Nested Virtualization

https://www.virtualizationhowto.com

Nested virtualization is a great technology to make use of in your home lab environment. You can use it with VMware ESXi, Proxmox, and other hypervisors. By using nested virtualization, you have the ability to run a hypervisor inside another hypervisor. This creates a great learning environment with the capabilities afforded by modern hypervisors, like snapshots, cloning, and scripted installations.

Table of Contents

Nested ESXi install in Proxmox: Step-by-Step

December 21, 2023
Proxmox

Vmware esxi on proxmox

If you have a Proxmox VE server in your home lab or production environment and want to play around with VMware ESXi, you can easily do that with Proxmox nested virtualization. Let's look at the steps required for a nested ESXi server install in Proxmox.

Table of contents

- Frequently Asked Questions About Nested ESXi in Proxmox

Nested Virtualization in Proxmox

Nested virtualization in Proxmox VE is easy to set up and has real benefits in learning and setting up rather complex architectures without the physical hardware that would otherwise be needed to set them up physically.

Now, you can use something like VMware Workstation to easily nest ESXi. However, if you already have a dedicated Proxmox host, it is a better platform for a dedicated lab experience. There is always running it on VMware ESXi if you have a physical VMware host.

Proxmox nested virtualization allows exposing the CPU's hardware virtualization characteristics to a nested hypervisor. This process to expose hardware assisted virtualization to the guest ESXi VM is required so the nested hypervisor can run virtual machines.

Preparing your Proxmox VE host to enable nested virtualization for ESXi

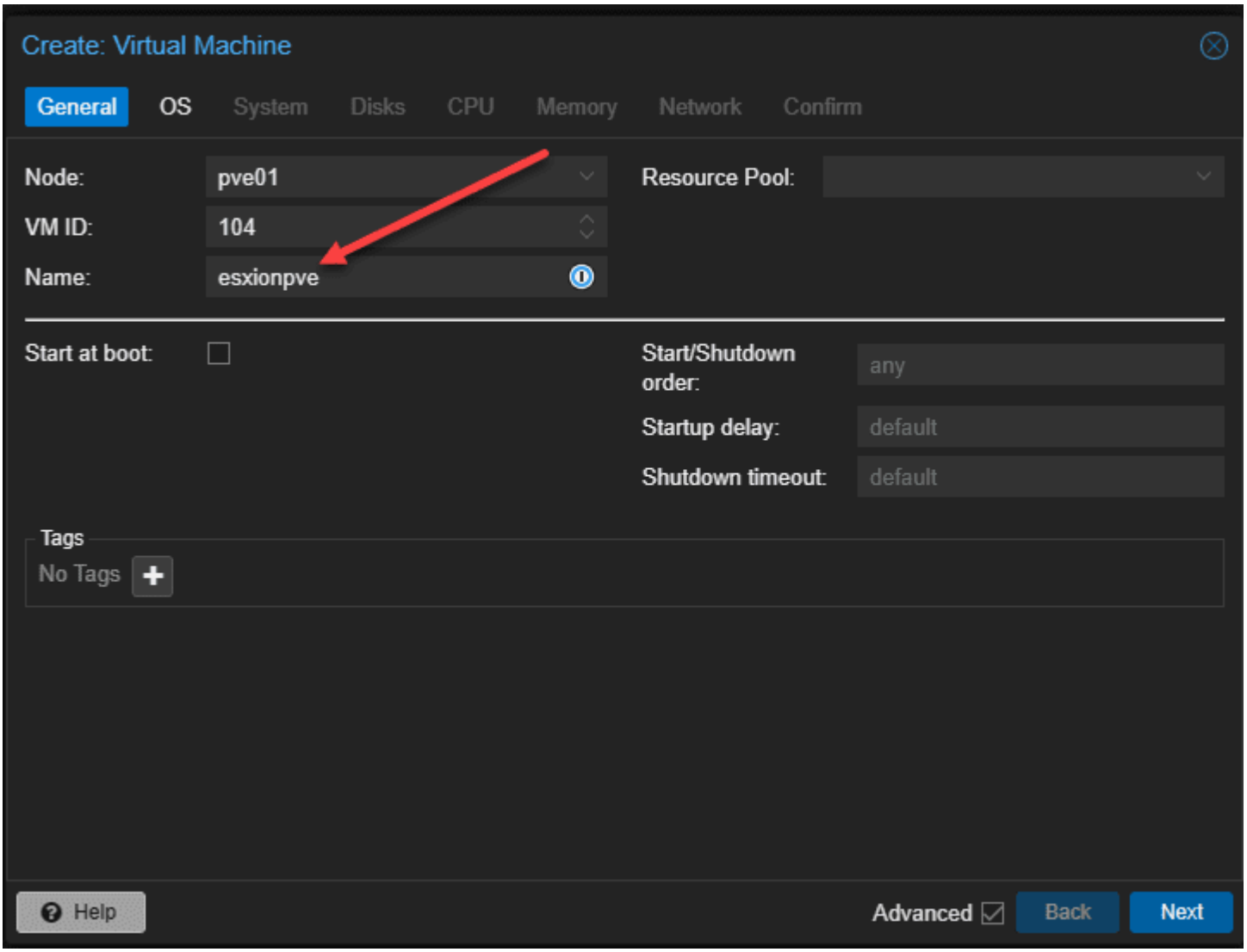
If you don't know how to configure Proxmox Nested Virtualization or enabling hardware assisted virtualization, you can see my recent guide to do that here: How to Enable Proxmox Nested Virtualization.

An overview of the few steps exist to enable nested virtualization for Proxmox and run a nested VM hypervisor are as follows:

- Make sure your CPU supports hardware-assisted virtualization
- Enable hardware-assisted virtualization if it isn't enabled already
- Enable nested virtualization on the nested ESXI installation VM

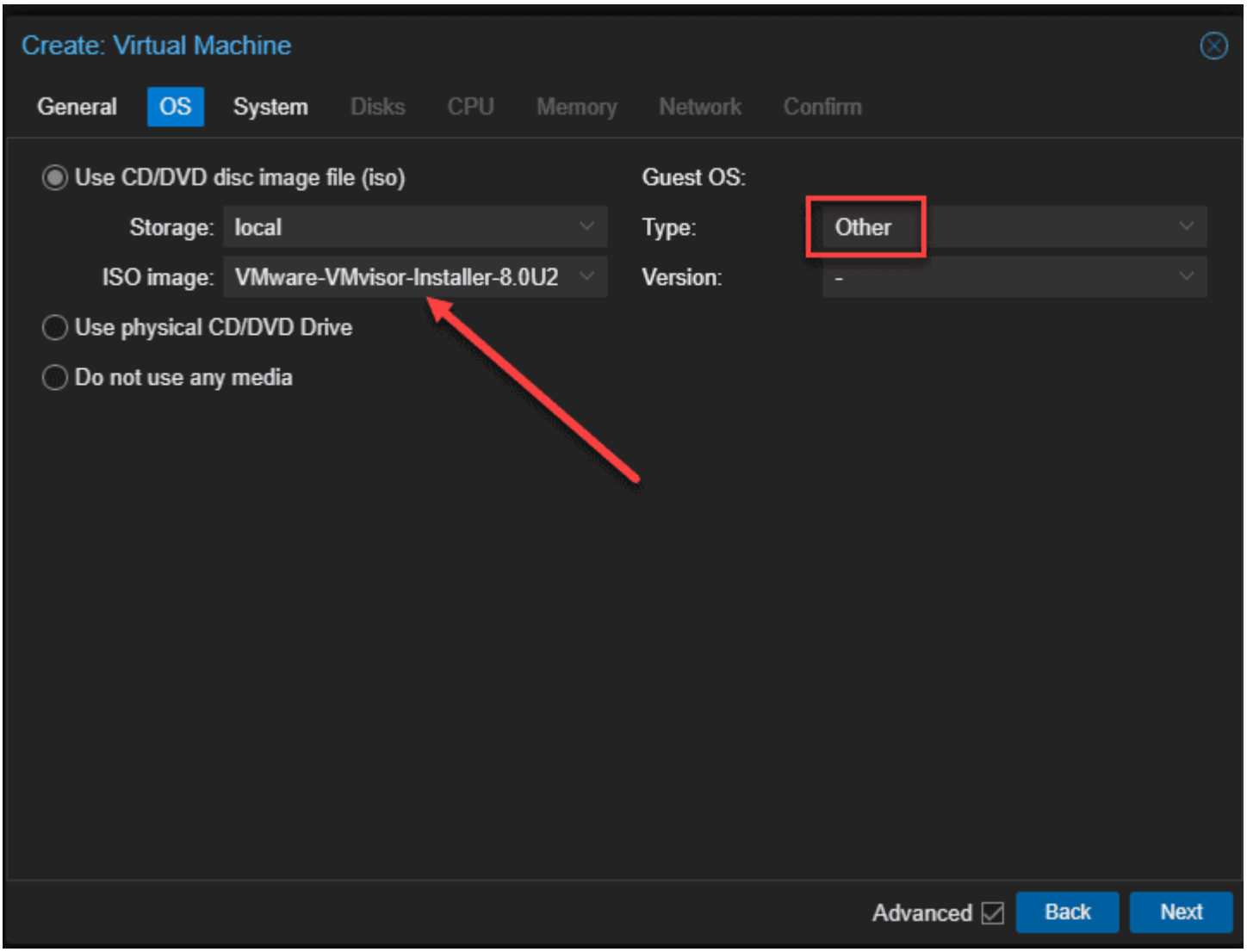
Creating the ESXi VM in Proxmox

VMware hypervisors are extremely popular in the enterprise. Let's look at the process to create the VMware ESXi VM in Proxmox. This is a normal creation process for the most part. I will show you guys one option I chose that didn't work, surprisingly when creating the VM running ESXi.



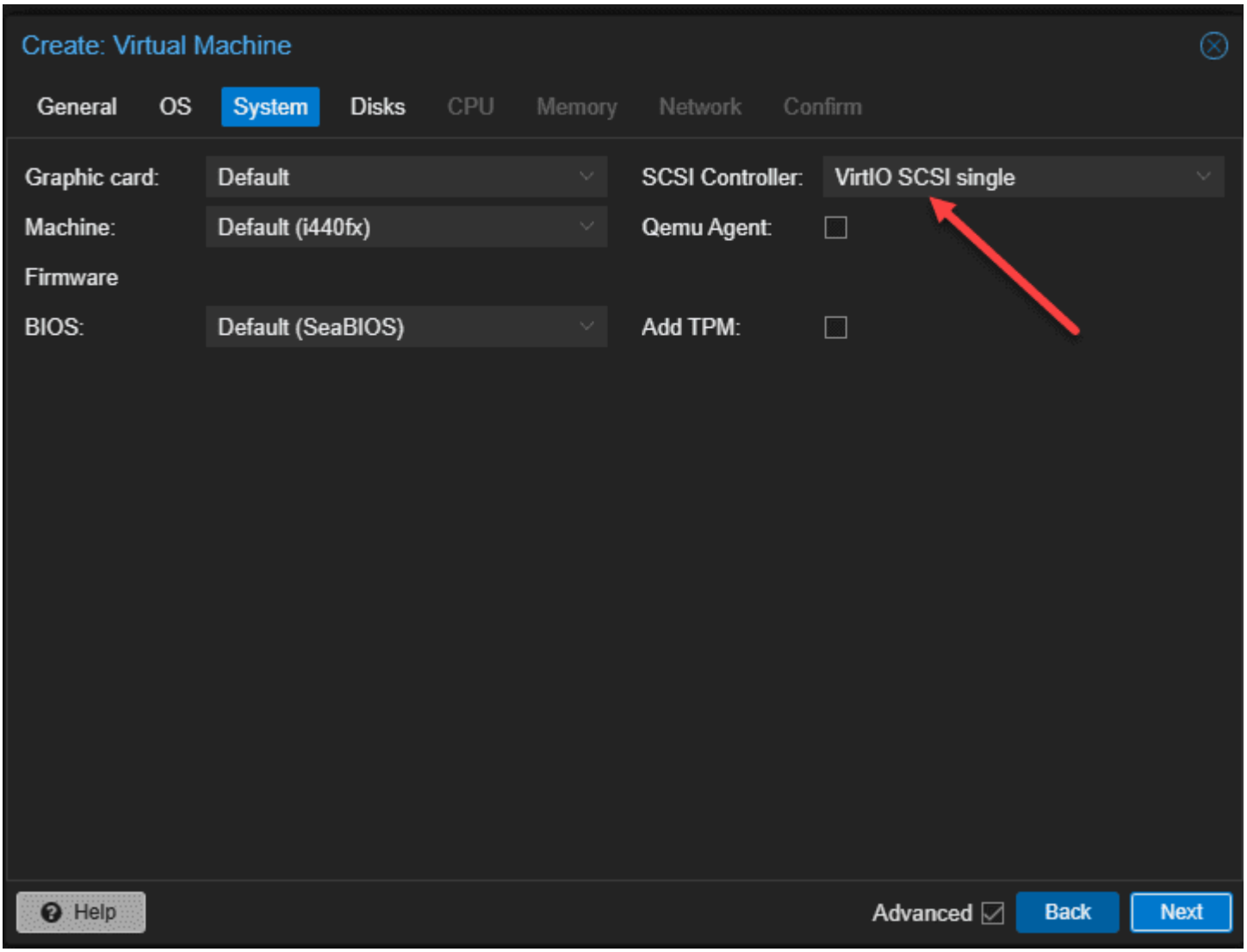
Beginning the create virtual machine wizard

Upload your VMware ESXi 8.0 U2 or other ESXi ISO to your Proxmox server and select this in the wizard. On the type, choose **Other** for the guest operating system.



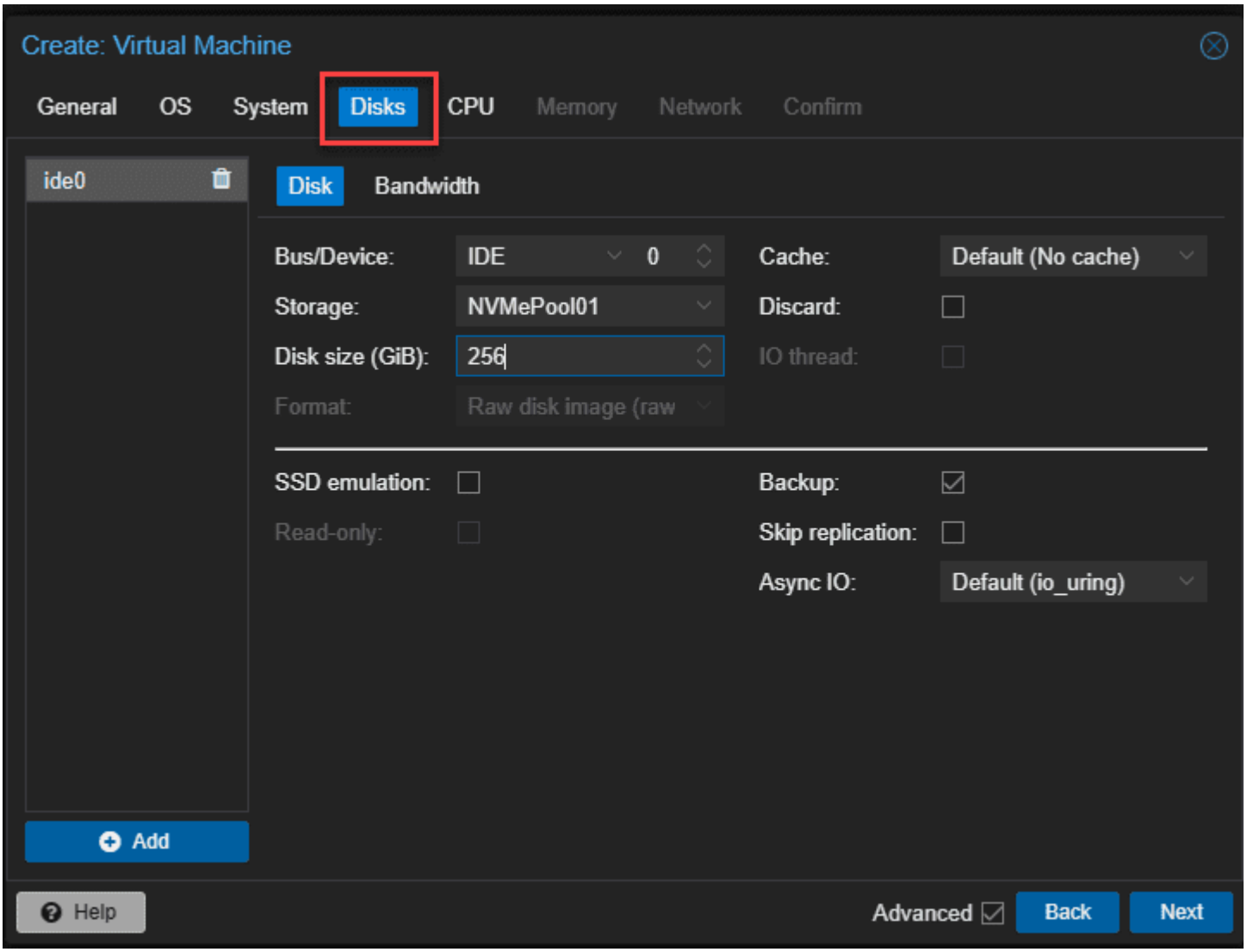
Select your esxi iso image under the os tab

Here I left **VirtIO SCSI single** selected for SCSI controller.



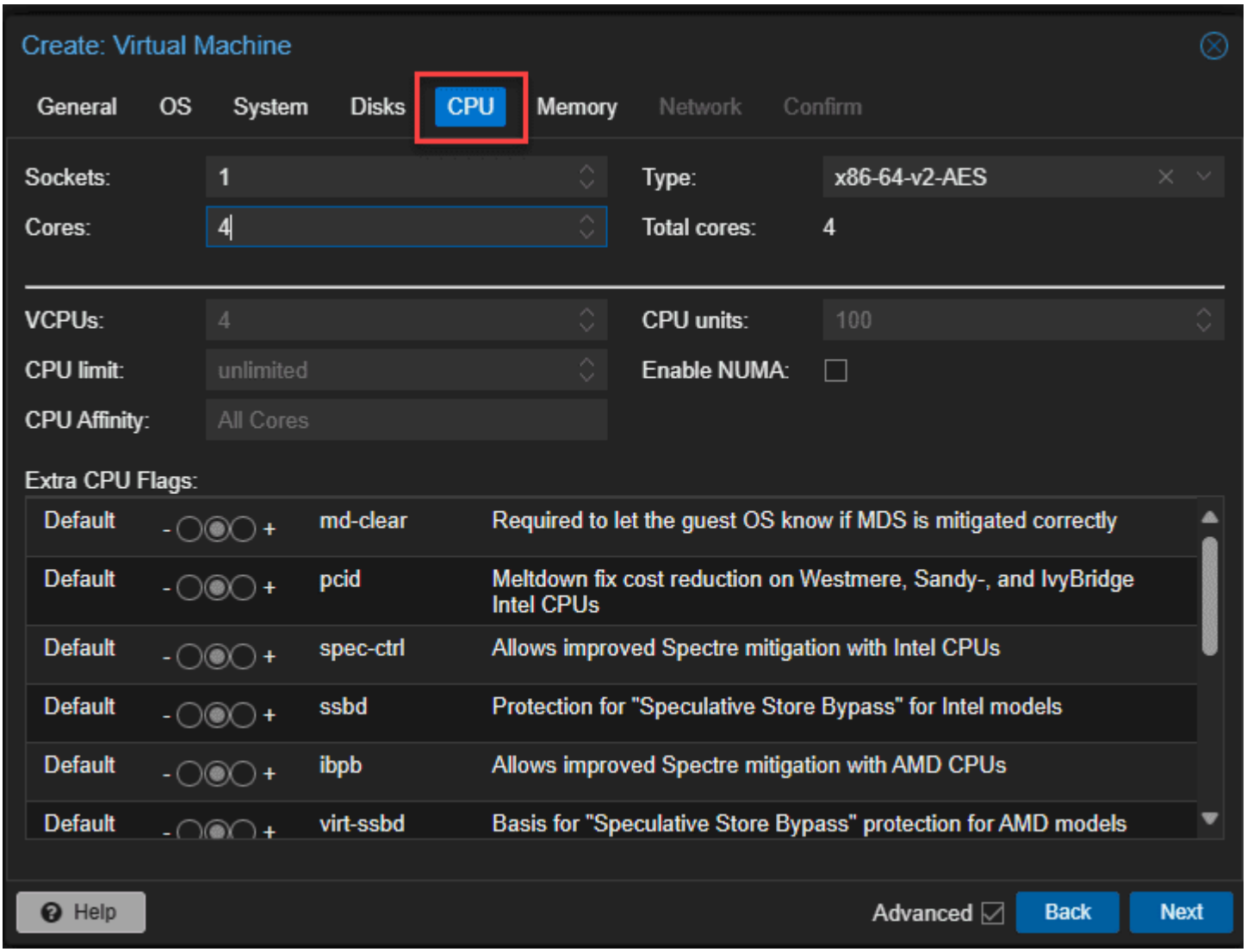
Leaving the defaults under system for esxi nested

On the **Disks** screen, configure the disk size you want and also the Storage location for your VM files and hit **Next**.



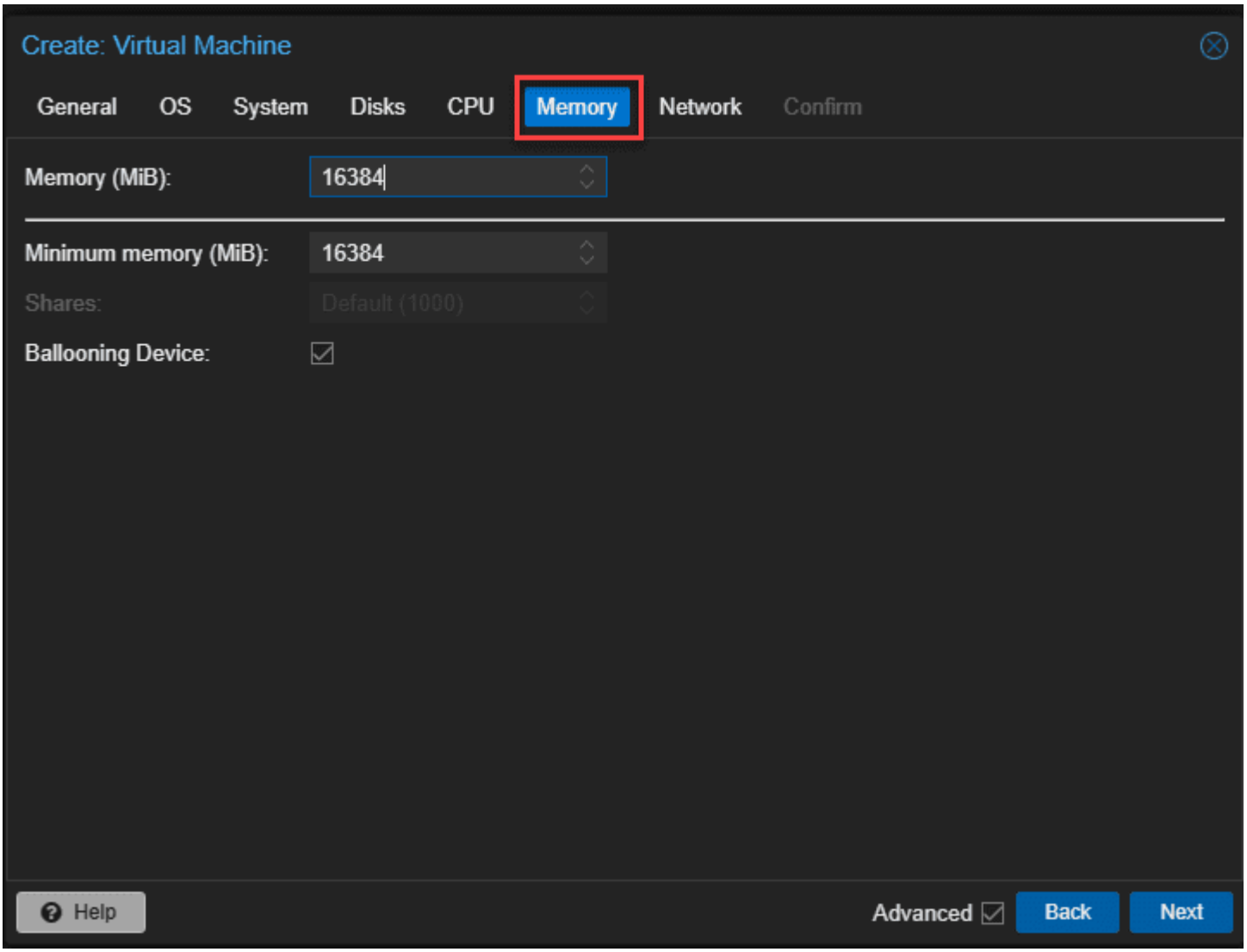
Setting up the storage for the esxi vm

Choose your CPU options.



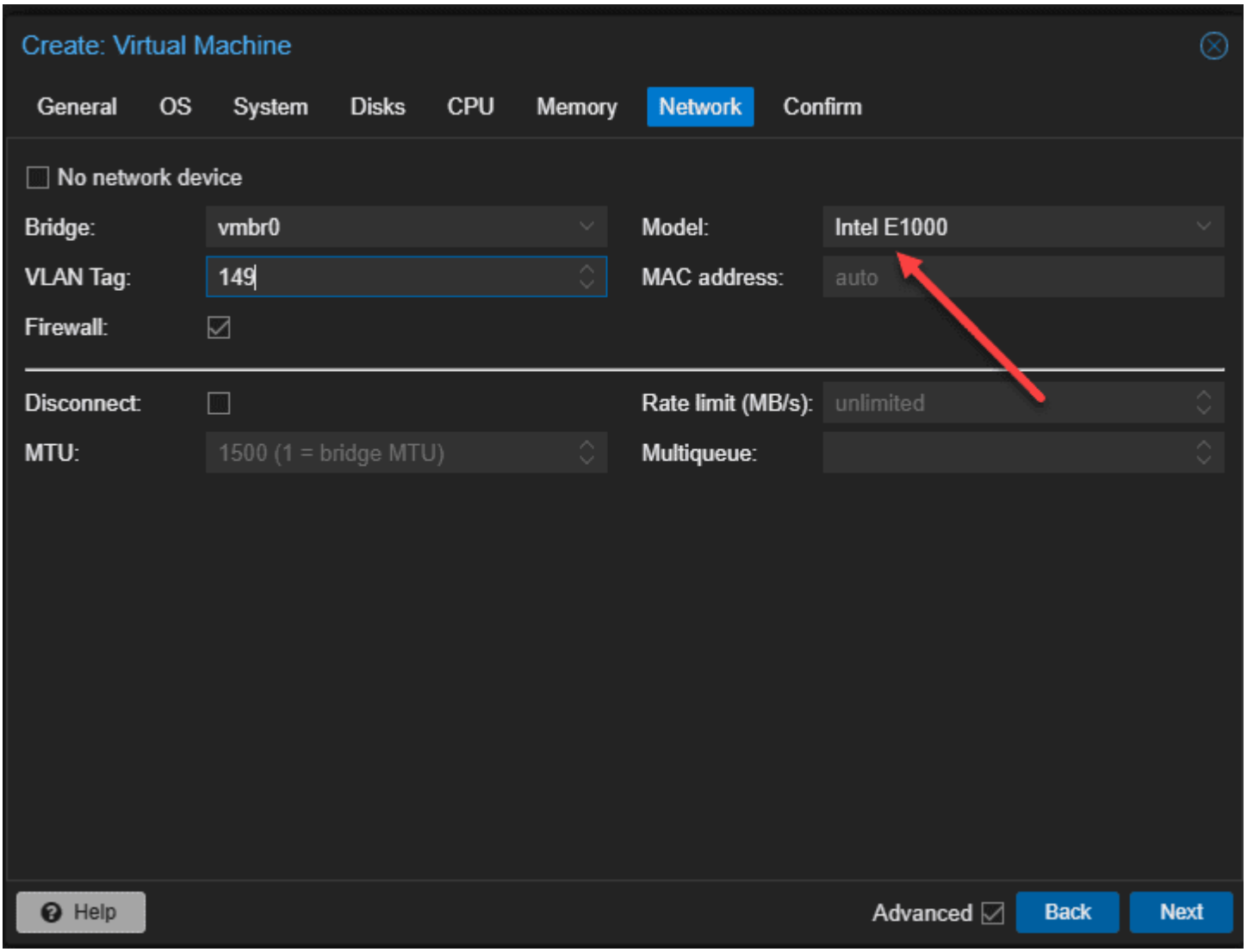
Cpu settings for the esxi nested vm

Configure your memory.



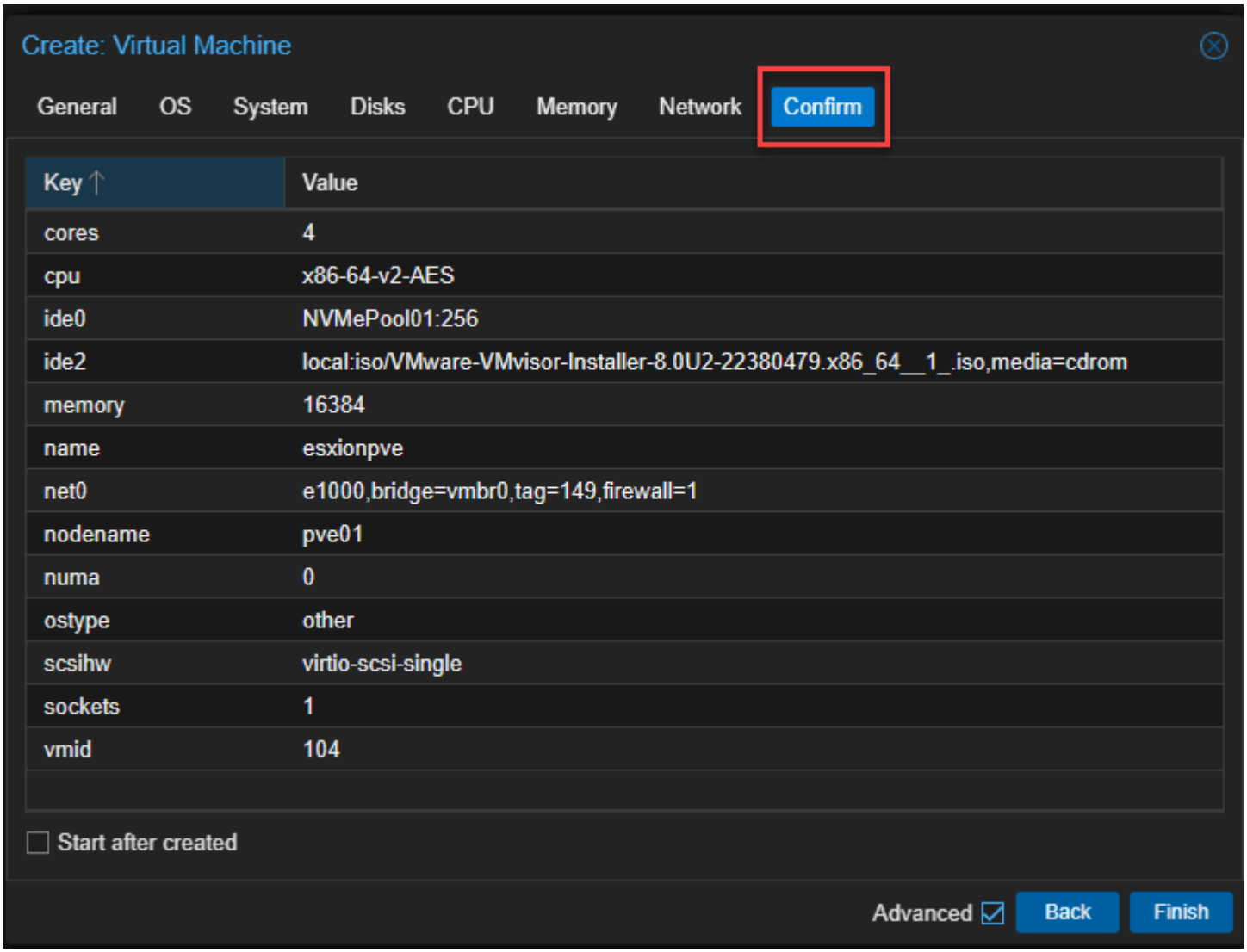
Memory configuration

Ok, so this is the step that surprised me a bit. I here selected **Intel E1000** which is a standard Intel driver. But I will show you what happens during the install.



Setting the network adapter to e1000

Confirm your configuration and click **Finish**.

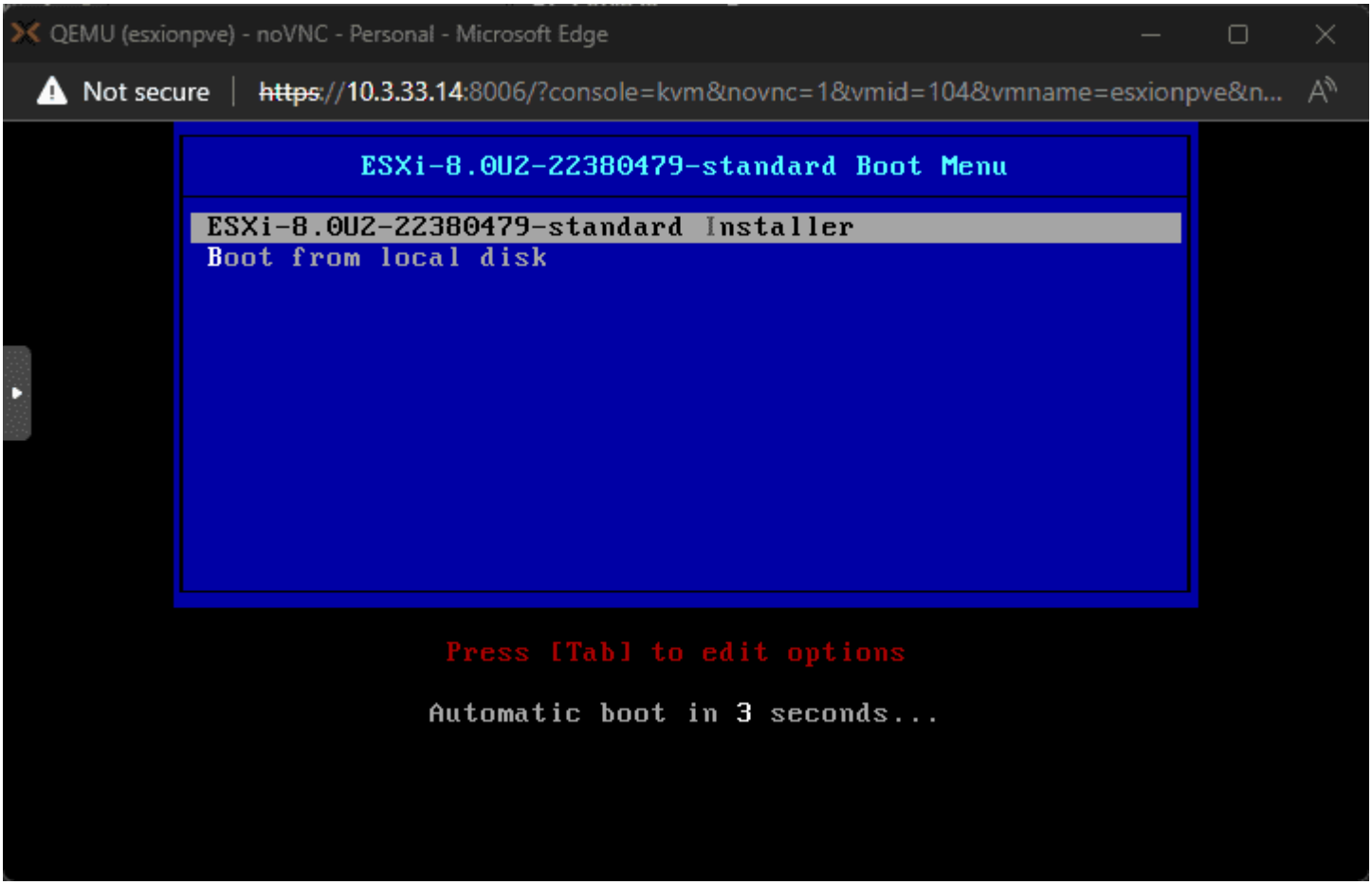


Confirm the installation options

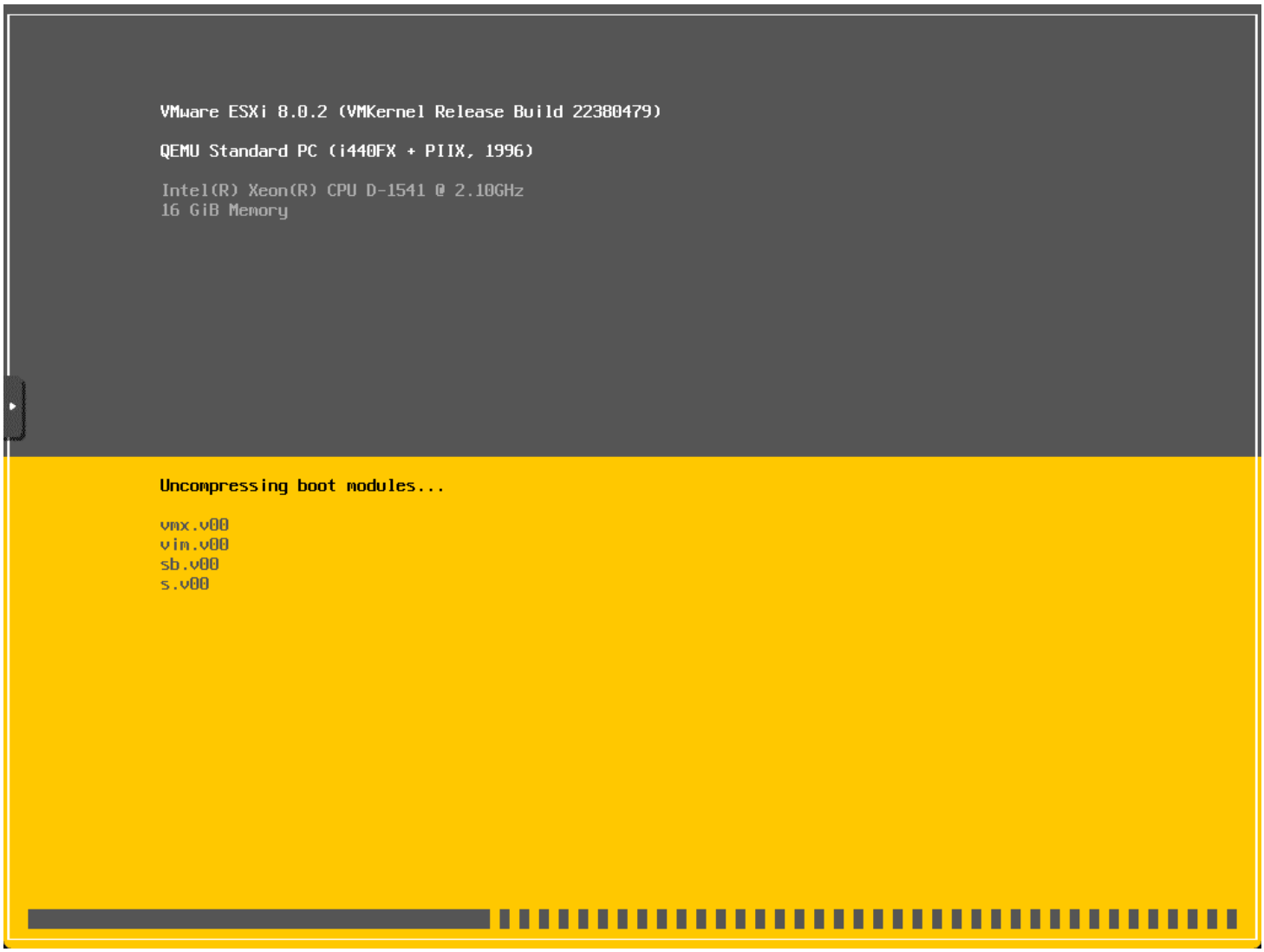
Step-by-Step Installation of Nested ESXi

Let's look at how to install ESXi in Proxmox after we have created the Proxmox virtual machine to house the nested virtual machine install.

Below is booting the VMware VM guest OS in Proxmox.



Beginning the esxi 8.0 u2 installation

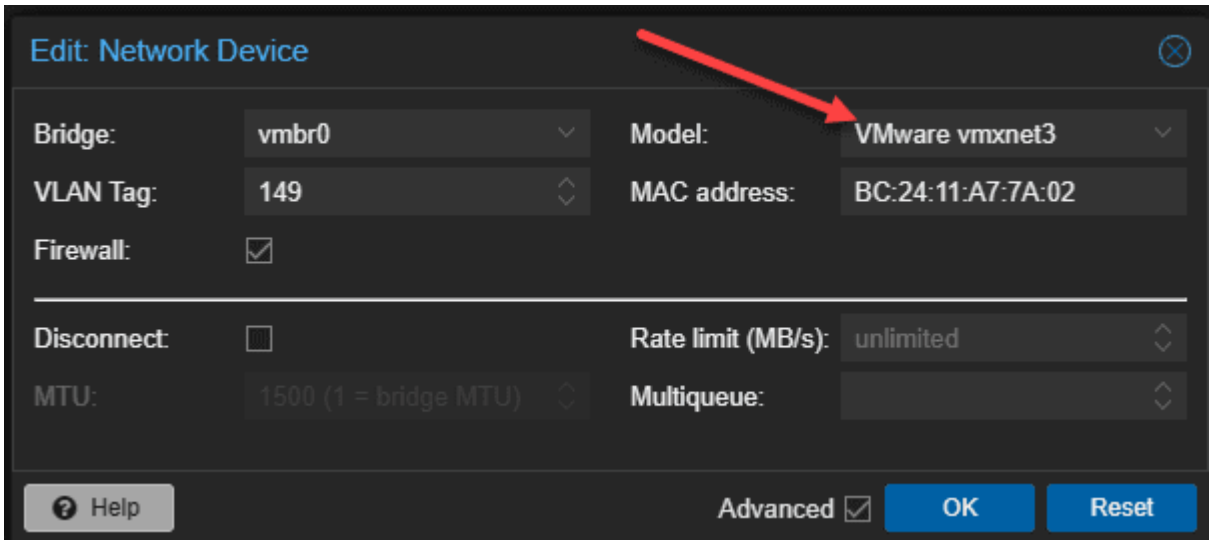


The esxi nested vm boots into the installation

OK, so I told you there was something unexpected happen with the Intel E1000 driver. It didn't detect the network adapter in ESXi.

No network card detected in esxi

I powered the ESXi VM down and went back and selected **VMware vmxnet3** adapter for the model.



Changing the network adapter model to vmware vmxnet3

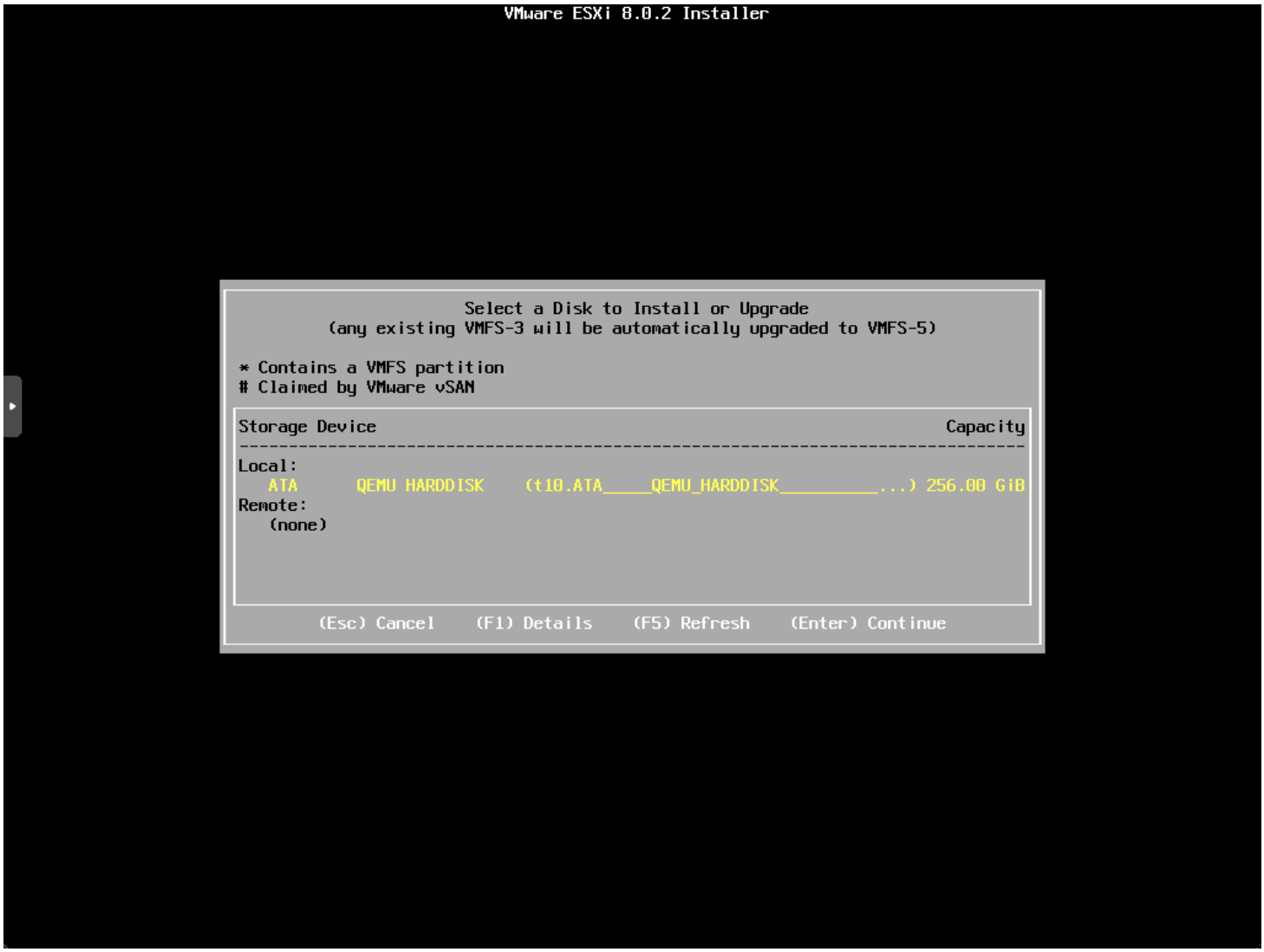
Now, the network adapter was recognized and the installation proceeded.

The installation of nested esxi continues

Now for the standard screens, but we will show them anyway. Accept the EULA.

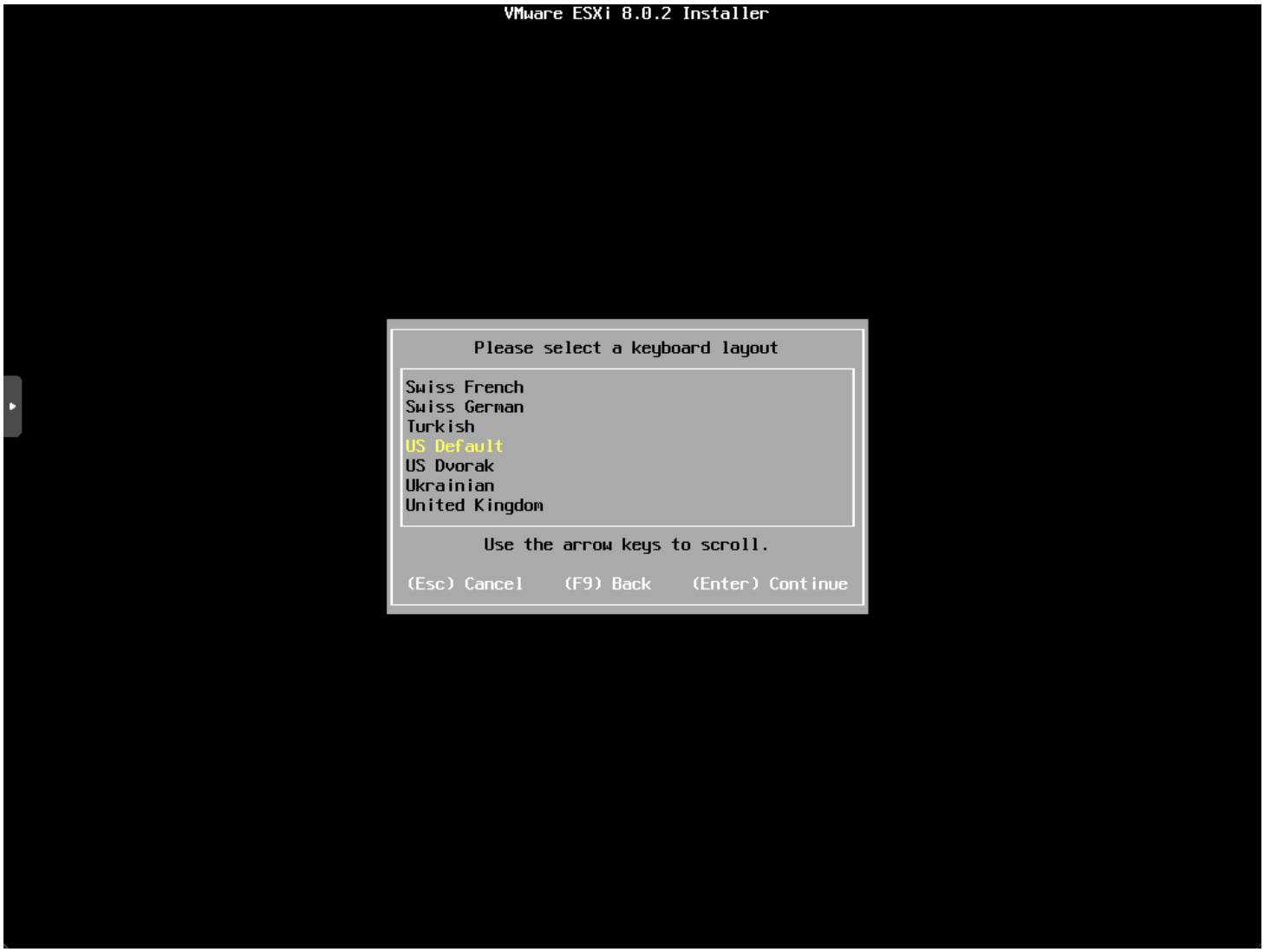
Accept the eula 1

Select the target storage for the installation.



Select the installation target storage for nested esxi

Select the location for the keyboard layout.



Select the esxi keyboard layout

Enter and confirm your password.

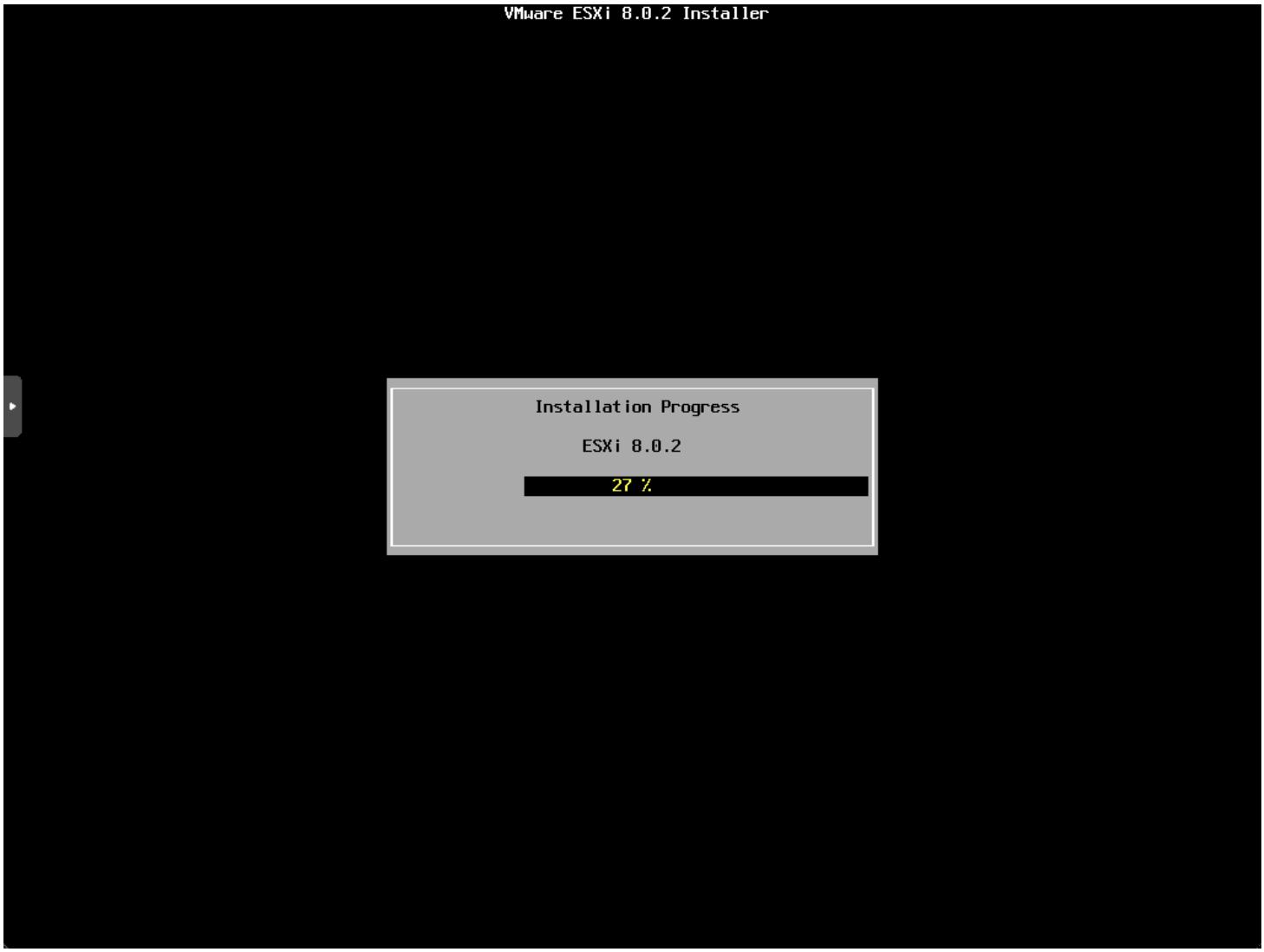
Configure the root password for esxi

I am running on an older Xeon D processor so we see the alert about an outdated processor that may not be supported in future releases. You will see the same error on bare metal.

Warning about older cpu support in esxi 8.0 update 2

Confirm the installation of esxi and repartitioning

The installation begins.

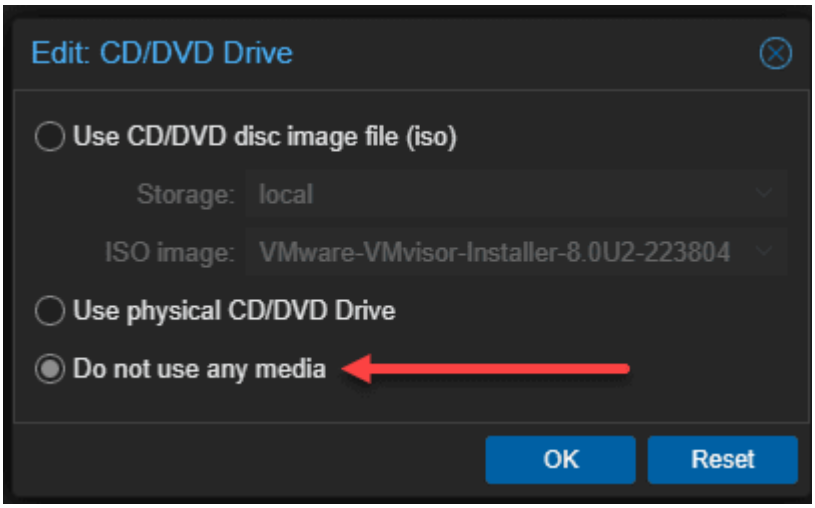


Esxi installation progress begins

Finally, we are prompted to remove the installation media and reboot.

Installation finished remove installation media

Hopping back over to Proxmox, I remove the ESXi ISO before rebooting.

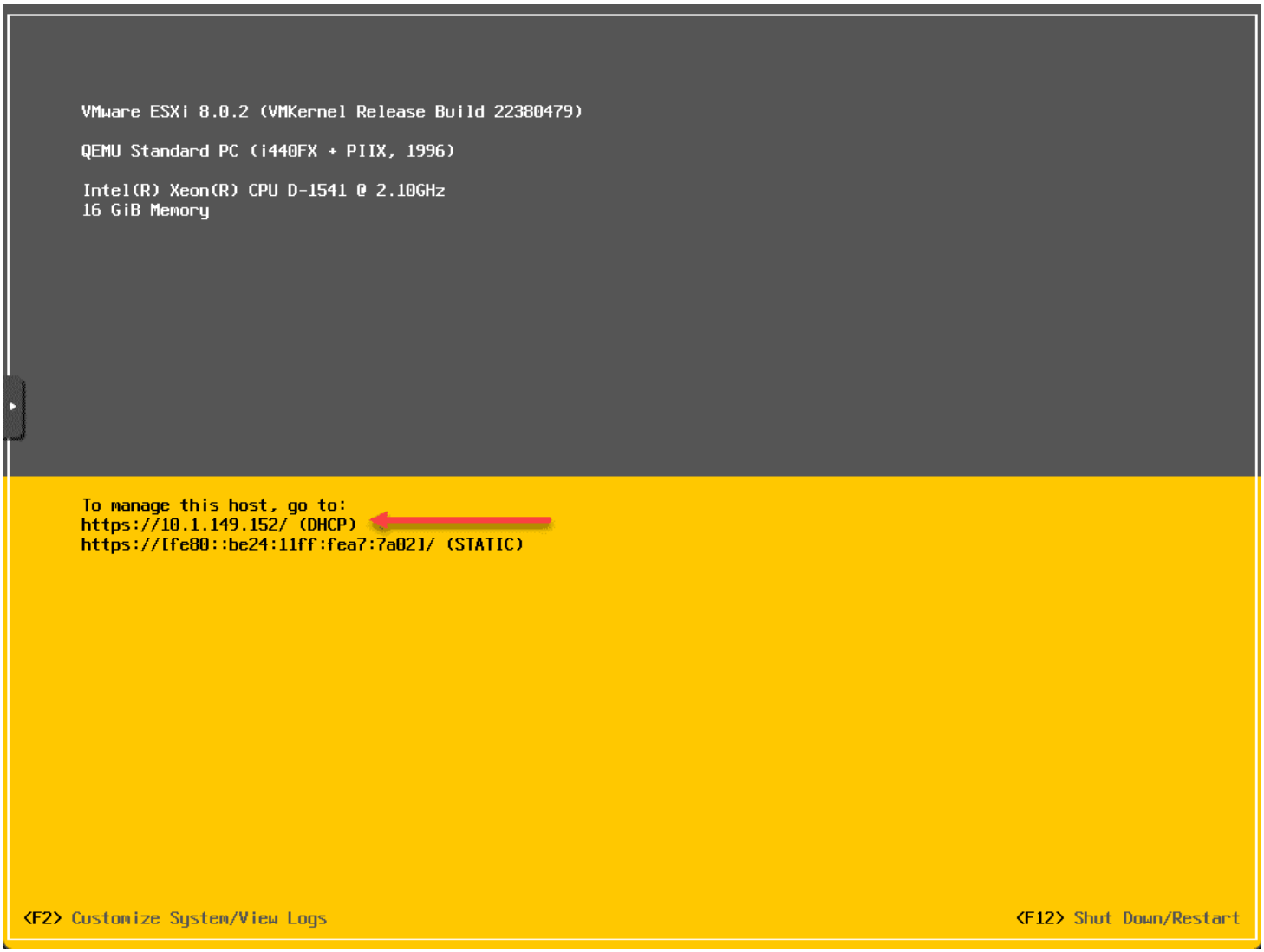


Removing the iso from the esxi vm in proxmox

After initiating a reboot.

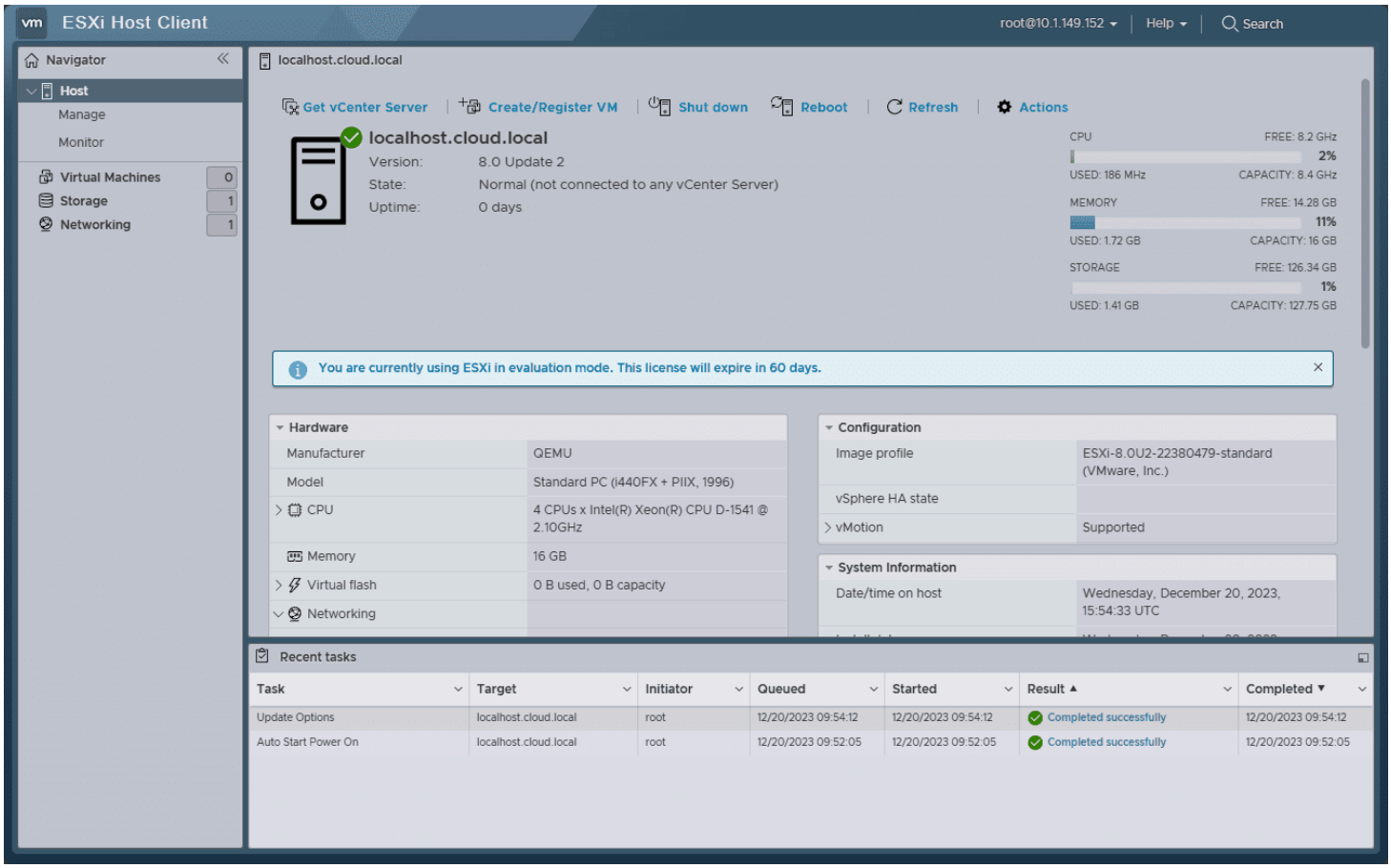
Rebooting the esxi installation

After the nested ESXi installation boots, we see it has correctly pulled an IP address from DHCP so the network adapter is working as expected.



Vmware esxi vm in proxmox boots and it correctly pulls a dhcp address

Below, I logged into the VMware host client to manage the ESXi host running in Proxmox.



Logged into the esxi host client

Managing Virtual Machines in a Nested Setup

The cool thing about working with ESXi that is nested in a Proxmox VM is that, for the most part, you won't notice much difference if you are used to accessing the ESXi host client or adding the ESXi host to the vCenter Server and managing it with vCenter.

Using advanced features in nested VMs

The great thing about running ESXi as a nested hypervisor, is you won't see any difference in the advanced features for nested VMs. You will still be able to do things like installing VMware Tools in Linux and your Windows Server operating system instances.

If you are configuring a cluster of ESXi hosts with vCenter, you can utilize features like vMotion and DRS within a nested VMware vSphere cluster.

Troubleshooting Common Issues in Nested Environments

Running nested ESXi in Proxmox can be a bit of a mind-bender on the networking side. However, this is not unique to Proxmox, as running nested ESXi on a physical ESXi host can be the same challenge.

First, though, you need to understand Proxmox VLANs. I just covered this recently as well. So, check out my post on Proxmox VLANs to first understand how to configure VLANs in Proxmox.

Just remember, on the nested VMware ESXi side, you can't tag VLANs on your port groups as this will lead to "double tagging". They will instead assume the tag from the Proxmox side.

What I like to do is set up the Proxmox Linux Bridge as a trunk bridge, which is the default configuration when you make it VLAN aware. Then, you can change the tag on your network adapter configured for your VMware ESXi VM to tag the traffic from the ESXi VM.

Frequently Asked Questions About Nested ESXi in Proxmox

How Does Nested ESXi Differ from Regular Virtualization in Proxmox?

Nested ESXi in Proxmox takes virtualization a step further by running a virtual machine (VM) within another VM. In nested setups, ESXi acts as a guest hypervisor within the VM to create and manage additional VMs in this second layer of virtualization.

Can I Run VMware Tools in a Nested ESXi VM?

Yes, VMware Tools can be installed and run within a VM running on nested ESXi in a Proxmox environment. This installation enhances the functionality and performance of the nested VMs. It provides better hardware compatibility and improved management capabilities.

What Are the Key Considerations for VM Hardware Settings in Nested Virtualization?

When configuring VM hardware in a nested virtualization setup, it's important to allocate sufficient resources, such as CPU and memory, to ensure smooth operation. Additionally, you should enable promiscuous mode in the virtual switch settings to allow communication between nested VMs.

Is Nested ESXi Suitable for Production Environments?

Not really in most scenarios. You definitely won't be supported by VMware in a nested environment and likely not Proxmox either. It is best to keep nested environments in their proper place, for learning and labbing and testing out configurations without the physical hardware to install on bare metal.

How Can I Optimize the Performance of Nested VMs in Proxmox?

Give attention to resource allocation, enabling hardware-assisted virtualization, and configuring network settings properly. Monitor your Proxmox VE host and nested ESXi VMs to make sure there are no performance issues.

Can Windows Server Be Used Effectively in a Nested ESXi Setup?

Windows Server can be run as a guest operating system in a nested ESXi VM. This setup allows for testing and development of Windows-based applications in a controlled, virtualized environment, leveraging the capabilities of both Proxmox and ESXi.

Are There Specific Network Configurations Required for Nested ESXi in Proxmox?

Nested ESXi in Proxmox requires specific network configurations, including setting up virtual switches and enabling promiscuous mode to allow proper network traffic flow between nested VMs. Proper configuration ensures seamless connectivity and communication within the nested environment.

What Are the Benefits of Using Intel VT-x in Nested Virtualization?

Using Intel VT-x in nested virtualization enhances the performance of nested virtualization. This technology enables more efficient emulation of hardware features. Really, you don't want to use nested virtualization without it.

Wrapping up

Hopefully, this blog post has been a help to any who are running Proxmox as your hypervisor running your home lab environments. It is easy to get a virtual machine running with VMware ESXi in a Proxmox nested environment. Keep in mind the need to use the VMware vmxnet3 adapter and the note on Proxmox VLAN tagging. If you are running guest VMs in your ESXi VM, you will also need to keep in mind the need to enable promiscuous mode for your Proxmox bridge.

Nested Proxmox VMware installation in ESXi

January 13, 2022
Proxmox

Booting the Proxmos 7.1 VE installer

In working with clients and different environments, you will definitely see many different hypervisors used across the landscape of enterprise organizations. While I recommend VMware vSphere for business-critical enterprise workloads to customers, there are use cases where I see other hypervisors used. Proxmox is a really great open-source, free hypervisor available for use and is even developed for use in enterprise applications. I also know of many in the community running Proxmox in their home lab environment. If you are like me and like to play around with technology, hypervisors, and other

cool geeky stuff, I find I load a lot of different solutions in the lab. Let's take a look at nested Proxmox VMware installation in ESXi and see how you can easily spin up a Proxmox host in a vSphere VM.

What is Proxmox?

Proxmox is easily administered using a rich, fully-featured web interface that actually looks and feels nice. While it is not in my opinion where the vSphere client is in look and feel, it is quite nice and does the job needed to administer the Proxmox environment.

Proxmox VE is an open-source hypervisor platform for enterprise virtualization. It provides many features needed to run production workloads, including virtual machines, containers, software-defined storage, networking, clustering, and other capabilities out-of-the-box. It is based on Linux, so you get the pure Linux experience for virtualization, containers, and other facets. Note some of the benefits:

- Open-source software
- No vendor lock-in
- Linux kernel
- Fast and easy installation
- Easy-to-use with the intuitive web-based management interface
- Low administration costs and simple deployment
- Huge active community

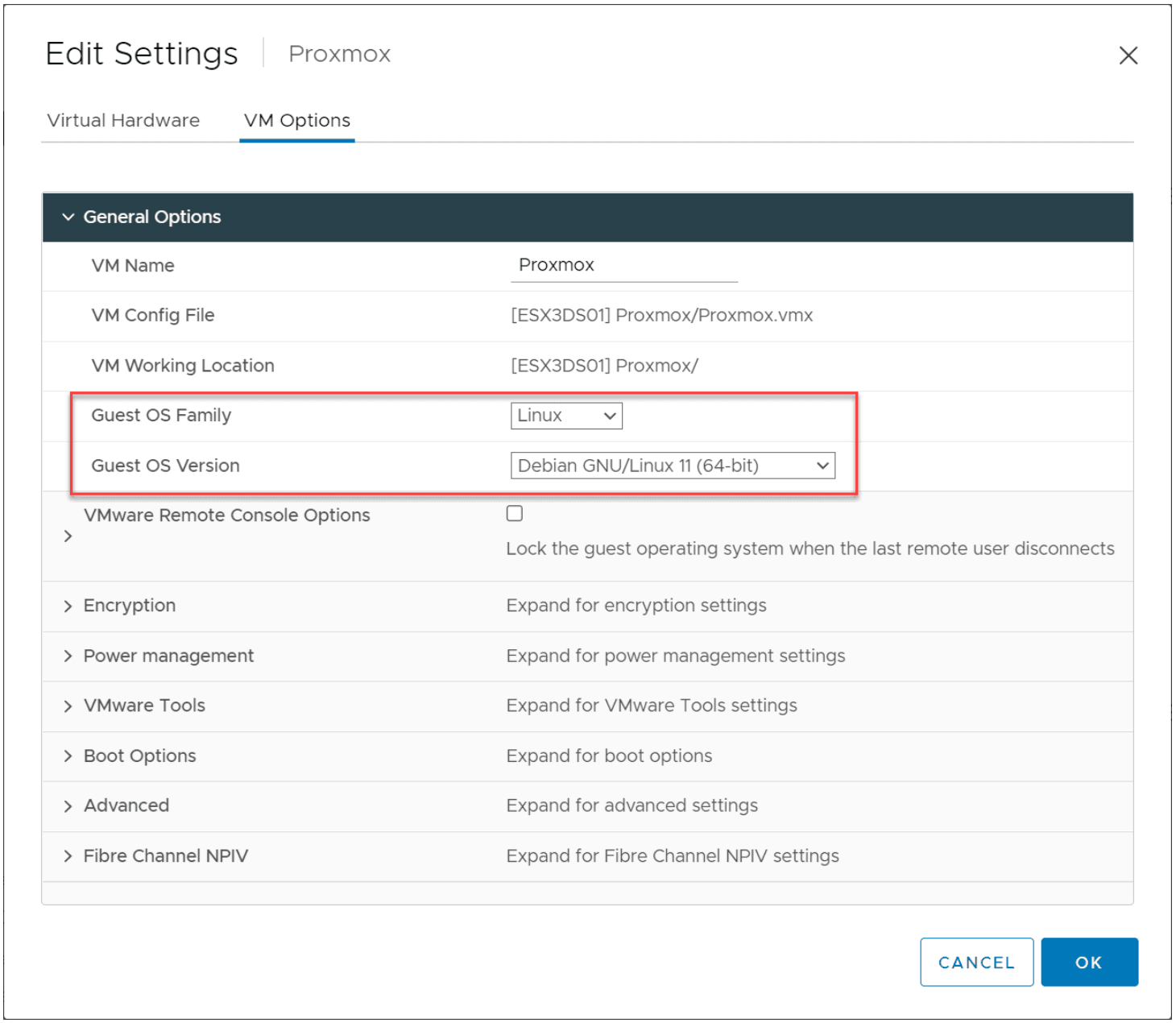
Nested Proxmox VMware installation in ESXi

The first thing you need for your nested Proxmox VMware installation in ESXi is to download the Proxmox ISO for installation. You can download the Proxmox ISO here:

- Get the free Proxmox VE ISO installer
- Current version Proxmox VE 7.1

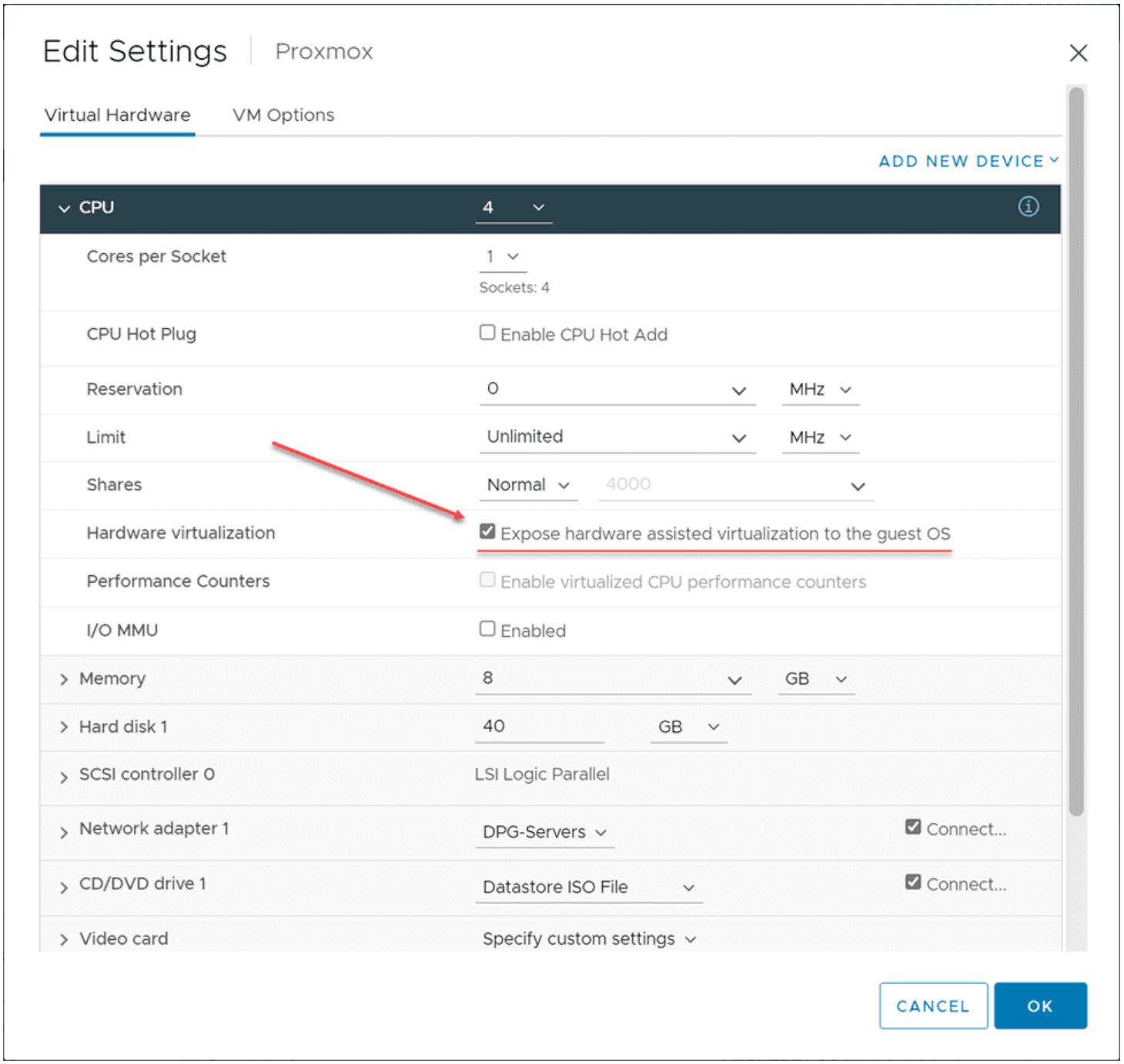
You will mount the ISO to your virtual machine in VMware vSphere like you would any other OS installation. Create a new VMware vSphere virtual machine with the following details:

- Guest OS Family – **Linux**
- Guest OS Version – **Debian GNU/Linux 11 (64-bit)**



Proxmox VMware virtual machine settings

Next, make sure to expose hardware-assisted virtualization to the guest OS for your soon-to-be Proxmox installation. As most of us are familiar with in our nested ESXi labs, this is a simple checkbox in the properties of your VMware ESXi virtual machine under the CPU settings.

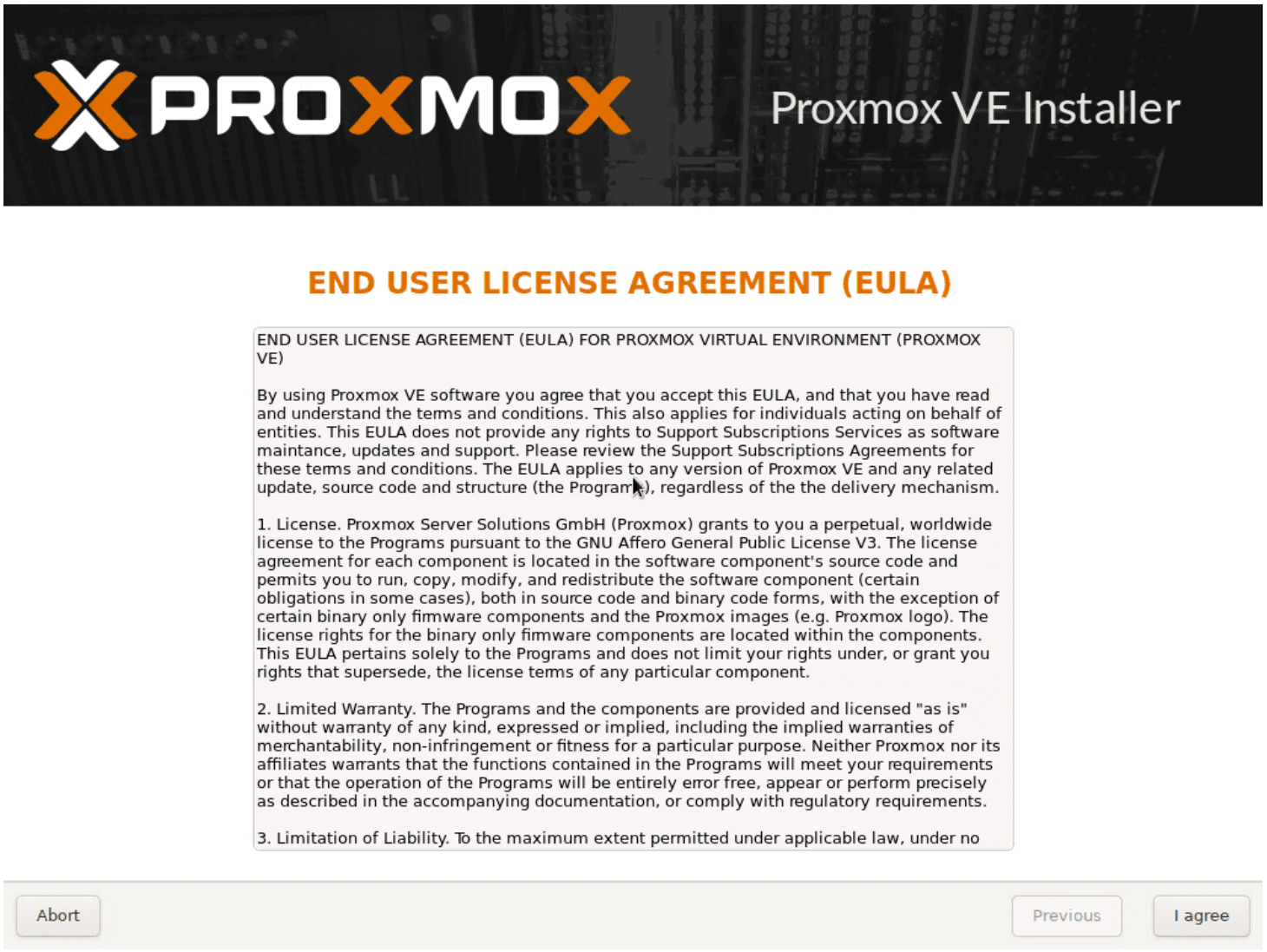


Exposing CPU hardware virtualization to the guest OS

After booting from the ISO, the Proxmox VE 7.1 installation begins. Select to **Install Proxmox VE**.

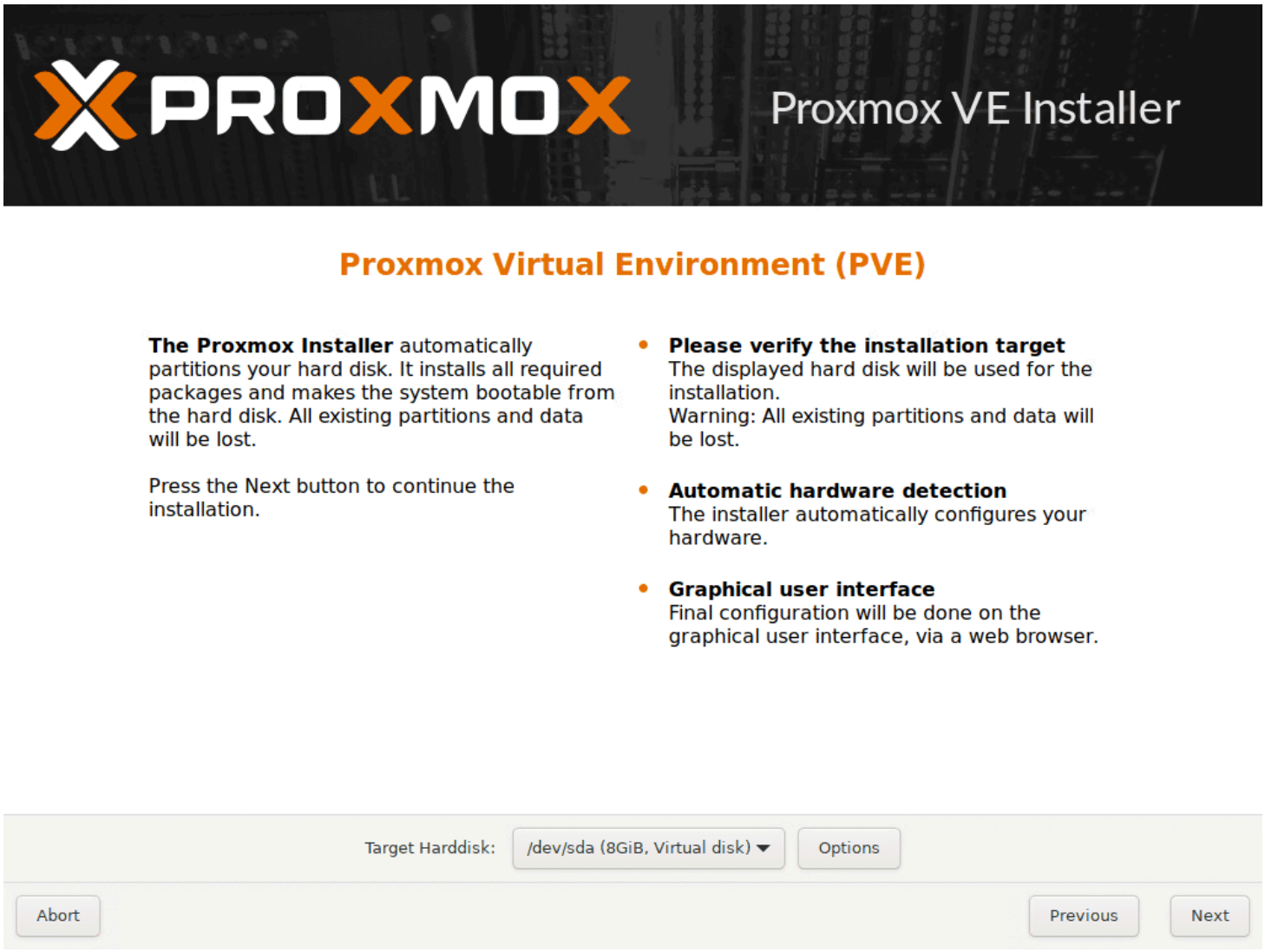
Booting the Proxmos 7.1 VE installer

First things first. Accept the EULA to proceed.



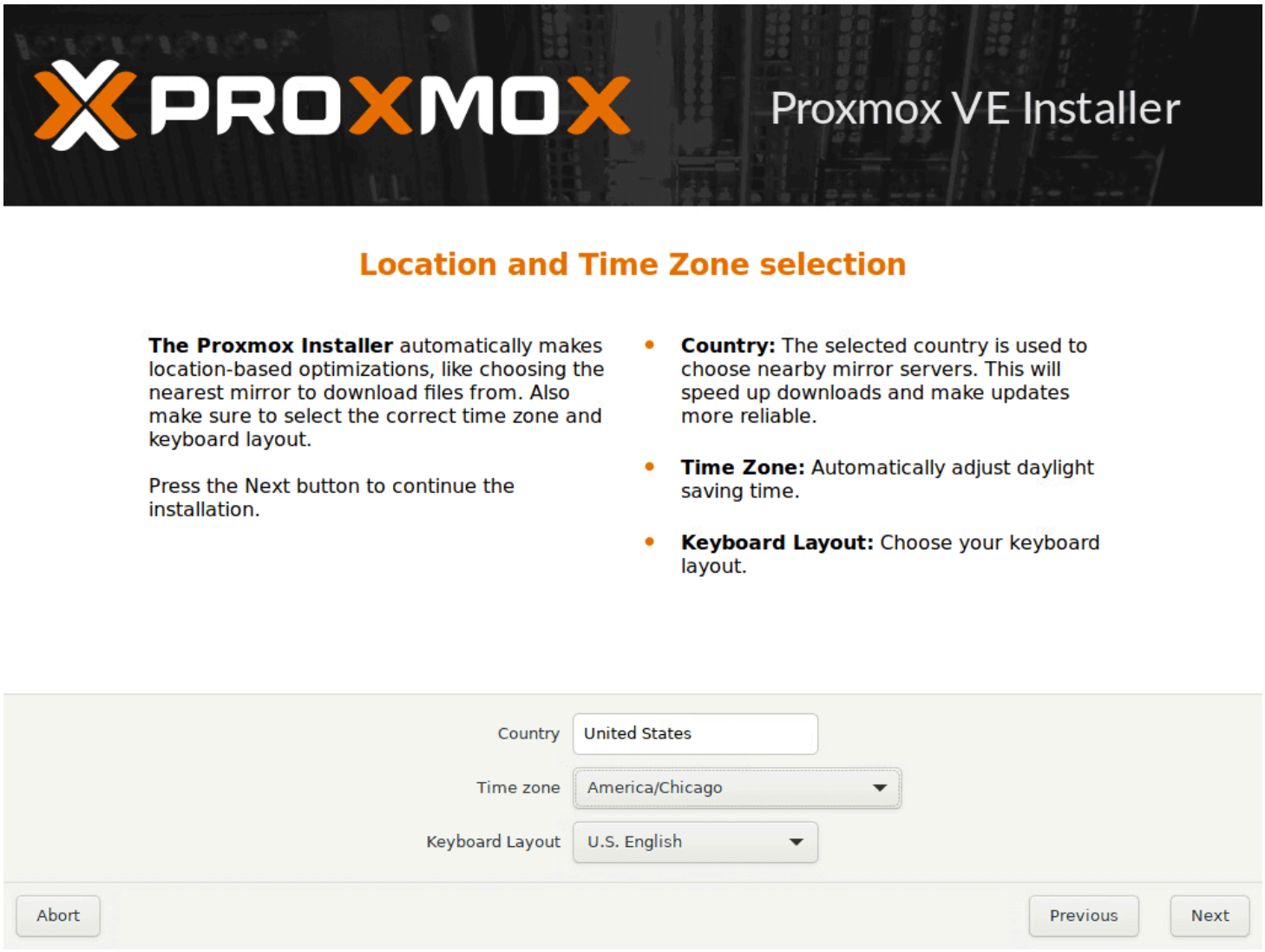
Accept the EULA for Proxmox VE 7.1

Next, you can customize the disk partition layout if you choose. However, for my nested Proxmox VMware installation, I am accepting the defaults.



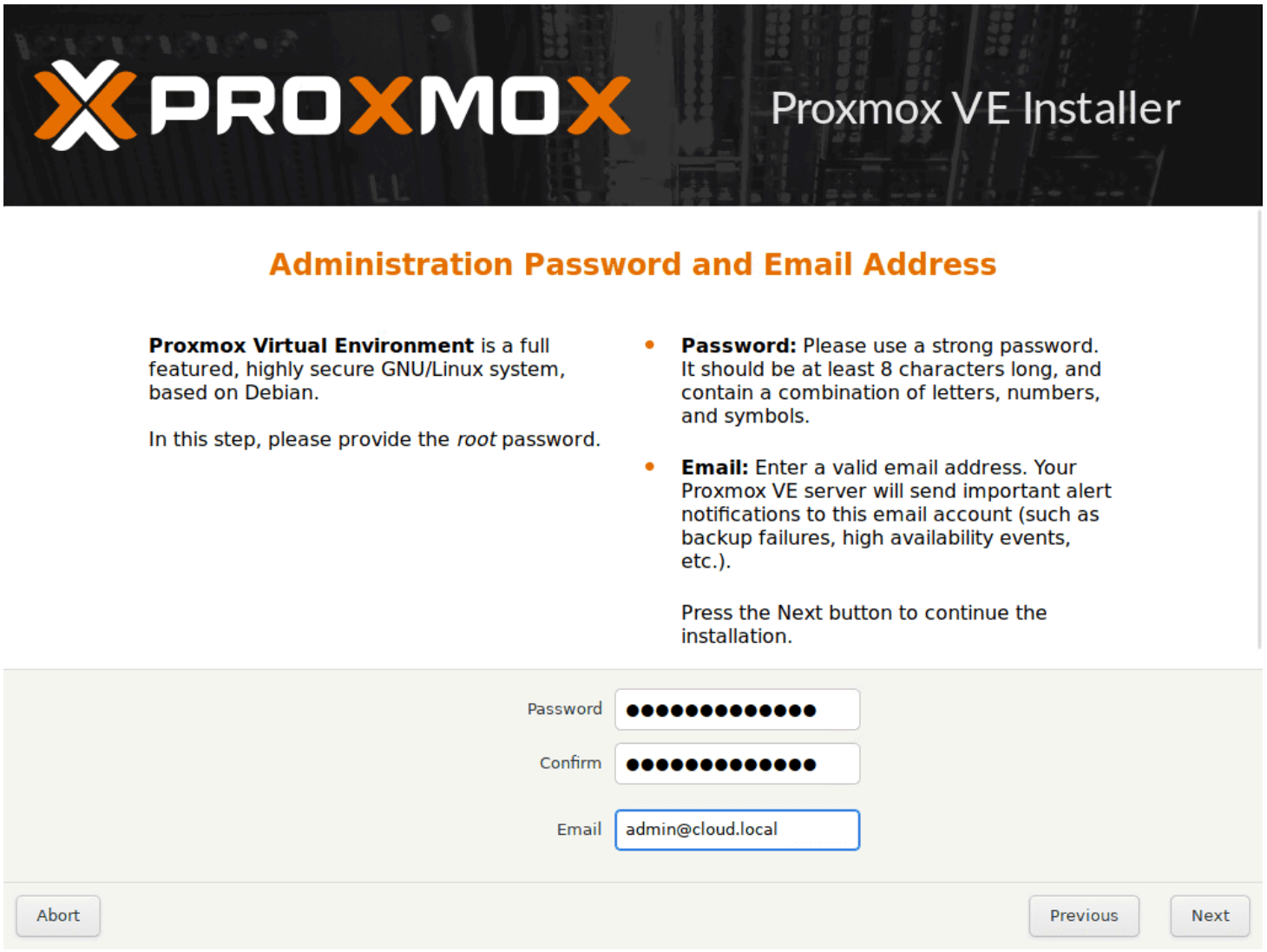
Select the disk partitioning to be used with the Proxmox VE 7.1 installation

Next up is setting your location and time zone configuration.



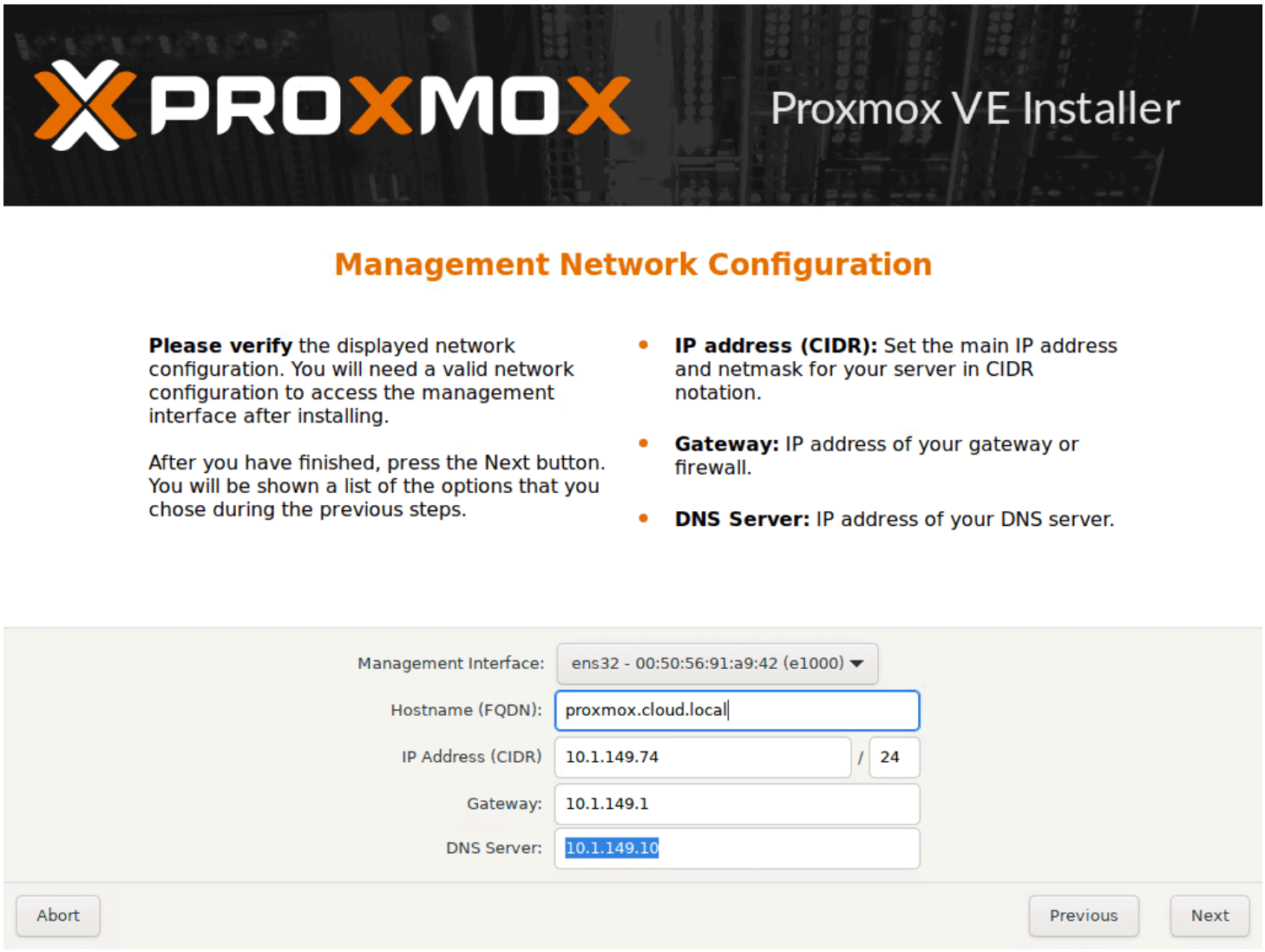
Set the location and time zone

Configure the password for your **root** account. Also, Proxmox has you enter an email address.



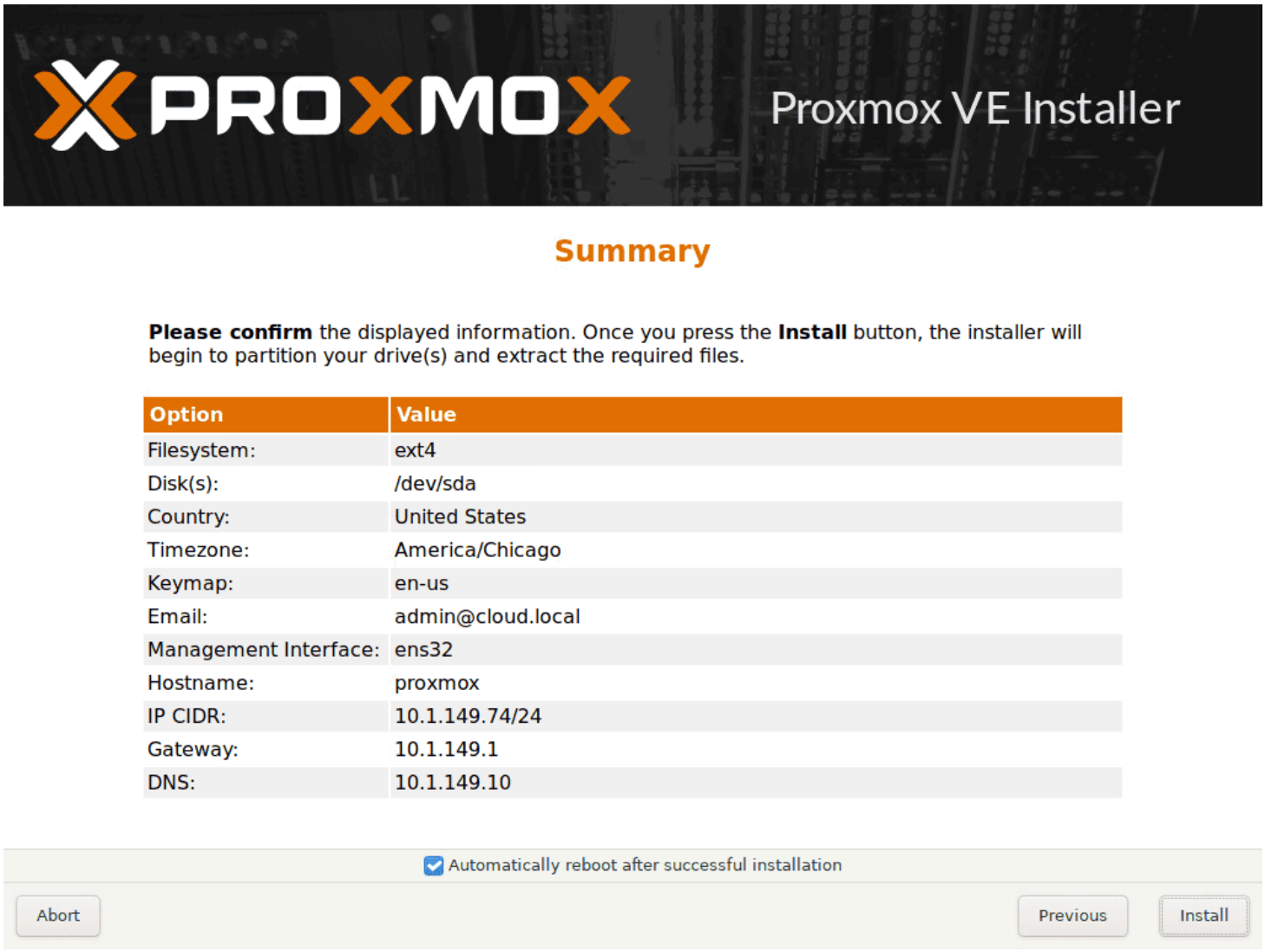
Set the administrator password and email address

Configure the Proxmox hostname and your network configuration.



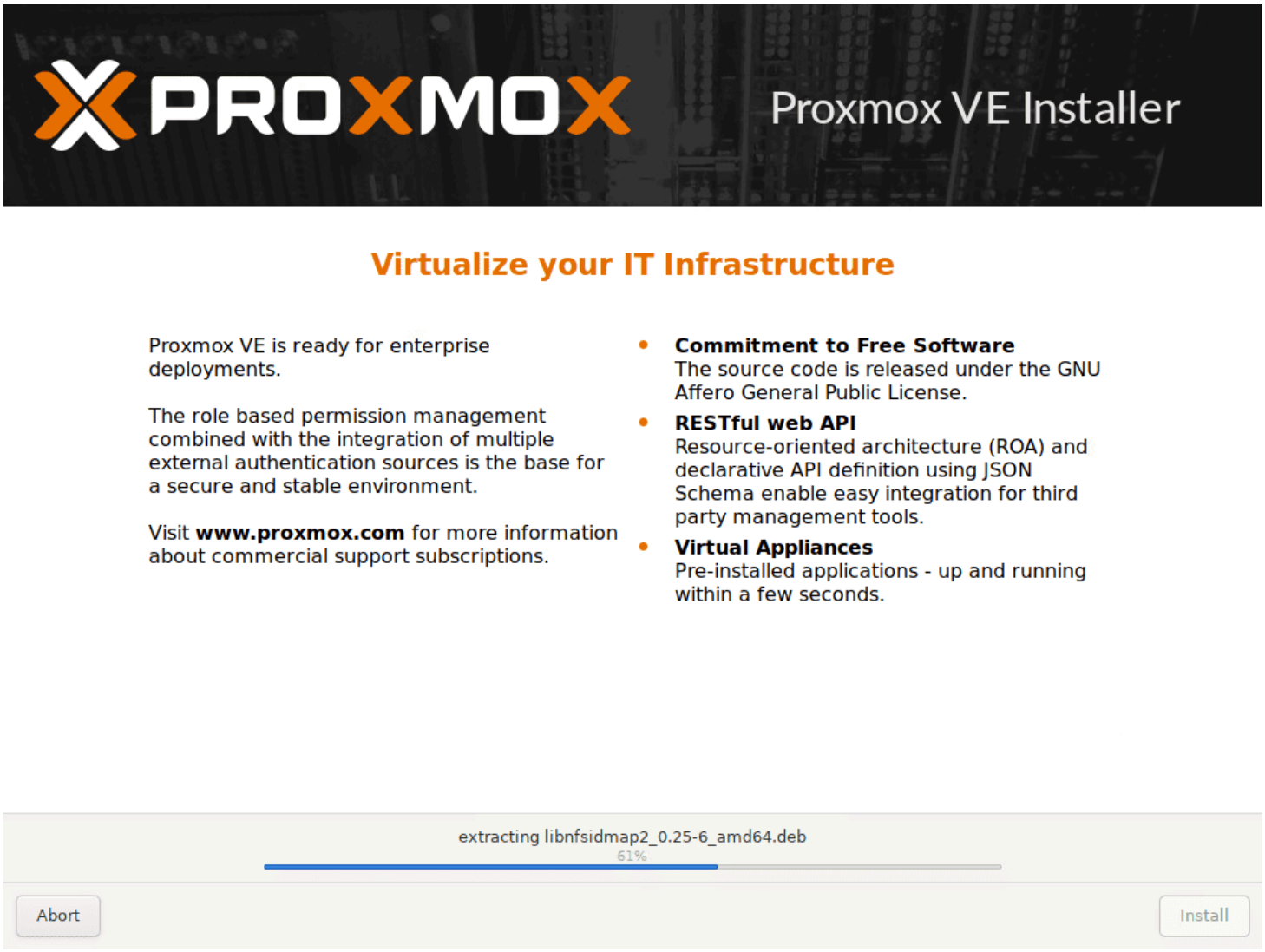
Set the hostname and network configuration

Finally, we come to the Summary screen. Here, review the configuration and validate your settings. Then, click **Install**.



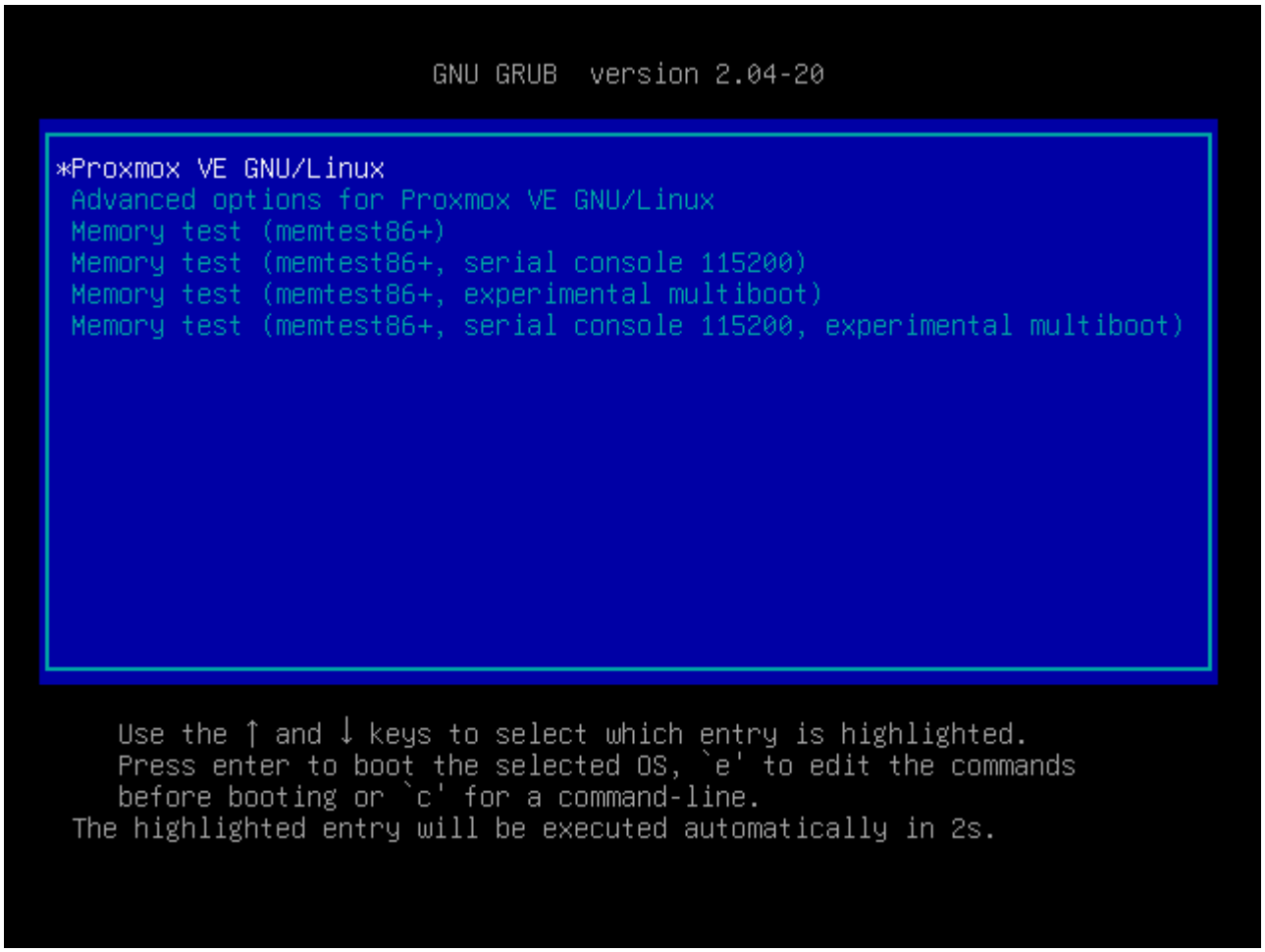
Summary of the Proxmox VE 7.1 installation

The installation process begins.



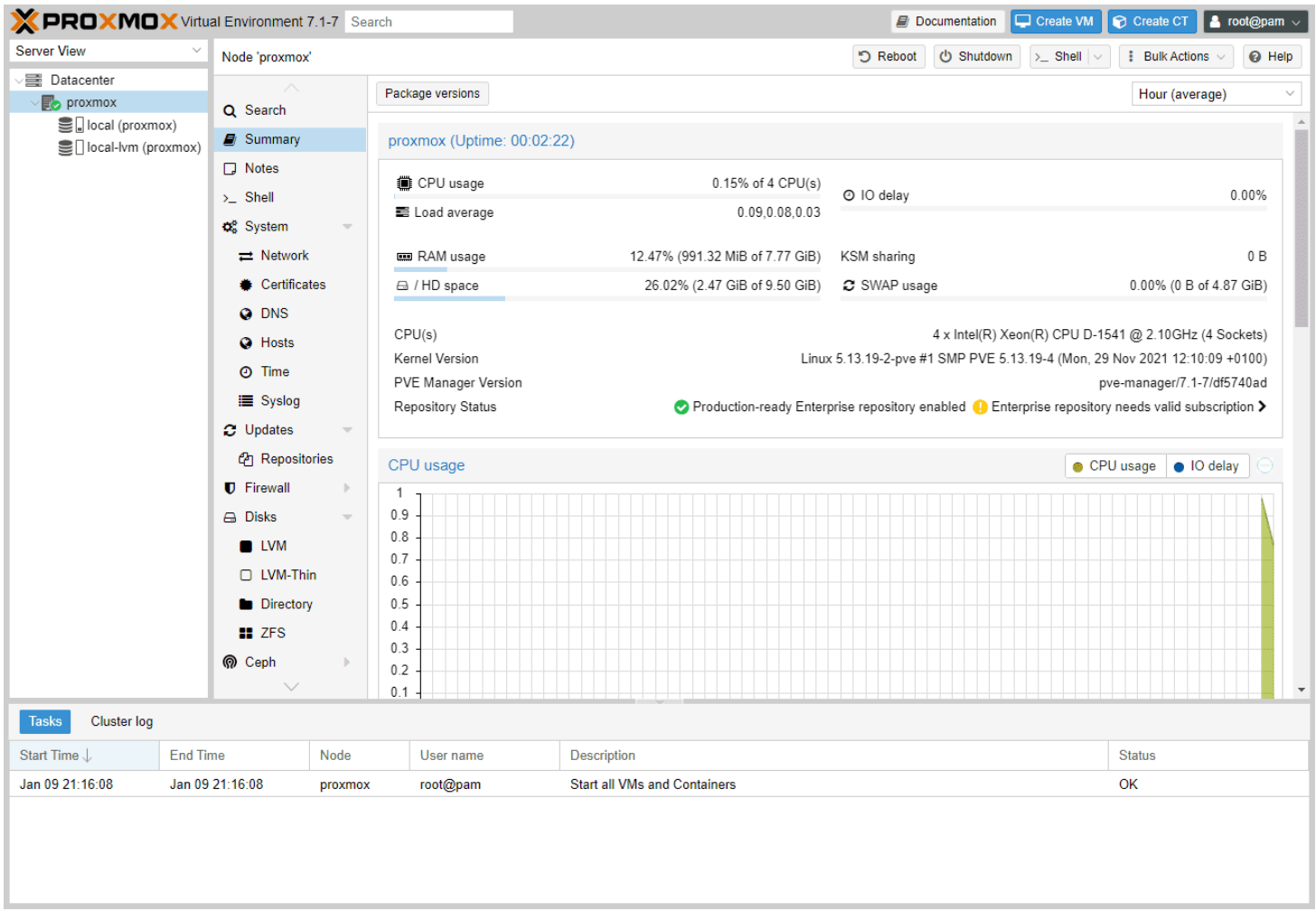
Proxmox VE 7.1 installation proceeds

After finishing the installation, the Proxmox server will reboot. Below is the boot screen captured as it reboots from the installation.



Proxmox VE 7.1 boots as a VMware ESXi VM

Finally, we are logged into the Proxmox web GUI using root and the password configured during the installation. Overall, the nested Proxmox VMware installation in ESXi was straightforward and easy. If you want to play around with Proxmox in a nested configuration, VMware vSphere provides a great way to do this using the basic functionality we have used for quite some time with nested ESXi installations.



Logged into the Proxmox VE 7.1 web interface

Wrapping Up

Proxmox is a cool hypervisor that provides a lot of features in an open-source, freely available download. The latest Proxmox VE 7.1 release has a lot of out-of-the-box features and can be used to run production workloads. If you want to play around with Proxmox, running the hypervisor inside a nested virtual machine in VMware ESXi is a great way to gain experience with installing, operating, troubleshooting, and other aspects of the virtualization solution.

You can learn more about Proxmox from their official page found here:

- Proxmox – Powerful open-source server solutions

How to Enable Proxmox Nested Virtualization

December 18, 2023
home lab

Proxmox nested virtualization

You may or may not have heard about nested virtualization. However, nested virtualization is a great way to set up virtualization labs and other learning environments. Nested virtualization in Proxmox is fairly easy to configure. With the Proxmox hypervisor, you can run nested hypervisors inside virtual machines. Let's see what nested virtualization is and how to set this up in Proxmox.

Table of contents

What is nested virtualization?

Nested virtualization allows running a hypervisor, like Proxmox VE (PVE) or others like VMware ESXi or Hyper-V, inside a virtual machine (VM), which itself runs on another hypervisor. With nested virtualization configured, a host hypervisor is hosting a guest hypervisor (as a VM). Then, the “nested” hypervisor is capable of hosting its own VMs. It allows utilizing the hardware acceleration from the host system and virtualization technology, enabling a VM within another VM.

Why do it?

You might wonder why you would want to run nested virtualization in Proxmox. Well, nested virtualization is a great way to learn. There may be other production use cases. However, with nested virtualization, you can set up one physical Proxmox host in the home lab and then run a cluster of Proxmox hosts as VMs.

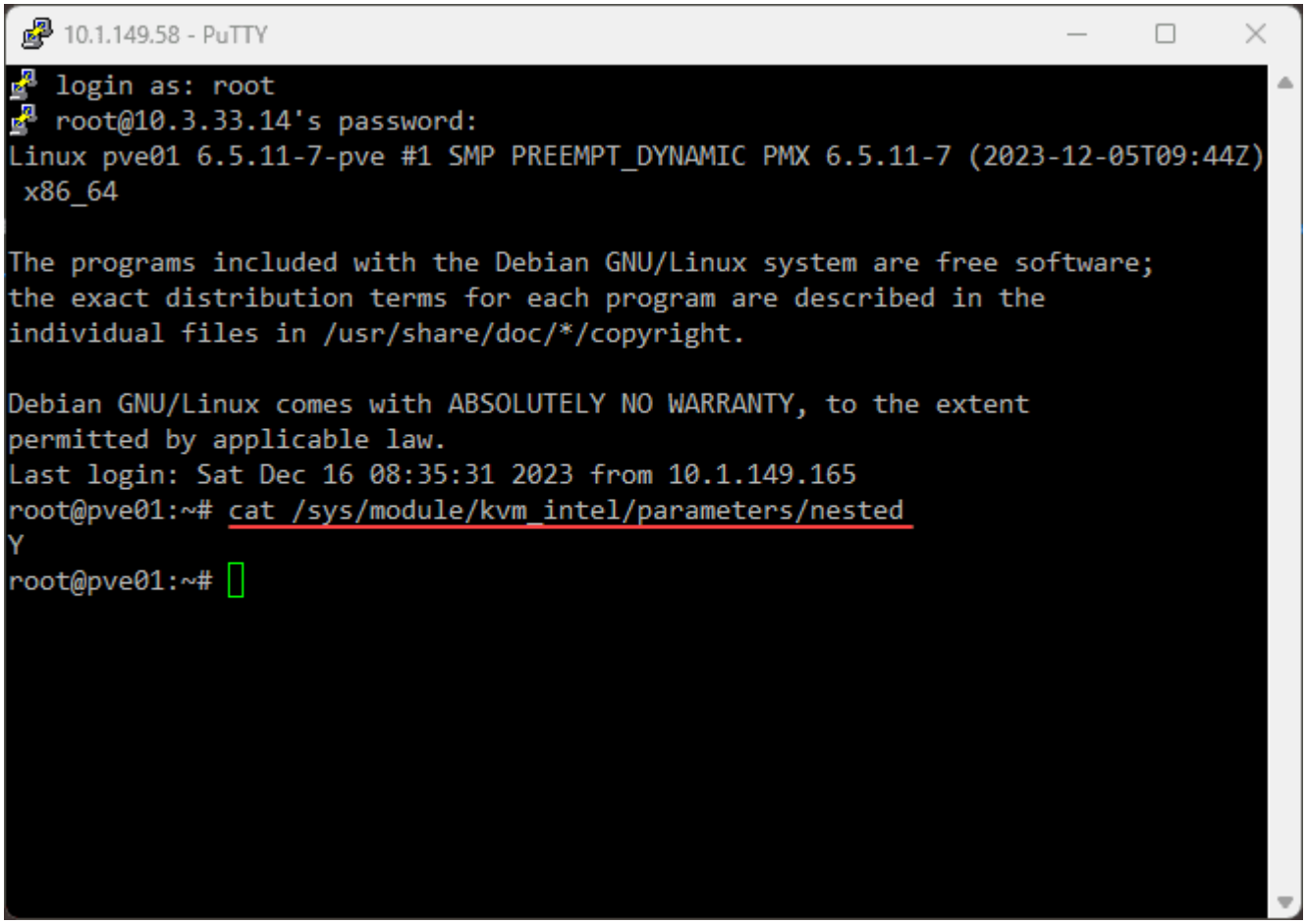
By configuring nested virtualization, you can treat the VM hypervisor like any other VM. It means you can use things like snapshots to capture the state of the VM. Then, you can roll it back like any other VM in your environment.

Using snapshots on these nested virtualization VMs allows playing around with and learning new skills, without worrying about breaking things.

Checking Processor Compatibility and Current Settings

First, determine if your Proxmox host’s processor supports nested virtualization. For Intel CPU, use the command SSH into your Proxmox host as root and run the following:

```
cat /sys/module/kvm_intel/parameters/nested
```

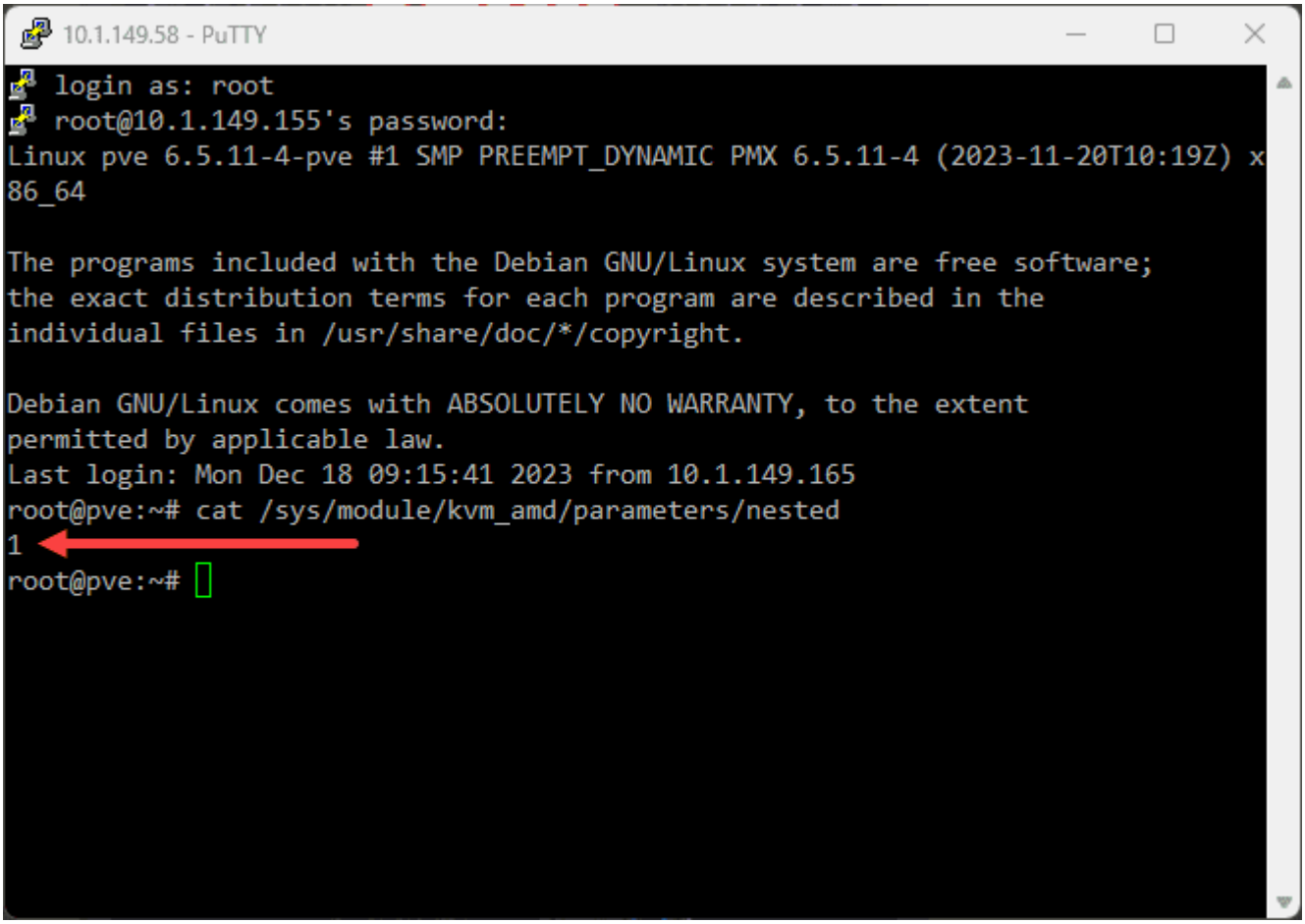


Checking intel processor compatibility with proxmox nested virtualization

For AMD CPU, use:

```
cat /sys/module/kvm_amd/parameters/nested
```

The output will be either “N” or “0” (not enabled) or “Y” or “1” (enabled).

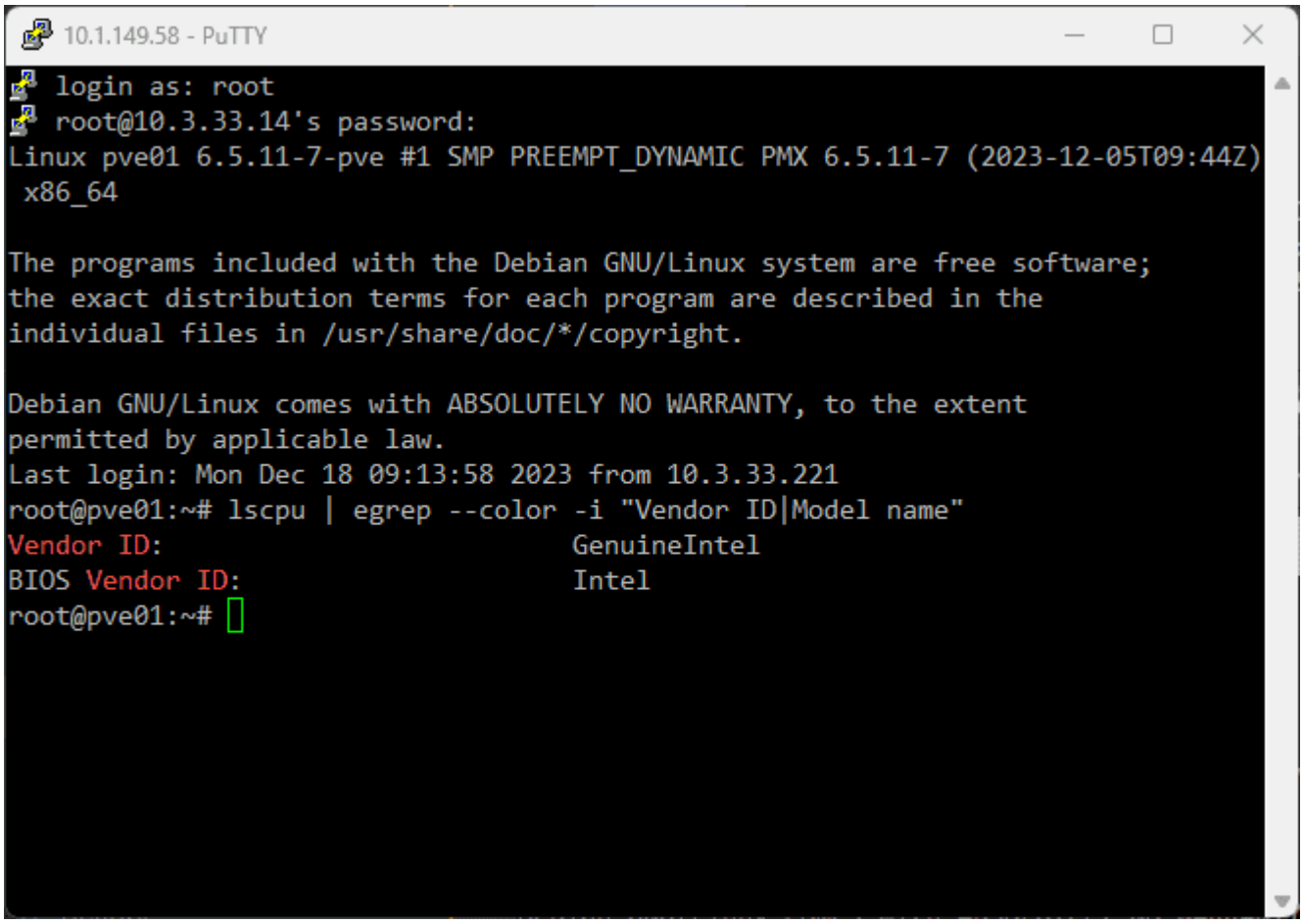


Checking amd processor compatibility for proxmox nested virtualization

Additionally, check the processor model using:

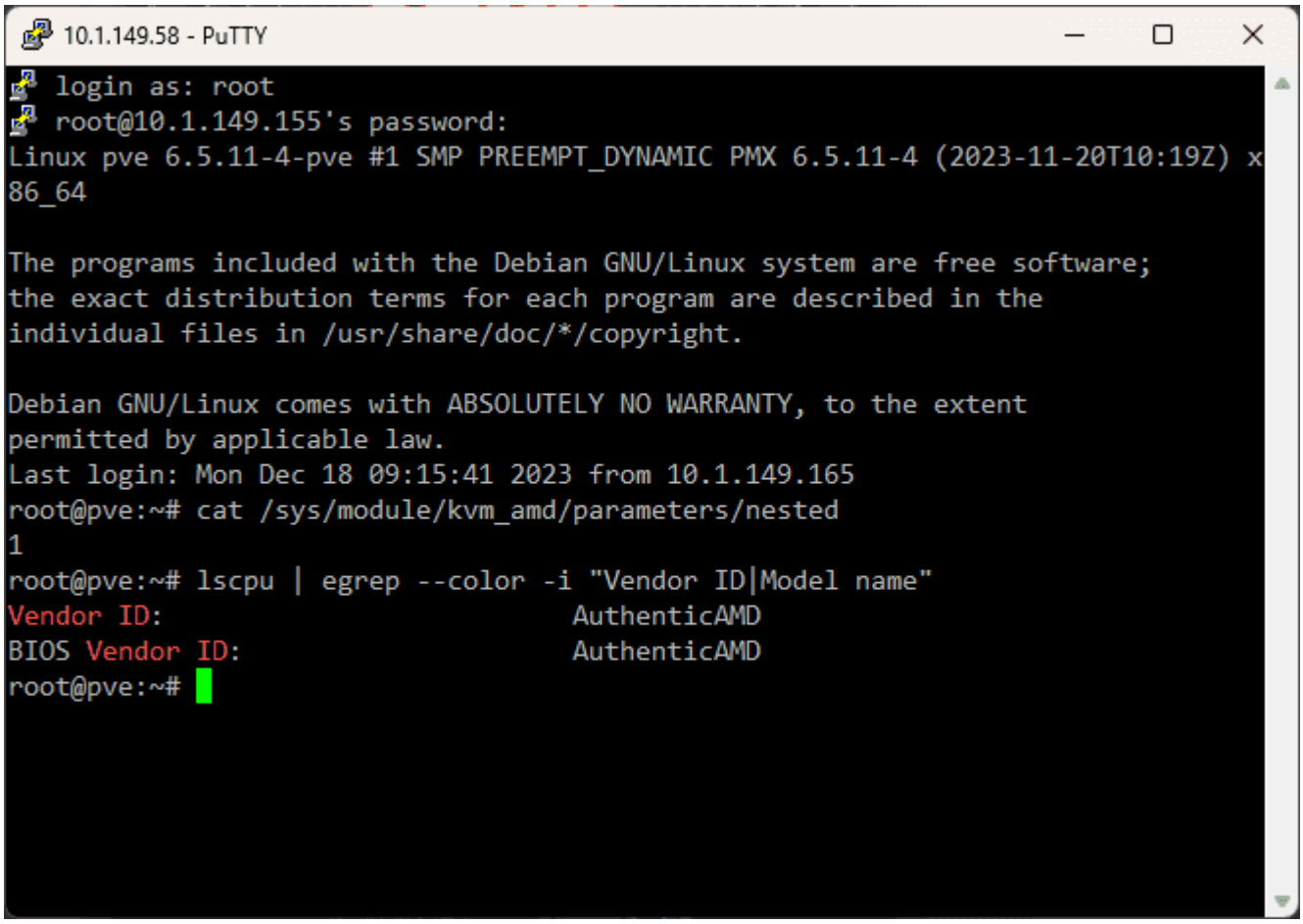
```
lscpu | egrep --color -i "Vendor ID|Model name"
```

Intel CPUs will look like the following:



Checking processor type when intel

AMD CPUs will look like the following:



Checking your processor type when amd

Enabling Nested Virtualization on Proxmox Host

You can activate it using specific commands if nested virtualization is not enabled. For Intel CPUs, use the commands:

```
sudo modprobe -r kvm_intel sudo modprobe kvm_intel nested=1 echo "options kvm-intel nested=Y" > /etc/modprobe.d/kvm-intel.conf

## Reload the kernel module
modprobe -r kvm_intel modprobe kvm_intel
```

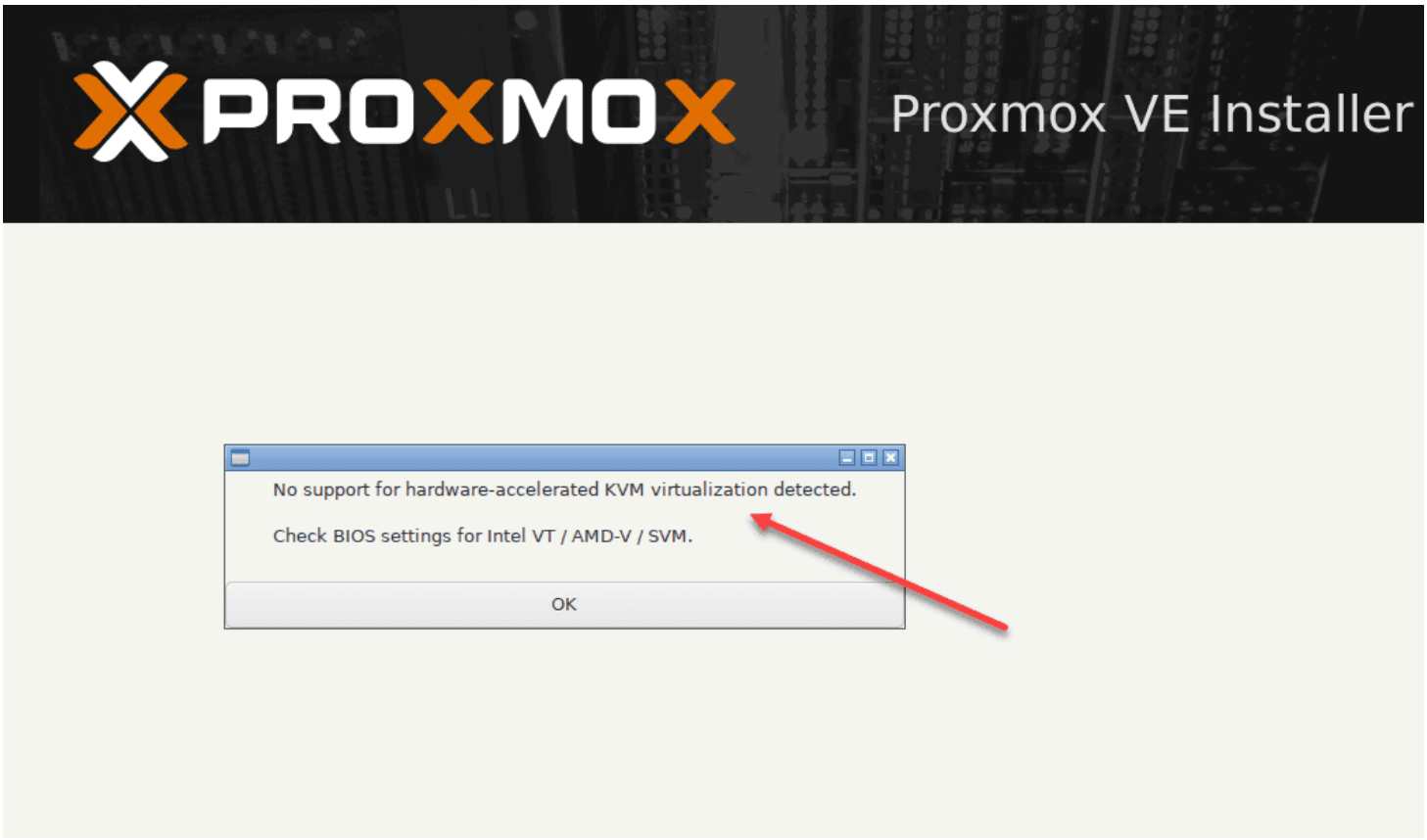
For AMD Proxmox host, the commands are slightly different:

```
sudo modprobe -r kvm_amd sudo modprobe kvm_amd nested=1 echo "options kvm-amd nested=1" > /etc/modprobe.d/kvm-amd.conf

## Reload the kernel module
modprobe -r kvm_amd modprobe kvm_amd
```

These commands effectively enable the nested virtualization feature for your CPU type.

What happens nested virtualization isn’t enabled?

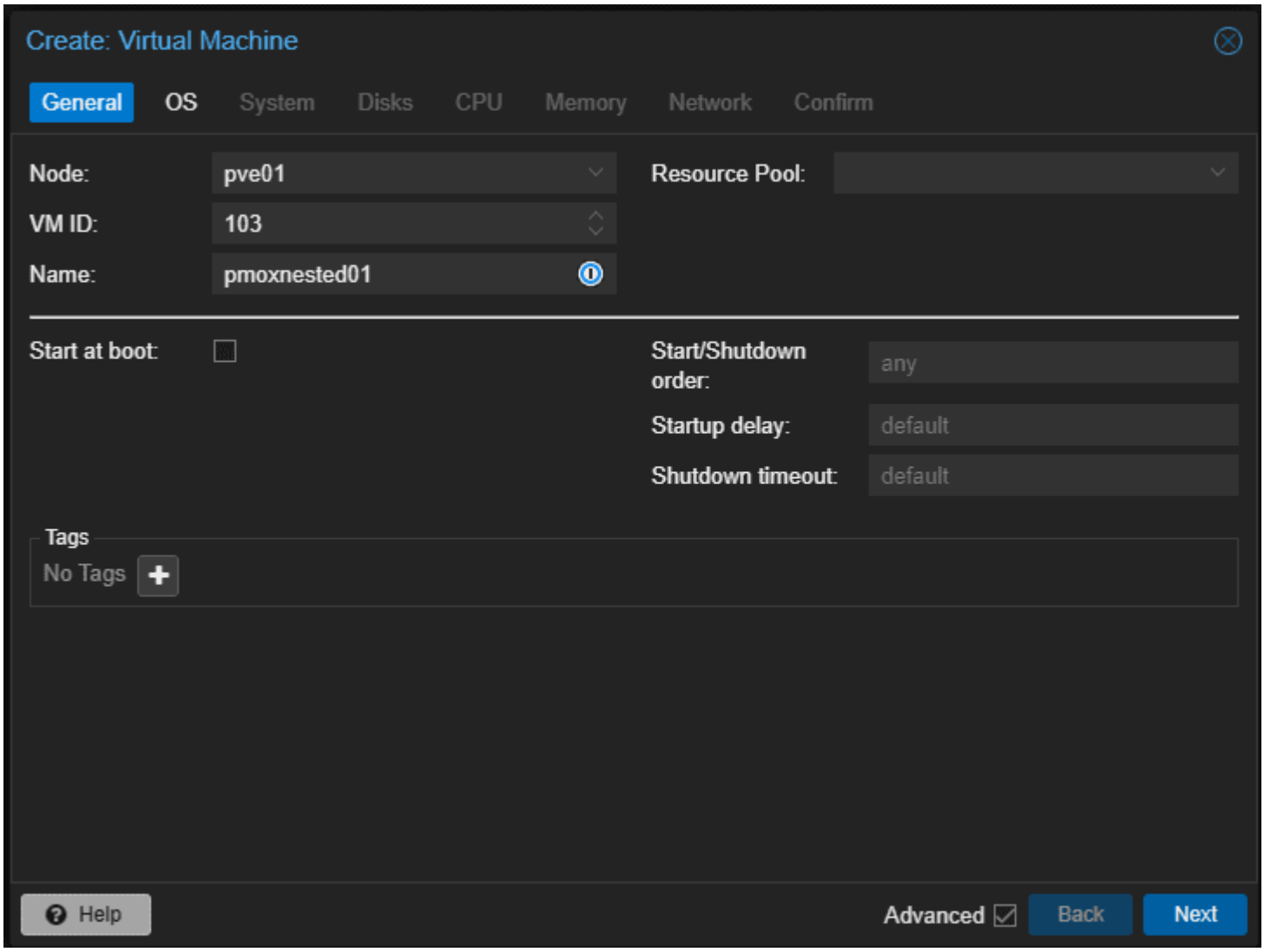
What will you see if there is no support for nested virtualization? Note the following error message displayed when trying to install a nested Proxmox hypervisor as a VM on a Proxmox physical host:



Error when nested virtualization is not enabled in proxmox

Configuring Nested Virtualization in VMs

After enabling nested virtualization on the Proxmox server, you must set up the virtual machines to utilize this feature when you create VM instances.



Creating a proxmox virtual machine with proxmox installed

Turn off the guest VM, then use the command to set the CPU type of the VM to “host”:

```
qm set <VMID> --cpu host
```

Setting the nested vm cpu to use the host cpu

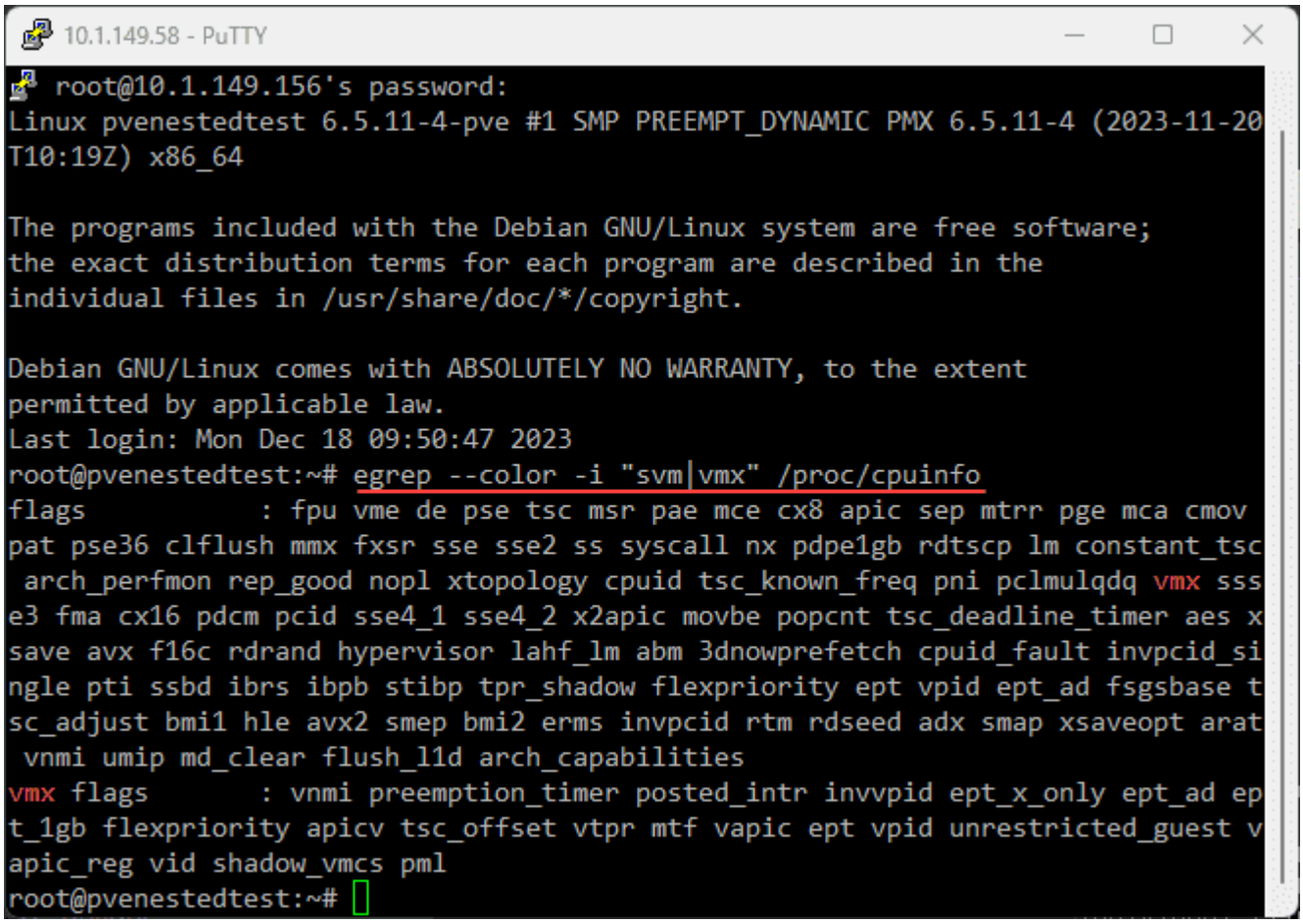
For AMD processors, you'll need to add additional parameters to the VM configuration file:

```
args: -cpu host,+svm.
```

Validating the Configuration

Finally, restart the VMs and validate the configuration. Check if VT-X is enabled in the VMs, which is necessary to host guest machines. The following command will help you verify if VT-X is enabled:

```
egrep --color -i "svm|vmx" /proc/cpuinfo
```



Verifying the nested cpu instructions on a nested proxmox hypervisor

FAQs on Proxmox Nested Virtualization

How Does Nested Virtualization Differ from Regular VM Setup?

Nested virtualization introduces a unique layer by allowing a virtual machine to function as a host for other VMs. This contrasts with standard virtualization, where VMs are directly managed by the physical host system. It's an advanced feature primarily used for development and testing environments.

Can I Use Both Intel and AMD CPUs for Nested Virtualization in Proxmox?

Yes, Proxmox supports nested virtualization on both Intel and AMD CPUs. However, the commands to enable this feature differ slightly. For Intel, use the **kvm_intel** module, and for AMD, use **kvm_amd**. The key is to adjust the kernel module settings to enable the nested feature.

Is It Possible to Enable Nested Virtualization from the Proxmox Web UI?

While some configurations can be managed through the Proxmox web interface, enabling nested virtualization involves command-line operations. Commands like **modprobe** and editing files in **/etc/modprobe.d/** are essential steps that are performed in the CLI environment.

What Should I Check Before Enabling Nested Virtualization?

Ensure that your Proxmox host's CPU supports virtualization and that the nested virtualization feature is not already enabled. Use **lscpu** to check your CPU type and **cat /sys/module/kvm_{intel/amd}/parameters/nested** to verify if nested virtualization is enabled.

How Do I Know if VT-X/AMD-V is Enabled in My VMs?

After setting up nested virtualization, use the command **egrep –color -i "svm|vmx" /proc/cpuinfo** within your VM. If the output includes "vmx" or "svm," it indicates that Intel VT-X or AMD-V, respectively, is enabled, allowing your VM to host other VMs.

Do I Need to Restart My Proxmox Host After Configuring Nested Virtualization?

While it's not always necessary to restart the entire Proxmox host, reloading the relevant kernel modules is crucial. Commands like **modprobe -r kvm_intel** and **modprobe kvm_intel** for Intel (and similar for AMD) are used to refresh the settings without needing a full system reboot.

Can Nested Virtualization Impact Proxmox Performance?

Nested virtualization can create additional overhead, as it adds another layer of virtualization. The impact largely depends on the host system's resources and the complexity of the nested environments. Proper resource allocation and monitoring are key to maintaining optimal performance.

Is Nested Virtualization Suitable for Production Environments?

While technically feasible, nested virtualization is typically used in testing, development, or educational scenarios. For production environments, direct virtualization on the physical host is preferred for better performance and stability.

Is networking harder in nested environments?

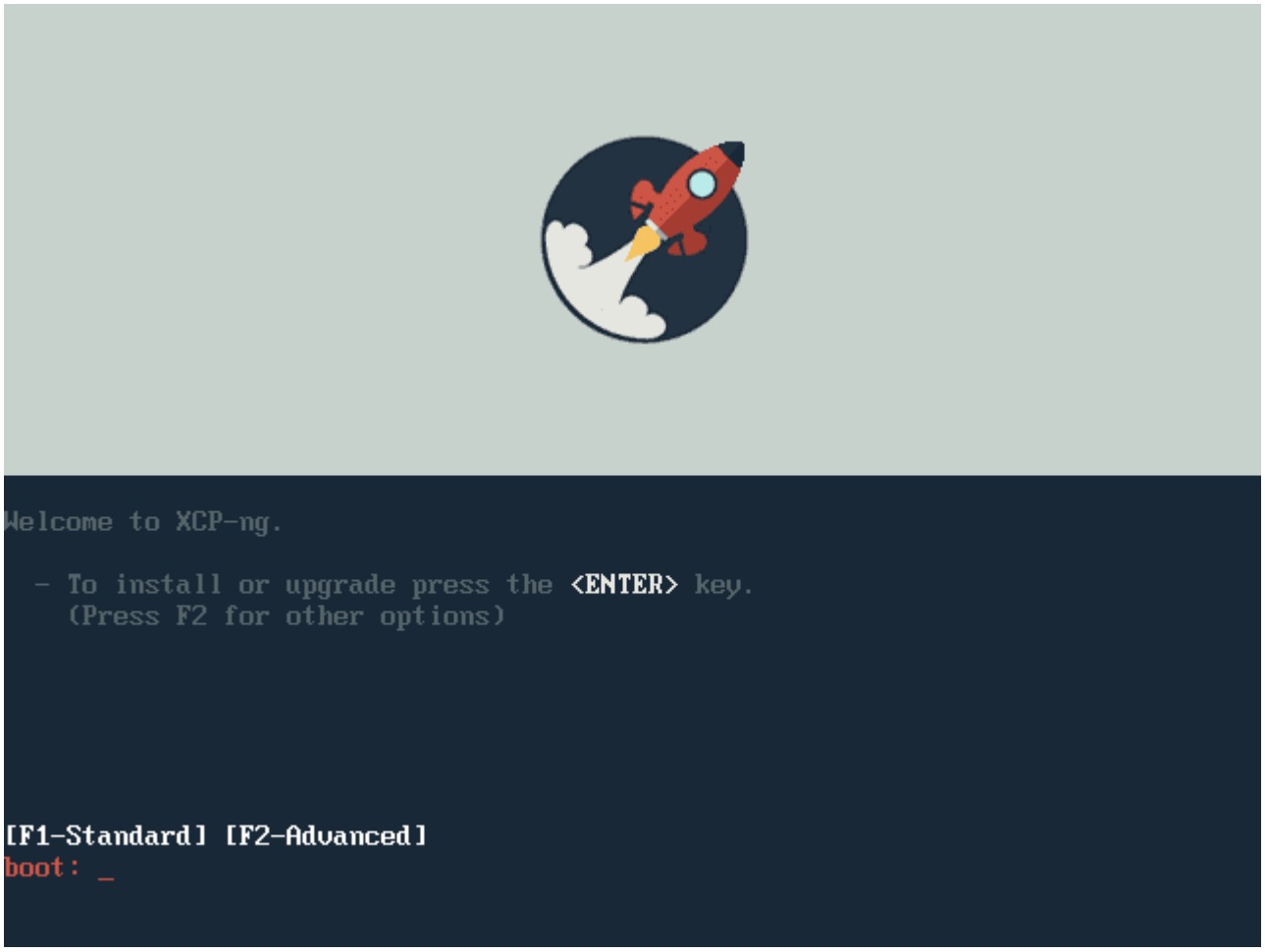
It can be. When you nest hypervisors as VMs, it creates another layer to understand with your network configuration.

Wrapping up

Nested virtualization in Proxmox is a great feature to take advantage of. With nested virtualization, you can run nested Proxmox VE servers, or even other hypervisors. It allows creating labs and other learning environments that can take advantage of a snapshot or even backups from Proxmox Backup Server as an example.

XCP-NG in VMware nested virtualization

July 1, 2022
Virtualization



XCP NG in VMware nested virtualization

As many of you may already know from reading my previous blog posts, I really like playing around with hypervisors and different technologies. The nested virtualization capabilities in VMware vSphere provide a great way to do this without the need for physical hardware laying around to install a hypervisor. I have had a hankering to play around with XCP-NG hypervisor in the lab environment and finally got around to doing this recently. Let's take a look at XCP-NG in VMware nested virtualization.

What is XCP-NG?

First of all, what is XCP-NG? XCP-NG is a virtualization platform based on the Xen Project hosted in the Linux Foundation. Being based on XenServer, it is a collaboration between individuals and businesses looking to deliver an open-source virtualization project that can meet the challenges of the enterprise today.

Its features include the following:

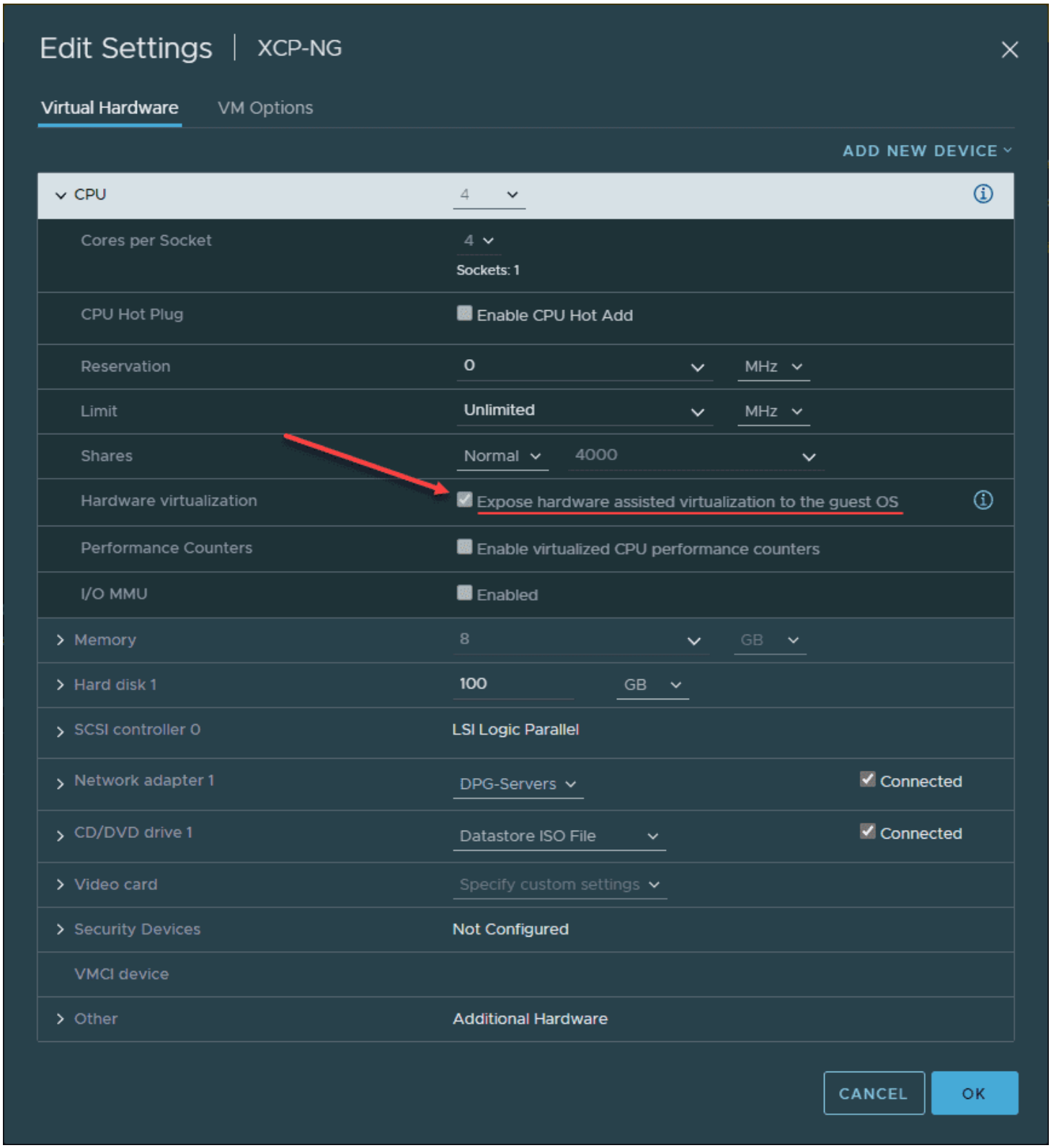
- **Modern UI** – It offers a modern UI for management called Xen Orchestra. There is also a Windows-based management tool called XCP-NG center that allows managing your XCP-NG hosts, etc.
- **Live migration** – This is probably what drew most of us to virtualization and got us hooked to begin with – live migration. XCP-NG offers the ability to move running virtual machines between hosts.
- **Scalability** – You can grow and scale your infrastructure as needed. Simply add a host to your XCP-NG environment
- **Security** – With XCP-NG, you can run your own private data center without the concerns of shared public cloud infrastructure.

You can learn more about XCP-NG and download the ISO installation from here:

- XCP-ng – XenServer Based, Community Powered

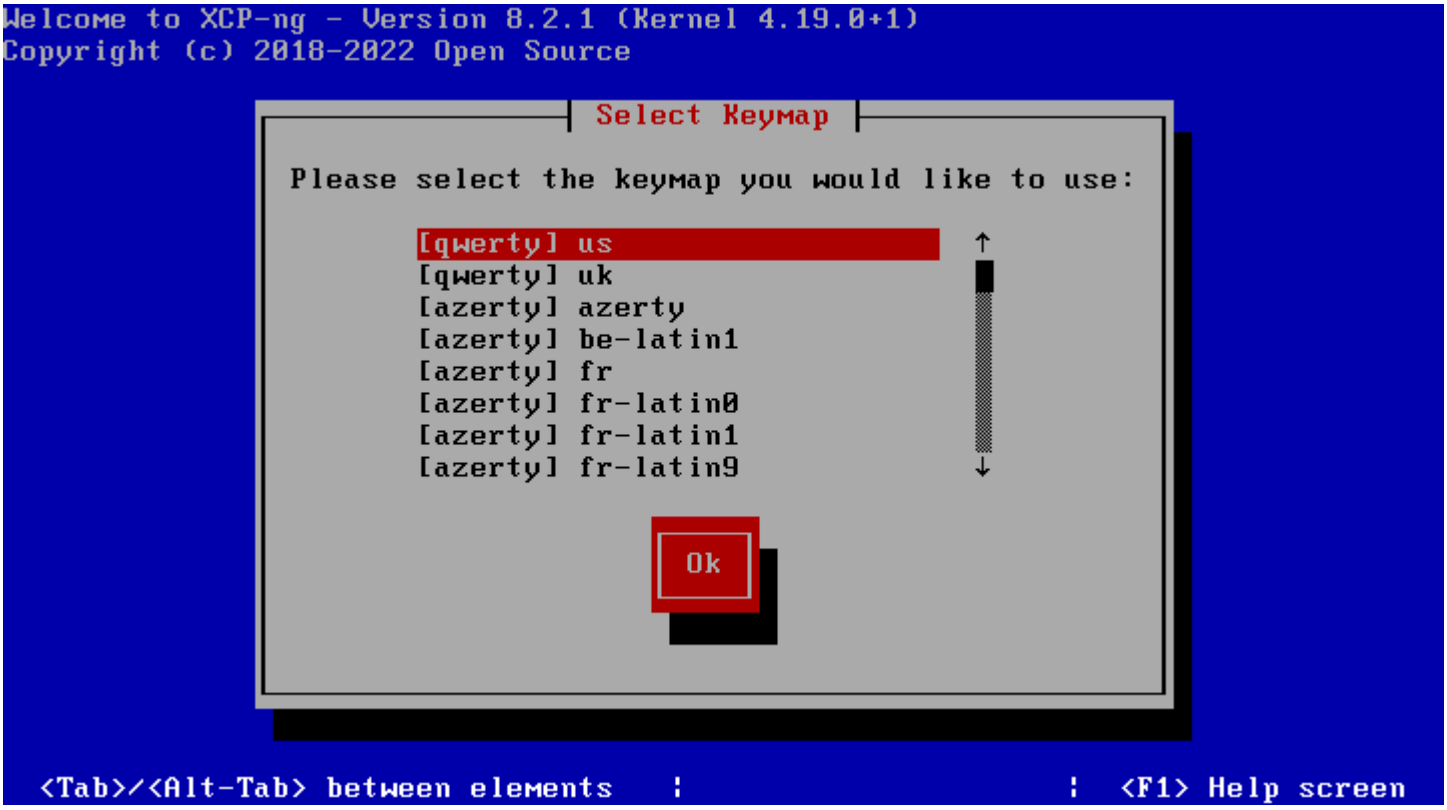
XCP-NG in VMware nested virtualization

Let's take a look at installing XCP-NG in VMware and see how you can do this with VMware nested virtualization, installing the XCP-NG hypervisor in your VMware vSphere environment is as simple as creating a new VMWare vSphere VM, enabling the CPU settings for nested virtualization and installing XCP-NG. Let's see what that looks like.



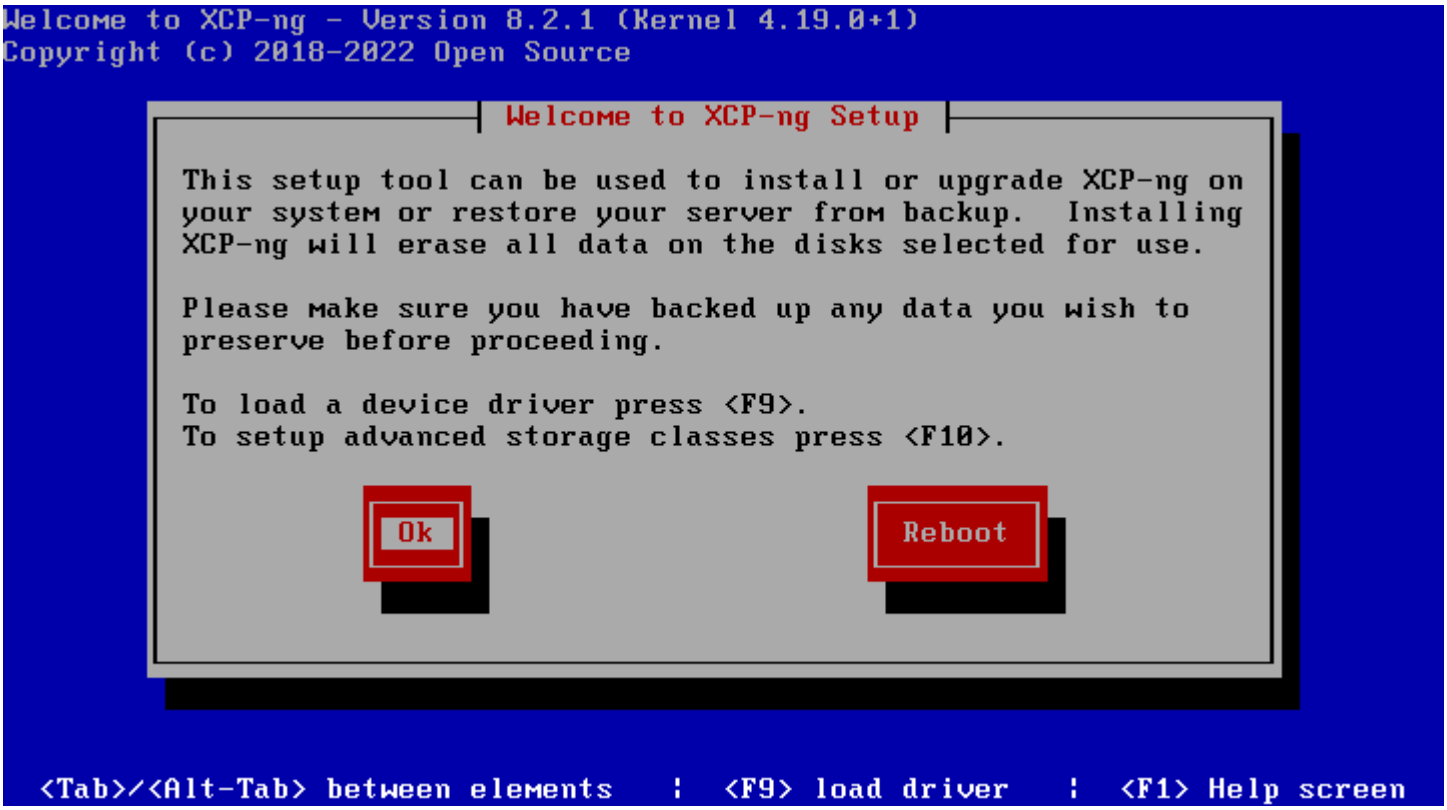
Enable nested virtualization for your XCP NG virtual machine

Select the locality for the keyboard layout.



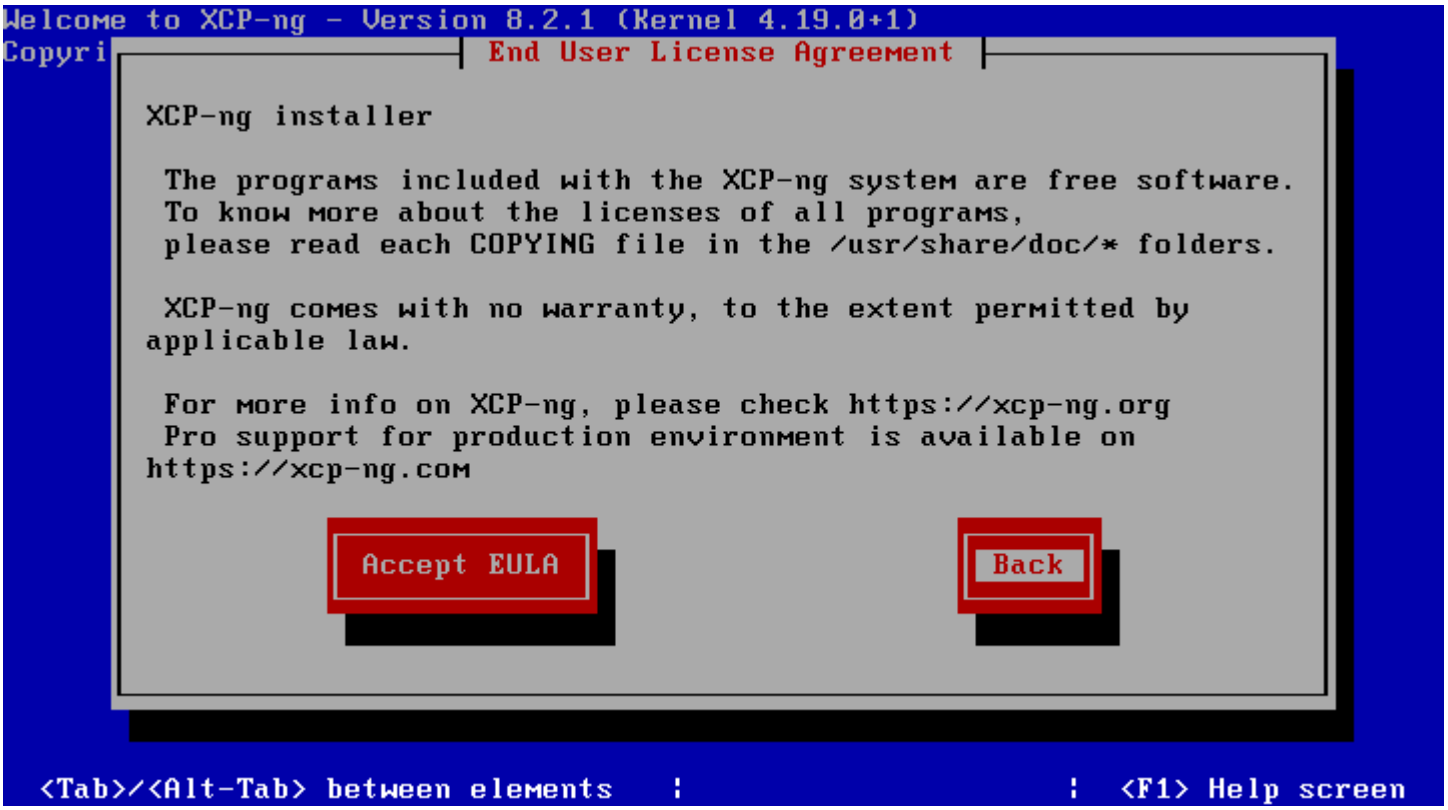
Choose the country for keyboard mapping

Press the OK button to move forward with the installation of XCP-NG.



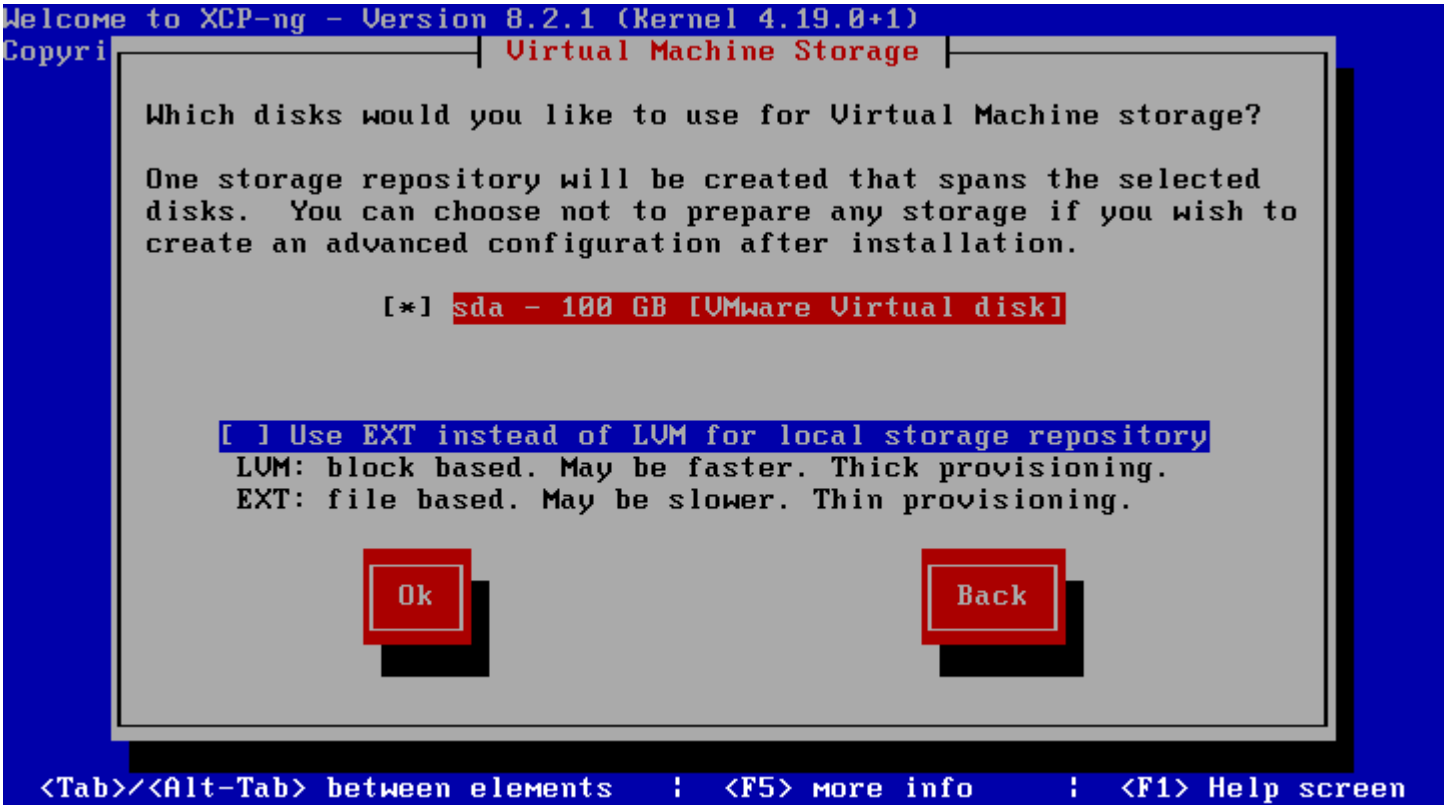
Beginning the setup wizard

Accept the EULA to move on with installation configuration.



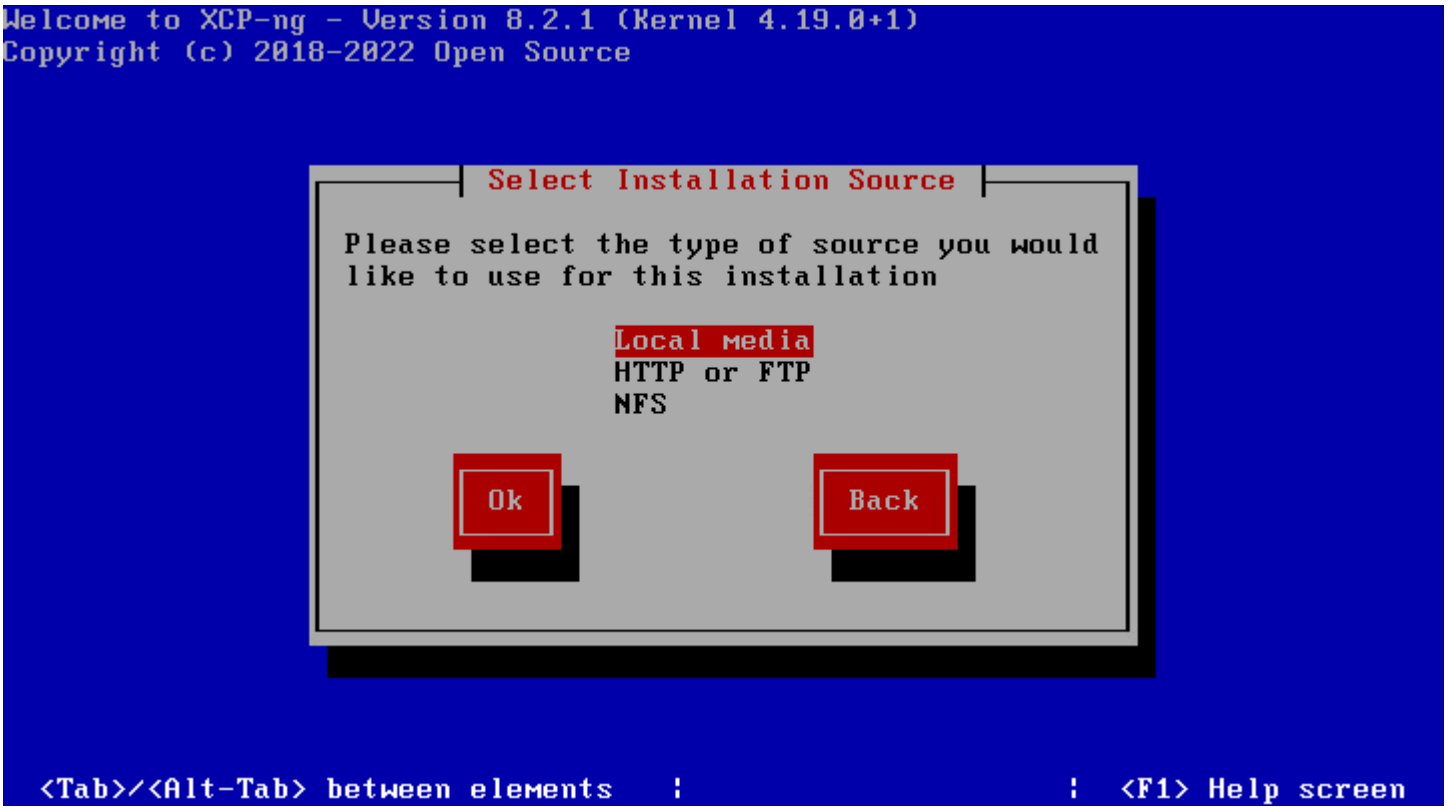
Accept the EULA for XCP NG installation

Select the local disk you want to install XCP-NG on and the provisioning method.



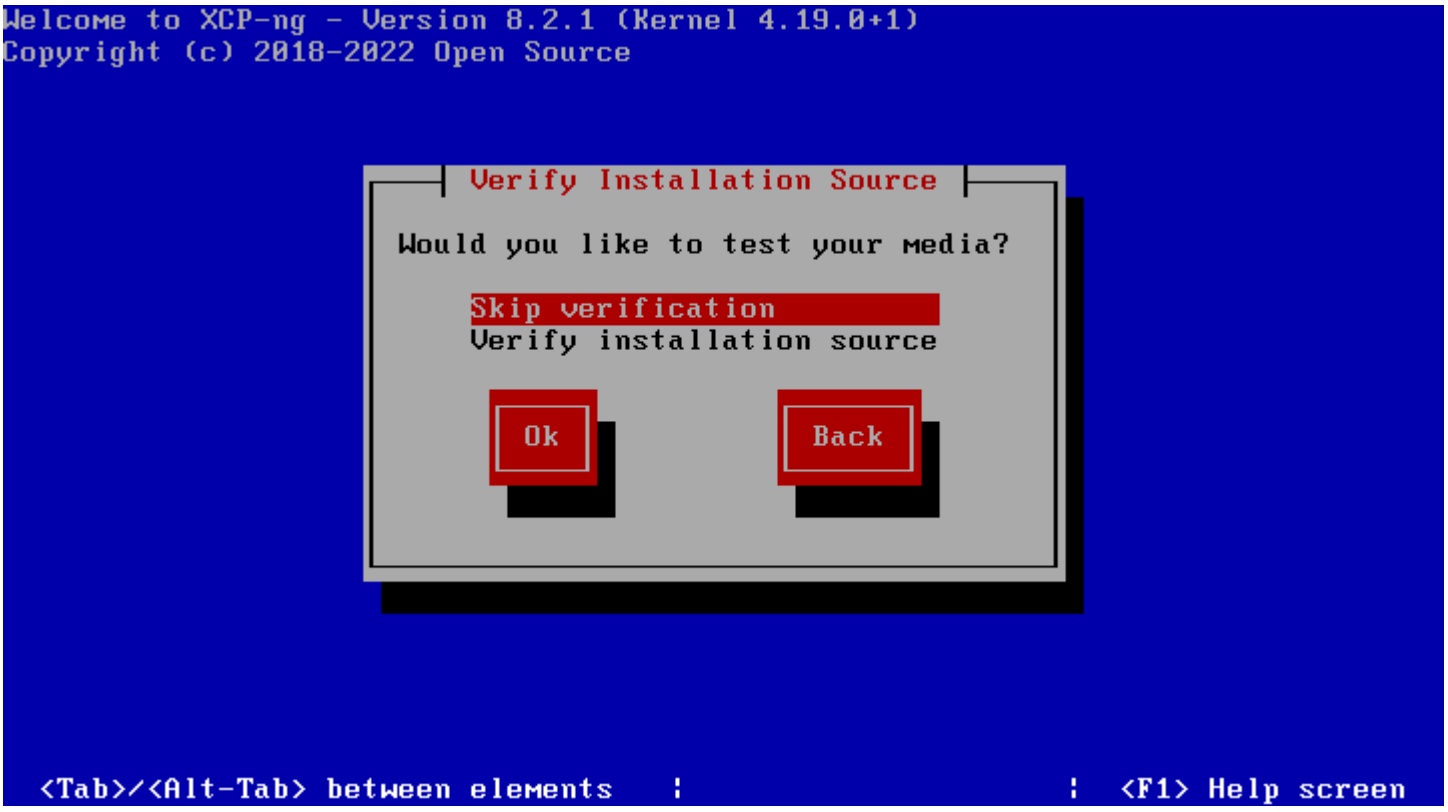
Choose the disk on which to install XCP NG

Select the media from which you want to install XCP-NG.



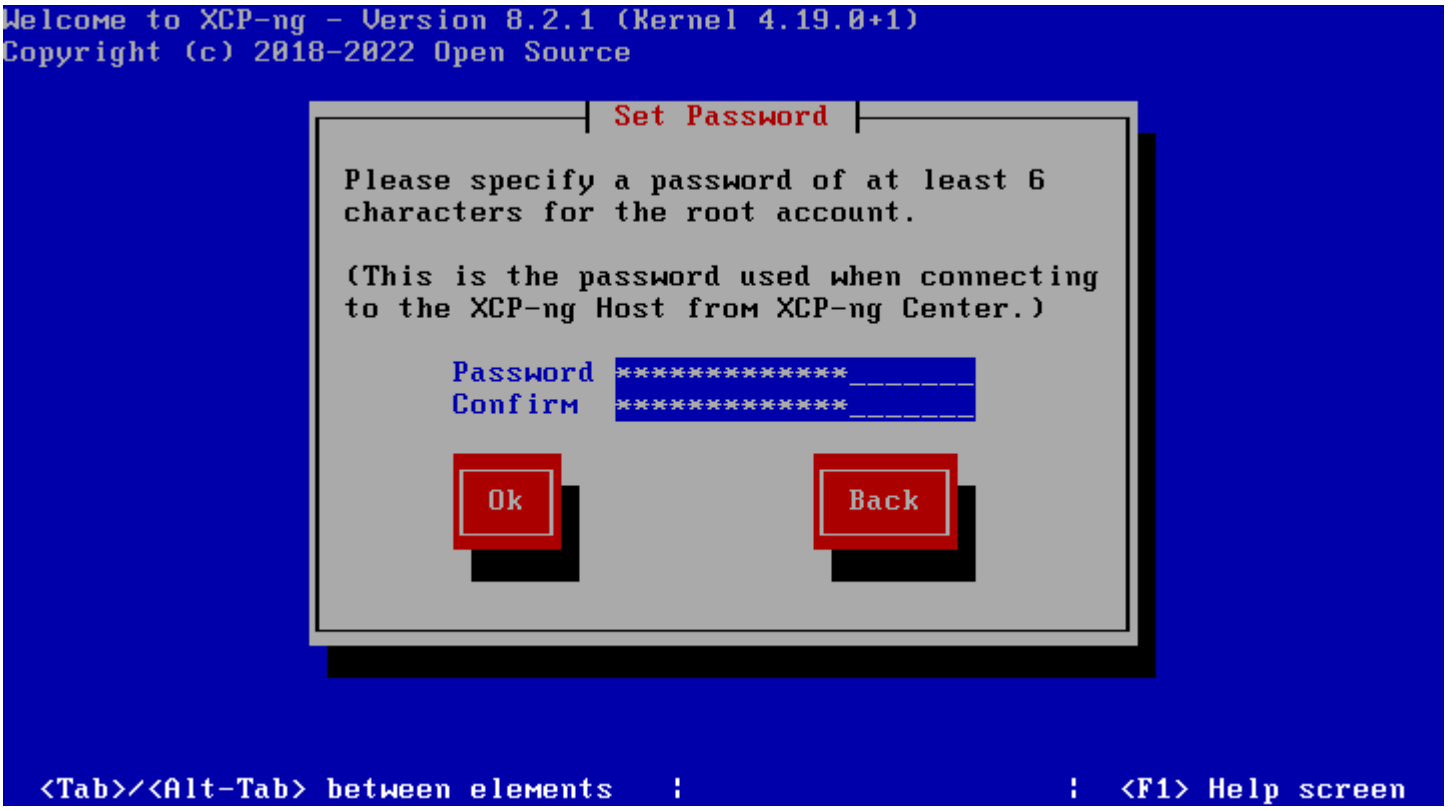
Select your installation source for your XCP NG install

You can choose to verify your media. Here I am skipping that step to save time. The media I am using is an ISO uploaded to the VMware vSphere datastore and selected in the virtual machine properties in the vSphere client.



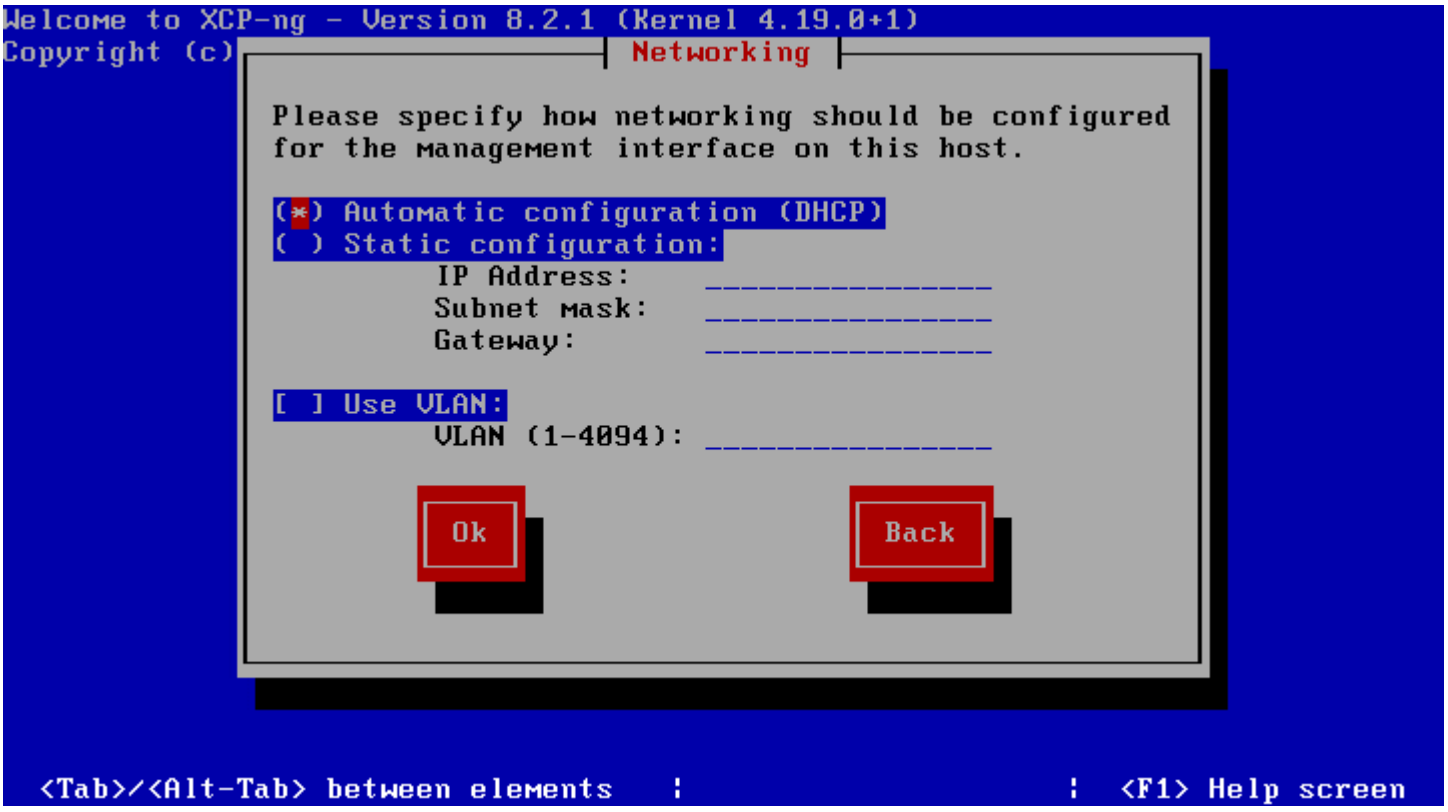
Choose media verification option

Set an administrator password for the root account.



Set your admin password for XCP NG

Choose your networking settings. You can set a static address or use DHCP. Also, you can choose the VLAN for the administrative interface also.



Configure networking for your XCP NG host

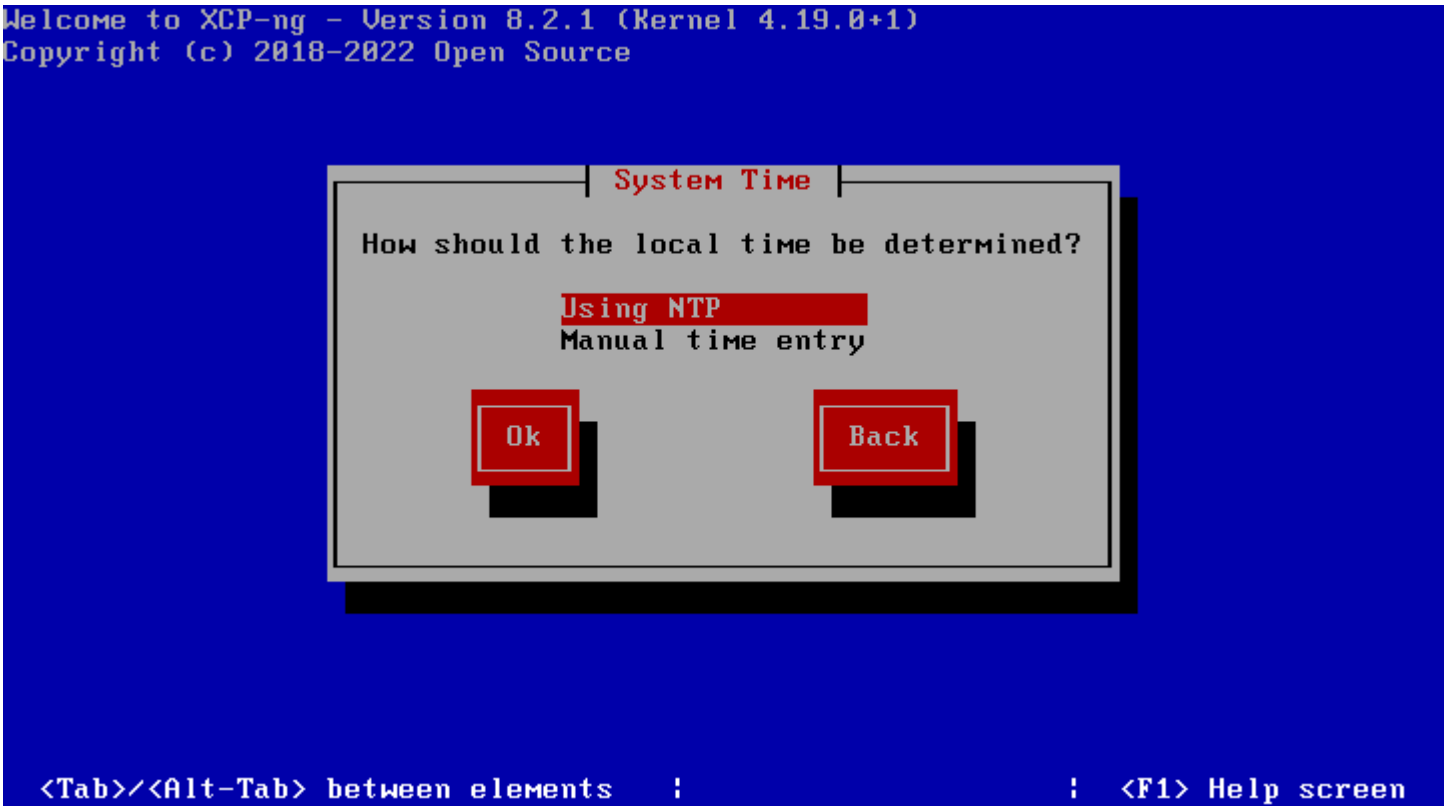
Select the geographic region for the XCP-NG server.

Choose your geographic region for XCP NG host

Set your locality for time zone purposes.

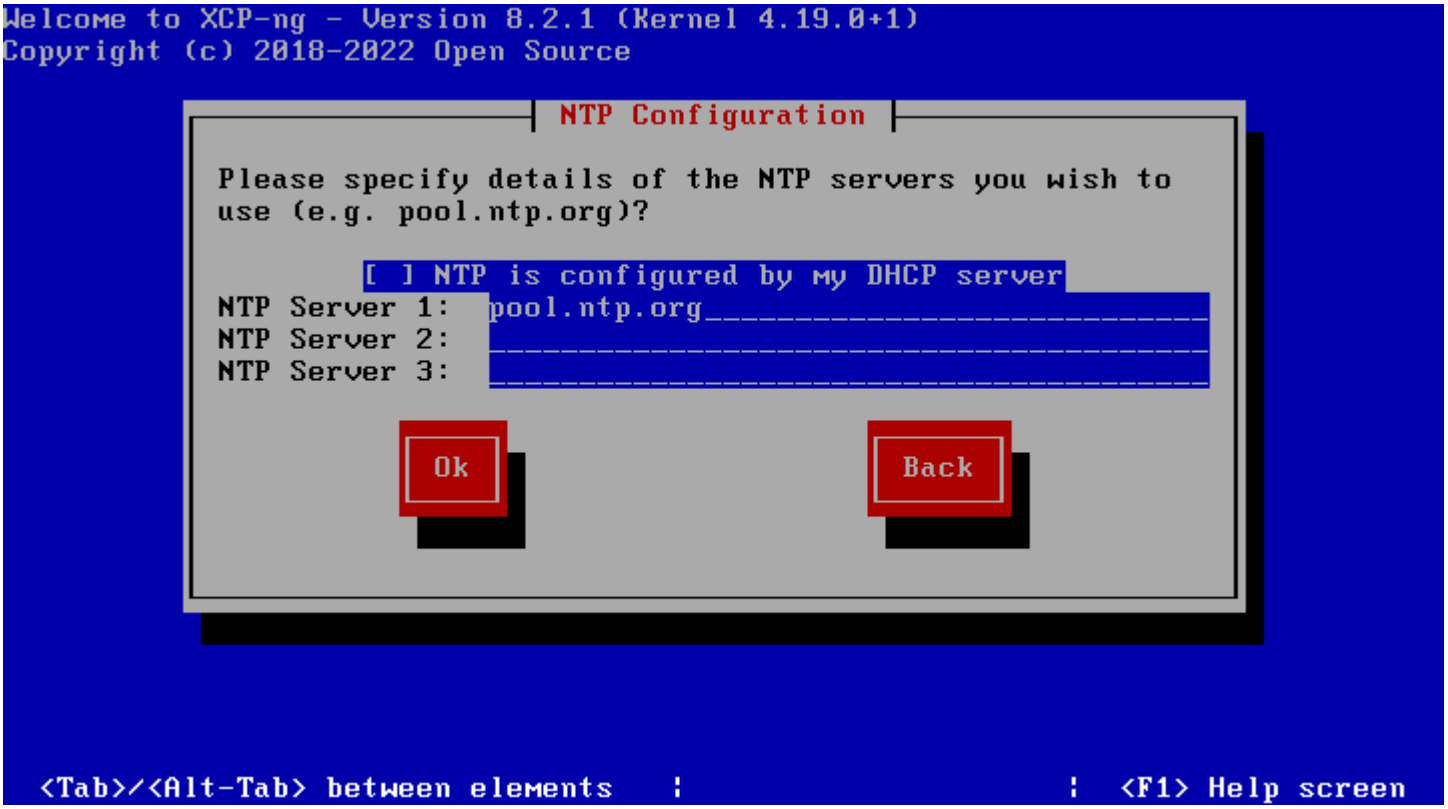
Choose your locality for time zone purposes

Choose how you want to configure the system time, using NTP or manually setting the time configuration.



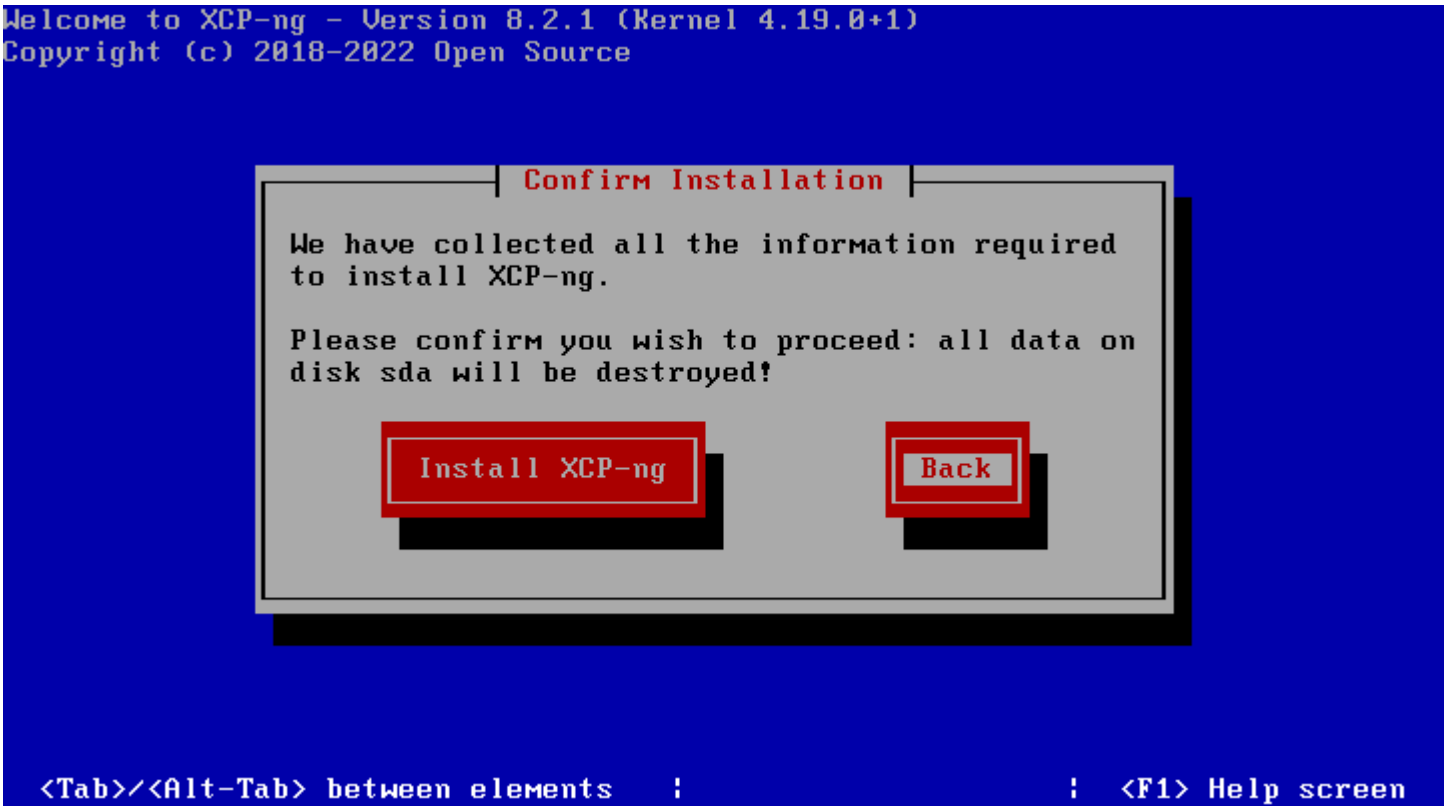
Choose how you want to set the XCP NG host system time

Configure the NTP server you want to use for time configuration.



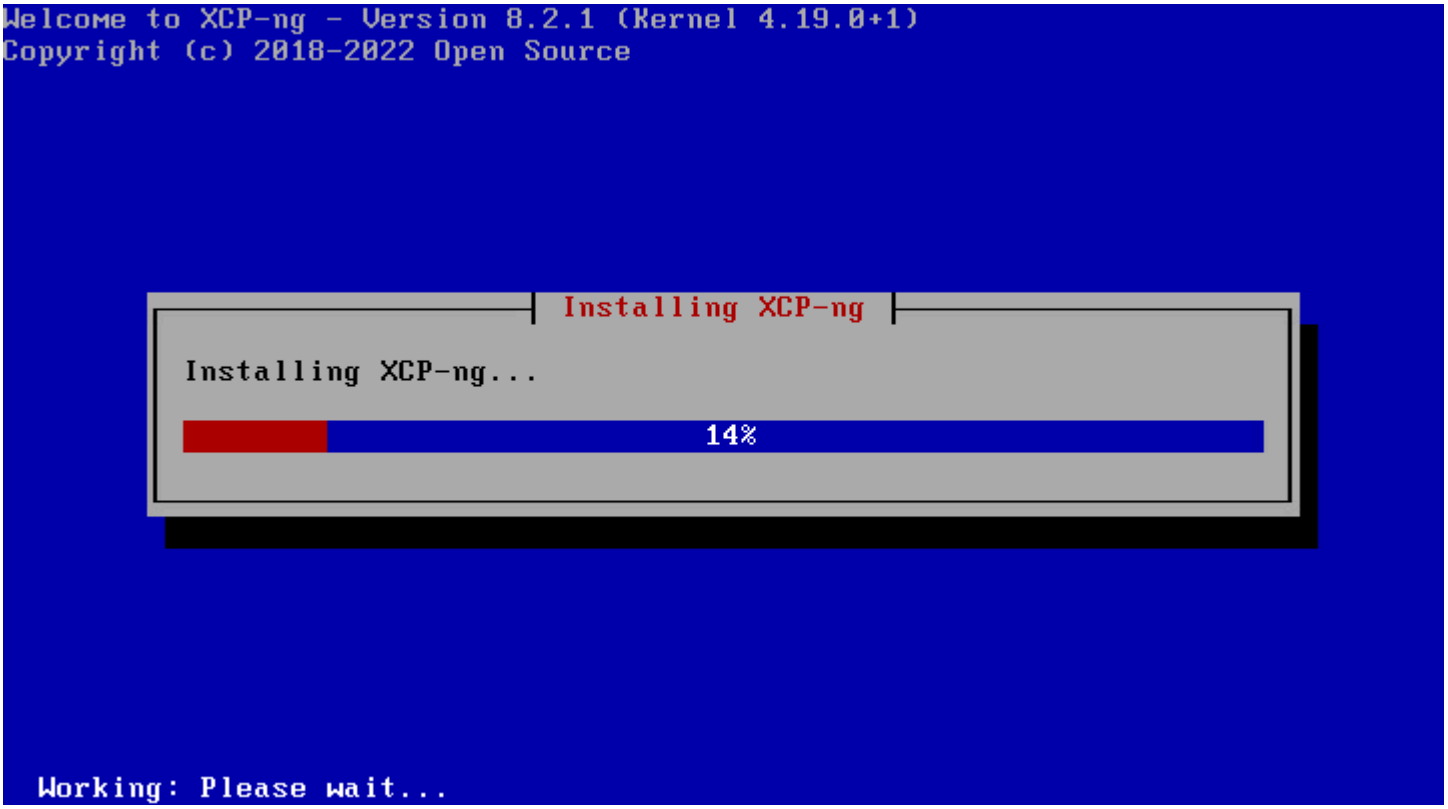
Configure your NTP servers

Choose the **install XCP-NG** button to begin the installation of XCP-NG.



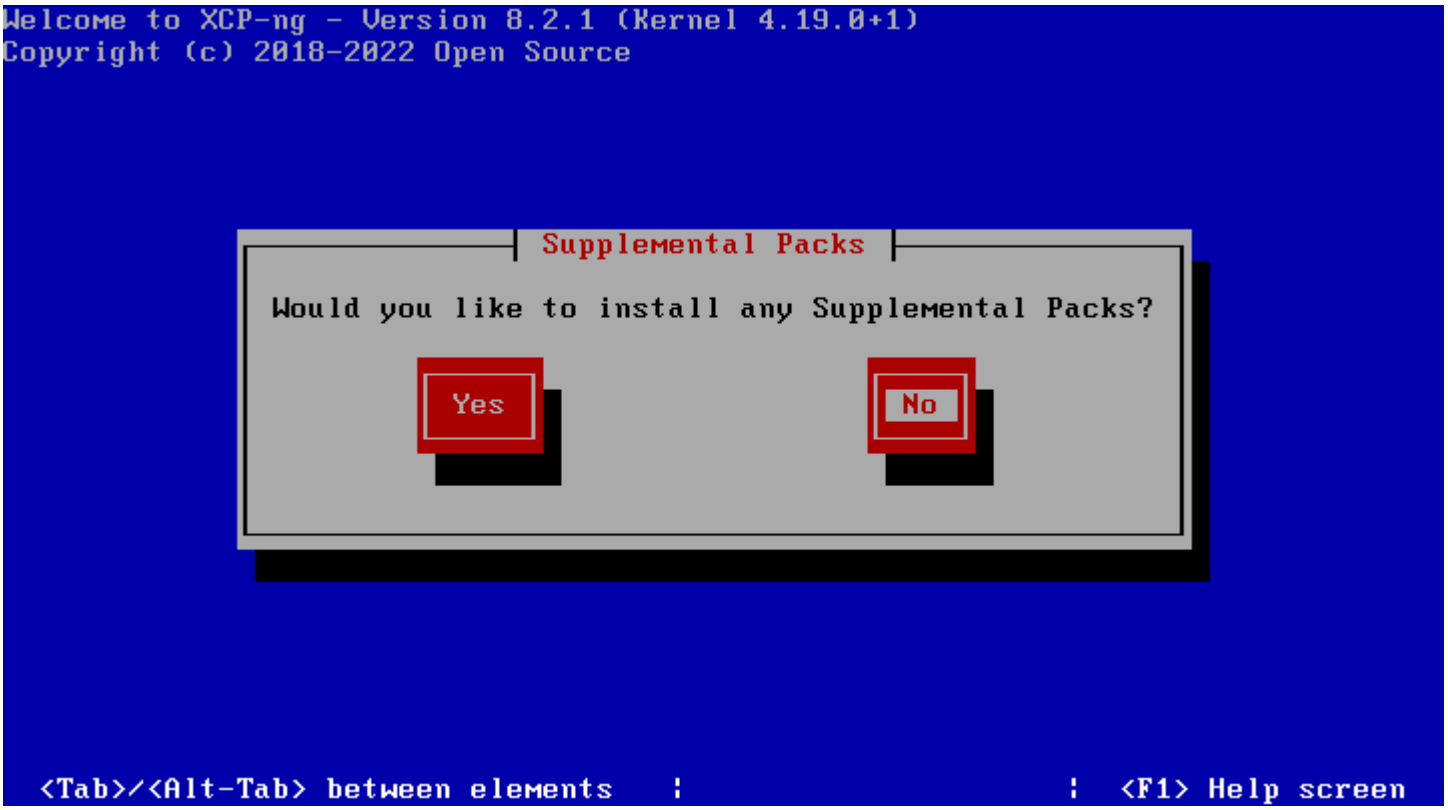
Choose to install XCP NG and begin the install with configuration options chosen

The installation of XCP-NG begins.



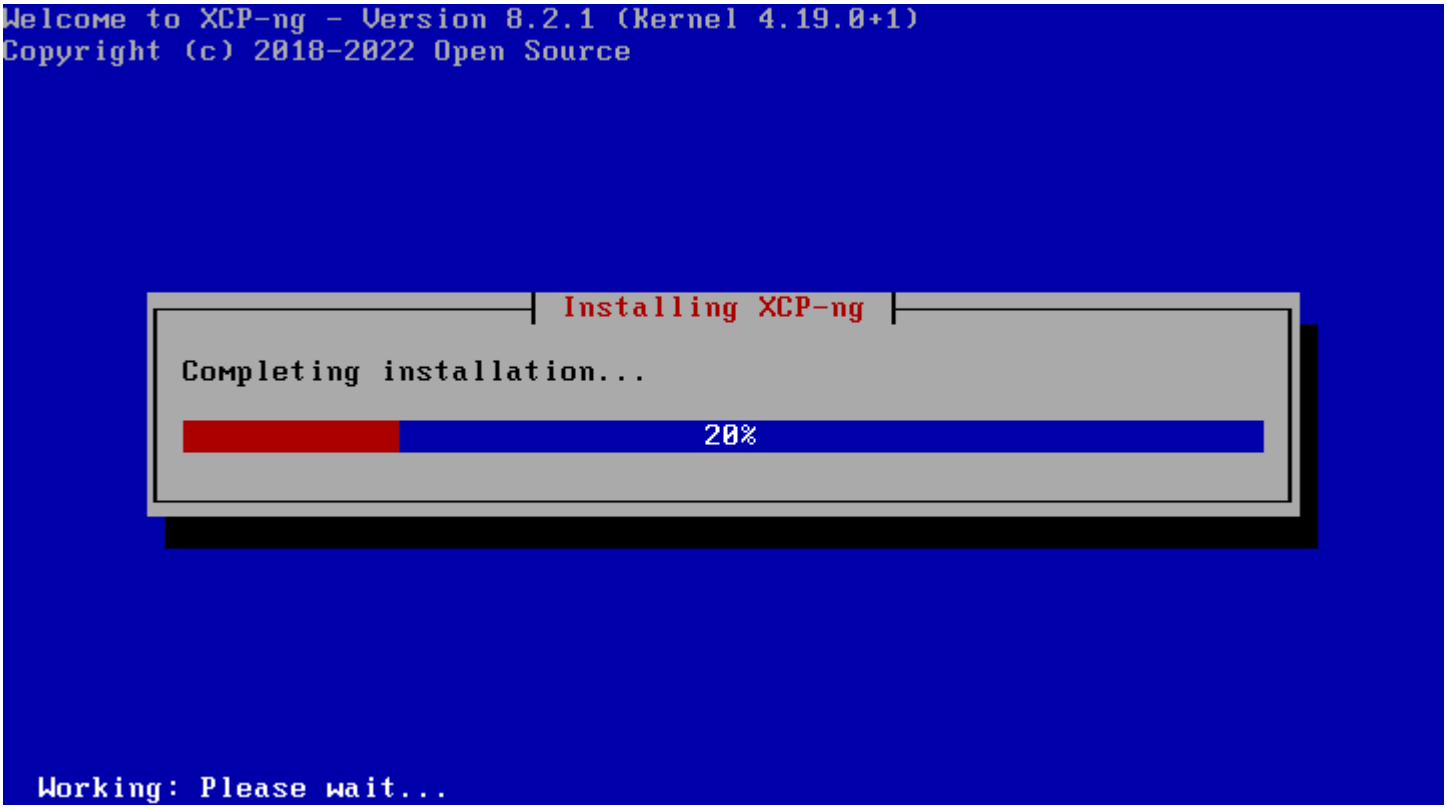
Installation of XCP NG begins

Answer the question of whether you want to install any supplemental packs in your XCP-NG installation or not.



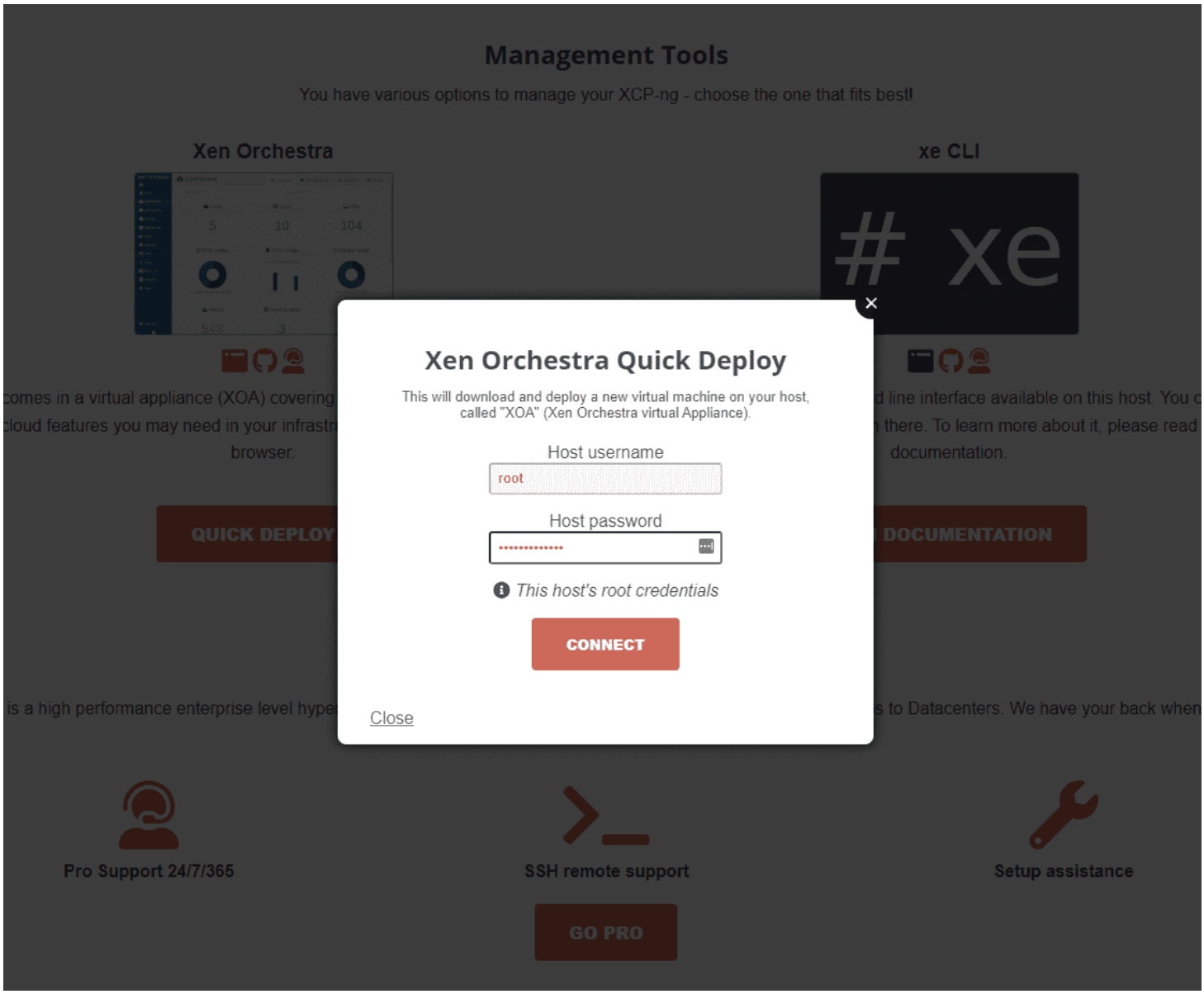
Answer question about installing supplemental packs

The installation completes and finalizes the settings.



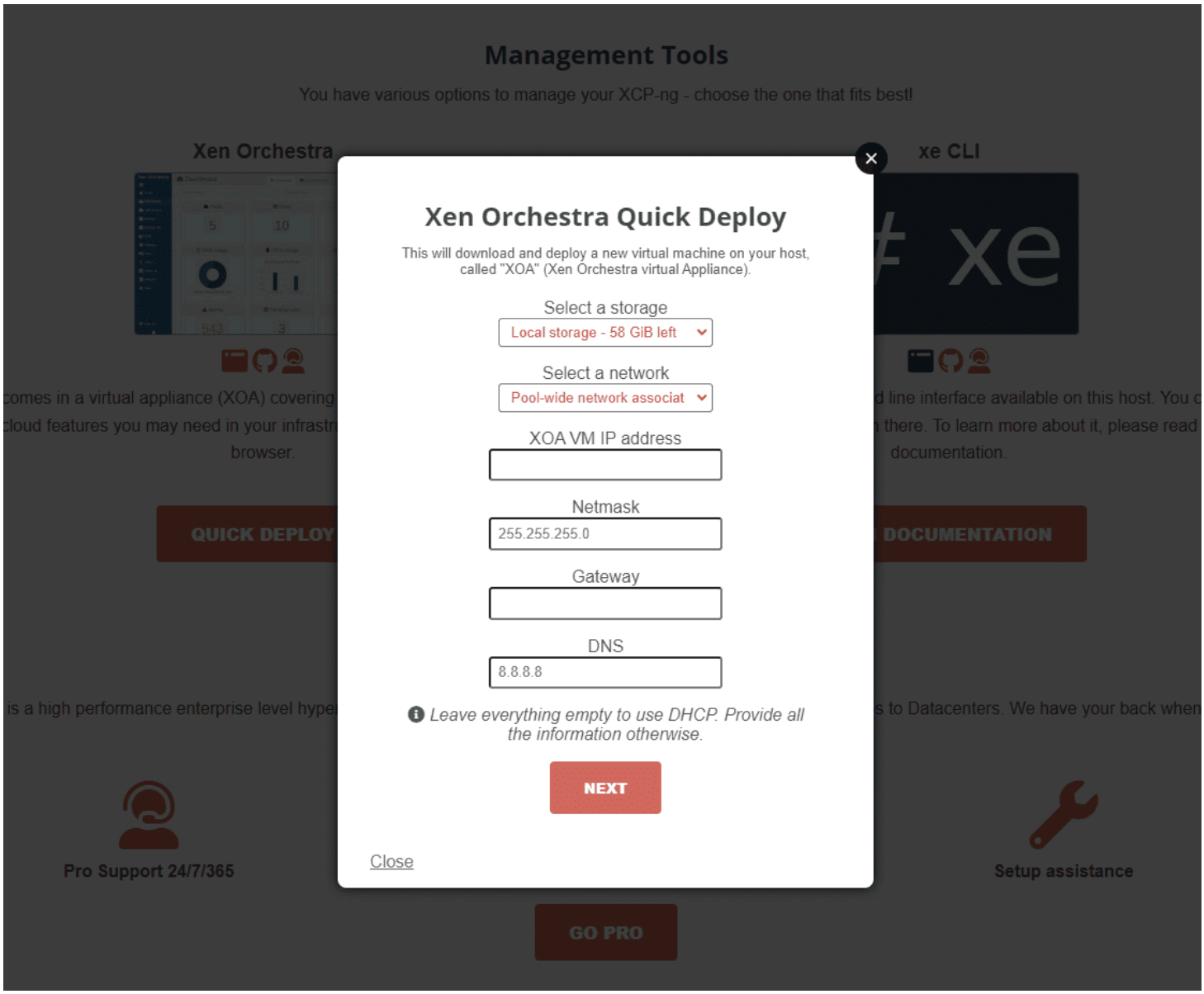
XCP NG installation begins to finalize

Once the host reboots, you can browse to the IP of your XCP-NG server and choose the **Xen Orchestra Quick Deploy** option. Enter the root password for your XCP-NG server.



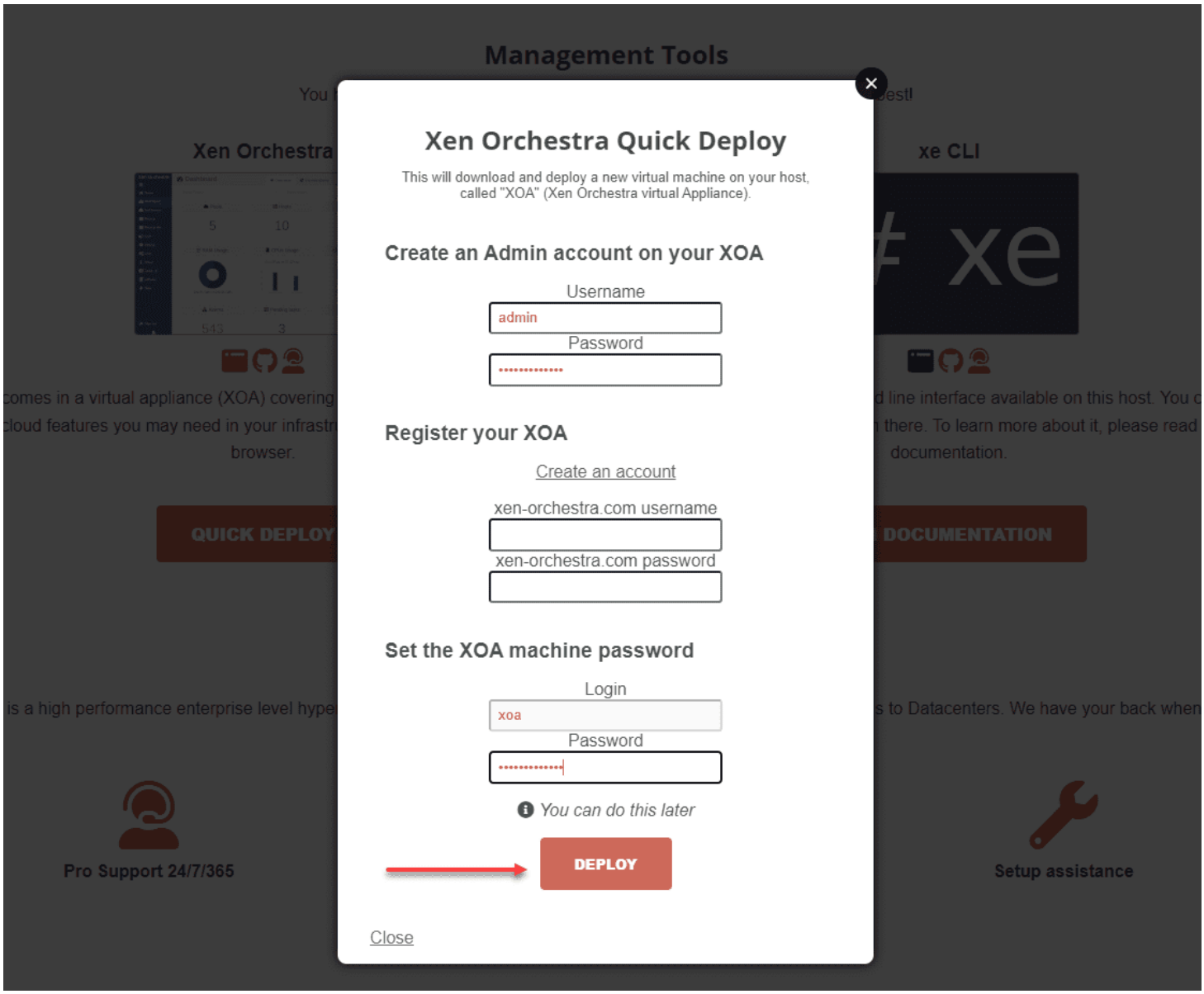
Beginning the installation of Xen Orchestra

Choose the storage and networking options for the deployed Xen Orchestra Server.



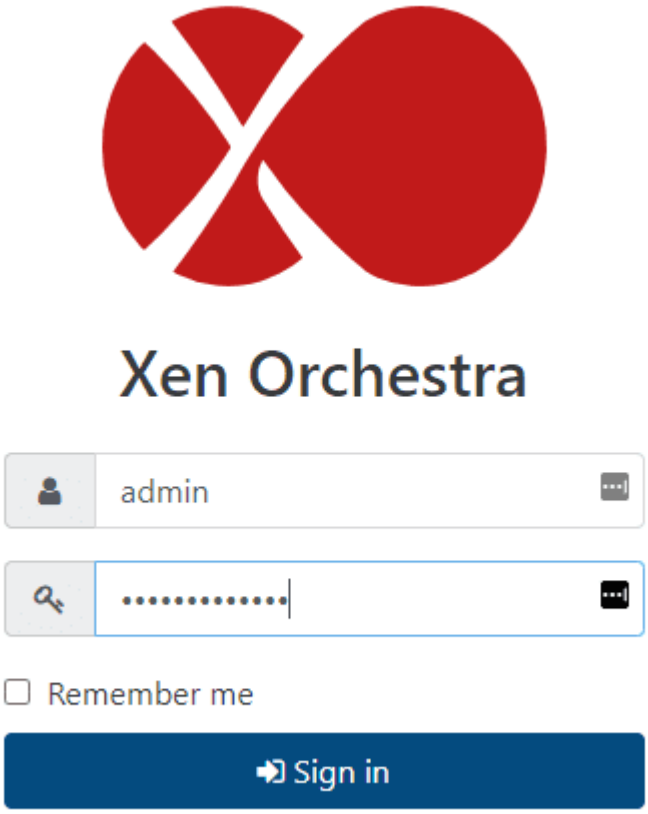
Xen Orchestra VM quick deploy storage and networking options

Set up the admin account for XOA, register the account, and set the machine password. If you choose not to register your Xen Orchestra server, note you will not be able to pull down updates for it. However, this process is free and just requires you complete the registration process. Click **Deploy**.



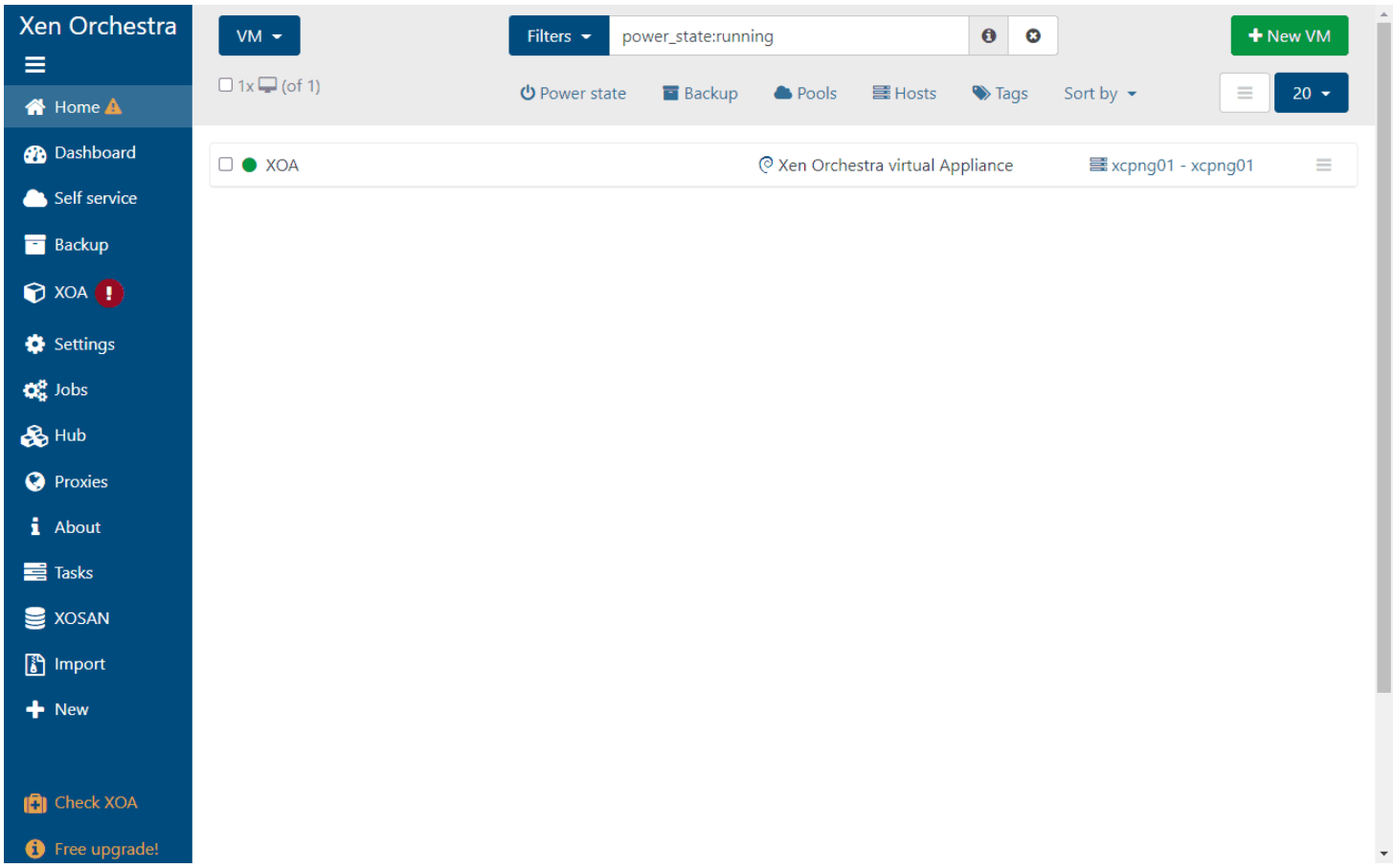
Set the admin account register Xen Orchestra and set the XOA machine account

After the XOA VM finishes deploying, browse to the IP you configured in the quick deploy process. Login with the admin password you configured.



Login to Xen Orchestra using the configured password

If you browse to the VMs section, you should see your XOA VM running in your Xen Orchestra inventory.



Viewing running virtual machines using Xen Orchestra

XCP-NG FAQs

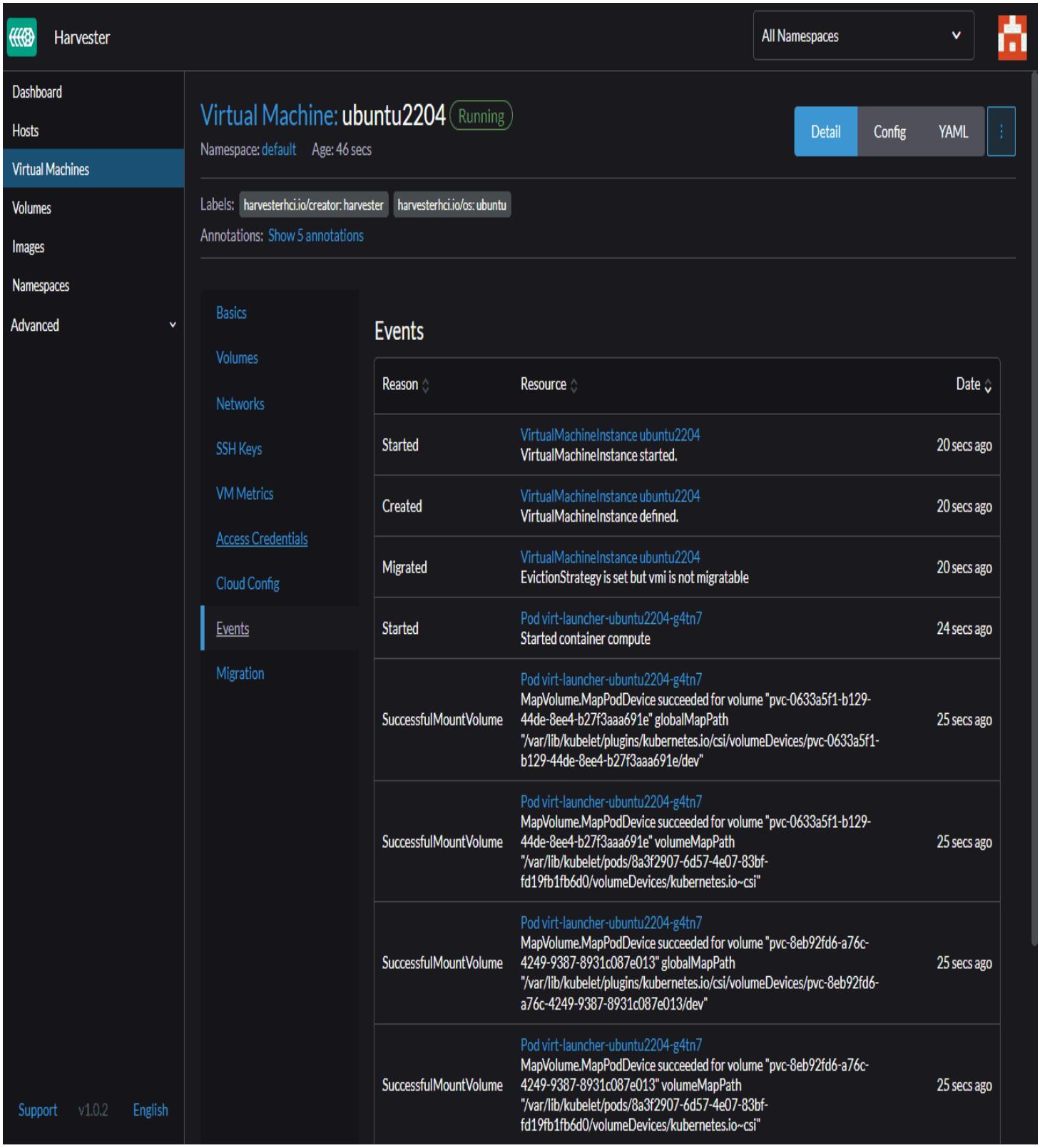
- **What is XCP-NG?** XCP-NG is an open-source hypervisor that allows easily spinning up workloads using the solution with a nice UI, live migration capabilities, and other features. It is free to download, along with the management platform – Xen Orchestra.
- **What is Xen Orchestra?** Xen Orchestra is a free, open-source management platform for your XCP-NG servers. It provides many management capabilities from a modern web UI and allows configuring VMs, hosts, and even backups for your XCP-NG environment.
- **How do you install XCP-NG in VMware?** As we have shown in the post, it is fairly easy. You just need to create a new VM with nested virtualization (Exposed hardware-assisted virtualization) enabled, mount the XCP-NG ISO, and begin the installation. There are several screens to configure your way through, but all-in-all, it is straightforward to get XCP-NG along with Xen Orchestra up and running.

Final Notes

Coming from VMware vSphere with vCenter Server, it is amazing how powerful vSphere is when compared to open-source solutions. However, many of these solutions are powerful in their own right, due to their open-source nature, full features, and the ability to quickly stand up a hypervisor to start running workloads. XCP-NG and Xen Orchestra are compelling solutions that can certainly be used in many use cases. As shown, you can easily get up and running with XCP-NG in VMware environments for labbing purposes and familiarizing yourself with other solutions out there.

Install Harvester in VMware ESXi

June 16, 2022
Kubernetes



Viewing events for the Harvester virtual machine

I have been playing around with Harvester by Rancher which is a pretty cool project that combines Kubernetes with Virtual Machines. For those that want to play around with Harvester and may not have a physical workstation to play around with, but you have your VMware ESXi lab, you can install Harvester inside of VMware ESXi as a nested hypervisor with nested

virtualization enabled. It is pretty easy and I will walk you through the steps to get this done. Let's look at installing harvester in VMware ESXi and see how you can setup a Harvester lab.

What is Harvester?

I just finished writing a pretty detailed blog post covering what Harvester is exactly. You can read that blog post here:

- Rancher Kubevirt with Harvester Virtual Machines with Kubernetes – Virtualization Howto

However, in brief, it is an open-source solution from Rancher that provides an HCI solution that combines the capabilities of running virtual machines and containers in the same platform. As you can imagine, since it is made by Rancher, you can integrate the solution in Rancher to have a cohesive platform to run VMs and containers. So, it is a pretty cool solution. I have written about Rancher quite a few times in the posts below:

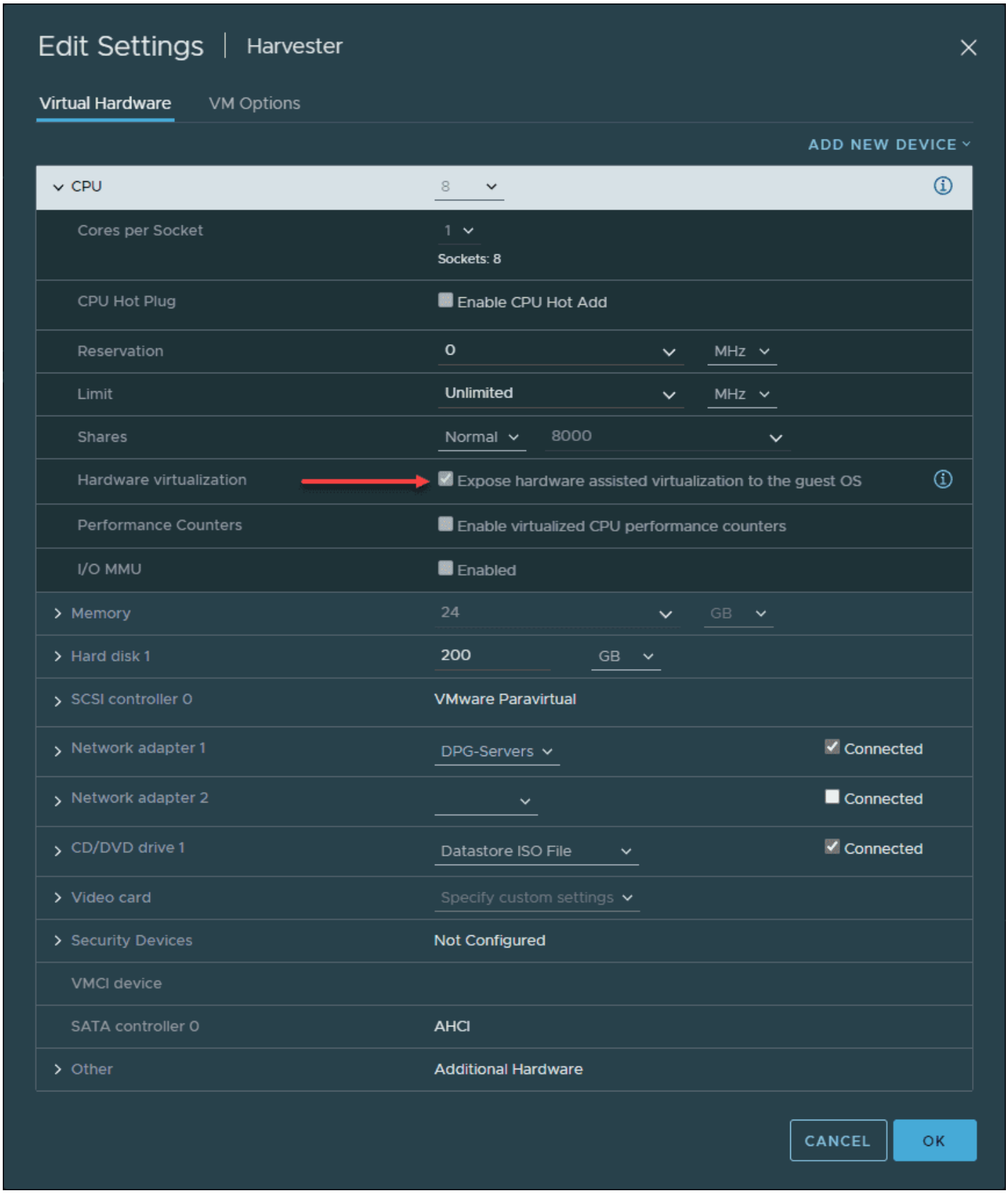
- Install Longhorn Rancher for Kubernetes Block Storage
- Rancher Node Template VMware ESXi – Ubuntu Cloud Image
- Rancher Desktop v1.0 Features Installation and Configuration
- Create Kubernetes Cluster with Rancher and VMware vSphere

Install Harvester in VMware ESXi

The process to install Harvester in VMware ESXi is fairly straightforward and aligns with installing any other nested hypervisor installation inside an ESXi virtual machine. You need to enable a couple of things to make sure nested virtualization works with an ESXi VM, including:

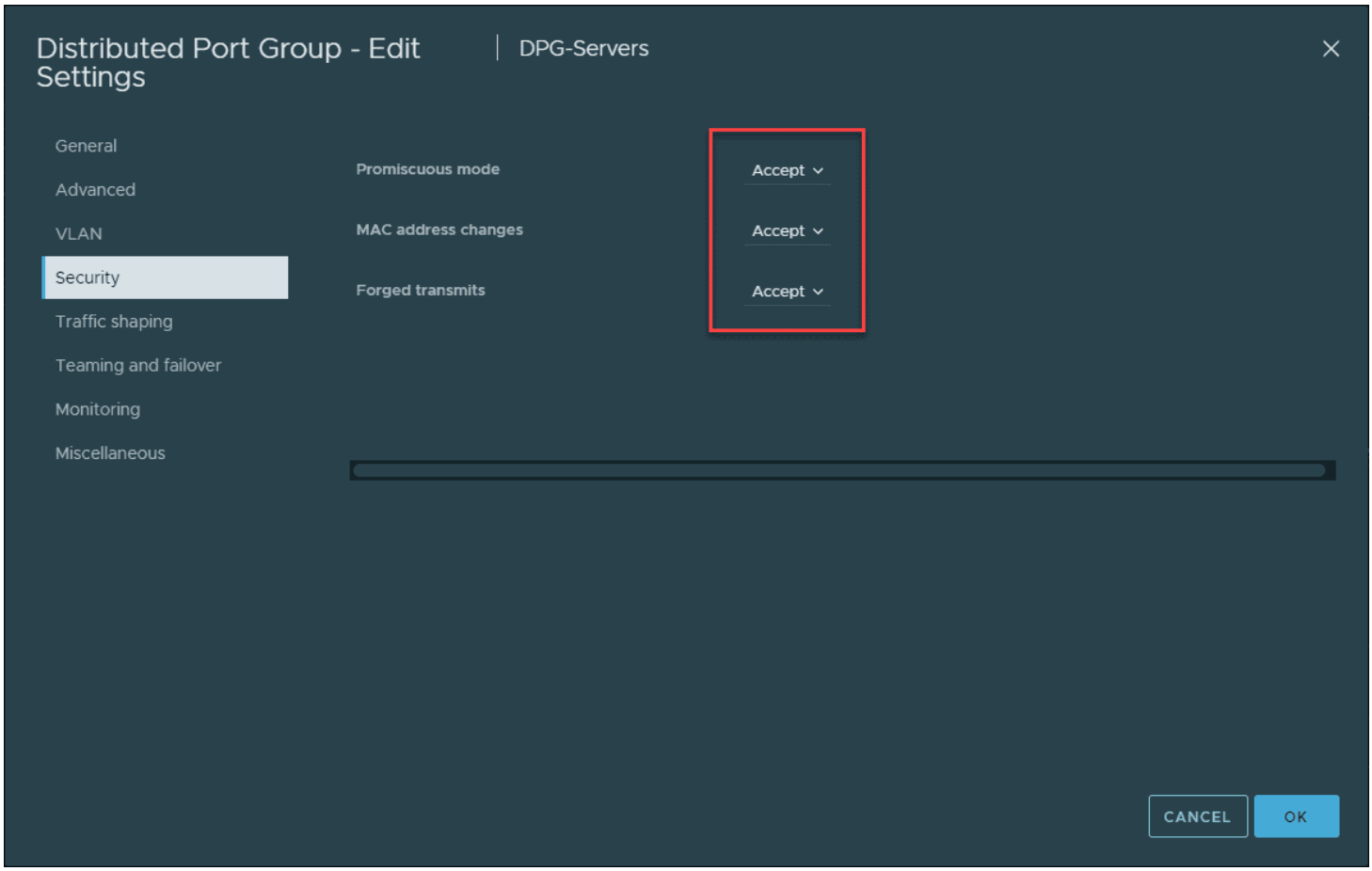
- Expose hardware assisted virtualization to the guest OS
- Change security settings for your vSphere Standard or vSphere Distributed Switch

Below, you can see the details of the CPU settings I have configured for the Harvester ESXi VM. Place a check next to the **Expose hardware assisted virtualization to the guest OS**.



Expose hardware assisted virtualization to the guest OS

Below is an example of how you can change the security permissions for your nested virtualization VM. Under the **Security** option for the vSwitch, you can change the settings for promiscuous mode, MAC address changes, and Forged transmits to **Accept**.



Edit the security settings for the vSphere standard or vSphere Distributed Switch

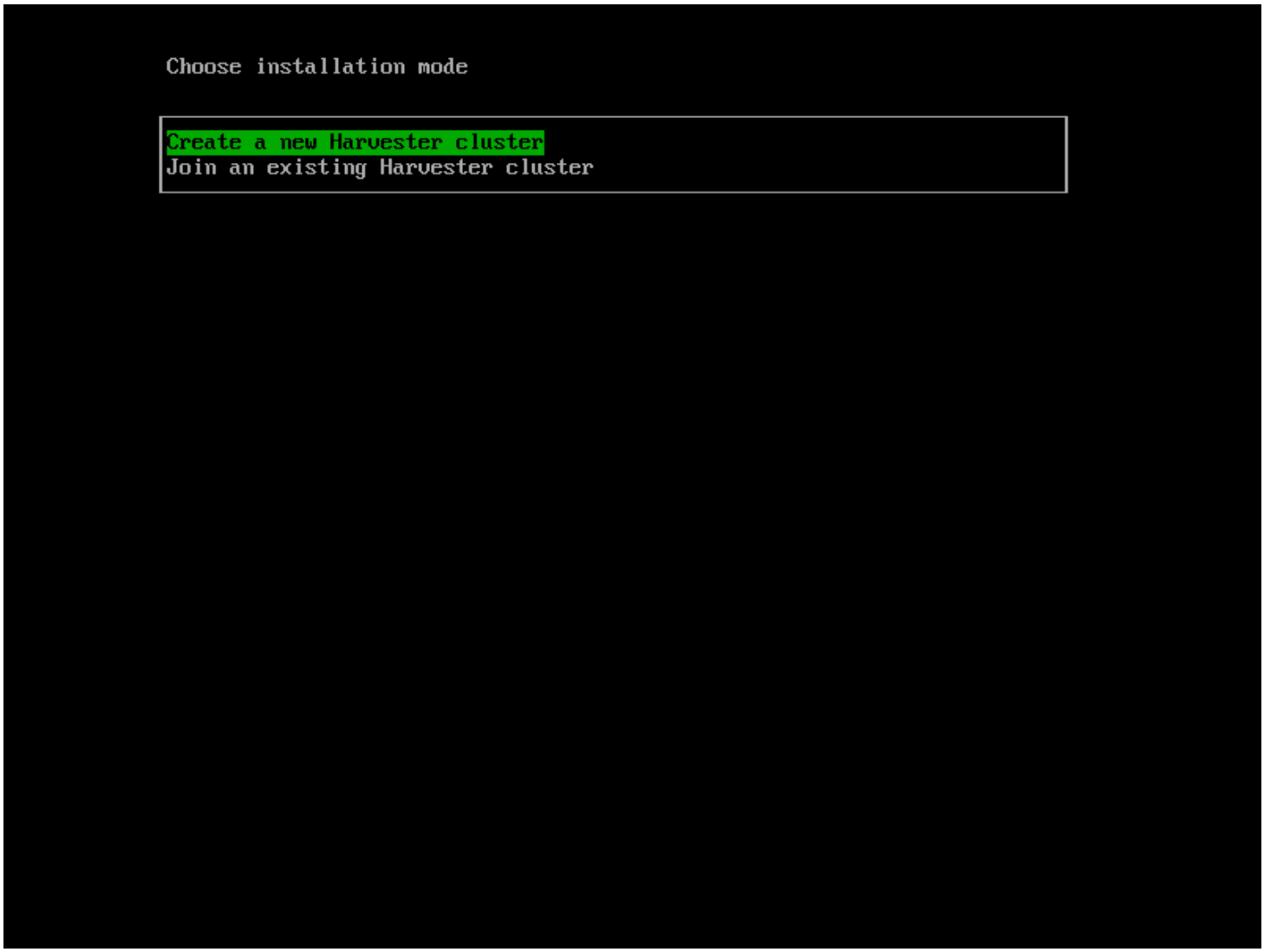
Once you have the virtual machine configured, you will need to have the Harvester ISO mounted to the CD ROM drive in the ESXi VM as well. You can download the Harvester ISO from here:

- Releases · harvester/harvester (github.com)

Boot the VM from the Harvester installation ISO, and begin the installation.

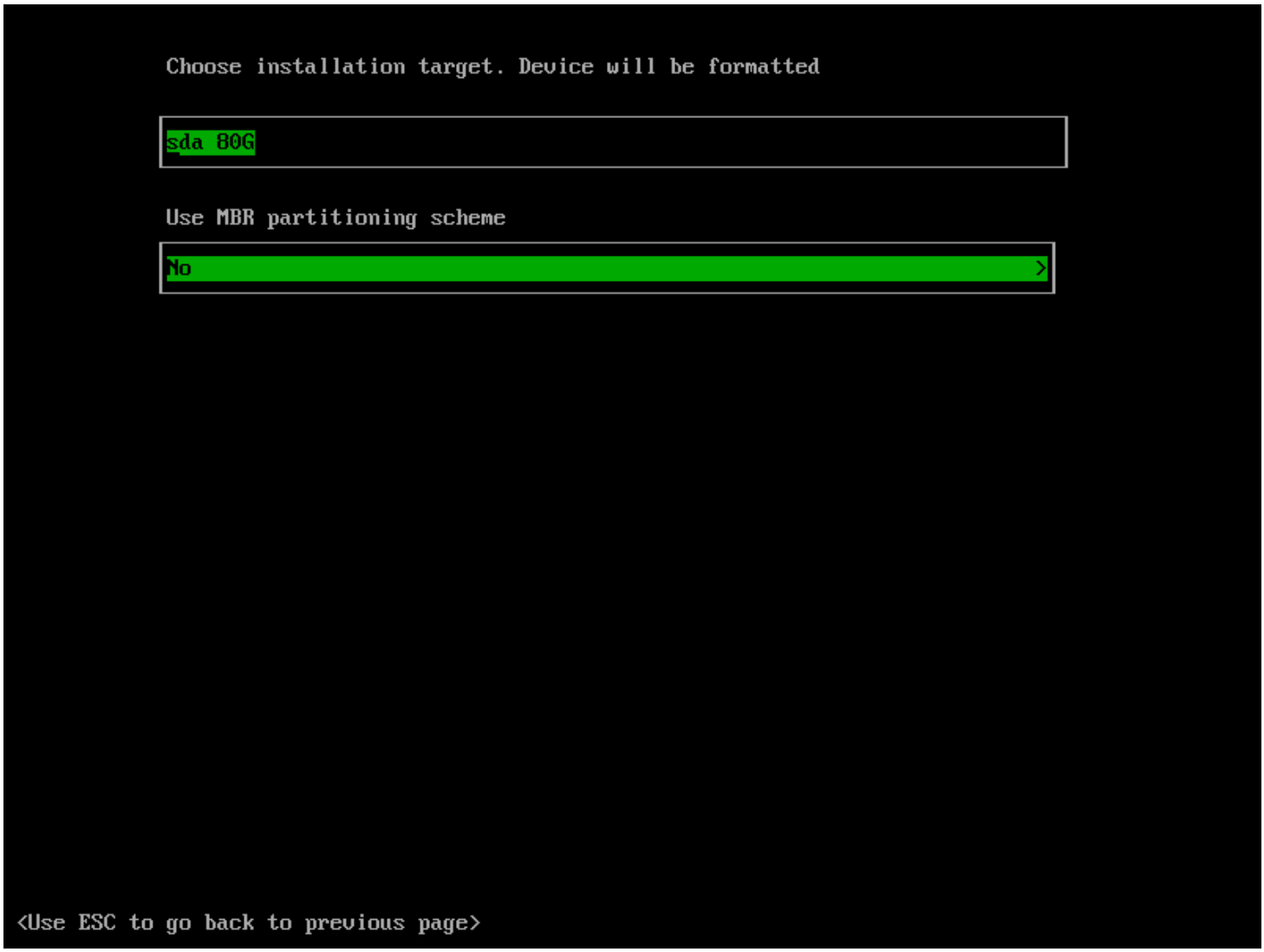
Starting the installation of Harvester in an VMware ESXi virtual machine

Here we choose to Create a new Harvester cluster. Even though, like me, you may just be installing a single node, it allows establishing the virtual IP address (VIP) and other configurations needed to expand the cluster in the future.



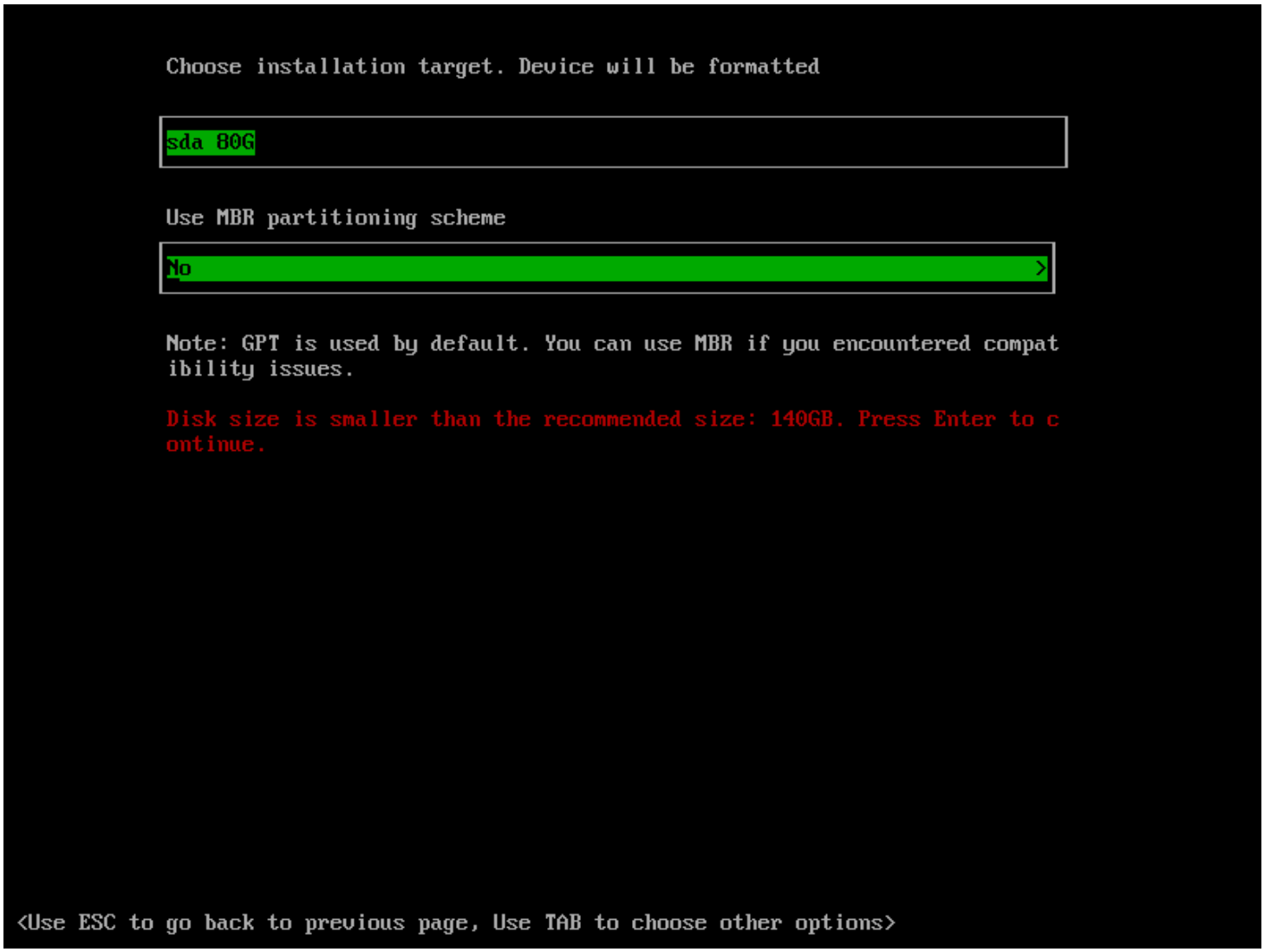
Create a new Harvester Cluster

On the installation target, you will be able to select the disk you want to install Harvester on and the partitioning scheme.



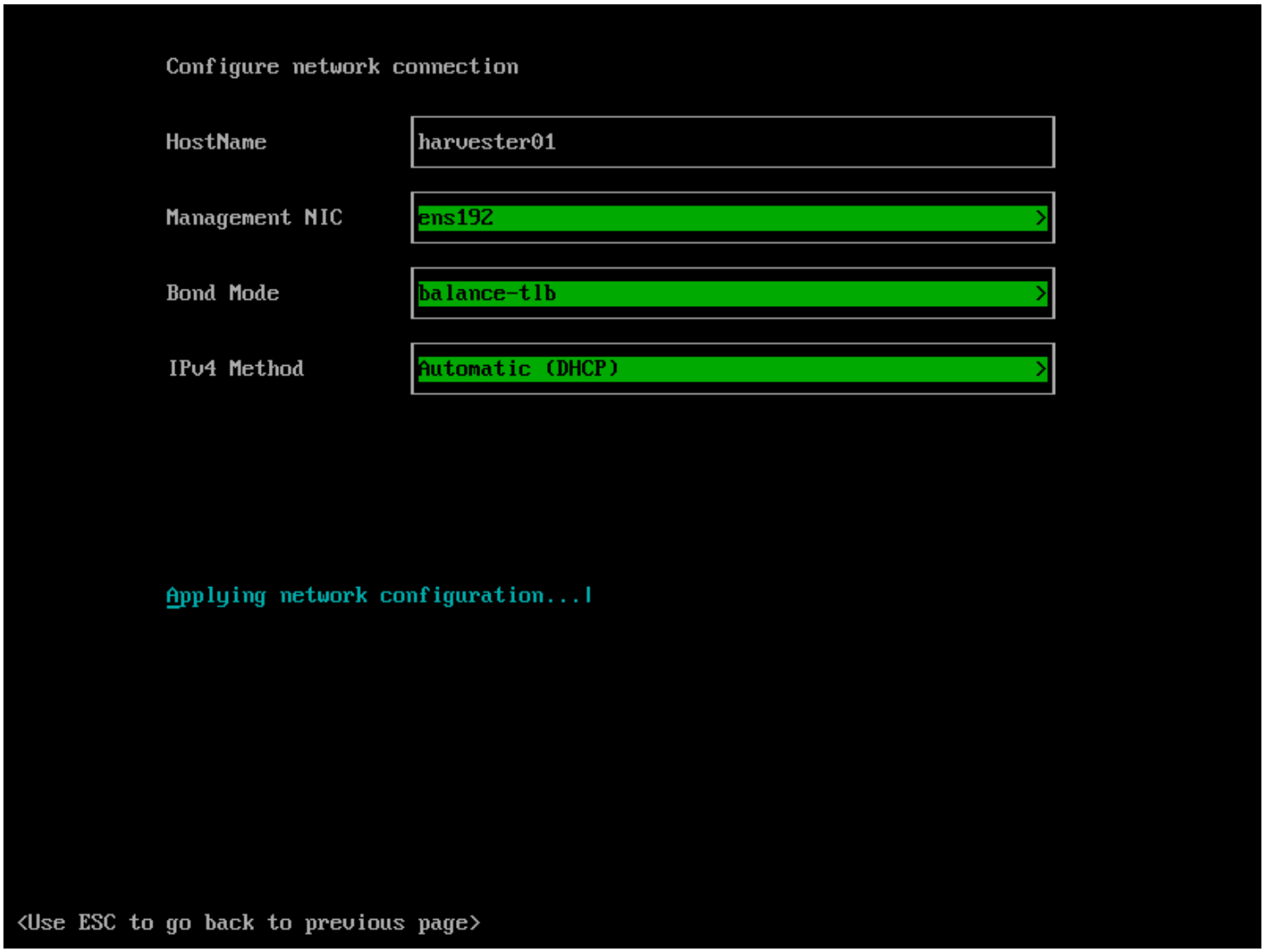
Choose the installation target

I wanted to post a screenshot on this. When I was grabbing the screens for the installation, I had only configured the VM with an 80 gig disk. After seeing this error, I went ahead and installed the node. However, when trying to configure a VM, I ran into storage issues. So, I reconfigured it with a 200 GB thin provisioned disk to give more headroom. Just beware of these requirements.



Error about disk size in Harvester

Set the hostname, Management NIC, Bond Mode, and IPv4 method.

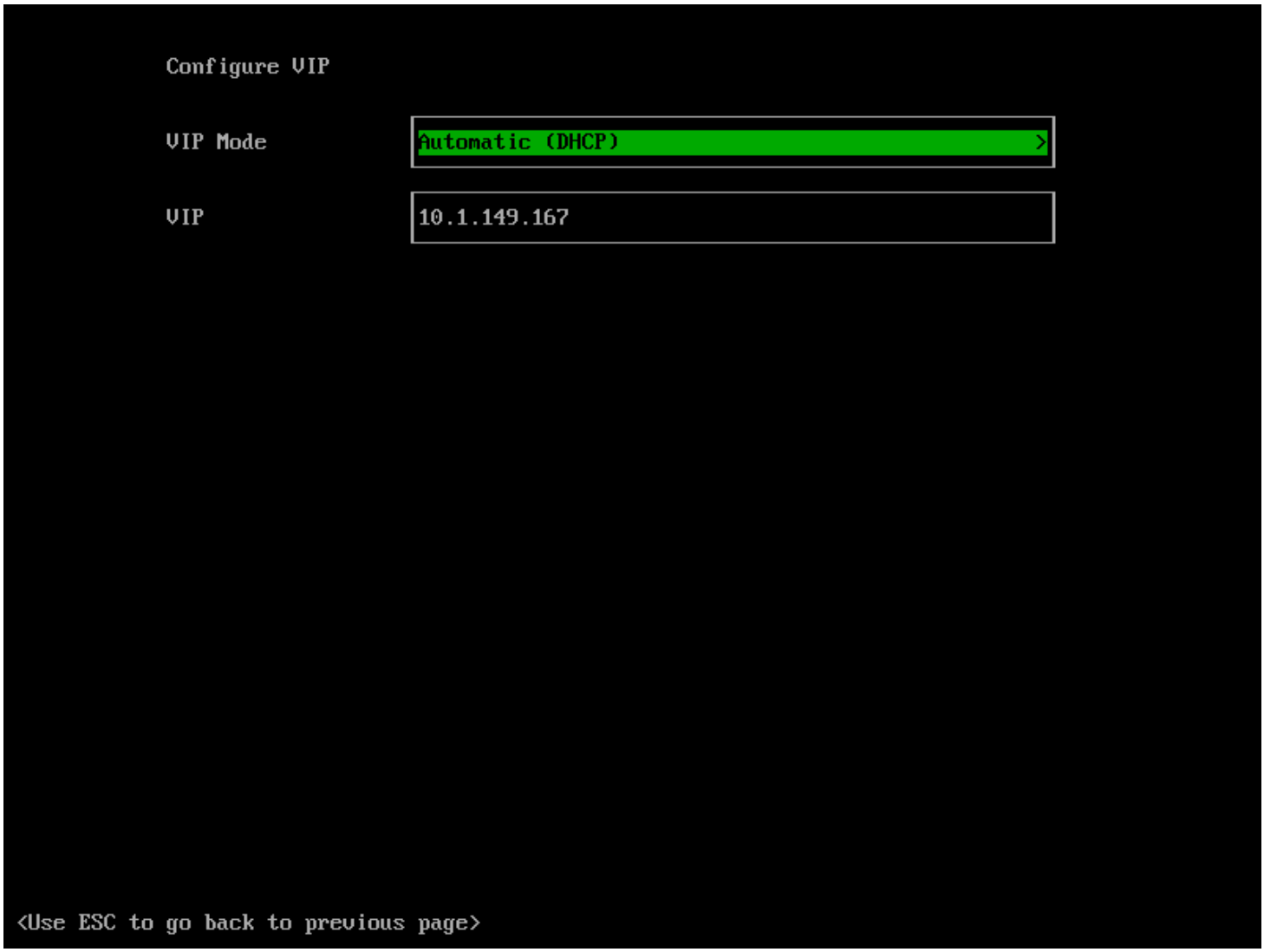


Setting management NIC bond mode and IPv4 configuration

Configure the DNS servers you want to use.

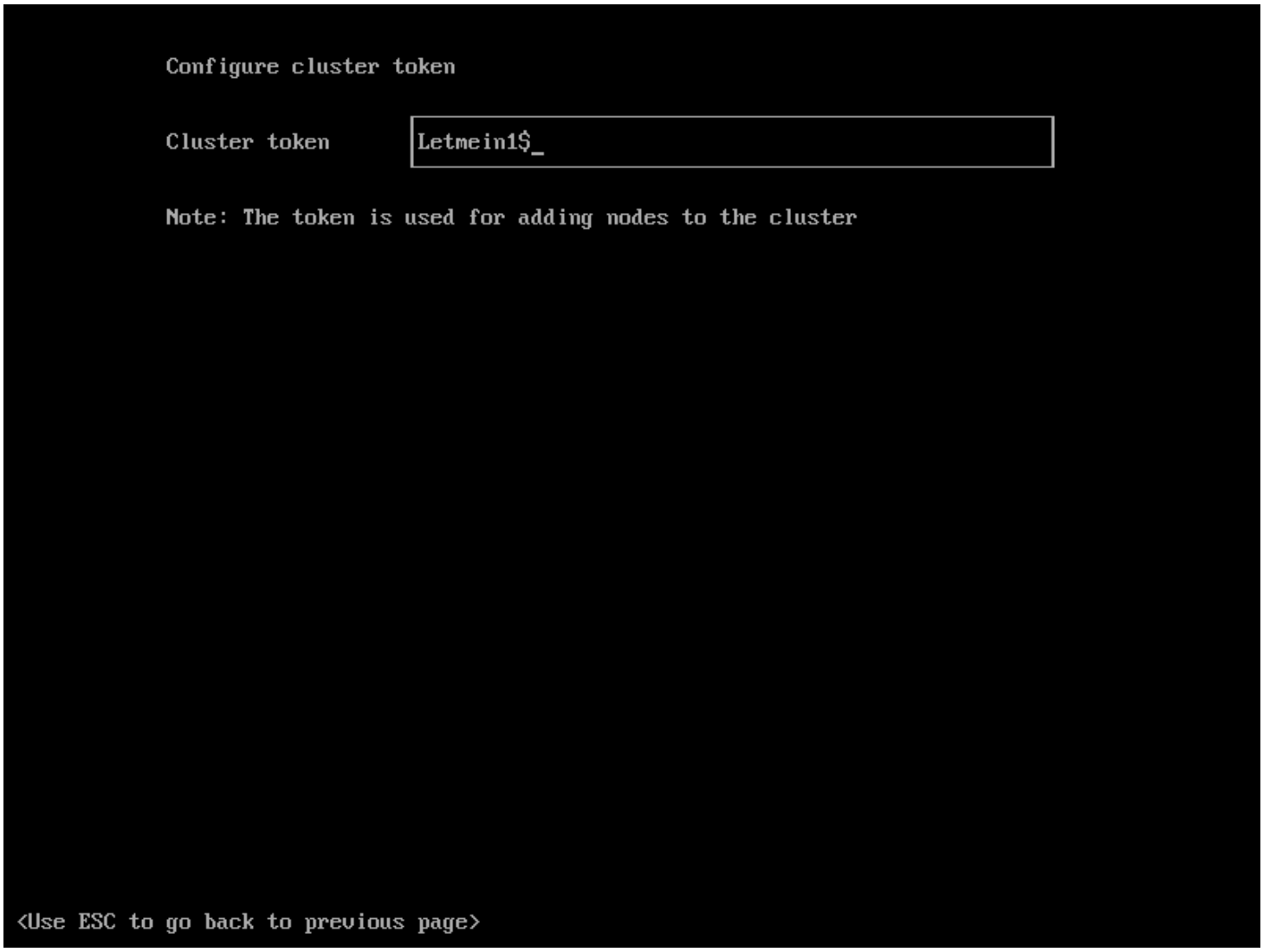
Configure DNS servers for the Harvester installation

Configure the virtual IP address (VIP). This is the IP address that will be assumed by the Harvester cluster.



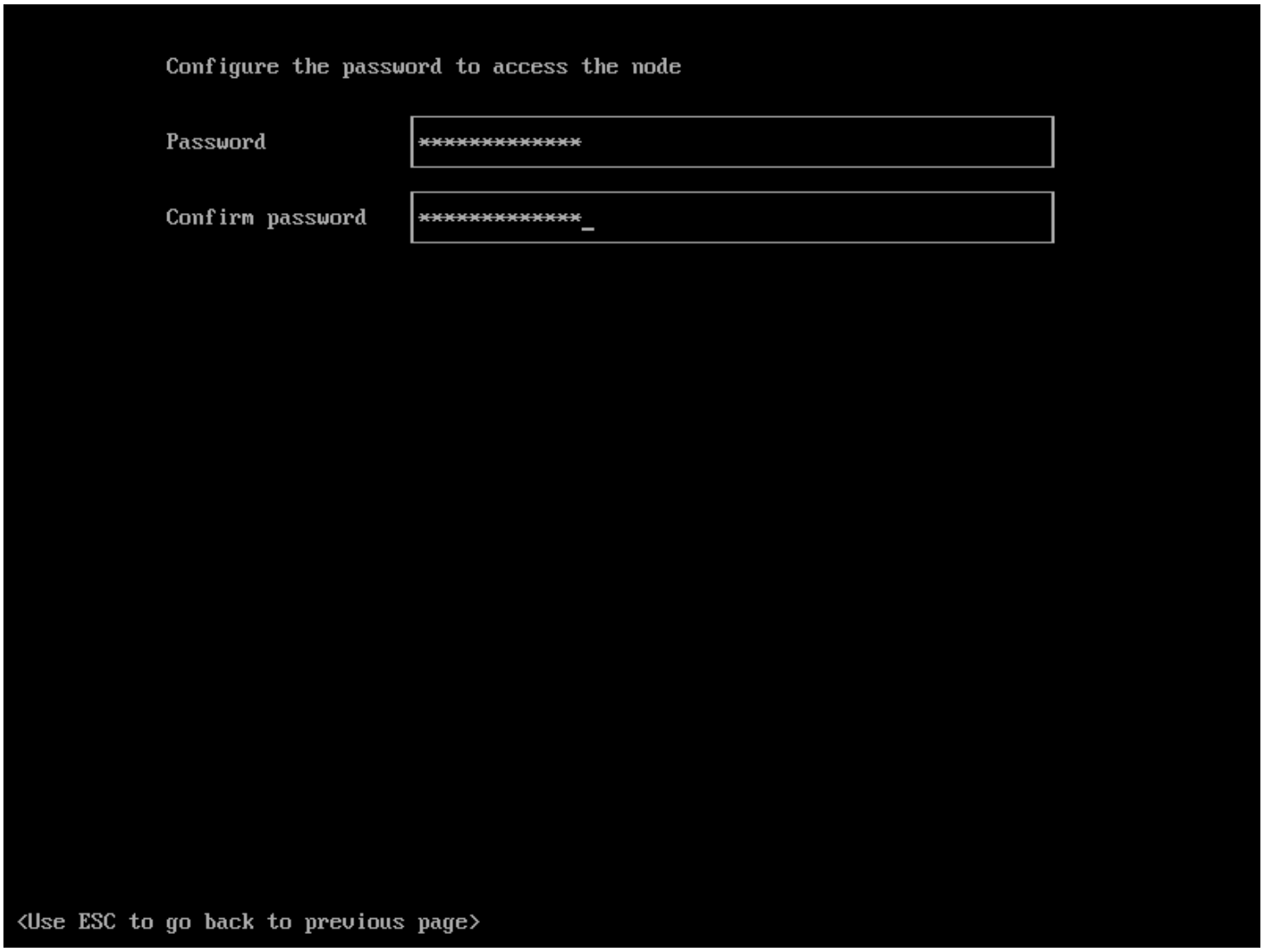
Configure the virtual IP address for the Harvester cluster

Next, you will be asked to configure a cluster token. This is the password that allows joining additional Harvester nodes to the Harvester cluster.



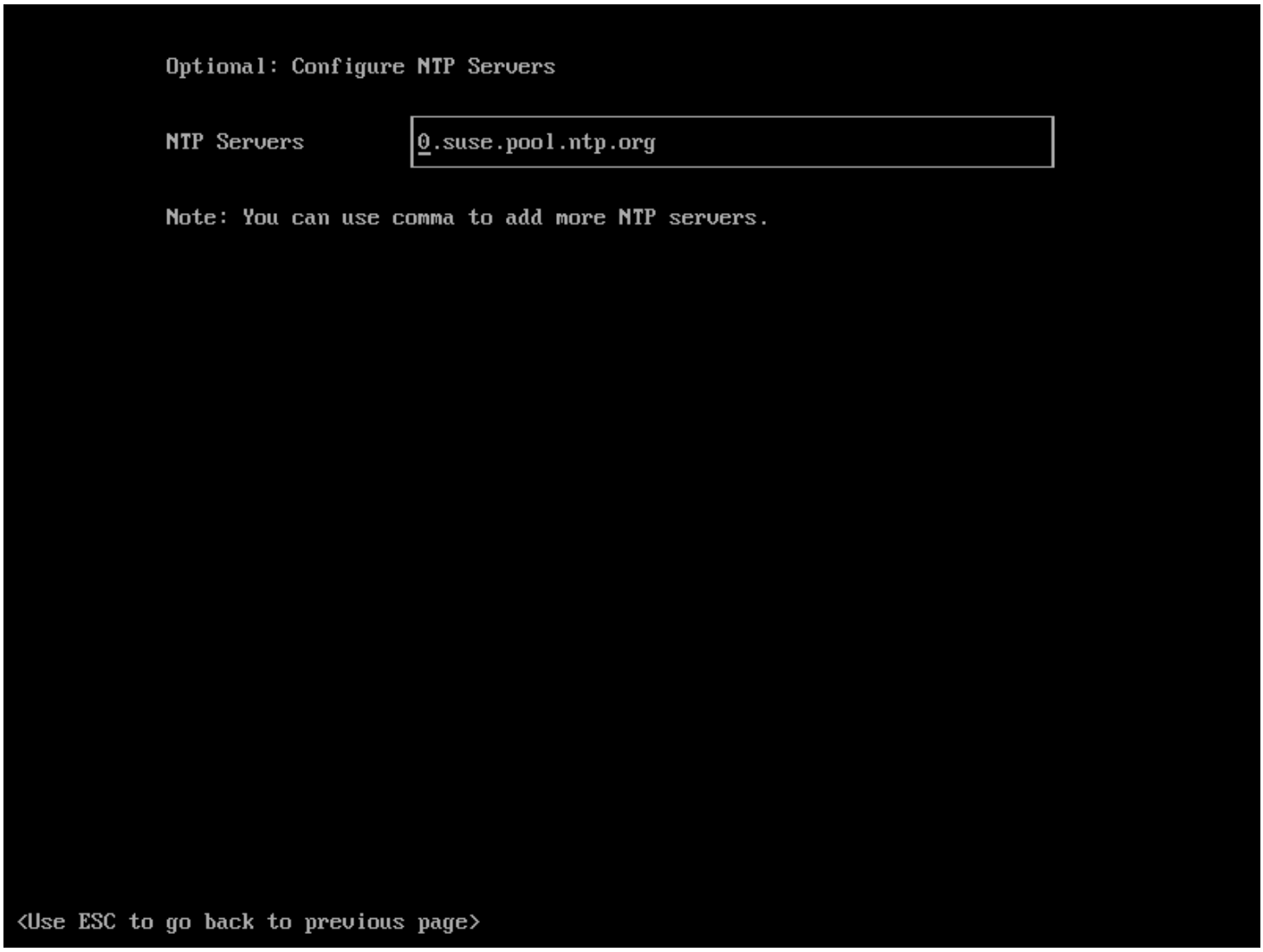
Configure your cluster token for Harvester

Configure your password to access the Harvester node.



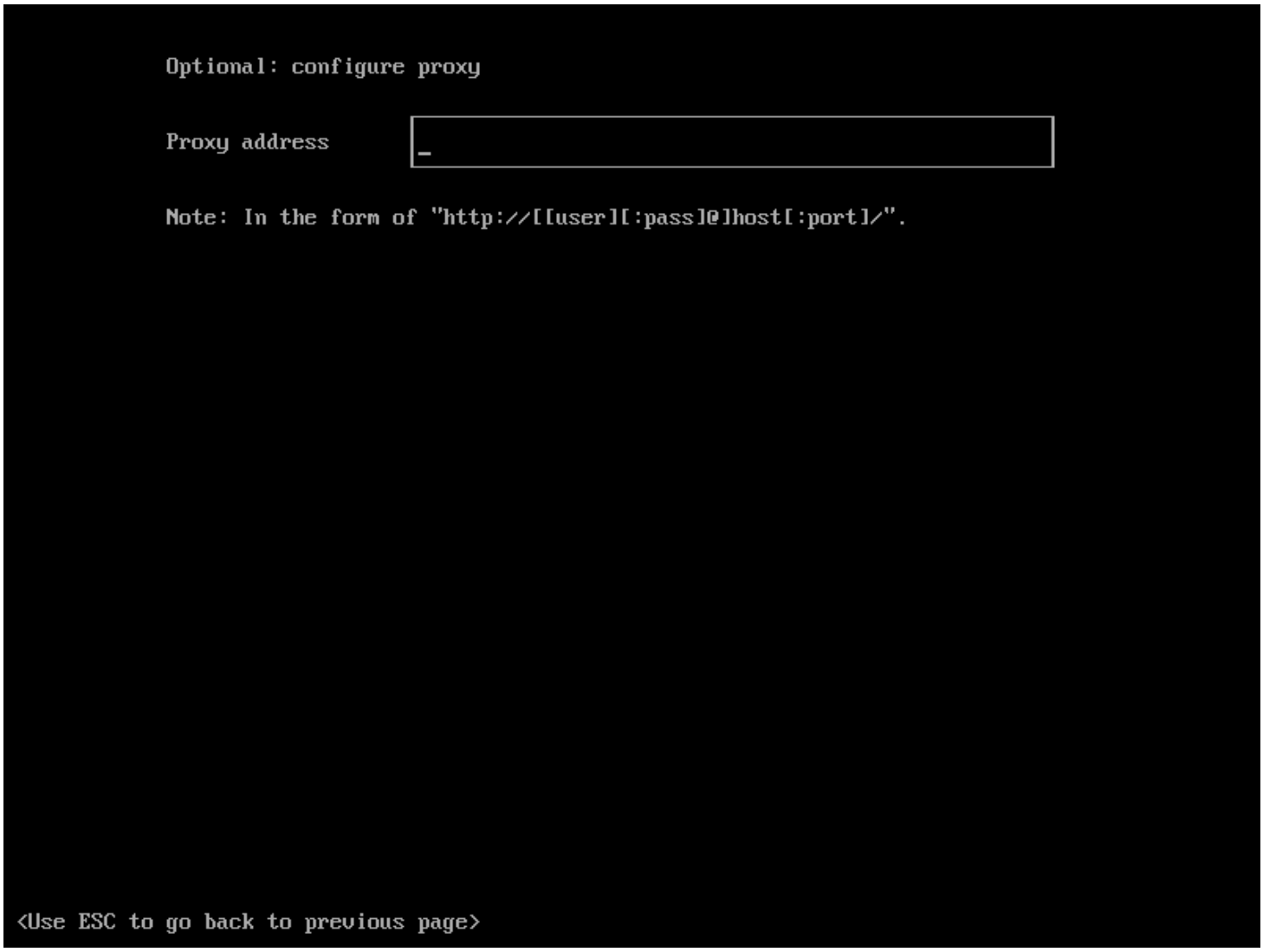
Configure the password for the Harvester node

Configure NTP. Below is the default NTP server that Harvester configures. You can change this here if needed.



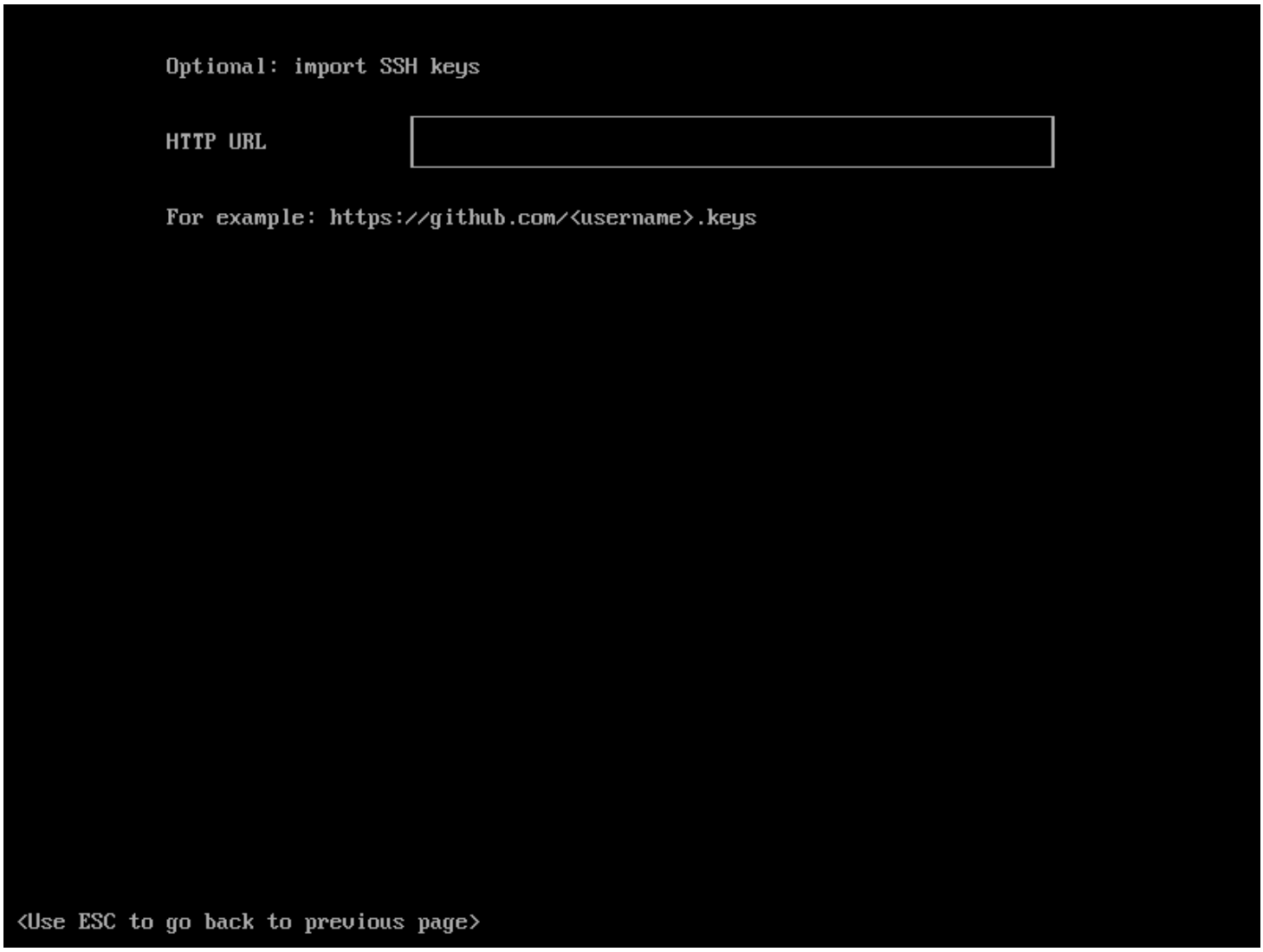
Configure NTP settings for Harvester

Set the optional proxy configuration if needed.



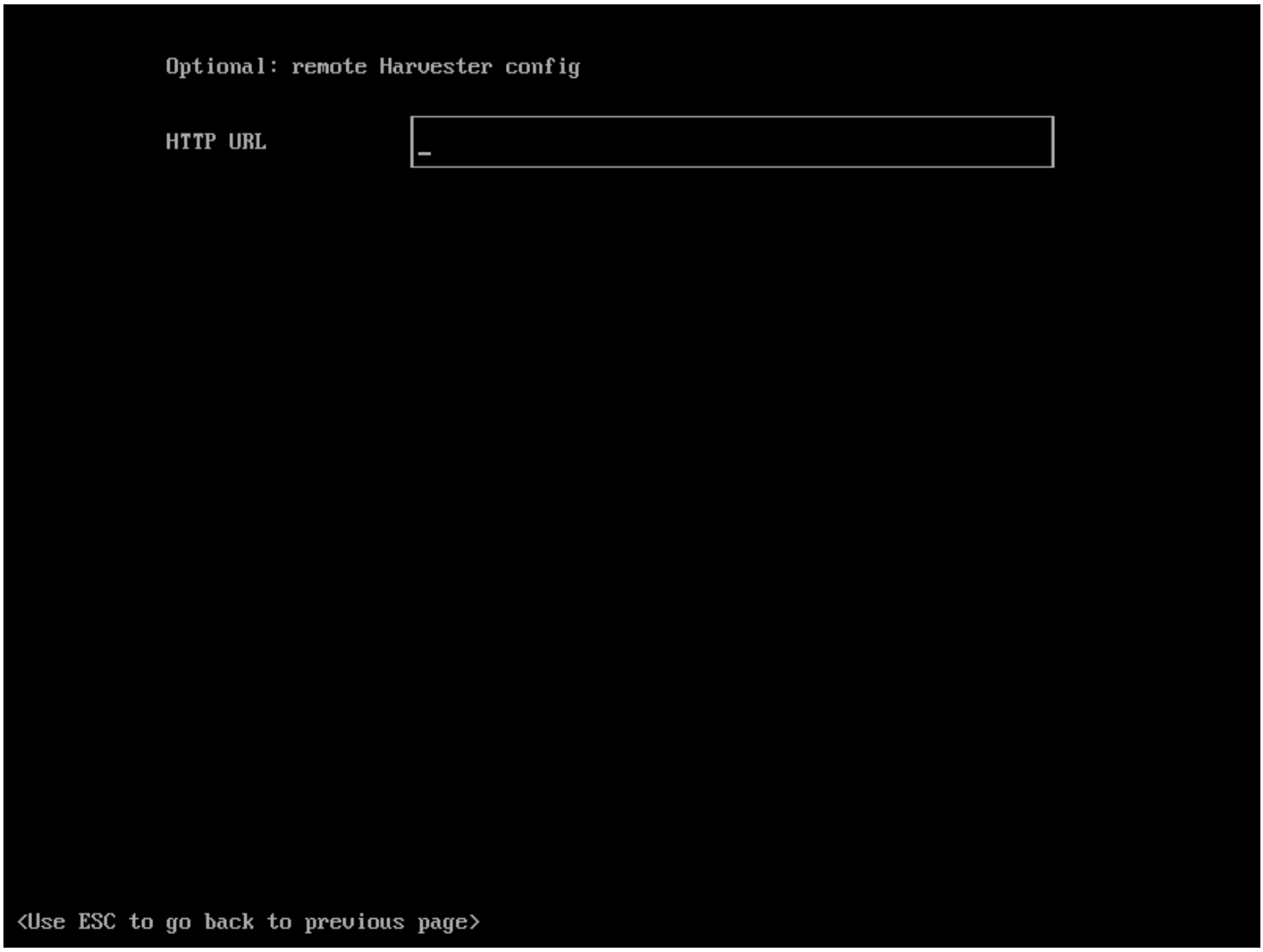
Set a proxy address if needed

Import SSH keys if needed.



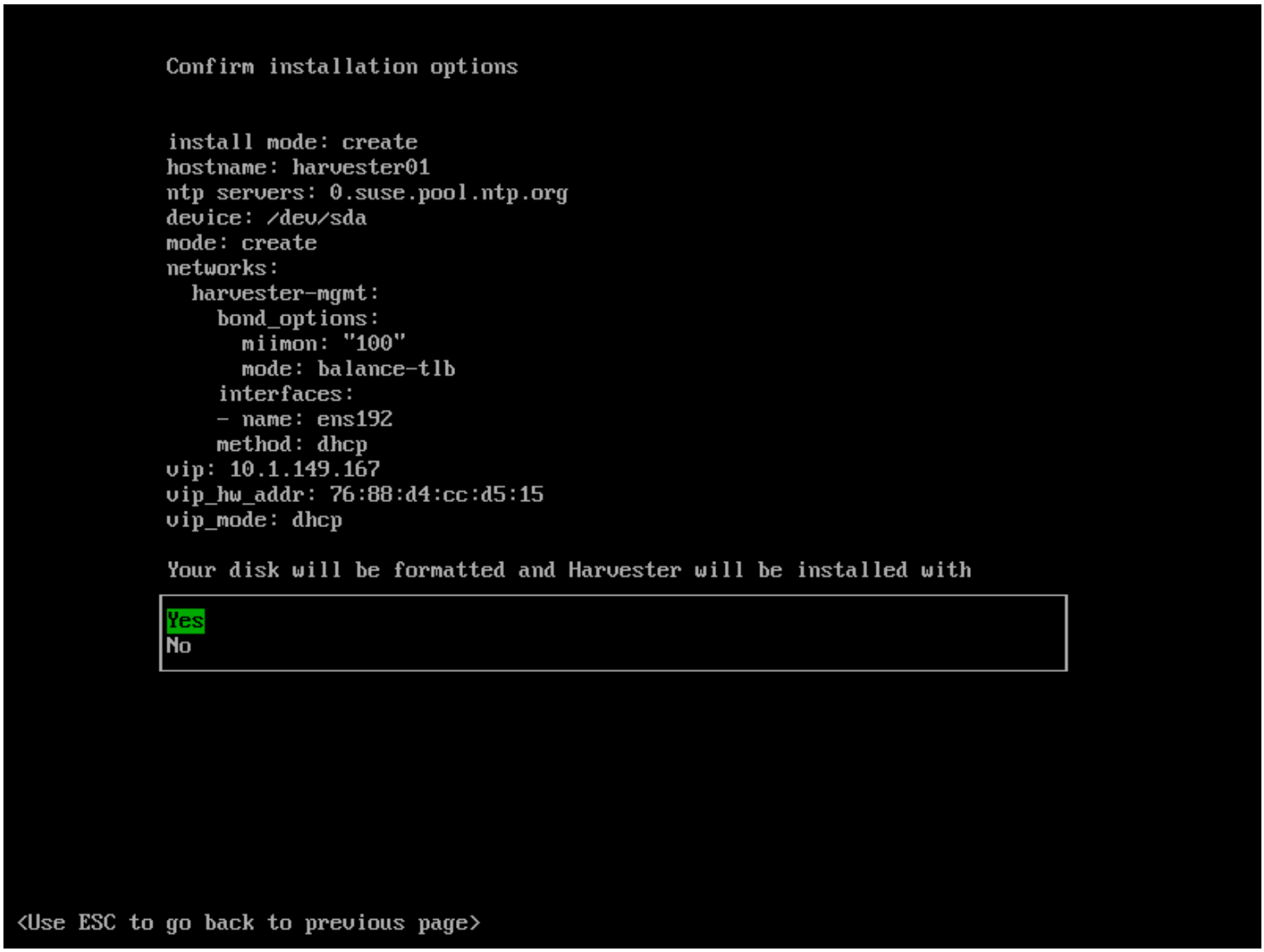
Import SSH keys for your Harvester node

Set the remote Harvester config if you have a config you want to use that is accessible via HTTP.



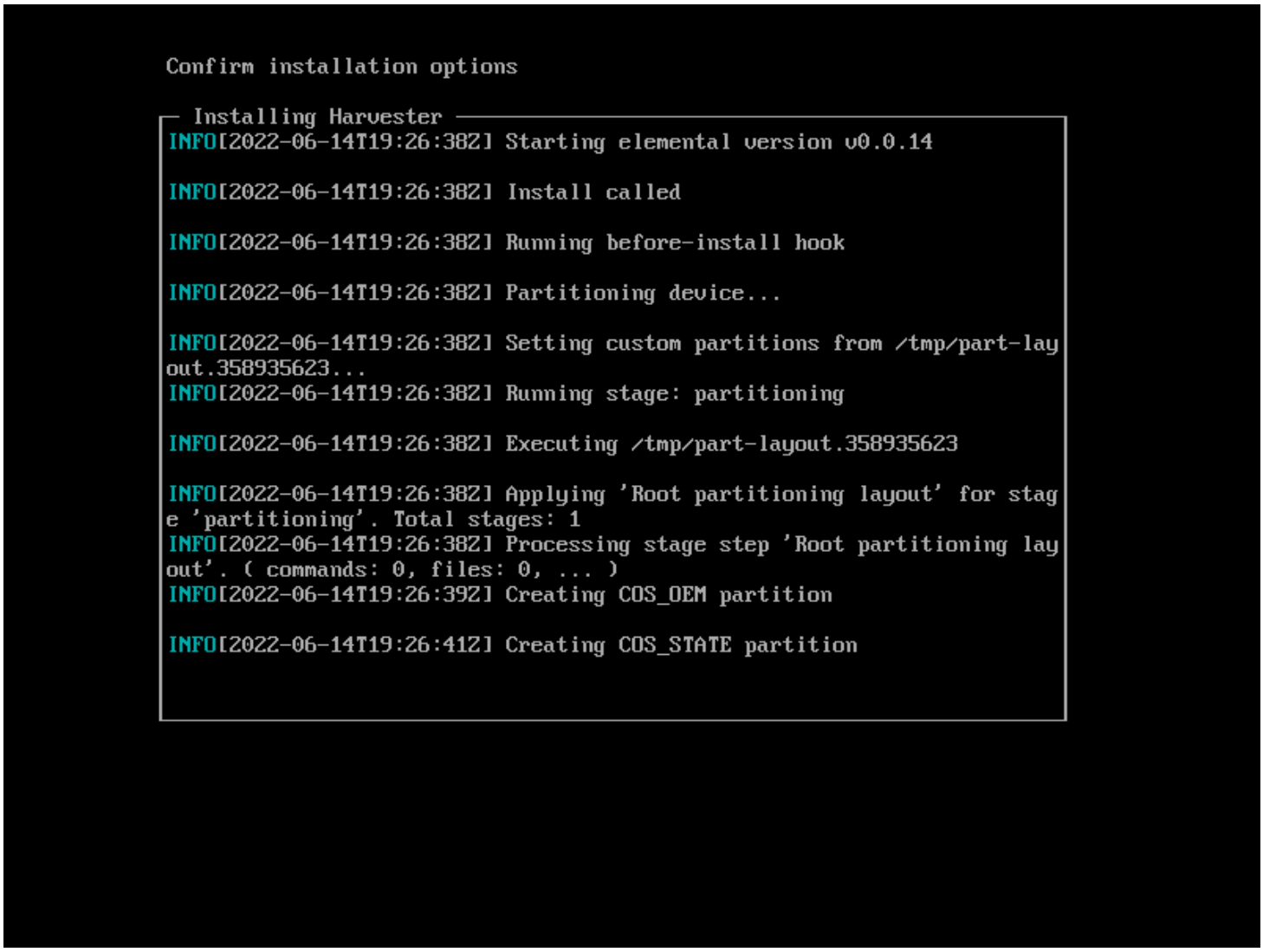
Remote Harvester configuration from HTTP

Finally, confirm your installation options. Select **Yes** to begin the Harvester installation.



Confirm the installation options of Harvester in VMware ESXi

The installation of Harvester begins.



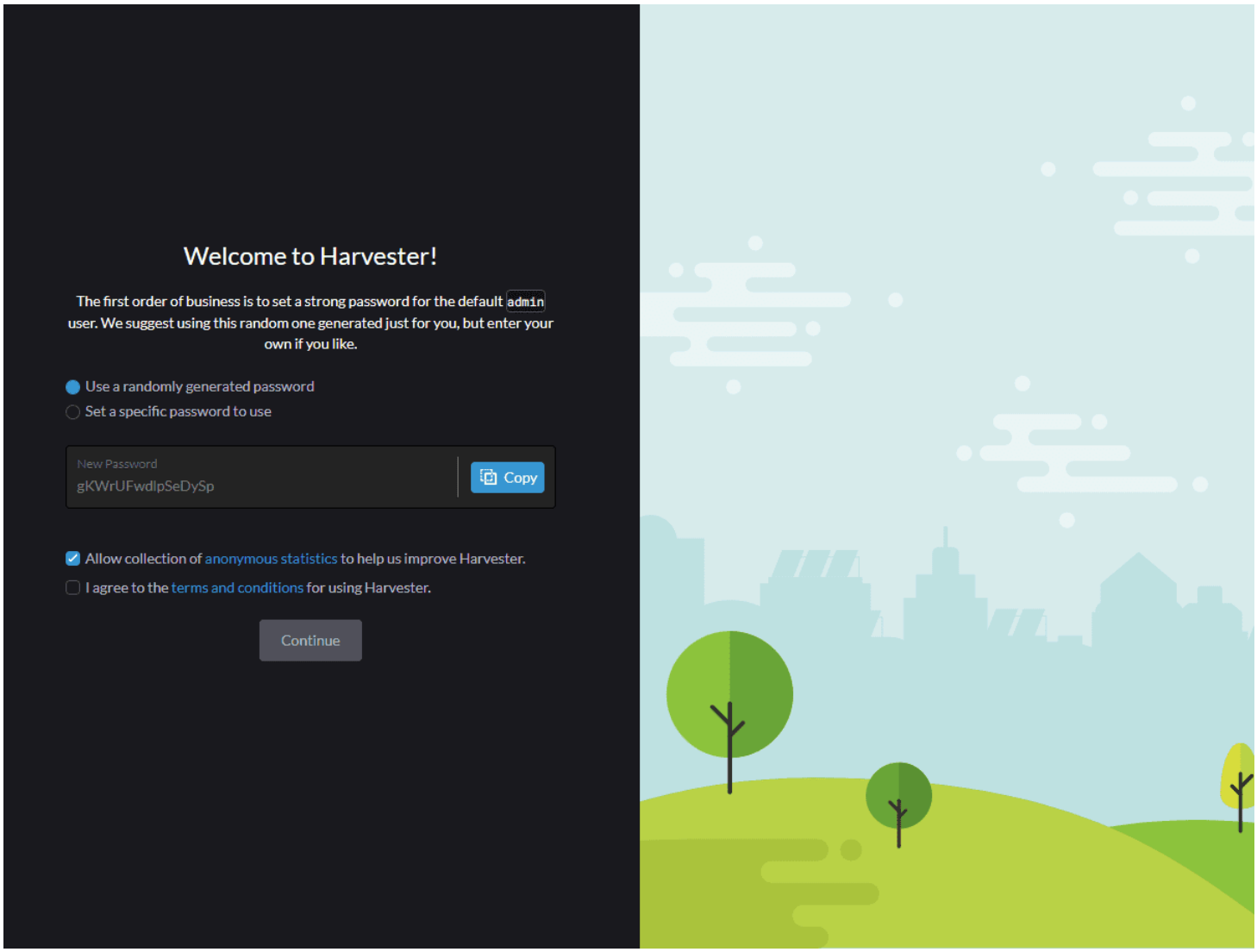
Harvester installation begins after confirming the installation options

Once the Harvester installation completes, you will see the **Current status** change to **Ready**. As a note, my node took a couple of minutes to change to the **Ready** state.

Harvester node is installed and in a Ready state

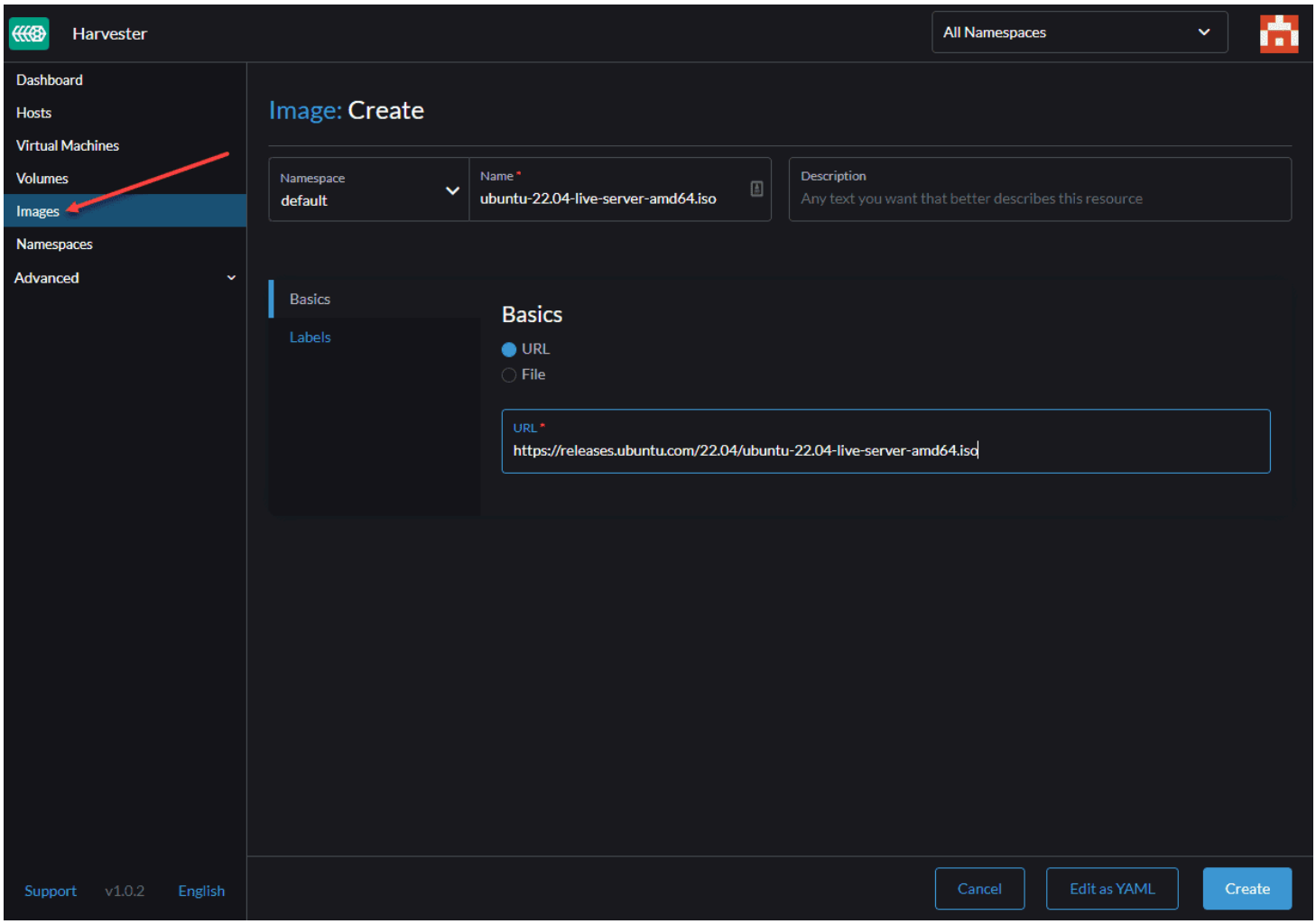
Accessing the Harvester admin page

Now that we have the Harvester node in the ready state, we should be able to access the management URL by browsing the VIP of the Harvester cluster. This page looks identical to the Rancher initial configuration. It will suggest a randomly generated password to use, or you can manually **set a specific password to use**.



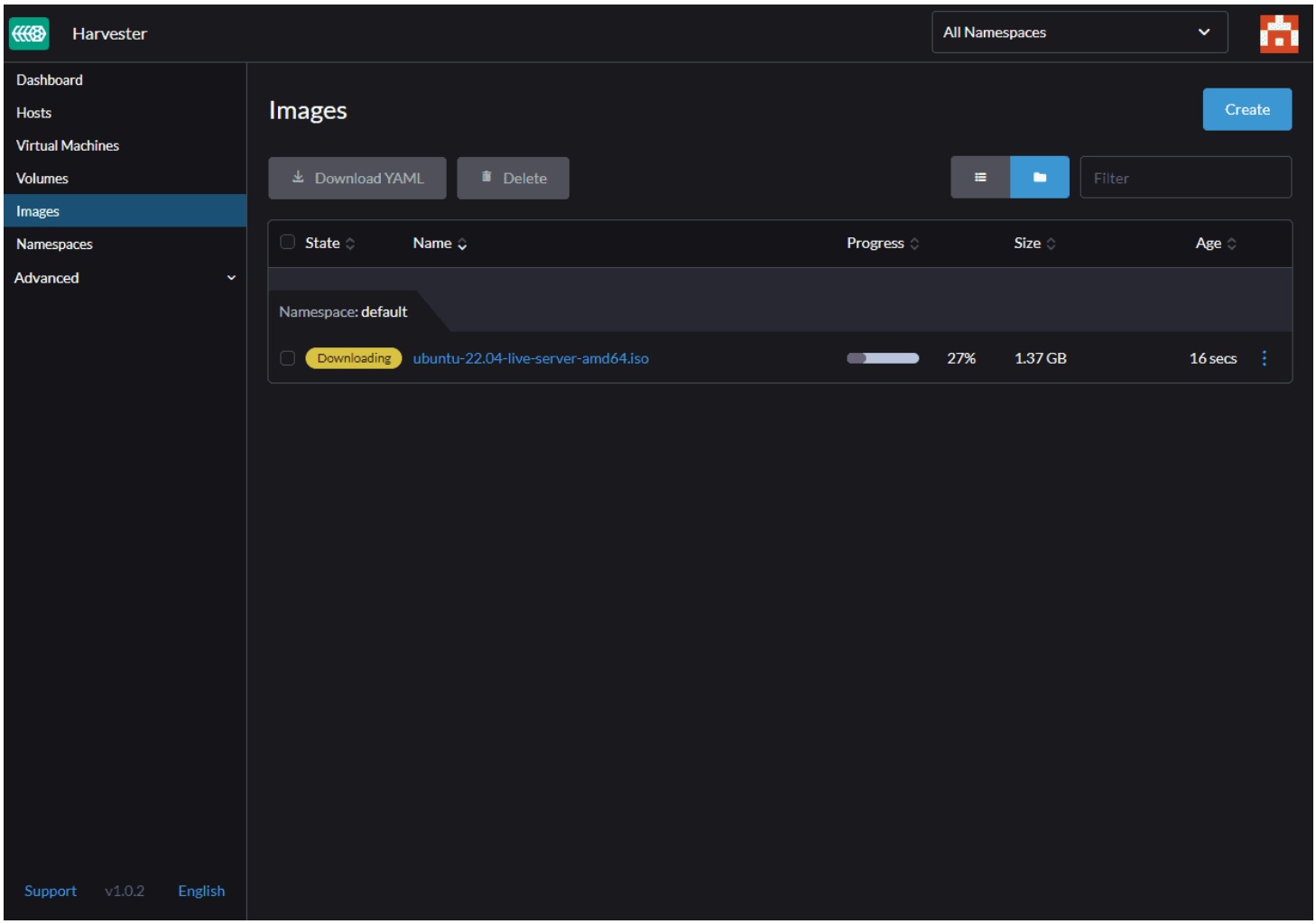
Setting your Harvester admin password 1

Now that we have access to the Harvester admin interface, navigate to **Images**. To install a virtual machine, we need to have an image to install from. Here, we can select **URL** and provide a download URL, such as for the latest Ubuntu 22.04 Server.



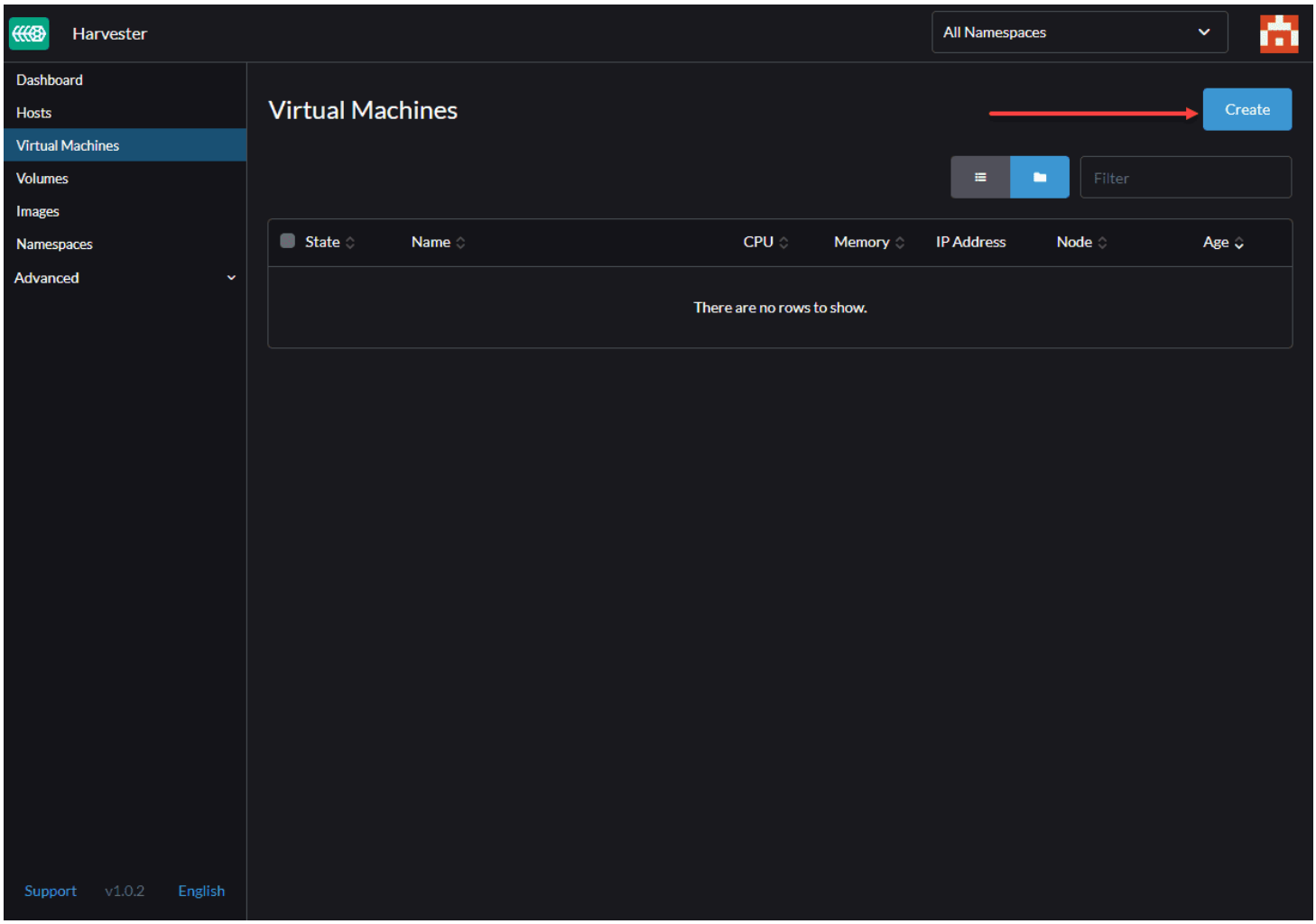
Uploading an Ubuntu image for installing Ubuntu 22.04 Server

The download of Ubuntu 22.04 ISO image begins.



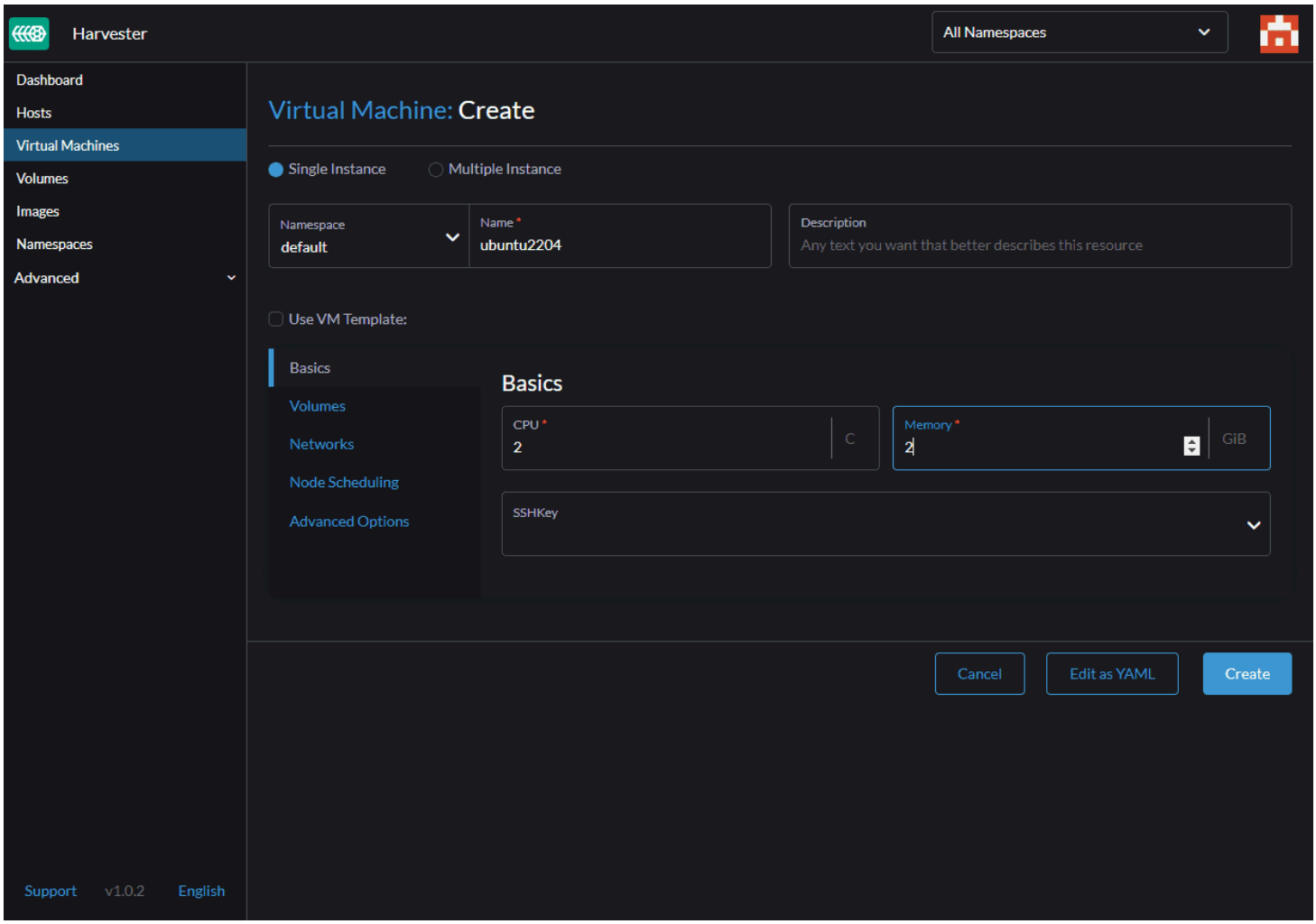
The Ubuntu 22.04 Server ISO begins downloading

Once the image finishes downloading, navigate to **Virtual Machines** and select **Create**.



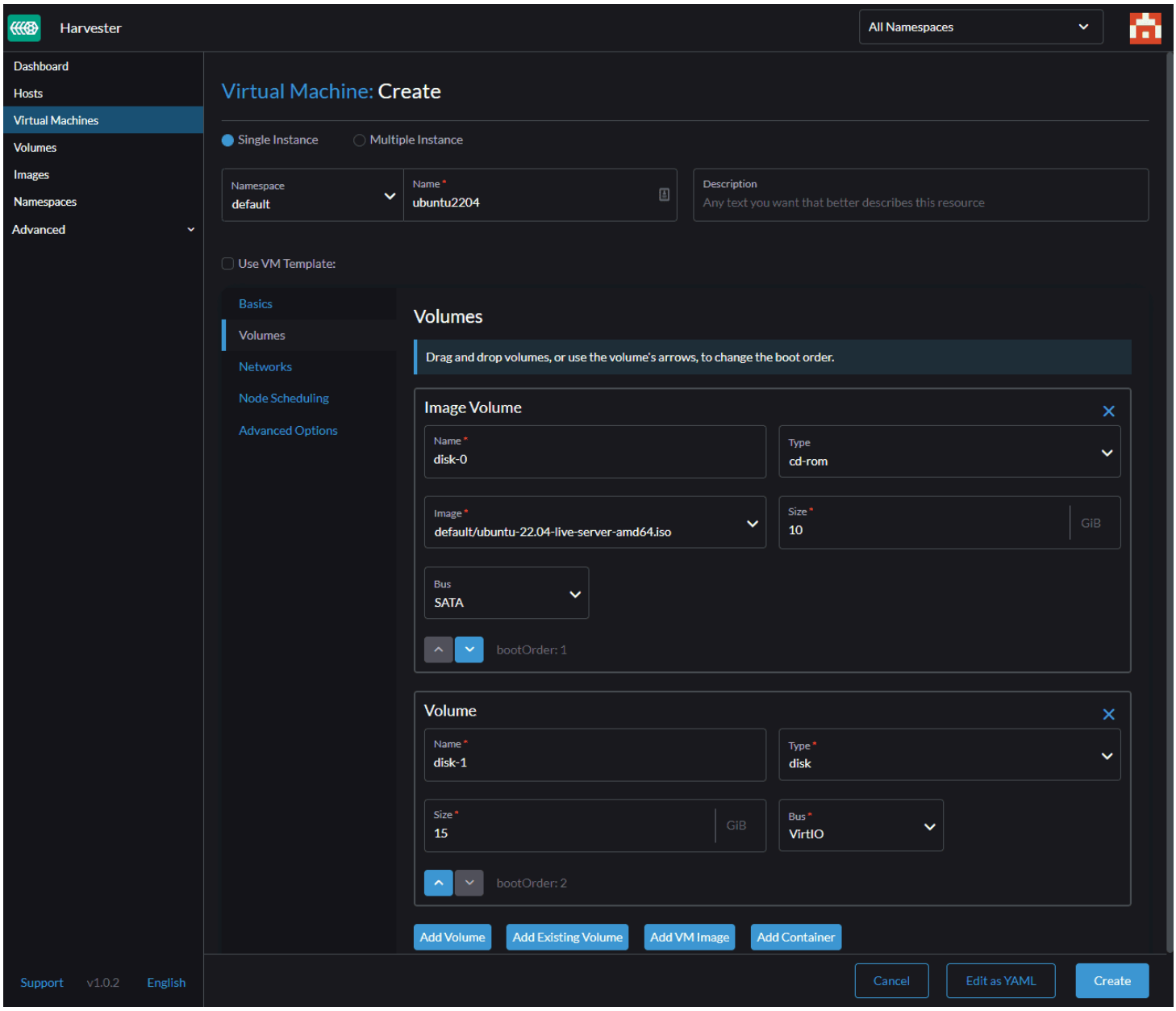
Beginning to create a new virtual machine in Harvester

On the **Basics** page, name the virtual machine and set the CPU and memory configuration.



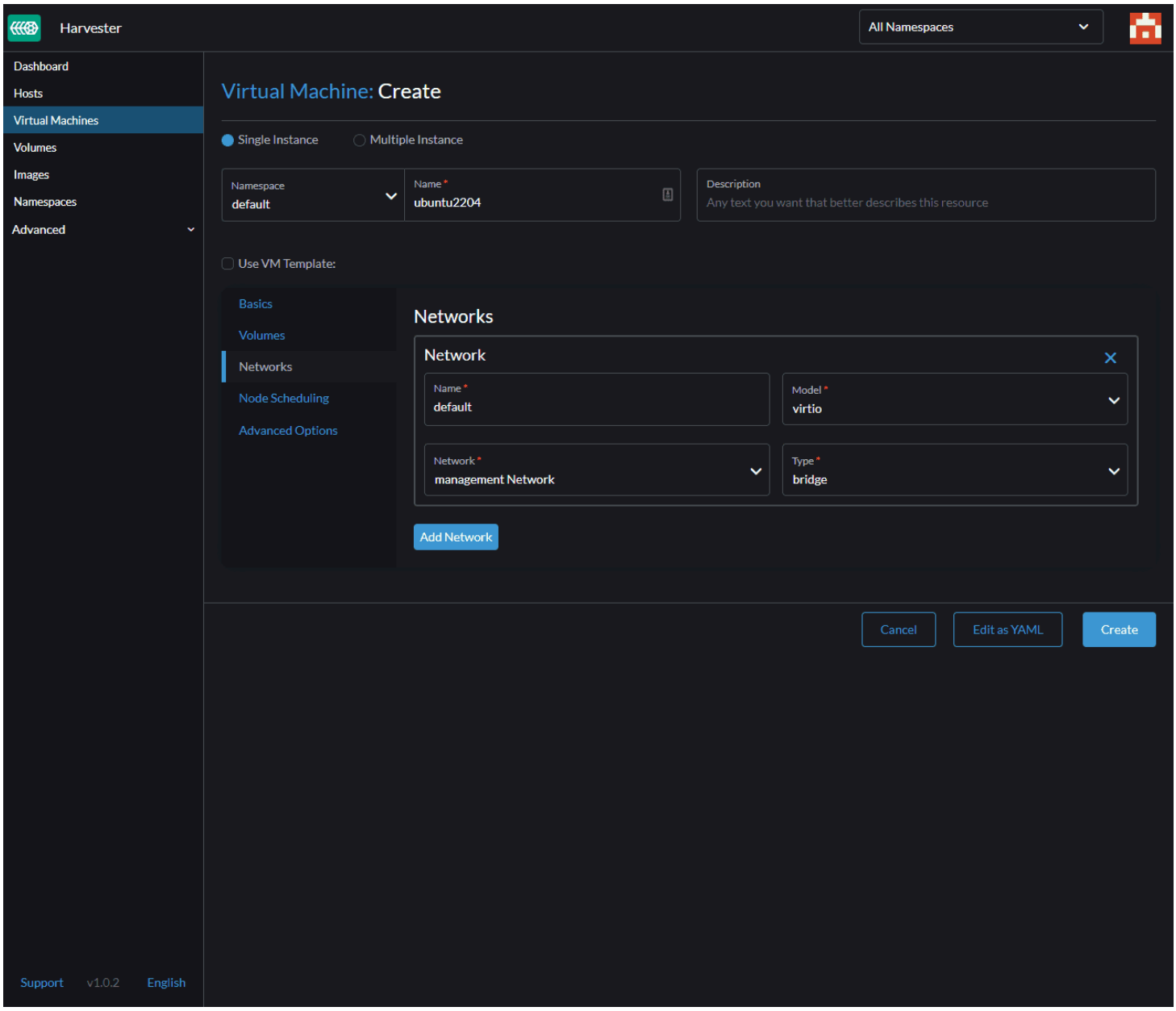
Set the CPU and memory configuration

On the **Volumes** configuration, for the first volume, change the type to **cd-rom** and select the image you downloaded under **Image**. Add an additional volume to serve as the hard disk for the VM. Here I am selecting the defaults for the most part. I have set the size to a meager 15 gigs just for testing purposes. I left the Bus configured for **VirtIO**.



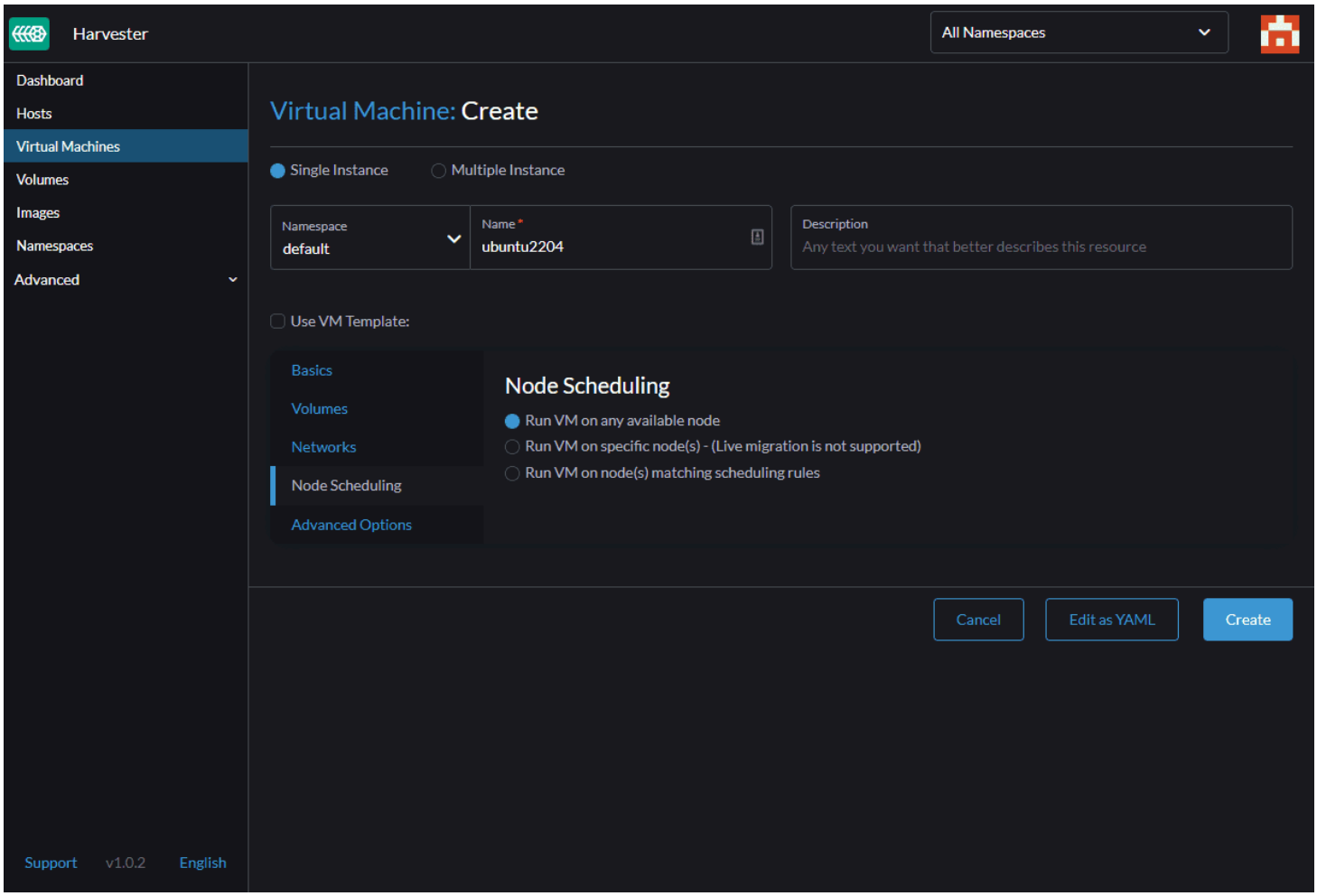
Configure the storage for the VM including the ISO image

Under **Networks**, you can configure the networking for the Ubuntu virtual machine. Here I have selected **Management Network** to share the management network and set the type to **Bridge**. Model is **virtio**.



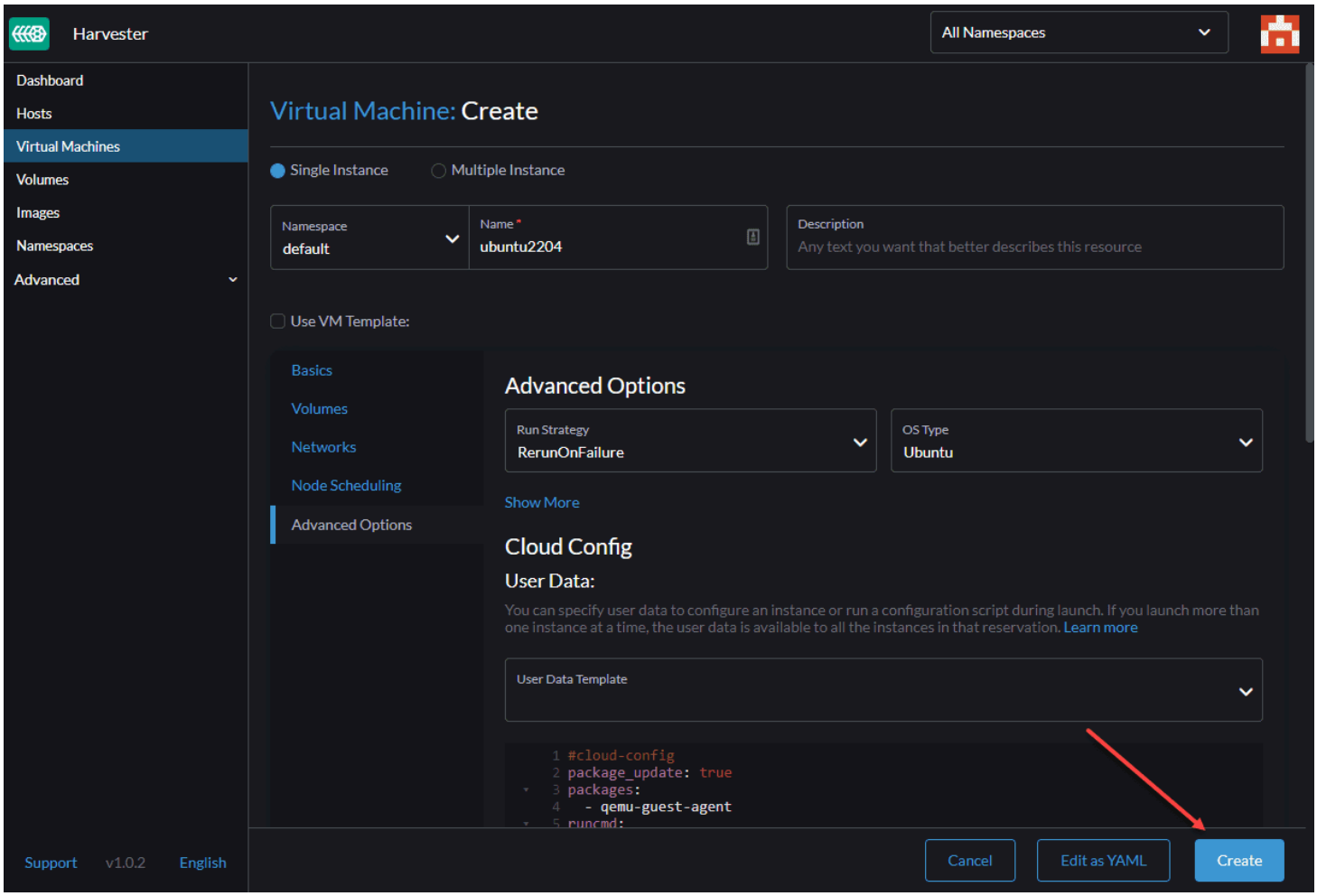
Configure your network options

I didn't change anything here, but if you have multiple Harvester nodes, these options are interesting. It affects how VMs live migrate or are pinned to specific hosts.



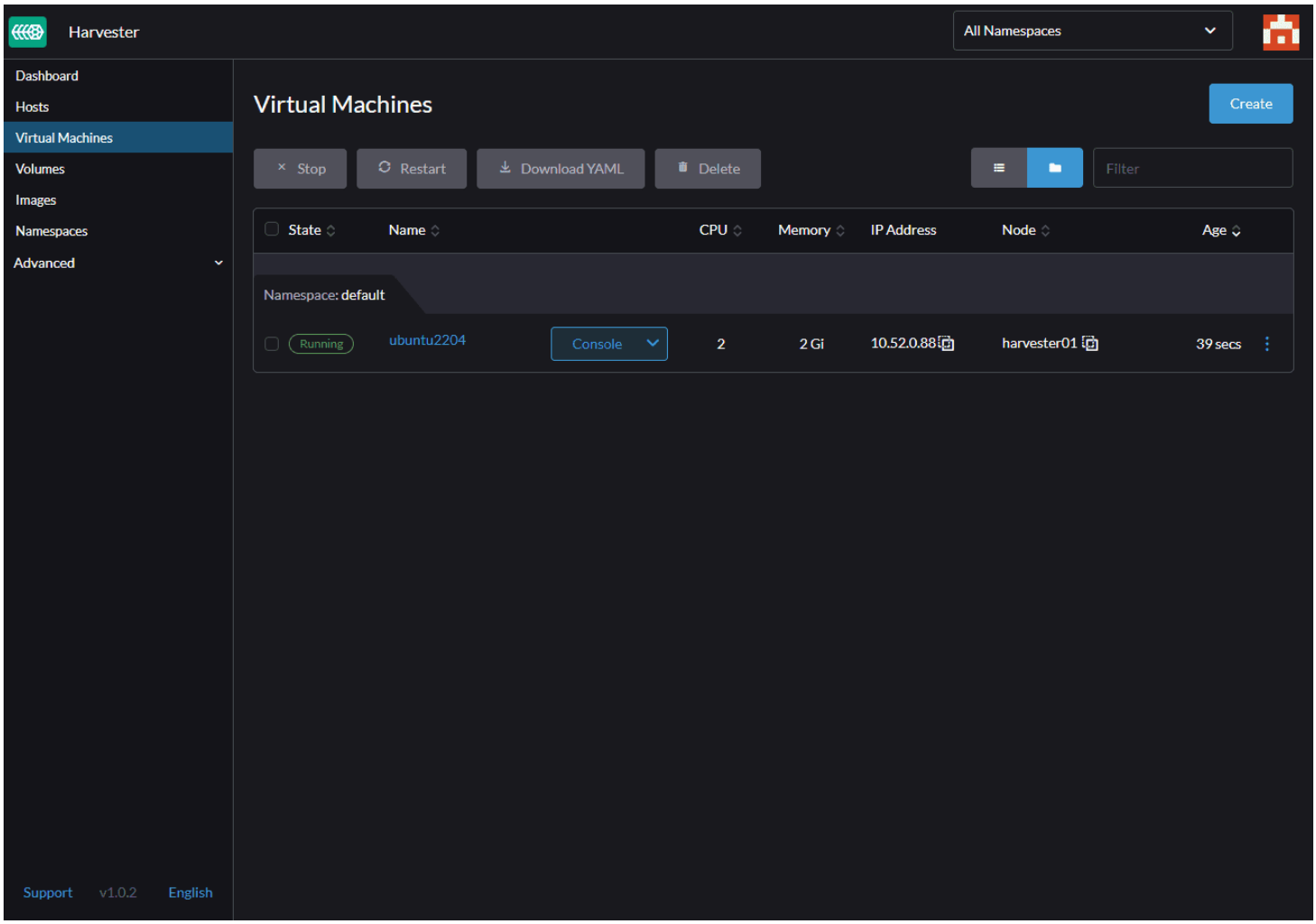
Viewing the node scheduling options for a Harvester VM

I also did not change anything in the advanced options screen, but again, lots of interesting options, including Cloud Config. When you are ready to create the VM, click **Create**.



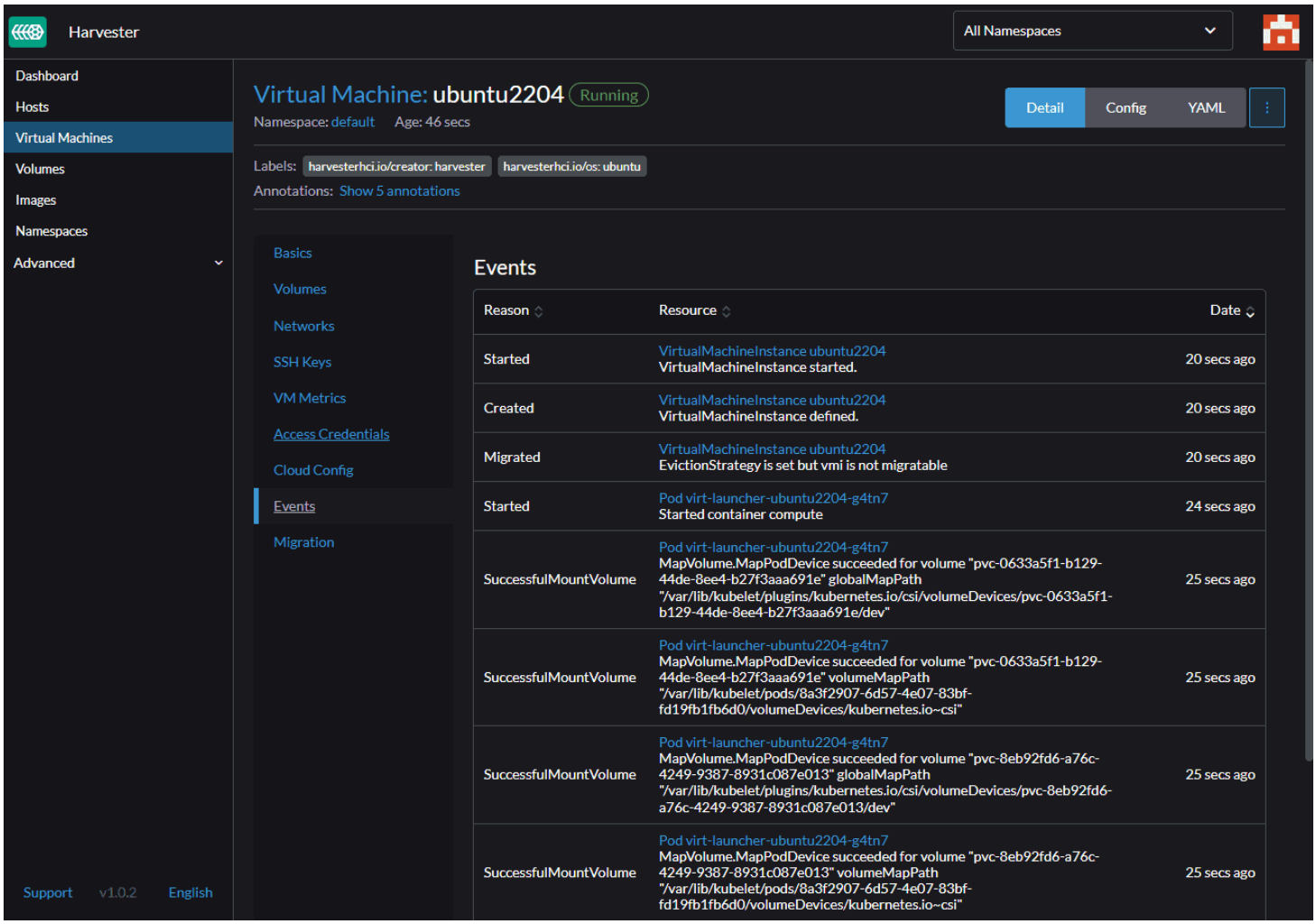
Viewing advanced options for the Harvester VM

The VM starts automatically and you should see it enter the **Running** state.



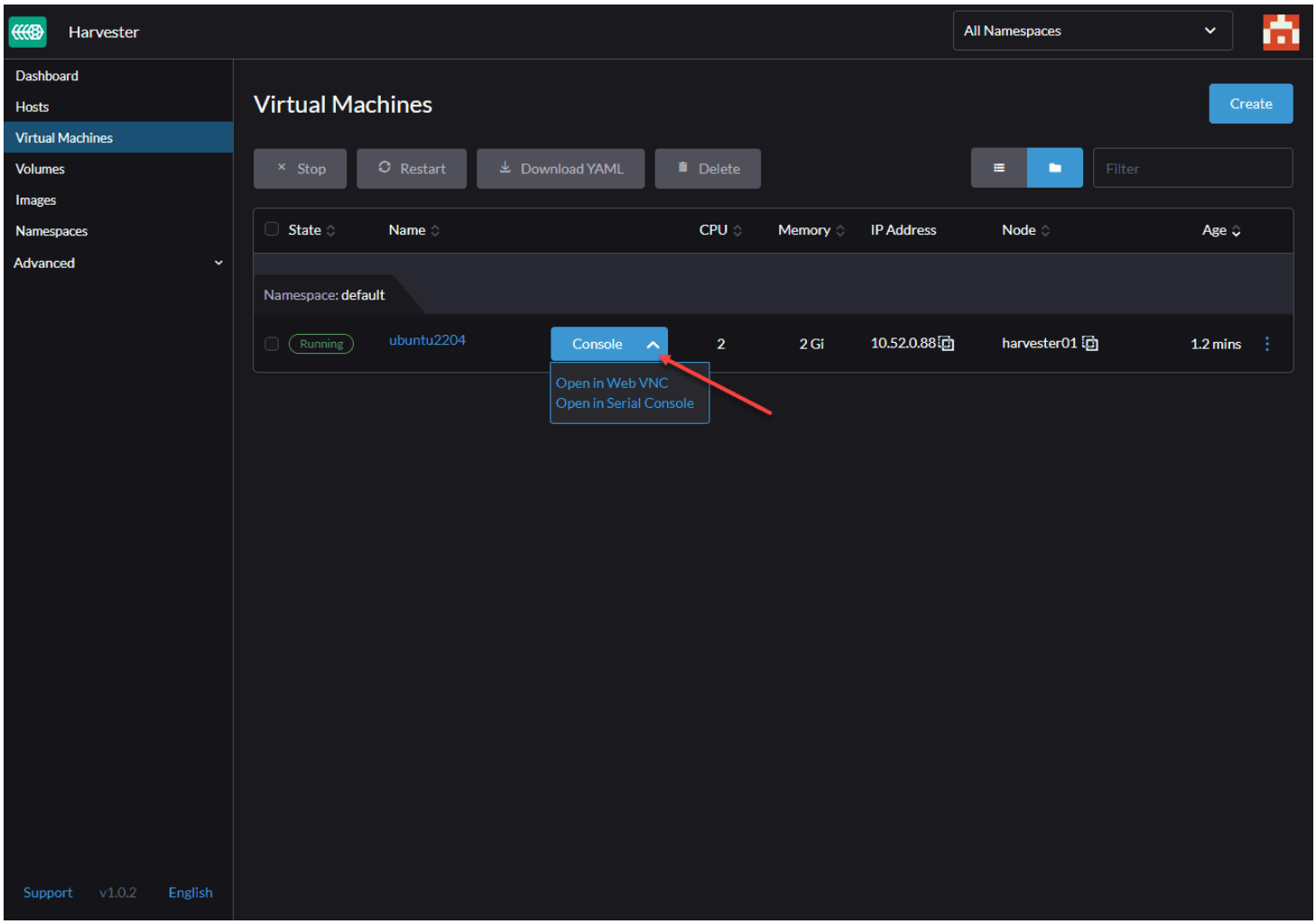
The virtual machine immediately starts running

If you have issues or see your VM get stuck starting, you can navigate to the **Detail > Events screen**, which shows the log entries. If you have any issues, they will be listed here.



Viewing events for the Harvester virtual machine

To open a console connection to your VM, go back to the Virtual Machines screen and click the arrow next to **Console**. You have a couple of choices here, but I am selecting **Open in Web VNC**.



Opening the virtual machine console in Harvester

The web VNC window opens, and you now have console access to your Ubuntu VM. Cool stuff.

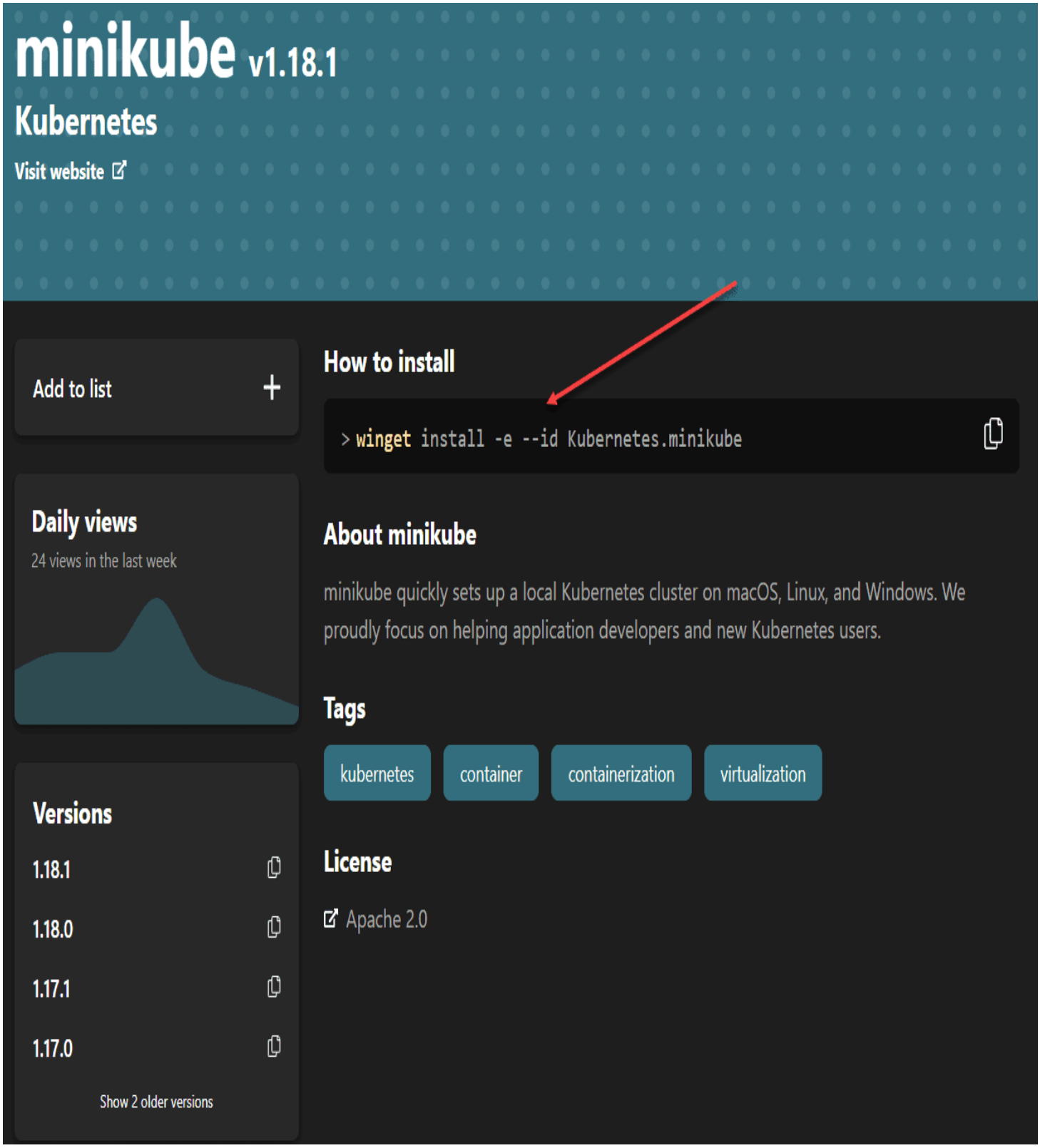
Ubuntu Server installation begins as it boots from the ISO image

Wrapping Up

As shown, the process to Install Harvester in VMware ESXi is pretty straightforward. This is an interesting solution that I would like to play around with more. I think solutions like Harvester have a long way to go before offering the enterprise features and capabilities businesses are used to with a mature, robust hypervisor like ESXi. However, for organizations already using Rancher and who want to stick with open source solutions for running virtual machines, Harvester has a lot of potential with seamless integration to their cloud-native stack.

Install Minikube in Windows Server 2022 using Winget

November 22, 2021
Kubernetes



Install Minikube using winget in Windows Server 2022

If you have not read my post about how to install Winget in Windows Server 2022 – No applicable app licenses error, you can see how to install the winget tool in Windows Server 2022. Building on that post, I want to show you guys how to install Minikube in Windows Server 2022 using Winget. Winget offers a very smooth and seamless way to get Minikube installed

and running in Windows Server 2022. However, it does require the workarounds I detailed in the blog post above to get Winget working. For the remainder of the post, I will assume you have winget working in Windows Server 2022 and we can go from there.

Install Minikube in Windows Server 2022 using Winget

There are just a few steps to begin working with Minkube in Windows Server 2022. However, there are a few prerequisites I will mention below:

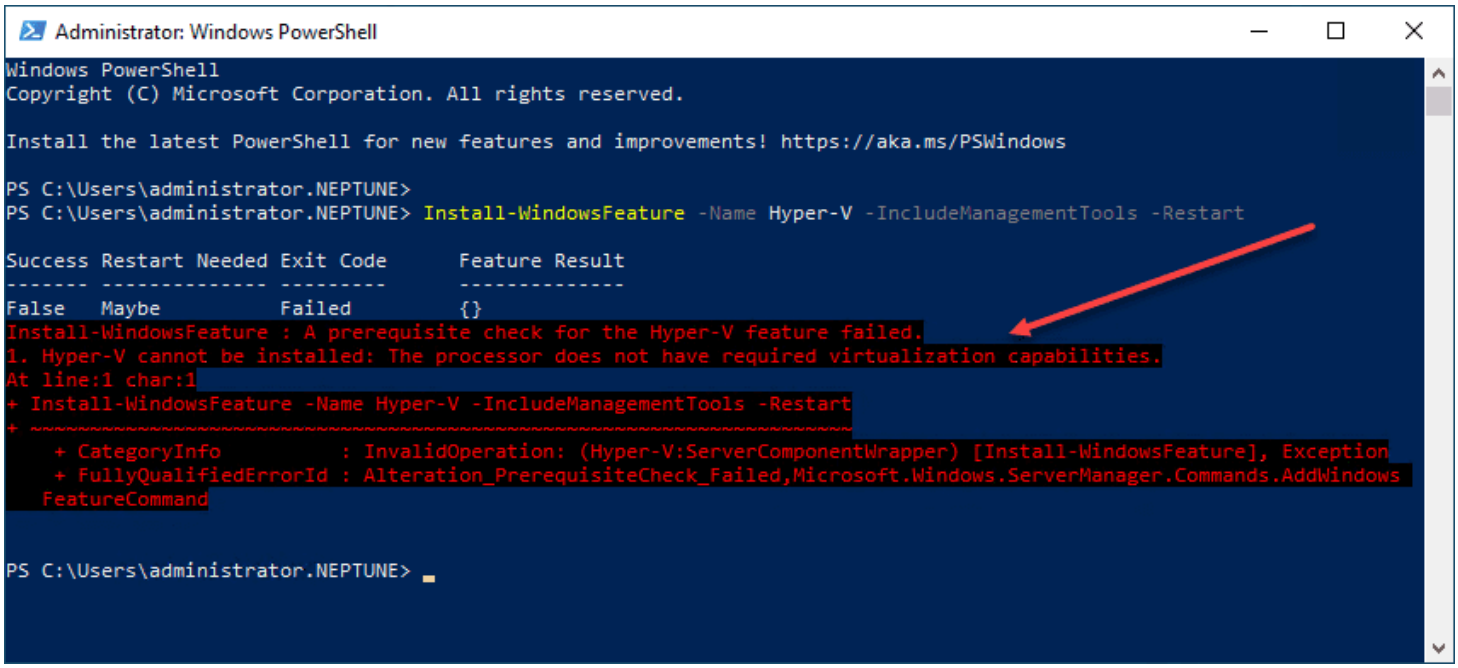
1. Installation of Hyper-V Server
    - Enable nested virtualization if using Hyper-V inside a VM
2. Install winget using the steps in the blog post linked above
3. Install Minikube in Windows Server 2022
4. Create a Hyper-V External Virtual Switch
5. Build your Minikube Kubernetes cluster
6. Kubernetes tools to work with Minikube

1. Installation of Hyper-V Server

To begin with, you will need to install Hyper-V Server in Windows Server 2022. To do that, we can pull out trusty PowerShell to install the role and management tools.

```
Install-WindowsFeature -Name Hyper-V -IncludeManagementTools -Restart
```

If you run into the following error when you attempt to enable the Hyper-V Role, it means you need to enable nested virtualization. The process varies between VMware and Hyper-V. However, both are easy to enable.



Error due to nested virtualization not being enabled

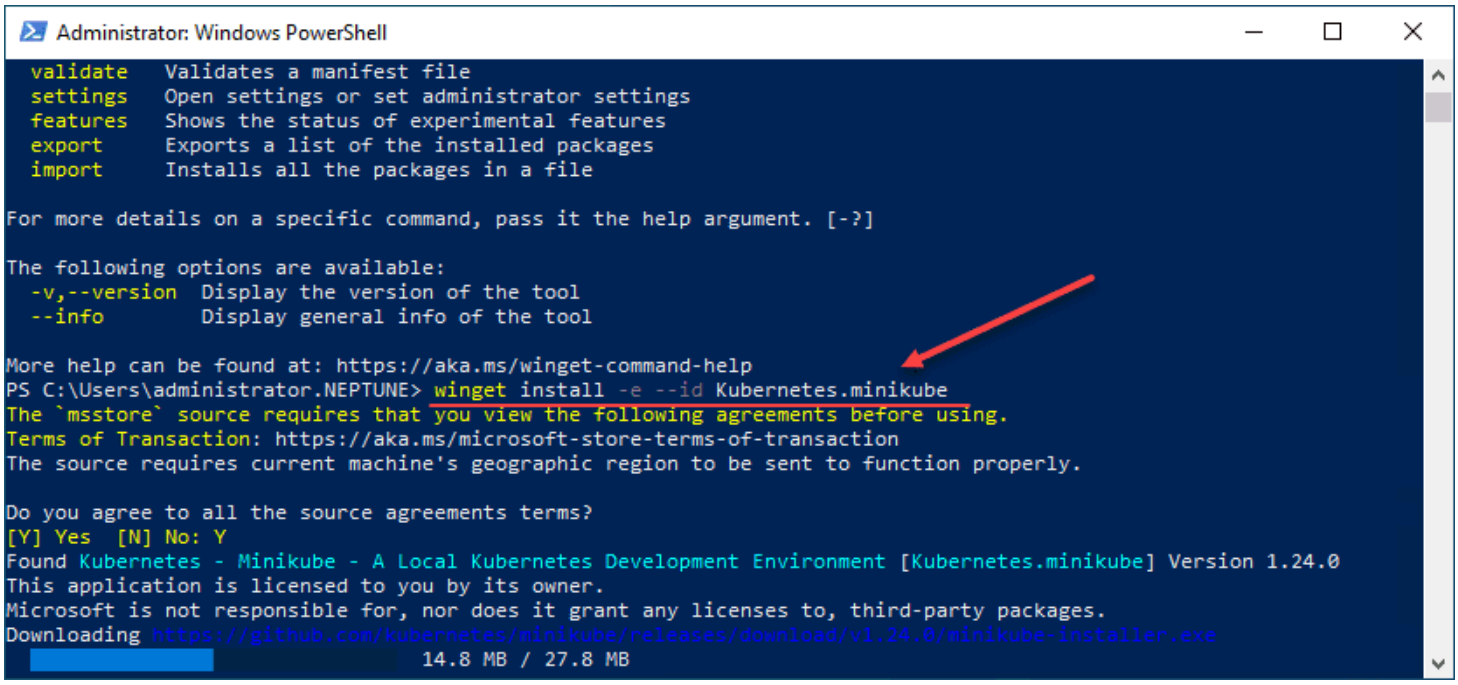
2. Install winget

Again, for your reference, you can see how to install winget in Windows Server 2022 here: Install Winget in Windows Server 2022 – No applicable app licenses error – Virtualization Howto

3. Install Minikube in Windows Server 2022

After you have installed winget on your Windows Server 2022 server, you can install Minikube. The very simple command to install Minikube in Windows Server 2022 once you have winget installed is the following:

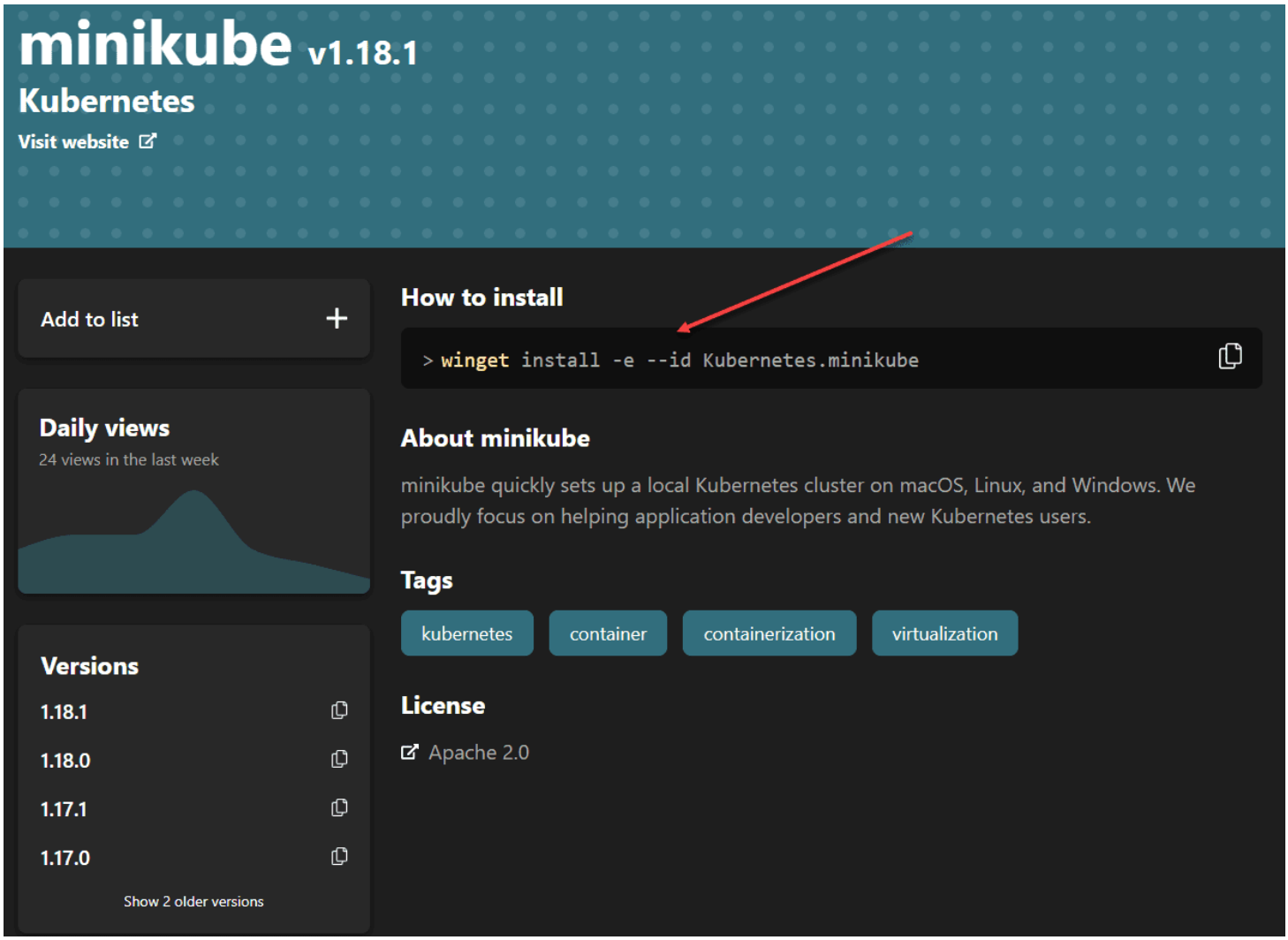
```
winget install -e --id Kubernetes.minikube
```



Install Minikube in Windows Server 2022

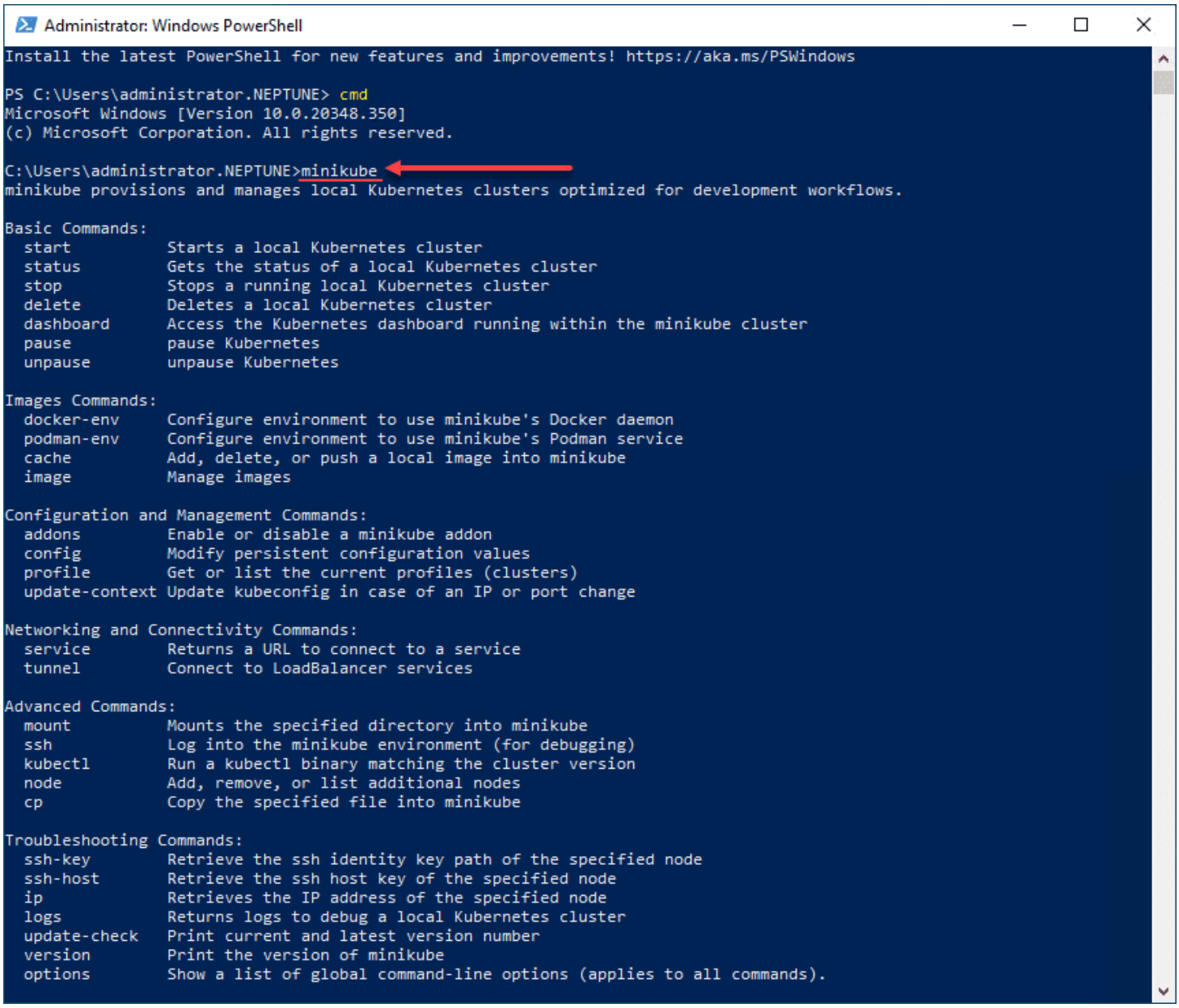
You can read the official documentation here:

- [Download and install minikube with winget]



Install Minikube using winget in Windows Server 2022

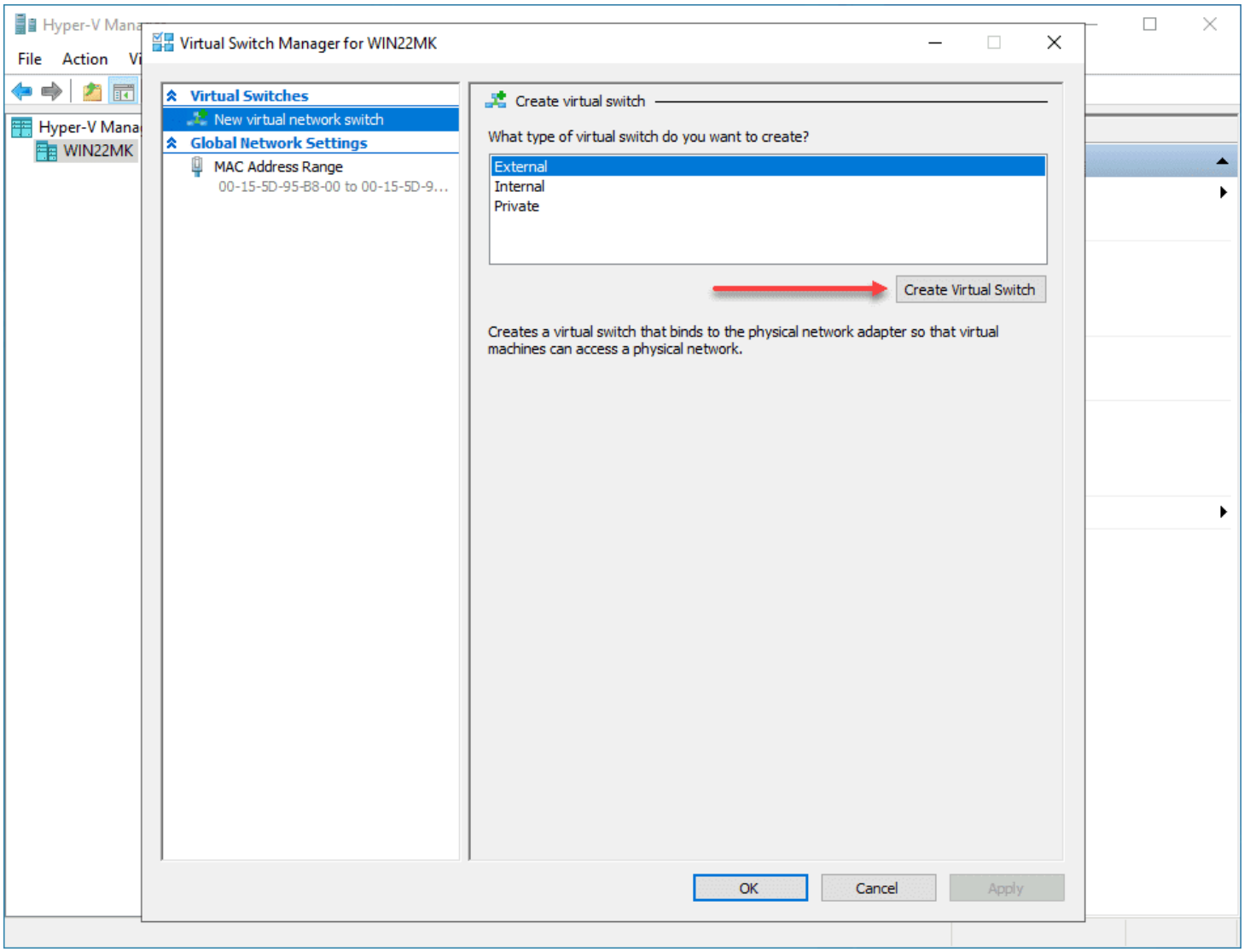
Check to see if Minikube is installed successfully by entering the command **minikube**.



Minikube is installed successfully viewing the parameters

4. Create a Hyper-V External Virtual Switch

Before you can start your Minikube Kubernetes cluster on the Windows Server 2022 machine with Hyper-V running, you need to create an external Hyper-V virtual network switch. You can do this in your Hyper-V Manager, using the **virtmgmt.msc** command. Navigate to the Virtual Switch Manager and create a new external virtual switch.



Create a new External Hyper V virtual switch

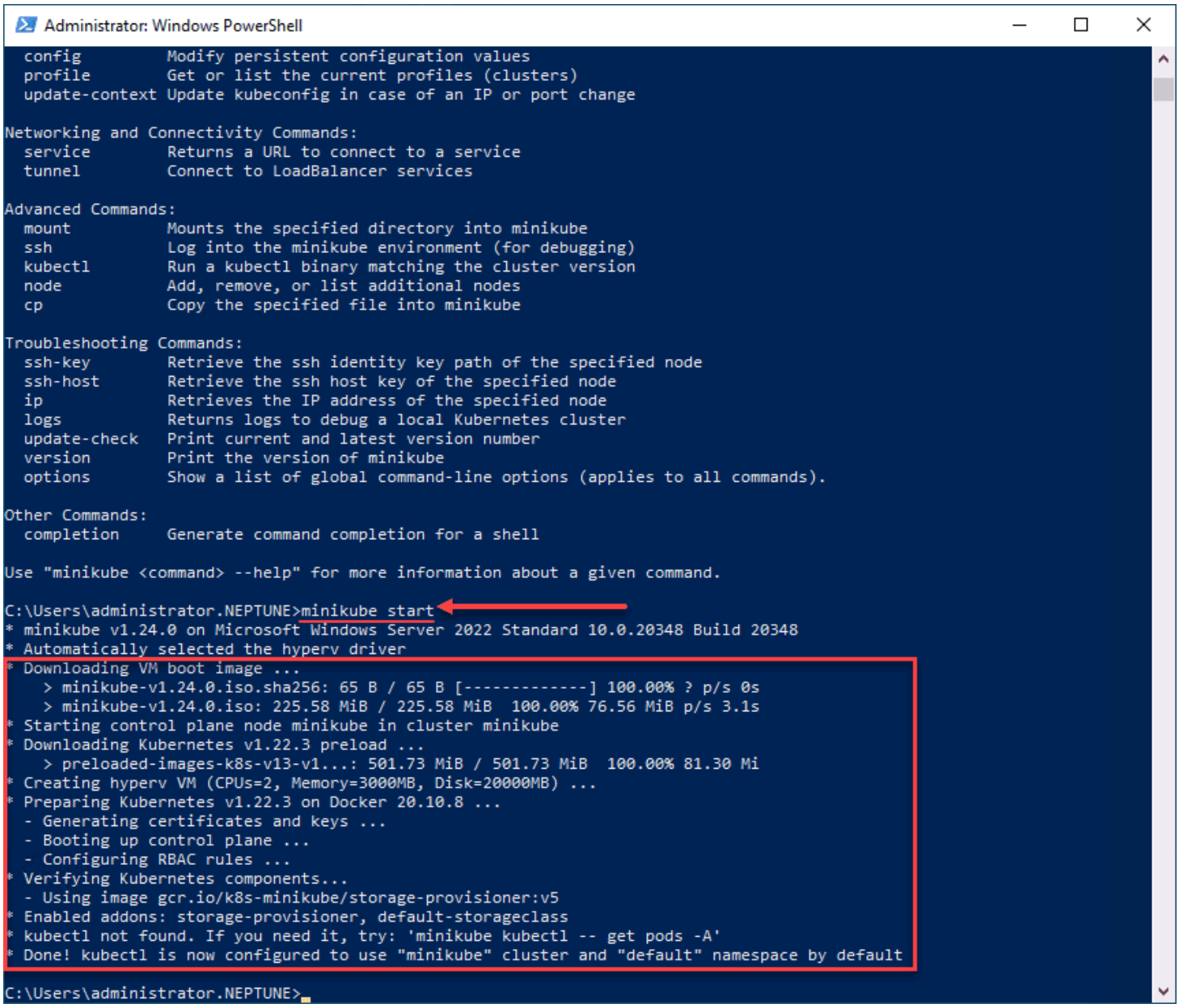
5. Build your Minikube Kubernetes Cluster

Once you have installed Minikube, the process to build your Minikube Kubernetes Cluster is fairly straightforward. On your Windows Server 2022 server, simply issue the command:

```
minikube start
```

This will go through the following steps:

1. Automatically selects the Hyper-V driver
2. Downloads the VM boot image
3. Starts the control plane node minikube in the cluster
4. Downloads the latest Kubernetes installation
5. Prepares Kubernetes
    - Generating certificates and keys
    - Boot up control plane VM
    - Configure RBAC rules
6. Verify Kubernetes components
7. Enable addons



Start the Minikube cluster in Windows Server 2022

6. Kubernetes tools to work with Minikube

Either before or after you install Minikube, you can install your Kubernetes tools that make life easier to work with the Kubernetes clusters. These include:

- kubectl
- helm

You can install kubectl by following the documentation provided here:

- Install and Set Up kubectl on Windows | Kubernetes

Follow the documentation here to install helm in Windows:

- Helm | Installing Helm

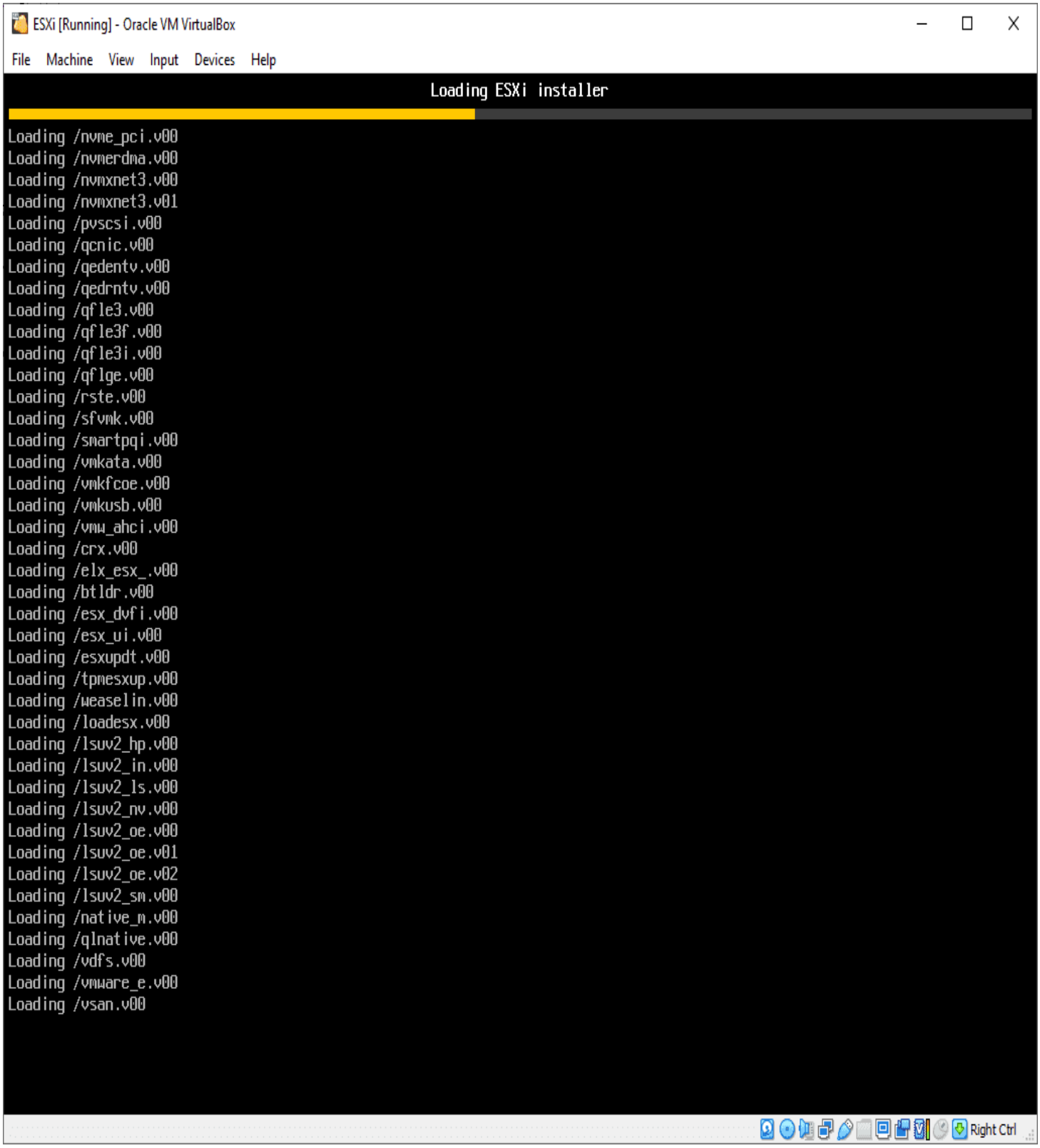
Wrapping Up

Minikube and really spinning up any Kubernetes framework is a great way to learn. There are many Kubernetes resources out there now that make learning about and getting up to speed quickly with Kubernetes extremely easy. Minikube is one

of the easiest to get up and running on both Linux and Windows. It makes the process of standing up the Kubernetes architecture a simple command that you run. Hopefully, this walkthrough of how to Install Minikube in Windows Server 2022 using Winget will help any who want to use Windows Server 2022 to host their first Minikube Kubernetes cluster.

VirtualBox Nested Virtualization with VMware ESXi

February 16, 2021
Virtualization



Powering on and booting your esxi vm in virtualbox

Oracle VirtualBox is an extremely popular enthusiast hypervisor that many use to learn various technologies and get their feet wet using virtualization technologies. VirtualBox is a freely available hypervisor that provides many great options, including a broad range of compatible host operating systems and the ability to run many different guest operating

systems. Let's take a look at VirtualBox nested virtualization with VMware ESXi. Nested virtualization with VirtualBox can go either way. You can use either nested VirtualBox inside of ESXi or ESXi inside VirtualBox. Or, what about both? First, let's take a look at how to enable nested virtualization in VirtualBox.

Enabling Nested Virtualization in VirtualBox

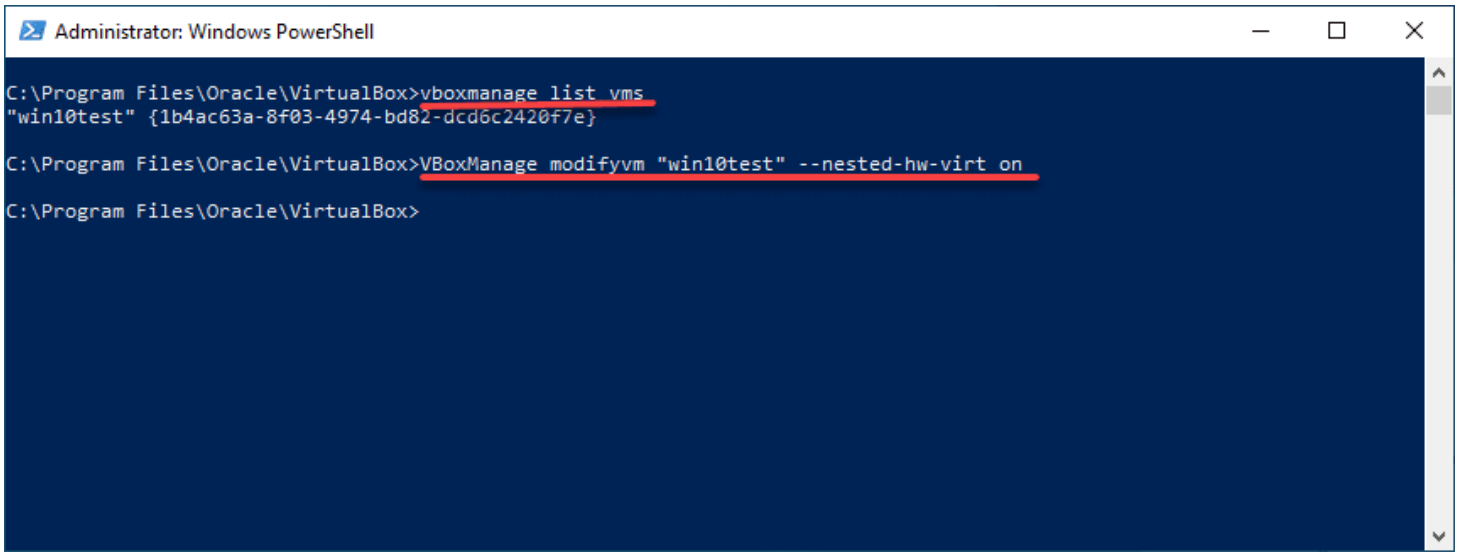
If you are not as familiar with Oracle VirtualBox, especially if you have mainly had your hands on VMware vSphere or Microsoft Hyper-V, you may not be as familiar with or have ever configured nested virtualization inside of VirtualBox.

With VirtualBox, you can enable nested virtualization using the VirtualBox GUI or using the command-line. For the majority who may be using Intel processors, you will need to enable nested virtualization settings from the VirtualBox command-line.

There is a special tool included with VirtualBox that allows enabling the nested virtualization flag. This is the **vboxmanage** utility. Using vboxmanage, you can easily enable nested virtualization on your VirtualBox VM.

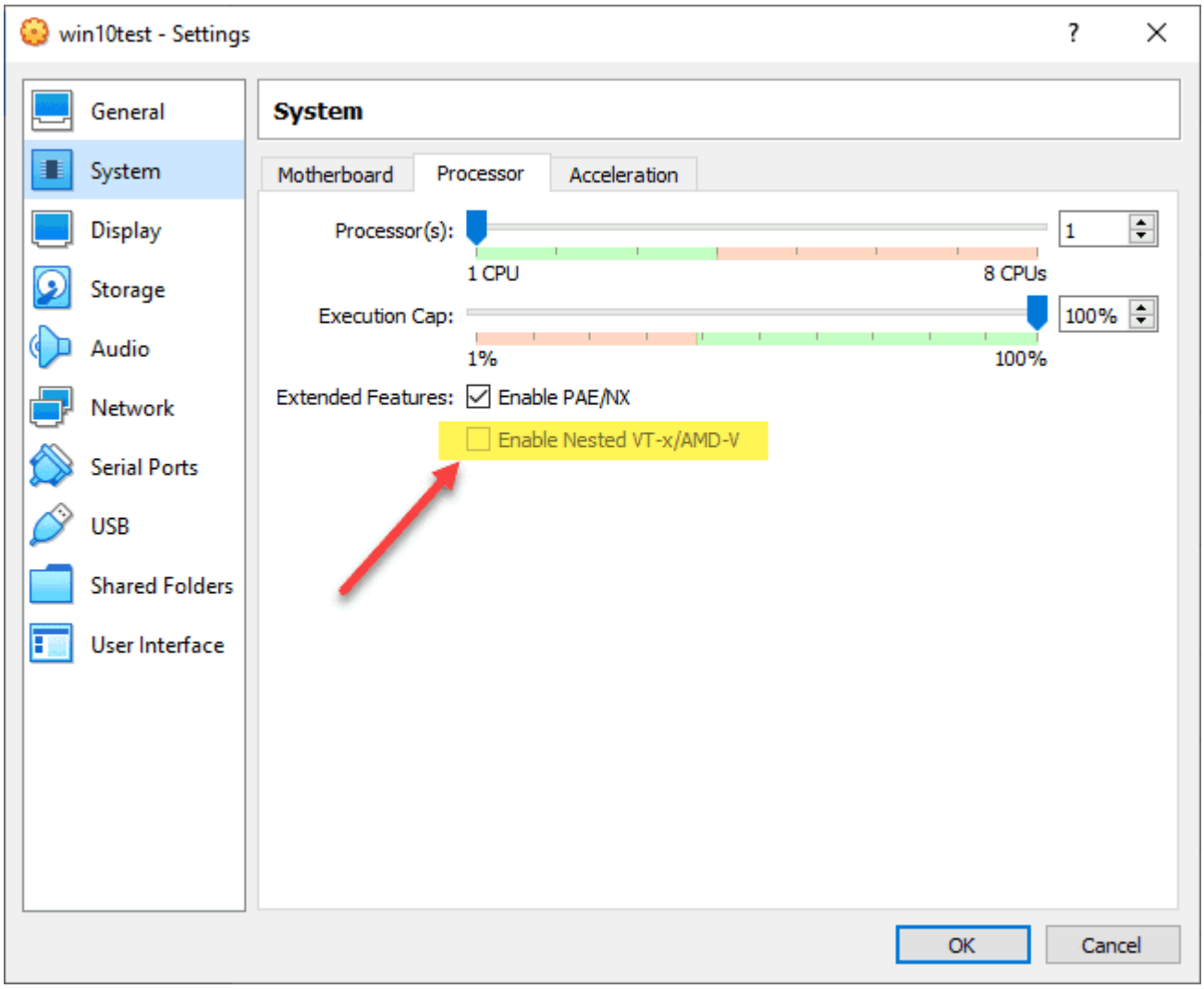
Below, we are using the **vboxmanage list vm** command to list out the VMs that are running inside VirtualBox. Then, you use the command:

```
vboxmanage modifyvm "<your vm name>" --nested-hw-virt on
```



Using the virtualbox command line to enable nested virtualization

As mentioned earlier, you can't enable nested virtualization for Intel CPUs using the GUI. As you can see below, the option is greyed out if you are running an Intel CPU.

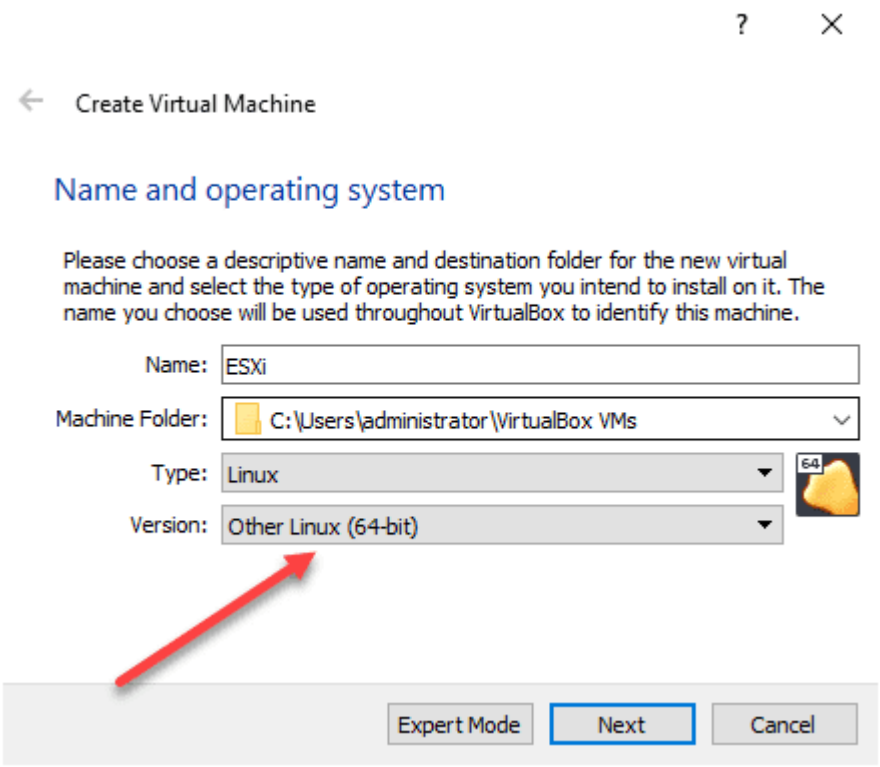


Using the virtualbox gui to enable nested virtualization

Now that nested virtualization is enabled for your VirtualBox VM, you can run a nested operating system that is virtualizing workloads underneath.

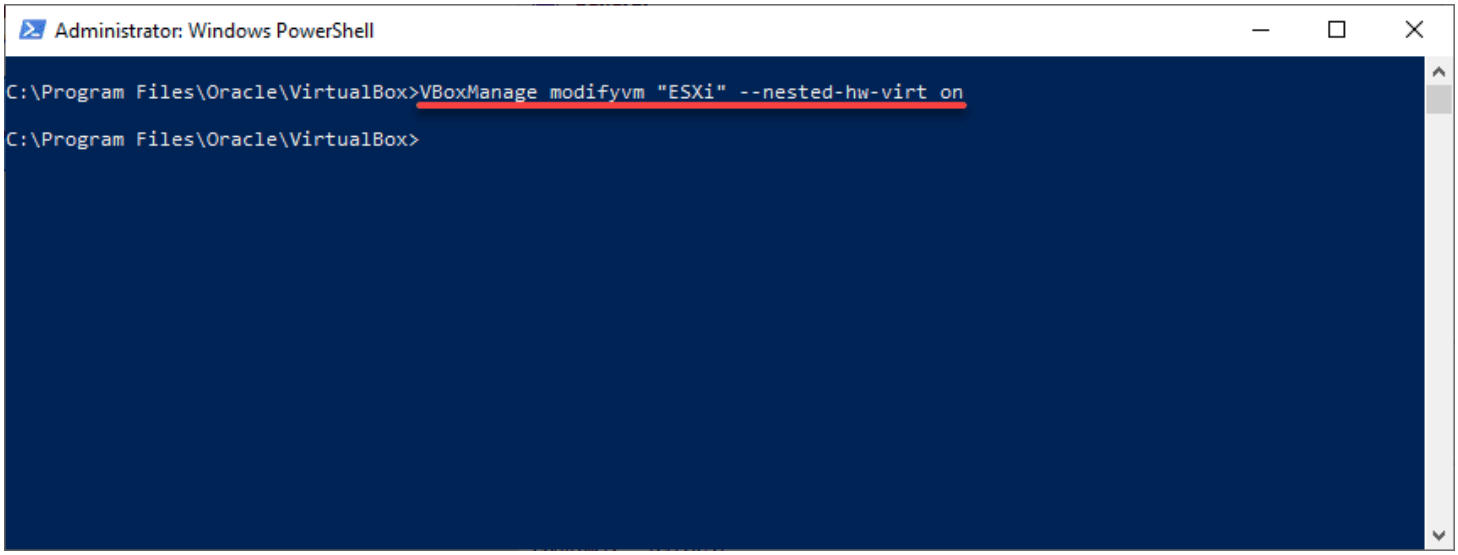
VirtualBox Nested VMware ESXi host

Since we have nested virtualization in VirtualBox fresh on the mind, let's look at nesting VMware ESXi inside a VirtualBox VM. The first thing we need to do is create a new VirtualBox virtual machine for the ESXi host. Use the version of **Other Linux (64-bit)** for your ESXi host.



Create a new esxi virtual machine in virtualbox

Using the same process as mentioned above for enabling nested virtualization, you will want to set the **nested-hw-virt** flag on the newly created ESXi host.



Enabling nested virtualization for the esxi host in virtualbox

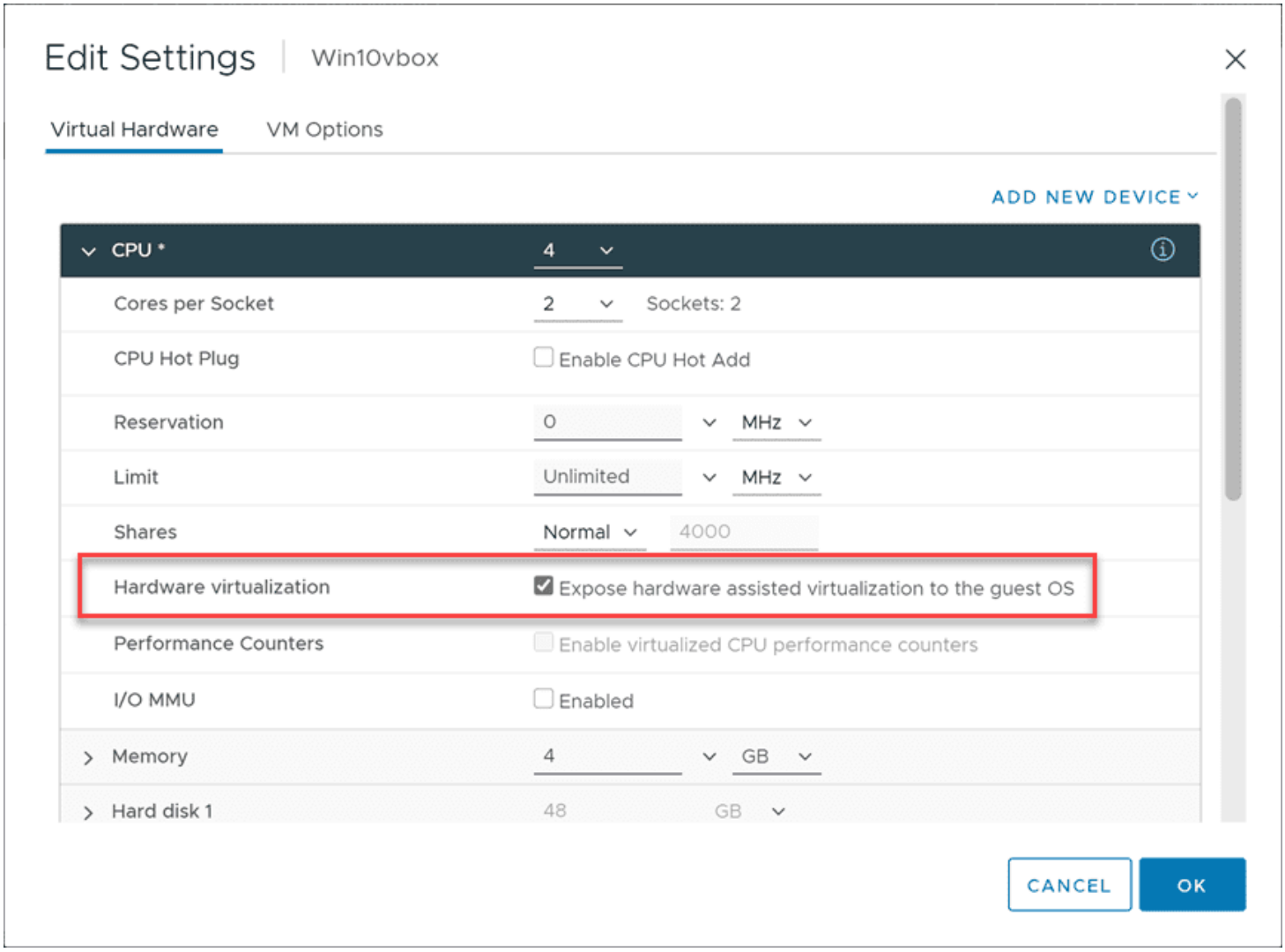
Upload your VMware vSphere ESXi media and load up the ESXi host in VirtualBox.

Nesting VirtualBox inside a VMware ESXi Virtual Machine

Let's now consider the other way around. How do you nest your VirtualBox installation inside a VMware ESXi virtual machine? This is fairly straightforward and requires the same thing from the VMware ESXi side – hardware virtualization for the CPU needs to be exposed to the guest operating system.

So, the virtual machine in VMware ESXi that you are using to house your VirtualBox installation will need to have the nested virtualization setting configured for the CPU. Under the **Virtual Hardware** configuration for the VirtualBox VM, CPU configuration, place a check in the box:

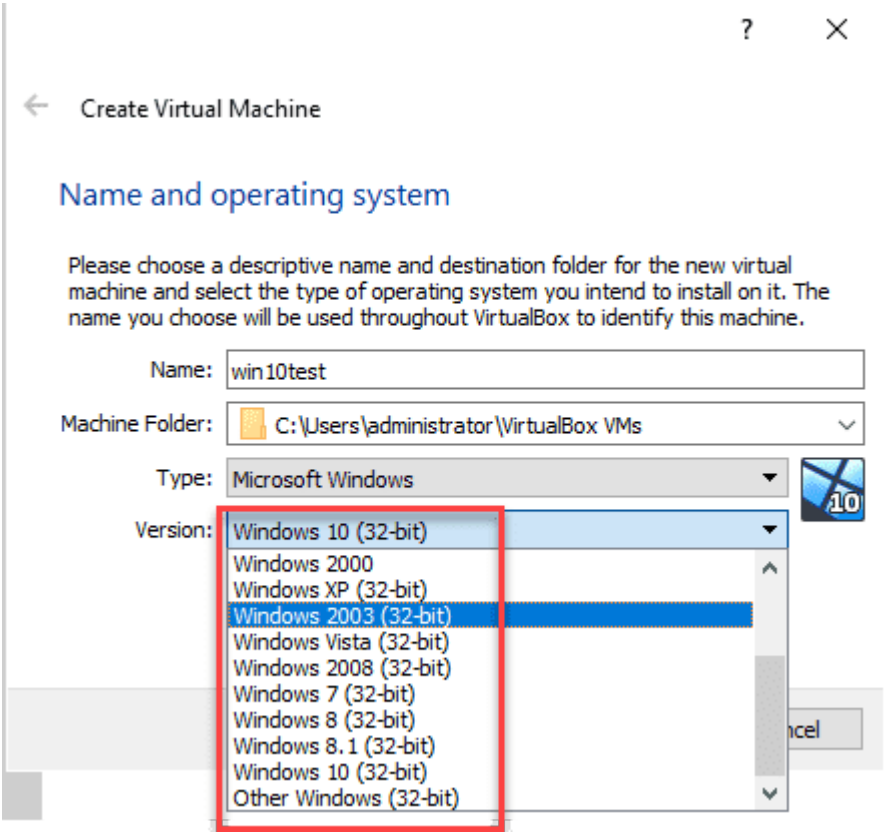
- **Expose hardware assisted virtualization to the guest OS**



Configuring the hardware virtualization setting for virtualbox nested virtualization inside vmware esxi

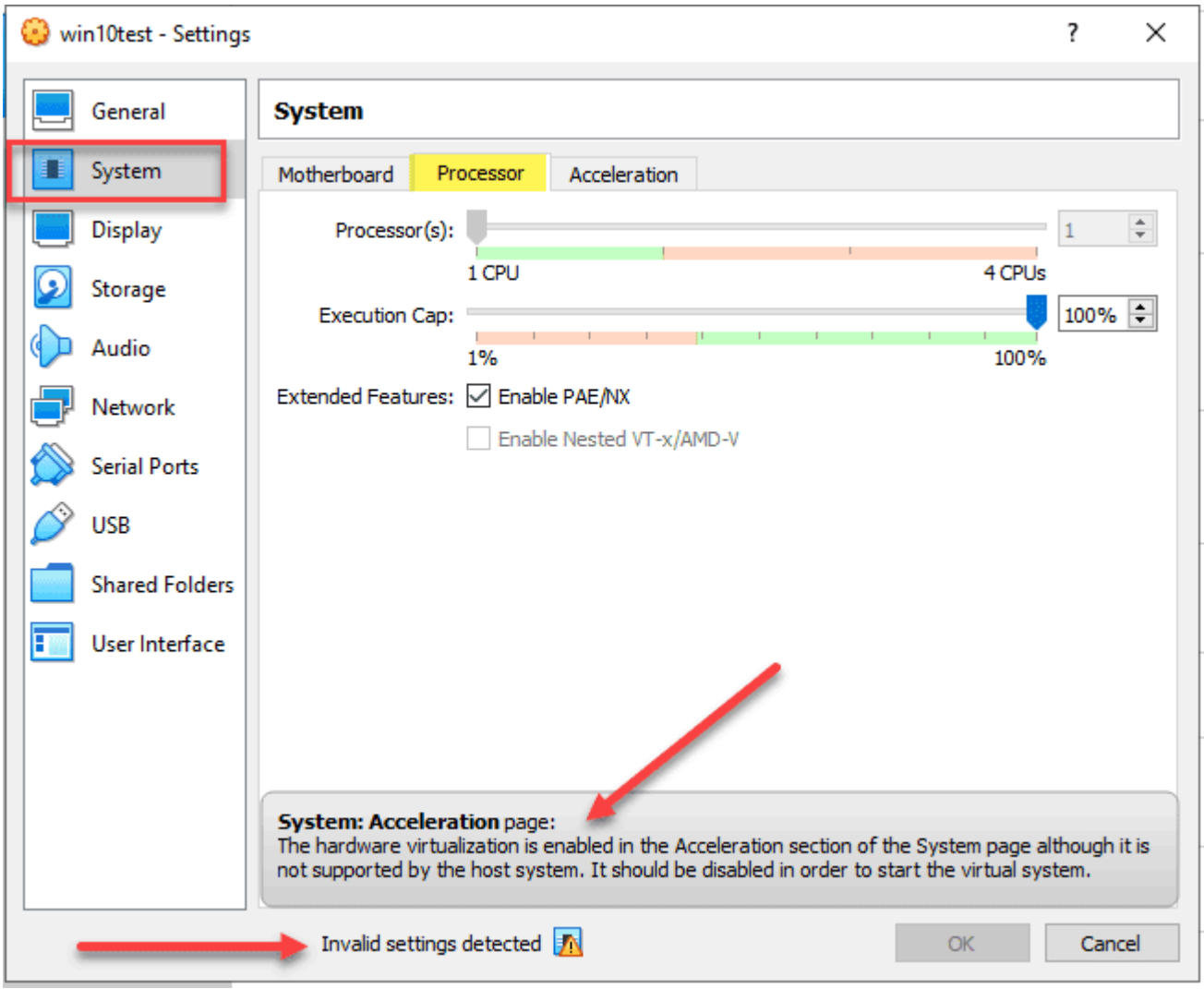
What happens if you don't configure the hardware virtualization setting on the properties of the VirtualBox virtual machine? Unlike installing Hyper-V or nested ESXi inside VMware ESXi, you won't receive an error during the installation of VirtualBox. The installation will behave the same as a virtual machine that has the setting enabled.

However, you will see a couple of odd things when you forget to enable hardware virtualization. When you go to create a virtual machine on a nested ESXi VirtualBox installation without the hardware virtualization flag configured, you will only see the options for 32-bit operating systems, and no Windows Server operating systems in the **Create Virtual Machine**.



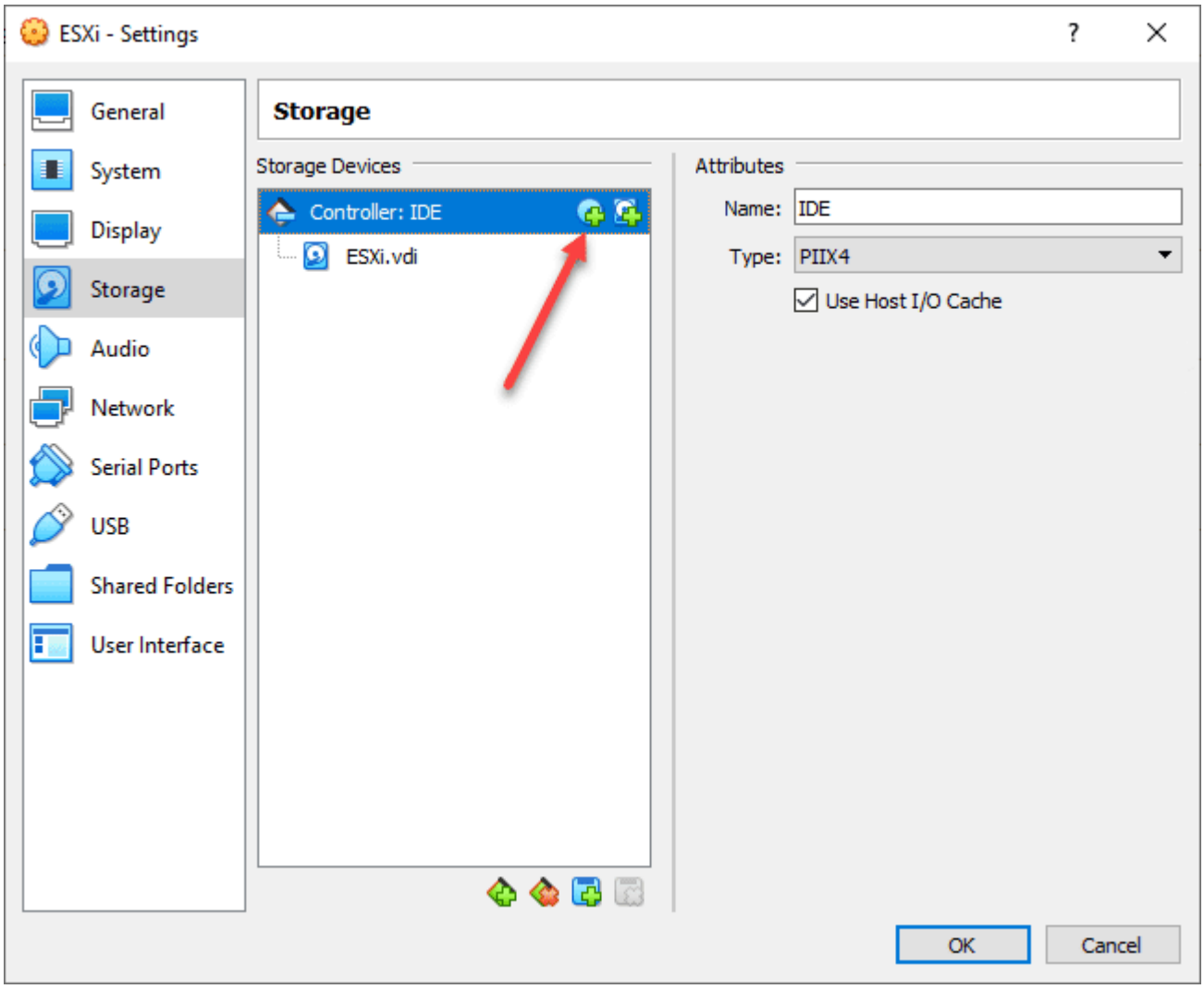
Only 32 bit operating systems are displayed and no server operating systems

Also, you will see an **Invalid settings detected** message on the System Settings screen for the CPU in VirtualBox. The note above the message details **the hardware virtualization is enabled in the cceleration section of the System page although it is not supported by the host system**. This is the tell tale sign that hardware virtualization was not enabled in VMware vSphere for the VirtualBox VM.



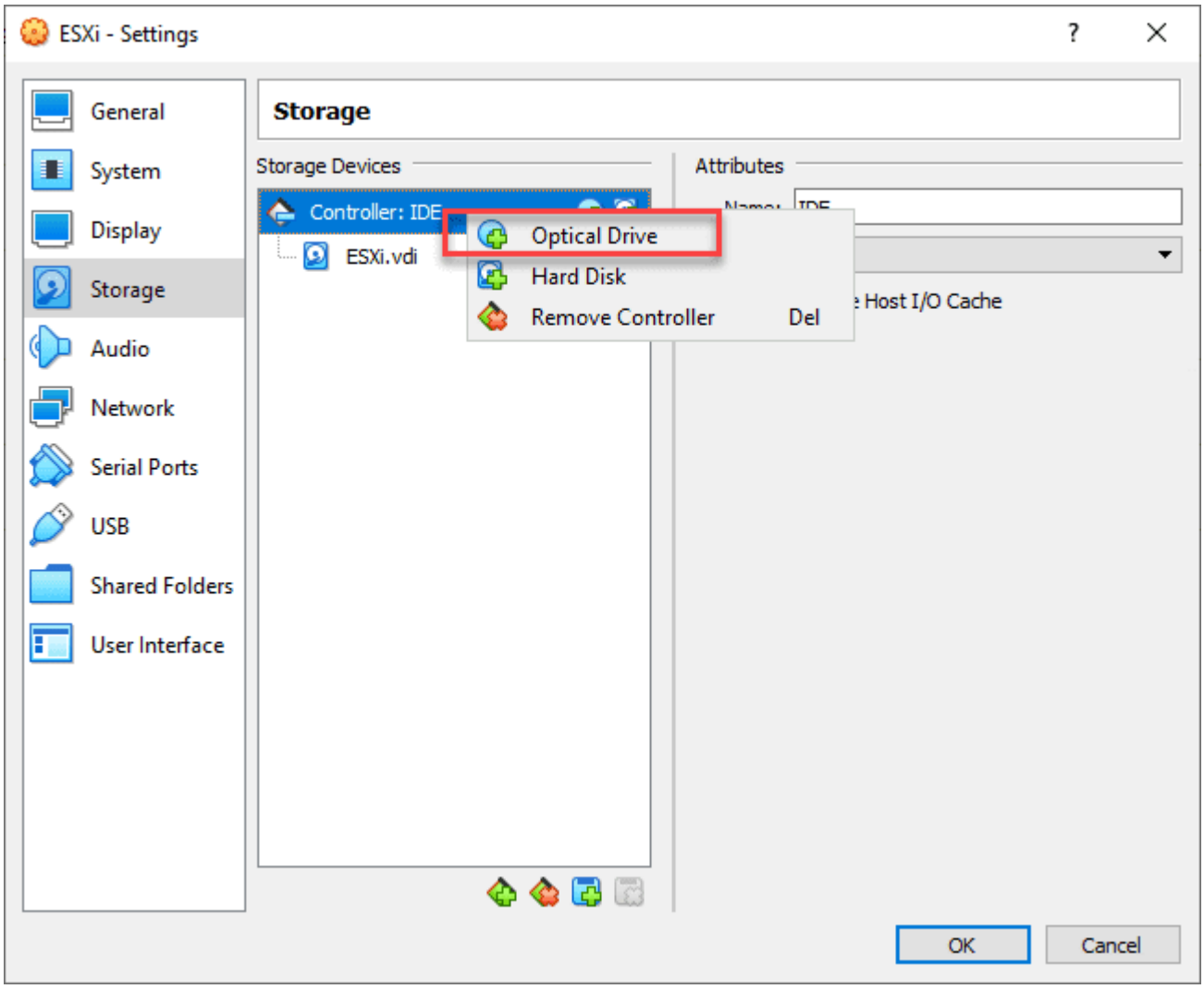
Invalid system settings error displayed in the virtualbox processor settings

Needless to say, you will want to have hardware virtualization enabled. Let's look at creating an ESXi VM in VirtualBox **nested inside a VMware vSphere ESXi Windows 10 VM**. Create a **Other Linux (64-bit)** VM. Under **Storage** Click the little Plus sign to add an optical drive. If you already have an optical drive listed, you can select to add optical media.



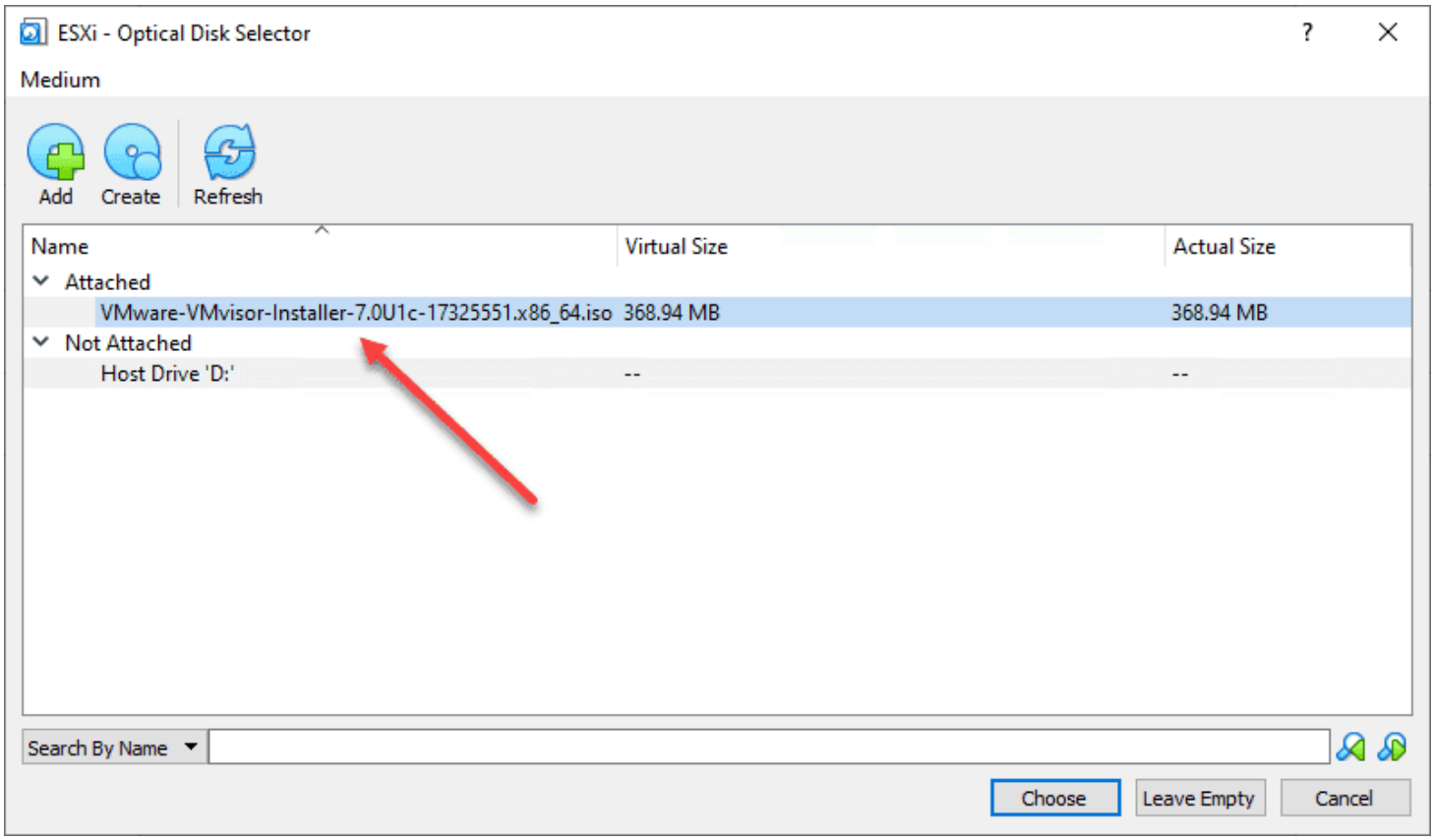
Adding an optical drive to the virtualbox vm

Select **Optical Drive**.



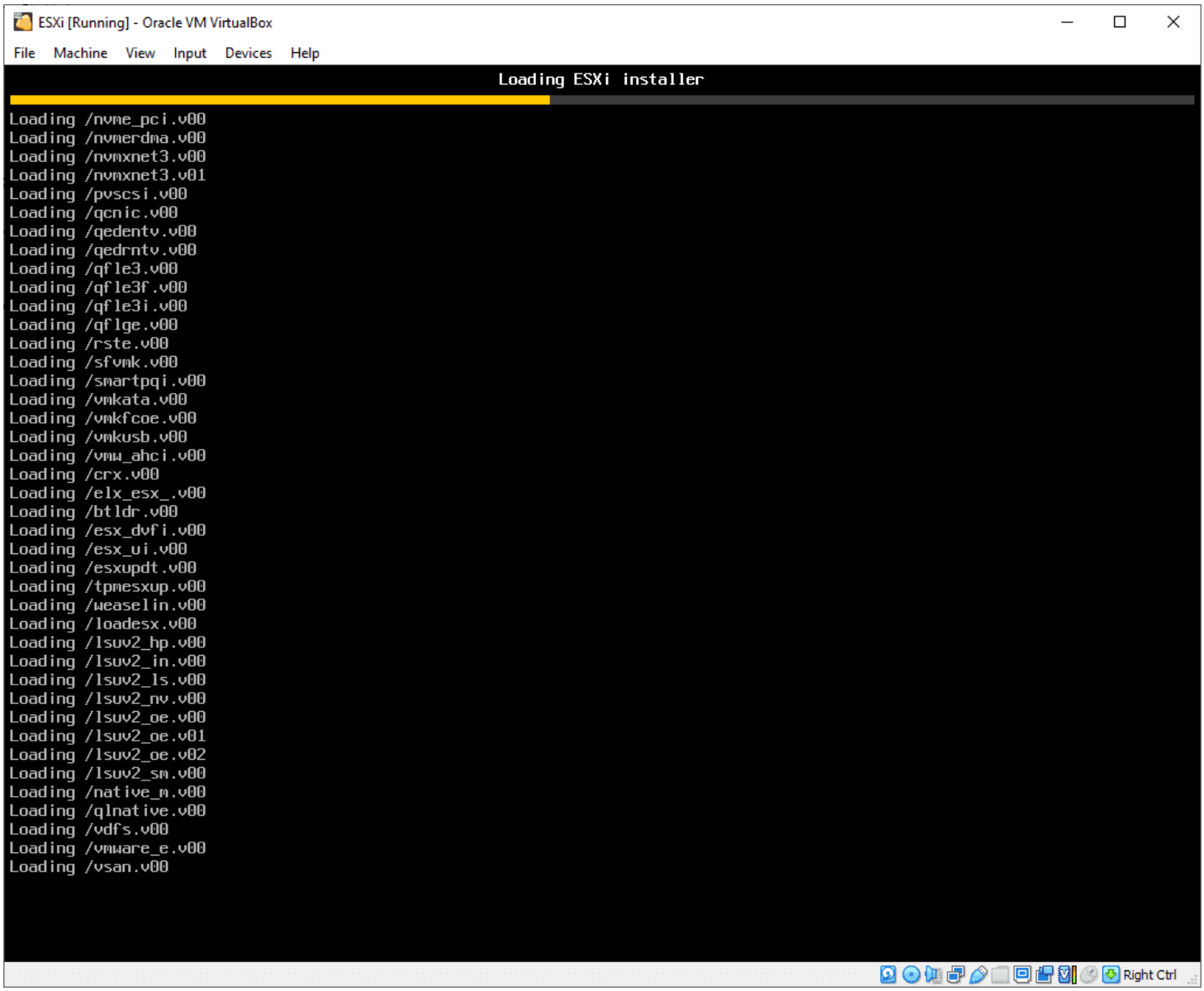
Choose optical drive for adding the esxi iso

Browse your computer and select the ESXi ISO file you downloaded from VMware.



Choose and select your esxi iso file you downloaded

After selecting the ESXi ISO file, you should be ready to power on and start loading ESXi in your VirtualBox VM.



Powering on and booting your esxi vm in virtualbox

Why would you nest VirtualBox inside ESXi?

The obvious question may be why would you do this? Well, the short answer is to play around with doing cool things! However, there are some real world use cases I can think of, especially if you already have an ESXi lab. You may want to play around with VirtualBox and want to do this in a nested environment that you can easily setup in your homelab. There are probably others such as setting up networking labs and other types of interesting uses.

- Check out Virtual Box here: Oracle VM VirtualBox
- Check out VMware ESXi free here: Download VMware vSphere Hypervisor for Free

VirtualBox Nested Walkthrough Video

Take a look at the video below showing the nested configuration for VirtualBox and running this in VMware ESXi.

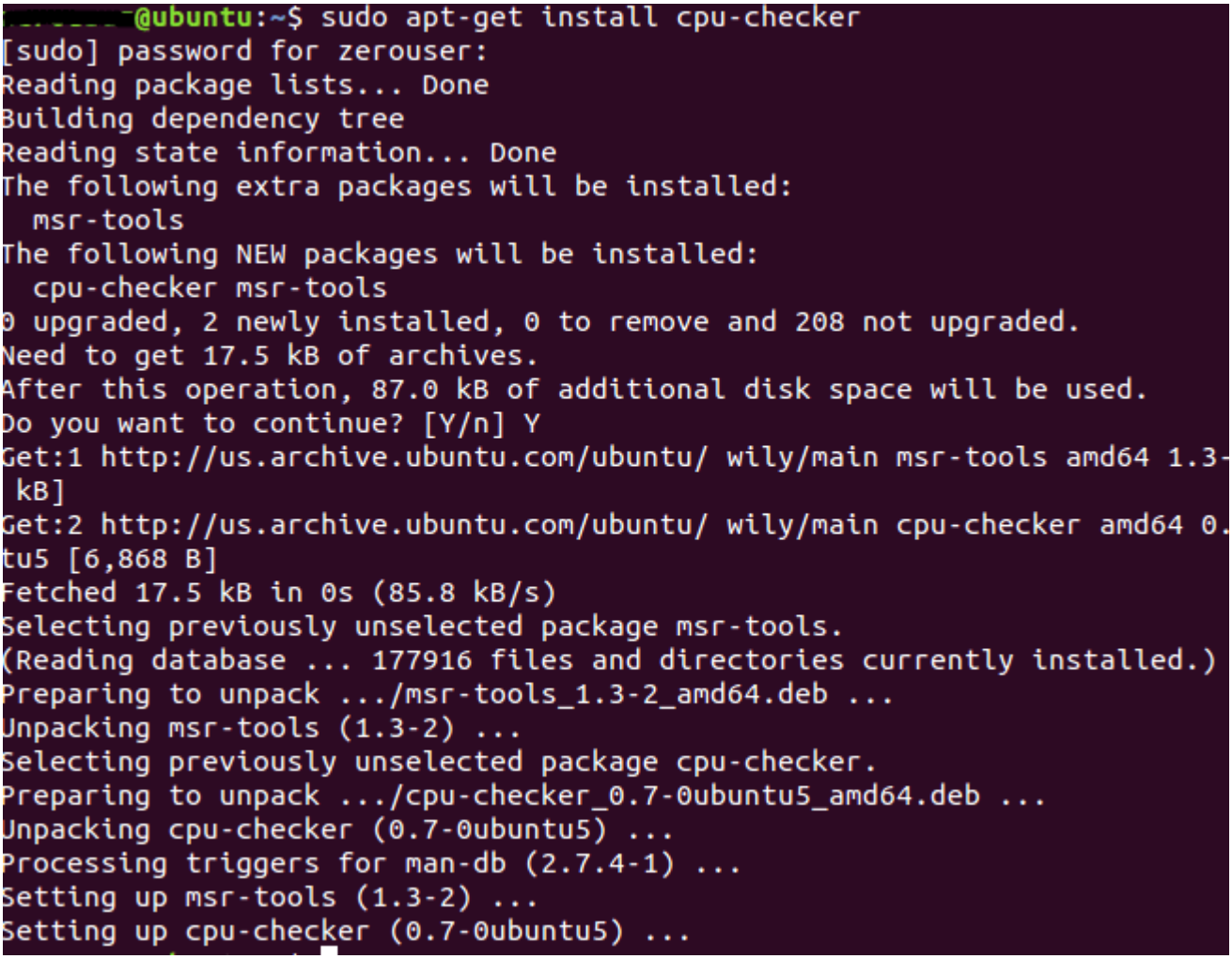
VirtualBox Nested Virtualization with VMware ESXi

https://youtube.com/watch?v=fWP0ii7gysg

VirtualBox Nested Virtualization

Install KVM Ubuntu Server VMware Workstation

April 18, 2016
Virtualization

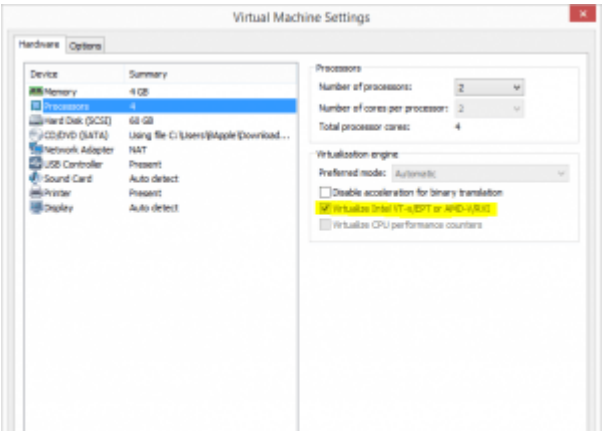


prereqs01

If you want to play around with KVM virtualization inside of VMware Workstation, this is quick and easy to do with a few steps. Let's take a look at how to install KVM Ubuntu Server VMware Workstation. The VM used for this post was an Ubuntu 14.04 64 bit server installation.

Install KVM Ubuntu Server VMware Workstation

The first thing that we need to do is properly set the virtual CPU inside of VMware Workstation so that it can perform nested virtualization. To do this, click on your "Processors" and make sure under the **virtualization engine** you have **Virtualize Intel VT-x/EPT or AMD-V/RVI** checked. This will allow KVM to function properly in the nested configuration.



Once your Ubuntu VM is installed and booted, you can check that you indeed have the KVM acceleration support enabled by issuing the following command:

```
egrep -c '(vmx|svm)' /proc/cpuinfo
```

Alternatively you can also issue the **kvm-ok** command. To do that however, you will need to install the **cpu-checker** package:

```
sudo apt-get install cpu-checker
```

```
@ubuntu:~$ sudo apt-get install cpu-checker
[sudo] password for zerouser:
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following extra packages will be installed:
  msr-tools
The following NEW packages will be installed:
  cpu-checker msr-tools
0 upgraded, 2 newly installed, 0 to remove and 208 not upgraded.
Need to get 17.5 kB of archives.
After this operation, 87.0 kB of additional disk space will be used.
Do you want to continue? [Y/n] Y
Get:1 http://us.archive.ubuntu.com/ubuntu/ wily/main msr-tools amd64 1.3-
 kB]
Get:2 http://us.archive.ubuntu.com/ubuntu/ wily/main cpu-checker amd64 0.
us5 [6,868 B]
Fetched 17.5 kB in 0s (85.8 kB/s)
Selecting previously unselected package msr-tools.
(Reading database ... 177916 files and directories currently installed.)
Preparing to unpack .../msr-tools_1.3-2_amd64.deb ...
Unpacking msr-tools (1.3-2) ...
Selecting previously unselected package cpu-checker.
Preparing to unpack .../cpu-checker_0.7-0ubuntu5_amd64.deb ...
Unpacking cpu-checker (0.7-0ubuntu5) ...
Processing triggers for man-db (2.7.4-1) ...
Setting up msr-tools (1.3-2) ...
Setting up cpu-checker (0.7-0ubuntu5) ...
```

You should see a message **kvm acceleration can be used** after issuing the **kvm-ok** command if you have the support enabled.

KVM Install:

Installing KVM is easy using this command:

```
sudo apt-get install qemu-kvm libvirt-bin ubuntu-vm-builder bridge-utils
```

You need to add your user to the **libvirtd** group. To do that, simply issue this command:

```
sudo adduser 'root' libvirtd
```

Then log out and back in.

Virtual Machine manager:

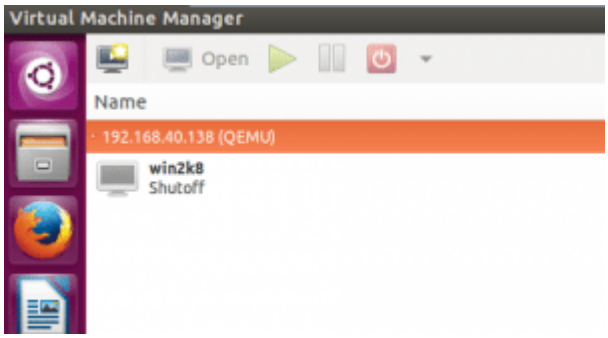
To manage my virtual machines, I used another Ubuntu 15.10 desktop machine to install the **virt-manager** package on. To install virt-manager, issue this command:

```
sudo apt-get install virt-manager
```

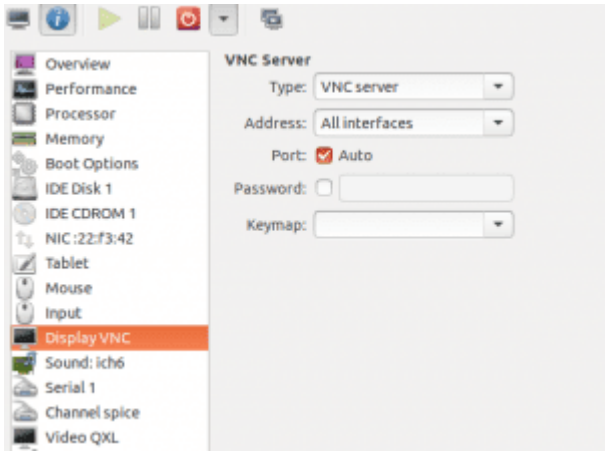
Now you are ready to start playing around with creating some Virtual machines.

Windows VM

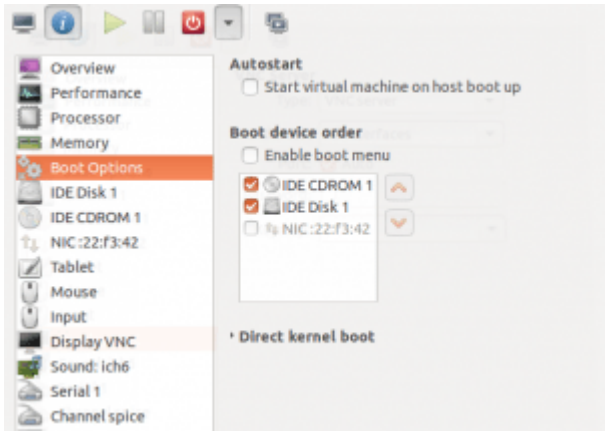
I played around with a Windows 2012 R2 VM as the first VM that I created in the KVM environment. This was fairly straightforward. After you launch Virtual Machine Manager, just click the button at the top left to “create a new virtual machine” and follow the prompts to assign storage, networking, and compute resources. As you can see below I have the Windows server VM ready to go.



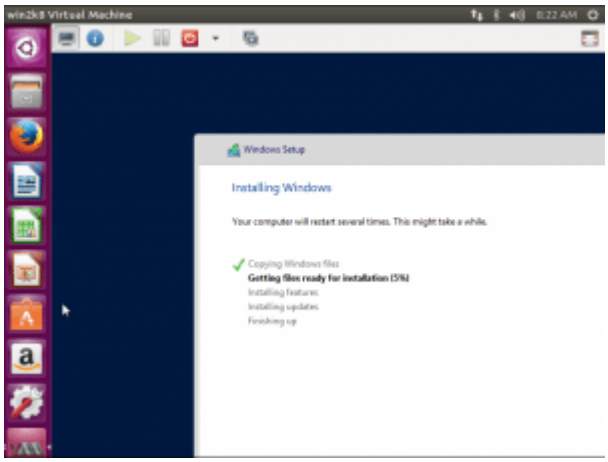
I had an issue right at first where I couldn’t see the console. I changed the Console display type to VNC Server and this seemed to resolve that issue.



Also, the VM wouldn’t boot from the ISO I had mounted at first. However, this is resolved under the **Boot Options** as you have to place a **check** by the Boot device for IDE CDROM to actually activate it.



After doing the above, I was able to get a Windows 2012 R2 VM up and running and then successfully install the OS.

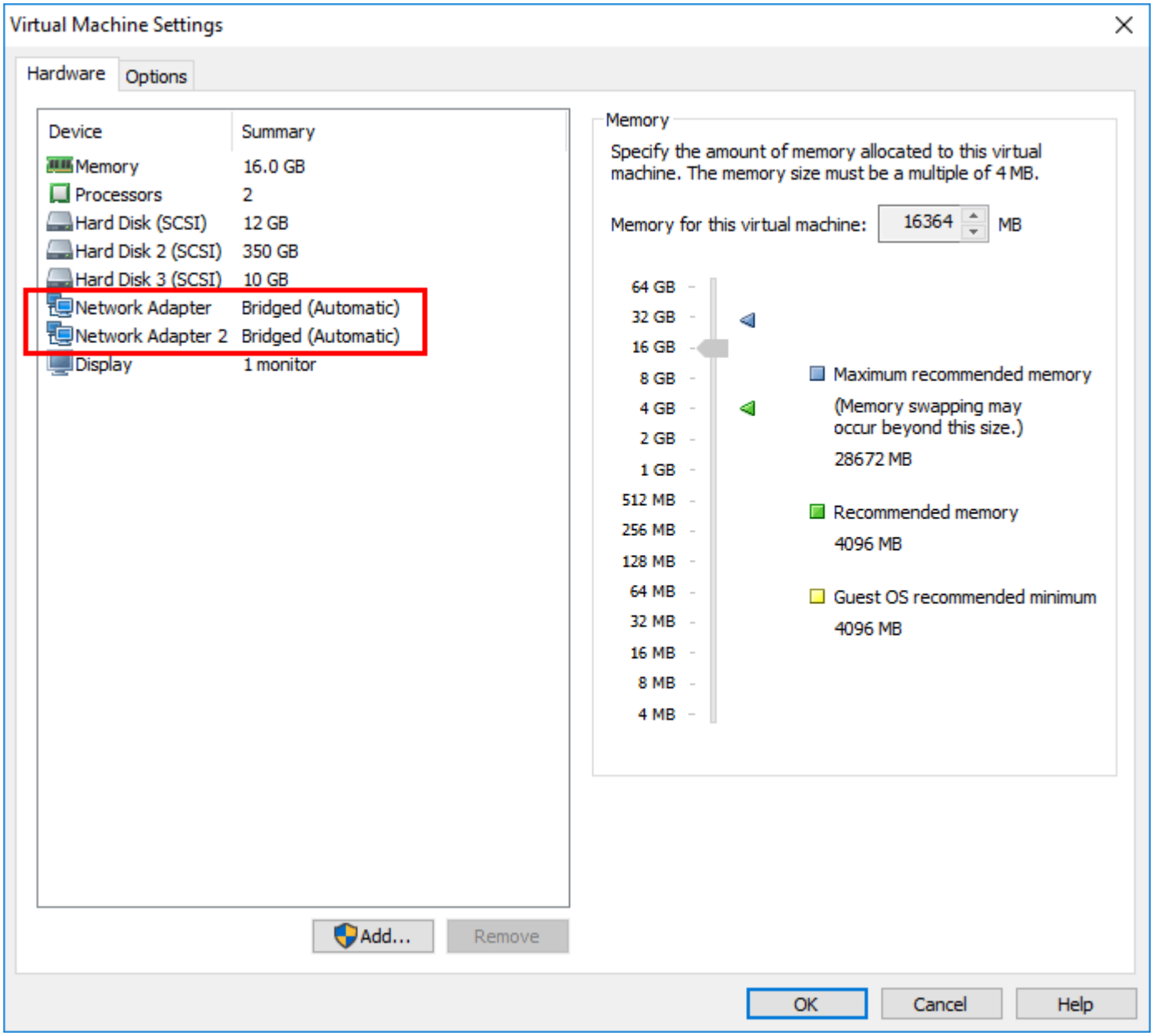


Final Thoughts

KVM is still a maturing technology but is widely available in most Linux distros. Hopefully this post on how to Install KVM Ubuntu Server VMware Workstation can help most with getting a lab version of KVM up and running inside of VMware Workstation.

Install and Configure VMware vSAN Witness Appliance in VMware Workstation

January 30, 2018
vSAN



A-look-at-the-configured-virtual-hardware-for-the-VMware-vSAN-Witness-Appliance

Update 1.30.2018 – Password issues with the vSAN Witness Appliance on legacy VMware Workstation versions

One of the really great features starting in VMware vSAN 6.1 was the introduction of the the new 2-node and stretched cluster implementations of vSAN. Both of these implementations are made possible by a new VMware vSAN component called the vSAN Witness Appliance. To borrow a thought from Windows Failover Clustering, the witness appliance provides the "quorum " or greater than 50% rule where you don't run into a "split brain" scenario. This allows making sure that greater than 50% of the components which make up a virtual machine's storage object are available. In revamping some of my home lab resources and turning up a 2-node vSAN cluster, I wanted to spin up a vSAN Witness Appliance in VMware Workstation as this allowed me to have the witness appliance running on separate physical hardware. Let's take a look at how to install and configure VMware vSAN Witness Appliance in VMware Workstation.

What is the VMware vSAN Witness Appliance?

Just a little background on the vSAN Witness appliance and how it is obtained from VMware. The vSAN Witness Appliance is a full-blown ESXi host that is obtained via an OVA appliance download from VMware. The nice thing is if you use the virtual appliance for the vSAN Witness Appliance, the ESXi host is fully licensed and supported from VMware in this configuration. So you don't have to provide your own license. Traditionally, ESXi has not been supported in a nested virtualization format from VMware. This changes that stance however.

The nested VMware vSAN ESXi host that serves as the vSAN Witness Appliance, stores vSAN metadata for the virtual machines running on the vSAN datastore. This provides the "witness" disk functionality for the virtual machines and provides an availability mechanism to those virtual machines. Additionally, the vSAN Witness Appliance doesn't require nearly the performance, capacity, or bandwidth of a "regular" hosts that provide the compute/memory and storage of the vSAN datastore.

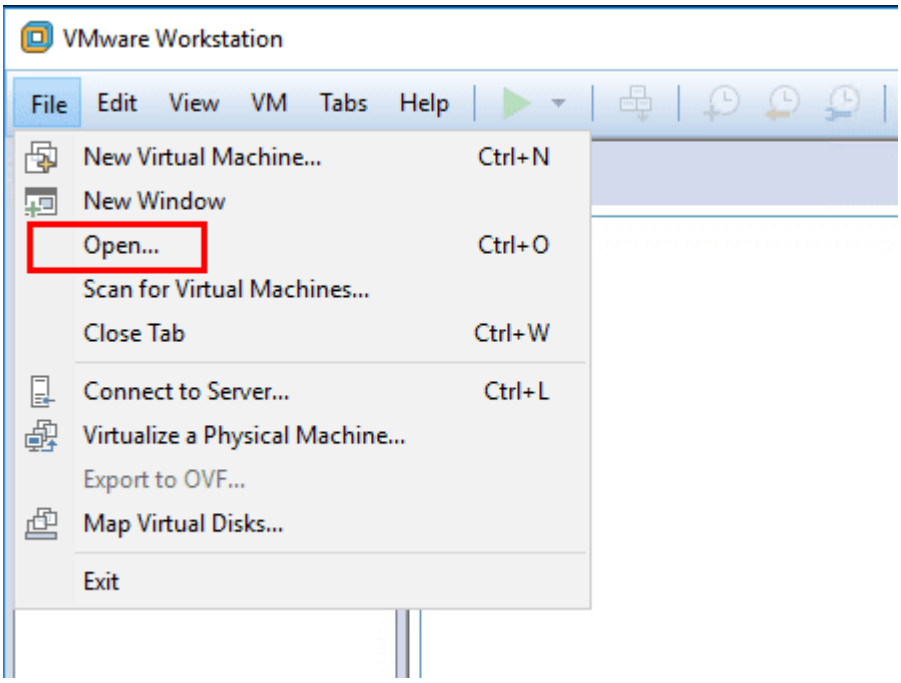
I recommend reading the following Cormac Hogan's resources on VMware vSAN Witness Appliance:

- A closer look at the VSAN Witness Appliance
- Step-by-step deployment of the VSAN Witness Appliance

Install and Configure VMware vSAN Witness Appliance in VMware Workstation

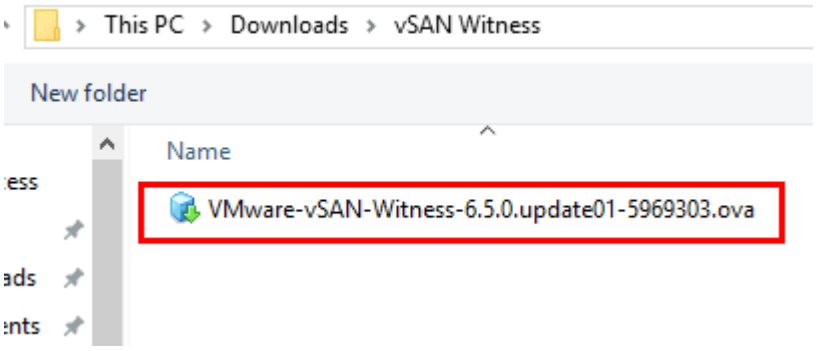
In my home lab environment, I have a standalone server outside of my 2-node ESXi cluster that serves other purposes for the lab environment. I wanted to utilize this server for the vSAN Witness Appliance version 6.5 U1 by using VMware Workstation 14. The steps to get up and running with the VMware vSAN Witness Appliance in VMware Workstation is extremely easy and is the same straightforward process you would use for any OVA appliance import.

After downloading the OVA appliance file, we will **Open** it in VMware Workstation.



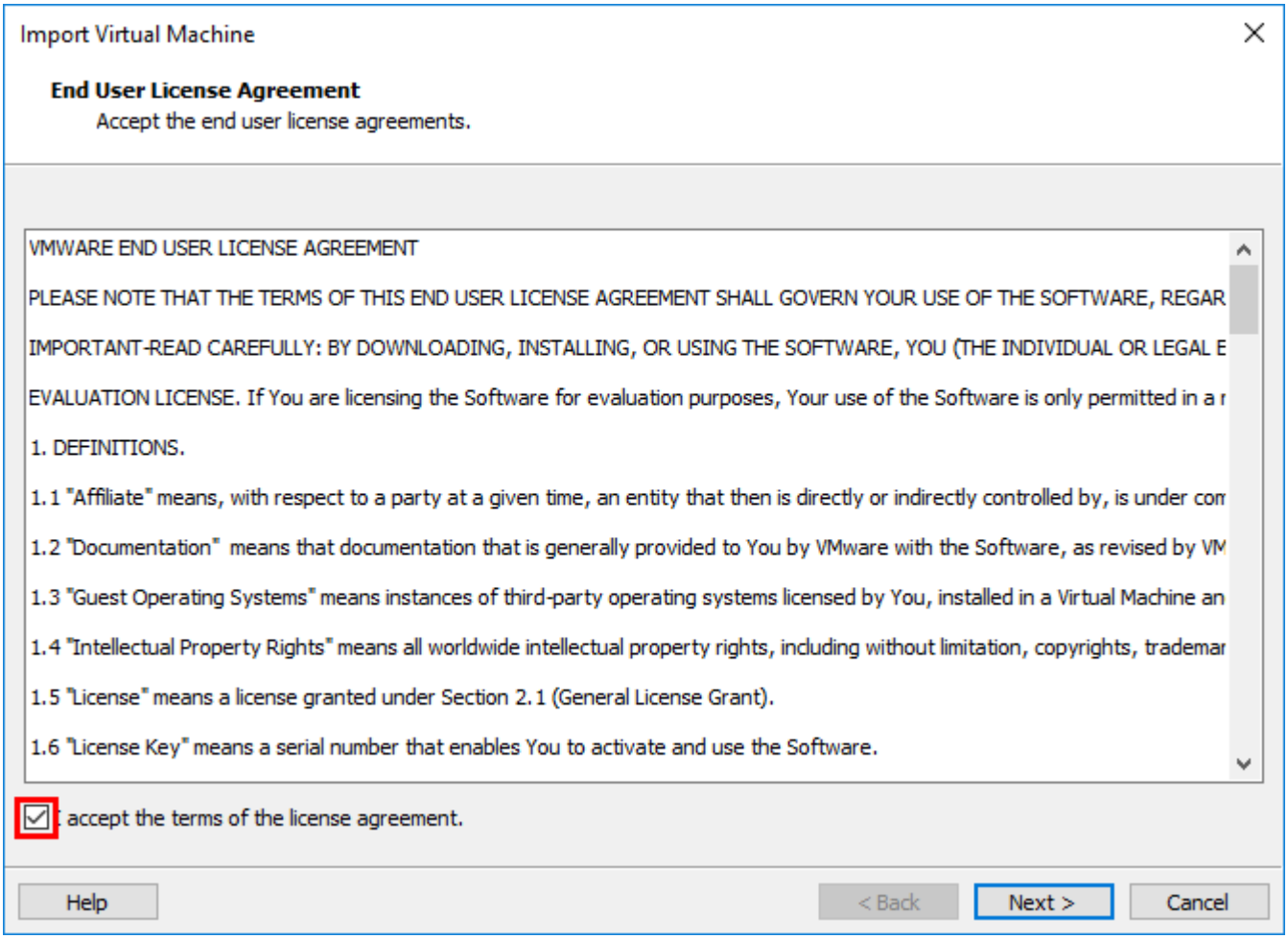
Open the VMware vSAN Witness Appliance in VMware Workstation

Here, we browse out to the downloaded OVA appliance file from VMware and choose this to import into VMware Workstation.



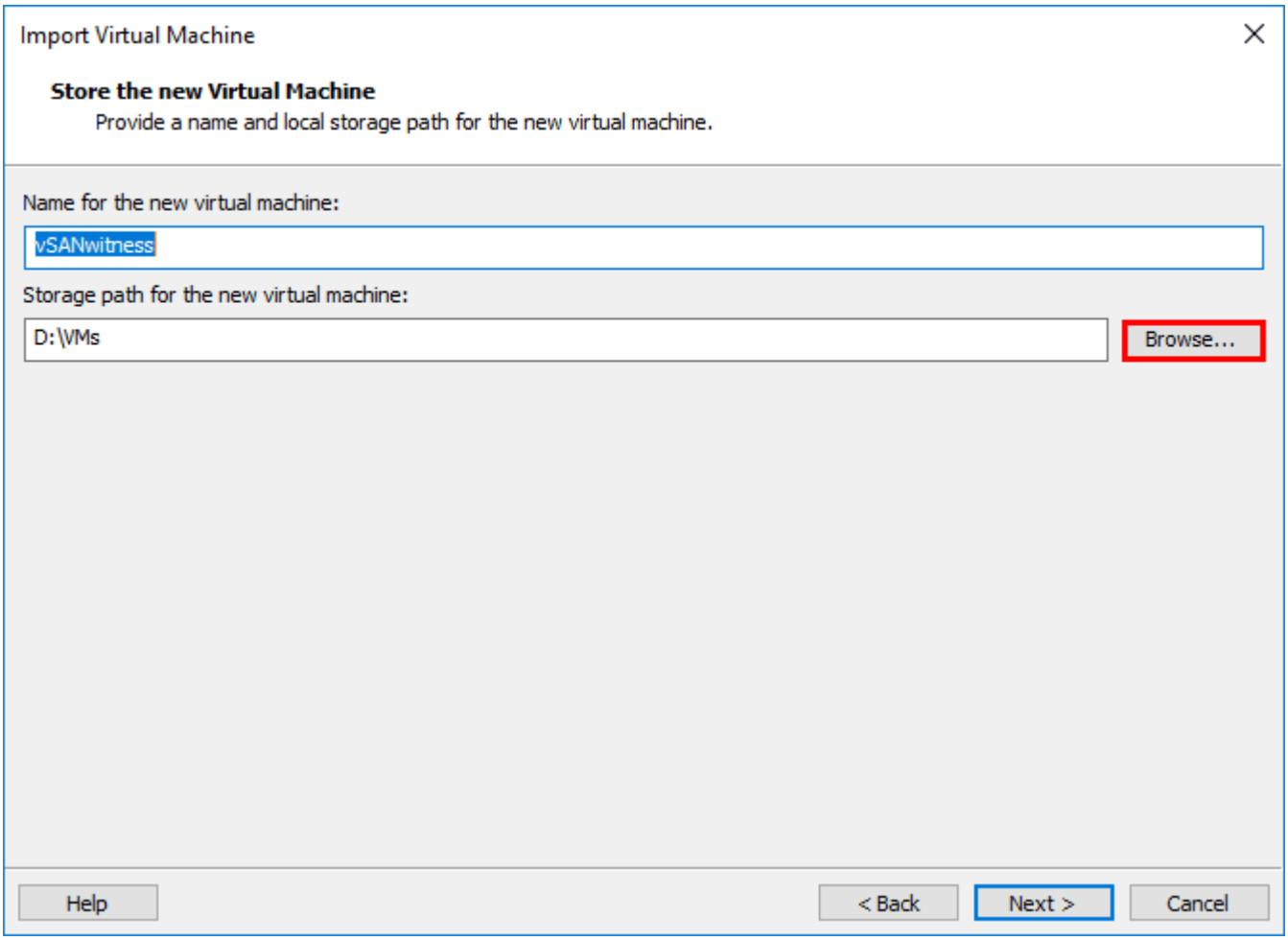
Choose the VMware vSAN Witness Appliance OVA file to import in VMware Workstation

Next, we need to accept the EULA of the VMware vSAN Witness Appliance that is shown.



Accept the EULA for the VMware vSAN Witness Appliance

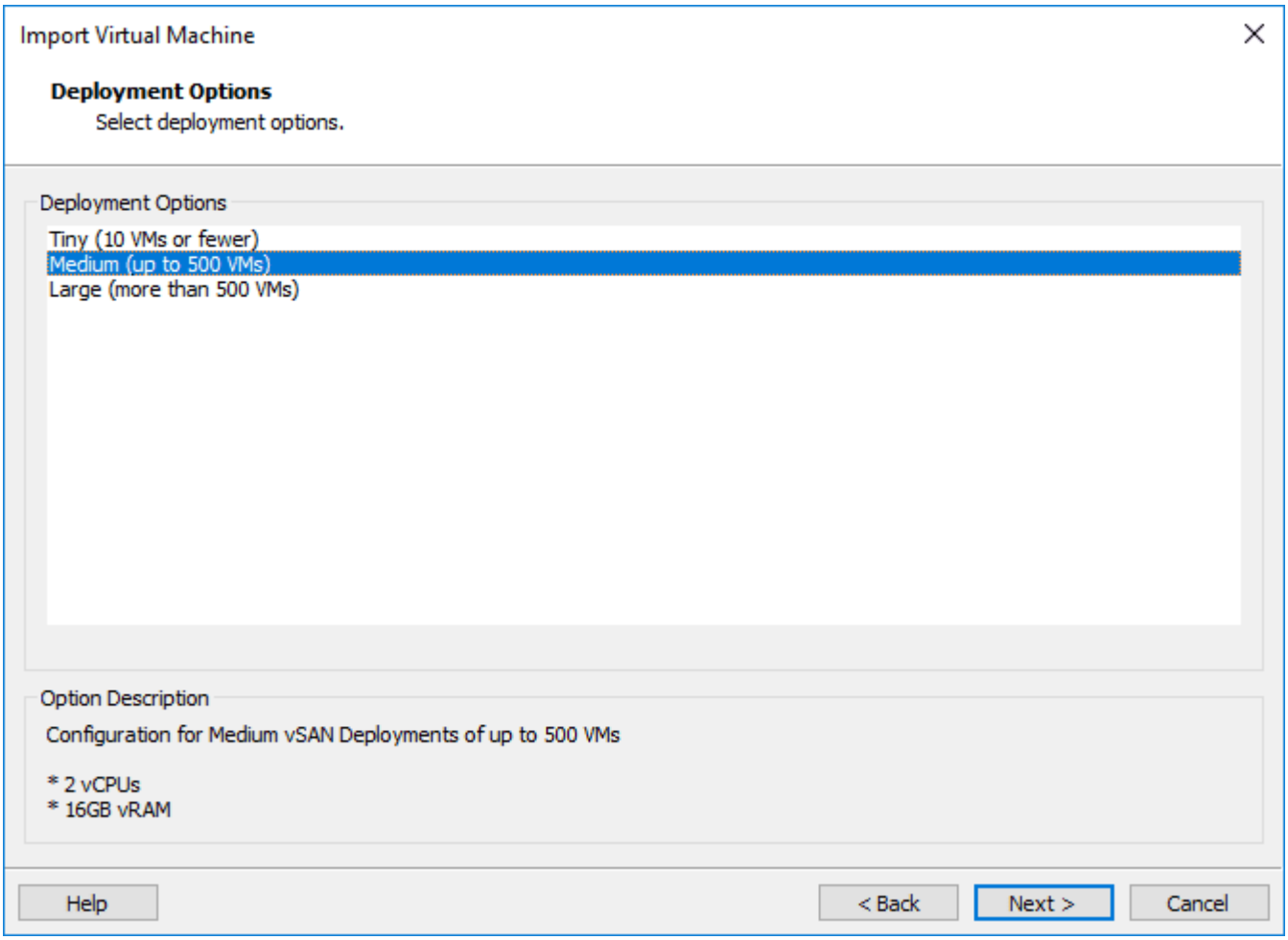
Name the vSAN Witness Appliance for the VMware Workstation inventory. Also, choose the location to store the virtual machine files for the vSAN Witness Appliance.



Choose where you want to store the VMware vSAN Witness Appliance VM files

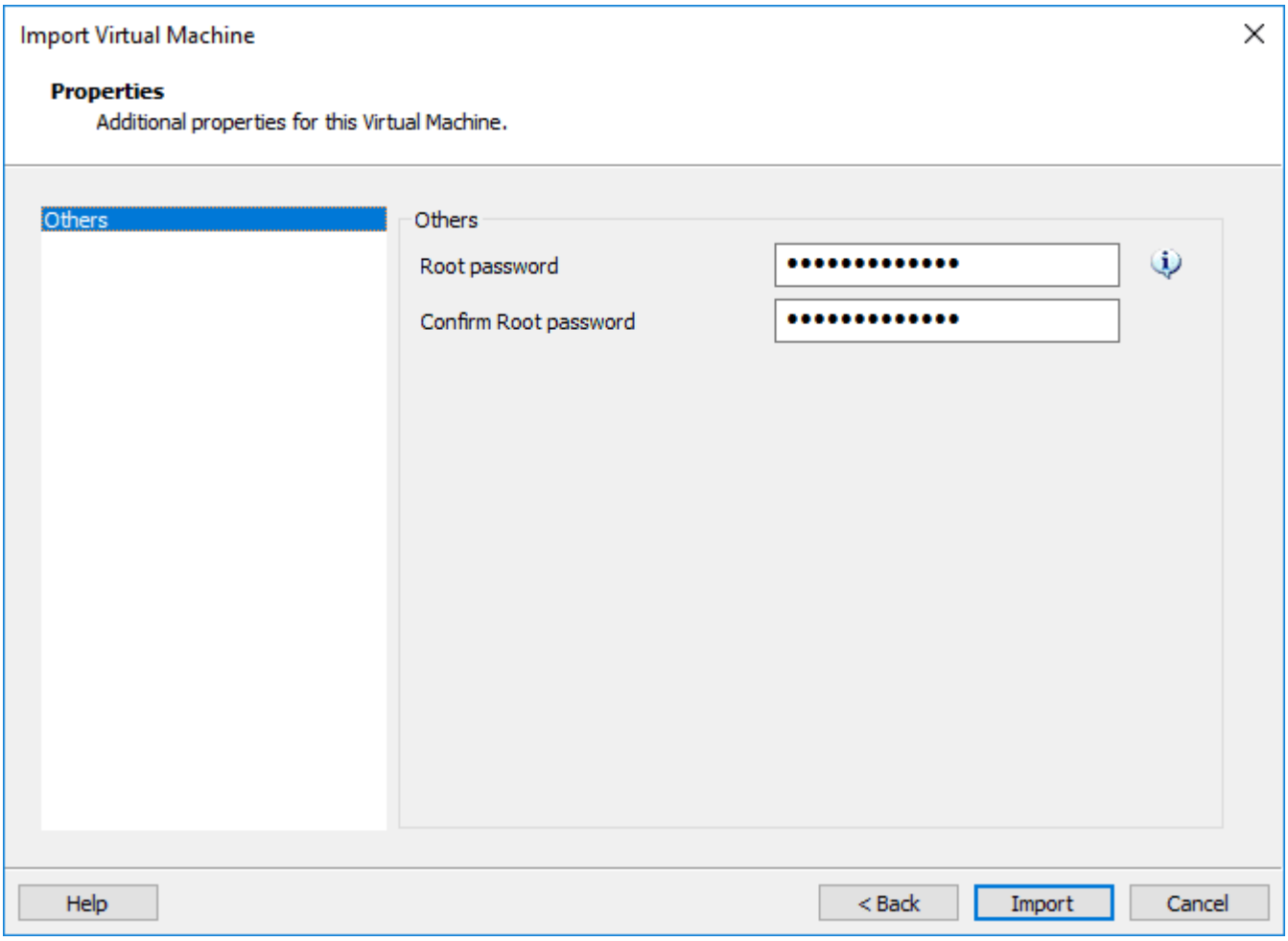
By default, on the **Deployment Options** screen, you will see the sizing options for the appliance. A brief sizing guideline is given beside each sizing option by way of "number of VMs". A further breakdown of the sizing options is as follows:

- **Tiny** (10 VMs and fewer) – 2 vCPUs, 8GB RAM, 8GB ESXi boot volume, one 10GB SSD, and one 15GB HDD
  - Maximum of 750 witness components
- **Medium** (up to 500 VMs) – 2 vCPUs, 16GB RAM, 8GB ESXi boot volume, one 10GB SSD, and one 350GB HDD
  - Maximum of 21,000 witness components
- **Large** (more than 500 VMs) – 2 vCPUs, 32GB RAM, 8 GB boot volume, one 10GB SSD, three 350GB HDDs
  - Maximum 45,000 witness components



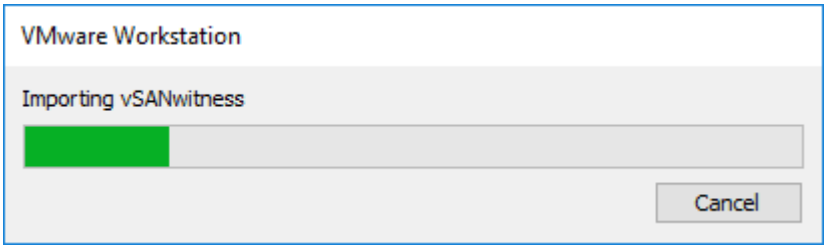
Choose the VMware vSAN Witness Appliance sizing in deployment options

Next, we set the root password for the appliance.



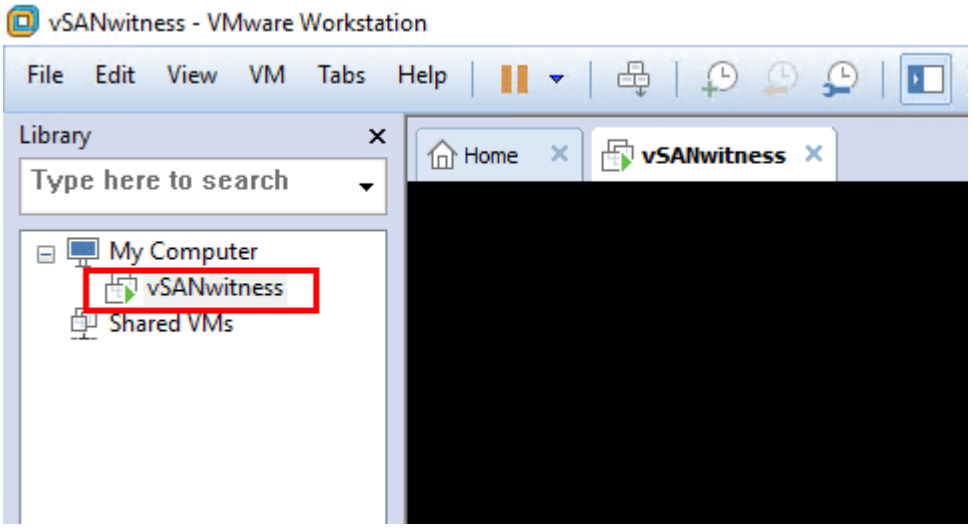
Set the VMware vSAN Witness Appliance root password

After clicking **Import**, the process of deploying the appliance into VMware Workstation begins.



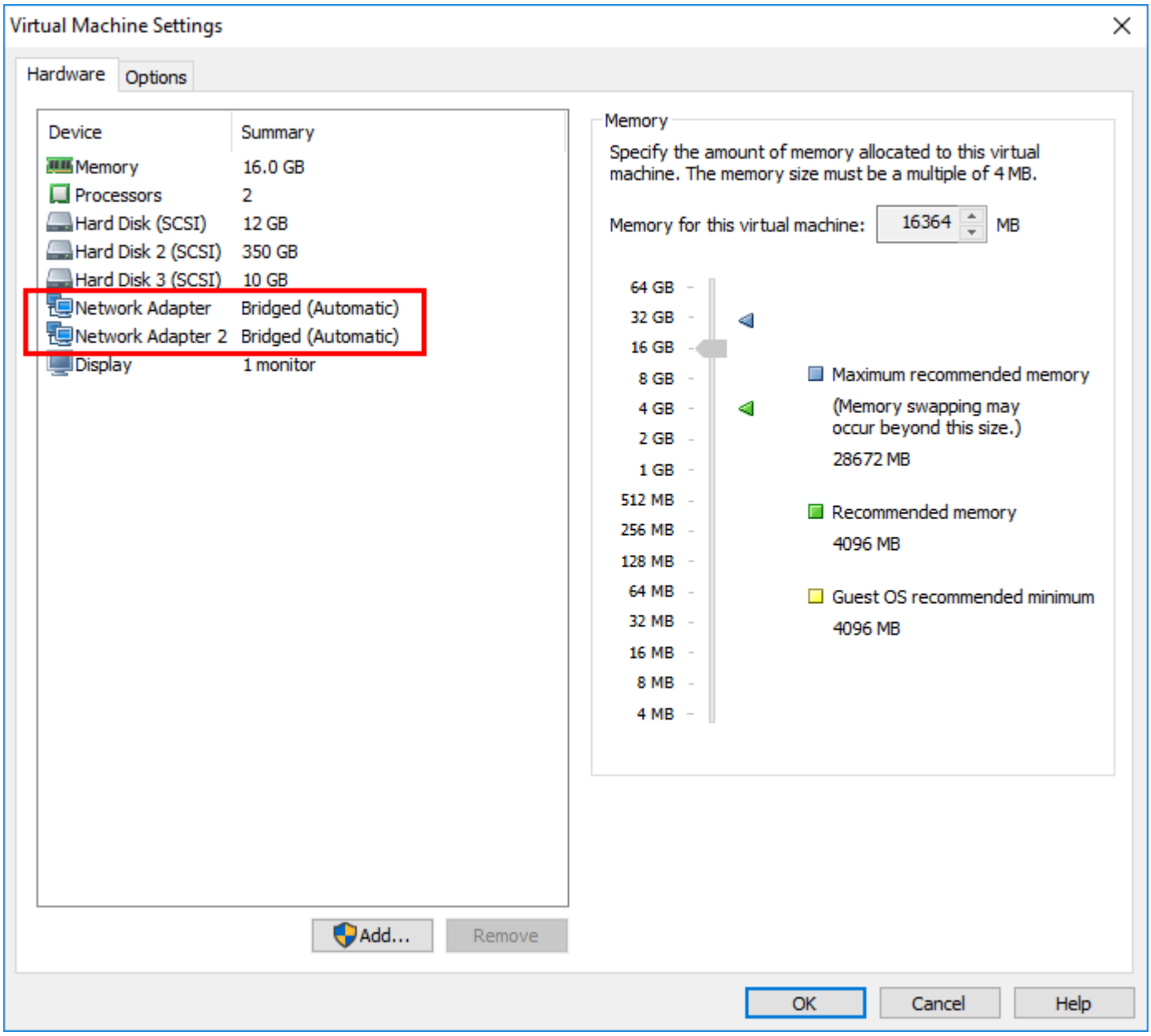
The import of the VMware vSAN Witness Appliance OVA appliance into VMware Workstation begins

The appliance import is successful and we can now boot the appliance for configuration.



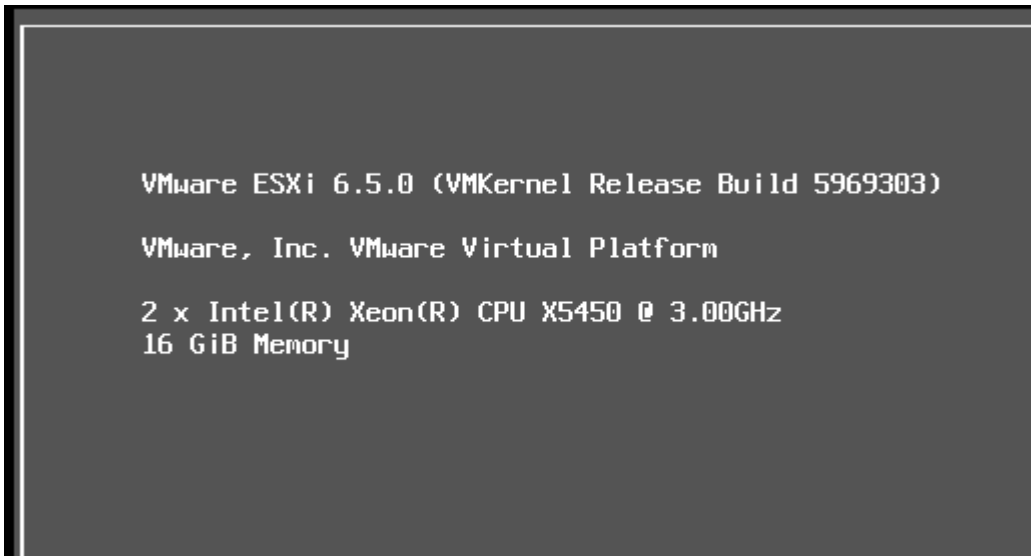
After importing the VMware vSAN Witness Appliance we power it on

A quick look at the hardware that is assigned by default in the **Medium** configuration.



A look at the configured virtual hardware for the VMware vSAN Witness Appliance

The appliance boots and as we can see, it is a nested **ESXi** host.



The VMware vSAN Witness Appliance is a nested copy of an ESXi host

VMware vSAN Witness Appliance Networking Configuration

The VMware vSAN Witness Appliance has a pretty unique network configuration directly from VMware. As Cormac describes in his blog series,

> the vmnic MAC address inside of the nested ESXi host matches the network adapter MAC address of the witness appliance (the outer MAC matches the inner MAC, so to speak). This means that we do not need to use promiscuous mode.

It is clear that VMware has thought through the "nested" virtualization configuration utilized here so there are no hoops to jump through. So the two vSwitches that come preconfigured allow the necessary communication both from a management perspective as well as a through a vSAN perspective. Keep in mind, if you deploy the vSAN Witness Appliance in VMware Workstation or elsewhere, it will need the plumbing necessary to be able to communicate with the vSAN network (i.e. associated VLANs, trunking, etc).

Update 1.30.2018 – Password issues with the vSAN Witness Appliance on legacy VMware Workstation versions

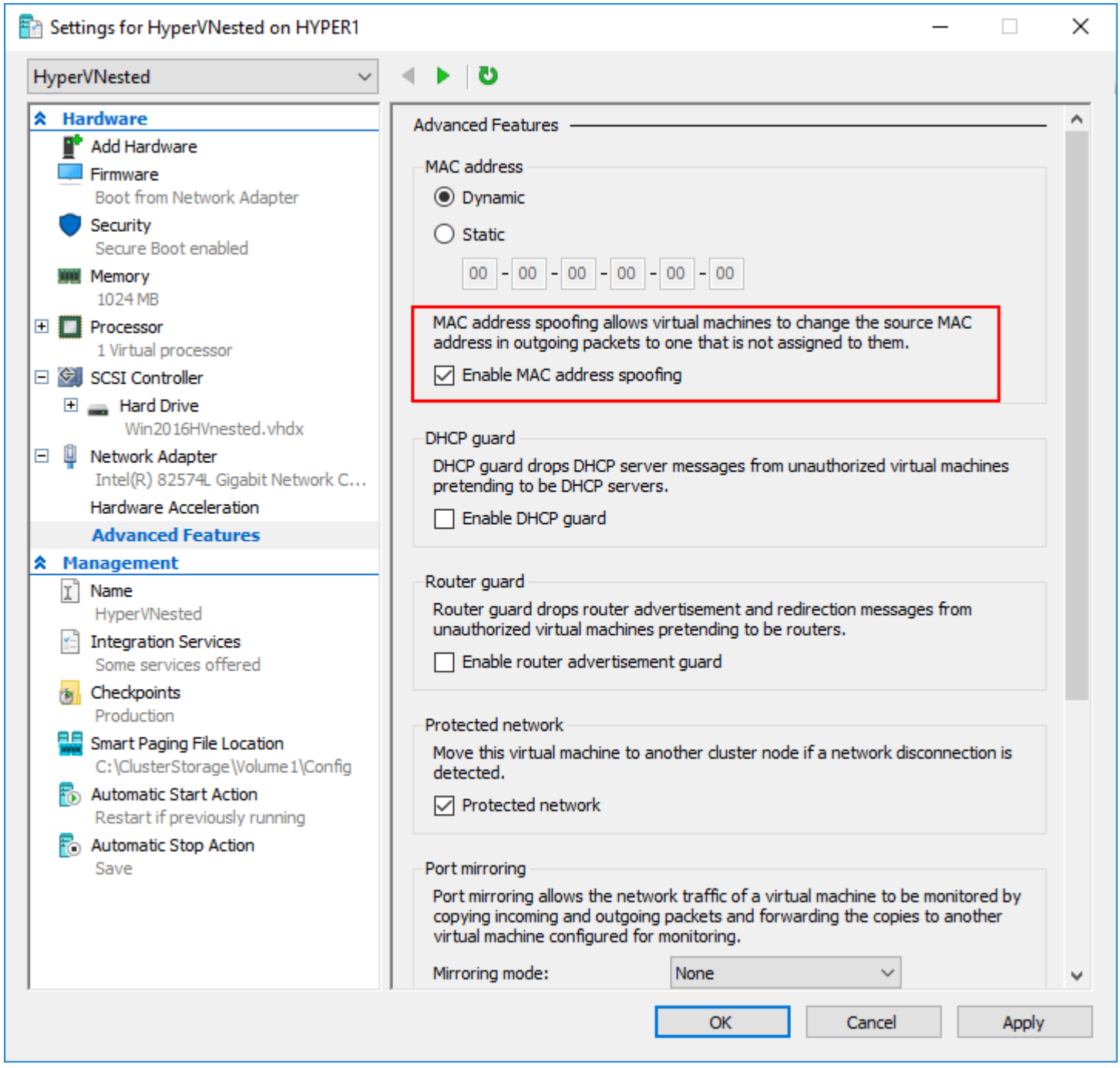
Using the method above, the vSAN appliance will deploy on VMware Workstation versions prior to Workstation 14, however, if you are using legacy workstation versions, you will need to follow, William Lam's workaround found here: https://www.virtuallyghetto.com/2015/09/how-to-deploy-and-run-the-vsan-6-1-witness-virtual-appliance-on-vmware-fusion-workstation.html

Thoughts

The VMware vSAN Witness Appliance is certainly a unique offering from VMware both from a licensing and architecture standpoint. The process to install and configure VMware vSAN Witness Appliance in VMware Workstation is very straightforward and effective, especially for home lab use. More to come on this implementation in the home lab environment.

VMware vs Hyper-V Nested Virtualization

July 23, 2018
Hyper-V



Configuring-MAC-address-spoofing-with-Hyper-V-nested-virtualization

Nested virtualization provides tremendous value for virtualization administrators whether running in VMware or now with Hyper-V environments running Windows Server 2016 as base for the Hyper-V role. Nested virtualization brings to the table many benefits including hypervisor hosted containers, dev/test environments, and pure lab or training environments that do well inside a nested environment. With VMware and Hyper-V what are the differences in running nested environments on top of both hypervisors? What configuration is involved with each platform to enable nested virtualization? Let's take a look at VMware vs Hyper-V Nested Virtualization and see the similarities and differences between both platforms as related to providing a nested virtualization environment.

Why Nested Virtualization?

Nested virtualization provides a very interesting bit of “inception” when thinking about running workloads. When running nested virtualization, you are installing a hypervisor platform inside another hypervisor platform. So you can have a virtual machine hosted on a hypervisor that is running **as a virtual machine** on top of an existing hypervisor loaded on a physical server. This creates very interesting use cases and possibilities when it comes to spinning up hypervisor resources and affords virtualization administrators yet another tool that enables solving real business problems with techniques in the virtual environment. This includes provisioning dev/test environments with hypervisors mimicking production. However, aside from the dev/test or lab environment that has commonly been the use case with “nested” hypervisor installations, there are some real world production use cases for running nested virtualization from both a VMware and Hyper-V perspective that we want to consider. Let’s take a look a comparison of enabling nested virtualization from a VMware and Hyper-V perspective as well as the differences between the two.

VMware vs Hyper-V Nested Virtualization

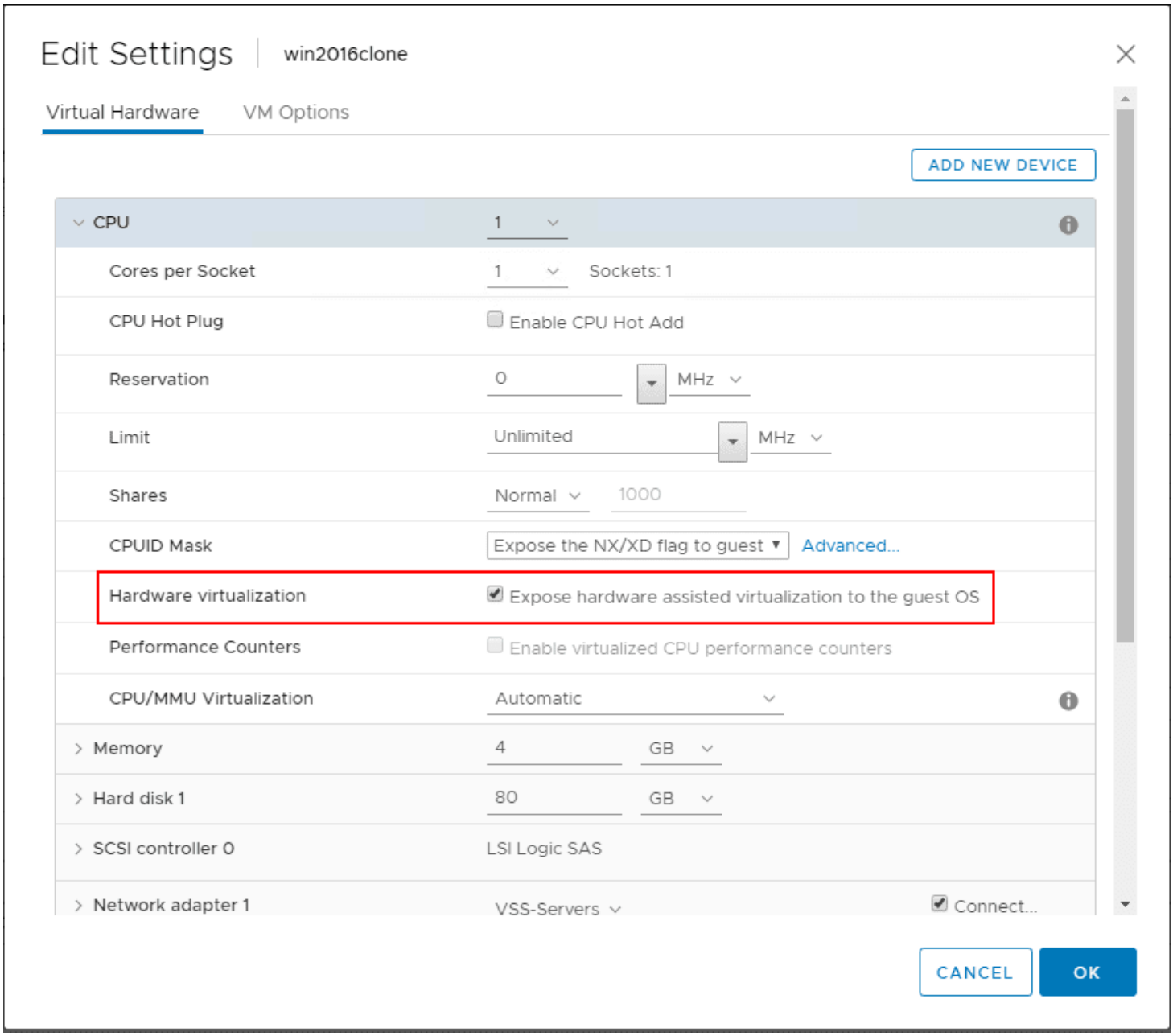
VMware has certainly been in the nested virtualization game for much longer than Hyper-V as Microsoft has only now released nested virtualization as a “thing” in Windows Server 2016 with Hyper-V. There are certainly similarities between the VMware and Hyper-V implementation of nested virtualization as to the requirements, especially in some of the virtual machine configuration required to make nested virtualization possible. We will compare the following:

- Processor requirements and extensions presented to the virtual machine
- Network requirements for connectivity to the production network
- Configuring a virtual machine for nested virtualization in both platforms
- Production use cases for nested virtualization

VMware vs Hyper-V Processor Requirements for Nested Virtualization

Both VMware and Hyper-V require the processors used for nested virtualization are enabled with the Intel VT virtualization technology or AMD-V virtualization technology. This is the hardware assisted technology that is provided from both processor vendors. A note here is that with Hyper-V, nested virtualization is not supported using AMD processors at least in any documentation that I have been able to find stating to the contrary as of yet.

For the guest virtual machine on both platforms, it is required to **expose** the **hardware assisted capabilities** of the process directly to the guest virtual machine running on top of both platforms.



VMware vs Hyper-V nested virtualization – Exposing hardware assisted virtualization to the guest OS

This can also be done via PowerCLI (thanks to LucD here)

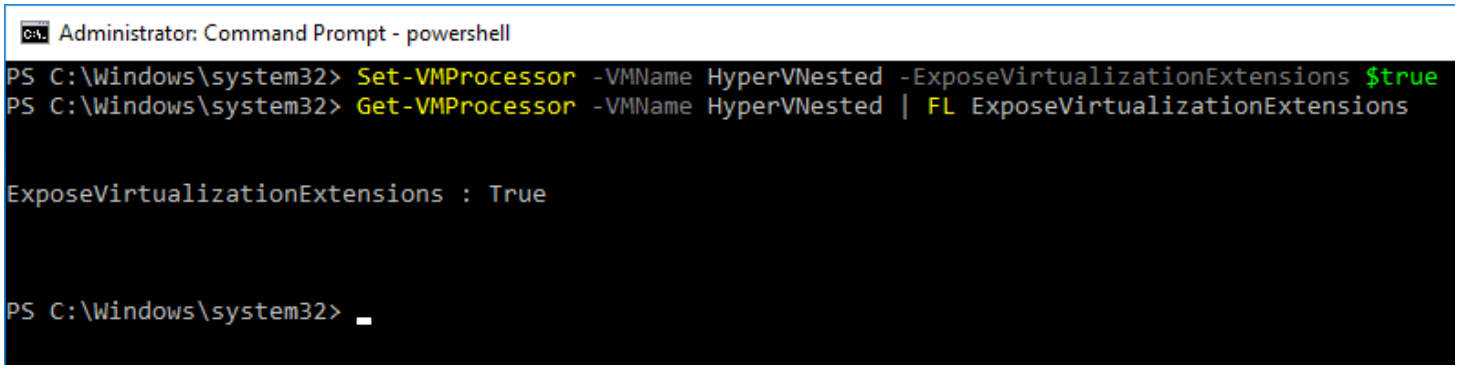
```
$vmName = 'MyVM'

$vm = Get-VM -Name $vmName


$spec = New-Object VMware.Vim.VirtualMachineConfigSpec

$spec.nestedHVEnabled = $true


$vm.ExtensionData.ReconfigVM($spec)
```

In Hyper-V, there is no way to do this from a GUI, so PowerShell makes quick work of this task with the following commandlet:

```
Set-VMProcessor -VMName <Hyper-V VM name> -ExposeVirtualizationExtensions $true
```

To check to see if the extensions are enabled in Hyper-V, you can also use PowerShell for that:

```
Get-VMProcessor -VMName <VMName> | FL ExposeVirtualizationExtensions
```



VMware vs Hyper-V nested virtualization – Exposing CPU Virtualization Extensions in Hyper-V

VMware vs Hyper-V Network Requirements for Nested Virtualization

Both VMware and Hyper-V have specific network requirements (with VMware depending on the version) as related to the nested virtual machine being able to communicate with the outside world. With VMware, for years VMware administrators have known about the requirement to enable **promiscuous** mode and **forged transmits** if they wanted to have nested virtual machine communicate to the production network outside of the nested ESXi host.

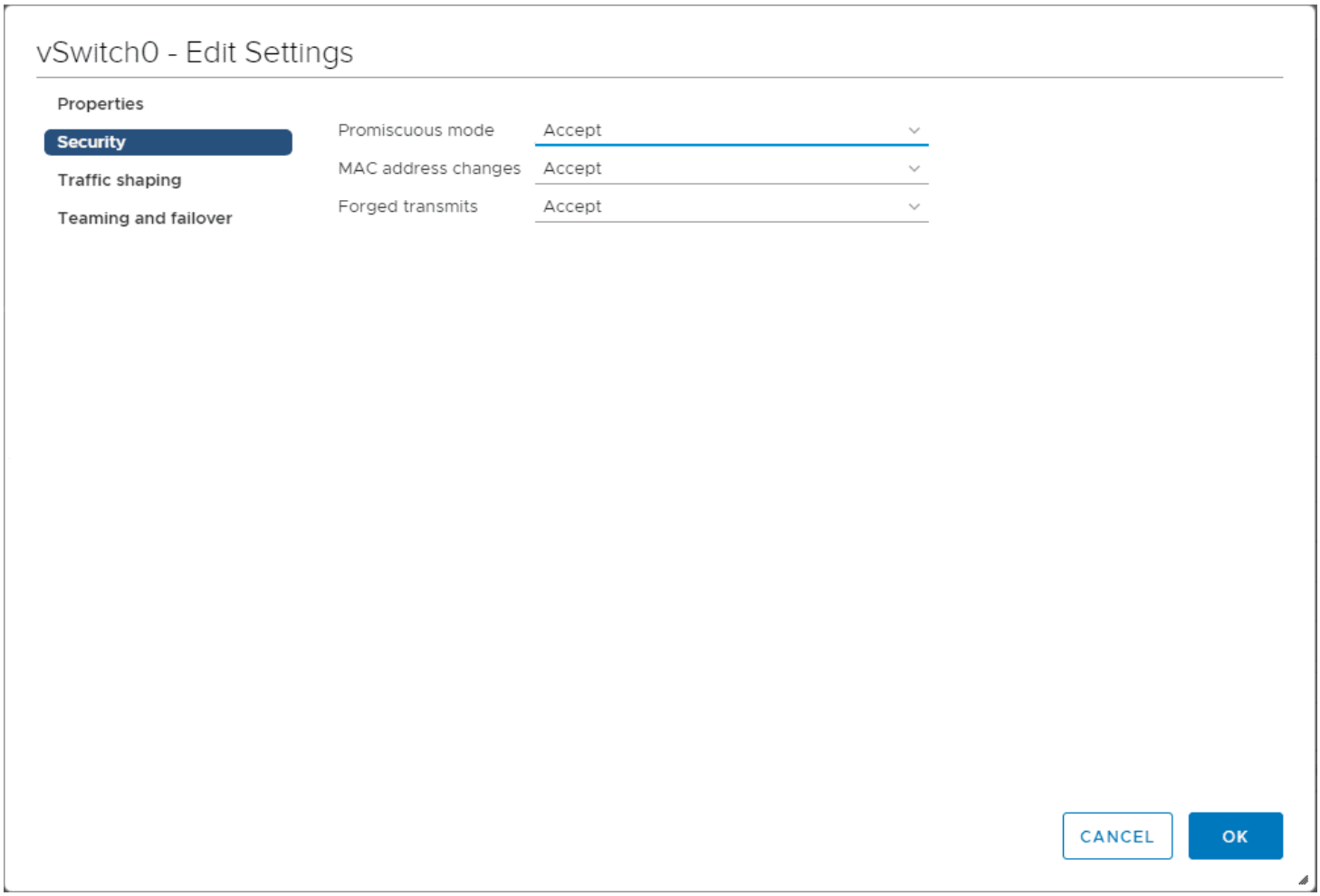
What is promiscuous mode? Promiscuous mode can be defined at the port group level or the virtual switch level in ESXi. Promiscuous mode allows seeing all the network traffic traversing the virtual switch. Prior to ESXi 6.7, the ESXi host virtual standard switch and vSphere Distributed Switch do not implement MAC learning like a traditional physical switch. With that being the case the virtual switch only forwards network packes to a virtual machine if the destination MAC address matches the ESXi vmnic's pNIC MAC address. In a nested environment these destination addresses differ so will be dropped. By enabling promiscuous mode, this introduces overhead on the virtual switch.

With forged transmits, this allows the **source MAC address** to not match the MAC address of the ESXi server so again the packets will not be dropped.

Resources for understanding

- Why is promiscuous mode & forged transmits required for nested ESXi?
- How the VMware forged transmits security policy works

Below, looking at the properties of a Virtual Standard Switch, under **Security** setting promiscuous mode and forged transmits to **Accept**.

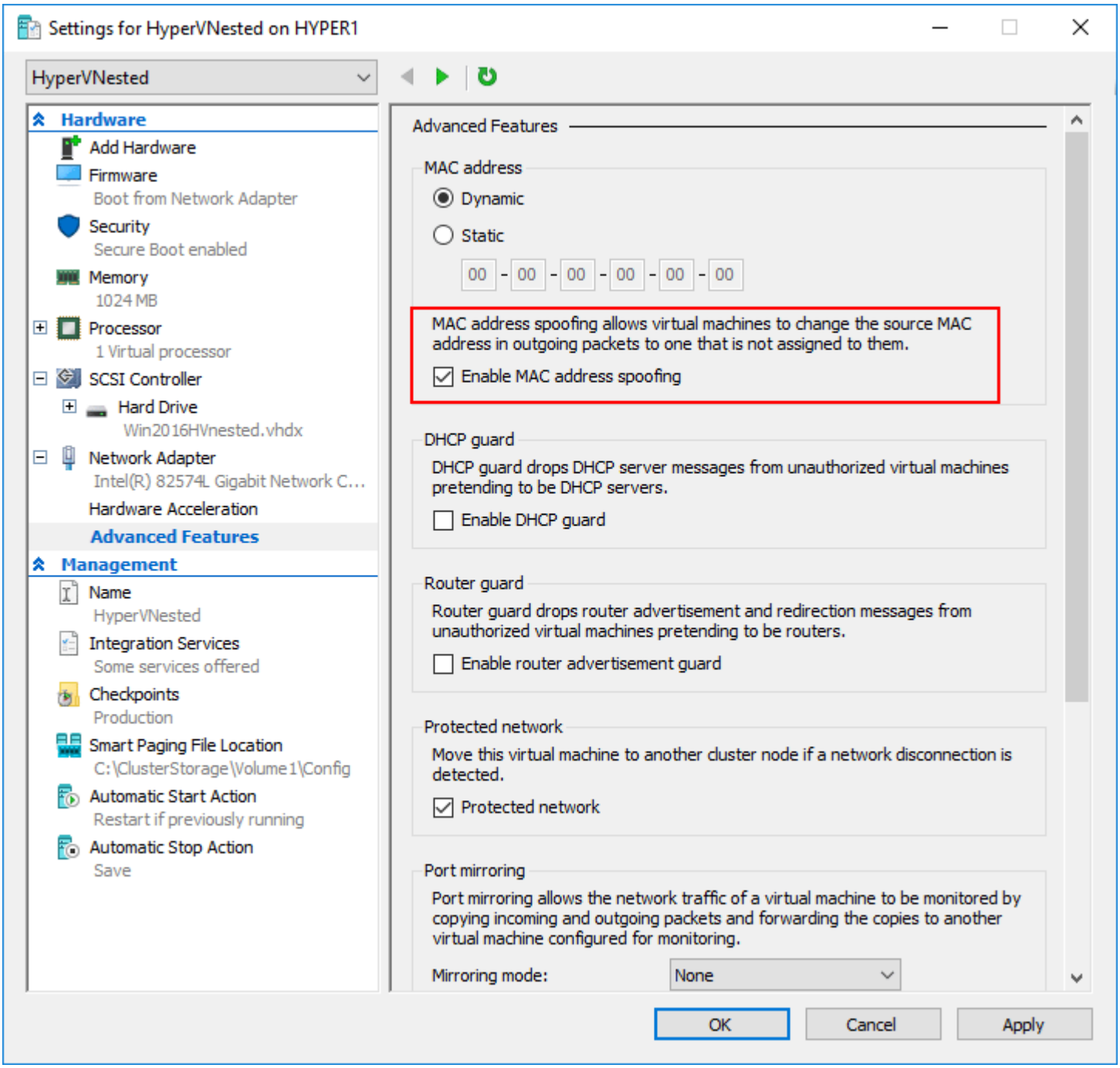


VMware vs Hyper-V nested virtualization – configuring VMware networking for nested virtualization pre ESXi 6.7

With VMware vSphere ESXi 6.7, VMware has implemented the work of a MAC learning Fling that has been around which adds the MAC learning functionality to the virtual switch. Promiscuous mode is no longer needed with ESXi 6.7 to run nested virtualization. See William Lam's great write up about this new functionality below:

- https://www.virtuallyghetto.com/2018/04/native-mac-learning-in-vsphere-6-7-removes-the-need-for-promiscuous-mode-for-nested-esxi.html

With Hyper-V, this is not enabled at the virtual switch level, but rather at the virtual machine level. Note on the properties of the virtual machine, under **Network Adapter >> Advanced Features >> MAC Address spoofing** this needs to be enabled. This is the same issue that VMware faced previous to ESXi 6.7 in running nested virtualization.



Configuring MAC address spoofing with Hyper-V nested virtualization

Additionally, with Hyper-V, running a NAT configuration for the nested Hyper-V virtual machine networking is also supported. To create the NAT configuration in Hyper-V for nested virtualization:

```
New-VMSwitch -Name NAT_switch -SwitchType Internal
New-NetNat –Name LocalNAT –InternalIPInterfaceAddressPrefix "192.168.50.0/24"
```

Other Hyper-V specific requirements

- Must be running Windows Server 2016
- Only Hyper-V is supported as the “guest” hypervisor
- Dynamic memory must be disabled
- SLAT must be supported by the processor

VMware vs Hyper-V Production Use Cases

There are differences between the platforms with use cases for nested virtualization. With VMware, there is really only one officially supported use case with nested virtualization and that is with the **vSAN Witness Appliance** which is nothing more than a nested ESXi appliance.

Take a look at the official documentation from VMware on supported nested virtualization use cases:

- https://kb.vmware.com/s/article/2009916

With Hyper-V, the main Microsoft mentioned use case is for **Hyper-V containers**. This provides further isolation for containers by running them inside a VM in Hyper-V.

- https://docs.microsoft.com/en-us/virtualization/hyper-v-on-windows/user-guide/nested-virtualization

Takeaways

In this look of VMware vs Hyper-V Nested Virtualization, it is easily seen there are similarities and differences between VMware and Hyper-V and their implementations and use cases involving nested virtualization. Depending on the platform you have running in your enterprise datacenter, the supported functionality with nested vitualization will vary based on the platform. Both VMware and Hyper-V provide really great functionality in the respective nested solutions. VMware certainly has the more robust and fully featured nested solution, especially with vSphere 6.7 and the new networking enhancements that have been made. Microsoft has at least finally broke ground and stepped into the world of supported nested virtualization. It will be interesting to see how this develops with subsequent Hyper-V releases.

Install Nested Nutanix CE in VMware vSphere ESXi 6.7 Update 1

April 11, 2019
Nutanix

Install-Nested-Nutanix-CE-in-VMware-vSphere-ESXi-6.7-Update-1

One of the great things about virtualization is "nested virtualization". This is where you can run a hypervisor inside a virtual machine running on another hypervisor. One of the really cool things with the lab environment is playing around with other hypervisors. Recently, I started taking a look at installing Nutanix Community Edition inside of VMware vSphere. With just a bit of Googling and stepping through the process, I was able to get a good install of Nutanix CE running inside a VMware

vSphere 6.7 Update 1 environment. Let's take a look at how to install Nested Nutanix CE in VMware vSphere ESXi 6.7 Update 1 and take a look at the steps required to do this.

Downloading Nutanix CE

While we won't detail how to get the Nutanix CE bits here, it is fairly straightforward by visting the following URL:

- https://portal.nutanix.com/#/page/docs/details?targetId=Nutanix-Community-Edition-Getting-Started:Nutanix-Community-Edition-Getting-Started

You will have go through the process of creating a community account (NEXT account) and verify your email address, etc. Basically, all of the normal stuff.

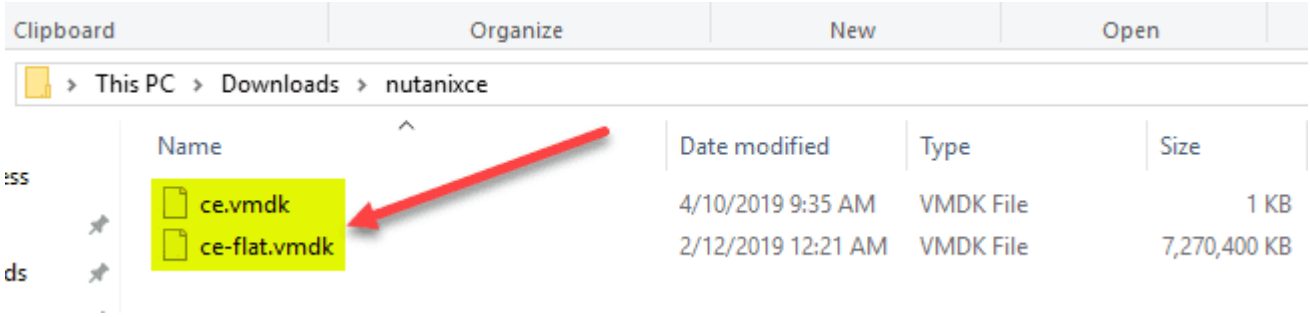
Getting the Nutanix VMDK Files Ready

I'm not going to go into overly detailed instructions for this part, however, once you download the Nutanix CE package from the community page, it will download as a **tar.gz** file. Extract this file.

The file that is extracted will end in an **.img** extension. Rename this file to a **.vmdk** extension. For me this was around an **8GB** file. This is the **flat** VMDK, so we need to add a descriptor VMDK file.

There is a great post here from Virtuallifestyle that actually has a descriptor file available for download. This is the file that I used in the lab to get Nutanix CE loaded.

When you are finished with the prep work including downloading the file from Nutanix and then creating a VMDK descriptor file, you will have two files, similar to this.

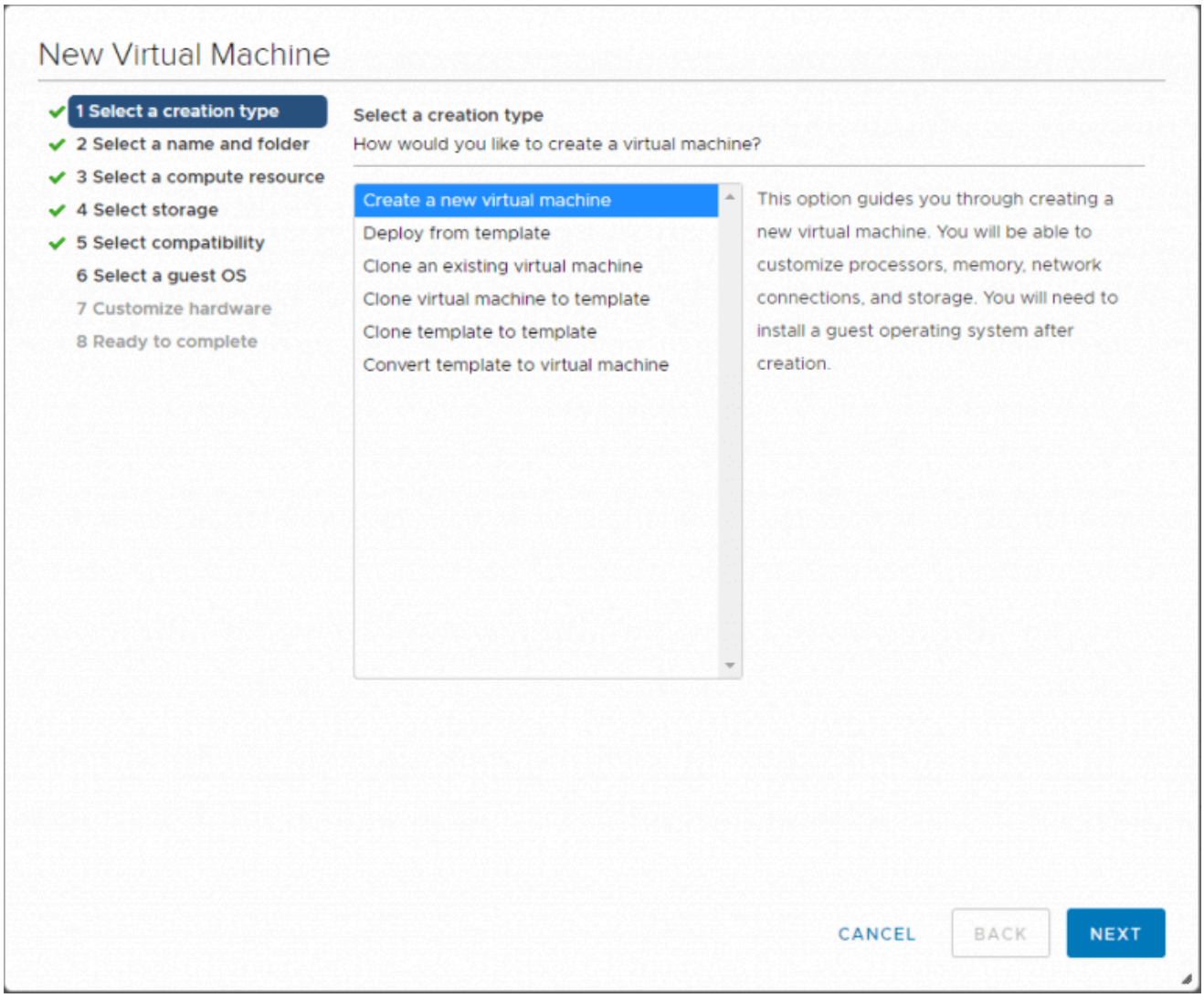


After downloading the Nutanix CE VMDK and creating a descriptor file

Now the last thing to do for getting ready to install Nutanix in VMware ESXi 6.7 Update 1, is upload the files to your datastore. Create a folder for the VM and upload the two files there.

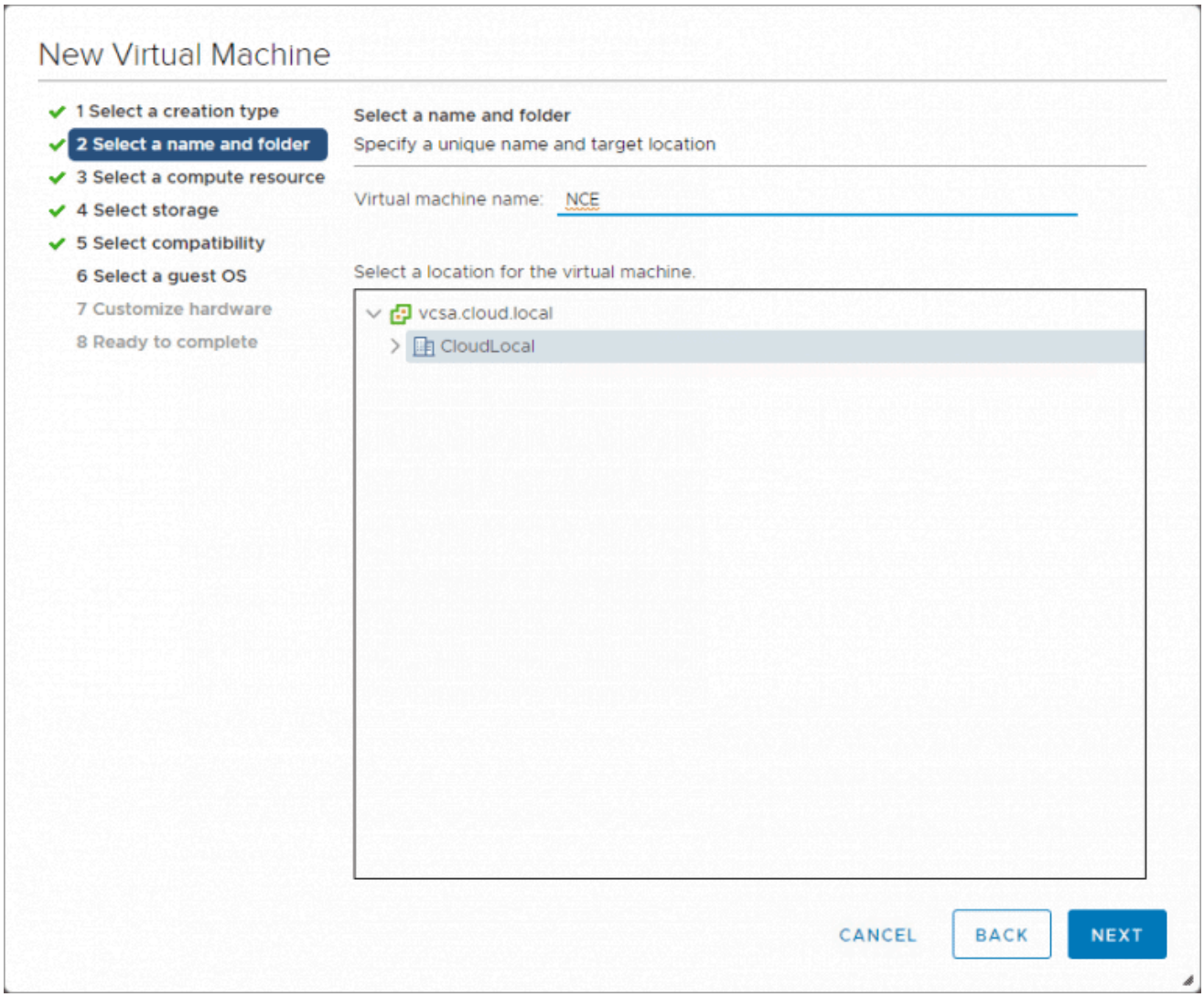
Install Nested Nutanix CE in VMware vSphere ESXi 6.7 Update 1

Let's take a look at the steps to get a VM created to house the Nutanix CE install. Here I am using the vSphere 6.7 Update vSphere client to get the VM created and provisioned. Begin by creating a new virtual machine.



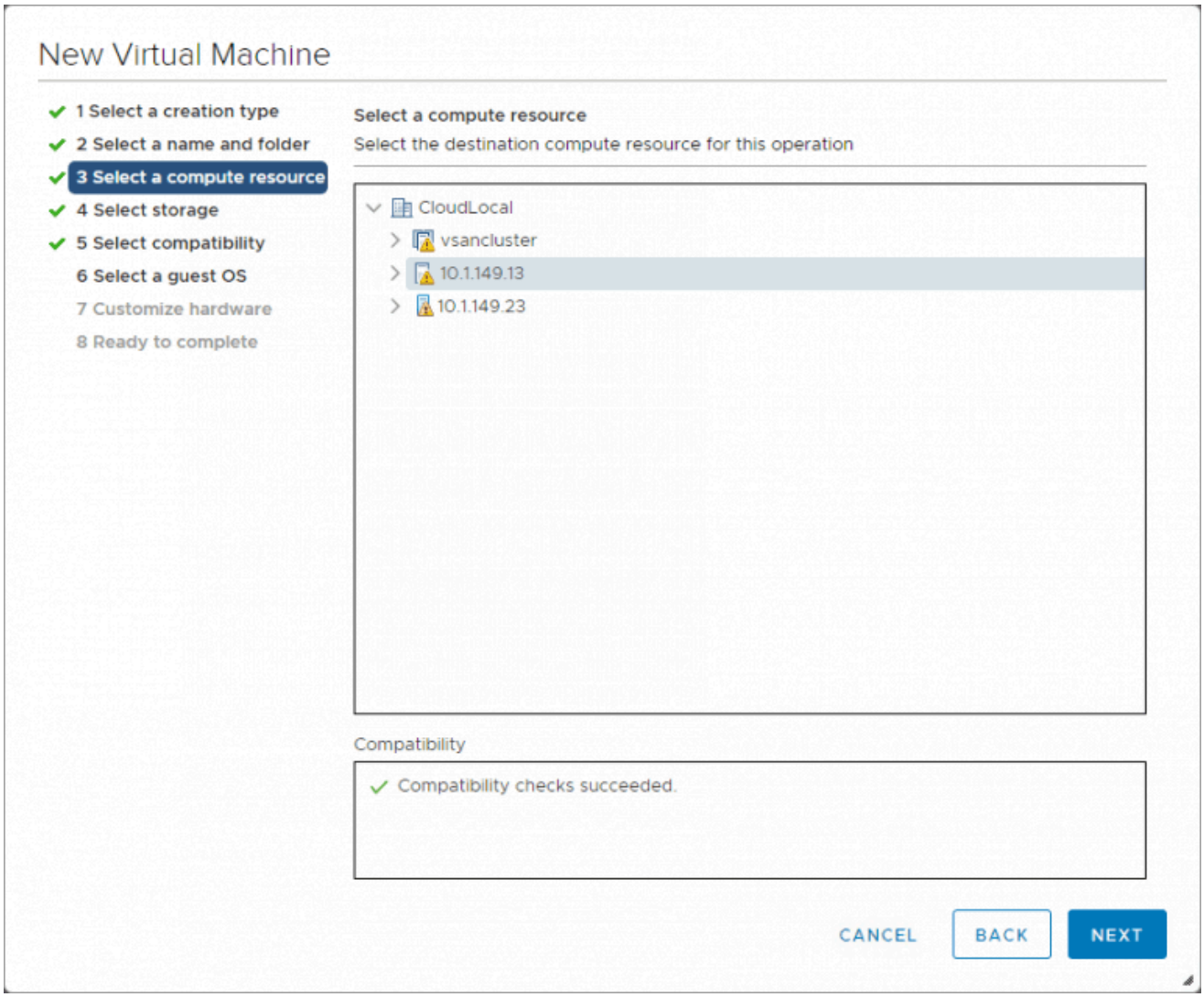
Creating a new VMware ESXi virtual machine for installing Nutanix CE

Choose a name and folder for the nested Nutanix CE VM.



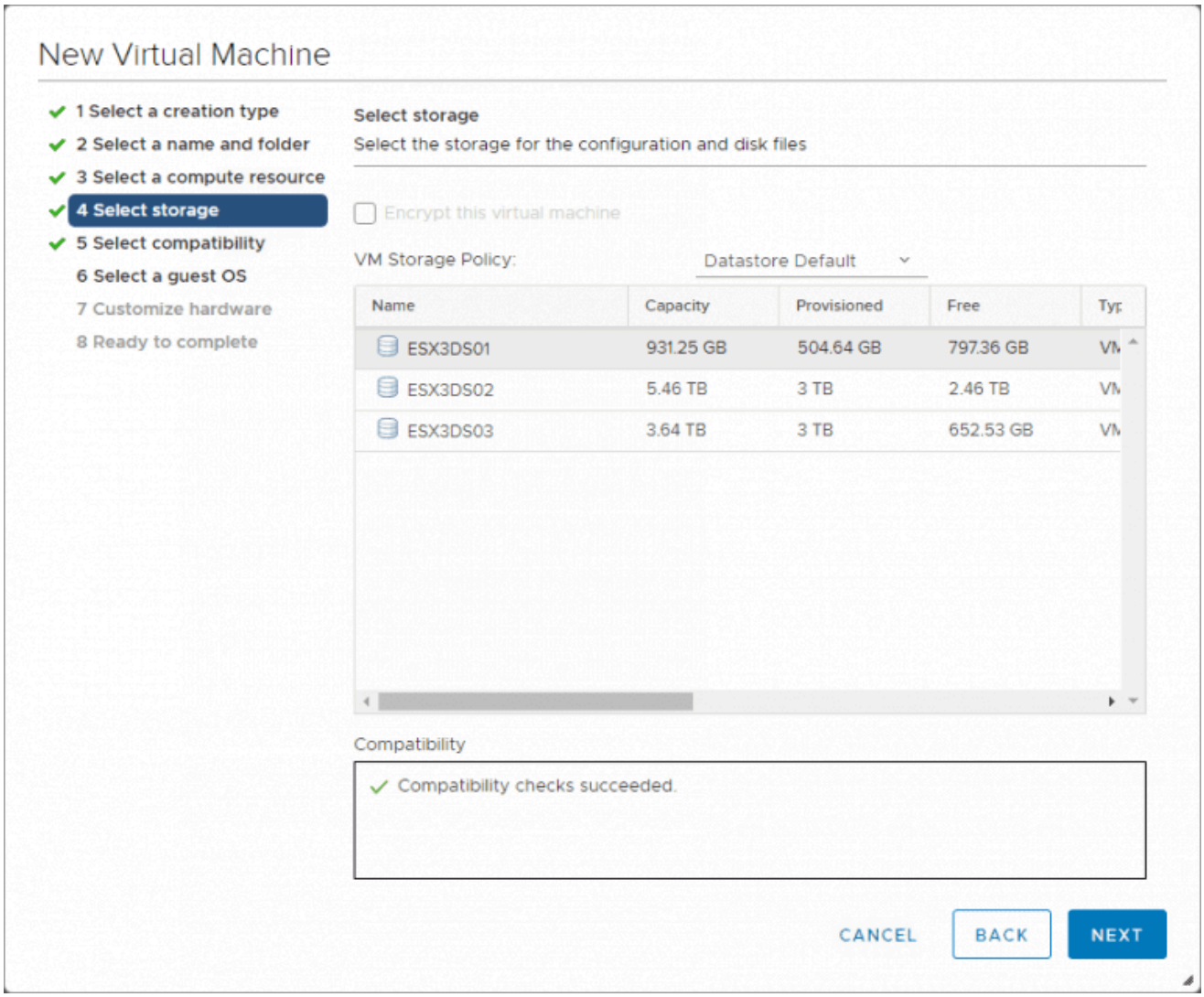
Name the VM and select the folder in teh datacenter for housing it

Choose your compute resource for the Nutanix CE installation.



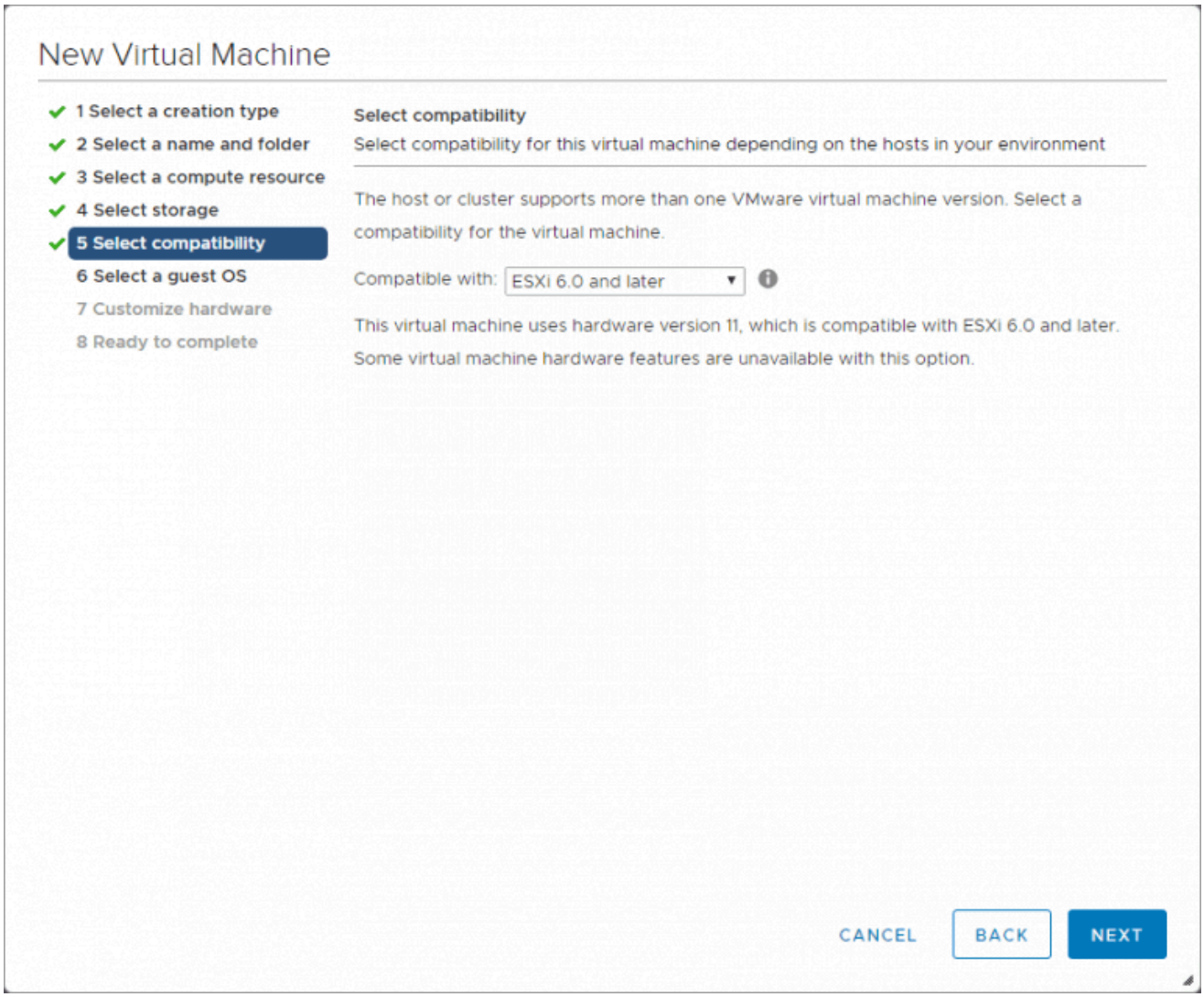
Select the compute resource in your vSphere environment for Nutanix CE nested installation

Choose a datastore. A note here, the two disks required for the Nutanix CE installation need to be at a minimum 500 GB and 200 GB. In the lab environment with NVMe storage, I am thin provisioning.



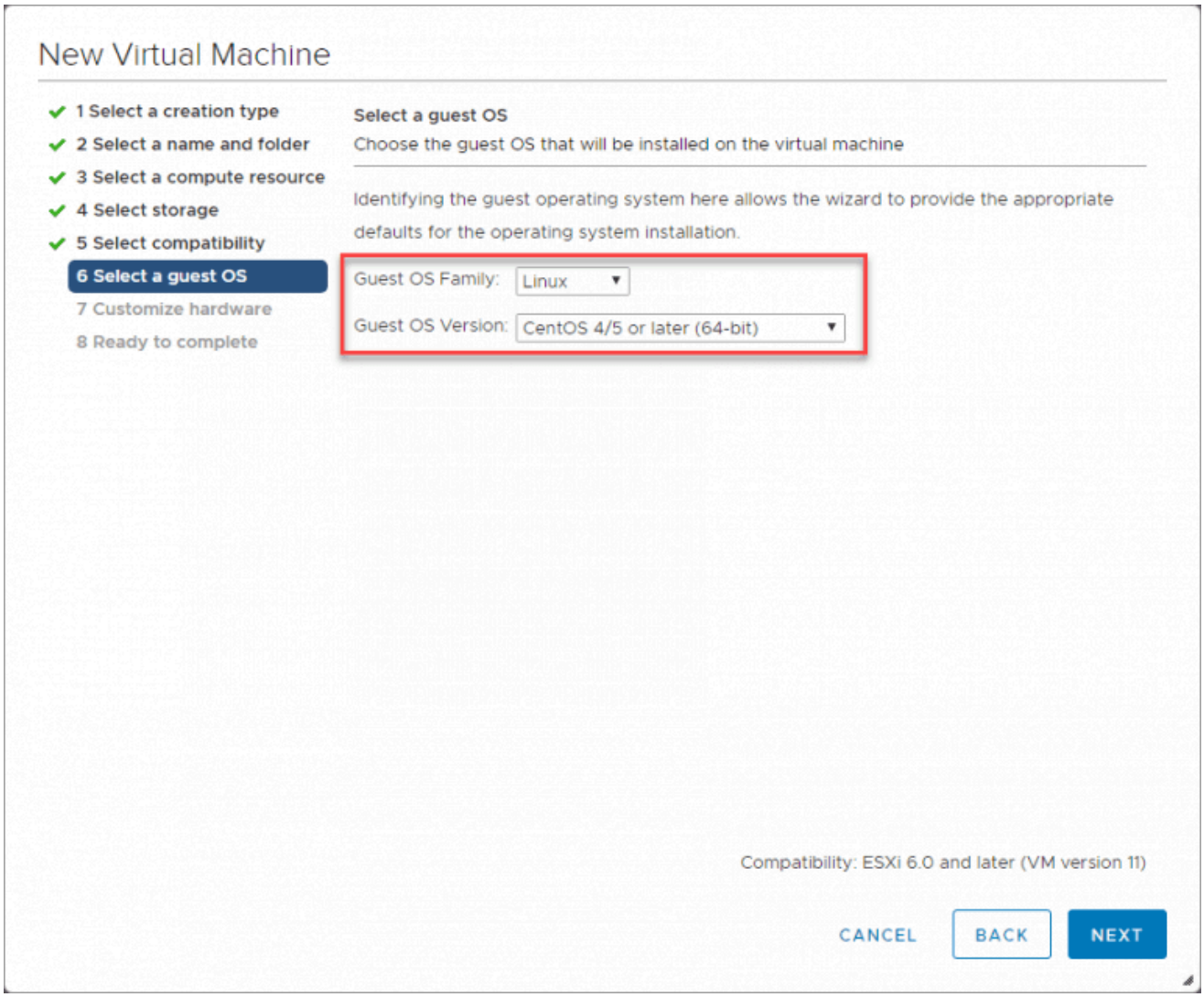
Select the datastore to house the Nutanix CE nested VM

Here, I am selecting ESXi 6.0 and later for the compatibility mode. For no other reason than some of the other posts on installing Nutanix CE in vSphere, I chose this compatibility level.



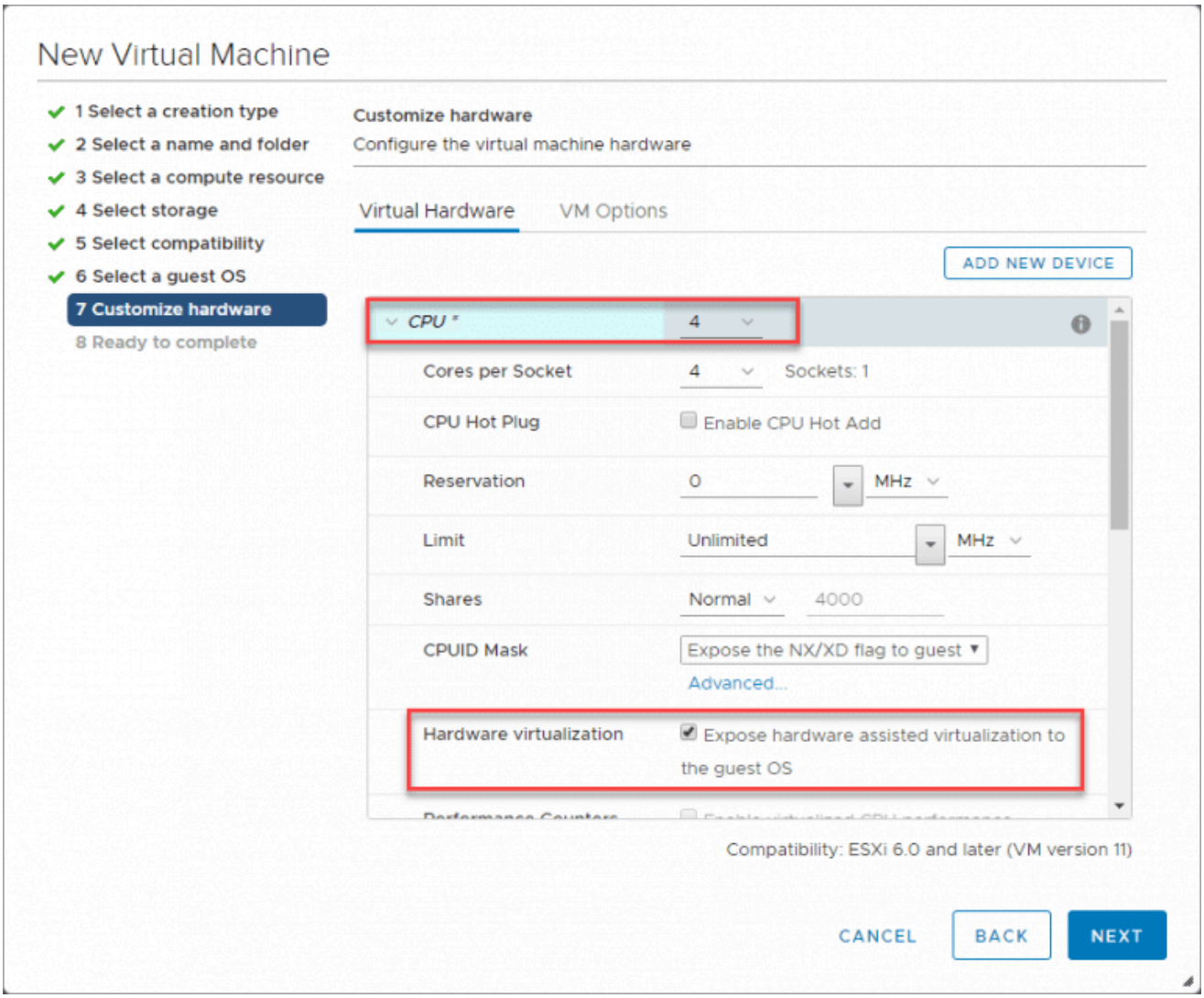
Select the compatibility mode for the Nutanix CE nested isntallation

For installing Nutanix CE, you need to choose **Linux** and **CentOS 4/5 or later (64-bit)**.



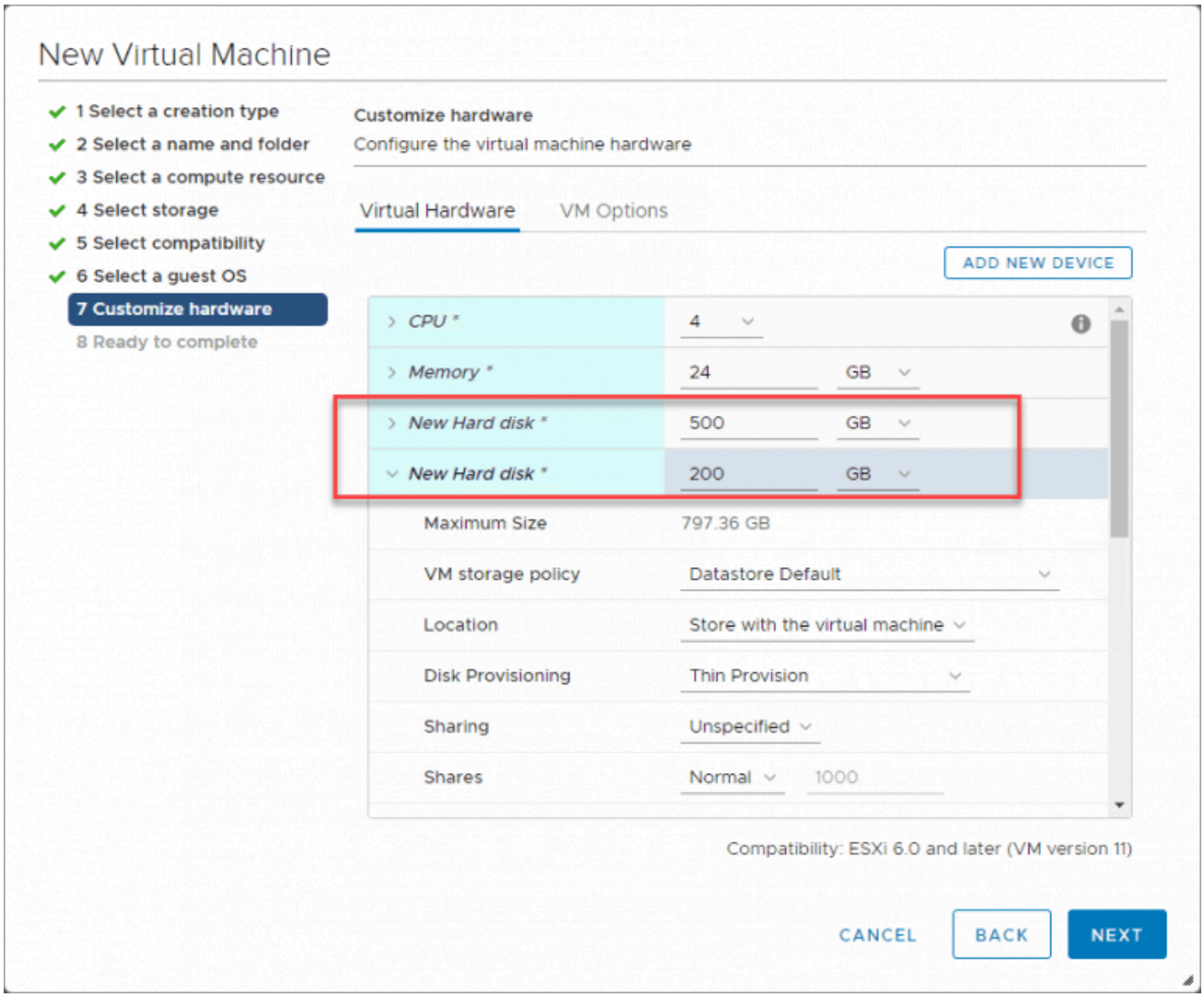
Select the guest OS for Nutanix CE

Now, we being the list of things we need to do to customize the virtual hardware for the Nutanix CE installation. The minimum here is **4 vCPUs**. Also, the same as needed with nested ESXi installations, you need to enable the **Expose hardware assisted virtualization to the guest OS** option.



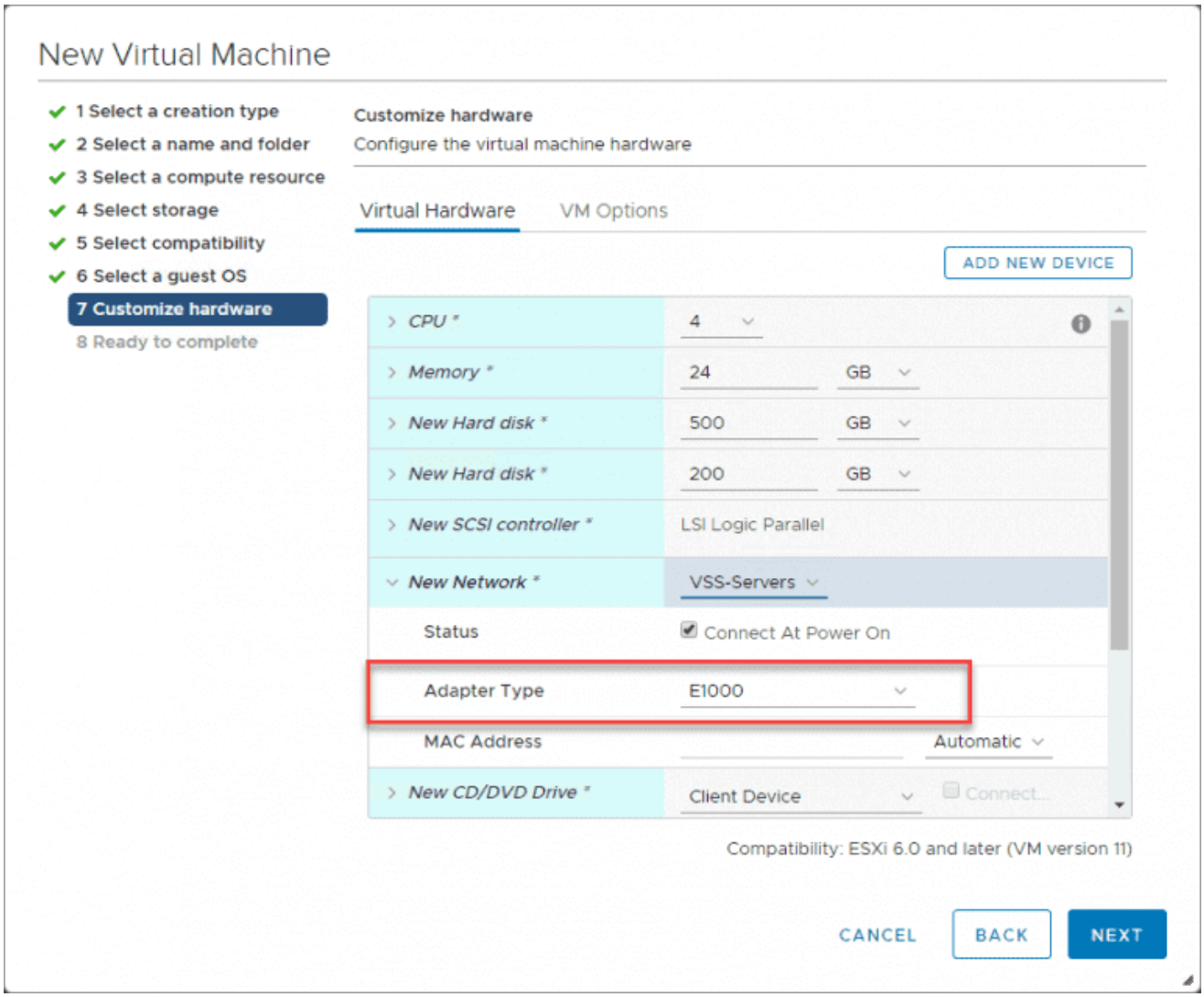
Customizing the CPU configuration for Nutanix CE

Configure and add hard disks here for the minimums if you want here. The first hard disk here is 500 GB and the second is 200 GB. Again, I am thin provisioning.



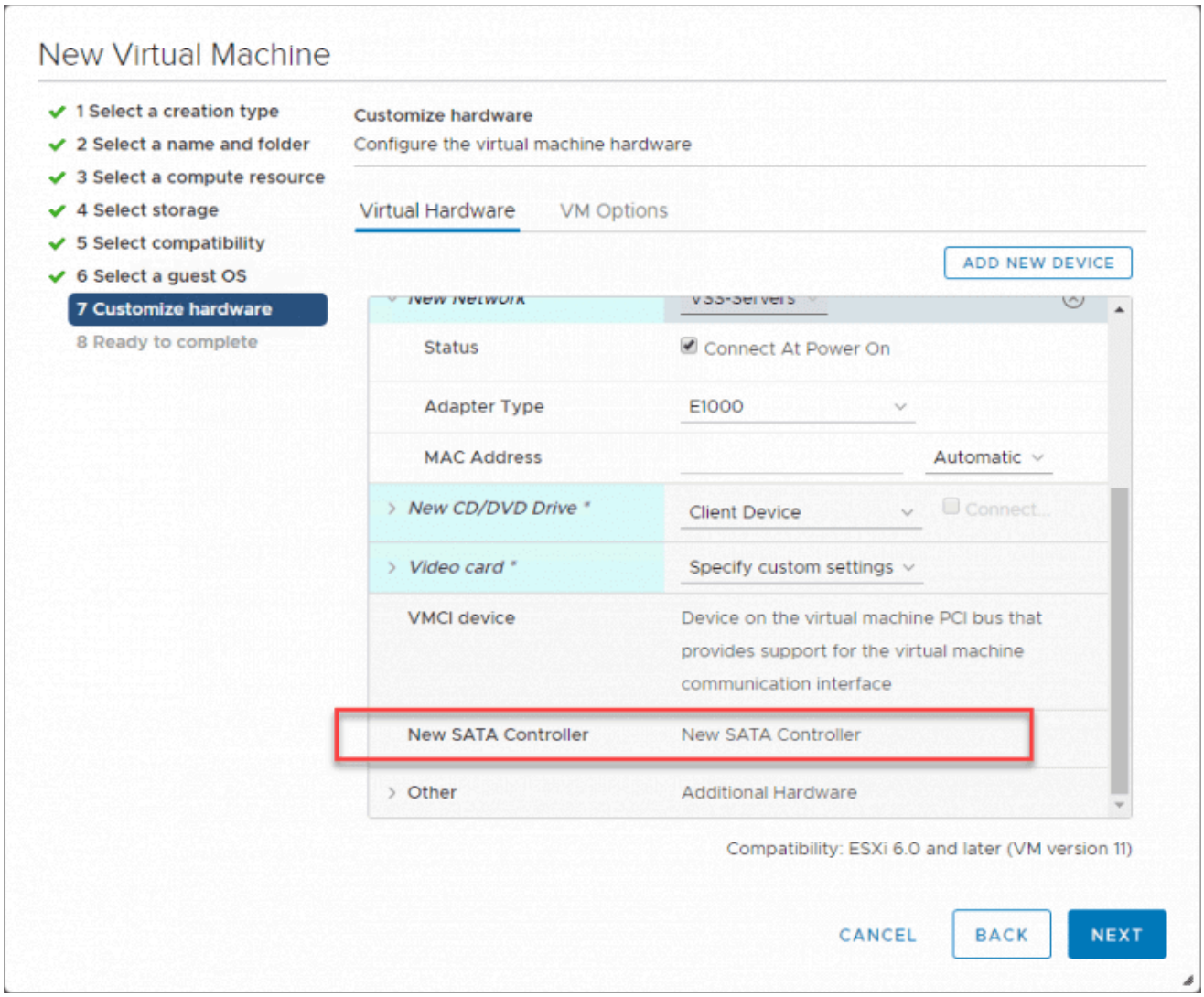
Adding the required hard disks for the Nutanix CE nested installation on ESXi

For network compatibility, choose the **E1000** adapter.

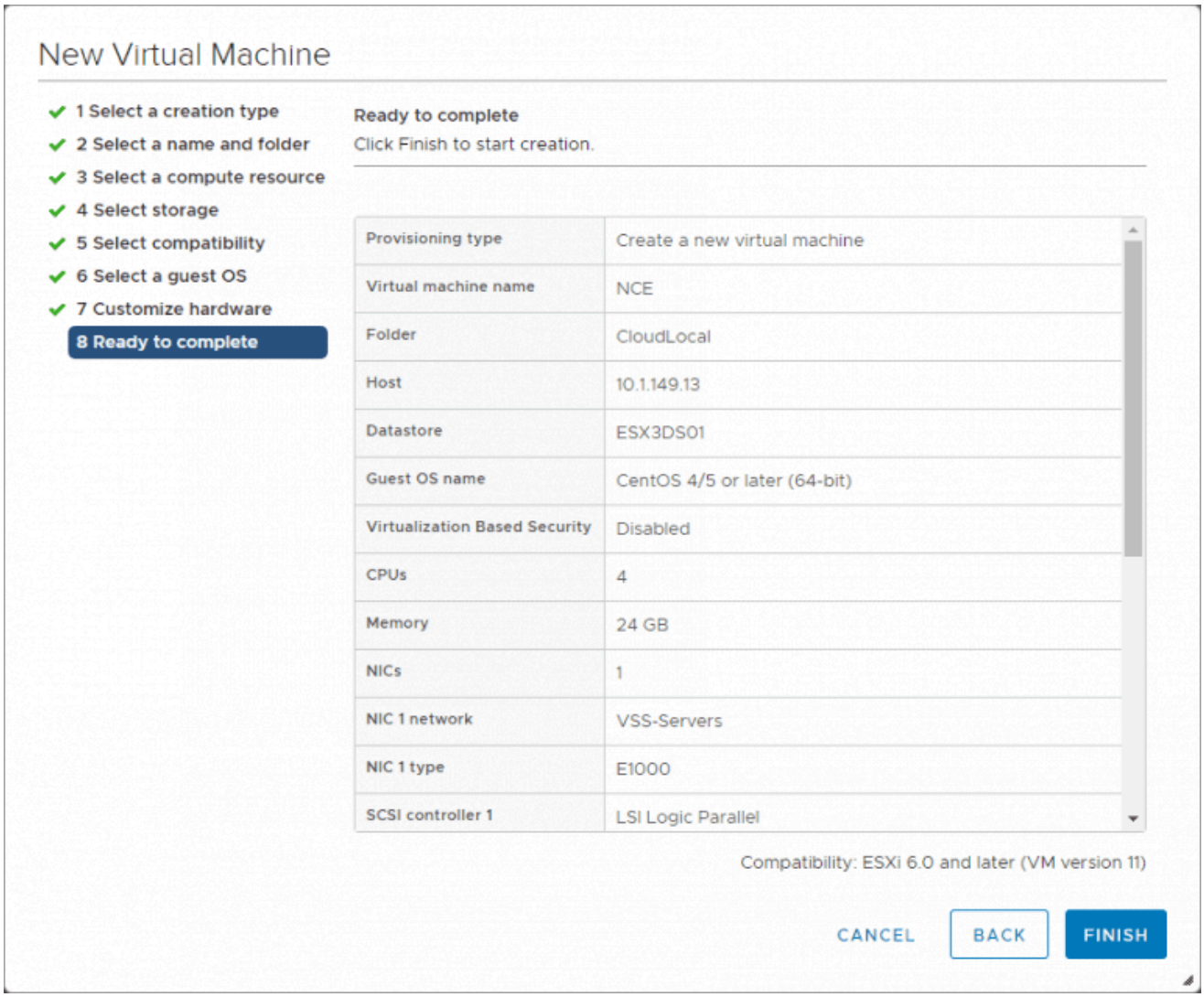


Configuring the network connection for Nutanix CE

Finally, in the wizard, add a **New SATA Controller**. This is the controller we will attach the downloaded Nutanix CE VMDK to.

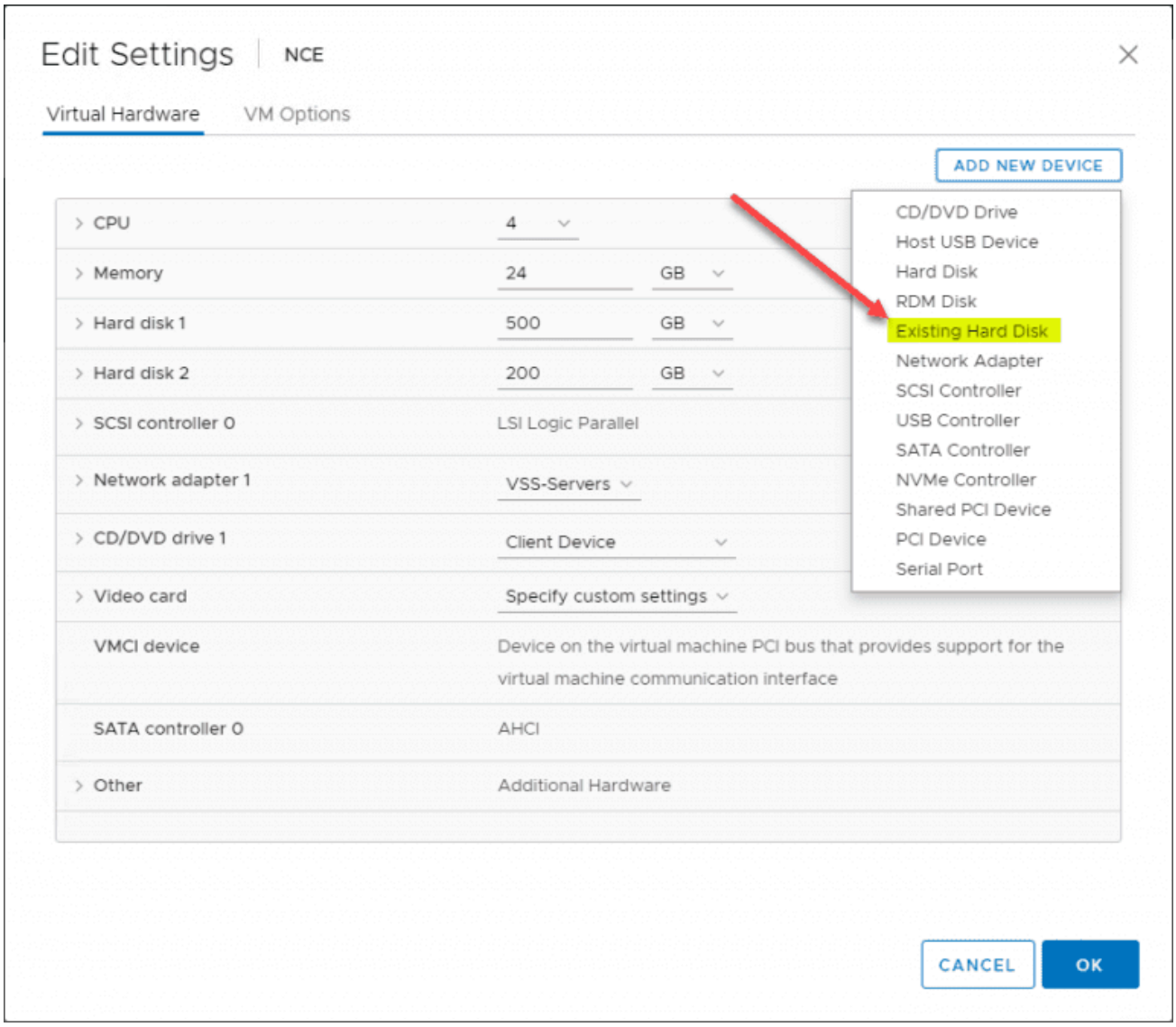


Finish out the wizard.



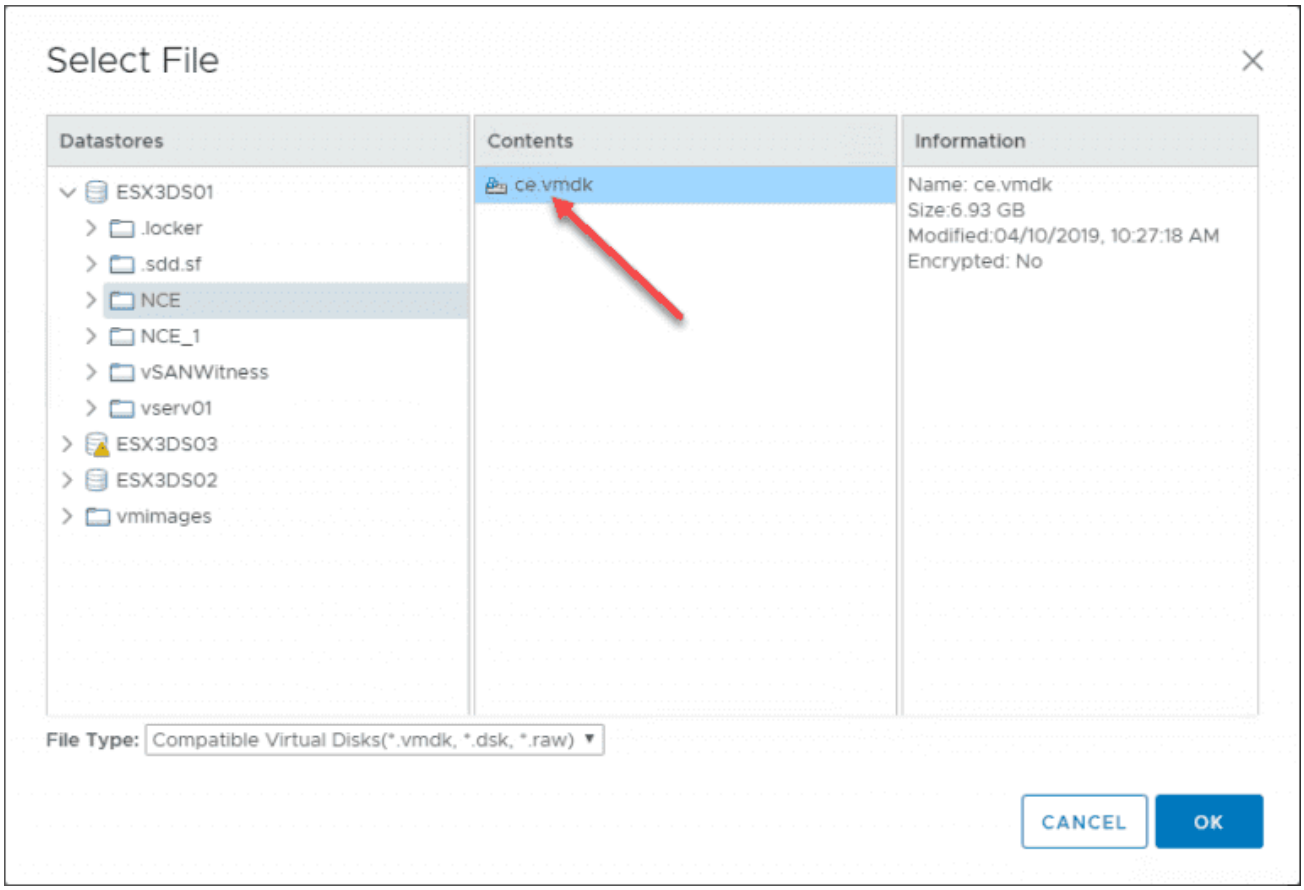
Finishing the initial VM creation wizard for Nutanix CE

Edit the resulting VM and choose to **Add New Device > Existing Hard Disk**.



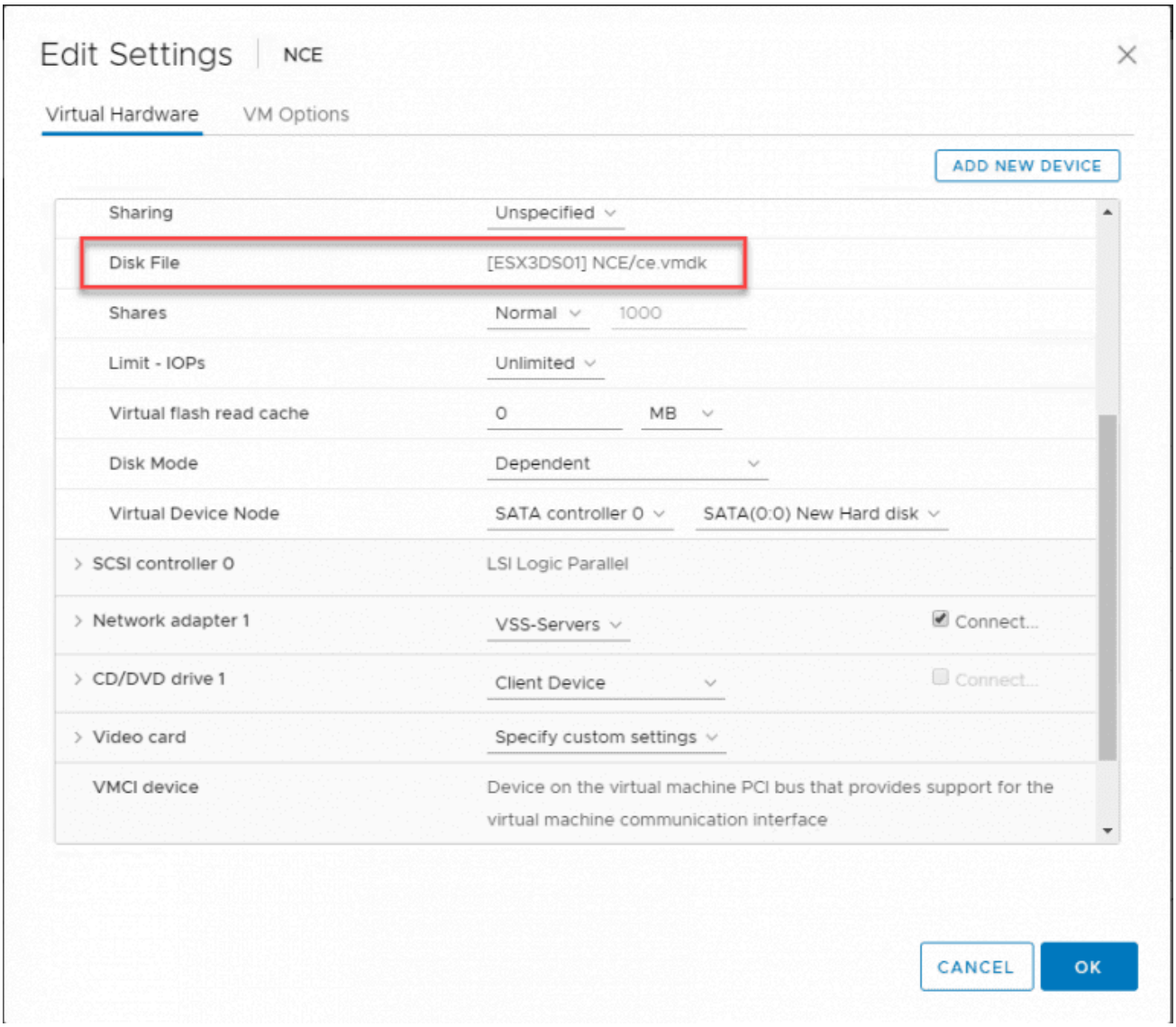
Adding a new Hard Disk to the newly created Nutanix CE virtual machine

Browse and choose the Nutanix CE VMDK on your datastore.



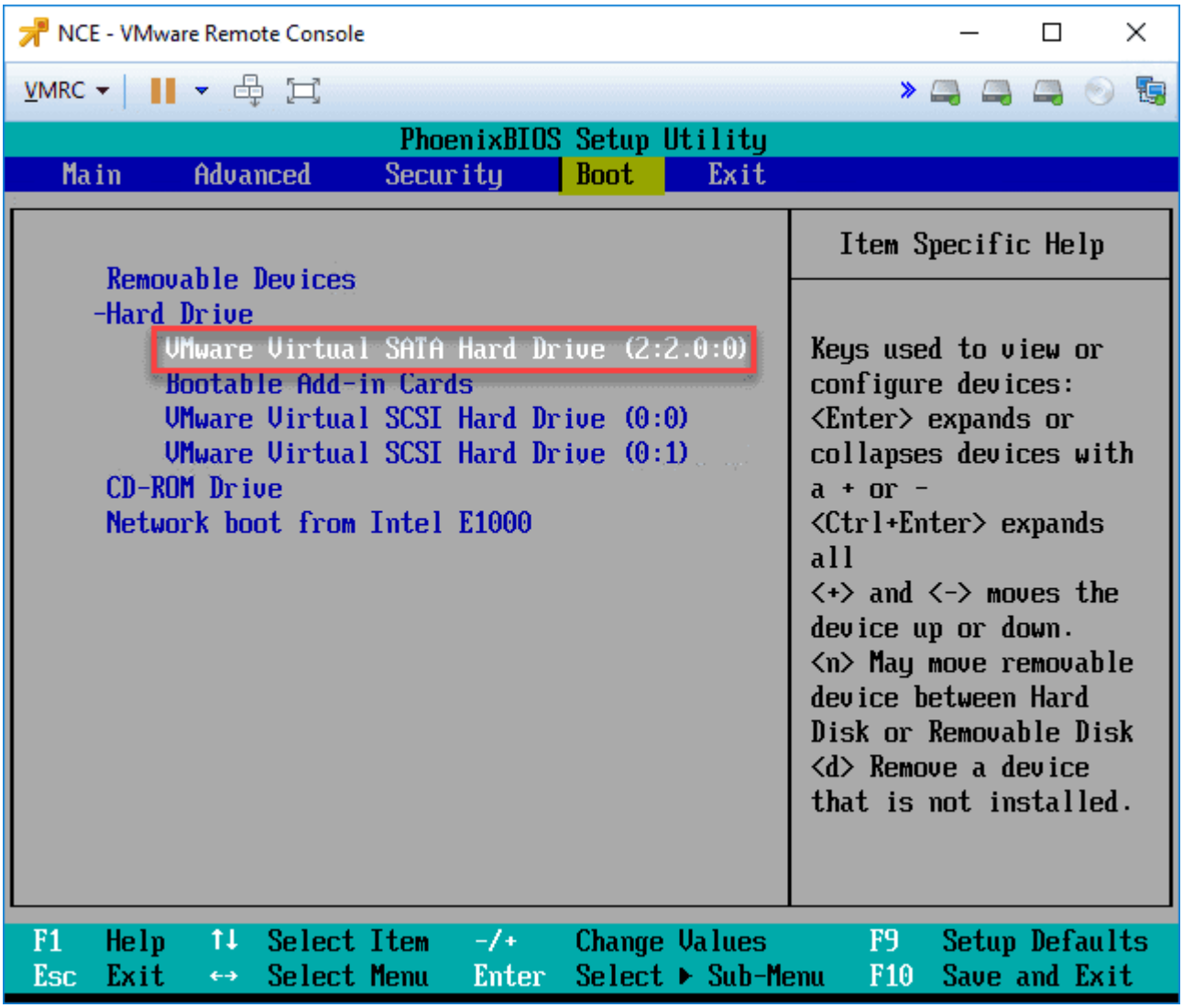
Select the downloaded Nutanix CE VMDK uploaded to the ESXi datastore

The downloaded Nutanix CE VMDK is now attached to the VM.



New SATA disk for Nutanix CE added to the VM

In the boot options for the VM, set the boot order to include the SATA Hard Drive first.

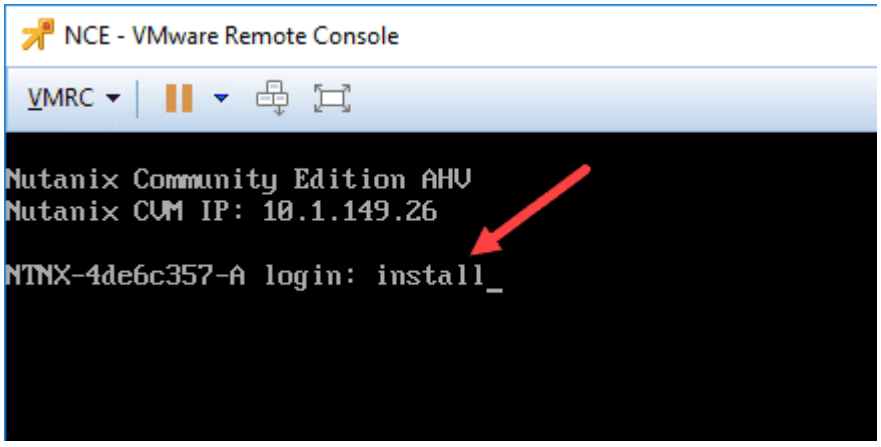


Change the default boot order of the VM to use the new SATA drive

Power on the Nutanix CE VM and you should see the VM boot Nutanix for installation.

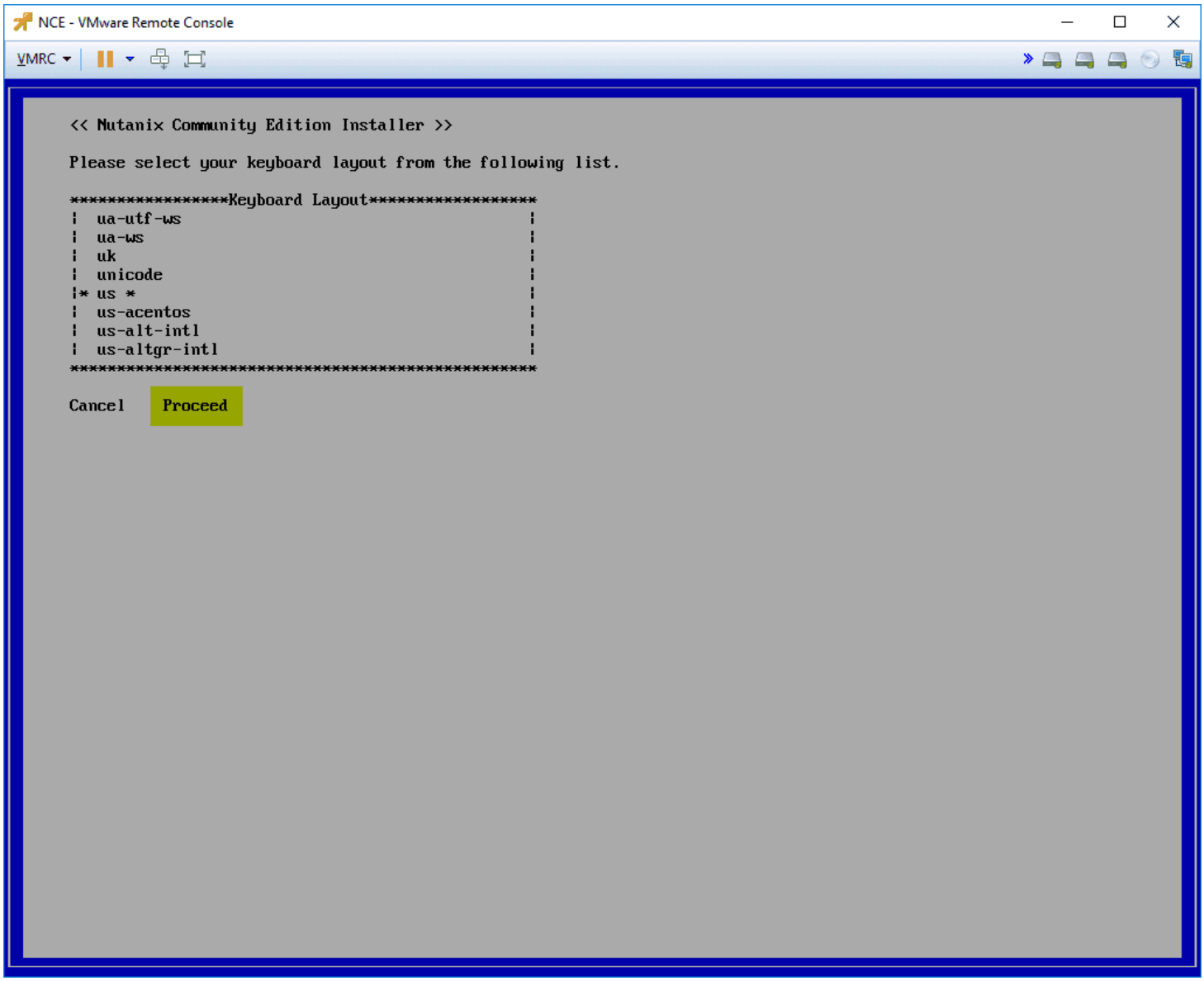
Booting the Nutanix CE VM to begin the installation

Once the VM is booted, on the login screen, enter the **install** user and hit ENTER. There is no password.



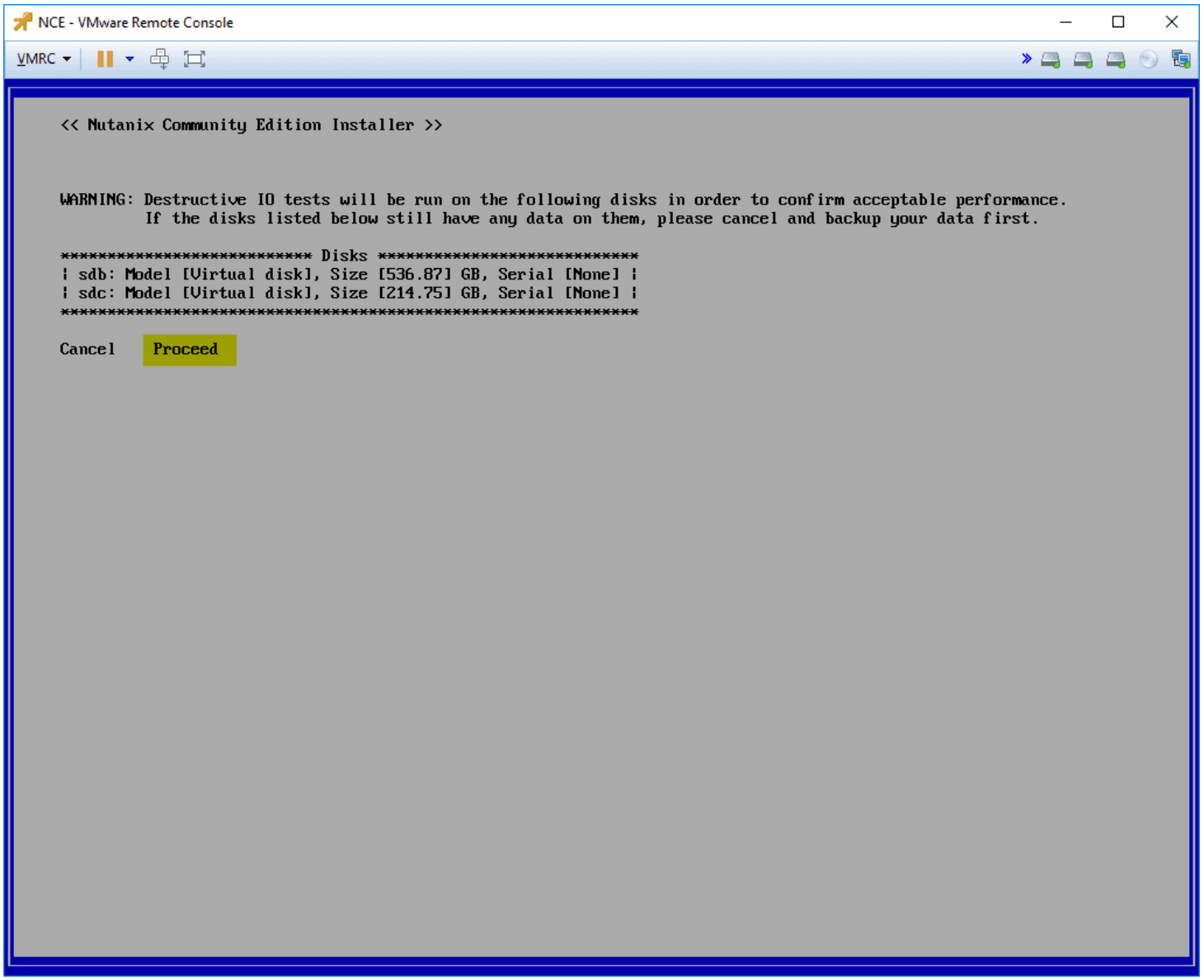
Login as the install user to begin installation

This will launch the Nutanix Community Edition Installer. Choose your location and then choose **Proceed**.



Nutanix Community Edition installer begins

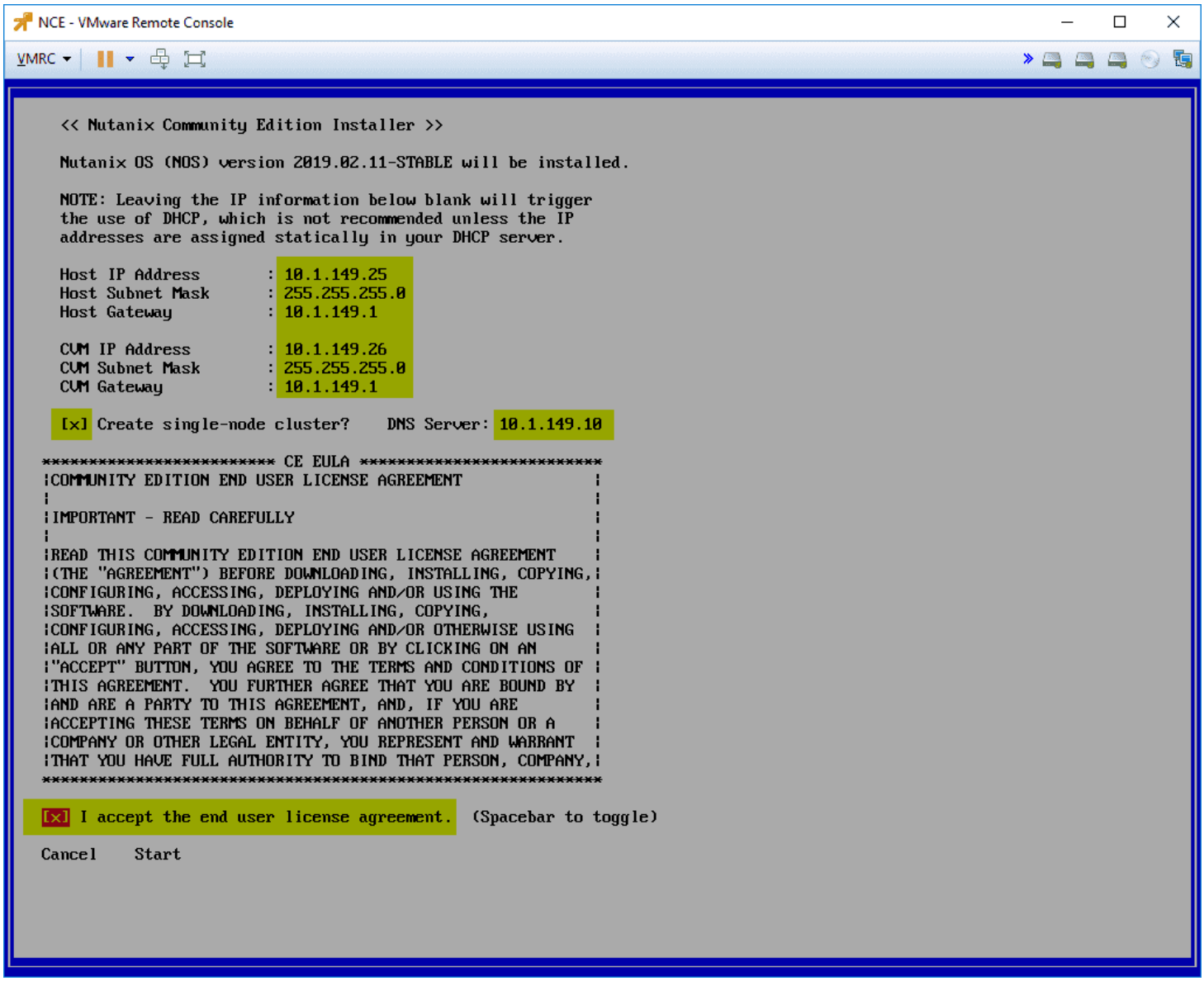
Next, you will be asked to confirm the destructive IO tests on your hard disks.



After the IO tests, there are several important network configuration parameters to configure. You will be configuring:

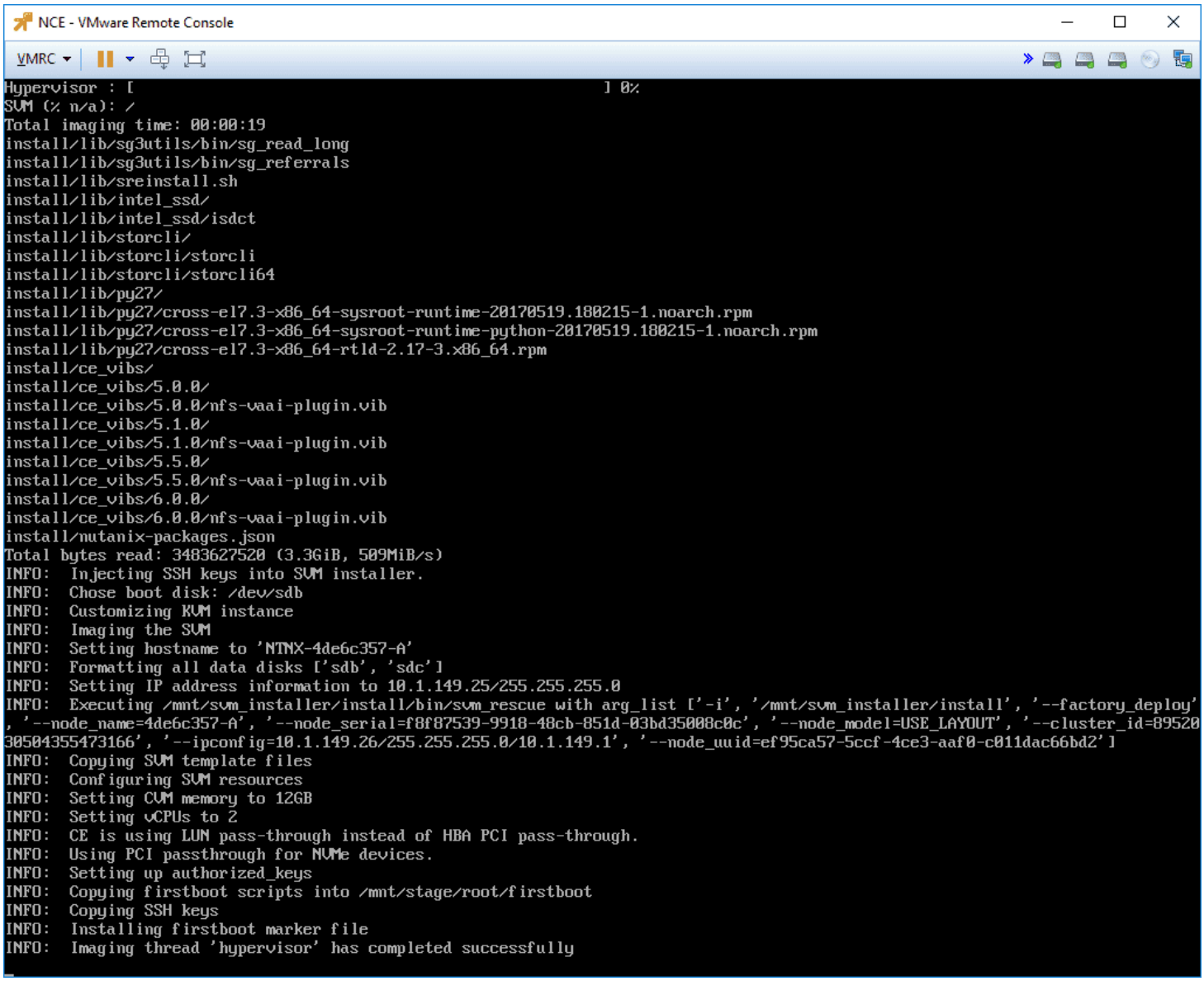
- Nutanix host IP, Subnet, and gateway
- Nutanix CVM IP, Subnet, and gateway
- Create single-node cluster (for testing “cluster” with single node)
- DNS Server
- Accept the EULA

Note if you do not scroll through the EULA, you will see an error after this configuration page that will have you go back to this page and actually scroll through the EULA before accepting. AFter doing that, navigate to **Start** which will begin the actual installation.



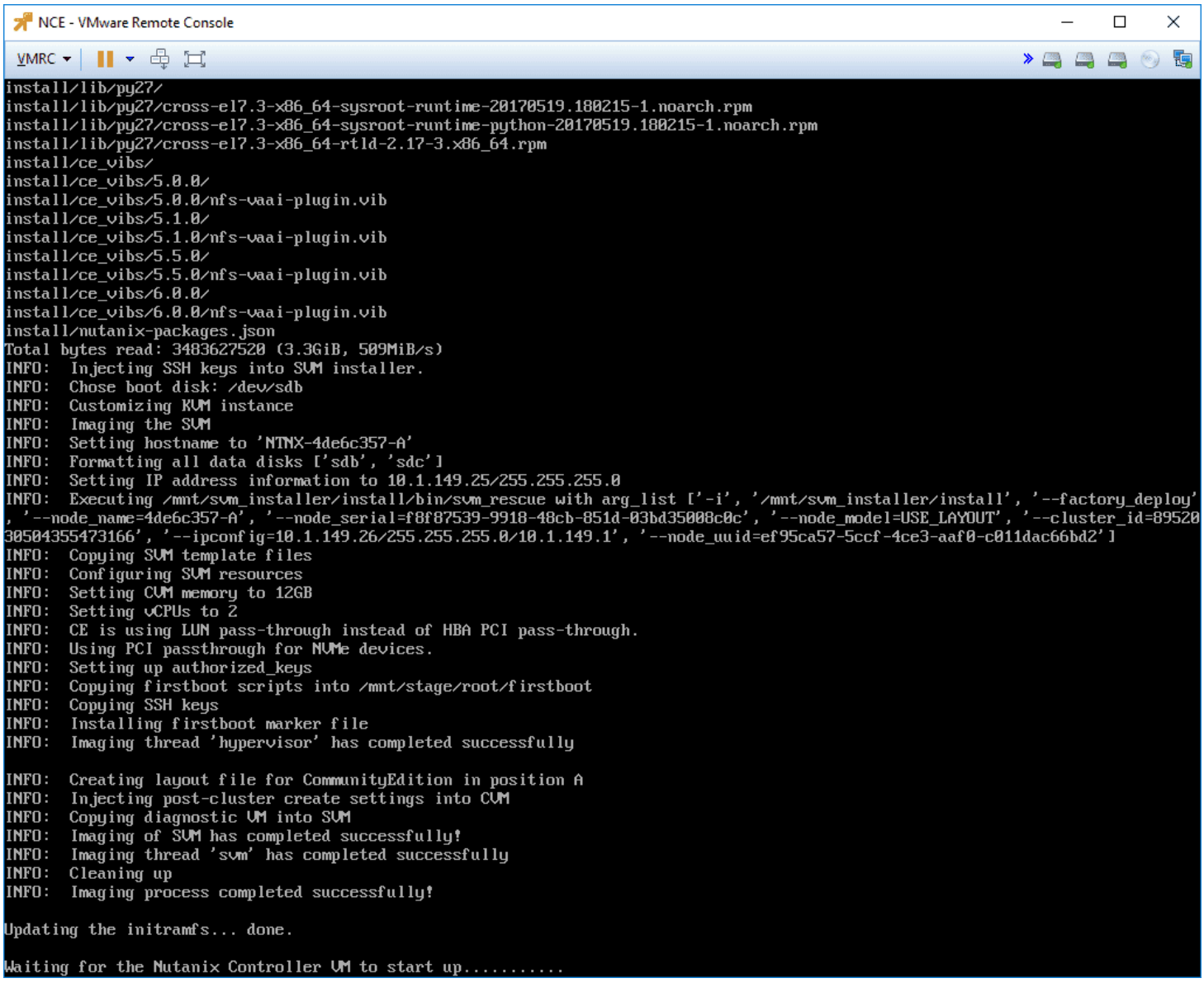
Configuring network cluster and EULA for the Nutanix CE installer

Installation of the Nutanix hypervisor begins.



Installation of the Nutanix CE in the VM begins

The CVM is provisioned and the installer waits for it to successfully boot.



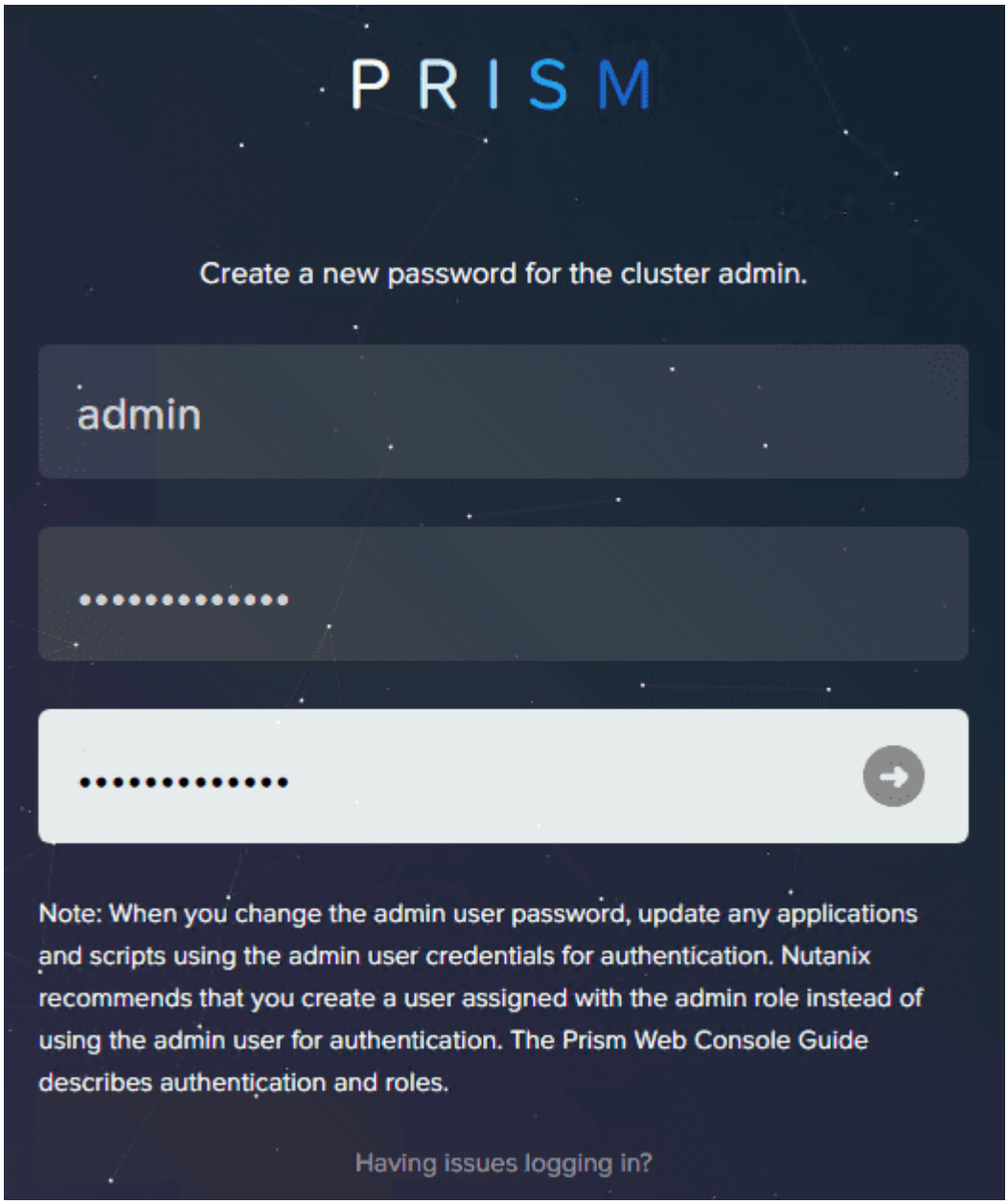
Waiting for the Nutanix CE controller VM to start after installation in VMware vSphere ESXi

Logging into Nutanix CE Prism Interface

After the installation finishes, you can browse to the CVM IP address **https://<your CVM IP>:9440** and login to the PRISM interface. Default user and password:

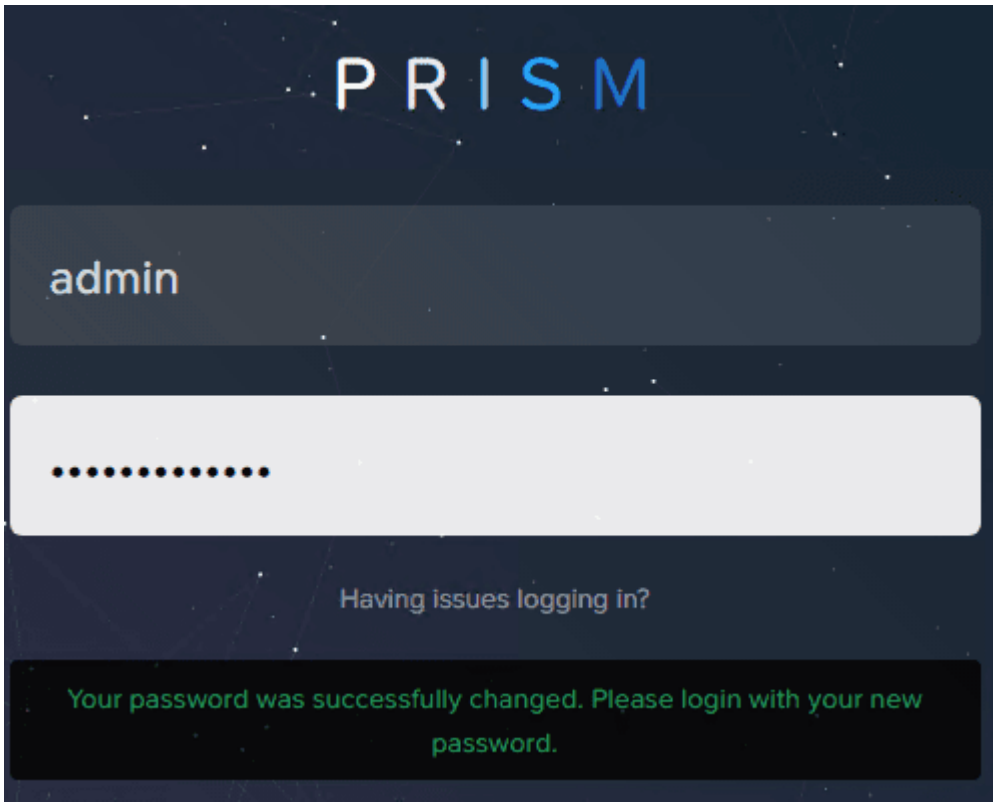
- **admin**
- **nutanix/4u**

This will immediately prompt you for a password change.



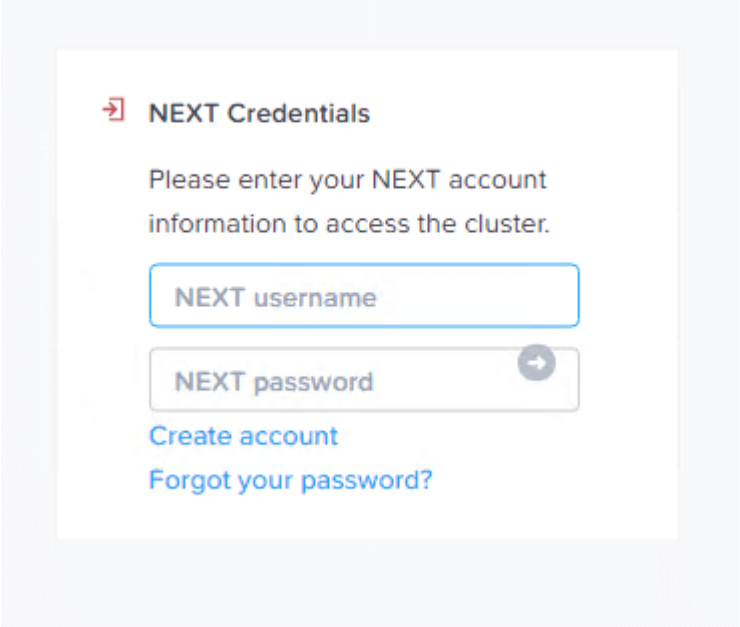
Logging intot he Nutanix CE interface with the default credentials

After changing your admin password, login to the interface.



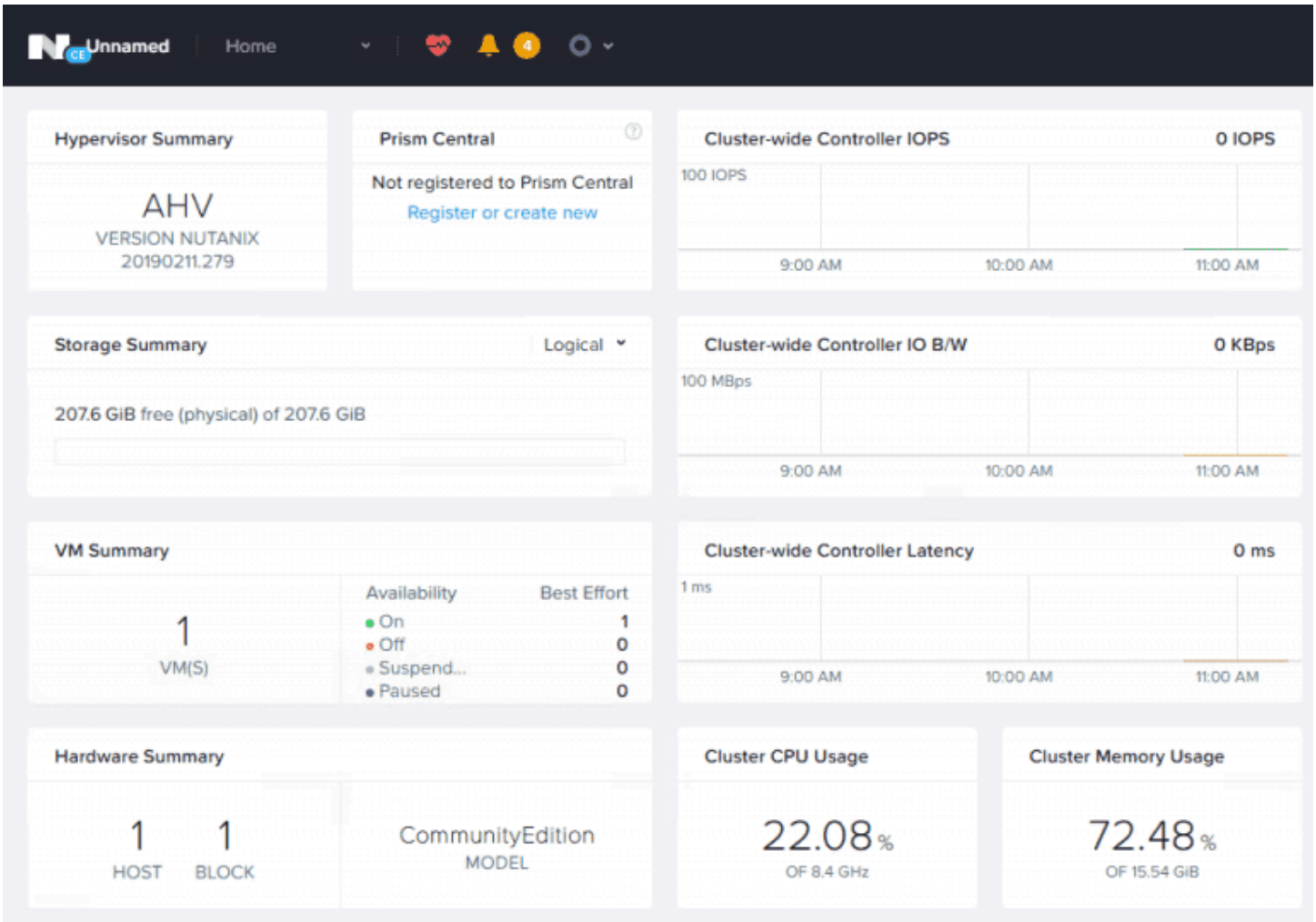
Prompted to change the default password for Nutanix CE

After logging into the main web interface, you will be asked to login to your Nutanix **NEXT** account. This should be the same account you signed up with to download the Nutanix CE bits.



Prompted to login to the Nutanix CE NEXT account

Afterward, you will see your Nutanix CE Prism interface dashboard. Success!



Want to Deploy via OVF?

*****Update 4.26.2018 ***** A great fellow vExpert and a very sharp guy, Jorge de la Cruz, has a great method of installing the Nutanix Community Edition via a custom OVF template file that he has built. This certainly shortcuts the process of modifying files yourself and streamlines the overall process. Check out his post here.

Concluding Thoughts

As you can see the process to Install Nested Nutanix CE in VMware vSphere ESXi 6.7 Update 1 is a bit lengthy, but straightforward. After downloading and getting the required files, the process basically consists of creating the new VM for Nutanix CE and customizing the hardware appropriately. After that, you then install Nutanix CE. If you want to play around with the Nutanix CE AHV hypervisor and have the space in your home lab or other vSphere environment, this is a great way to see and play around with AHV configuration and features. Stay tuned for more Nutanix CE AHV adventures in the lab.

Run into an issue with your Nutanix CE nested VMs not booting after installing in VMware vSphere? Check out the resolution here:

- Guest VM Running in Nested Nutanix CE on VMware vSphere Won't Boot

VMware NSX Home Lab Setup

June 2, 2020
NSX-T

Supermicro-mini-tower-server-for-VMware-NSX-home-lab

When you work in the virtualization field, networking, or information technology in general, having a home lab can be one of the best investments that you can make in your career and your learning experience. While many are moving towards utilizing cloud resources for labs and such, there is just something that is lost in my opinion by not having your own hardware to get your hands on. Understanding how things go together and understanding how to troubleshoot up the networking stack can provide you with valuable skills in your job and applied to other disciplines. I have been running a VMware home lab now for a few years and it has been a great tool for learning, POC'ing, playing around with technology, and just having fun in general. One of the technologies that I working with and enjoy learning and honing skills is VMware NSX. In this post of a series of forthcoming posts, I am going give you guys an overview of VMware NSX home lab setup and diferent aspects of configuring yourself a learning environment for VMware NSX.

This will consist of my recommendations for:

1. **Hardware**
2. **Software/licensing**
3. **Installation**

Hardware – VMware NSX nested home lab

By far, the easiest and one of the most efficient ways to run your VMware NSX home lab is running nested virtual machines either inside of VMware Workstation or VMware vSphere ESXi on a hypervisor host with nested ESXi hosts.

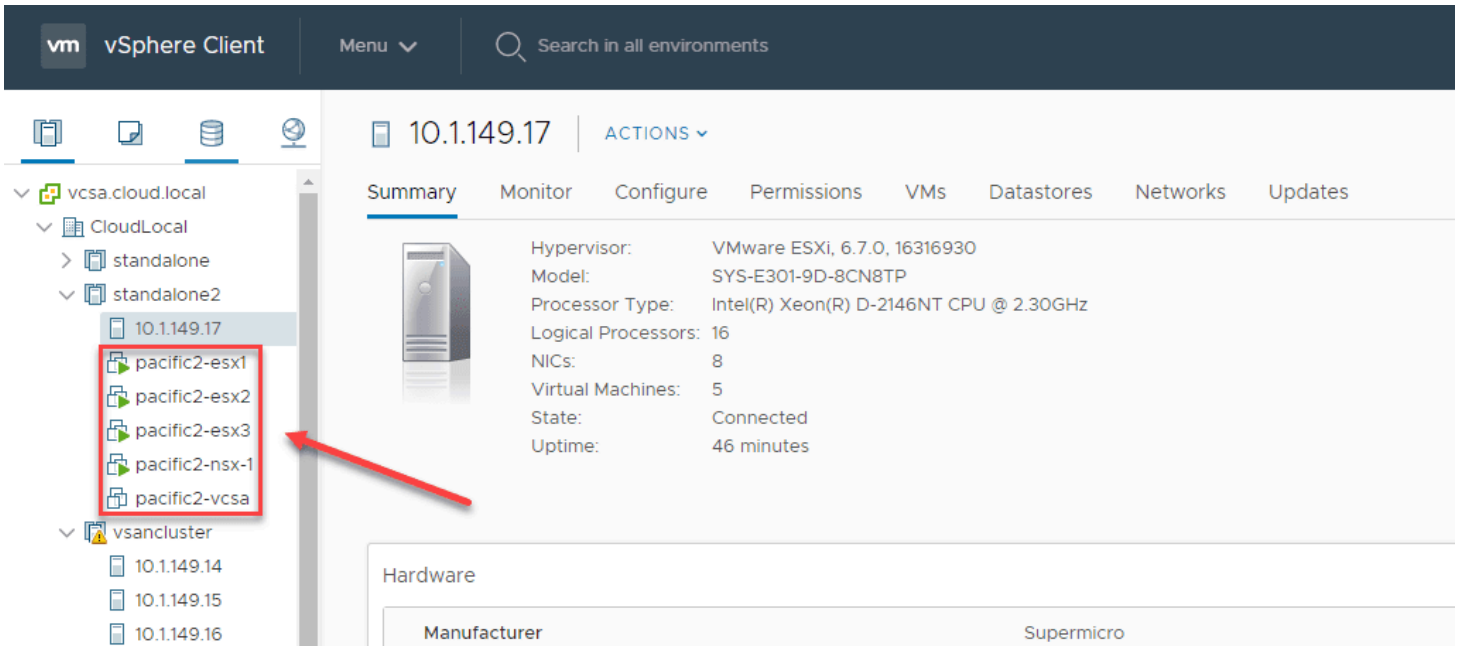
With nested virtualization, you can essentially build "datacenters" of ESXi hosts inside your host hypervisor. "Nested virtualization" essentially means that you are running a hypervisor inside another hypervisor.

These Supermicro servers have plenty of horsepower to run these types of nested virtualization labs.

Supermicro mini tower server for VMware NSX home lab

The server that is referenced above comes with 64 GB of memory. With previous versions of vSphere ESXi such as 6.7 running VMware NSX 2.4-2.5, the memory here would probably be sufficient to have a nested three host cluster running vSAN along with the NSX Manager and an Edge appliance.

However, with vSphere 7 and NSX-T 3.0, you will want to have more RAM than this. This is why, even though you can use a laptop running VMware Workstation to host nested workloads, the limiting factor there is generally RAM.



Nested ESXi NSX-T lab running on a Supermicro E301-9D

As an example of the resources you will need for a VMware NSX home lab setup, you William Lam lists the following as requirements for his vGhetto automated deployed for vSphere 7 including NSX-T 3.0.

Resource Requirements

- Compute
  - Ability to provision **VMs with up to 8 vCPU**
  - Ability to provision **up to 116-140 GB of memory**
- Network
  - Single Standard or Distributed Portgroup (Native VLAN) used to deploy all VMs
    - 6 x IP Addresses for VCSA, ESXi, NSX-T UA and Edge VM
    - 5 x Consecutive IP Addresses for Kubernetes Control Plane VMs
    - 1 x IP Address for T0 Static Route
    - 32 x IP Addresses (/27) for Egress CIDR range is the minimum (must not overlap with Ingress CIDR)

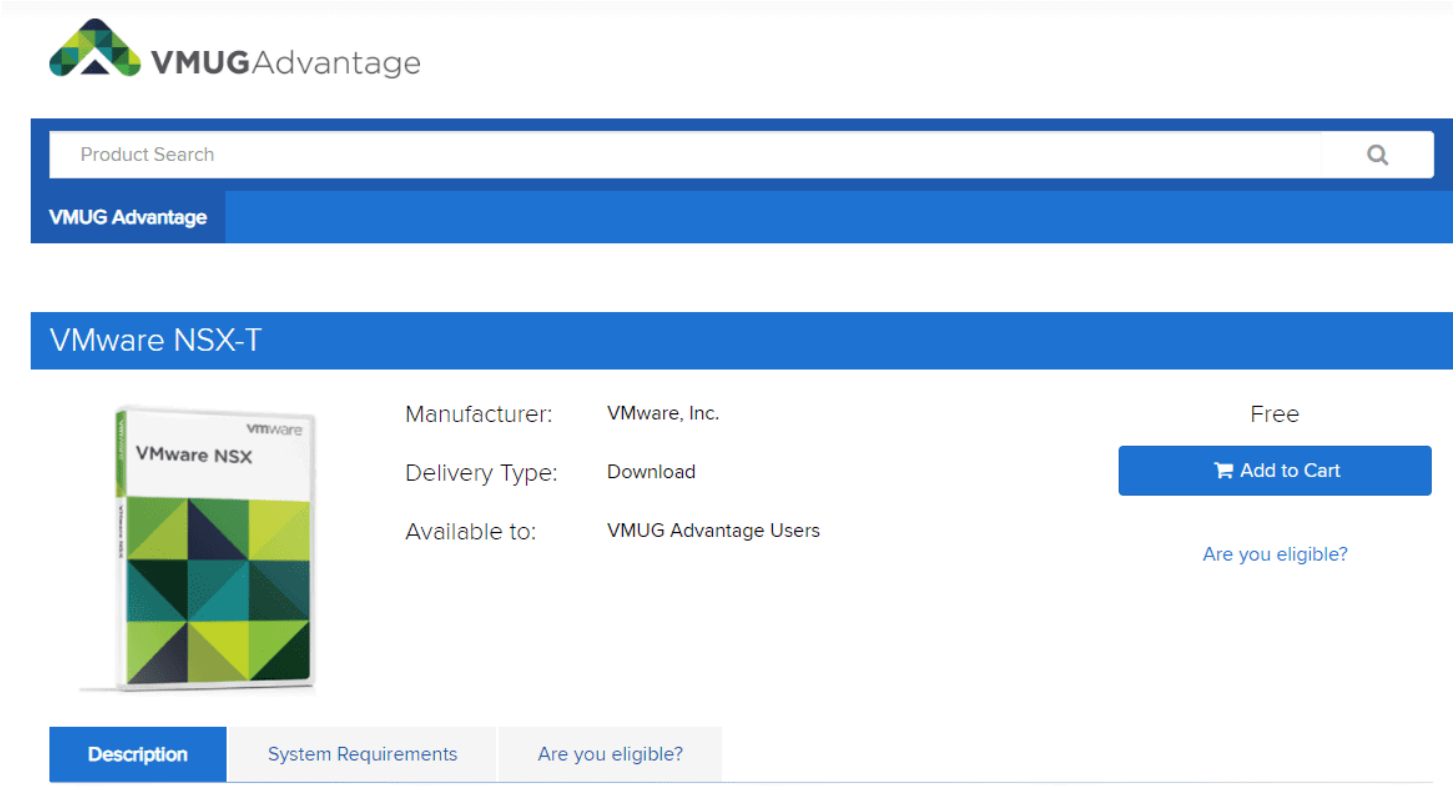
    - 32 x IP Addresses (/27) for Ingress CIDR range is the minimum (must not overlap with Egress CIDR)
    - All IP Addresses should be able to communicate with each other
- Storage
    - Ability to provision up to **1TB of storage**

Check out my write up on building a Supermicro VMware home lab here where I detail the servers I use and reasons:

- Supermicro VMware home lab

Software – VMUG Licensing

There is really not a better deal out there than you will find with VMUG Advantage EvalExperience. With the EvalExperience component of the VMUG Advantage subscription for a mere $200 (less if you catch it at the right time as they typically run 10% off coupons often), you can have access to most of the catalog of VMware products, including the recently added NSX-T 3.0! It is an amazing value and one that if you are serious about learning at home, you just can't beat it.



VMUG Advantage EvalExperience for VMware NSX home lab setup

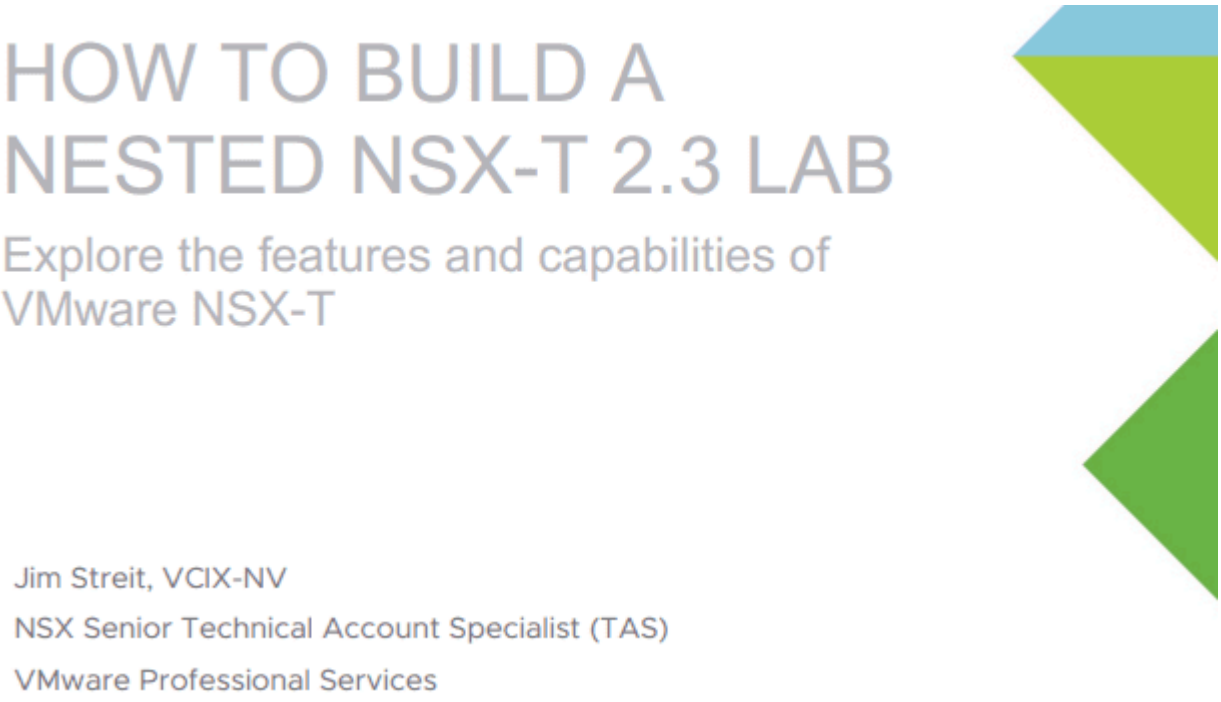
With the VMUG licensing, you will have access to all the software and licensing components you will need for your VMware NSX home lab with the exception of Windows licensing if you want to have something other than evaluation licensing there.

VMware NSX Home Lab Installation

Ok, now you have hardware and software. What about the installation and configuration of your VMware NSX home lab? Well, there are a couple of great resources out there that I highly recommend.

- VMware How to build a nested NSX-T 2.3 lab
- William Lam's nested lab deployment script

The first resource – VMware how to build a nested NSX-T 2.3 lab is a guide that VMware released that will step you through building your NSX-T home lab and gives you the guidance on how to set this up. This includes how to install your hosts, configure your networking, setup transport zones, deploy edge appliances, and much more.



VMware how to build a nested NSX-T 2.3 lab guide

If you want to get into the nuts and bolts of the deployment and really see how to fit the pieces together and how they work the way they do, this is a great place to start. Working through the lab guide will help you wrap your head around many of the NSX-T concepts and workflows.

What though if you are looking to just get the lab built and have your hosts deployed and ready to go so you can start playing around with the cool NSX-T functionality including distributed firewalling, IDS, and other things?

William Lam's vGhetto Automated vSphere with Kubernetes Lab Deployment is definitely a great resource! William has over the years provided many great tools and utilities including his automated nested lab deployment scripts.

With the current version of the script, you will be able to fully deploy a vSphere 7 lab with Kubernetes and NSX-T 3.0 without any input from you other than customizing the script file in the beginning to fit your environment.

Additionally, in the directions given in the Github download of the vGhetto script, you can customize the script to deploy only vSphere 7 and no Kubernetes or NSX-T, or you can have it deploy vSphere 7 with NSX-T and no Kubernetes. So, you can customize the script for resource availability.

For a strictly VMware NSX home lab, you can opt out of deploying Kubernetes which will reduce the resource requirements considerably.

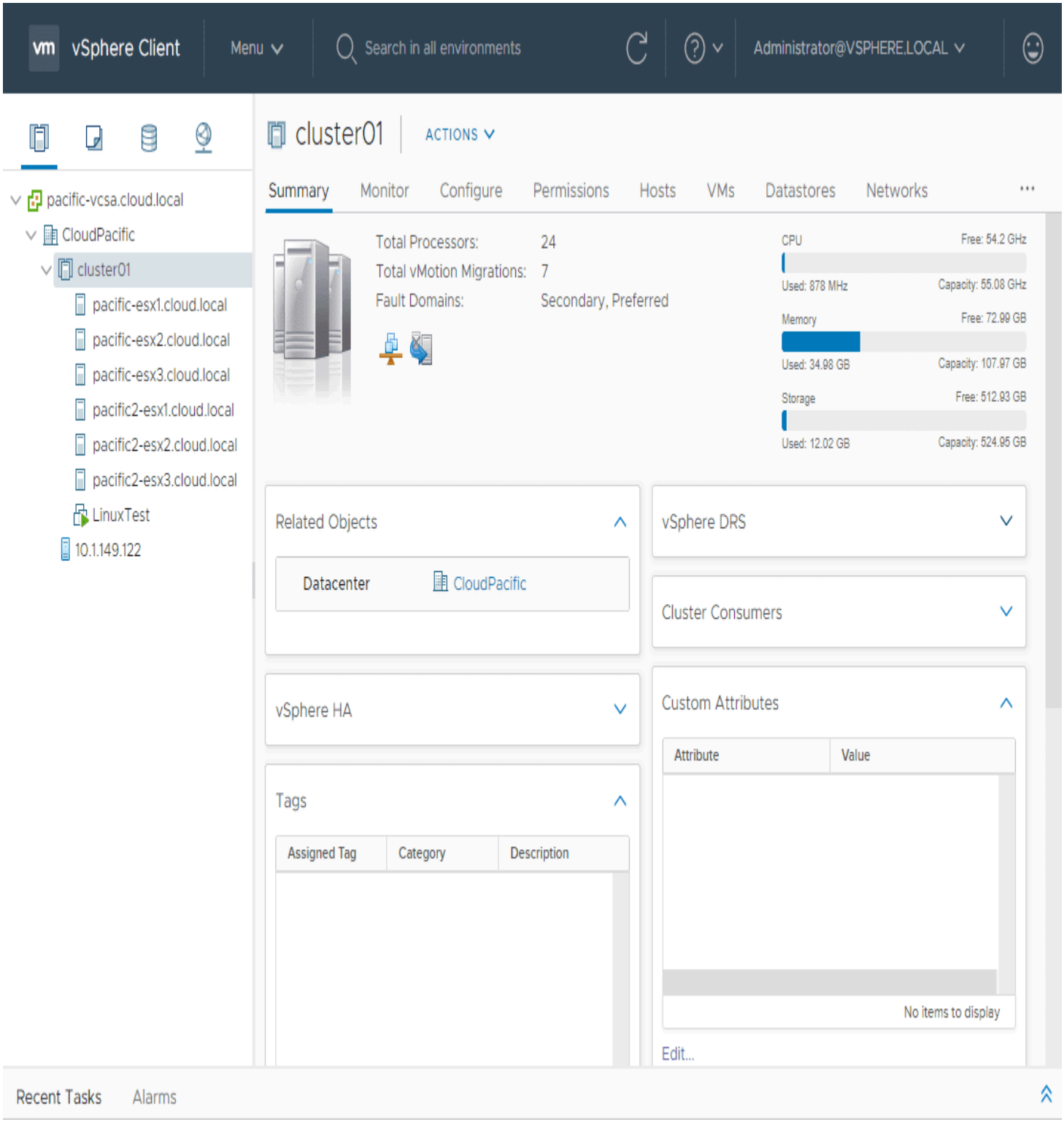
Concluding Thoughts

If you are interested in VMware NSX home lab setup and what would be required for you to have an environment at home for learning and playing around with VMware NSX-T, hopefully this post with a few of my thoughts around hardware, software/licensing, and installation will help fill in a few gaps of what you need and how to go about it.

Check out Supermicro VMware home lab to see which servers I am using and the reasons, configurations, etc.

Nested ESXi Lab Build Networking and Hardware

September 12, 2020
home lab



Nested-ESXi-lab-environment-with-stretched-vSAN-cluster-with-Witness-host

Many who are learning VMware vSphere and those who may already work with VMware vSphere products each and every day benefit from having a lab environment. I have always been a proponent of lab environments. Even with many touting moving to the cloud and such, there will always be a need for people who actually know what goes on behind the scenes in the data center. Having a lab environment is a great way to build, break, troubleshoot, upgrade, and most of all, learn. To learn VMware vSphere, having a nested ESXi lab is a great way to learn the fundamentals of the VMware vSphere

hypervisor, without breaking the bank on numerous physical lab hosts. Let's take a look at nested ESXi lab build networking and hardware to see how you can go about successfully building your nested ESXi lab from scratch.

What is nested virtualization?

Nested virtualization is where you are basically running a hypervisor "on top of" another hypervisor. Think of the movie "Inception". You can run the ESXi hypervisor as a VM on top of another physical ESXi hypervisor host. Now, you may be wondering why you would want to do this. The answer is simple – labs.

When you run a nested ESXi lab as VMs running on a physical ESXi hypervisor host, you have all of the benefits that you normally have with a VM running inside of vSphere. This includes being able to control and work with the ESXi host VMs as you would any other VM. This allows you to spin up multiple ESXi hosts that can build a nested ESXi **cluster** so you can start playing around with features such as HA, DRS, and vMotion as well as the other enterprise features that you want to work with.

With nested virtualization, you can learn and play around with many great technologies including vSAN, NSX, and stretched clustering.

Build a Nested ESXi Lab

How do you build a nested ESXi lab? There are many different ways that you can build a nested ESXi Lab. One of the first things that you need is hardware to run the nested lab.

This can be as simple as a laptop or workstation class machine running VMware Workstation. VMware Workstation Pro 16 contains great features for running a nested ESXi lab.

With VMware Workstation Pro 16, it supports vSphere 7 as well as containers, so lots of great features for learning new technologies and working with the latest technologies. You can read my post here about VMware Workstation Pro 16:

- VMware Workstation Pro 16 and Fusion 12 New Features

A Dell Precision laptop provides a powerful nested ESXi lab platform

For my purposes, I have the need for a more permanent lab solution with dedicated hardware that I run 24×7. I use Supermicro servers for this purpose. You can check out my post here about using Supermicro servers for a VMware home lab for specifics about the models I use and what I have configured in my lab environment.

- Supermicro VMware Home Lab

Check out my article below on how to setup your Supermicro server to automatically power on and power off to save money on your electric bill.

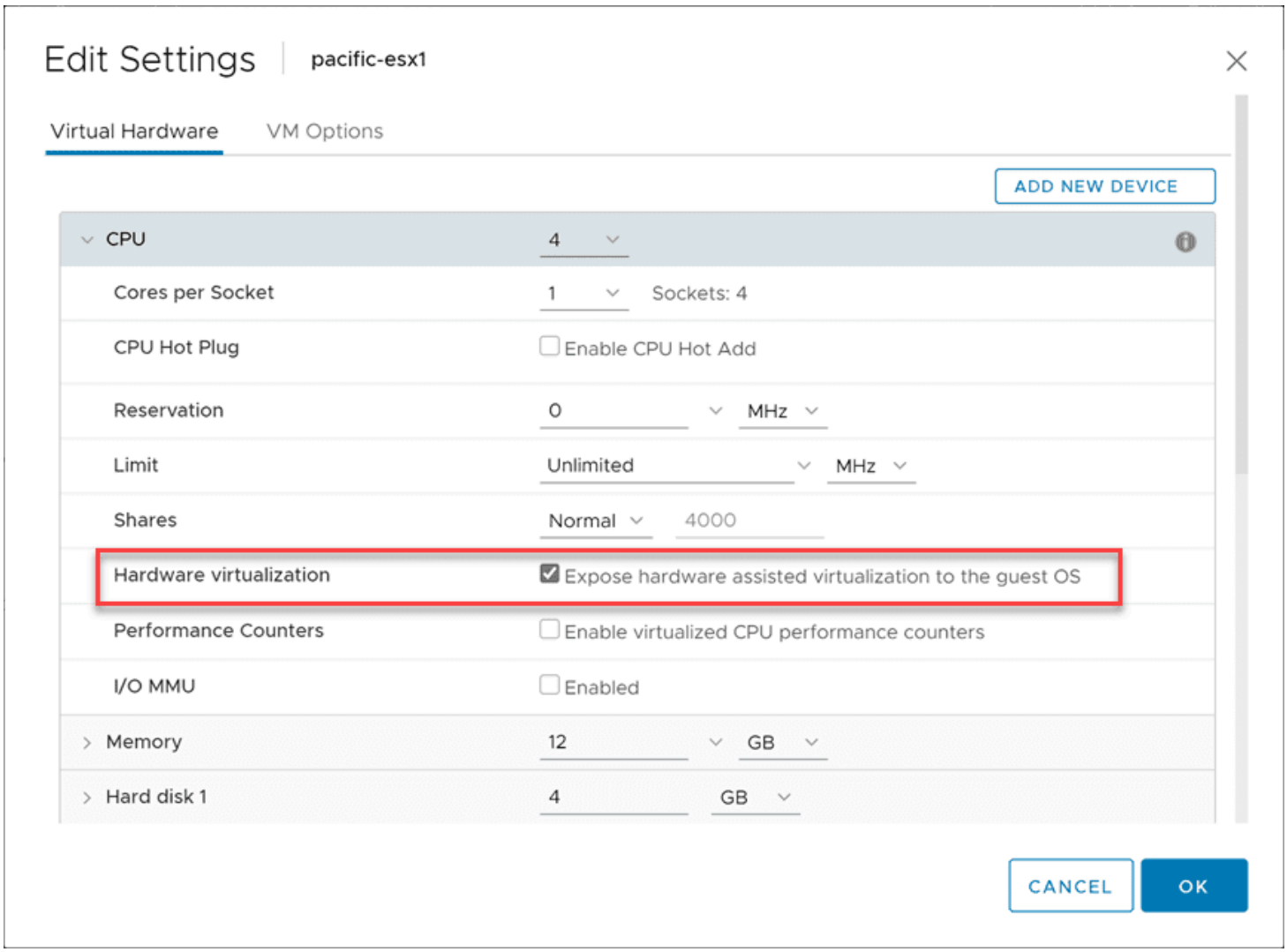
- Supermicro VMware Home Lab Automatic Power On and Power Off

Deploying a Nested ESXi lab

Can you deploy your ESXi hosts one-by-one by simply deploying the ISO into a VM? Yes. This may be the way you want to deploy your ESXi hosts to begin with to see the inner workings of how the install works, initial setup, etc.

If you load your ESXi hosts manually, be sure to set the processor setting:

- Expose hardware assisted virtualization to guest OS



Expose hardware assisted virtualization to the guest

However, if you want to progress to an automated deployment, THE best way to deploy a nested ESXi lab is with William Lam's nested ESXi lab script. William has updated his script to deploy ESXi 7.0.

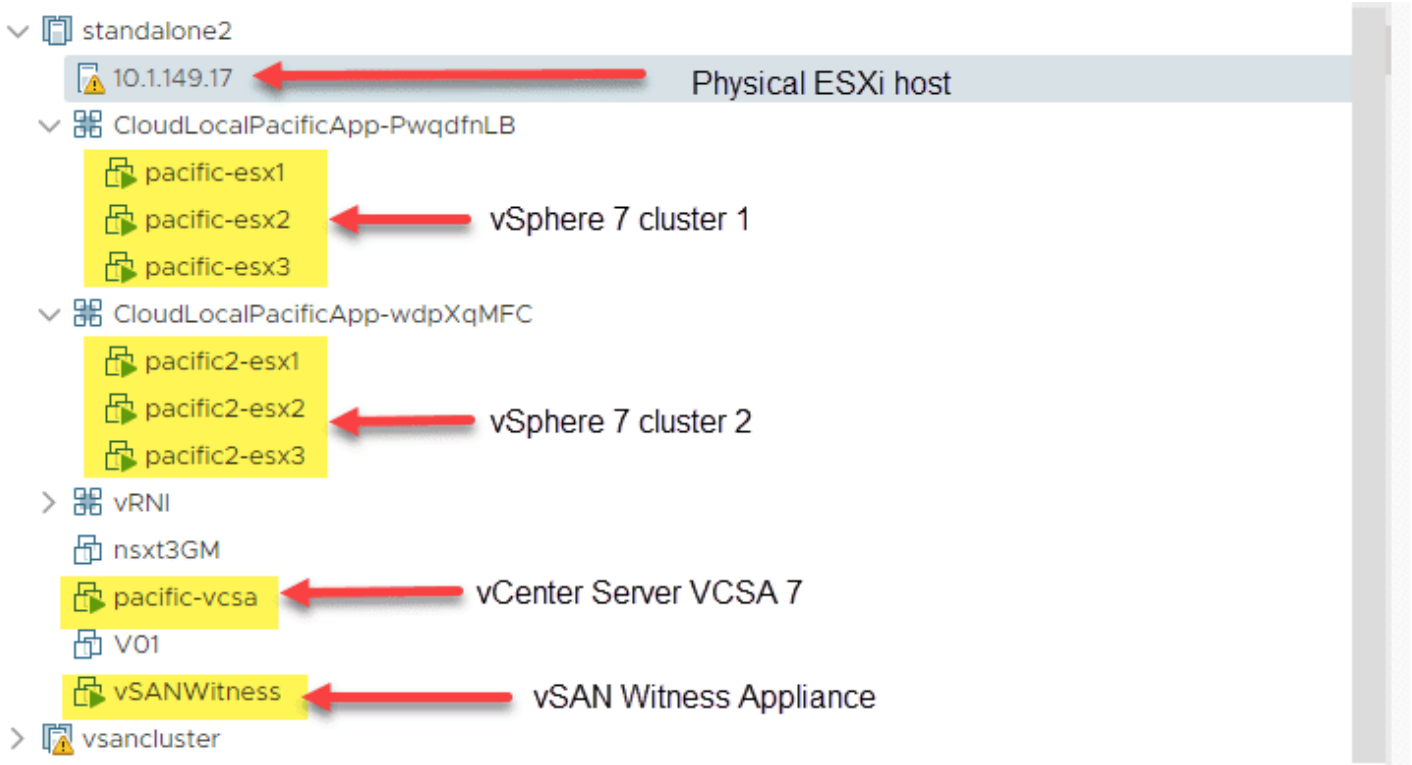
Check out the nested deployment here:

- https://www.virtuallyghetto.com/2020/04/automated-vsphere-7-and-vsphere-with-kubernetes-lab-deployment-script.html

What does a Nested ESXi lab look like?

In case you are having difficulty visualizing what is going on with a nested ESXi lab, let's take a look at one of my nested ESXi labs that I have running on one of my home lab hosts.

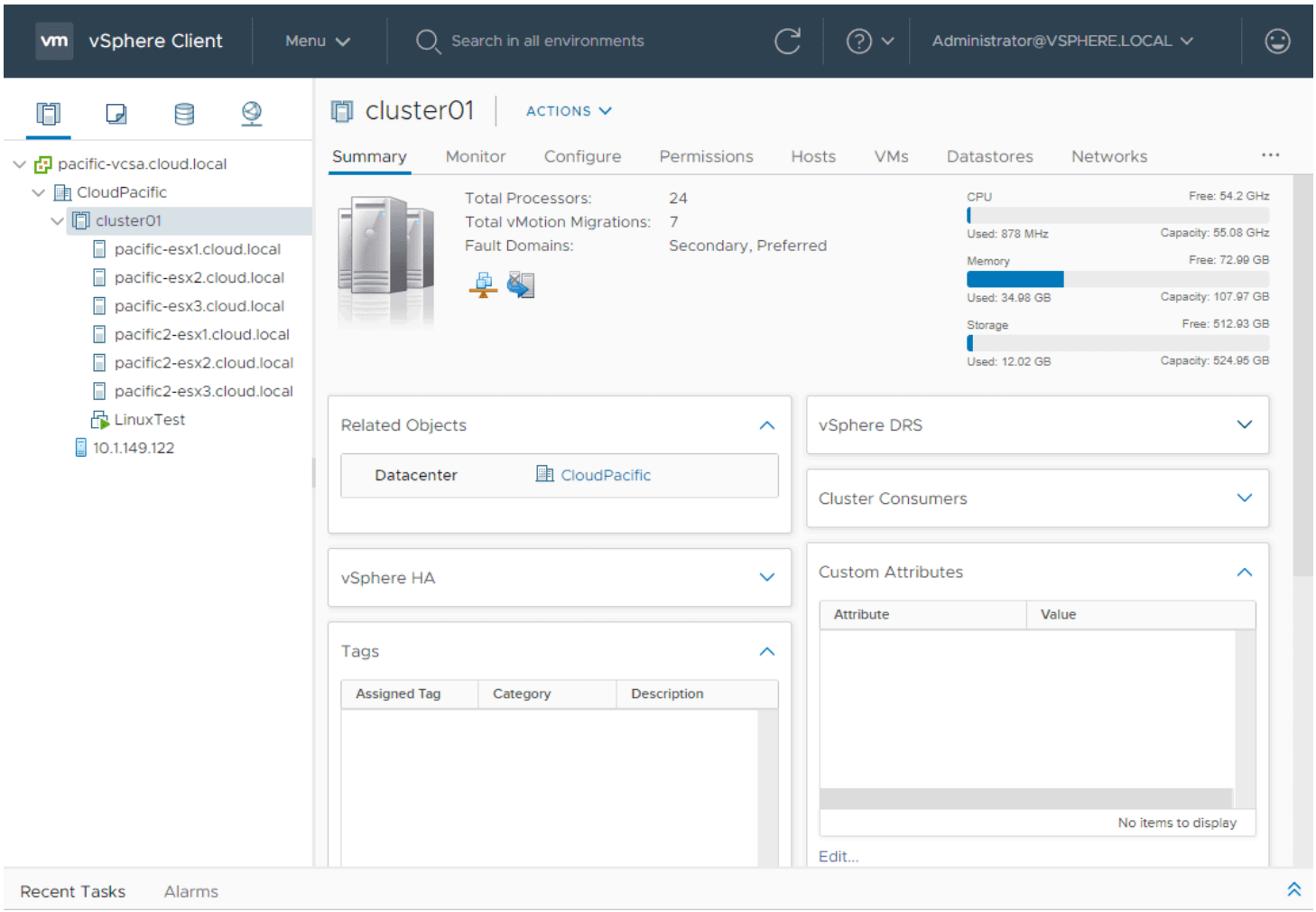
If you notice below, I have two vAPPs running with (3) nested ESXi hosts running inside each vAPP. Each of the vAPPs with ESXi hosts represents a vSphere cluster that I have running. As you can see on the same physical ESXi host, I also have a vCenter Server running as well.



Nested ESXi lab VMs running on a physical ESXi server

What does this look like from the nested ESXi lab side? You can see below, after you login to your nested ESXi lab vCenter Server, it looks like any other vSphere environment would look.

In fact, you can do more complex nested environments such as I have below. Below is a stretched vSAN cluster running "all-flash" vSAN with a Witness node provisioned. So, as you can see, nested environments allow you to do very complex lab environments, testing many different technologies.



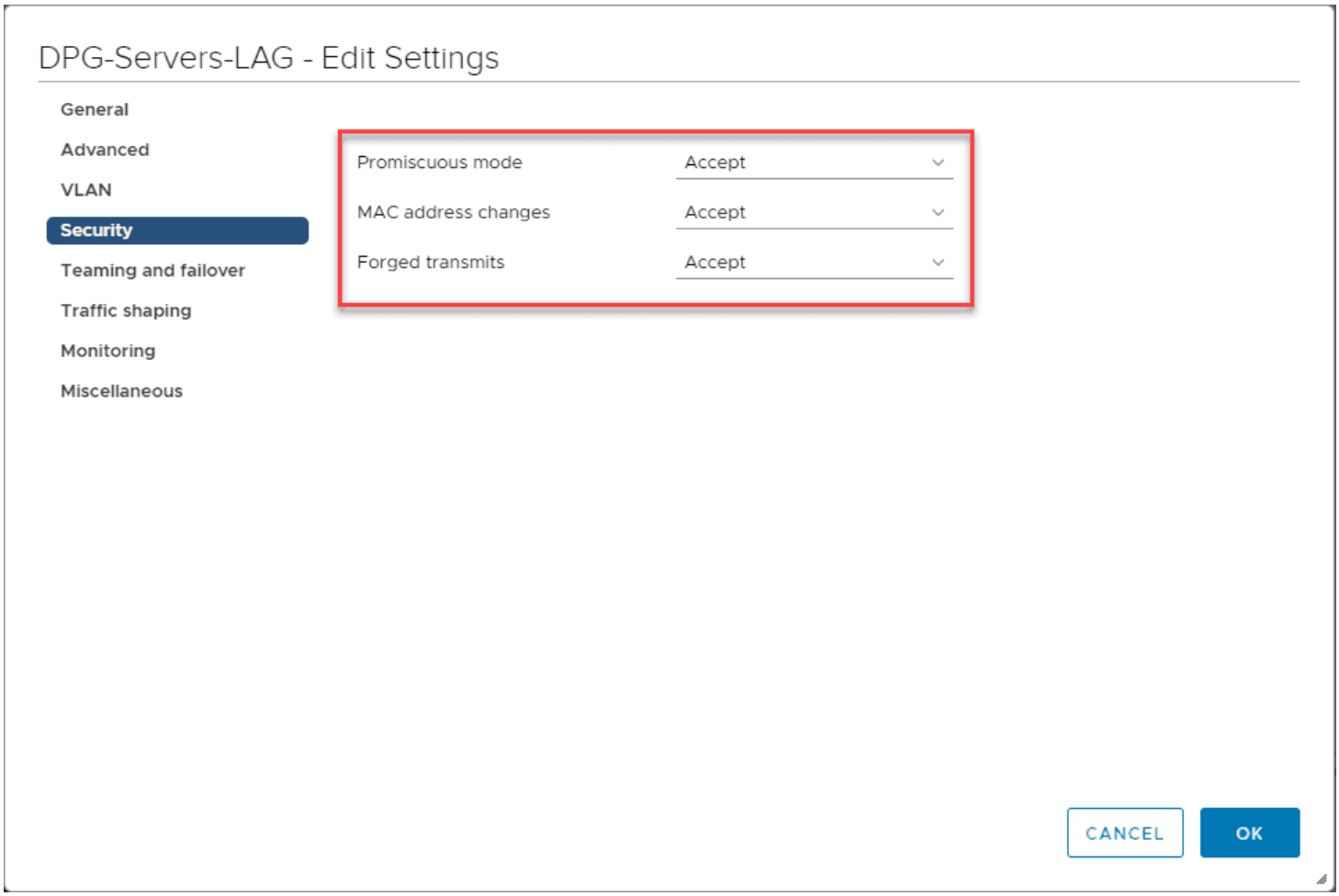
Nested ESXi lab environment with stretched vSAN cluster with Witness host

How to Setup a Nested ESXi Lab Networking

For many, it is kind of a mind bender to visualize the nested ESXi lab networking and how this is configured. However, the easiest way to think of the lab is to think of the fact that your nested ESXi hosts are simply VMs running on your physical ESXi host.

The virtual networking of the ESXi hosts is simply running on top of the vSphere networking on the physical ESXi host. However, there are some special things that need to take place with ESXi networking. Since your nested ESXi lab has the capability of actually running virtual machines **inside** the nested environment, there are settings that must be enabled to make this work properly.

With nested virtualization, the nested ESXi host is hosting traffic for a number of other “nested VMs” that are ran on the nested instance of ESXi. When this is the case, multiple MAC addresses will appear in the source address 802.3 field. Each virtual workload hosted by the nested ESXi host would need to communicate using the virtual ESXi network adapter. These extra MAC addresses are rejected as forged transmits.



Setting the security policy on your virtual switch for nested ESXi lab environment

Chris Wahl has a great writeup on the specifics of why this is needed here:

- How The VMware Forged Transmits Security Policy Works

The solution is to enable promiscuous mode and forged transmits on your virtual switches on the physical ESXi host, that you want to carry traffic for the nested ESXi host. There is another solution that has been covered in great detail by William Lam. That is the ability now of vSphere 6.7 ESXi with the vSphere Distributed Switch to learn the MAC addresses of nested ESXi traffic.

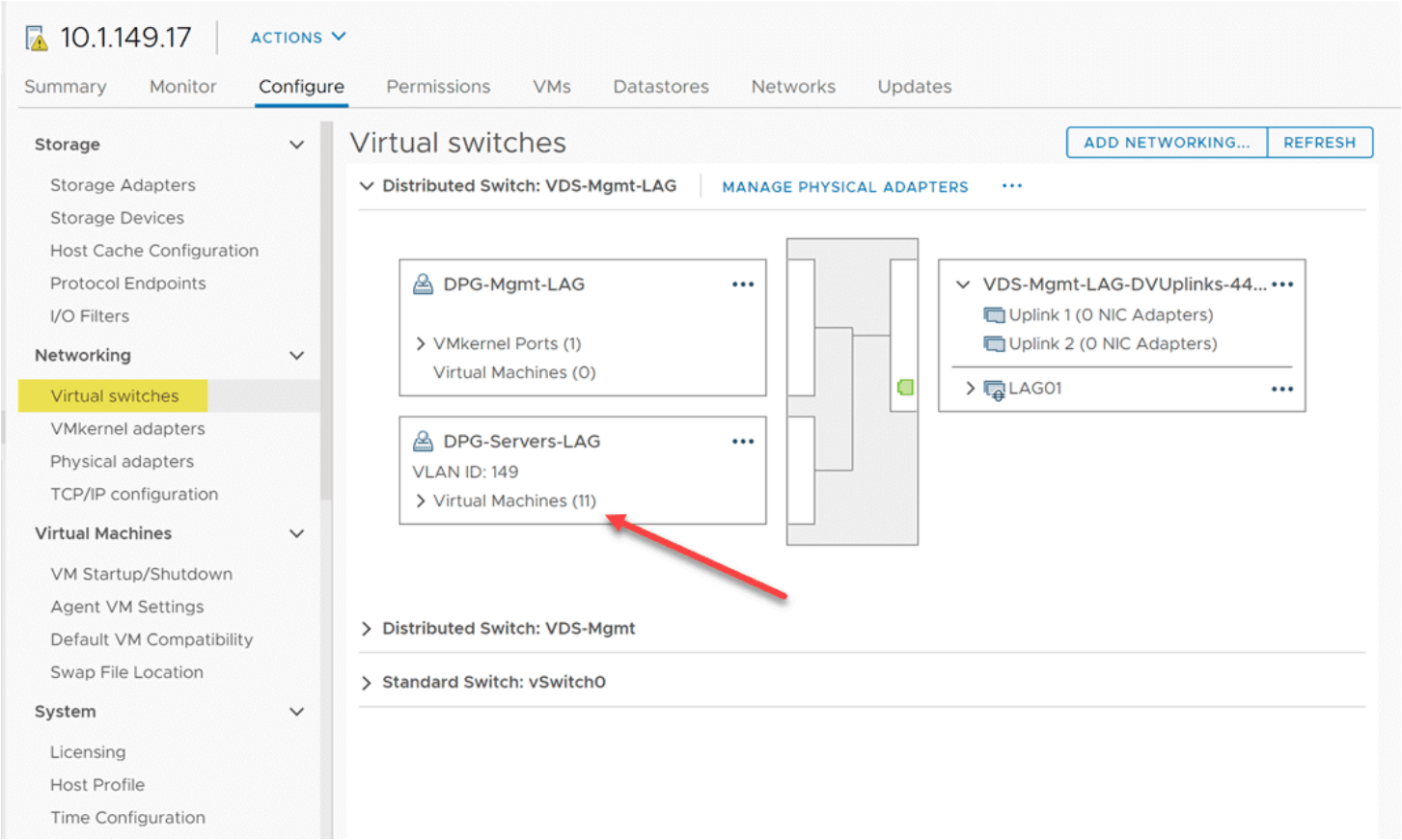
Check out William's post here on that subject:

- Native MAC Learning in vSphere 6.7 removes the need for Promiscuous mode for Nested ESXi

Either solution is required to enable nested virtualization network traffic for VMs that you have running in your nested ESXi environment to pass traffic to your real physical network.

Now, thinking about the physical host networking, you can house your nested ESXi lab on any VSS or VDS port group of your choosing. The vSwitch that carries the nested ESXi lab traffic can be untagged frames or VLAN tagged frames.

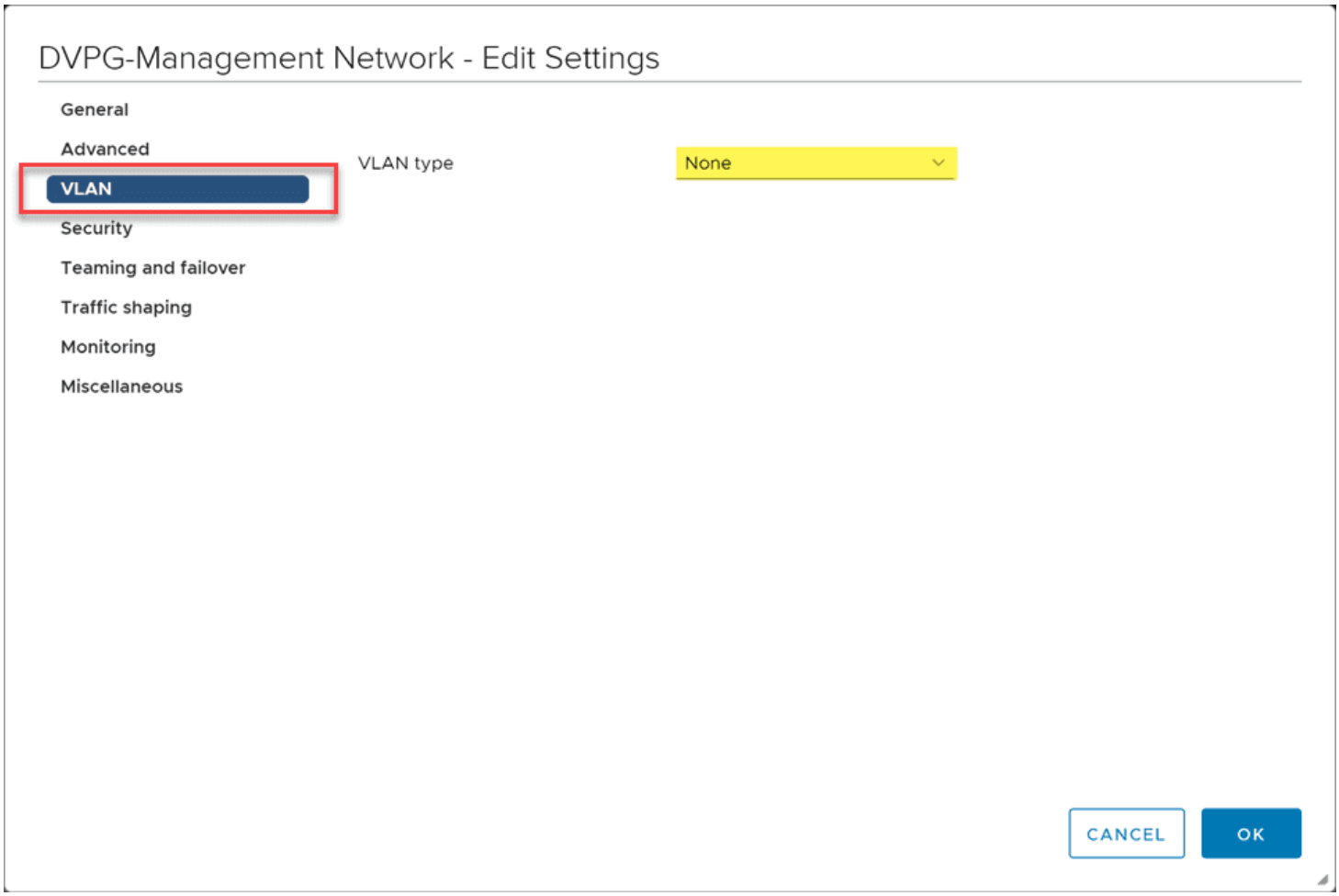
Below, the port group my nested ESXi hosts will be connected to is a VLAN-backed port group called **Servers**.



Physical ESXi host networking

Now, the nested ESXi hosts have a vSphere Distributed Switch provisioned that is running on top of the VDS on the physical ESXi host. Note, you can't tag frames in your nested ESXi host configuration. However, keep in mind the ESXi traffic from the nested server will be tagged with the VLAN of the physical ESXi host port group it is attached to.

So, with that being said you can split off your "virtual network adapters" connected to your nested ESXi hosts and connect those to different port groups on your physical ESXi hosts to connect nested VMs to different VLANs.



Nested ESXi host vSphere Distributed Switch settings

Nested ESXi Lab Licensing

I am and have always been a fan of the VMUG Advantage subscription. You simply won't find a better value for a home lab than the VMUG subscription.

With a VMUG subscription, you get vSphere licensing for products like: vSphere, vSAN, NSX-V & T, VCF, SRM, Horizon, and others.

- Price $200! You won't find a better deal than this anywhere. Period.

Read my articles on VMUG coverage, subscription, features, etc here:

- VMware Cloud Foundation vCF 4.0 added to VMUG Advantage
- VMware Cloud Foundation added to VMUG Advantage Eval Experience
- VMware TestDrive added to VMUG Advantage Subscriptions
- VMware Site Recovery Manager SRM added to VMUG Advantage
- VMUG Advantage adds vRealize Automation 7.3 Enterprise
- VMUG VMware vRNI bundled with NSX offering

Wrapping Up

Hopefully this Nested ESXi Lab Build Networking and Hardware discussion will help any who are wanting to delve into setting up a nested ESXi lab. The process is really easy.

You can get started with just a bit of hardware and even just VMware Workstation. Or if you have dedicated hardware you can allocate, you can load ESXi on physical hardware and then load your nested ESXi hosts on top of that.

There are great resources available such as a VMUG subscription, William Lam's vSphere 7 lab scripts, and other community resources.