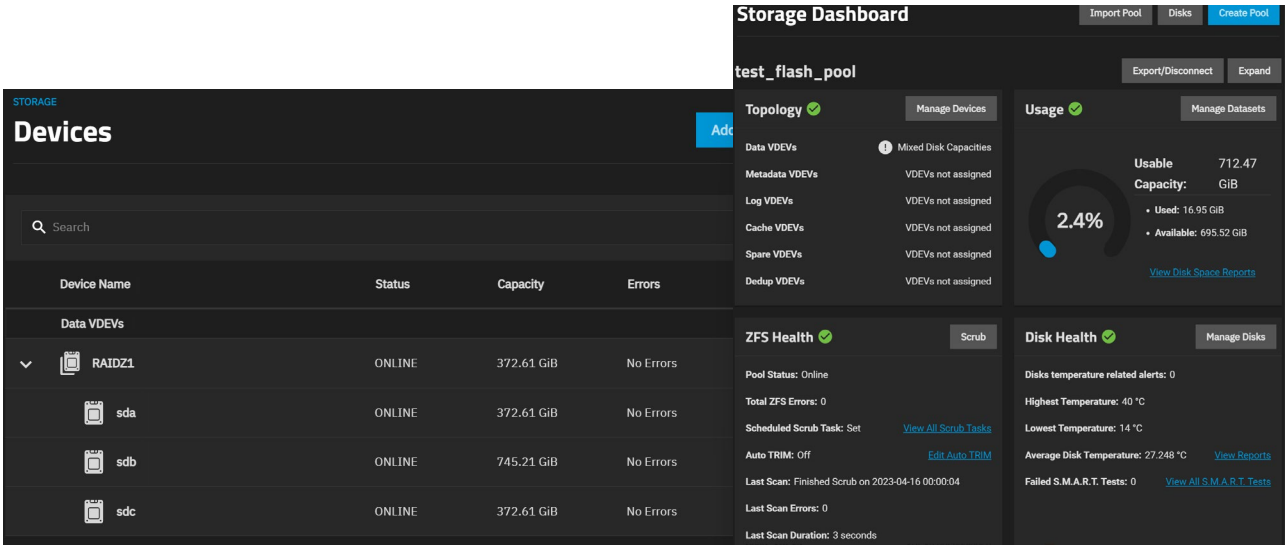
Initial Setup

For the sake of brevity, we're going to assume that you have been able to install SCALE to your chosen hardware and have the web interface available. IXSystems has an official "Getting Started" documentation section if you need help. I will be using an older FreeNAS Mini for demonstration purposes here, with SCALE 22.12.2 installed.



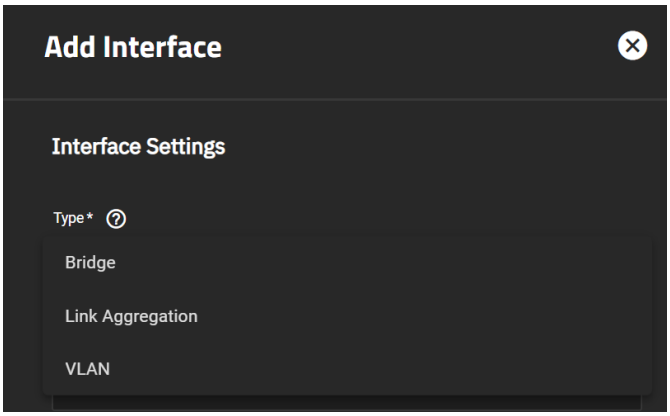
For storage I have selected to install 3 Intel S3710 Enterprise Grade Solid state drives, configured as a RAIDZ-1. For the best performance, especially in a VM workload it is better to use a mirrored VDEV layout. Do as I say, not as I do 😊

For more information on performance and VDEV layouts, IXSystems released a white paper describing the differences between each of the different layouts. If you are new to ZFS in general, I wrote an introduction back in 2020 on ServeTheHome.



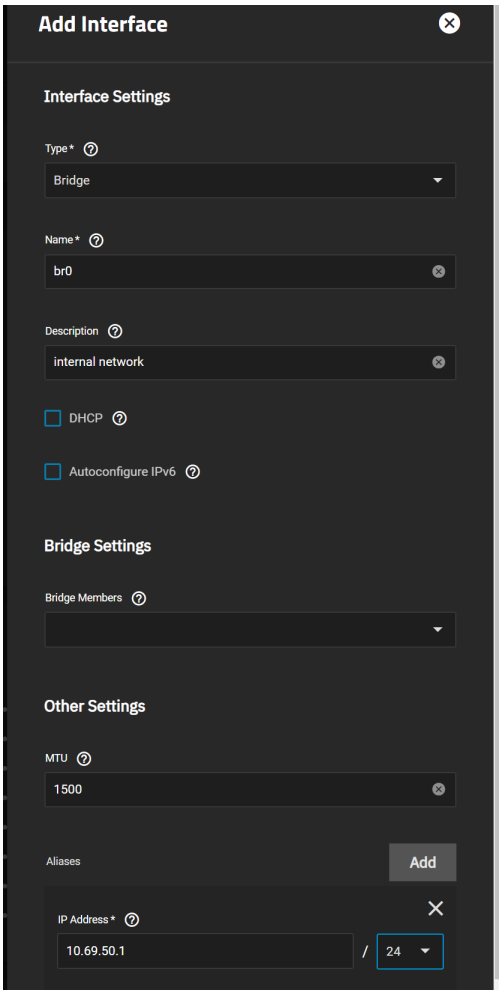
With my pool ready, we can begin to configure our network stack.

If you go to the networking tab of SCALE and press “ADD” you will see the different types available, a Bridge, a Link Aggregation, and a VLAN. We’re going to use all three.

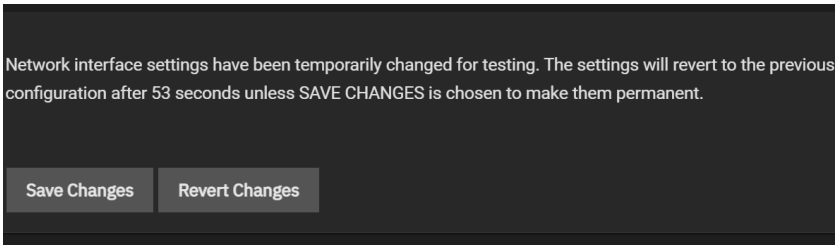


Let’s first talk about a bridge network. By default in TrueNAS SCALE, a virtual machine running inside of SCALE ***will not*** be able to talk to SCALE itself. This can be both a good and a bad thing, and is a topic which I’ve talked about at length. We’re going to be building a bridge that is **not attached to parent network card**. Essentially we will be creating a virtual switch that will allow our VMs to talk to either TrueNAS SCALE or each other *without having to leave* our server.

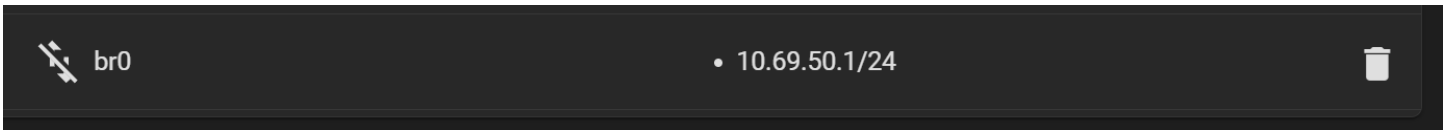
We’re going to name our bridge br0 and give it a description of internal network. Do not add any members to the bridge. If we want our guest virtual machines to be able to talk to a file share, we will have to give our internal network an IP address “Alias” in a network range that’s not already in use. We’ll use our reserved 10.69.50.0/24 network, and save.



TrueNAS will ask you to Test Changes, and then Save changes.



Now we have our internal network ready, it shows as offline, but that's only because nothing is connected to it yet.

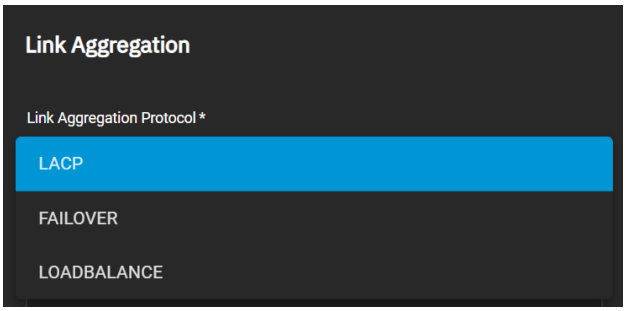


Moving on to the world outside of our plucky little server. In TrueNAS SCALE there is a concept of a parent/child interface. A parent is essentially either a physical network card or a group of physical network cards. A child is essentially a logical subset of that parent that meets a specific criteria. In the case of our topology, we will have a "grand-parent" interface as well, which is a parent of a parent. Sounds confusing! Let's visualize it.

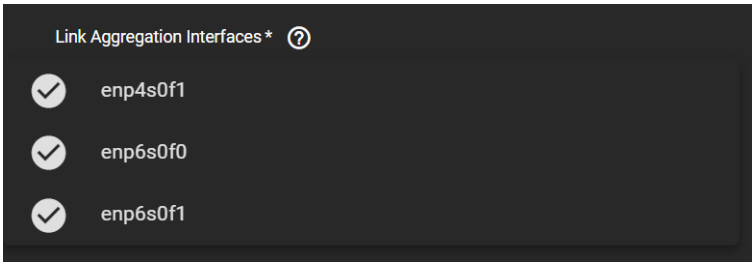
I have four physical network interfaces:

These 4 physical interfaces are going to be our "grand-parent" interfaces. We are going to want to bond the 4 network interfaces together so that we have increased redundancy if one of them comes unplugged, and to also increase our throughput. We are essentially widening the street from a 1 lane road to a 4 lane highway. However, we are not increasing the speed limit from 1 Gigabit to 4 Gigabit, we are just opening up the road to allow more cars to drive through.

Their child is called a bond, sometimes also called a link aggregation or port channel. In the networking tab, click on Add again, and select Link Aggregation. We're going to name our interface bond0, and give it the description of highway. If you have a switch, like mine, that supports LACP then that is what you should select. If you are using an unmanaged switch, you can select failover or loadbalance.



Add your interfaces to the group:



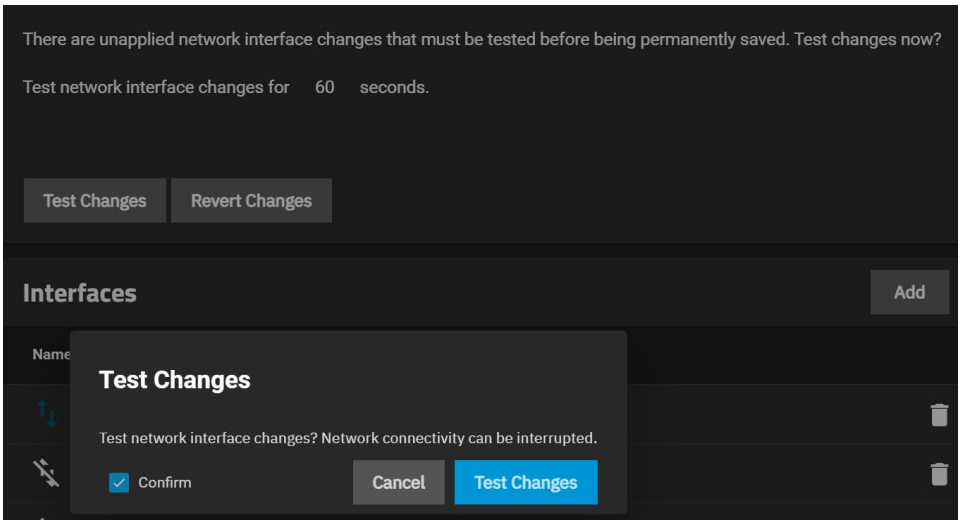
And press save.

Now I am going to go over to my switch and configure it to accept LACP PDUs by creating a Link Aggregation on it as well. This process will differ from brand-to-brand, but I will show you how it's done on Brocade/Ruckus. SSH into your switch and enter config mode, type

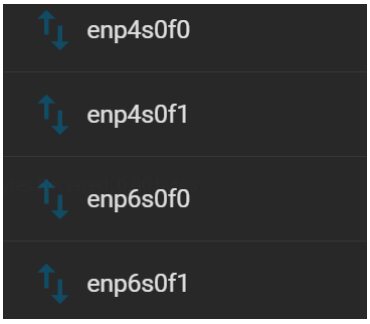
lag highway dynamic id 1
ports ethernet 1/1/1
ports ethernet 1/1/2
ports ethernet 1/1/3
ports ethernet 1/1/4

You should get a message stating: LAG highway deployed successfully!

Now plug your cables in between the ports you configured on your server and the ports you added to the lag on your switch. Go back to TrueNAS and press "Test Changes" and then "Save Changes"



The interfaces should now all show as up:

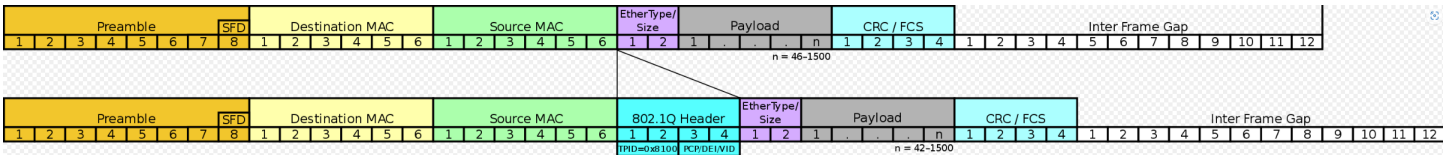


On our logs on our switch, we should see the peering come up and the interfaces becoming active:

```
Dynamic Log Buffer (4000 lines):
Apr 17 12:42:44:I:Security: running-config was potentially changed by manager from ssh client 10.69.
10.57
Apr 17 12:42:15:I:STP: VLAN 4094 Port lg1 STP State -> FORWARDING (DOT1wTransition)
Apr 17 12:42:15:I:STP: VLAN 4094 Port lg1 STP State -> LEARNING (DOT1wTransition)
Apr 17 12:42:12:I:System: Logical link on dynamic lag interface ethernet 1/1/4 is up.
Apr 17 12:42:12:I:System: Logical link on dynamic lag interface ethernet 1/1/2 is up.
Apr 17 12:42:11:I:System: Logical link on dynamic lag interface ethernet 1/1/3 is up.
Apr 17 12:42:11:I:STP: VLAN 4094 Port lg1 STP State -> BLOCKING (DOT1wTransition)
Apr 17 12:42:11:I:Trunk: Group (1/1/2, 1/1/3, 1/1/4) created by 802.3ad link-aggregation module.
Apr 17 12:42:11:I:System: dynamic lag 1, has new peer info (priority=65535,id=d050.99d4.c96a,key=9)
(N/A)
```

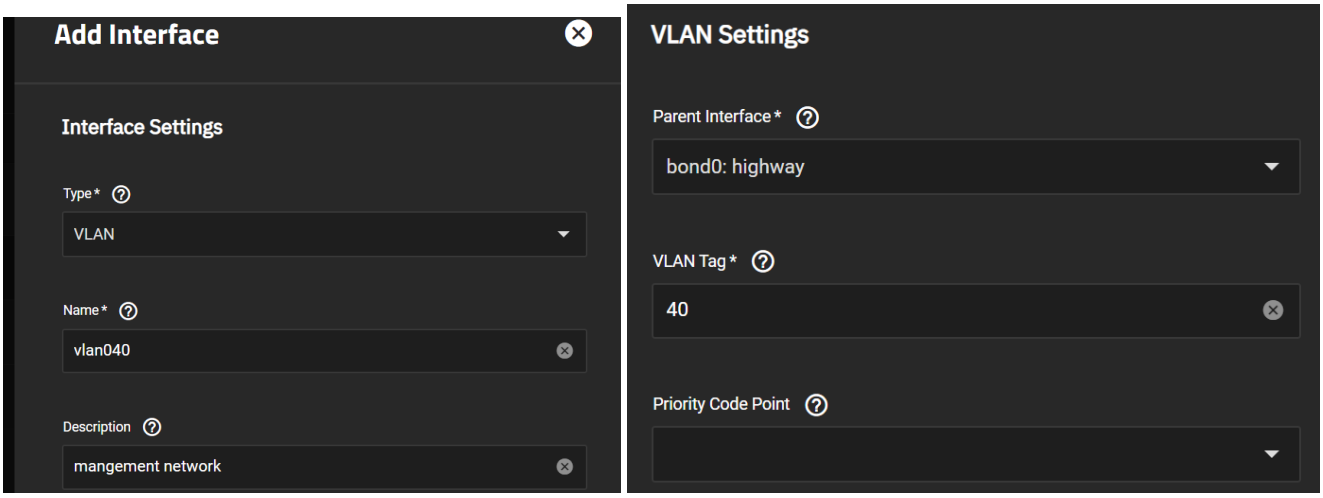
Understanding VLANs, 802.1q

We talked a bit prior about what VLANs are used for, but we haven't touch upon what they actually *are.* Remember when I said we would be discussing things about Layer 2 concepts? The maximum standard frame size for ethernet is 1512 bytes. The IEEE 802.1q standard extends that size to 1522. The Linux kernel and our switch both comply with that standard, so we can use those extra ten bytes to tell them which VLAN a particular frame is supposed to go to. If there is a frame sent from a VM on our server, with a Q-tag for VLAN 60, the switch will know that and forward the frame into VLAN 60.

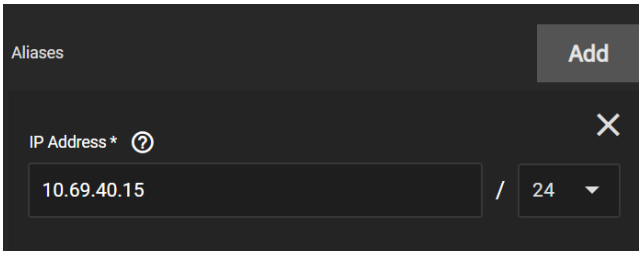


For the sake of this article, we will be focusing entirely on tagged VLANs, or in Cisco parlance, trunks. To do all of this in TrueNAS, we need to make some child interfaces. The criteria for traffic going into and out of each of those child interfaces is determined by what Q tag they have.

Back to our TrueNAS Network tab we should press Add again. This time our type is going to be a VLAN. I'm going to name mine "vlan040" for my "management network" as the description. I am going to select my parent interface as bond0 and provide it a VLAN tag of 40.



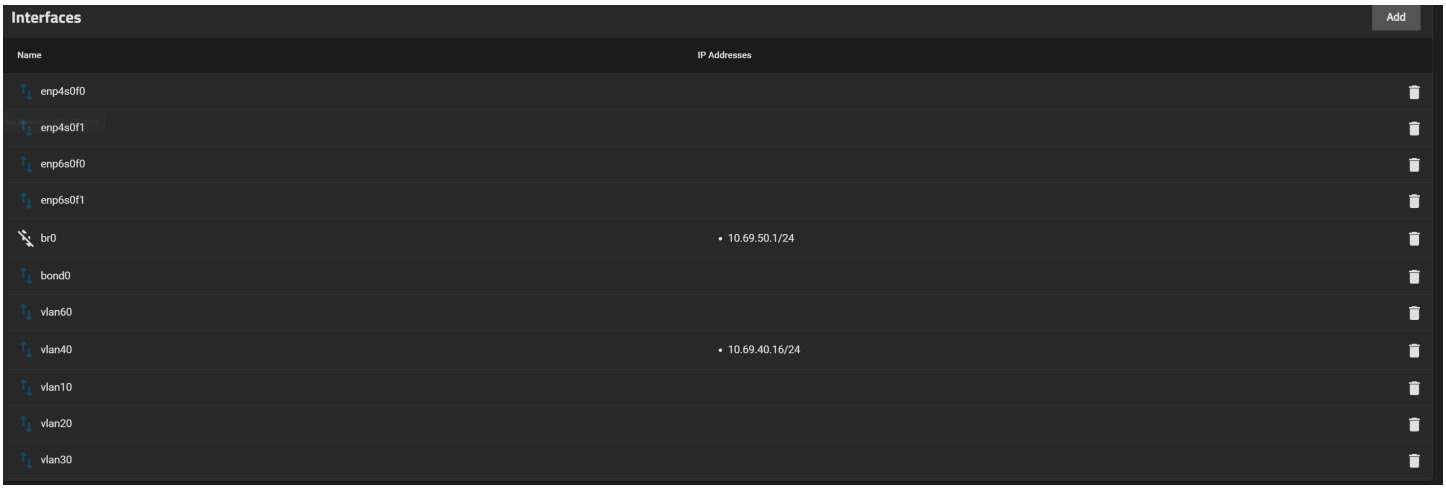
Since I intend to use this VLAN as my management interface from my PC, I am going to need to give it an address, so under the alias section I've added a free IP.



Press Save, Test, and Save again. Repeat these steps for each of the VLANs you wish to host virtual machines in. Supplying an additional alias for each of those is optional and is not necessary in order to proceed. With the TrueNAS side configuration for VLANs complete, we now need to do the same for our LAG on our switch.

Vlan 10
tagged lag 1
Vlan 20
tagged lag 1
Vlan 30
tagged lag 1
Vlan 40
tagged lag 1
Vlan 60
tagged lag 1

When all is said and done your TrueNAS UI should look like this:

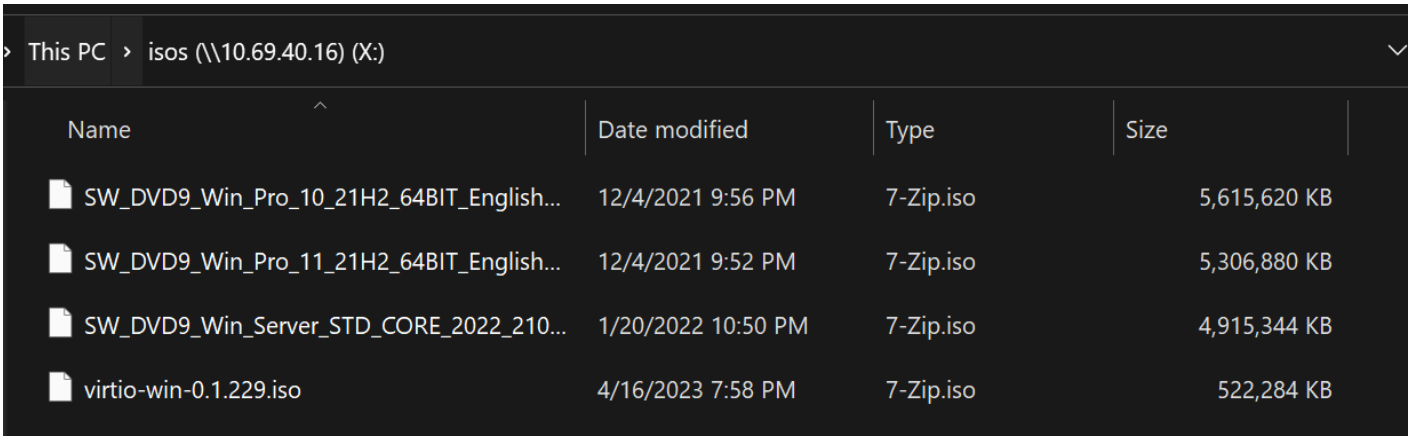


And all of the VLANs are tagged on the uplink from the switch. Whew! We're finally ready to move off of networking.

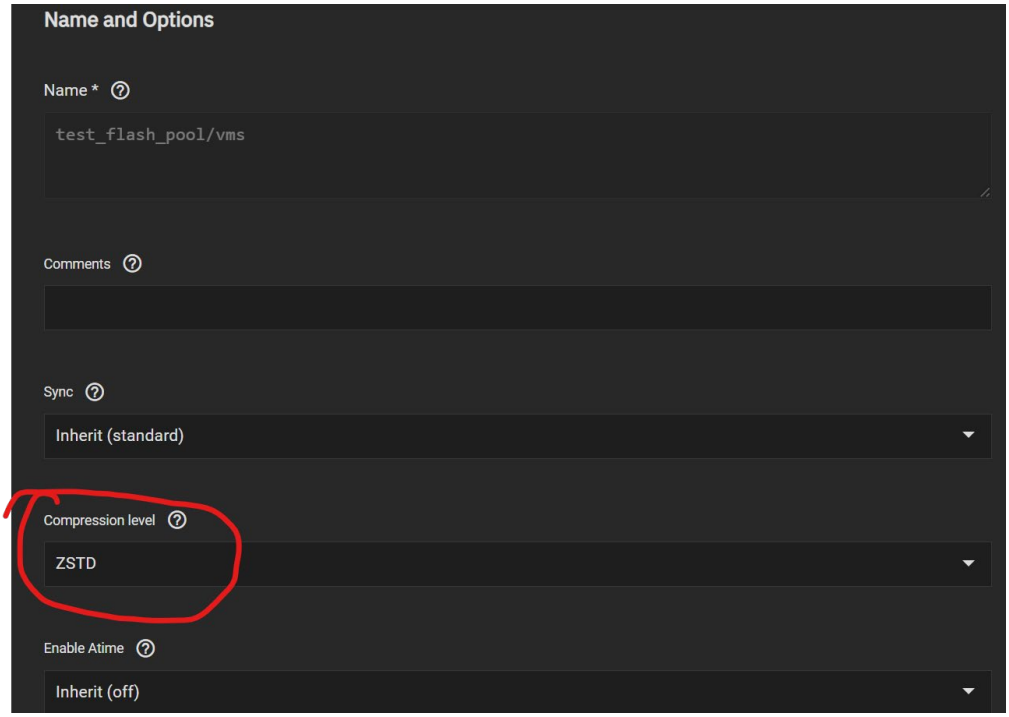
Storage Configuration: Your First Datasets

Generally I like to lay out my pools such that there are several cascading trees of datasets. This allows me to create snapshot retention policies based on the classification of data later on. The first couple of datasets I've created are called "isos" and "vms". I'm going to create a SMB share for the ISO dataset and I am going to use the vms dataset as a parent for my virtual machines hard drives, this is done through creating zvols.

I've uploaded some Windows ISOs and the virtio drivers with guest tools to my ISOs folder.



And on my new vms dataset I have changed the default compression algorithm from lz4 to ztd



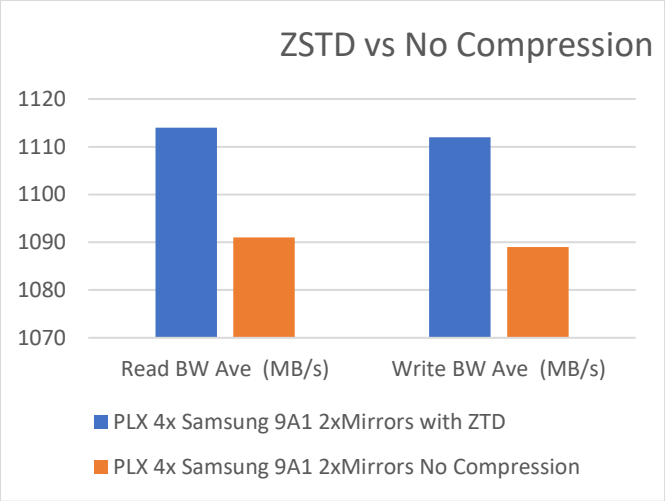
In doing so, we are sacrificing performance for additional space savings. This is a personal preference on my part, but I think that the results are worth it. The faster your CPU is, the less you will feel the pain.

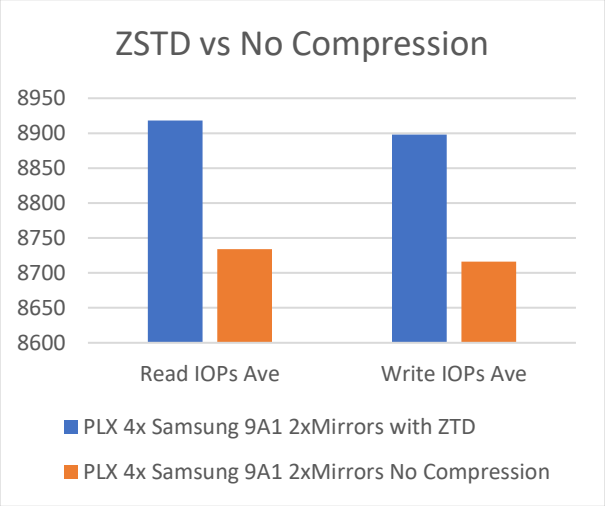
From Facebook Data:

	Ratio	Comp Speed MB/s	Decomp Speed MB/s
lz4	2.10	444.69	2165.93
zstd	3.14	136.18	536.36
zlib	3.11	23.21	281.52
xz	4.31	2.37	62.97

On my AMD EPYC System, I have run benchmarks, and I feel that the difference in performance was reasonable. On my TrueNAS mini, the verdict is still out, though I may update this section after I've done some additional testing.

fio --bs=128k --direct=1 --directory=Blahblah --gtod_reduce=1 --iodepth=32 --group_reporting --name=randrw --numjobs=12 --ramp_time=10 --runtime=60 --rw=randrw --size=256M --time_based





Provisioning our ZVols

The TrueNAS VM Creation wizard generally does this for you, but I prefer to manually create my Zvols, for one simple reason: Thin Provisioning. **Because of the magic of compression, we can potentially store more data on our pool than there is actual space.** Neat huh? But to quote Spiderman:

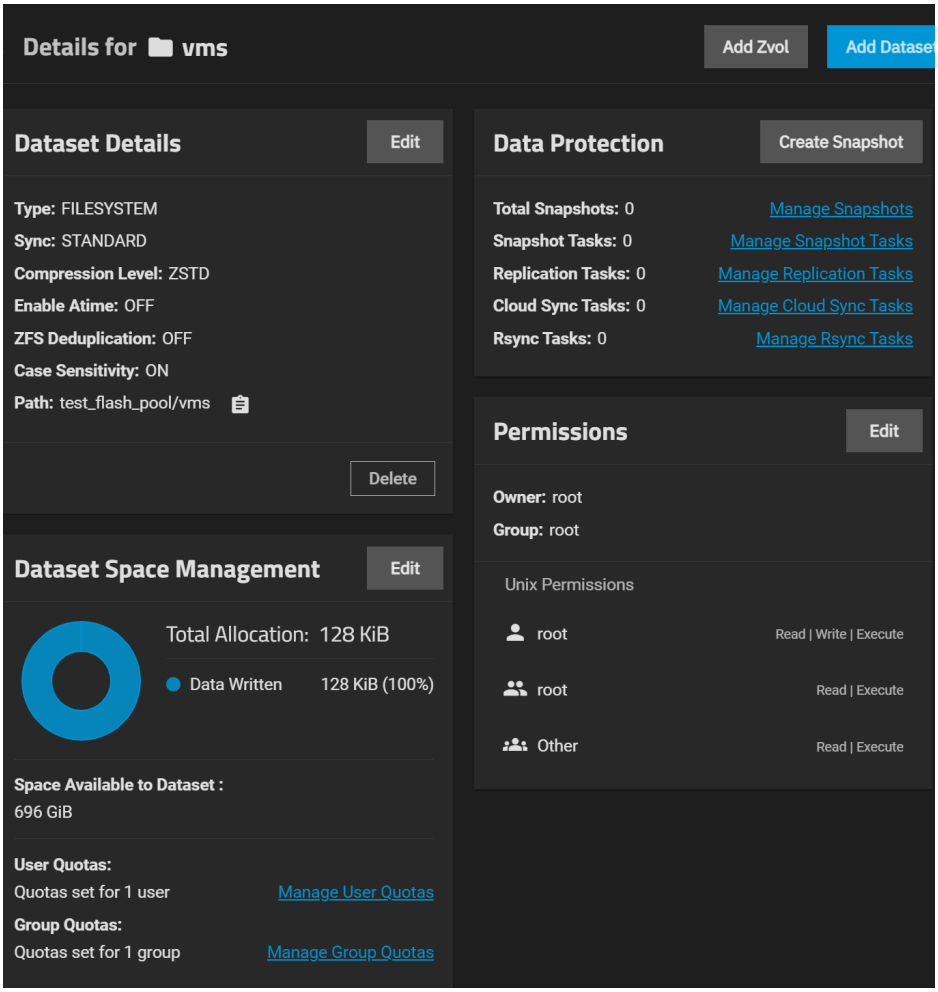
If we overprovision our zpool and we use more space than there is available, we will cause a write blockade. That means all of our VMs will crash and our pool will be stuck in a read-only state in order to try and prevent data corruption. But since TrueNAS is smarter than we are, it will alert us once our drive is 80% actually full, and really, really alert us when it's 95% full. There's a really cool script that will make those alerts even louder and email you every night when your pool starts to get full. Thanks @JoeSchmuck!

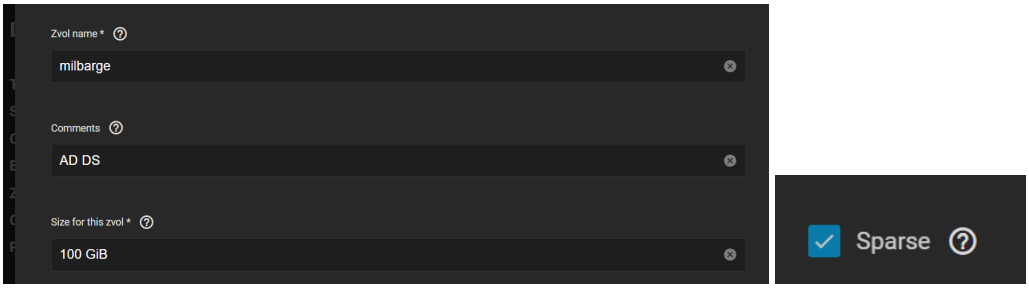
Running the command on my production storage server, I am seeing the following:

root@prod[~]# zfs get all optane_vm | grep compressratio

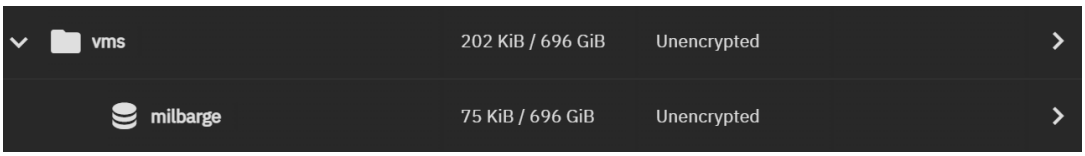
optane_vm compressratio 1.48x

In other words, I am getting 48% more storage right now than I have physical space. By thin provisioning I can actually use this space for more VMs 😊 Back to the TrueNAS UI I am going to make my first zvol

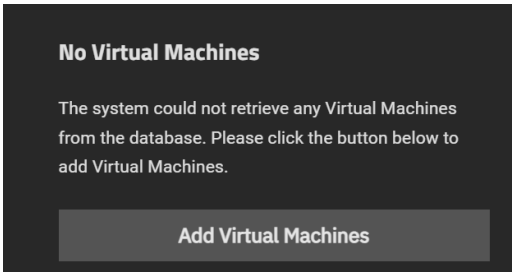




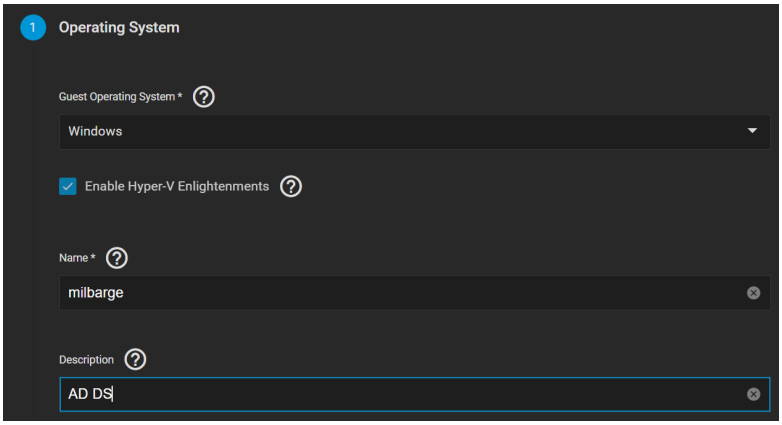
With my new zvol created, I can move on to the VM Wizard



In the VM tab, I can see there are no VMs yet. Press Add Virtual Machine

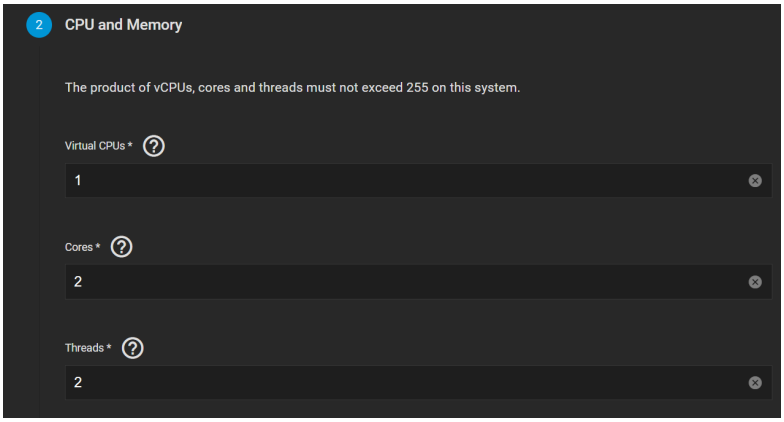


I'm going to select Windows and I am going to Enable Hyper-V Enlightments, this may help improve performance. I'm going to name it milbarge

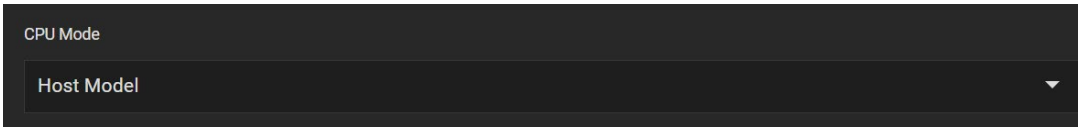


For now I am going to leave the rest of the settings at default in the Operating System tab. Press Next.

I am going to give my machine 2 cores and 2 threads.

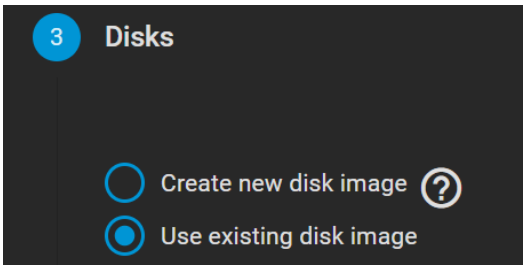


And I am going to select Host Model for the CPU mode

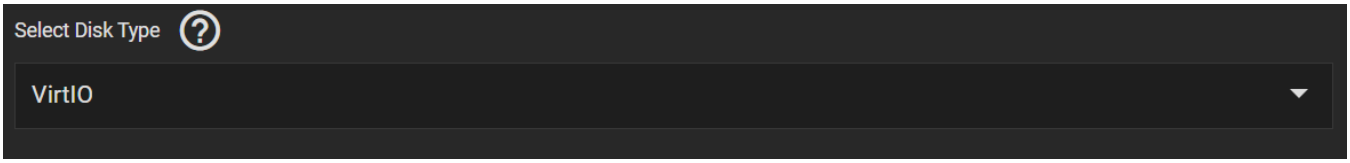


I'm going to leave the memory size at the default 4GB and I am going to press next.

Because we have already created our Zvol, I am going to tell SCALE to use an existing disk image

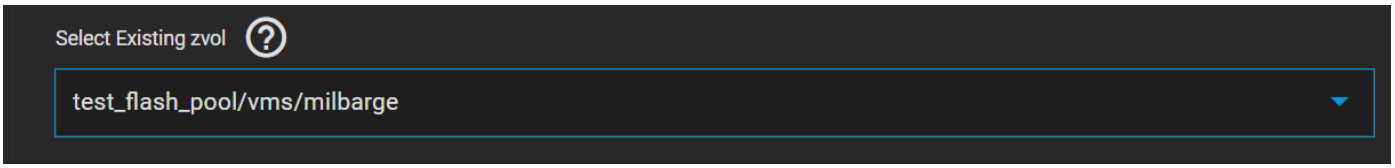


Disk Type requires a bit more explanation. I am selecting VirtIO



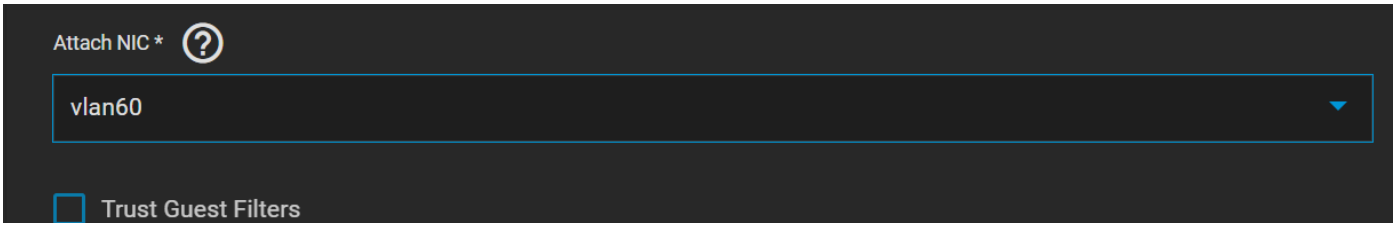
There are huge performance gains to be had by selecting VirtIO over AHCI. This is because VirtIO is a paravirtualization technology. The alternative is literally a software emulator for an fake LSI SCSI driver which adds a substantial amount of overhead. The only downside is that we have to install a driver at the time of installation, which is a trivial task.

Finally we are going to select our Zvol from earlier and press next.

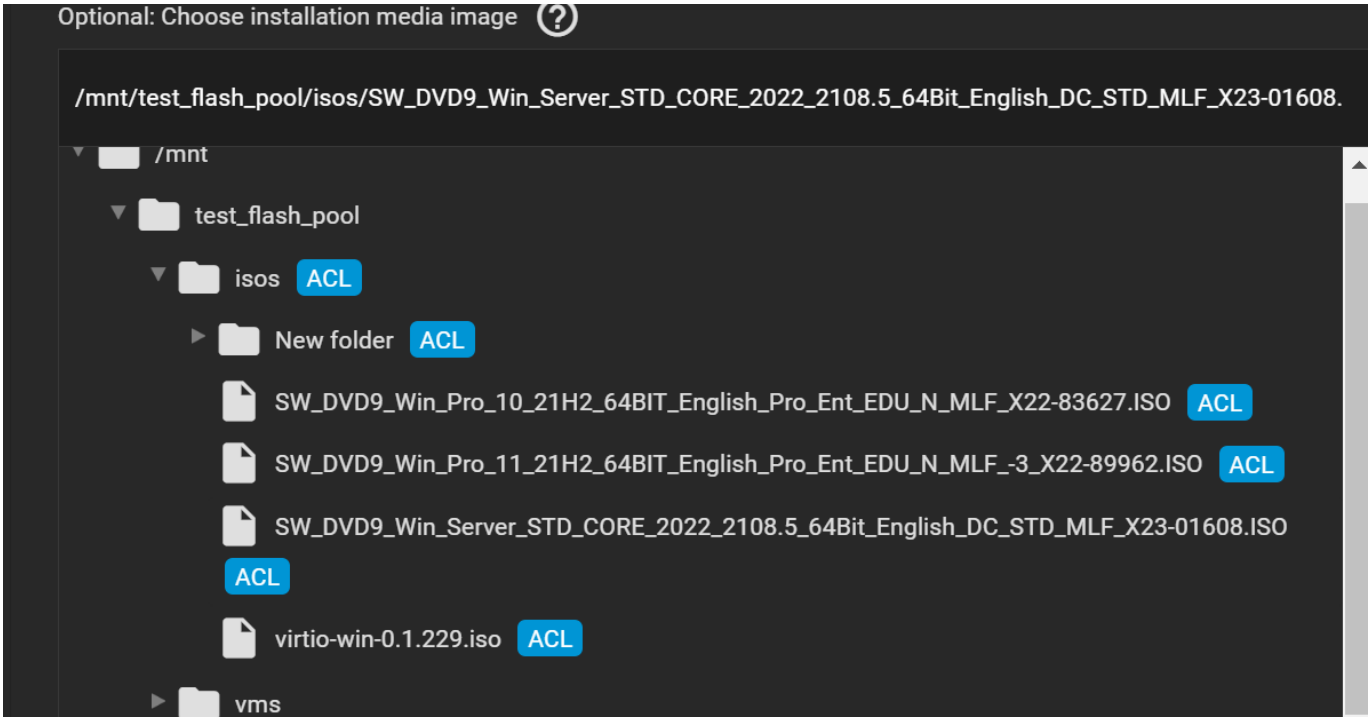


Like before, we are going to select the paravirtualization-based VirtIO adapter type

And I am going to select the server vlan I created earlier, vlan 60



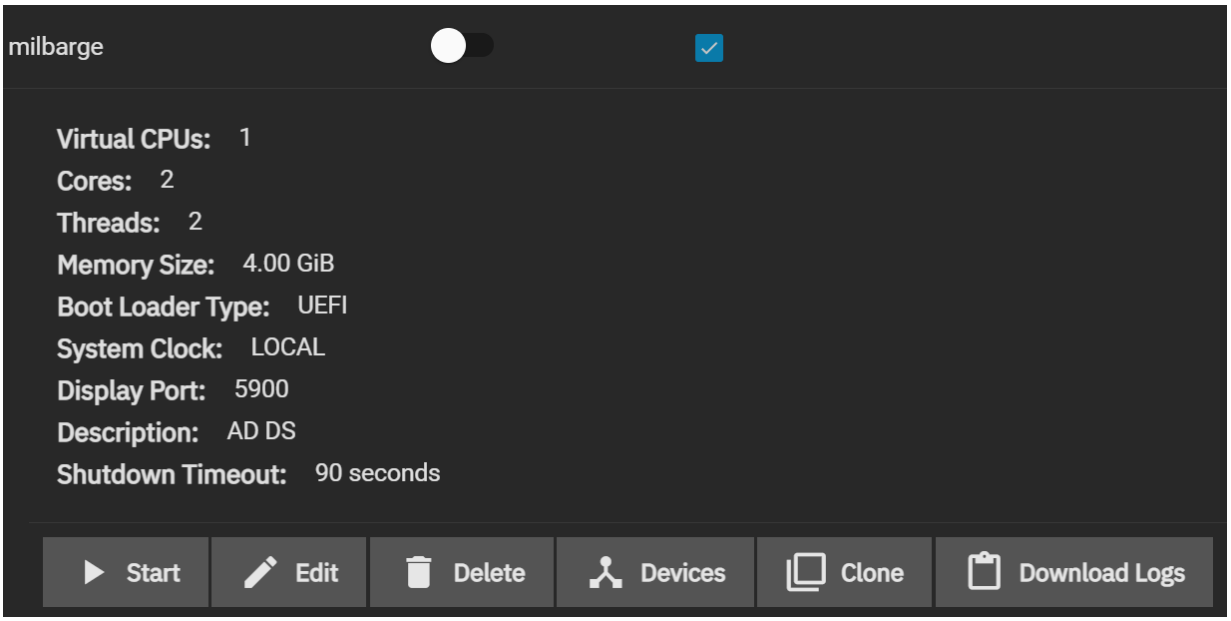
Now we will be able to select out Windows Server installation ISO and press Next, Press Next again because we aren't passing through a GPU, and Save.



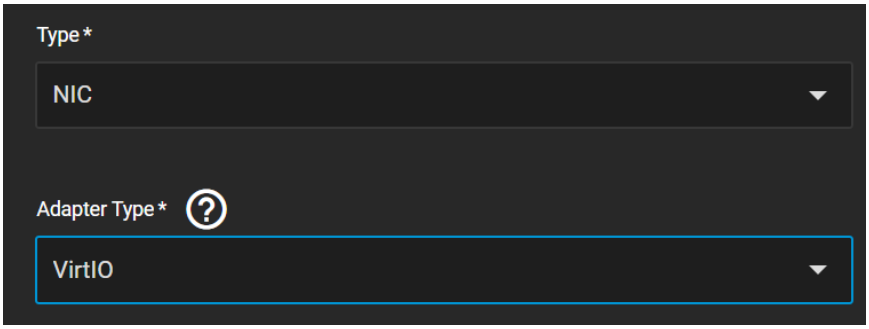
Before we can turn on our VM, we have two additional steps to take. Remember way back to the beginning where we created that bridge adapter?

We need to give our new VM a second network interface in that network so it can talk to other VMs on the SCALE host, and so that it can talk to our ISO SMB share we created.

Click on the Devices button

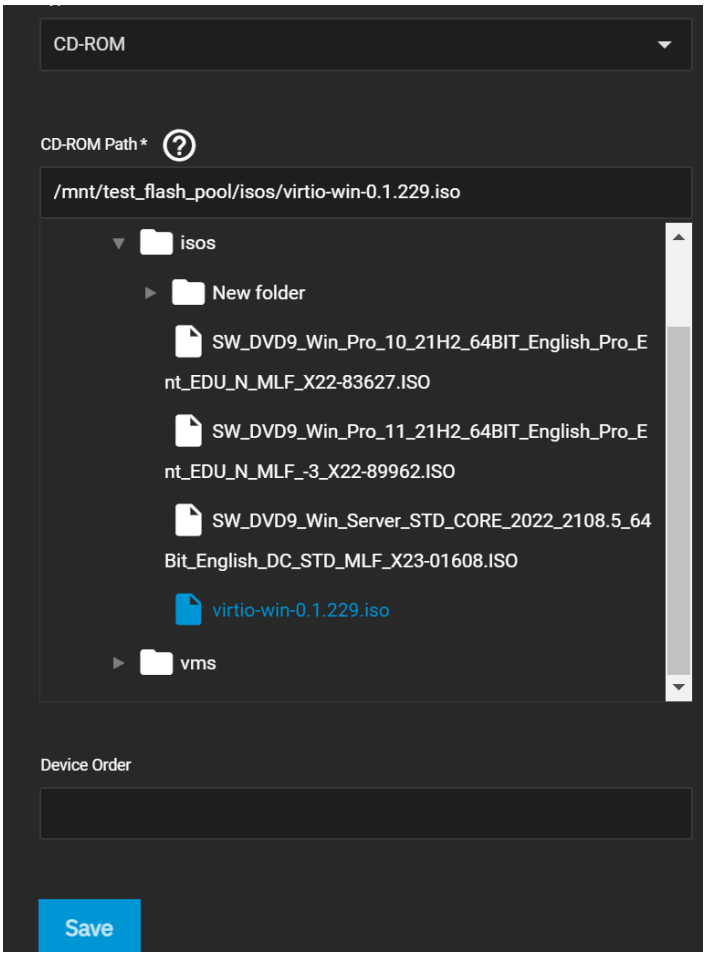


Press Add and change the type to NIC. Once again select the adapter type as VirtIO



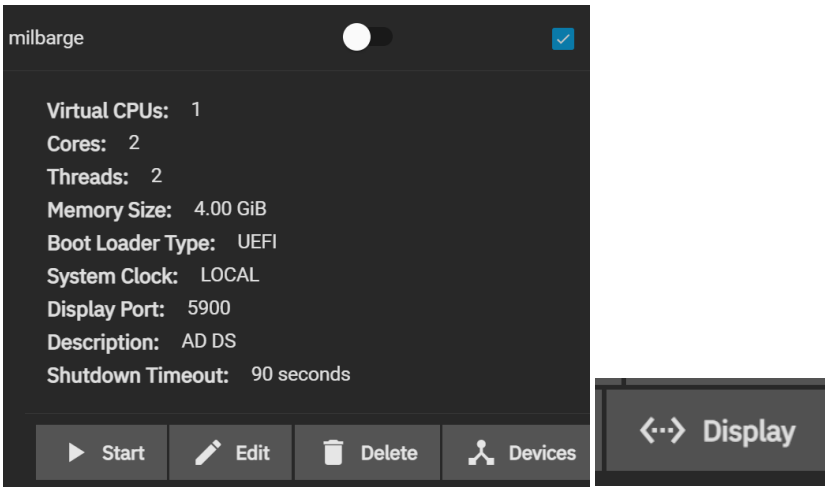
Attach it to the Bridge Network and press save

One final step. In order to install the OS, we're going to need to mount the driver CD for VirtIO as well. Let's add another CD drive. Press Add again, select the virtio CD and press save



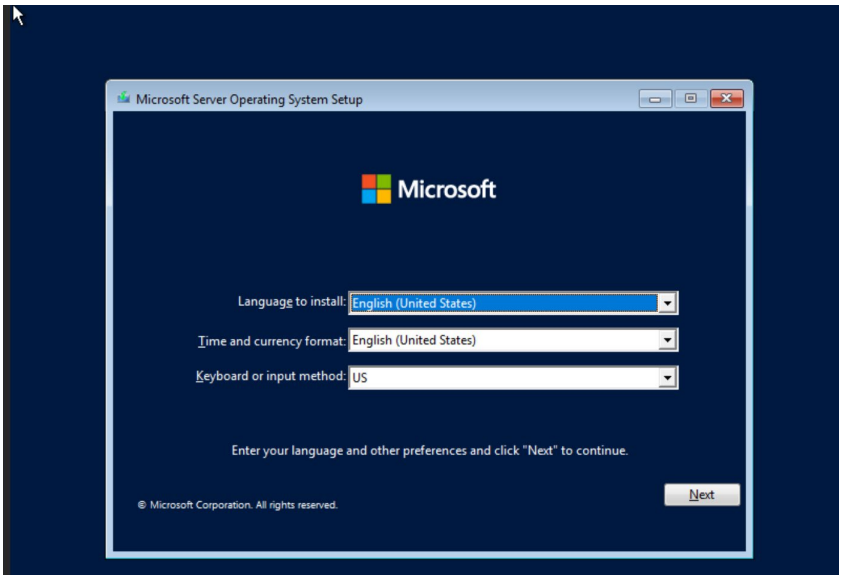
Now it’s time to Start Me Up!

Go back to your VM and press “Start” and the “Display” . Press a Key to launch the CD installer.



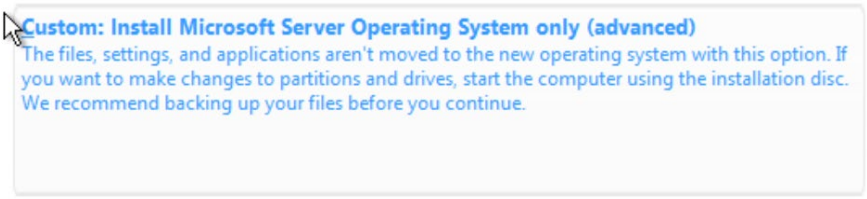
Installing Windows

Now we are ready to start installing Windows!



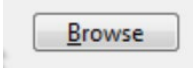
Press Next, Then Install Now, Select the version of Windows you have a license key for, and press Next again. Accept the EULA and press Next.

Select Custom:

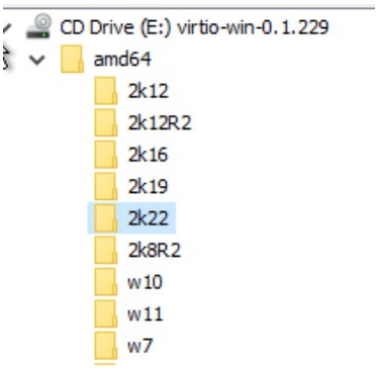


And then Select Load Driver:

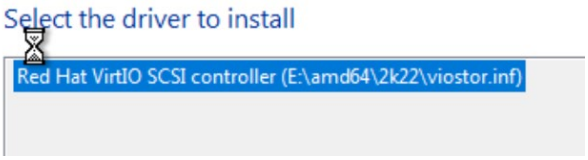
Press Browse:



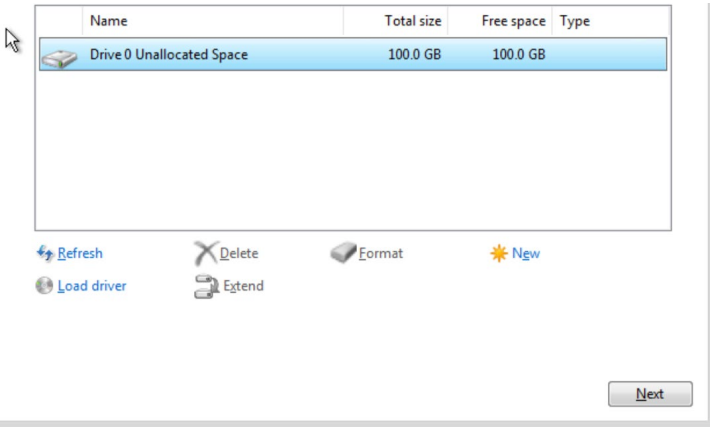
And select the folder for the VirtIO drivers that matches the version of Windows you are installing:



It will then start installing the driver:



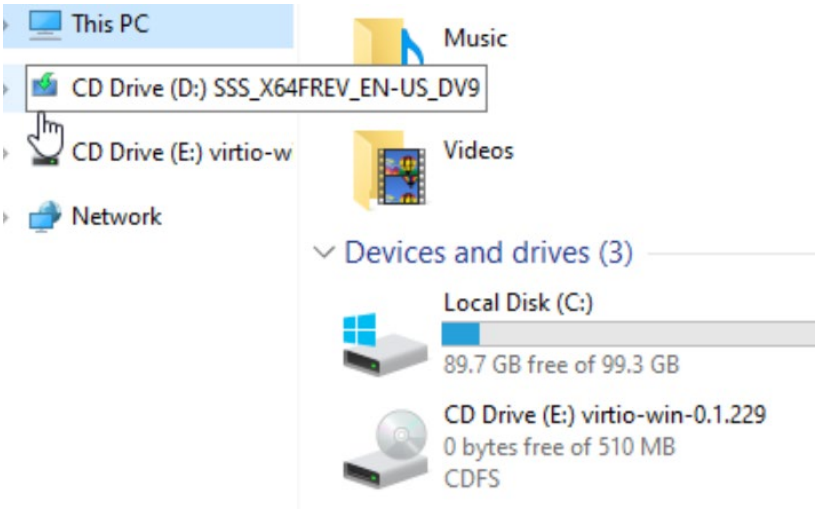
Now you can select Drive 0 and press Next



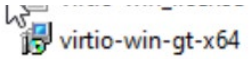
The OS will install and eventually reboot. Go grab some coffee or a snack.

When it comes back up, enter an administrator password and press Finish. Use the NoVNC toolbar to press Control/Alt/Delete, type your password and you are in!

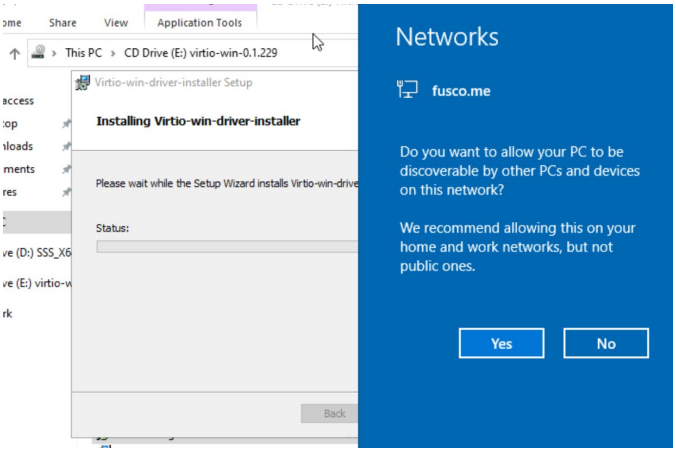
Now, lets get the rest of the VirtIO drivers installed so our network adapters work. Go to this PC and double click on the CD drive where VirtIO lives



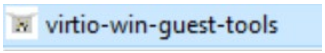
Click on the virtio-win-gt-x64 MSI



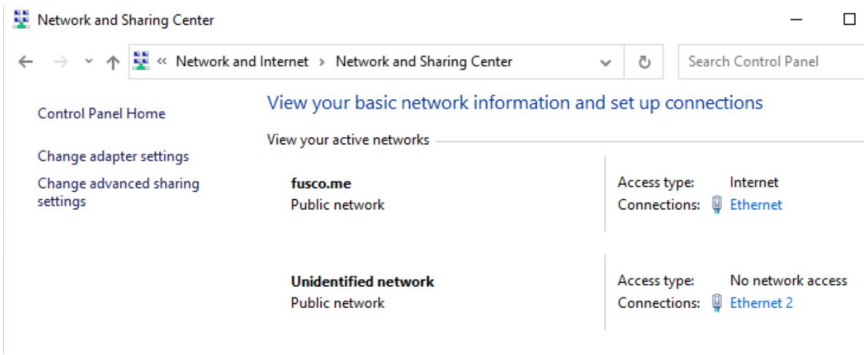
Next and yes your way through it, and all of the drivers will get installed and your network adapters should come up:



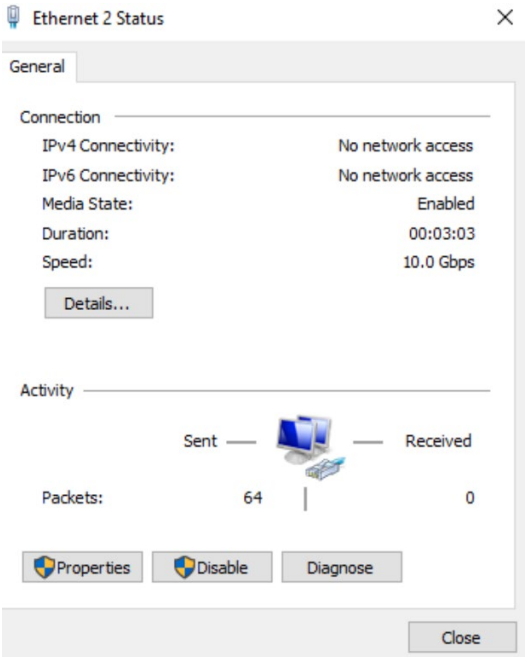
Now lets install the Guest Tools, so that TrueNAS can safely shut down the VMs when you reboot TrueNAS. Just like before, next and yes your way through.



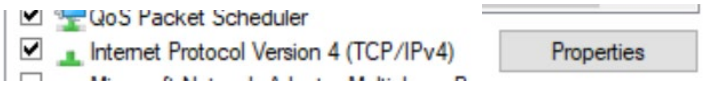
With that taken care of, we need to give our second network adapter an IP address.



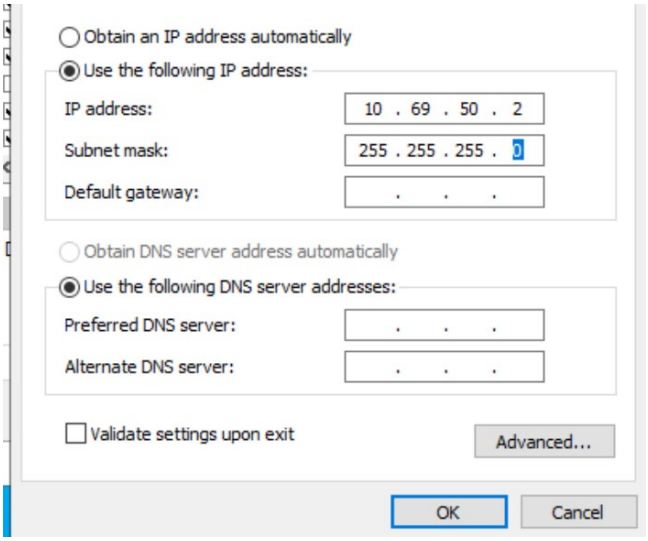
Click on Properties



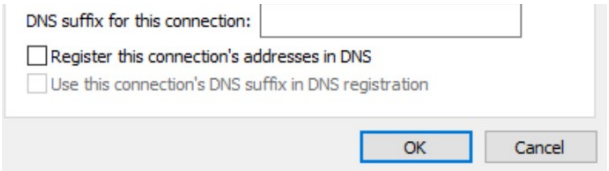
Then click on IP V4 and click Properties again



Give it an IP Address and subnet mask. My TrueNAS box is 10.69.50.1 so this server will be 10.69.50.2. Do not give it a gateway, and do not give it DNS. Then Press Advanced.



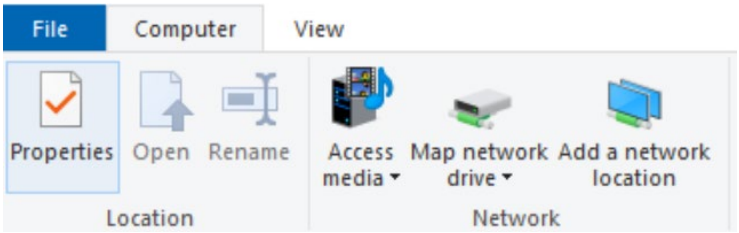
On the top click DNS and uncheck the box that says register this conections addresses in DNS. This will save you from stupid problems later. Press OK, OK, Close then Close again.



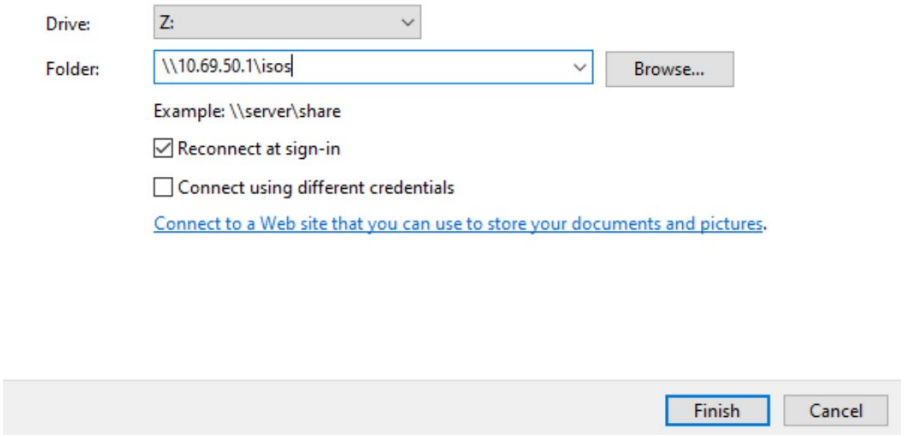
Lets open up a command prompt and make sure we can talk to TrueNAS SCALE:

```
C:\Users\Administrator>ping 10.69.50.1

Pinging 10.69.50.1 with 32 bytes of data:
Reply from 10.69.50.1: bytes=32 time<1ms TTL=64
Reply from 10.69.50.1: bytes=32 time<1ms TTL=64
Reply from 10.69.50.1: bytes=32 time<1ms TTL=64
Reply from 10.69.50.1: bytes=32 time<1ms TTL=64

Ping statistics for 10.69.50.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

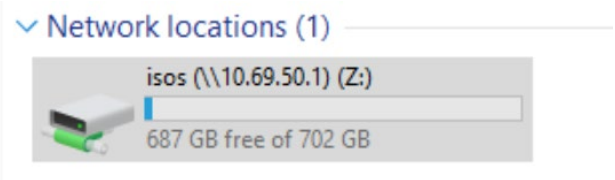
Now let's map our SMB share, go to this PC and then on the top select Map Network Drive:



Put the path to your SMB share on the IP address of your SCALE server inside of the Internal Network, tick the box that says connect using different credentials and press Finish. It will ask you to enter your username and password for the share on SCALE.

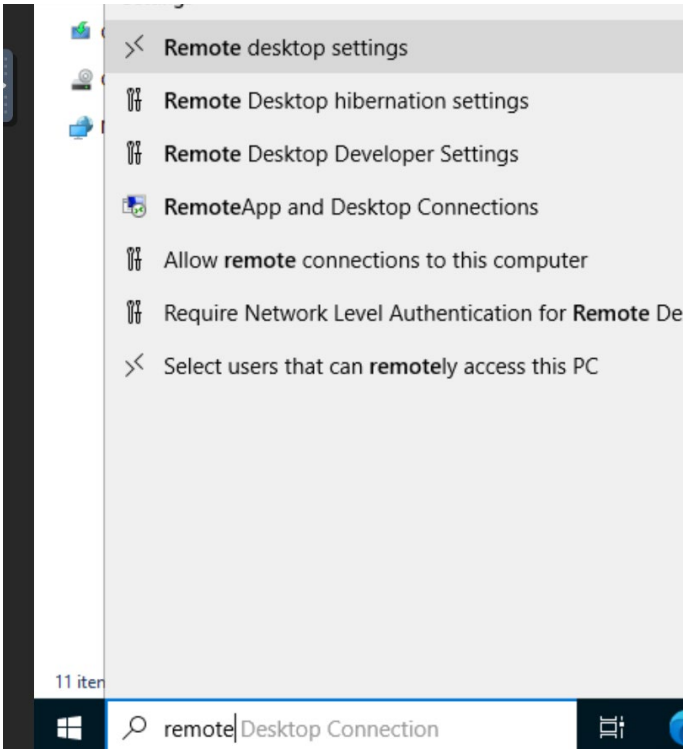


Now we are cooking with fire!

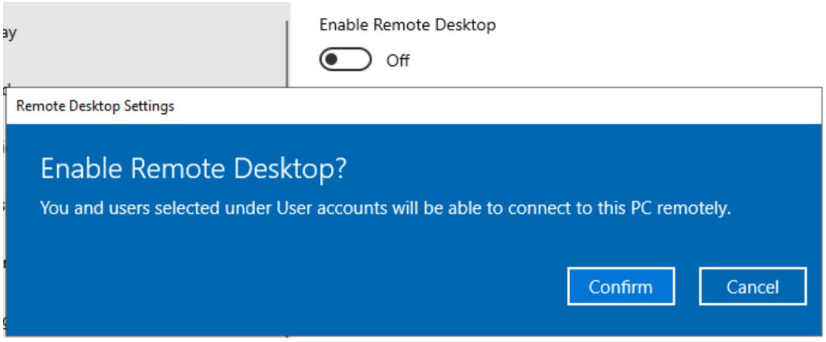


Now let's enable Micrsoft RDP so we can remove the VNC monitor which is only intended for initial setup and troubleshooting. Leaving VNC enabled is a security risk.

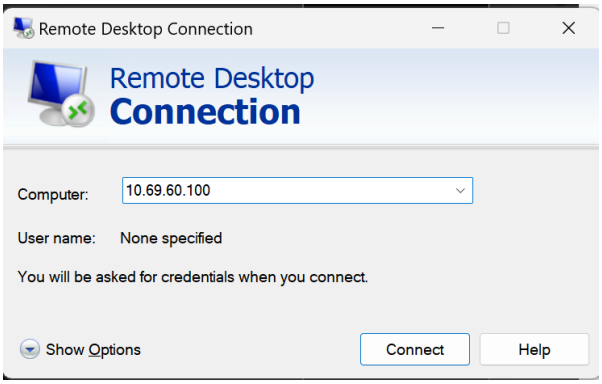
Go to start and type Remote Desktop Settings



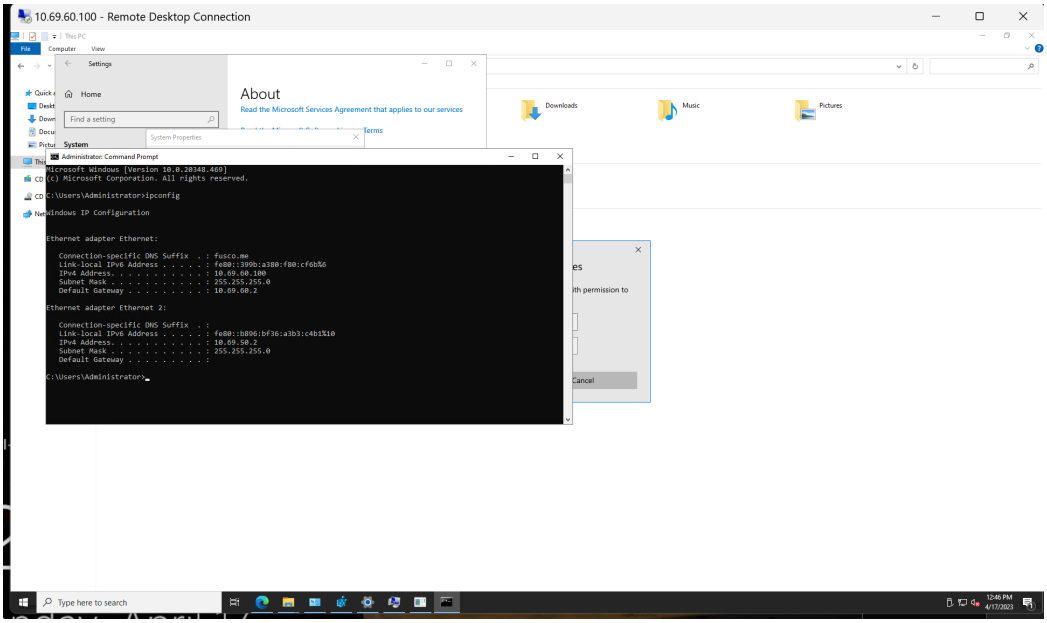
Tick the box to enable it and press confirm:



Verify you can access the RDP from another computer:

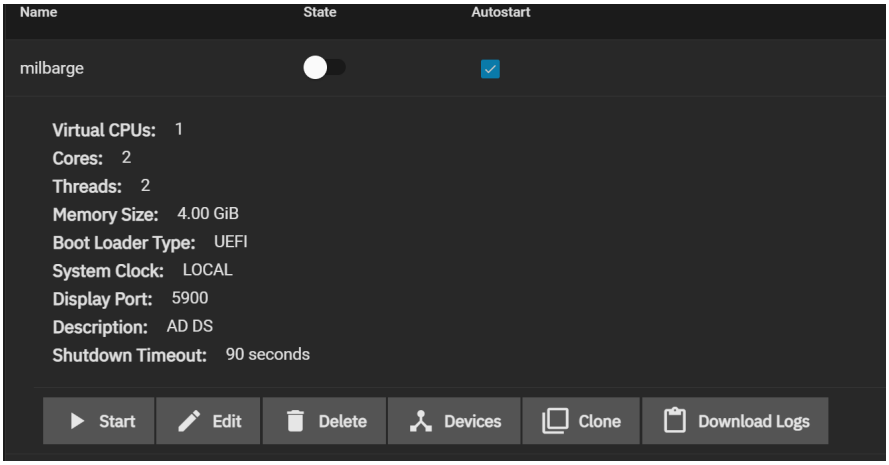


Huzah! It is working.

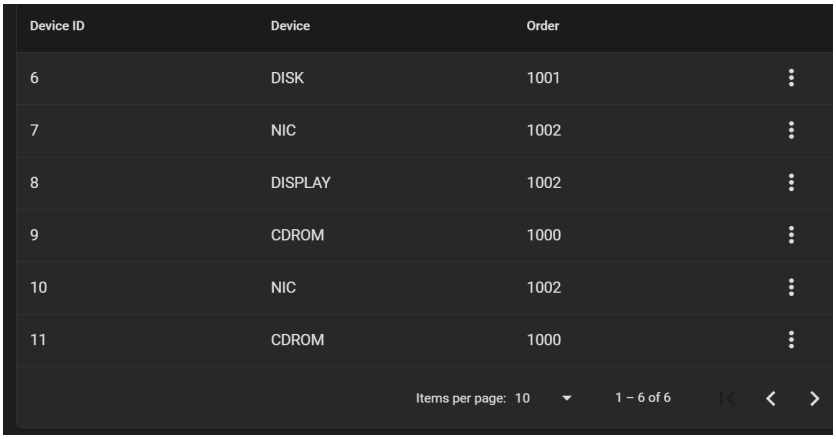


Shut the VM down and go back to your VM Page on SCALE.

Click on Devices:



Click on the Elipses(3 dots) and remove the two CDROM devices and the DISPLAY device



Turn your VM back on, make sure you can RDP back to it and you’re FINALLY done!

Conclusion

I hope this was helpful for someone in the future. Please let me know if I should update any sections or add/remove any information.