

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.

Copyright © 2024 by Brandon Lee
All rights reserved.

No portion of this book may be reproduced in any form without written permission from the publisher or author, except as permitted by U.S. copyright law.

This publication is designed to provide accurate and authoritative information in regard to the subject matter covered. It is distributed with the understanding that neither the author nor the publisher is engaged in rendering any type of professional services. While the publisher and author have used their best efforts in preparing this book, they make no representations or warranties with respect to the accuracy or completeness of the contents of this book and specifically disclaim any implied warranties of merchantability or fitness for a particular purpose. No warranty may be created or extended by sales representatives or written sales materials. The technical advice and strategies contained herein may not be suitable for your situation. You should consult with a technical professional when appropriate in your environment. Neither the publisher nor the author shall be liable for any loss of profit or any other commercial damages, including but not limited to special, incidental, consequential, personal, or other damages.

Table of Contents

[Nested ESXi install in Proxmox: Step-by-Step](#)
[Nested Proxmox VMware installation in ESXi](#)
[How to Enable Proxmox Nested Virtualization](#)
[XCP-NG in VMware nested virtualization](#)
[Install Harvester in VMware ESXi](#)
[Install Minikube in Windows Server 2022 using Winget](#)
[VirtualBox Nested Virtualization with VMware ESXi](#)
[Install KVM Ubuntu Server VMware Workstation](#)
[Install and Configure VMware vSAN Witness Appliance in VMware Workstation](#)
[VMware vs Hyper-V Nested Virtualization](#)
[Install Nested Nutanix CE in VMware vSphere ESXi 6.7 Update 1](#)
[VMware NSX Home Lab Setup](#)
[Nested ESXi Lab Build Networking and Hardware](#)

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

- [Nested Virtualization in Proxmox](#)
- [Preparing your Proxmox VE host to enable nested virtualization for ESXi](#)
- [Creating the ESXi VM in Proxmox](#)
- [Step-by-Step Installation of Nested ESXi](#)
- [Managing Virtual Machines in a Nested Setup](#)
 - [Using advanced features in nested VMs](#)
- [Troubleshooting Common Issues in Nested Environments](#)

- [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.

Create: Virtual Machine ✕

General OS System Disks CPU Memory Network Confirm

Node: pve01 ▼ Resource Pool: ▼

VM ID: 104 ⬆️⬇️⬆️

Name: esxionpve 📘

Start at boot:

Start/Shutdown order: any

Startup delay: default

Shutdown timeout: default

Tags

No Tags +

🔍 Help Advanced Back Next

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.

Create: Virtual Machine ✕

General **OS** System Disks CPU Memory Network Confirm

Use CD/DVD disc image file (iso) Guest OS:

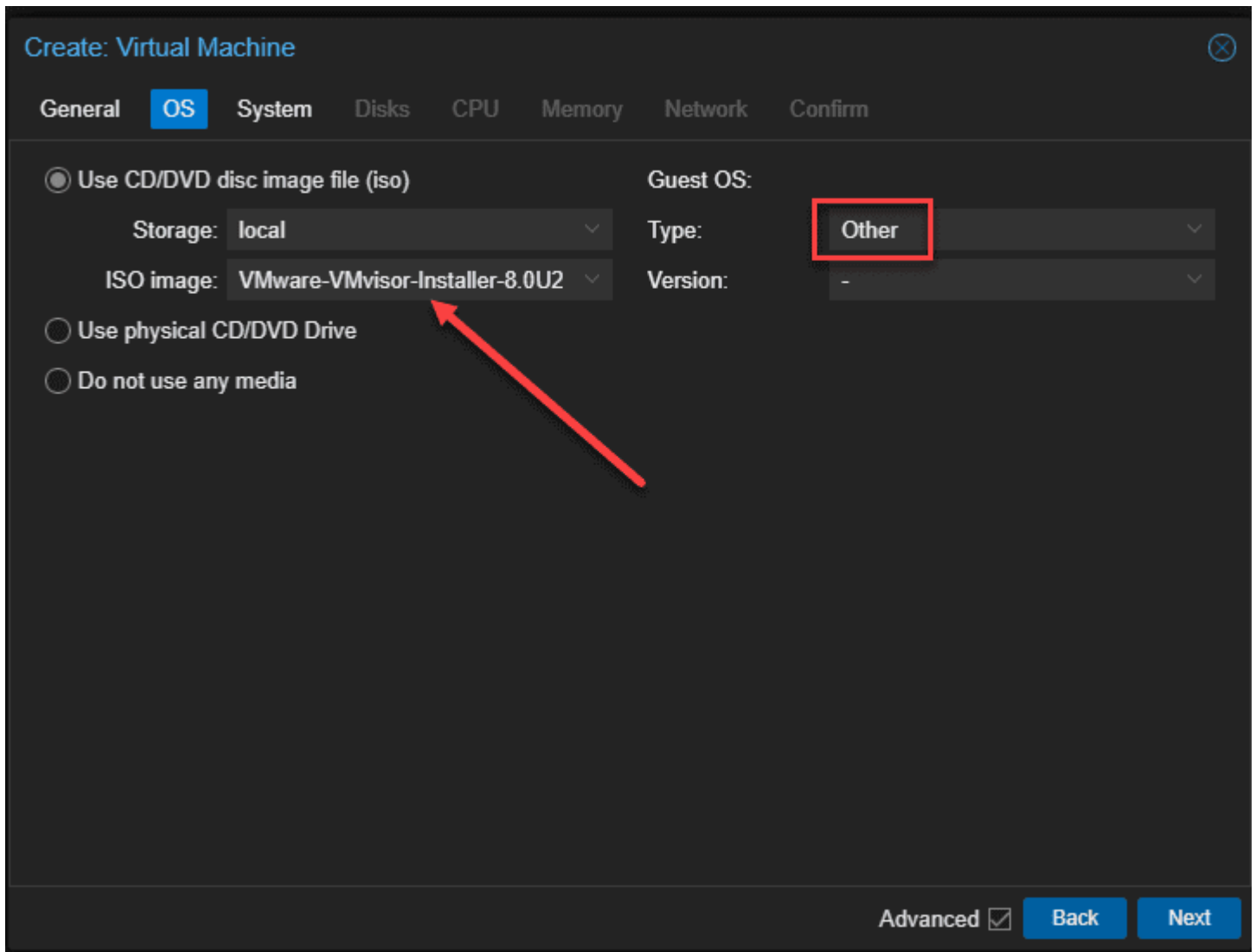
Storage: local Type: **Other**

ISO image: VMware-VMvisor-Installer-8.0U2 Version: -

Use physical CD/DVD Drive

Do not use any media

Advanced Back Next



Select your esxi iso image under the os tab

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

Create: Virtual Machine ✕

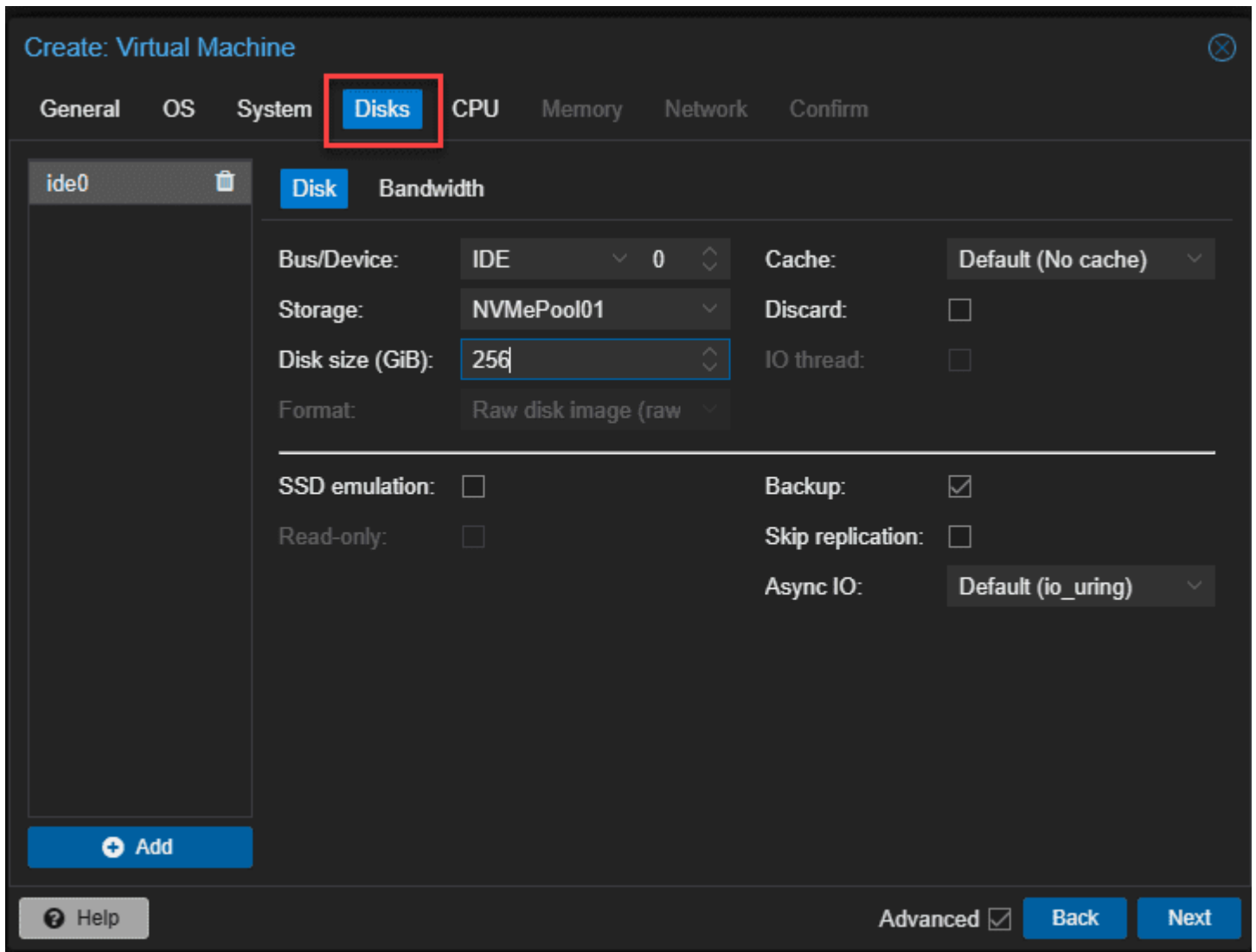
General OS **System** Disks CPU Memory Network Confirm

Graphic card:	Default	SCSI Controller:	VirtIO SCSI single
Machine:	Default (i440fx)	Qemu Agent:	<input type="checkbox"/>
Firmware			
BIOS:	Default (SeaBIOS)	Add TPM:	<input type="checkbox"/>

Help Advanced Back Next

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.

Create: Virtual Machine

General OS System Disks **CPU** Memory Network Confirm

Sockets: 1 Type: x86-64-v2-AES

Cores: 4 Total cores: 4

VCPUs: 4 CPU units: 100

CPU limit: unlimited Enable NUMA:

CPU Affinity: All Cores

Extra CPU Flags:

Default	- ○ ● ○ +	md-clear	Required to let the guest OS know if MDS is mitigated correctly
Default	- ○ ● ○ +	pcid	Meltdown fix cost reduction on Westmere, Sandy-, and IvyBridge Intel CPUs
Default	- ○ ● ○ +	spec-ctrl	Allows improved Spectre mitigation with Intel CPUs
Default	- ○ ● ○ +	ssbd	Protection for "Speculative Store Bypass" for Intel models
Default	- ○ ● ○ +	ibpb	Allows improved Spectre mitigation with AMD CPUs
Default	- ○ ● ○ +	virt-ssbd	Basis for "Speculative Store Bypass" protection for AMD models

Help Advanced Back Next

Cpu settings for the esxi nested vm

Configure your memory.

Create: Virtual Machine ⌵

General OS System Disks CPU **Memory** Network Confirm

Memory (MiB): ⌵

Minimum memory (MiB): ⌵

Shares: ⌵

Ballooning Device:

ⓘ Help Advanced Back Next

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.

Create: Virtual Machine ✕

General OS System Disks CPU Memory **Network** Confirm

No network device

Bridge: Model:

VLAN Tag: MAC address:

Firewall:

Disconnect: Rate limit (MB/s):

MTU: Multiqueue:

Advanced

Setting the network adapter to e1000

Confirm your configuration and click **Finish**.

Create: Virtual Machine

General OS System Disks CPU Memory Network **Confirm**

Key ↑	Value
cores	4
cpu	x86-64-v2-AES
ide0	NVMePool01:256
ide2	local:iso/VMware-VMvisor-Installer-8.0U2-22380479.x86_64__1_.iso,media=cdrom
memory	16384
name	esxionpve
net0	e1000,bridge=vibr0,tag=149,firewall=1
nodename	pve01
numa	0
ostype	other
scsihw	virtio-scsi-single
sockets	1
vmid	104

Start after created

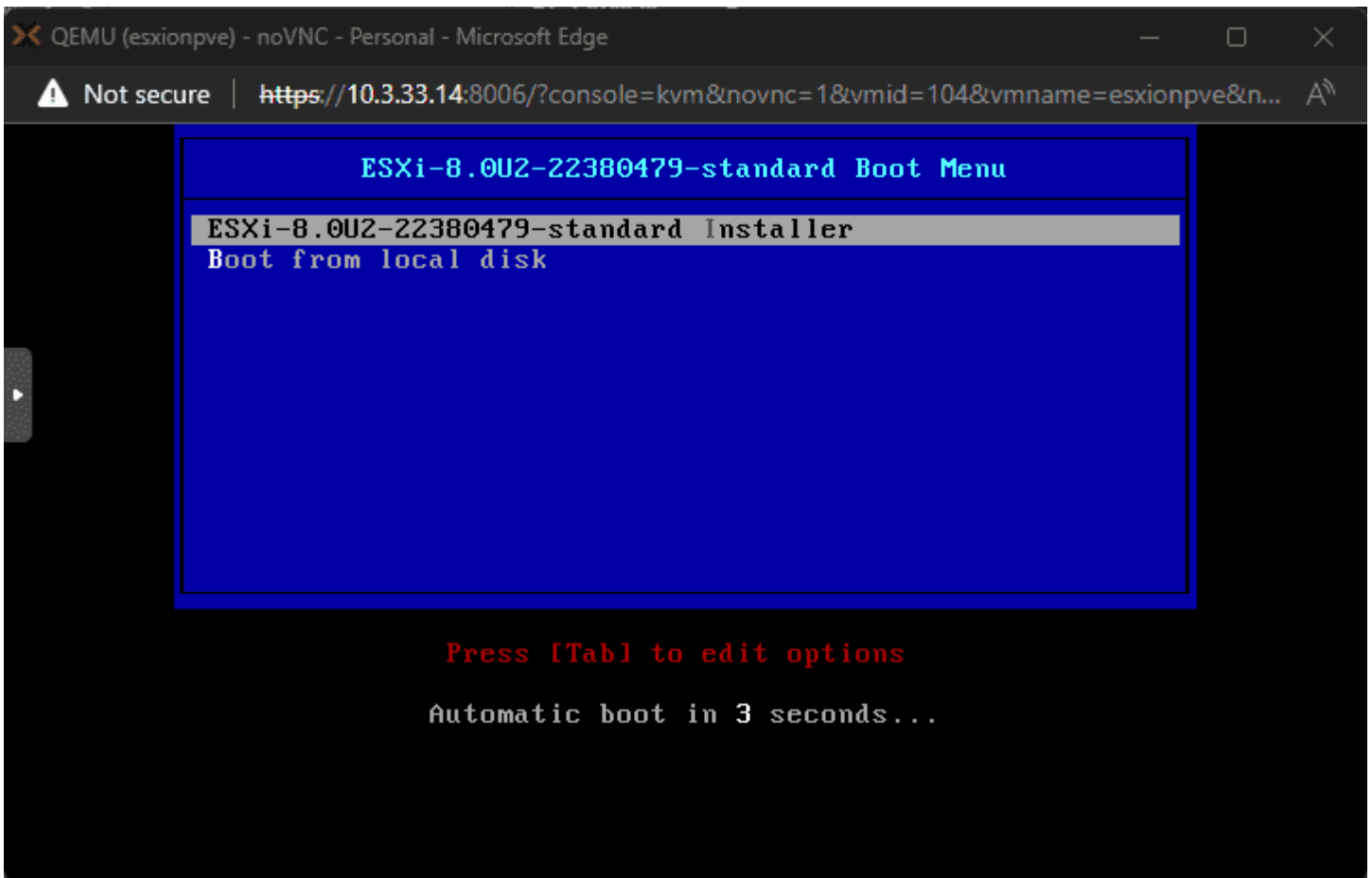
Advanced **Back** **Finish**

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

VMware ESXi 8.0.2 (VMKernel Release Build 22380479)

QEMU Standard PC (i440FX + PIIX, 1996)

Intel(R) Xeon(R) CPU D-1541 @ 2.10GHz
16 GiB Memory

Uncompressing boot modules...

vmx.v00
vin.v00
sb.v00
s.v00

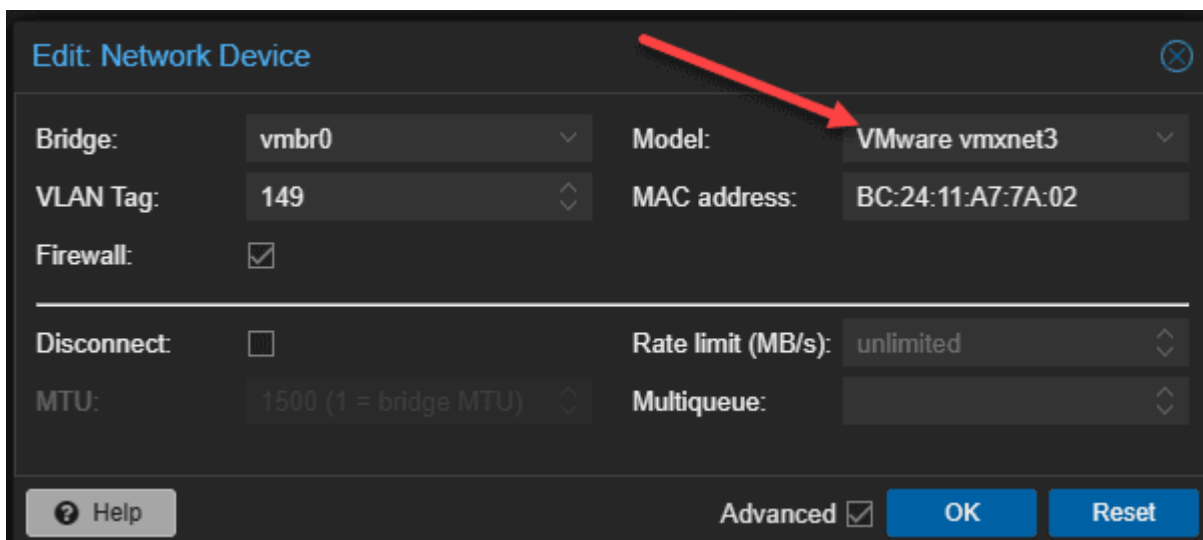
The esxi nested vm boots into the installation

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.

Welcome to the VMware ESXi 8.0.2 Installation

VMware ESXi 8.0.2 installs on most systems but only systems on VMware's Compatibility Guide are supported.

Consult the VMware Compatibility Guide at:
<http://www.vmware.com/resources/compatibility>

Select the operation to perform.

(Esc) Cancel (Enter) Continue

The installation of nested esxi continues

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

End User License Agreement (EULA)

VMWARE GENERAL TERMS

Last updated: 16 June 2022

By downloading or using an Offering, Customer agrees to be bound by the terms of the Agreement.

1. OFFERINGS.

1.1. Applicable Terms. The terms of the Order and these General Terms, including applicable Exhibits and Offering-specific Notes (collectively, the "Agreement") govern Customer's use of the Offerings. The following descending order of precedence applies: (a) the Order; (b) the General Terms; (c) the Exhibits; and (d) the Offering-specific Notes.

1.2. Users. Customer is responsible for its Users' compliance with the Agreement.

1.3. Restrictions. Customer may use the Offerings only for

Use the arrow keys to scroll the EULA text

(ESC) Do not Accept

(F11) Accept and Continue

Accept the eula 1

Select the target storage for the installation.

Select a Disk to Install or Upgrade
(any existing VMFS-3 will be automatically upgraded to VMFS-5)

* Contains a VMFS partition
Claimed by VMware vSAN

Storage Device	Capacity
Local:	
ATA QEMU HARDDISK (t10.ATA____QEMU_HARDDISK____...)	256.00 GiB
Remote:	
(none)	

(Esc) Cancel (F1) Details (F5) Refresh (Enter) Continue

Select the installation target storage for nested esxi

Select the location for the keyboard layout.

Please select a keyboard layout

- Swiss French
- Swiss German
- Turkish
- US Default
- US Dvorak
- Ukrainian
- United Kingdom

Use the arrow keys to scroll.

(Esc) Cancel (F9) Back (Enter) Continue

Select the esxi keyboard layout

Enter and confirm your password.

Enter a root password

Root password: *****
Confirm password: *****_

Passwords match.

(Esc) Cancel (F9) Back (Enter) Continue

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](#).

Error(s)/Warning(s) Found During System Scan

The system encountered the following warning(s).

Warning(s)

<CPU_SUPPORT WARNING: The CPU on this host may not be supported in future ESXi releases. Please plan accordingly. Please refer to KB 82794 for more details.>

<BIOS_FIRMWARE_TYPE WARNING: Legacy boot detected. ESXi servers running legacy BIOS are encouraged to move to UEFI. Please refer to KB 84233 for more details.>

Use the arrow keys to scroll

(Esc) Cancel (F9) Back (Enter) Continue

Warning about older cpu support in esxi 8.0 update 2

```
Confirm Install

The installer is configured to :
- install ESXi 8.0.2 on t10.ATA_____QEMU_HARDDISK_____
  _____QM00001_____

Warning: This disk will be repartitioned.

(Esc) Cancel      (F9) Back      (F11) Install
```

Confirm the installation of esxi and repartitioning

The installation begins.



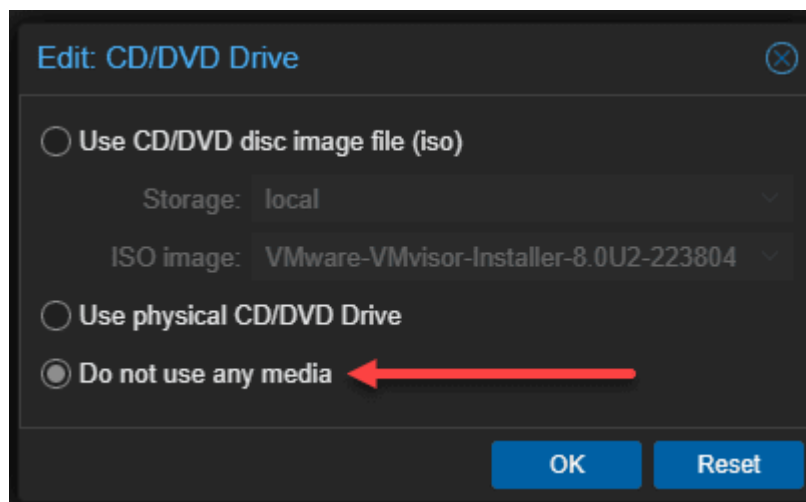
Esxi installation progress begins

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 Server

The server will shut down and reboot.

The process will take a short time to complete.


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 8.0.2 (VMKernel Release Build 22380479)

QEMU Standard PC (i440FX + PIIX, 1996)

Intel(R) Xeon(R) CPU D-1541 @ 2.10GHz
16 GiB Memory

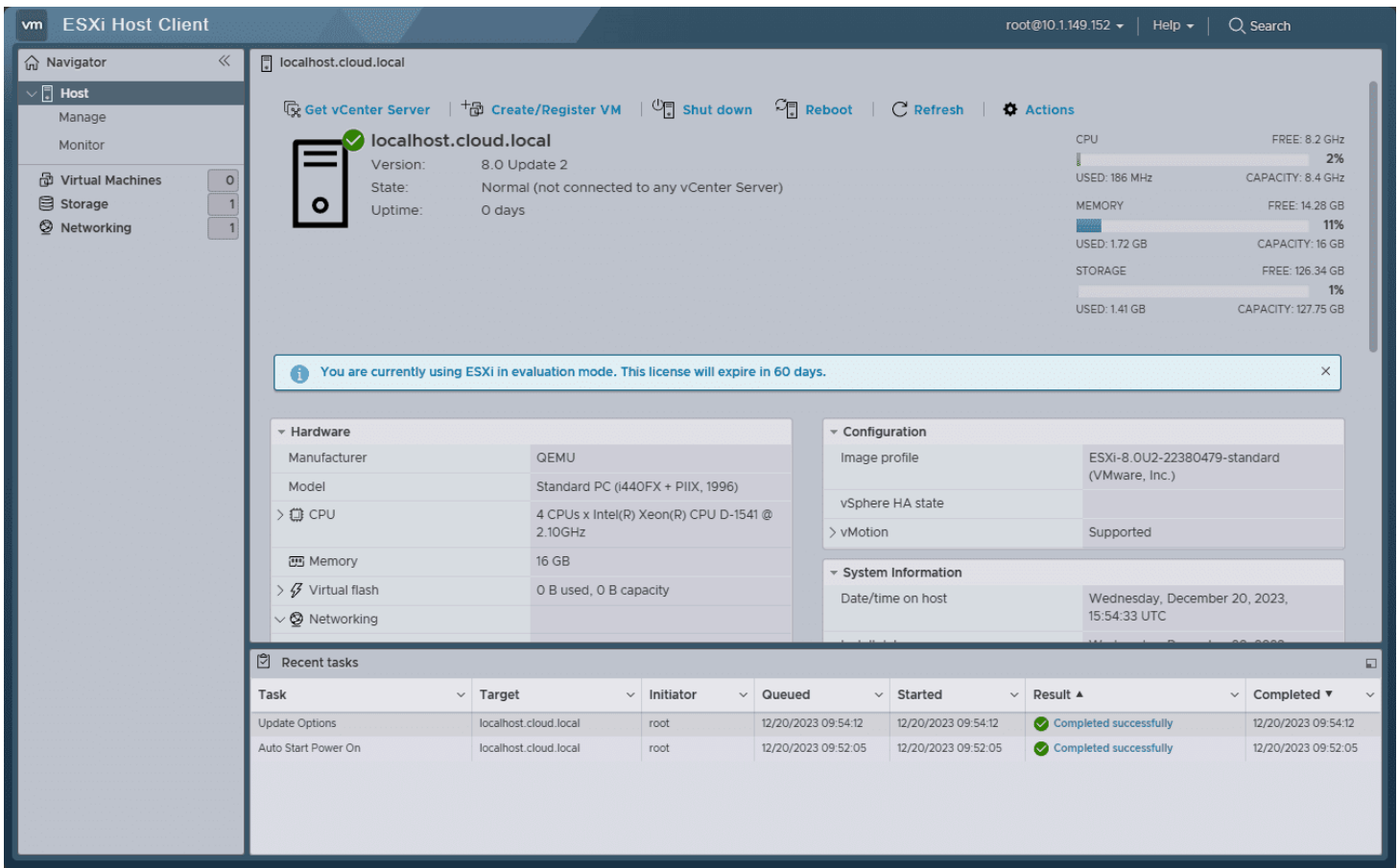
To manage this host, go to:
<https://10.1.149.152/> (DHCP) 
[https://\[fe80::be24:11ff:fea7:7a021\]/](https://[fe80::be24:11ff:fea7:7a021]/) (STATIC)

<F2> Customize System/View Logs

<F12> Shut Down/Restart

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](#)

Proxmox VE 7.1 (iso release 2) - <https://www.proxmox.com/>



Welcome to Proxmox Virtual Environment

Install Proxmox VE

Install Proxmox VE (Debug mode)

Rescue Boot

Test memory (Legacy BIOS)

Booting the Proxmox 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)**

Edit Settings | Proxmox



Virtual Hardware | **VM Options**

General Options	
VM Name	Proxmox
VM Config File	[ESX3DS01] Proxmox/Proxmox.vmx
VM Working Location	[ESX3DS01] Proxmox/
Guest OS Family	Linux
Guest OS Version	Debian GNU/Linux 11 (64-bit)
VMware Remote Console Options	<input type="checkbox"/> Lock the guest operating system when the last remote user disconnects
> Encryption	Expand for encryption settings
> Power management	Expand for power management settings
> VMware Tools	Expand for VMware Tools settings
> Boot Options	Expand for boot options
> Advanced	Expand for advanced settings
> Fibre Channel NPIV	Expand for Fibre Channel NPIV settings

CANCEL OK

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.

Edit Settings | Proxmox



Virtual Hardware | VM Options

ADD NEW DEVICE ▾

CPU		4 ▾	(i)
Cores per Socket	1 ▾	Sockets: 4	
CPU Hot Plug	<input type="checkbox"/> Enable CPU Hot Add		
Reservation	0 ▾	MHz ▾	
Limit	Unlimited ▾	MHz ▾	
Shares	Normal ▾	4000 ▾	
Hardware virtualization	<input checked="" type="checkbox"/> <u>Expose hardware assisted virtualization to the guest OS</u>		
Performance Counters	<input type="checkbox"/> Enable virtualized CPU performance counters		
I/O MMU	<input type="checkbox"/> Enabled		
> Memory	8 ▾	GB ▾	
> Hard disk 1	40 ▾	GB ▾	
> SCSI controller 0	LSI Logic Parallel		
> Network adapter 1	DPG-Servers ▾	<input checked="" type="checkbox"/> Connect...	
> CD/DVD drive 1	Datastore ISO File ▾	<input checked="" type="checkbox"/> Connect...	
> Video card	Specify custom settings ▾		

CANCEL OK

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**.



Welcome to Proxmox Virtual Environment

Install Proxmox VE
Install Proxmox VE (Debug mode)
Rescue Boot
Test memory (Legacy BIOS)

Booting the Proxmox 7.1 VE installer

First things first. Accept the EULA to proceed.



END USER LICENSE AGREEMENT (EULA)

END USER LICENSE AGREEMENT (EULA) FOR PROXMOX VIRTUAL ENVIRONMENT (PROXMOX VE)

By using Proxmox VE software you agree that you accept this EULA, and that you have read and understand the terms and conditions. This also applies for individuals acting on behalf of entities. This EULA does not provide any rights to Support Subscriptions Services as software maintenance, updates and support. Please review the Support Subscriptions Agreements for these terms and conditions. The EULA applies to any version of Proxmox VE and any related update, source code and structure (the Program), regardless of the the delivery mechanism.

1. License. Proxmox Server Solutions GmbH (Proxmox) grants to you a perpetual, worldwide license to the Programs pursuant to the GNU Affero General Public License V3. The license agreement for each component is located in the software component's source code and permits you to run, copy, modify, and redistribute the software component (certain obligations in some cases), both in source code and binary code forms, with the exception of certain binary only firmware components and the Proxmox images (e.g. Proxmox logo). The license rights for the binary only firmware components are located within the components. This EULA pertains solely to the Programs and does not limit your rights under, or grant you rights that supersede, the license terms of any particular component.

2. Limited Warranty. The Programs and the components are provided and licensed "as is" without warranty of any kind, expressed or implied, including the implied warranties of merchantability, non-infringement or fitness for a particular purpose. Neither Proxmox nor its affiliates warrants that the functions contained in the Programs will meet your requirements or that the operation of the Programs will be entirely error free, appear or perform precisely as described in the accompanying documentation, or comply with regulatory requirements.

3. Limitation of Liability. To the maximum extent permitted under applicable law, under no

Abort

Previous

I agree

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.

Proxmox Virtual Environment (PVE)

The Proxmox Installer automatically partitions your hard disk. It installs all required packages and makes the system bootable from the hard disk. All existing partitions and data will be lost.

Press the Next button to continue the installation.

- **Please verify the installation target**
The displayed hard disk will be used for the installation.
Warning: All existing partitions and data will be lost.
- **Automatic hardware detection**
The installer automatically configures your hardware.
- **Graphical user interface**
Final configuration will be done on the graphical user interface, via a web browser.

Target Harddisk: /dev/sda (8GiB, Virtual disk) ▾ Options

Abort Previous Next

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

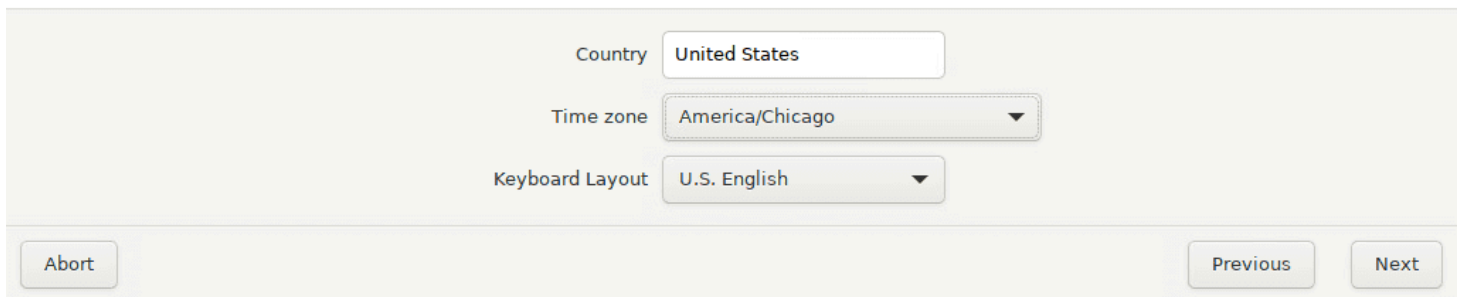
Next up is setting your location and time zone configuration.

Location and Time Zone selection

The Proxmox Installer automatically makes location-based optimizations, like choosing the nearest mirror to download files from. Also make sure to select the correct time zone and keyboard layout.

Press the Next button to continue the installation.

- **Country:** The selected country is used to choose nearby mirror servers. This will speed up downloads and make updates more reliable.
- **Time Zone:** Automatically adjust daylight saving time.
- **Keyboard Layout:** Choose your keyboard layout.



The screenshot shows the Proxmox VE installer's location and time zone selection screen. It features three input fields: 'Country' with 'United States', 'Time zone' with 'America/Chicago', and 'Keyboard Layout' with 'U.S. English'. At the bottom, there are three buttons: 'Abort', 'Previous', and 'Next'.

Set the location and time zone

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

Administration Password and Email Address

Proxmox Virtual Environment is a full featured, highly secure GNU/Linux system, based on Debian.

In this step, please provide the *root* password.

- **Password:** Please use a strong password. It should be at least 8 characters long, and contain a combination of letters, numbers, and symbols.
- **Email:** Enter a valid email address. Your Proxmox VE server will send important alert notifications to this email account (such as backup failures, high availability events, etc.).

Press the Next button to continue the installation.

Form fields for setting administrator password and email address:

- Password: [Masked]
- Confirm: [Masked]
- Email:

Buttons: Abort, Previous, Next

Set the administrator password and email address

Configure the Proxmox hostname and your network configuration.

Management Network Configuration

Please verify the displayed network configuration. You will need a valid network configuration to access the management interface after installing.

After you have finished, press the Next button. You will be shown a list of the options that you chose during the previous steps.

- **IP address (CIDR):** Set the main IP address and netmask for your server in CIDR notation.
- **Gateway:** IP address of your gateway or firewall.
- **DNS Server:** IP address of your DNS server.

Management Interface:

Hostname (FQDN):

IP Address (CIDR): /

Gateway:

DNS Server:

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

Please confirm the displayed information. Once you press the **Install** button, the installer will begin to partition your drive(s) and extract the required files.

Option	Value
Filesystem:	ext4
Disk(s):	/dev/sda
Country:	United States
Timezone:	America/Chicago
Keymap:	en-us
Email:	admin@cloud.local
Management Interface:	ens32
Hostname:	proxmox
IP CIDR:	10.1.149.74/24
Gateway:	10.1.149.1
DNS:	10.1.149.10

Automatically reboot after successful installation

Abort

Previous

Install

Summary of the Proxmox VE 7.1 installation

The installation process begins.



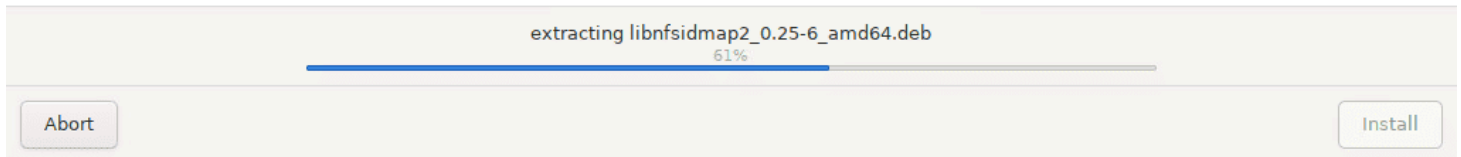
Virtualize your IT Infrastructure

Proxmox VE is ready for enterprise deployments.

The role based permission management combined with the integration of multiple external authentication sources is the base for a secure and stable environment.

Visit www.proxmox.com for more information about commercial support subscriptions.

- **Commitment to Free Software**
The source code is released under the GNU Affero General Public License.
- **RESTful web API**
Resource-oriented architecture (ROA) and declarative API definition using JSON Schema enable easy integration for third party management tools.
- **Virtual Appliances**
Pre-installed applications - up and running within a few seconds.



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.

GNU GRUB version 2.04-20

```
*Proxmox VE GNU/Linux
Advanced options for Proxmox VE GNU/Linux
Memory test (memtest86+)
Memory test (memtest86+, serial console 115200)
Memory test (memtest86+, experimental multiboot)
Memory test (memtest86+, serial console 115200, experimental multiboot)
```

Use the ↑ and ↓ keys to select which entry is highlighted.
Press enter to boot the selected OS, `e` to edit the commands
before booting or `c` for a command-line.
The highlighted entry will be executed automatically in 2s.

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.

The screenshot displays the Proxmox VE 7.1 web interface. The top navigation bar includes the Proxmox logo, version information, a search bar, and buttons for 'Documentation', 'Create VM', 'Create CT', and the user 'root@pam'. The left sidebar shows a 'Server View' with a tree structure including 'Datacenter', 'proxmox', 'local (proxmox)', and 'local-lvm (proxmox)'. The main content area is titled 'Node proxmox' and features a 'Summary' tab. The summary shows system metrics: CPU usage at 0.15% of 4 CPU(s), Load average at 0.09, 0.08, 0.03, RAM usage at 12.47% (991.32 MiB of 7.77 GiB), and / HD space at 26.02% (2.47 GiB of 9.50 GiB). It also lists hardware (4 x Intel(R) Xeon(R) CPU D-1541 @ 2.10GHz) and software (Linux 5.13.19-2-pve #1 SMP PVE 5.13.19-4). A 'CPU usage' graph is visible below the summary. At the bottom, a 'Tasks' table shows a recent task: 'Start all VMs and Containers' performed by 'root@pam' on 'Jan 09 21:16:08' with a status of 'OK'.

Start Time ↓	End Time	Node	User name	Description	Status
Jan 09 21:16:08	Jan 09 21:16:08	proxmox	root@pam	Start all VMs and Containers	OK

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?](#)
- [Why do it?](#)
- [Checking Processor Compatibility and Current Settings](#)
- [Enabling Nested Virtualization on Proxmox Host](#)
- [What happens nested virtualization isn't enabled?](#)
- [Configuring Nested Virtualization in VMs](#)

- [Validating the Configuration](#)
- [FAQs on Proxmox Nested Virtualization](#)

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.

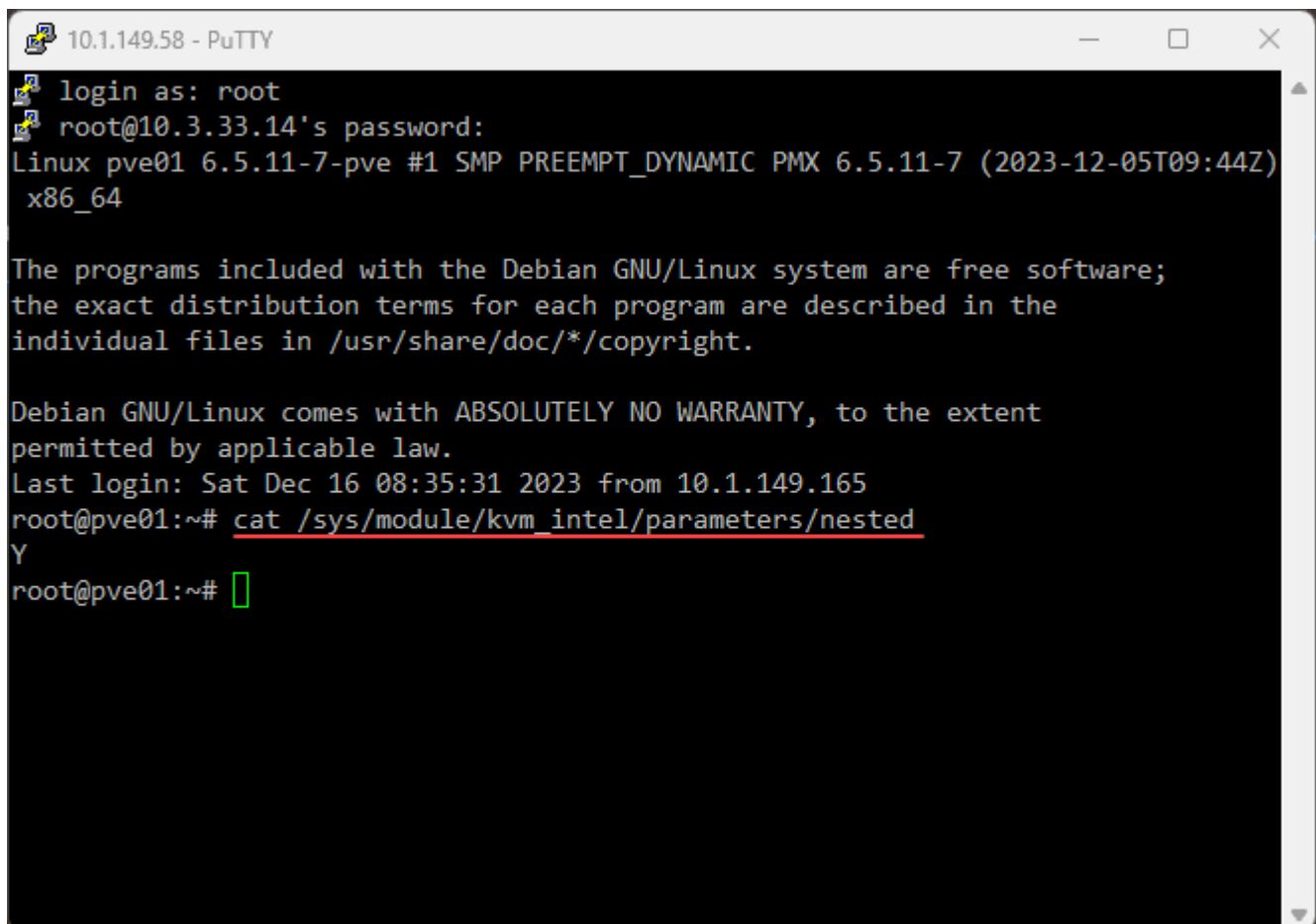
By configuring nested virtualization, you can treat the VM hypervisor like any other VM. It means you can use things like snapshots to capture the state of the VM. Then, you can roll it back like any other VM in your environment.

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
```



```
10.1.149.58 - PuTTY
login as: root
root@10.3.33.14's password:
Linux pve01 6.5.11-7-pve #1 SMP PREEMPT_DYNAMIC PMX 6.5.11-7 (2023-12-05T09:44Z)
x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

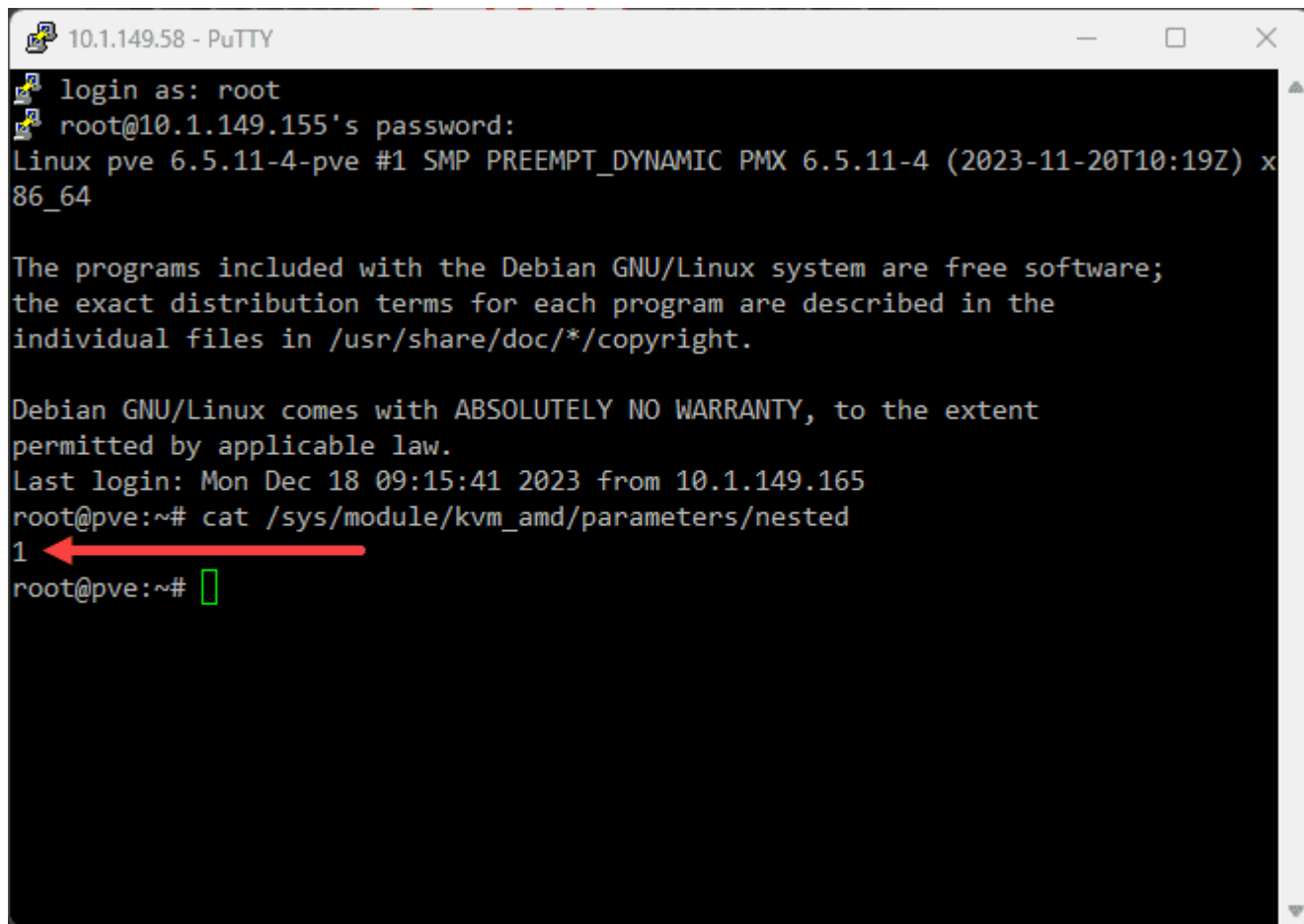
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Sat Dec 16 08:35:31 2023 from 10.1.149.165
root@pve01:~# cat /sys/module/kvm_intel/parameters/nested
Y
root@pve01:~# █
```

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).



```
10.1.149.58 - PuTTY
login as: root
root@10.1.149.155's password:
Linux pve 6.5.11-4-pve #1 SMP PREEMPT_DYNAMIC PMX 6.5.11-4 (2023-11-20T10:19Z) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Mon Dec 18 09:15:41 2023 from 10.1.149.165
root@pve:~# cat /sys/module/kvm_amd/parameters/nested
1
root@pve:~#
```

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:

```
10.1.149.58 - PuTTY
login as: root
root@10.3.33.14's password:
Linux pve01 6.5.11-7-pve #1 SMP PREEMPT_DYNAMIC PMX 6.5.11-7 (2023-12-05T09:44Z)
x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Mon Dec 18 09:13:58 2023 from 10.3.33.221
root@pve01:~# lscpu | egrep --color -i "Vendor ID|Model name"
Vendor ID:           GenuineIntel
BIOS Vendor ID:     Intel
root@pve01:~#
```

Checking processor type when intel

AMD CPUs will look like the following:

```
10.1.149.58 - PuTTY
login as: root
root@10.1.149.155's password:
Linux pve 6.5.11-4-pve #1 SMP PREEMPT_DYNAMIC PMX 6.5.11-4 (2023-11-20T10:19Z) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Mon Dec 18 09:15:41 2023 from 10.1.149.165
root@pve:~# cat /sys/module/kvm_amd/parameters/nested
1
root@pve:~# lscpu | egrep --color -i "Vendor ID|Model name"
Vendor ID: AuthenticAMD
BIOS Vendor ID: AuthenticAMD
root@pve:~#
```

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:

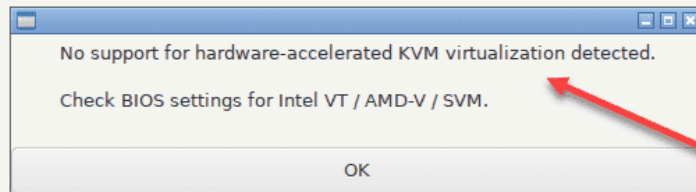
```
sudo modprobe -r kvm_amd sudo modprobe kvm_amd nested=1 echo "options kvm-amd nested=1" > /etc/modprobe.d/kvm-amd.conf
```

```
## 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.

Create: Virtual Machine ⌵

General OS System Disks CPU Memory Network Confirm

Node: pve01 ⌵ Resource Pool: ⌵

VM ID: 103 ⬆

Name: pmoxnested01 ⓘ

Start at boot:

Start/Shutdown order: any

Startup delay: default

Shutdown timeout: default

Tags

No Tags +

? Help Advanced Back Next

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
```



```
10.1.149.58 - PuTTY
root@pve01:~# qm set 103 --cpu host
update VM 103: -cpu host
root@pve01:~#
```

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
```

```
10.1.149.58 - PuTTY
root@10.1.149.156's password:
Linux pvenestedtest 6.5.11-4-pve #1 SMP PREEMPT_DYNAMIC PMX 6.5.11-4 (2023-11-20
T10:19Z) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Mon Dec 18 09:50:47 2023
root@pvenestedtest:~# egrep --color -i "svm|vmx" /proc/cpuinfo
flags          : fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov
pat pse36 clflush mmx fxsr sse sse2 ss syscall nx pdpe1gb rdtscp lm constant_tsc
arch_perfmon rep_good nopl xtopology cpuid tsc_known_freq pni pclmulqdq vmx sss
e3 fma cx16 pdcm pcid sse4_1 sse4_2 x2apic movbe popcnt tsc_deadline_timer aes x
save avx f16c rdrand hypervisor lahf_lm abm 3dnowprefetch cpuid_fault invpcid_si
ngle pti ssbd ibrs ibpb stibp tpr_shadow flexpriority ept vpid ept_ad fsgsbase t
sc_adjust bmi1 hle avx2 smep bmi2 erms invpcid rtm rdseed adx smap xsaveopt arat
vnmi umip md_clear flush_l1d arch_capabilities
vmx flags      : vnmi preemption_timer posted_intr invvpid ept_x_only ept_ad ep
t_l1gb flexpriority apicv tsc_offset vtpr mtf vapic ept vpid unrestricted_guest v
apic_reg vid shadow_vmcs pml
root@pvenestedtest:~# █
```

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?

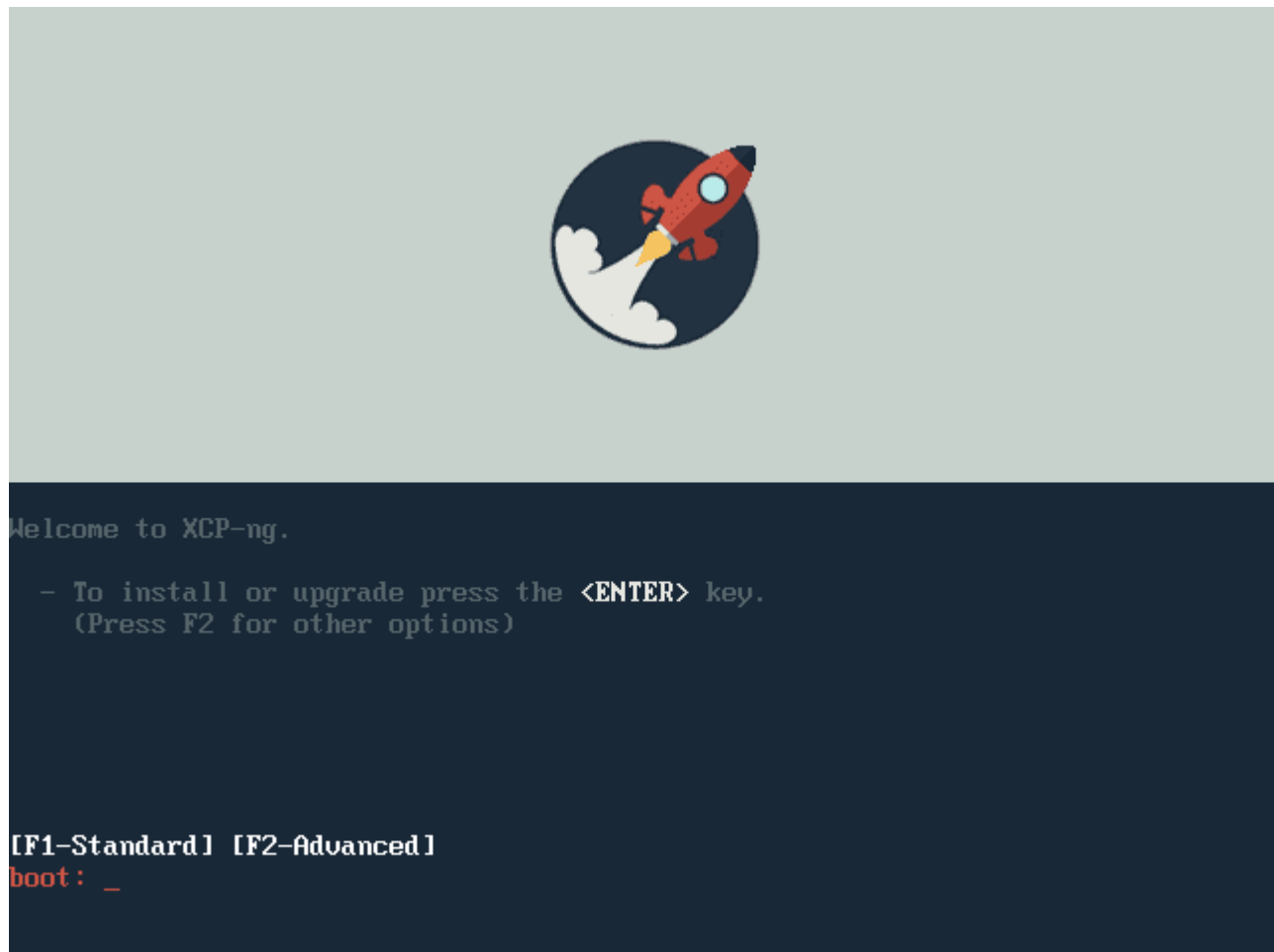
It can be. When you nest hypervisors as VMs, it creates another layer to understand with your network configuration.

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:

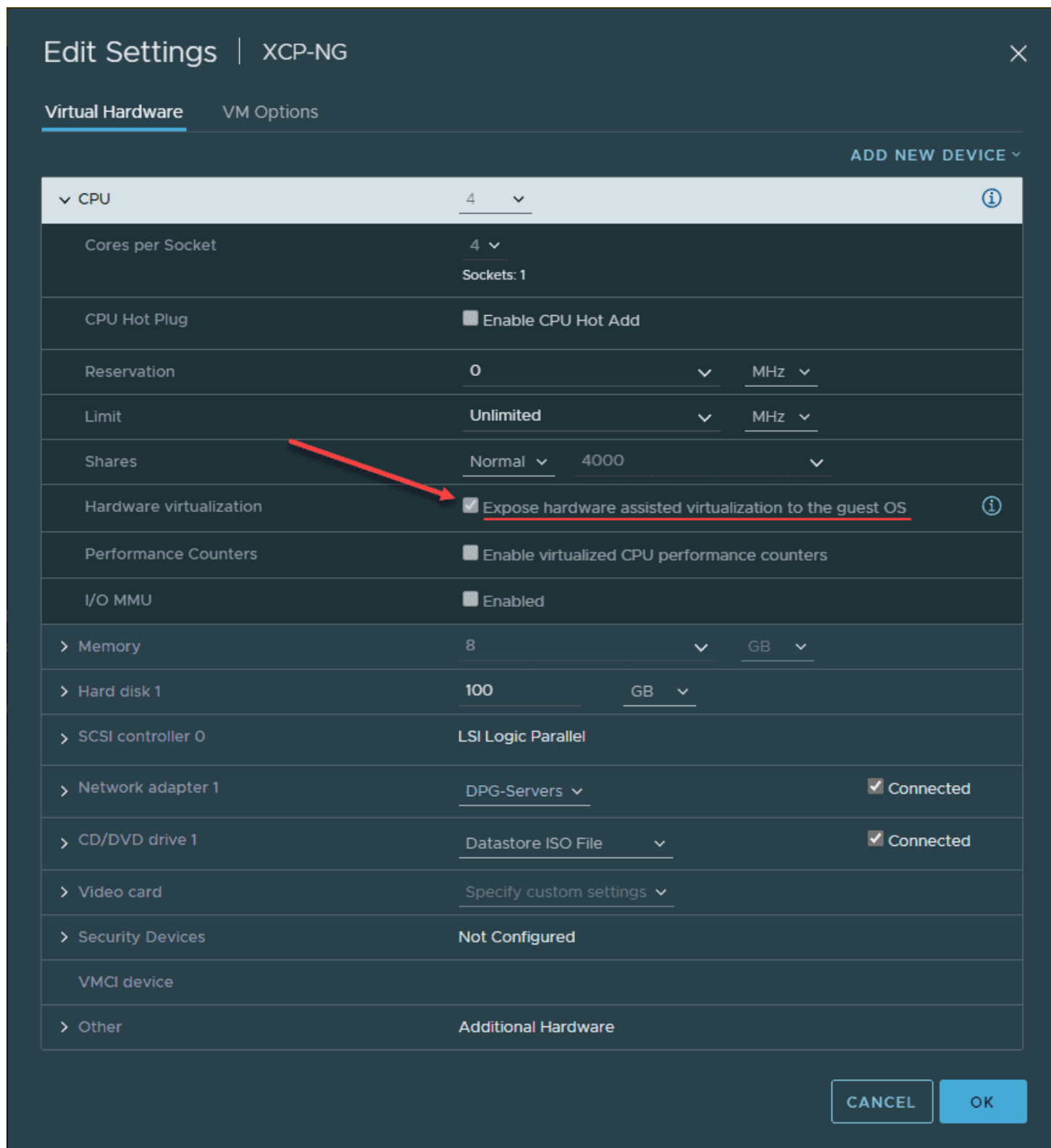
- **Modern UI** – It offers a modern UI for management called [Xen Orchestra](#). There is also a Windows-based management tool called XCP-NG center that allows managing your XCP-NG hosts, etc.
- **Live migration** – This is probably what drew most of us to virtualization and got us hooked to begin with – [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.



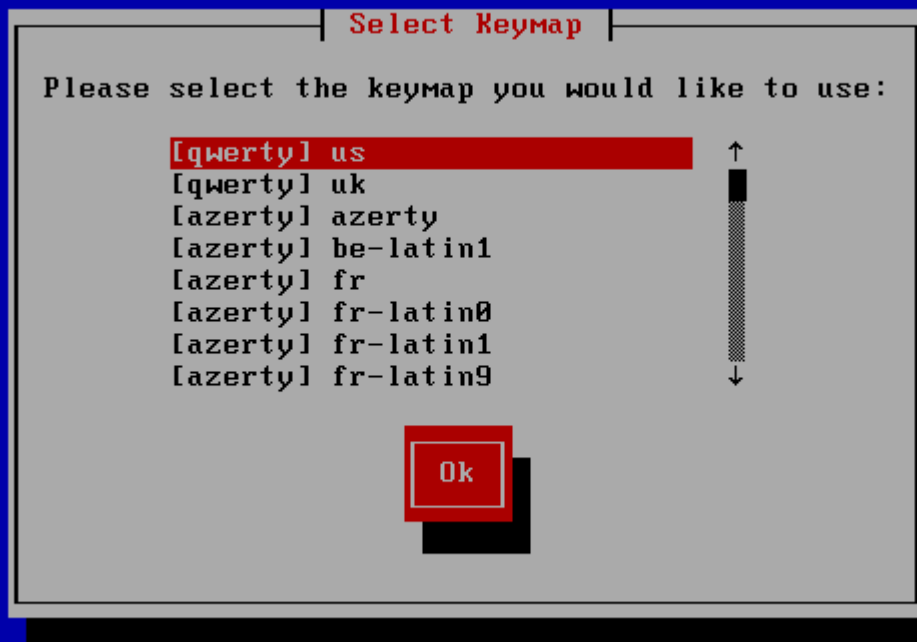
The screenshot shows the 'Edit Settings' window for an XCP-NG virtual machine. The 'Virtual Hardware' tab is selected. The 'CPU' section is expanded, showing various settings. A red arrow points to the 'Expose hardware assisted virtualization to the guest OS' checkbox, which is checked. The 'Hardware virtualization' section also includes 'Performance Counters' (unchecked) and 'I/O MMU' (checked). Other hardware sections like Memory, Hard disk, SCSI controller, Network adapter, CD/DVD drive, Video card, Security Devices, and VMCI device are also visible.

Setting	Value
CPU	4
Cores per Socket	4
Sockets	1
CPU Hot Plug	Enable CPU Hot Add
Reservation	0 MHz
Limit	Unlimited MHz
Shares	Normal 4000
Hardware virtualization	Expose hardware assisted virtualization to the guest OS (checked)
Performance Counters	Enable virtualized CPU performance counters (unchecked)
I/O MMU	Enabled (checked)
Memory	8 GB
Hard disk 1	100 GB
SCSI controller 0	LSI Logic Parallel
Network adapter 1	DPG-Servers (checked Connected)
CD/DVD drive 1	Datastore ISO File (checked Connected)
Video card	Specify custom settings
Security Devices	Not Configured
VMCI device	
Other	Additional Hardware

Enable nested virtualization for your XCP NG virtual machine

Select the locality for the keyboard layout.

```
Welcome to XCP-ng - Version 8.2.1 (Kernel 4.19.0+1)
Copyright (c) 2018-2022 Open Source
```

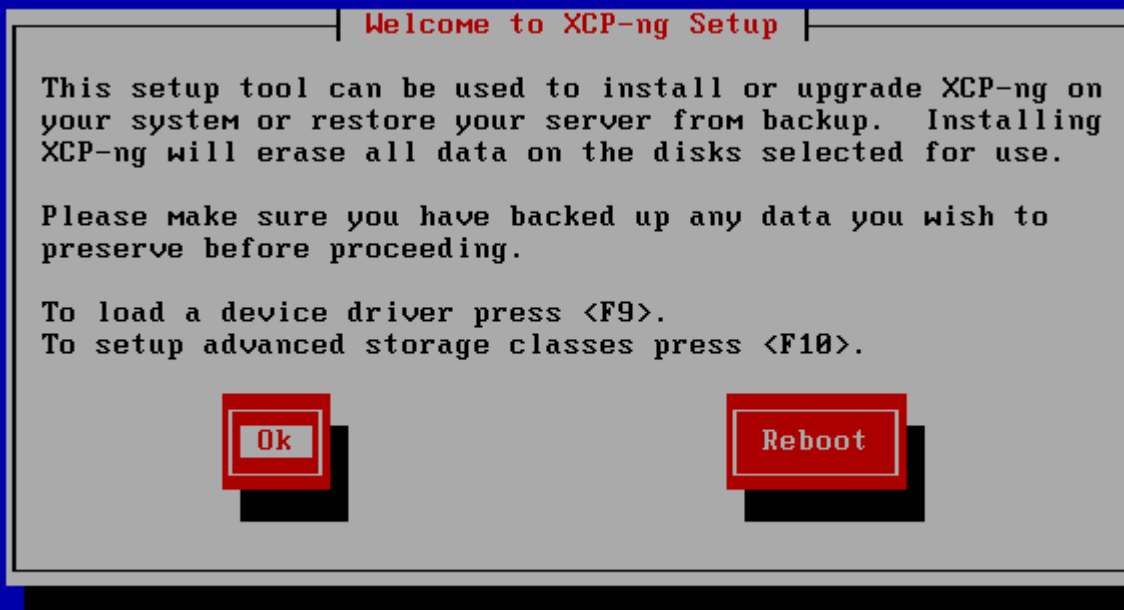


<Tab>/<Alt-Tab> between elements ; ; <F1> Help screen

Choose the country for keyboard mapping

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

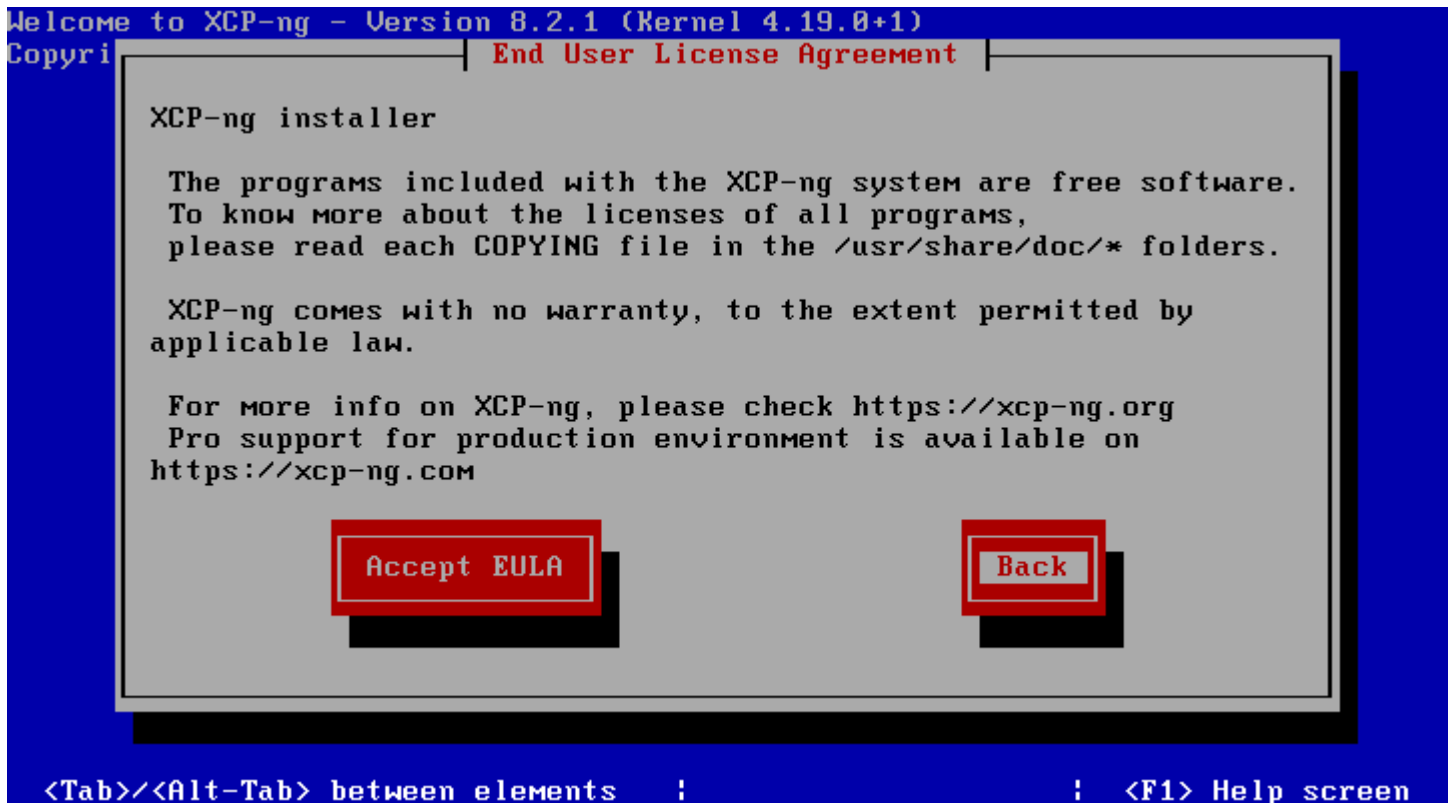
```
Welcome to XCP-ng - Version 8.2.1 (Kernel 4.19.0+1)
Copyright (c) 2018-2022 Open Source
```



<Tab>/<Alt-Tab> between elements ; <F9> load driver ; <F1> Help screen

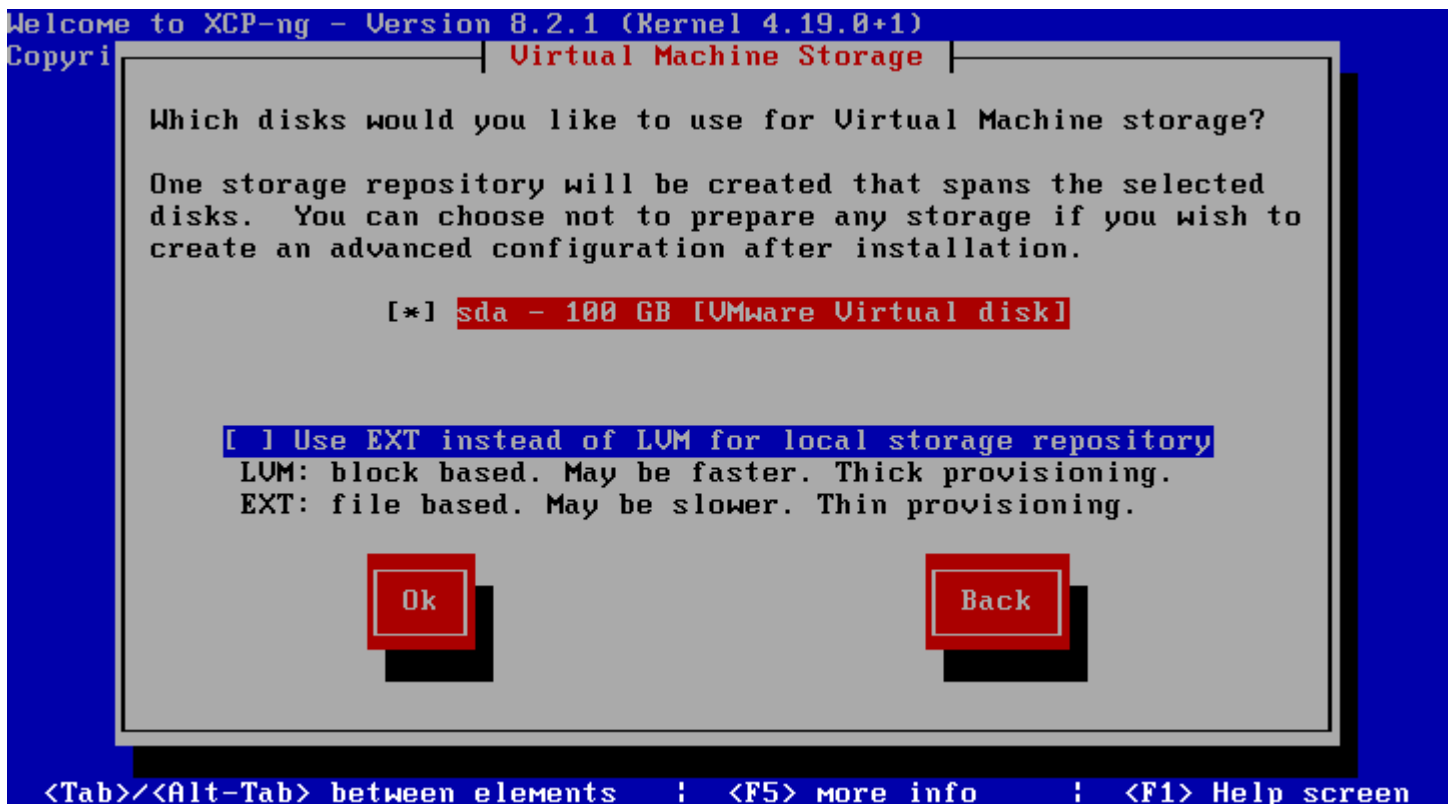
Beginning the setup wizard

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.

Welcome to XCP-ng - Version 8.2.1 (Kernel 4.19.0+1)
Copyright (c) 2018-2022 Open Source

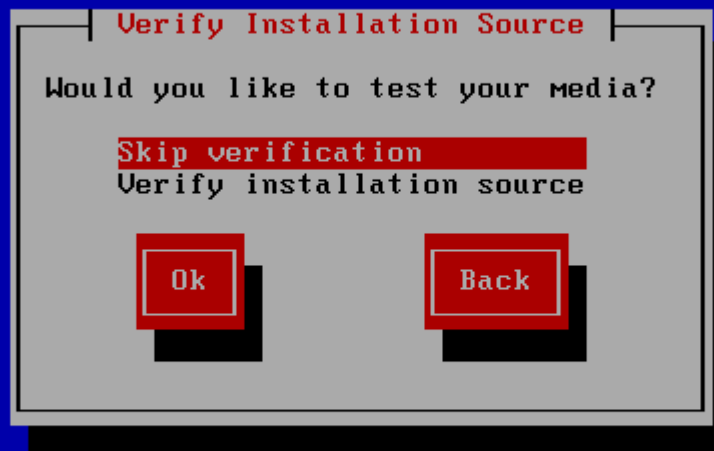


<Tab>/<Alt-Tab> between elements ; : <F1> Help screen

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.

Welcome to XCP-ng - Version 8.2.1 (Kernel 4.19.0+1)
Copyright (c) 2018-2022 Open Source



<Tab>/<Alt-Tab> between elements ; : <F1> Help screen

Choose media verification option

Set an administrator password for the root account.

Welcome to XCP-ng - Version 8.2.1 (Kernel 4.19.0+1)
Copyright (c) 2018-2022 Open Source

Set Password

Please specify a password of at least 6 characters for the root account.
(This is the password used when connecting to the XCP-ng Host from XCP-ng Center.)

Password *****
Confirm *****

Ok **Back**

<Tab>/<Alt-Tab> between elements ; ; <F1> Help screen

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.

Welcome to XCP-ng - Version 8.2.1 (Kernel 4.19.0+1)
Copyright (c)

Networking

Please specify how networking should be configured for the management interface on this host.

Automatic configuration (DHCP)
 Static configuration:
 IP Address: _____
 Subnet mask: _____
 Gateway: _____

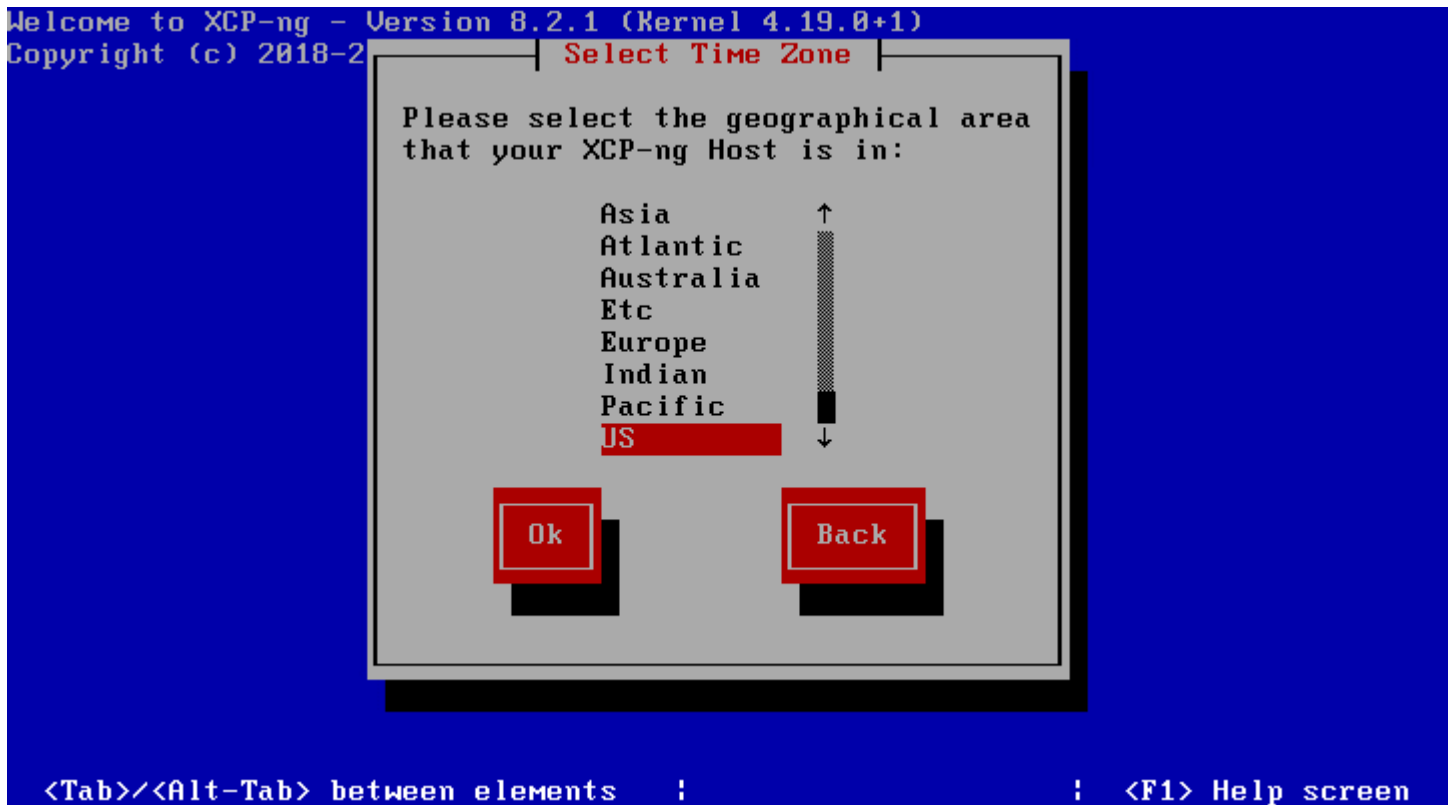
Use VLAN:
 VLAN (1-4094): _____

Ok **Back**

<Tab>/<Alt-Tab> between elements ; ; <F1> Help screen

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.

Welcome to XCP-ng - Version 8.2.1 (Kernel 4.19.0+1)
Copyright (c) 2018-2022 Open Source

System Time

How should the local time be determined?

Using NTP
 Manual time entry

<Tab>/<Alt-Tab> between elements ; ; <F1> Help screen

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

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

Welcome to XCP-ng - Version 8.2.1 (Kernel 4.19.0+1)
Copyright (c) 2018-2022 Open Source

NTP Configuration

Please specify details of the NTP servers you wish to use (e.g. pool.ntp.org)?

NTP is configured by my DHCP server

NTP Server 1: pool.ntp.org_____

NTP Server 2: _____

NTP Server 3: _____

<Tab>/<Alt-Tab> between elements ; ; <F1> Help screen

Configure your NTP servers

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

Welcome to XCP-ng - Version 8.2.1 (Kernel 4.19.0+1)
Copyright (c) 2018-2022 Open Source

Confirm Installation

We have collected all the information required to install XCP-ng.

Please confirm you wish to proceed: all data on disk sda will be destroyed!

Install XCP-ng **Back**

<Tab>/<Alt-Tab> between elements ; ; <F1> Help screen

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

The installation of XCP-NG begins.

Welcome to XCP-ng - Version 8.2.1 (Kernel 4.19.0+1)
Copyright (c) 2018-2022 Open Source

Installing XCP-ng

Installing XCP-ng...

14%

Working: Please wait...

Installation of XCP NG begins

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

Welcome to XCP-ng - Version 8.2.1 (Kernel 4.19.0+1)
Copyright (c) 2018-2022 Open Source

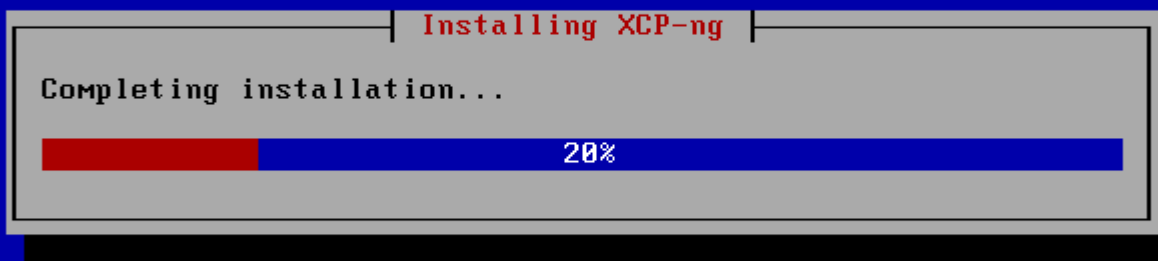


<Tab>/<Alt-Tab> between elements ; ; <F1> Help screen

Answer question about installing supplemental packs

The installation completes and finalizes the settings.

Welcome to XCP-ng - Version 8.2.1 (Kernel 4.19.0+1)
Copyright (c) 2018-2022 Open Source



Working: Please wait...

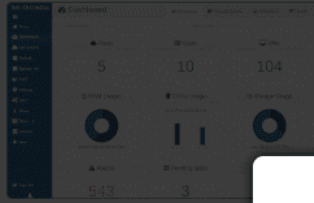
XCP NG installation begins to finalize

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.

Management Tools

You have various options to manage your XCP-ng - choose the one that fits best!

Xen Orchestra



xe CLI




Xen Orchestra Quick Deploy

This will download and deploy a new virtual machine on your host, called "XOA" (Xen Orchestra virtual Appliance).

Host username

Host password

 This host's root credentials

CONNECT

[Close](#)

comes in a virtual appliance (XOA) covering cloud features you may need in your infrastructure browser.

QUICK DEPLOY

and line interface available on this host. You can find more there. To learn more about it, please read the documentation.

DOCUMENTATION

is a high performance enterprise level hypervisor

to Datacenters. We have your back when



Pro Support 24/7/365



SSH remote support



Setup assistance

GO PRO

Beginning the installation of Xen Orchestra

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

Management Tools

You have various options to manage your XCP-ng - choose the one that fits best!

Xen Orchestra



Xen CLI



Xen Orchestra Quick Deploy

This will download and deploy a new virtual machine on your host, called "XOA" (Xen Orchestra virtual Appliance).

Select a storage

Local storage - 58 GiB left

Select a network

Pool-wide network associat

XOA VM IP address

Netmask

255.255.255.0

Gateway

DNS

8.8.8.8

i Leave everything empty to use DHCP. Provide all the information otherwise.

NEXT

[Close](#)

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**.

Management Tools

Xen Orchestra Quick Deploy

This will download and deploy a new virtual machine on your host, called "XOA" (Xen Orchestra virtual Appliance).

Create an Admin account on your XOA

Username

Password

Register your XOA

[Create an account](#)


xen-orchestra.com username


xen-orchestra.com password

Set the XOA machine password

Login

Password

 You can do this later

 **DEPLOY**

[Close](#)

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.

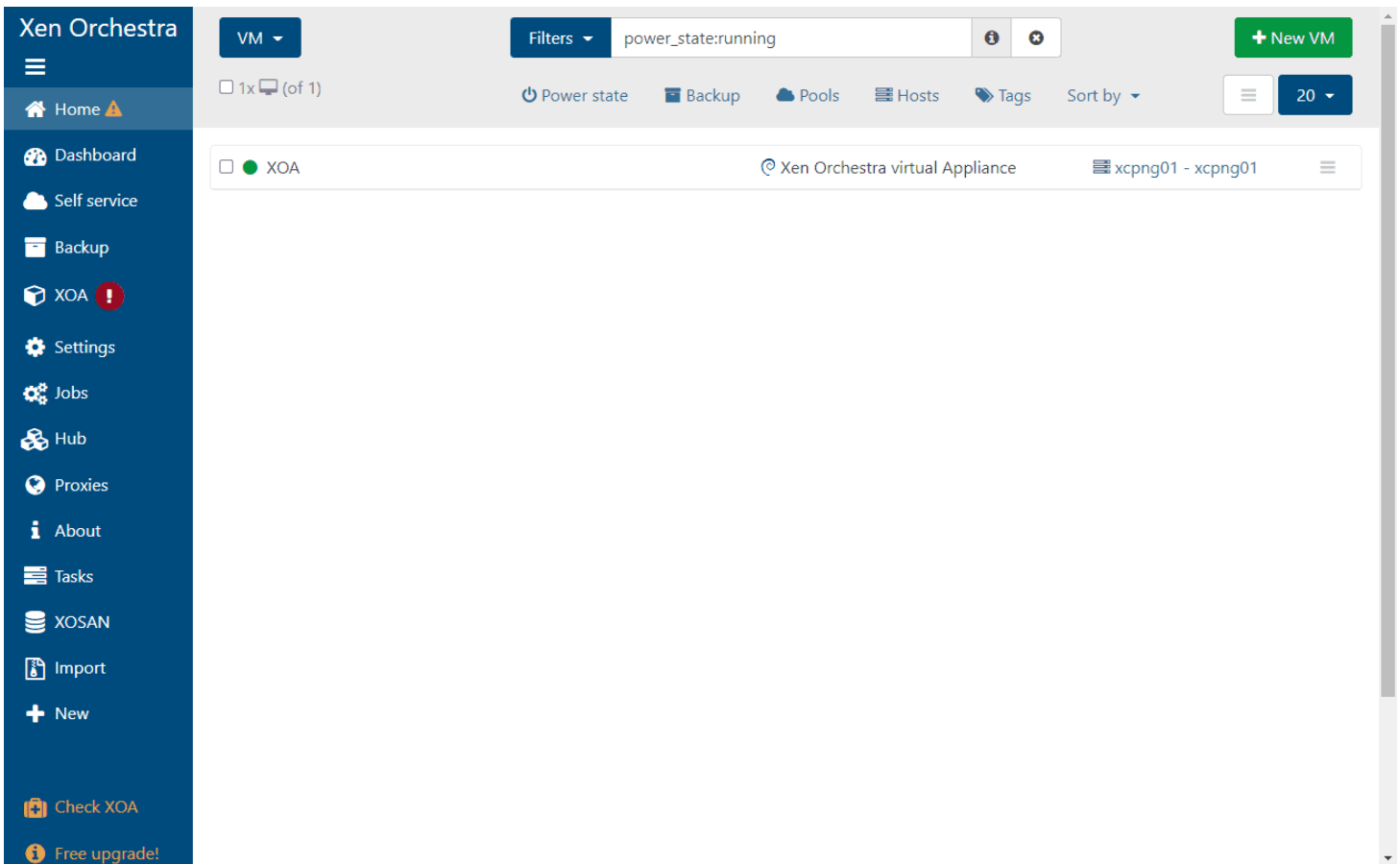


Xen Orchestra

Remember me

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](#)

The screenshot shows the Harvester web interface. The top navigation bar includes the Harvester logo, the text 'All Namespaces', and a user profile icon. The left sidebar contains a menu with items: Dashboard, Hosts, Virtual Machines (selected), Volumes, Images, Namespaces, and Advanced. The main content area displays details for a 'Virtual Machine: ubuntu2204' which is in a 'Running' state. It shows the namespace as 'default' and an age of '46 secs'. There are buttons for 'Detail', 'Config', and 'YAML'. Below this, there are labels for 'harvesterhci.io/creator: harvester' and 'harvesterhci.io/os: ubuntu', and an annotation link 'Show 5 annotations'. A left-hand sub-menu lists various categories: Basics, Volumes, Networks, SSH Keys, VM Metrics, Access Credentials, Cloud Config, Events (selected), and Migration. The 'Events' section is a table with columns for Reason, Resource, and Date. It lists several events, including 'Started', 'Created', 'Migrated', and 'SuccessfulMountVolume'.

Reason	Resource	Date
Started	VirtualMachineInstance ubuntu2204 VirtualMachineInstance started.	20 secs ago
Created	VirtualMachineInstance ubuntu2204 VirtualMachineInstance defined.	20 secs ago
Migrated	VirtualMachineInstance ubuntu2204 EvictionStrategy is set but vmi is not migratable	20 secs ago
Started	Pod virt-launcher-ubuntu2204-g4tn7 Started container compute	24 secs ago
SuccessfulMountVolume	Pod virt-launcher-ubuntu2204-g4tn7 MapVolume.MapPodDevice succeeded for volume "pvc-0633a5f1-b129-44de-8ee4-b27f3aaa691e" globalMapPath "/var/lib/kubelet/plugins/kubernetes.io/csi/volumeDevices/pvc-0633a5f1-b129-44de-8ee4-b27f3aaa691e/dev"	25 secs ago
SuccessfulMountVolume	Pod virt-launcher-ubuntu2204-g4tn7 MapVolume.MapPodDevice succeeded for volume "pvc-0633a5f1-b129-44de-8ee4-b27f3aaa691e" volumeMapPath "/var/lib/kubelet/pods/8a3f2907-6d57-4e07-83bf-fd19fb1fb6d0/volumeDevices/kubernetes.io~csi"	25 secs ago
SuccessfulMountVolume	Pod virt-launcher-ubuntu2204-g4tn7 MapVolume.MapPodDevice succeeded for volume "pvc-8eb92fd6-a76c-4249-9387-8931c087e013" globalMapPath "/var/lib/kubelet/plugins/kubernetes.io/csi/volumeDevices/pvc-8eb92fd6-a76c-4249-9387-8931c087e013/dev"	25 secs ago
SuccessfulMountVolume	Pod virt-launcher-ubuntu2204-g4tn7 MapVolume.MapPodDevice succeeded for volume "pvc-8eb92fd6-a76c-4249-9387-8931c087e013" volumeMapPath "/var/lib/kubelet/pods/8a3f2907-6d57-4e07-83bf-fd19fb1fb6d0/volumeDevices/kubernetes.io~csi"	25 secs ago

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**.

Edit Settings | Harvester



Virtual Hardware | VM Options

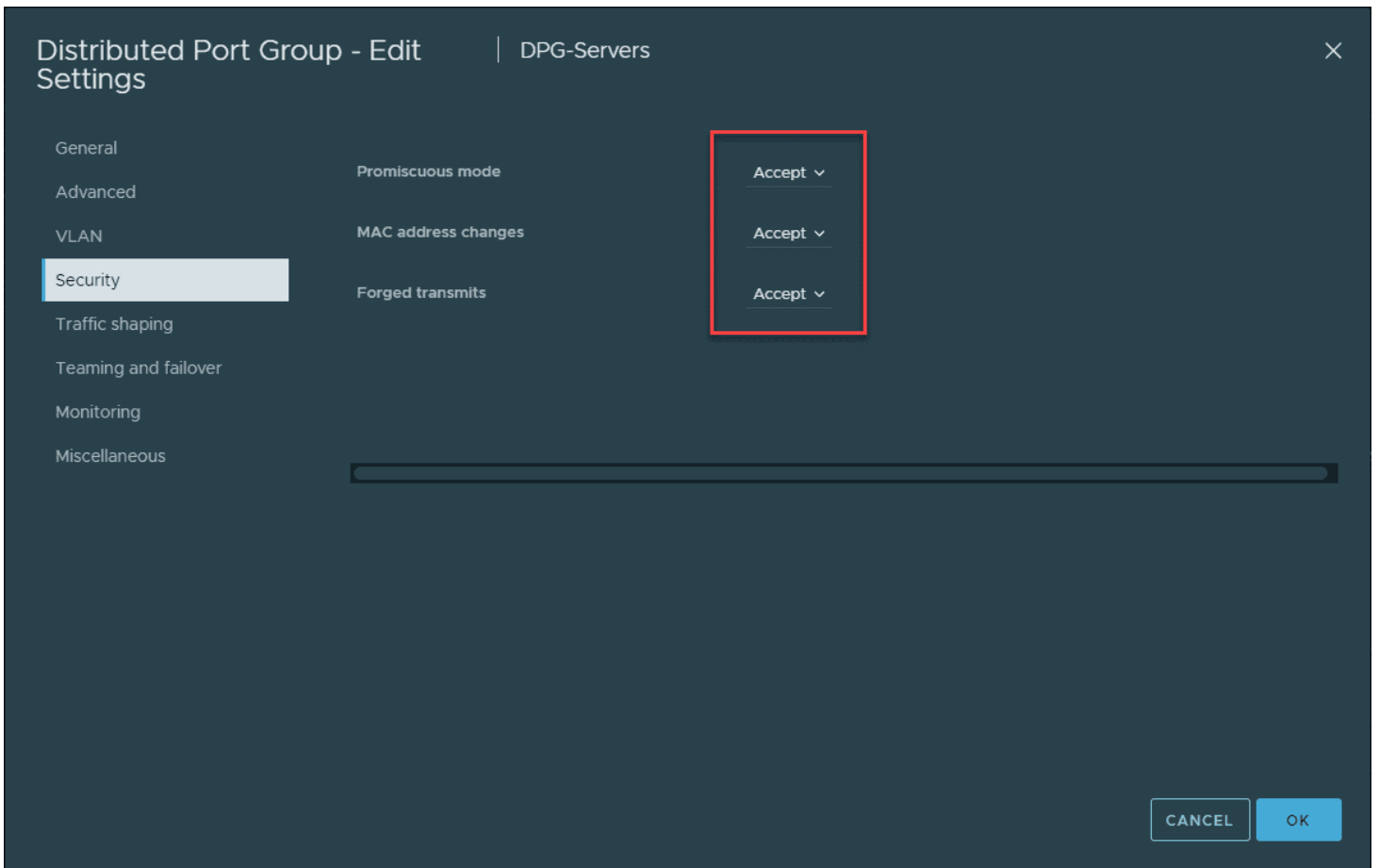
ADD NEW DEVICE ▾

▼ CPU	8 ▾	
Cores per Socket	1 ▾	Sockets: 8
CPU Hot Plug	<input type="checkbox"/> Enable CPU Hot Add	
Reservation	0 ▾	MHz ▾
Limit	Unlimited ▾	MHz ▾
Shares	Normal ▾	8000 ▾
Hardware virtualization	<input checked="" type="checkbox"/> Expose hardware assisted virtualization to the guest OS	
Performance Counters	<input type="checkbox"/> Enable virtualized CPU performance counters	
I/O MMU	<input type="checkbox"/> Enabled	
> Memory	24 ▾	GB ▾
> Hard disk 1	200 ▾	GB ▾
> SCSI controller 0	VMware Paravirtual	
> Network adapter 1	DPG-Servers ▾	<input checked="" type="checkbox"/> Connected
> Network adapter 2	▾	<input type="checkbox"/> Connected
> CD/DVD drive 1	Datastore ISO File ▾	<input checked="" type="checkbox"/> Connected
> Video card	Specify custom settings ▾	
> Security Devices	Not Configured	
VMCI device		
SATA controller 0	AHCI	
> Other	Additional Hardware	

CANCEL OK

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 \(github.com\)](https://github.com/harvester/harvester/releases)

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

GNU GRUB version 2.04

*Harvester Installer v1.0.2

Use the ↑ and ↓ keys to select which entry is highlighted.
Press enter to boot the selected OS, 'e' to edit the commands
before booting or 'c' for a command-line.
The highlighted entry will be executed automatically in 4s.

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.

Choose installation mode

Create a new Harvester cluster
Join an existing Harvester cluster

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 installation target. Device will be formatted

sda 80G

Use MBR partitioning scheme

No

<Use ESC to go back to previous page>

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.

Choose installation target. Device will be formatted

sda 80G

Use MBR partitioning scheme

No

Note: GPT is used by default. You can use MBR if you encountered compatibility issues.

Disk size is smaller than the recommended size: 140GB. Press Enter to continue.

<Use ESC to go back to previous page, Use TAB to choose other options>

Error about disk size in Harvester

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

Configure network connection

HostName

harvester01

Management NIC

ens192 >

Bond Mode

balance-tlb >

IPv4 Method

Automatic (DHCP) >

Applying network configuration...!

<Use ESC to go back to previous page>

Setting management NIC bond mode and IPv4 configuration

Configure the DNS servers you want to use.

Configure DNS Servers

DNS Servers

Note: You can use comma to add more DNS servers. Leave blank to use def

<Use ESC to go back to previous page>

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 VIP

VIP Mode

Automatic (DHCP) >

VIP

10.1.149.167

<Use ESC to go back to previous page>

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 cluster token

Cluster token

Note: The token is used for adding nodes to the cluster

<Use ESC to go back to previous page>

Configure your cluster token for Harvester

Configure your password to access the Harvester node.

Configure the password to access the node

Password

Confirm password

*****_

<Use ESC to go back to previous page>

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.

Optional: Configure NTP Servers

NTP Servers

Note: You can use comma to add more NTP servers.

<Use ESC to go back to previous page>

Configure NTP settings for Harvester

Set the optional proxy configuration if needed.

Optional: configure proxy

Proxy address

Note: In the form of "http://[[user][:pass]@]host[:port]/".

<Use ESC to go back to previous page>

Set a proxy address if needed

Import SSH keys if needed.

Optional: import SSH keys

HTTP URL

For example: `https://github.com/<username>.keys`

<Use ESC to go back to previous page>

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.

Optional: remote Harvester config

HTTP URL

<Use ESC to go back to previous page>

Remote Harvester configuration from HTTP

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

Confirm installation options

```
install mode: create
hostname: harvester01
ntp servers: 0.suse.pool.ntp.org
device: /dev/sda
mode: create
networks:
  harvester-mgmt:
    bond_options:
      miimon: "100"
      mode: balance-tlb
    interfaces:
      - name: ens192
      method: dhcp
vip: 10.1.149.167
vip_hw_addr: 76:88:d4:cc:d5:15
vip_mode: dhcp
```

Your disk will be formatted and Harvester will be installed with

```
Yes
No
```

<Use ESC to go back to previous page>

Confirm the installation options of Harvester in VMware ESXi

The installation of Harvester begins.

Confirm installation options

```
Installing Harvester
INFO[2022-06-14T19:26:38Z] Starting elemental version v0.0.14
INFO[2022-06-14T19:26:38Z] Install called
INFO[2022-06-14T19:26:38Z] Running before-install hook
INFO[2022-06-14T19:26:38Z] Partitioning device...
INFO[2022-06-14T19:26:38Z] Setting custom partitions from /tmp/part-lay
out.358935623...
INFO[2022-06-14T19:26:38Z] Running stage: partitioning
INFO[2022-06-14T19:26:38Z] Executing /tmp/part-layout.358935623
INFO[2022-06-14T19:26:38Z] Applying 'Root partitioning layout' for stag
e 'partitioning'. Total stages: 1
INFO[2022-06-14T19:26:38Z] Processing stage step 'Root partitioning lay
out'. ( commands: 0, files: 0, ... )
INFO[2022-06-14T19:26:39Z] Creating COS_OEM partition
INFO[2022-06-14T19:26:41Z] Creating COS_STATE partition
```

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
version: v1.0.2
```

```
Harvester management URL:
```

```
https://10.1.149.169
```

```
Current status:
```

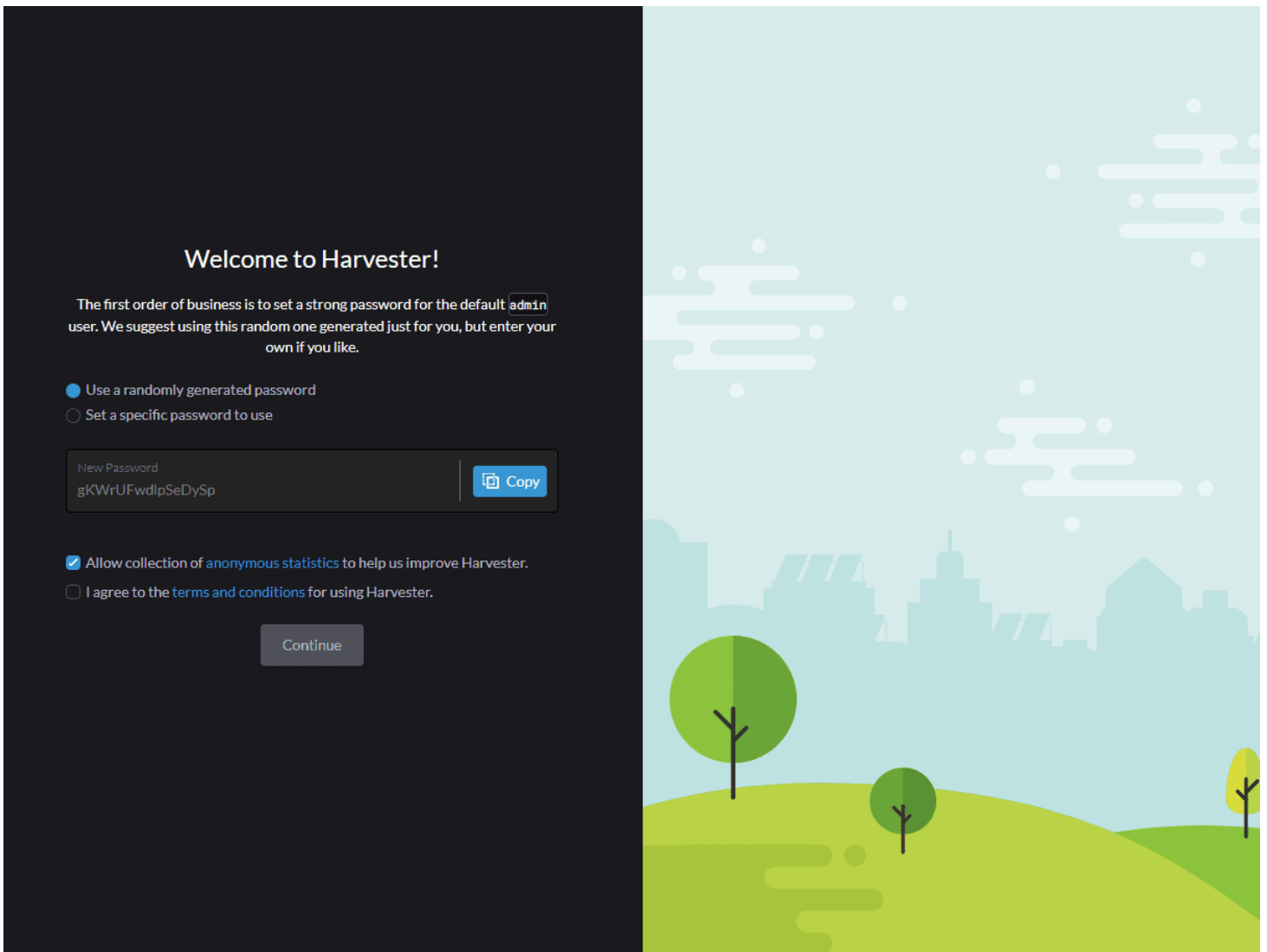
```
Ready
```

```
<Use F12 to switch between Harvester console and Shell>
```

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.

The screenshot shows the Harvester web interface for creating a new image. The left sidebar contains a navigation menu with items: Dashboard, Hosts, Virtual Machines, Volumes, Images (highlighted with a red arrow), Namespaces, and Advanced. The main content area is titled 'Image: Create' and features a form with the following fields:

- Namespace:** default
- Name:** ubuntu-22.04-live-server-amd64.iso
- Description:** Any text you want that better describes this resource

Below the form, there are two tabs: 'Basics' (selected) and 'Labels'. Under the 'Basics' tab, there are two radio buttons: 'URL' (selected) and 'File'. A text input field under the 'URL' radio button contains the following text:

```
https://releases.ubuntu.com/22.04/ubuntu-22.04-live-server-amd64.iso
```

At the bottom of the form, there are three buttons: 'Cancel', 'Edit as YAML', and 'Create'. The footer of the interface includes 'Support', 'v1.0.2', and 'English'.

Uploading an Ubuntu image for installing Ubuntu 22.04 Server

The download of Ubuntu 22.04 ISO image begins.

Harvester

All Namespaces

Dashboard
Hosts
Virtual Machines
Volumes
Images
Namespaces
Advanced

Images

Create

Download YAML Delete

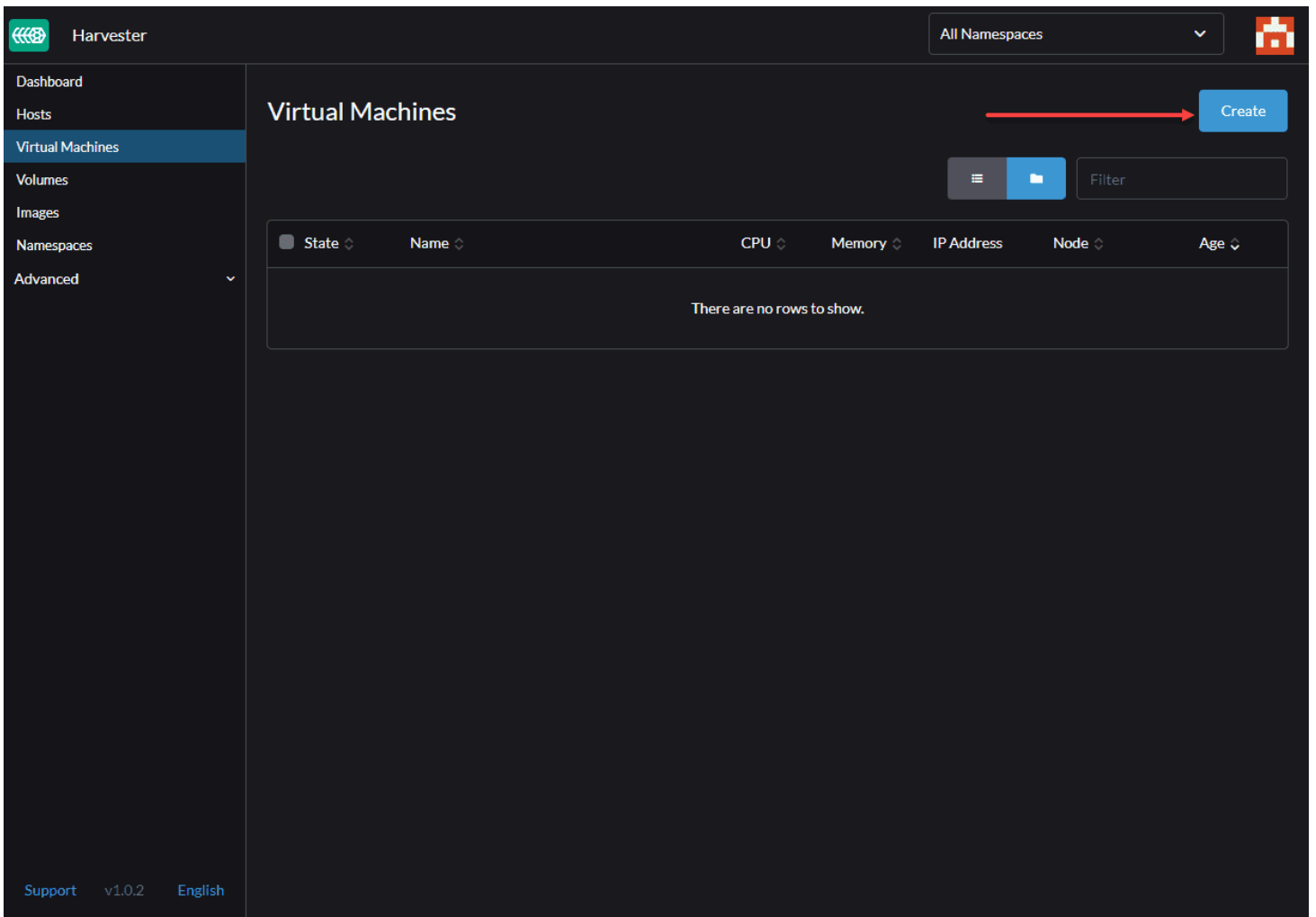
Filter

State	Name	Progress	Size	Age
Namespace: default				
Downloading	ubuntu-22.04-live-server-amd64.iso	27%	1.37 GB	16 secs

Support v1.0.2 English

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.

Harvester

All Namespaces

Dashboard
Hosts
Virtual Machines
Volumes
Images
Namespaces
Advanced

Virtual Machine: Create

Single Instance Multiple Instance

Namespace: default | Name: **ubuntu2204** | Description: Any text you want that better describes this resource

Use VM Template:

- Basics
- Volumes
- Networks
- Node Scheduling
- Advanced Options

Basics

CPU: 2 | Memory: 2 GIB

SSHKey

Cancel Edit as YAML Create

Support v1.0.2 English

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**.

Harvester

All Namespaces

Dashboard

Hosts

Virtual Machines

Volumes

Images

Namespaces

Advanced

Virtual Machine: Create

Single Instance Multiple Instance

Namespace: default

Name: ubuntu2204

Description: Any text you want that better describes this resource

Use VM Template:

- Basics
- Volumes
- Networks
- Node Scheduling
- Advanced Options

Volumes

Drag and drop volumes, or use the volume's arrows, to change the boot order.

Image Volume

Name: disk-0

Type: cd-rom

Image: default/ubuntu-22.04-live-server-amd64.iso

Size: 10 GiB

Bus: SATA

bootOrder: 1

Volume

Name: disk-1

Type: disk

Size: 15 GiB

Bus: VirtIO

bootOrder: 2

Add Volume Add Existing Volume Add VM Image Add Container

Support v1.0.2 English

Cancel Edit as YAML Create

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**.

Harvester

All Namespaces

Dashboard
Hosts
Virtual Machines
Volumes
Images
Namespaces
Advanced

Virtual Machine: Create

Single Instance Multiple Instance

Namespace: default Name: ubuntu2204 Description: Any text you want that better describes this resource

Use VM Template:

- Basics
- Volumes
- Networks
- Node Scheduling
- Advanced Options

Networks

Network		✕
Name *	default	
Model *	virtio	
Network *	management Network	
Type *	bridge	

Add Network

Cancel Edit as YAML Create

Support v1.0.2 English

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.

The screenshot shows the Harvester web interface for creating a virtual machine. The left sidebar contains navigation links: Dashboard, Hosts, Virtual Machines (selected), Volumes, Images, Namespaces, and Advanced. The main content area is titled 'Virtual Machine: Create' and features a 'All Namespaces' dropdown in the top right. Below the title, there are radio buttons for 'Single Instance' (selected) and 'Multiple Instance'. A form contains a 'Namespace' dropdown set to 'default', a 'Name' field with 'ubuntu2204', and a 'Description' field with the placeholder text 'Any text you want that better describes this resource'. A checkbox for 'Use VM Template:' is present but unchecked. A vertical sidebar on the left of the main content area lists configuration sections: Basics, Volumes, Networks, Node Scheduling (highlighted), and Advanced Options. The 'Node Scheduling' section is expanded, showing three radio button options: 'Run VM on any available node' (selected), 'Run VM on specific node(s) - (Live migration is not supported)', and 'Run VM on node(s) matching scheduling rules'. At the bottom right of the form, there are three buttons: 'Cancel', 'Edit as YAML', and 'Create'. The footer of the interface includes 'Support', 'v1.0.2', and 'English'.

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**.

Harvester

All Namespaces

Dashboard
Hosts
Virtual Machines
Volumes
Images
Namespaces
Advanced

Virtual Machine: Create

Single Instance Multiple Instance

Namespace: default | Name: ubuntu2204 | Description: Any text you want that better describes this resource

Use VM Template:

Basics
Volumes
Networks
Node Scheduling
Advanced Options

Advanced Options

Run Strategy: RerunOnFailure | OS Type: Ubuntu

[Show More](#)

Cloud Config

User Data:

You can specify user data to configure an instance or run a configuration script during launch. If you launch more than one instance at a time, the user data is available to all the instances in that reservation. [Learn more](#)

User Data Template

```
1 #cloud-config
2 package_update: true
3 packages:
4   - qemu-guest-agent
5 runcmd:
```

Support v1.0.2 English

Cancel Edit as YAML Create

Viewing advanced options for the Harvester VM

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

The screenshot displays the Harvester web interface. On the left is a navigation sidebar with options: Dashboard, Hosts, Virtual Machines (highlighted), Volumes, Images, Namespaces, and Advanced. The main content area is titled 'Virtual Machines' and includes a 'Create' button. Below the title are action buttons: Stop, Restart, Download YAML, and Delete. A filter input is also present. A table lists the VMs with columns for State, Name, CPU, Memory, IP Address, Node, and Age. One VM is shown in the 'default' namespace, named 'ubuntu2204', with a state of 'Running', 2 CPU, 2 Gi memory, IP 10.52.0.88, on node 'harvester01', and has been running for 39 seconds. A 'Console' button is visible next to the VM name. At the bottom left, there are links for 'Support', 'v1.0.2', and 'English'.

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.

Harvester

All Namespaces

Dashboard
Hosts
Virtual Machines
Volumes
Images
Namespaces
Advanced

Virtual Machine: ubuntu2204 Running

Namespace: default Age: 46 secs

Labels: `harvesterhci.io/creator: harvester` `harvesterhci.io/os: ubuntu`

Annotations: Show 5 annotations

- Basics
- Volumes
- Networks
- SSH Keys
- VM Metrics
- Access Credentials
- Cloud Config
- Events
- Migration

Events

Reason	Resource	Date
Started	VirtualMachineInstance ubuntu2204 VirtualMachineInstance started.	20 secs ago
Created	VirtualMachineInstance ubuntu2204 VirtualMachineInstance defined.	20 secs ago
Migrated	VirtualMachineInstance ubuntu2204 EvictionStrategy is set but vmi is not migratable	20 secs ago
Started	Pod virt-launcher-ubuntu2204-g4tn7 Started container compute	24 secs ago
SuccessfulMountVolume	Pod virt-launcher-ubuntu2204-g4tn7 MapVolume.MapPodDevice succeeded for volume "pvc-0633a5f1-b129-44de-8ee4-b27f3aaa691e" globalMapPath "/var/lib/kubelet/plugins/kubernetes.io/csi/volumeDevices/pvc-0633a5f1-b129-44de-8ee4-b27f3aaa691e/dev"	25 secs ago
SuccessfulMountVolume	Pod virt-launcher-ubuntu2204-g4tn7 MapVolume.MapPodDevice succeeded for volume "pvc-0633a5f1-b129-44de-8ee4-b27f3aaa691e" volumeMapPath "/var/lib/kubelet/pods/8a3f2907-6d57-4e07-83bf-fd19fb1fb6d0/volumeDevices/kubernetes.io~csi"	25 secs ago
SuccessfulMountVolume	Pod virt-launcher-ubuntu2204-g4tn7 MapVolume.MapPodDevice succeeded for volume "pvc-8eb92fd6-a76c-4249-9387-8931c087e013" globalMapPath "/var/lib/kubelet/plugins/kubernetes.io/csi/volumeDevices/pvc-8eb92fd6-a76c-4249-9387-8931c087e013/dev"	25 secs ago
SuccessfulMountVolume	Pod virt-launcher-ubuntu2204-g4tn7 MapVolume.MapPodDevice succeeded for volume "pvc-8eb92fd6-a76c-4249-9387-8931c087e013" volumeMapPath "/var/lib/kubelet/pods/8a3f2907-6d57-4e07-83bf-fd19fb1fb6d0/volumeDevices/kubernetes.io~csi"	25 secs ago

Support v1.0.2 English

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**.

The screenshot shows the Harvester web interface. On the left is a sidebar with navigation links: Dashboard, Hosts, Virtual Machines (selected), Volumes, Images, Namespaces, and Advanced. The main area is titled 'Virtual Machines' and includes a 'Create' button and action buttons for 'Stop', 'Restart', 'Download YAML', and 'Delete'. Below these is a table of virtual machines. The table has columns for State, Name, CPU, Memory, IP Address, Node, and Age. A single row is visible for a VM named 'ubuntu2204' in the 'default' namespace, which is in a 'Running' state. It has 2 CPUs, 2 Gi of memory, IP address 10.52.0.88, and is located on node 'harvester01'. A 'Console' button is highlighted with a red arrow, and a dropdown menu is open showing two options: 'Open in Web VNC' and 'Open in Serial Console'.

State	Name	CPU	Memory	IP Address	Node	Age
Running	ubuntu2204	2	2 Gi	10.52.0.88	harvester01	1.2 mins

Opening the virtual machine console in Harvester

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

```
[ OK ] Listening on Open-iscsi iscsid Socket.
Starting Socket activation for snappy daemon...
[ OK ] Listening on UUID daemon activation socket.
[ OK ] Reached target Preparation for Remote File Systems.
[ OK ] Reached target Remote File Systems.
[ OK ] Finished Availability of block devices.
[ OK ] Listening on Socket activation for snappy daemon.
[ OK ] Reached target Socket Units.
[ OK ] Reached target Basic System.
Starting LSB: automatic crash report generation...
Starting casper-md5check Verify Live ISO checksums...
[ OK ] Started Regular background program processing daemon.
[ OK ] Started D-Bus System Message Bus.
[ OK ] Started Save Initial Kernel messages after boot.
Starting Remove Stale Online ext4 Metadata Check Snapshots...
[ OK ] Reached target Login Prompts.
[ OK ] Started irqbalance daemon.
Starting Dispatcher daemon for systemd-networkd...
Starting Authorization Manager...
Starting Pollinate to seed the pseudo random number generator...
Starting System Logging Service...
Starting Snap Daemon...
Starting User Login Management...
Starting Permit User Sessions...
Starting Disk Manager...
[ OK ] Finished Permit User Sessions.
Starting Hold until boot process finishes up...
Starting Terminate Plymouth Boot Screen...
[ OK ] Finished Hold until boot process finishes up.
Starting Set console scheme...
[ OK ] Finished Terminate Plymouth Boot Screen.
[ OK ] Started LSB: automatic crash report generation.
[ OK ] Finished Set console scheme.
[ OK ] Started System Logging Service.
Starting Hostname Service...
[ OK ] Started Hostname Service.
[ OK ] Started User Login Management.
[ OK ] Started Unattended Upgrades Shutdown.
[ OK ] Started Authorization Manager.
Starting Modem Manager...
[ OK ] Started Dispatcher daemon for systemd-networkd.
[ OK ] Finished Remove Stale Online ext4 Metadata Check Snapshots.
[ OK ] Started Disk Manager.
[ OK ] Started Modem Manager.
[ OK ] Finished Pollinate to seed the pseudo random number generator.
Starting OpenBSD Secure Shell server...
[ OK ] Started OpenBSD Secure Shell server.
```

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](#)

minikube v1.18.1
Kubernetes
Visit website [↗](#)

Add to list +

How to install

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

Daily views
24 views in the last week

About minikube

minikube quickly sets up a local Kubernetes cluster on macOS, Linux, and Windows. We proudly focus on helping application developers and new Kubernetes users.

Tags

kubernetes container containerization virtualization

License

[↗](#) Apache 2.0

Versions

- 1.18.1 [↗](#)
- 1.18.0 [↗](#)
- 1.17.1 [↗](#)
- 1.17.0 [↗](#)

Show 2 older versions

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 Minikube in Windows Server 2022. However, there are a few prerequisites I will mention below:

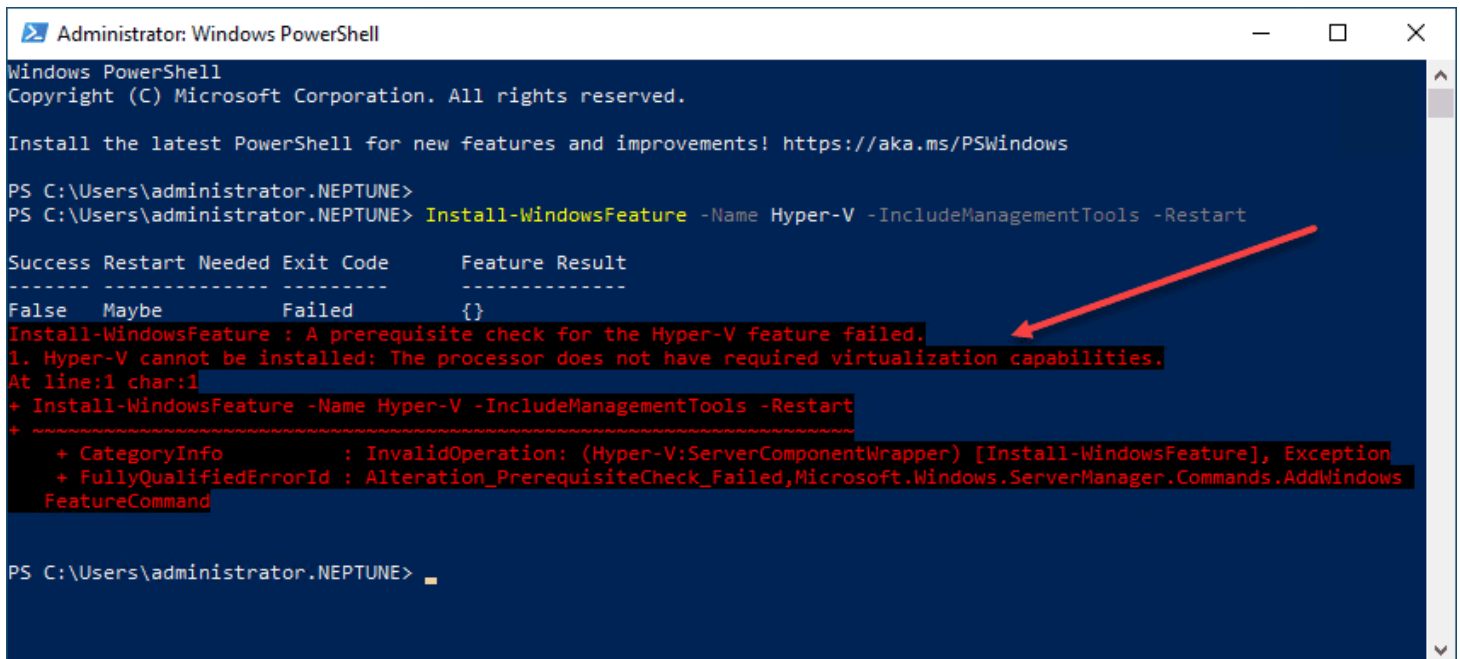
1. Installation of Hyper-V Server
 - Enable nested virtualization if using Hyper-V inside a VM
2. Install winget using the steps in the blog post linked above
3. [Install Minikube in Windows Server 2022](#)
4. Create a Hyper-V External Virtual Switch
5. Build your Minikube [Kubernetes](#) cluster
6.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



```
Administrator: Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\administrator.NEPTUNE>
PS C:\Users\administrator.NEPTUNE> Install-WindowsFeature -Name Hyper-V -IncludeManagementTools -Restart

Success Restart Needed Exit Code      Feature Result
-----
False  Maybe          Failed          {}

Install-WindowsFeature : A prerequisite check for the Hyper-V feature failed.
1. Hyper-V cannot be installed: The processor does not have required virtualization capabilities.
At line:1 char:1
+ Install-WindowsFeature -Name Hyper-V -IncludeManagementTools -Restart
+ ~~~~~
+ CategoryInfo          : (InvalidOperation: (Hyper-V:ServerComponentWrapper) [Install-WindowsFeature], Exception
+ FullyQualifiedErrorId : Alteration_PrerequisiteCheck_Failed,Microsoft.Windows.ServerManager.Commands.AddWindows
FeatureCommand

PS C:\Users\administrator.NEPTUNE> _
```

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
```

```
Administrator: Windows PowerShell

validate  Validates a manifest file
settings  Open settings or set administrator settings
features  Shows the status of experimental features
export    Exports a list of the installed packages
import    Installs all the packages in a file

For more details on a specific command, pass it the help argument. [-?]

The following options are available:
-v,--version  Display the version of the tool
--info        Display general info of the tool

More help can be found at: https://aka.ms/winget-command-help
PS C:\Users\administrator.NEPTUNE> winget install -e --id Kubernetes.minikube
The `msstore` source requires that you view the following agreements before using.
Terms of Transaction: https://aka.ms/microsoft-store-terms-of-transaction
The source requires current machine's geographic region to be sent to function properly.

Do you agree to all the source agreements terms?
[Y] Yes [N] No: Y
Found Kubernetes - Minikube - A Local Kubernetes Development Environment [Kubernetes.minikube] Version 1.24.0
This application is licensed to you by its owner.
Microsoft is not responsible for, nor does it grant any licenses to, third-party packages.
Downloading https://github.com/kubernetes/minikube/releases/download/v1.24.0/minikube-installer.exe
14.8 MB / 27.8 MB
```

Install Minikube in Windows Server 2022

You can read the official documentation here:

- [Download and install minikube with winget](#)

minikube v1.18.1

Kubernetes

Visit website [↗](#)

Add to list



Daily views

24 views in the last week



Versions

1.18.1



1.18.0



1.17.1



1.17.0



Show 2 older versions

How to install

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



About minikube

minikube quickly sets up a local Kubernetes cluster on macOS, Linux, and Windows. We proudly focus on helping application developers and new Kubernetes users.

Tags

kubernetes

container

containerization

virtualization

License

[↗](#) Apache 2.0

Install Minikube using winget in Windows Server 2022

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


```
Administrator: Windows PowerShell
Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\administrator.NEPTUNE> cmd
Microsoft Windows [Version 10.0.20348.350]
(c) Microsoft Corporation. All rights reserved.

C:\Users\administrator.NEPTUNE>minikube ←
minikube provisions and manages local Kubernetes clusters optimized for development workflows.

Basic Commands:
start           Starts a local Kubernetes cluster
status         Gets the status of a local Kubernetes cluster
stop           Stops a running local Kubernetes cluster
delete         Deletes a local Kubernetes cluster
dashboard      Access the Kubernetes dashboard running within the minikube cluster
pause          pause Kubernetes
unpause        unpause Kubernetes

Images Commands:
docker-env     Configure environment to use minikube's Docker daemon
podman-env     Configure environment to use minikube's Podman service
cache          Add, delete, or push a local image into minikube
image          Manage images

Configuration and Management Commands:
addons         Enable or disable a minikube addon
config         Modify persistent configuration values
profile        Get or list the current profiles (clusters)
update-context Update kubeconfig in case of an IP or port change

Networking and Connectivity Commands:
service        Returns a URL to connect to a service
tunnel         Connect to LoadBalancer services

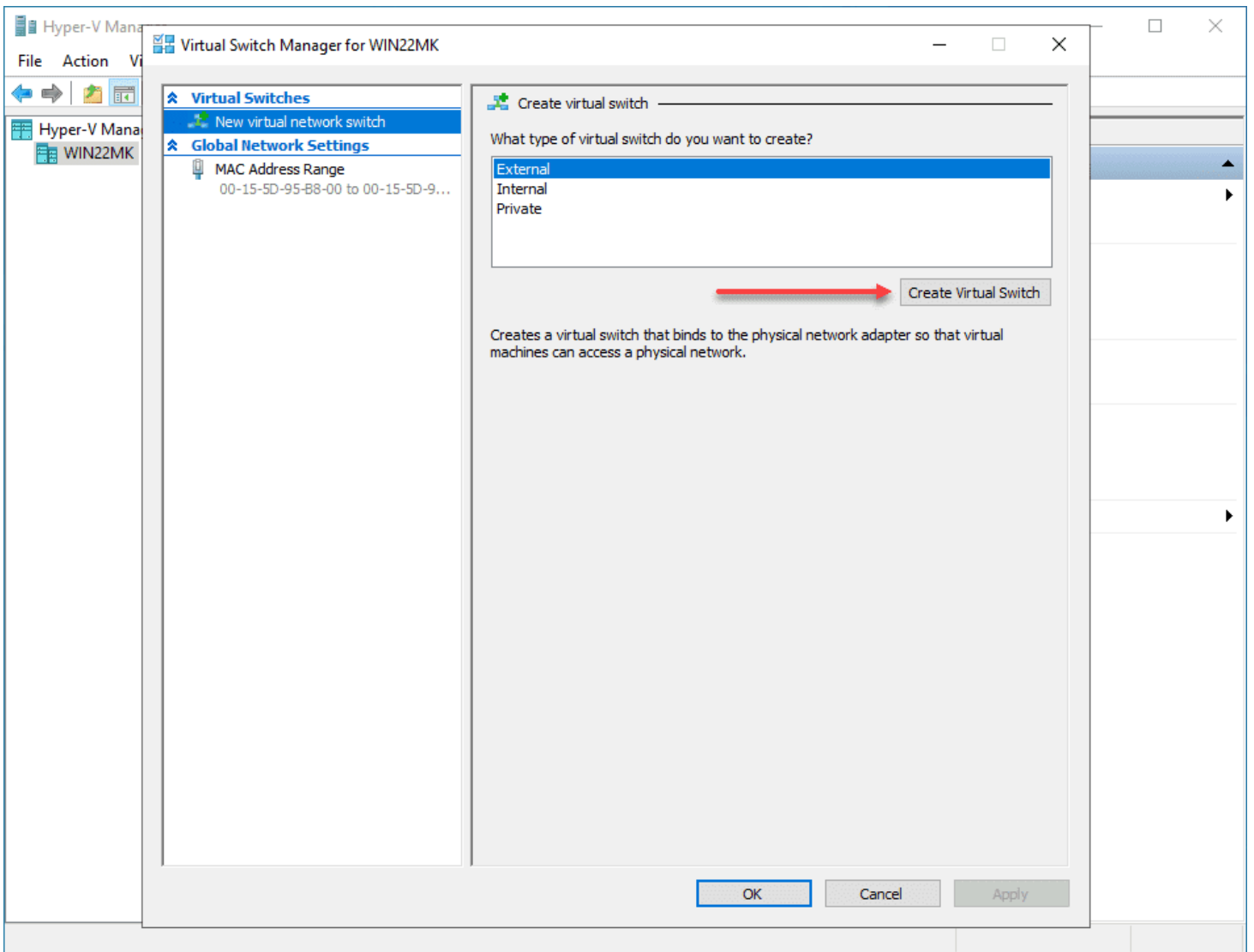
Advanced Commands:
mount          Mounts the specified directory into minikube
ssh            Log into the minikube environment (for debugging)
kubectl        Run a kubectl binary matching the cluster version
node           Add, remove, or list additional nodes
cp             Copy the specified file into minikube

Troubleshooting Commands:
ssh-key        Retrieve the ssh identity key path of the specified node
ssh-host       Retrieve the ssh host key of the specified node
ip             Retrieves the IP address of the specified node
logs           Returns logs to debug a local Kubernetes cluster
update-check   Print current and latest version number
version        Print the version of minikube
options        Show a list of global command-line options (applies to all commands).
```

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
 - o Generating certificates and keys
 - o Boot up control plane VM
 - o Configure RBAC rules
6. Verify Kubernetes components
7. Enable addons

```
Administrator: Windows PowerShell

config      Modify persistent configuration values
profile     Get or list the current profiles (clusters)
update-context Update kubeconfig in case of an IP or port change

Networking and Connectivity Commands:
service     Returns a URL to connect to a service
tunnel      Connect to LoadBalancer services

Advanced Commands:
mount       Mounts the specified directory into minikube
ssh         Log into the minikube environment (for debugging)
kubectl     Run a kubectl binary matching the cluster version
node        Add, remove, or list additional nodes
cp          Copy the specified file into minikube

Troubleshooting Commands:
ssh-key     Retrieve the ssh identity key path of the specified node
ssh-host    Retrieve the ssh host key of the specified node
ip          Retrieves the IP address of the specified node
logs        Returns logs to debug a local Kubernetes cluster
update-check Print current and latest version number
version     Print the version of minikube
options     Show a list of global command-line options (applies to all commands).

Other Commands:
completion  Generate command completion for a shell

Use "minikube <command> --help" for more information about a given command.

C:\Users\administrator.NEPTUNE>minikube start
* minikube v1.24.0 on Microsoft Windows Server 2022 Standard 10.0.20348 Build 20348
* Automatically selected the hyperv driver
* Downloading VM boot image ...
  > minikube-v1.24.0.iso.sha256: 65 B / 65 B [-----] 100.00% ? p/s 0s
  > minikube-v1.24.0.iso: 225.58 MiB / 225.58 MiB 100.00% 76.56 MiB p/s 3.1s
* Starting control plane node minikube in cluster minikube
* Downloading Kubernetes v1.22.3 preload ...
  > preloaded-images-k8s-v13-v1...: 501.73 MiB / 501.73 MiB 100.00% 81.30 Mi
* Creating hyperv VM (CPUs=2, Memory=3000MB, Disk=20000MB) ...
* Preparing Kubernetes v1.22.3 on Docker 20.10.8 ...
  - Generating certificates and keys ...
  - Booting up control plane ...
  - Configuring RBAC rules ...
* Verifying Kubernetes components...
  - Using image gcr.io/k8s-minikube/storage-provisioner:v5
* Enabled addons: storage-provisioner, default-storageclass
* kubectl not found. If you need it, try: 'minikube kubectl -- get pods -A'
* Done! kubectl is now configured to use "minikube" cluster and "default" namespace by default

C:\Users\administrator.NEPTUNE>
```

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

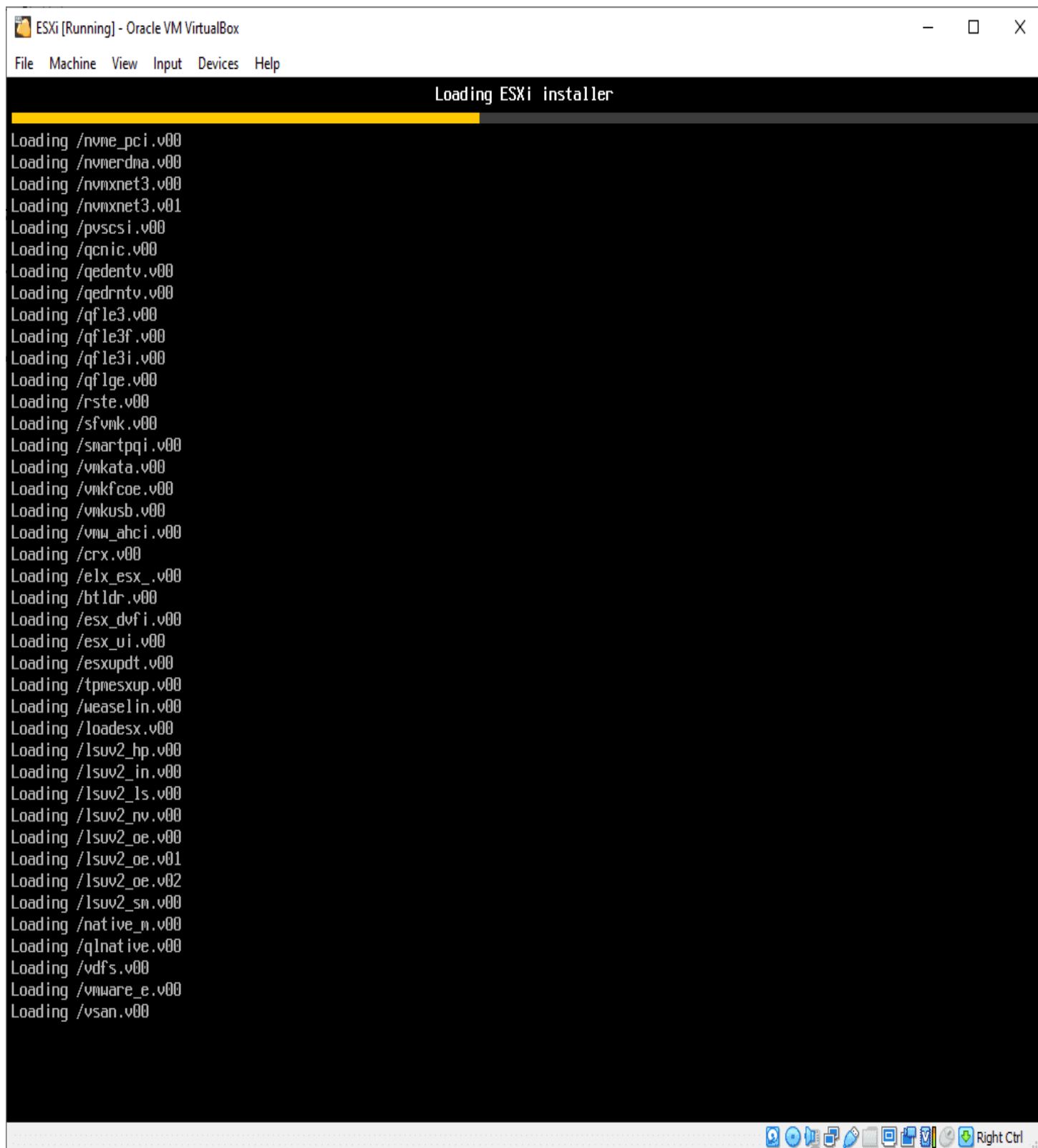
Minikube and really spinning up any Kubernetes framework is a great way to learn. There are many Kubernetes resources out there now that make learning about and getting up to speed quickly with Kubernetes extremely easy. Minikube is one

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

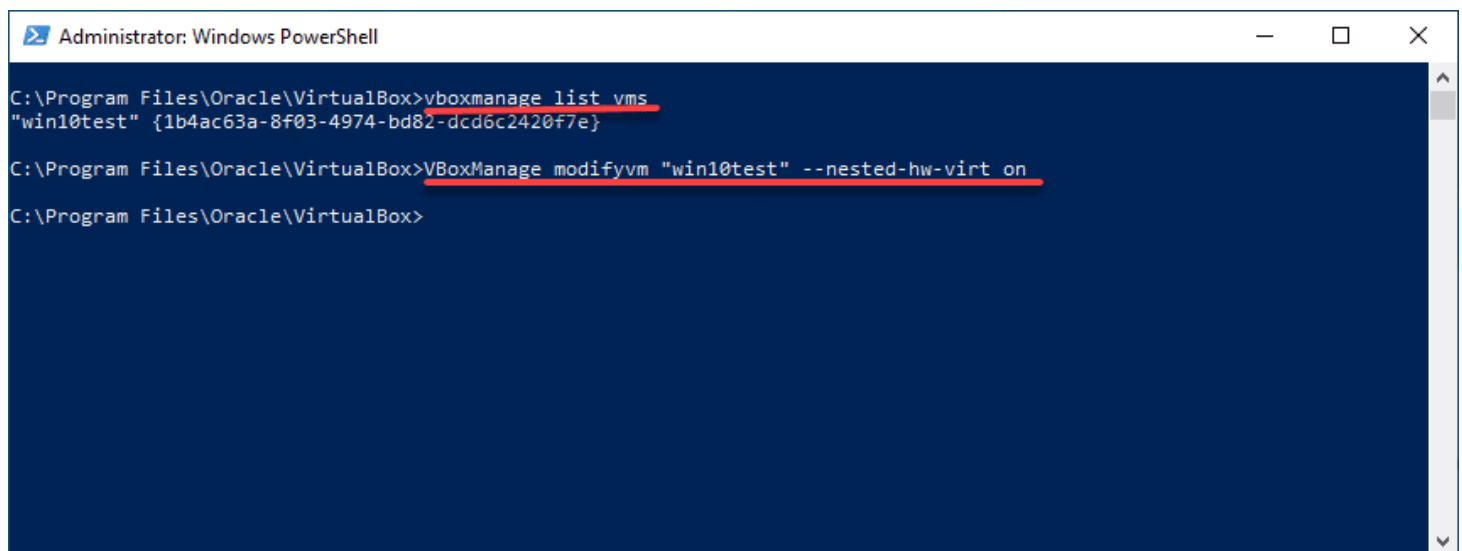
If you are not as familiar with Oracle VirtualBox, especially if you have mainly had your hands on [VMware vSphere](#) or Microsoft [Hyper-V](#), you may not be as familiar with or have ever configured nested virtualization inside of VirtualBox.

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
```

A screenshot of a Windows PowerShell terminal window titled "Administrator: Windows PowerShell". The terminal shows the following commands and output:

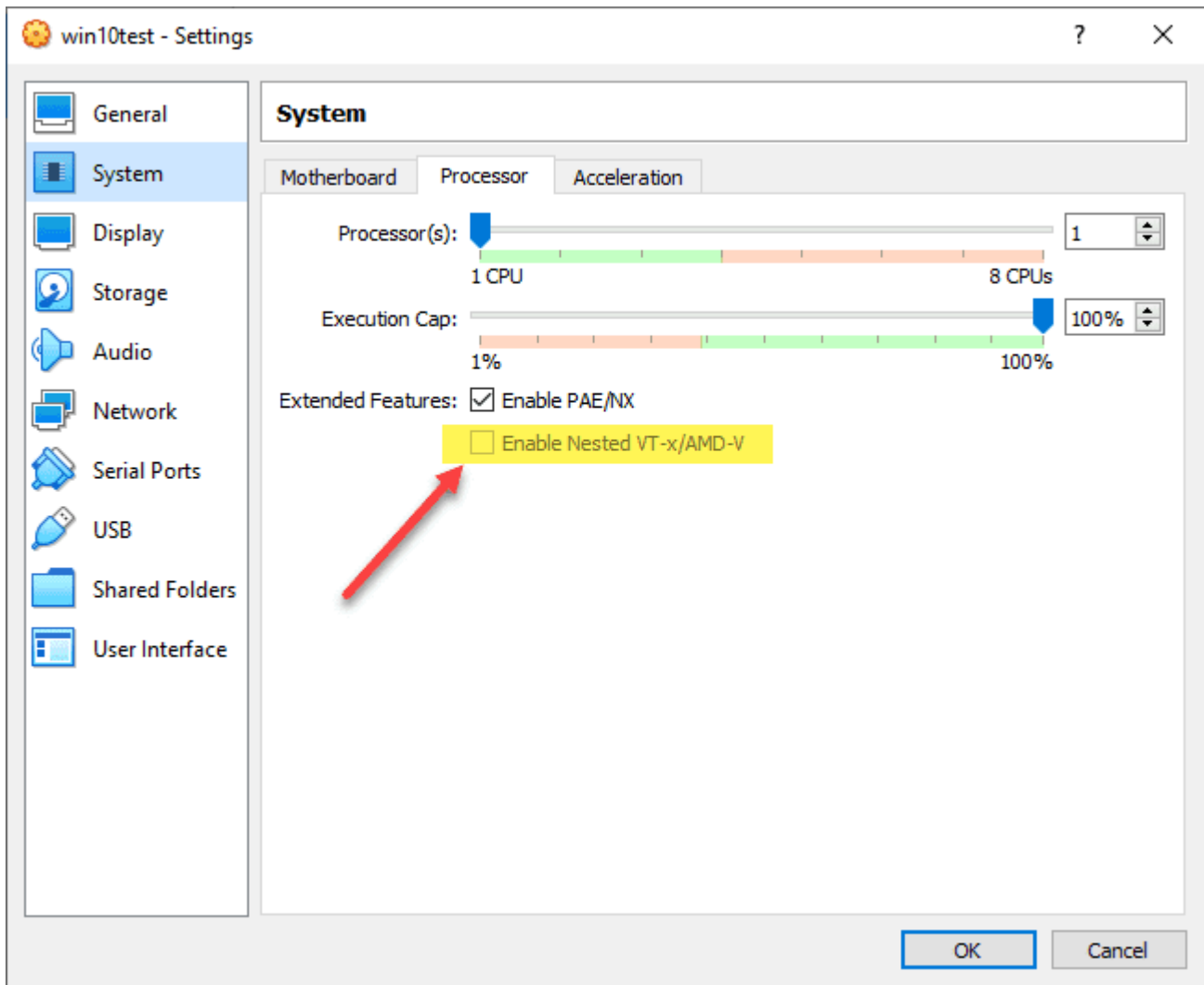
```
C:\Program Files\Oracle\VirtualBox>vboxmanage list vms
"win10test" {1b4ac63a-8f03-4974-bd82-dcd6c2420f7e}

C:\Program Files\Oracle\VirtualBox>VBoxManage modifyvm "win10test" --nested-hw-virt on

C:\Program Files\Oracle\VirtualBox>
```

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 Virtual Machine

Name and operating system

Please choose a descriptive name and destination folder for the new virtual machine and select the type of operating system you intend to install on it. The name you choose will be used throughout VirtualBox to identify this machine.

Name:

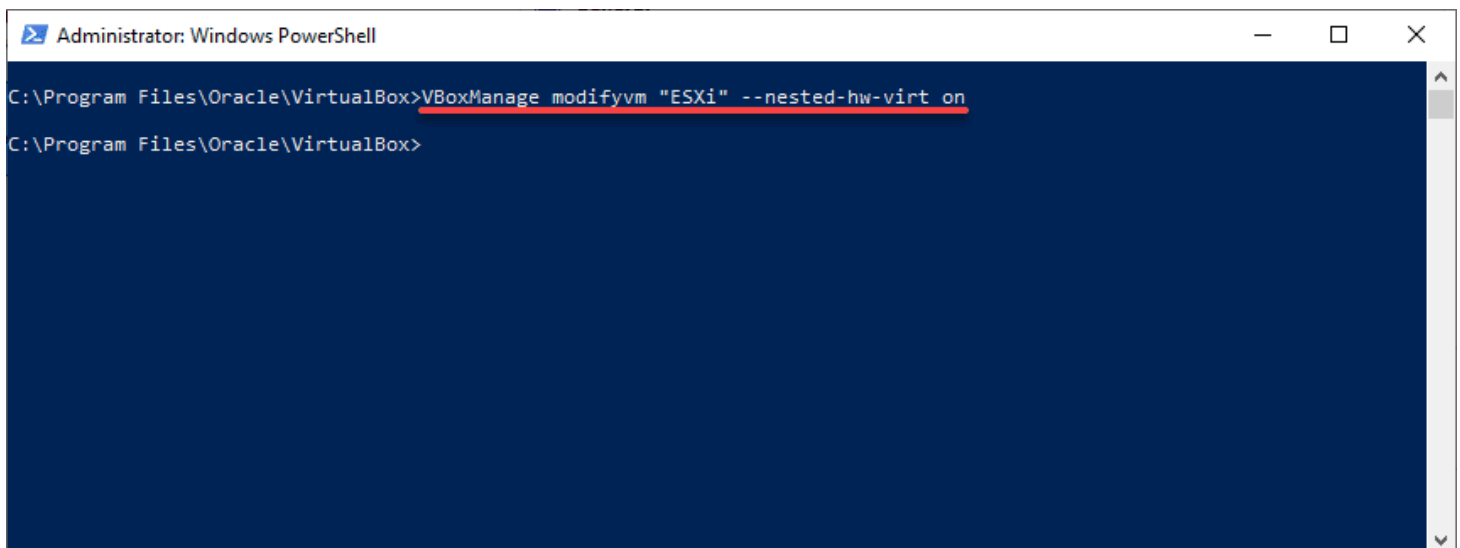
Machine Folder:

Type: 

Version:

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.



```
Administrator: Windows PowerShell
C:\Program Files\Oracle\VirtualBox>VBoxManage modifyvm "ESXi" --nested-hw-virt on
C:\Program Files\Oracle\VirtualBox>
```

Enabling nested virtualization for the esxi host in virtualbox

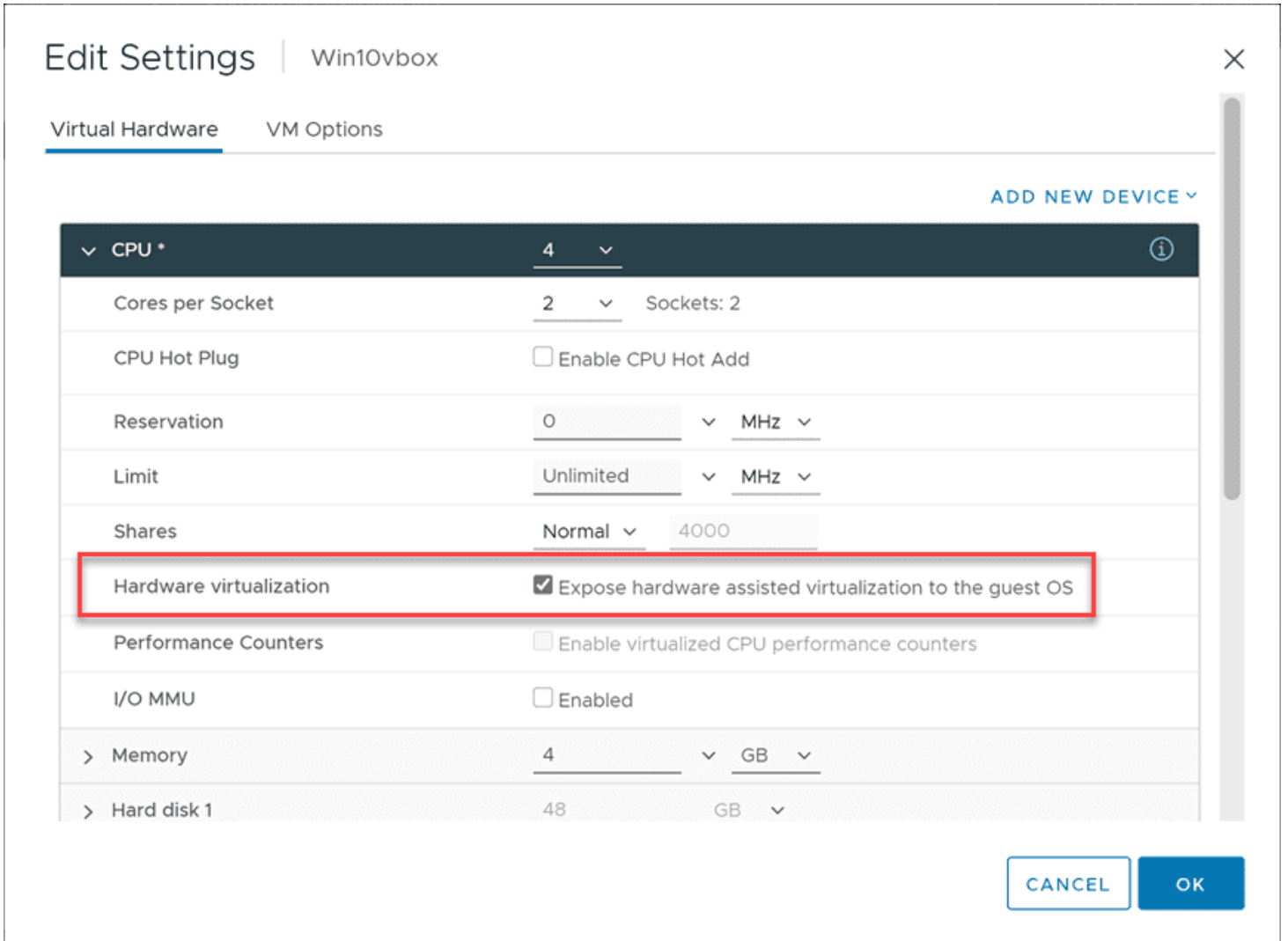
Upload your VMware vSphere ESXi media and load up the ESXi host in VirtualBox.

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:

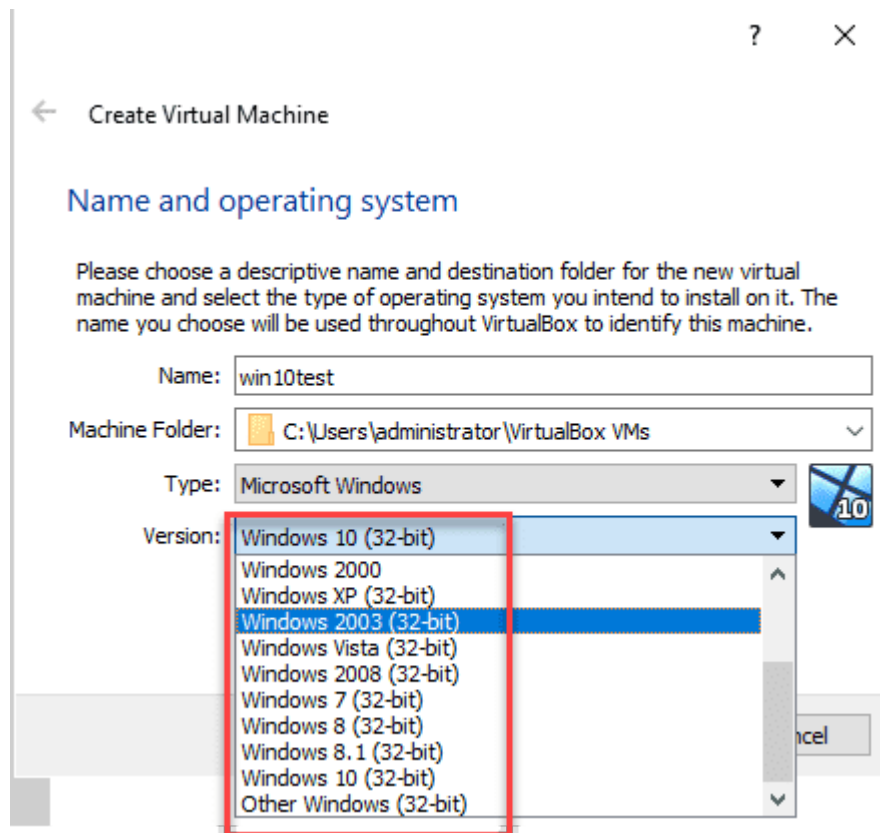
- Expose hardware assisted virtualization to the guest OS



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

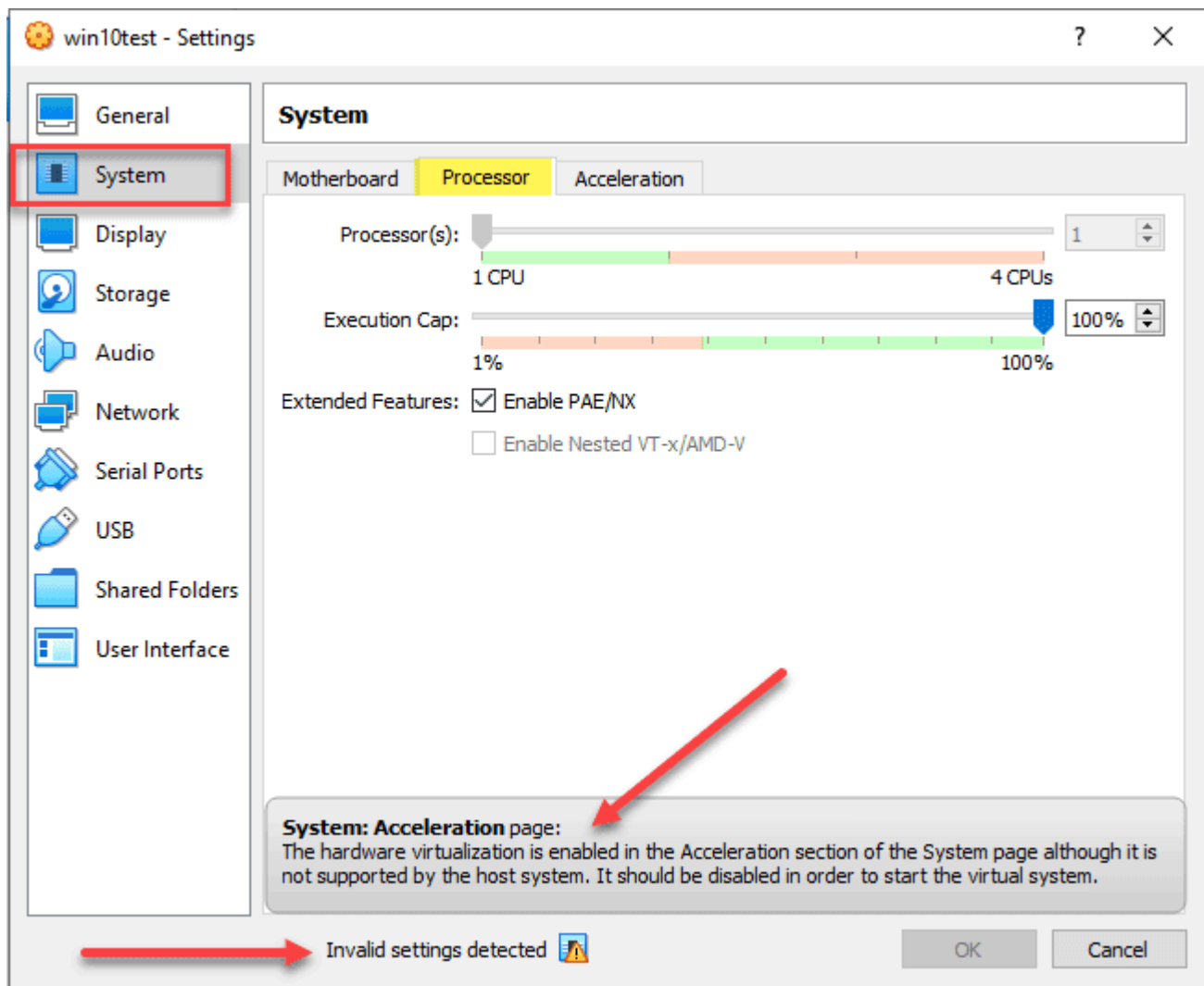
What happens if you don't configure the hardware virtualization setting on the properties of the VirtualBox virtual machine? Unlike [installing Hyper-V or nested ESXi](#) inside VMware ESXi, you won't receive an error during the installation of VirtualBox. The installation will behave the same as a virtual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



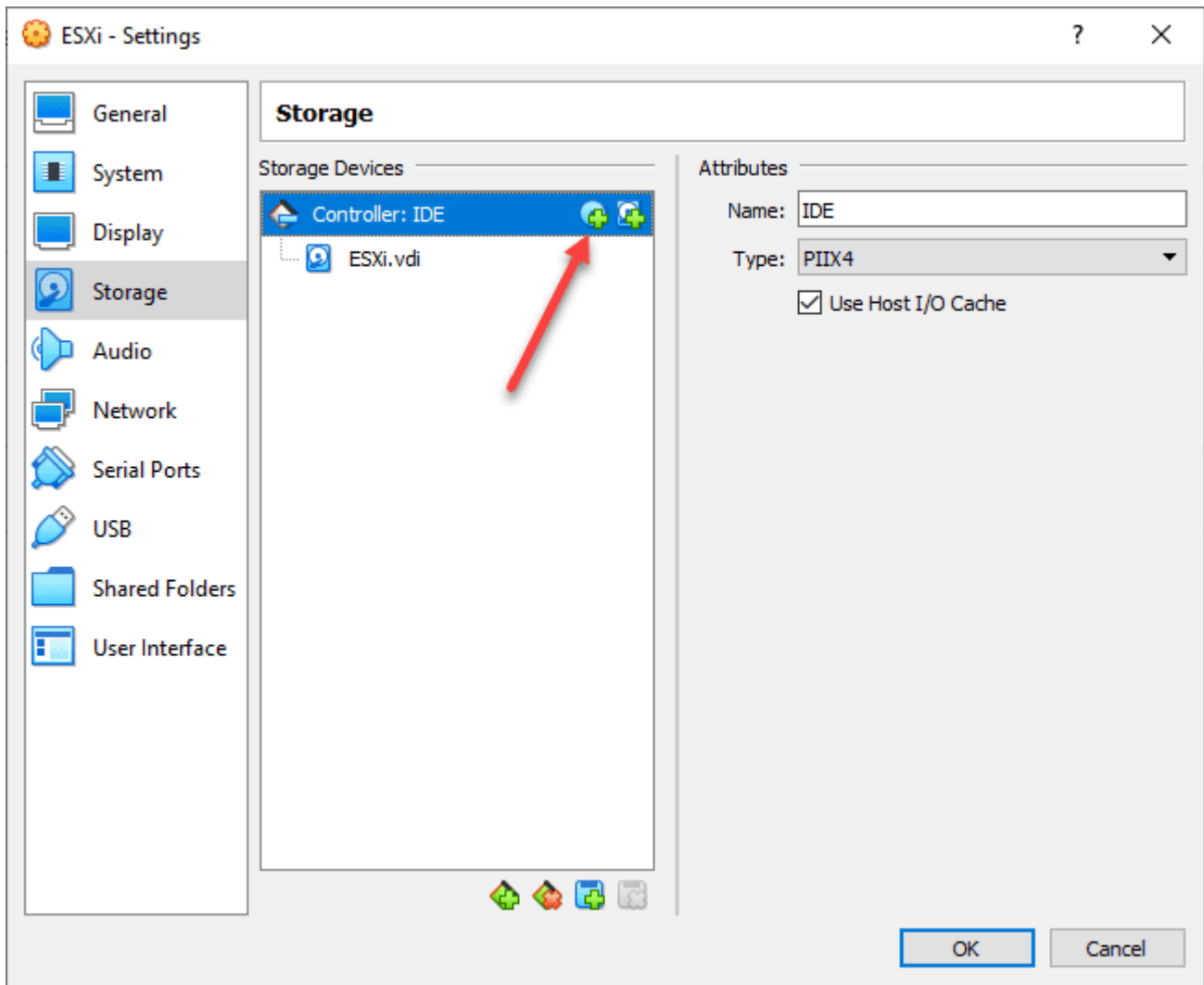
Only 32 bit operating systems are displayed and no server operating systems

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.



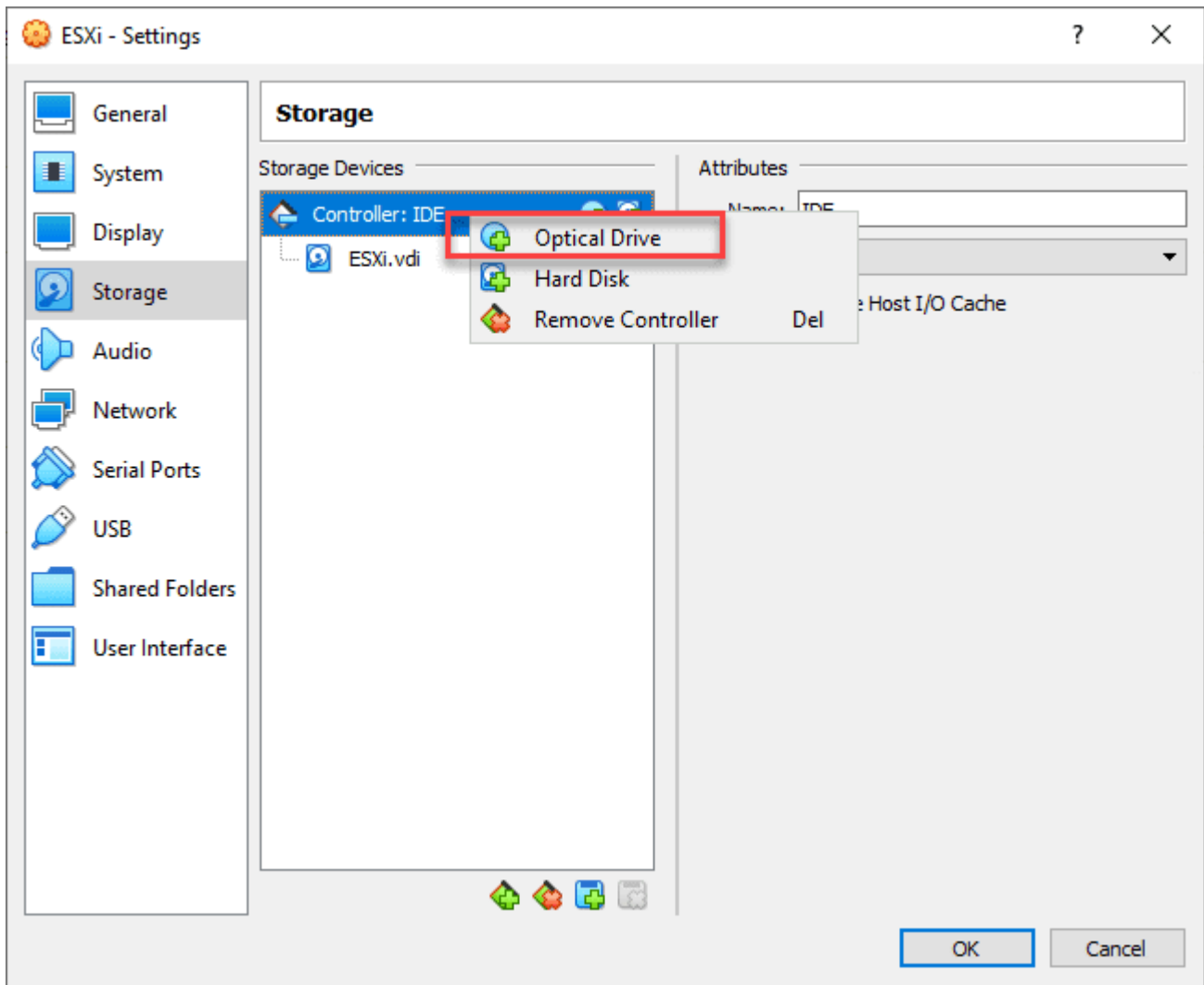
Invalid system settings error displayed in the virtualbox processor settings

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.



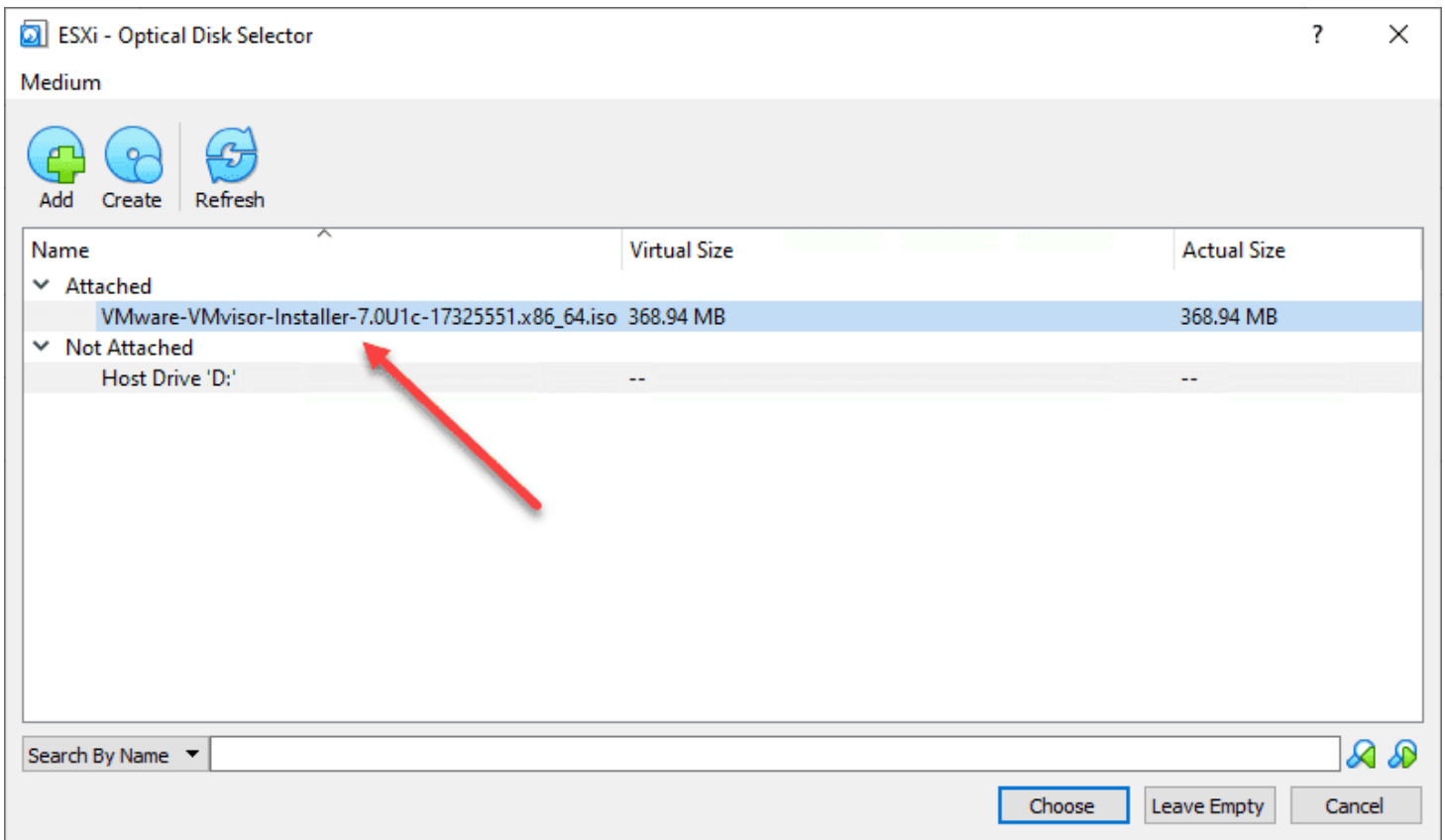
Adding an optical drive to the virtualbox vm

Select **Optical Drive**.



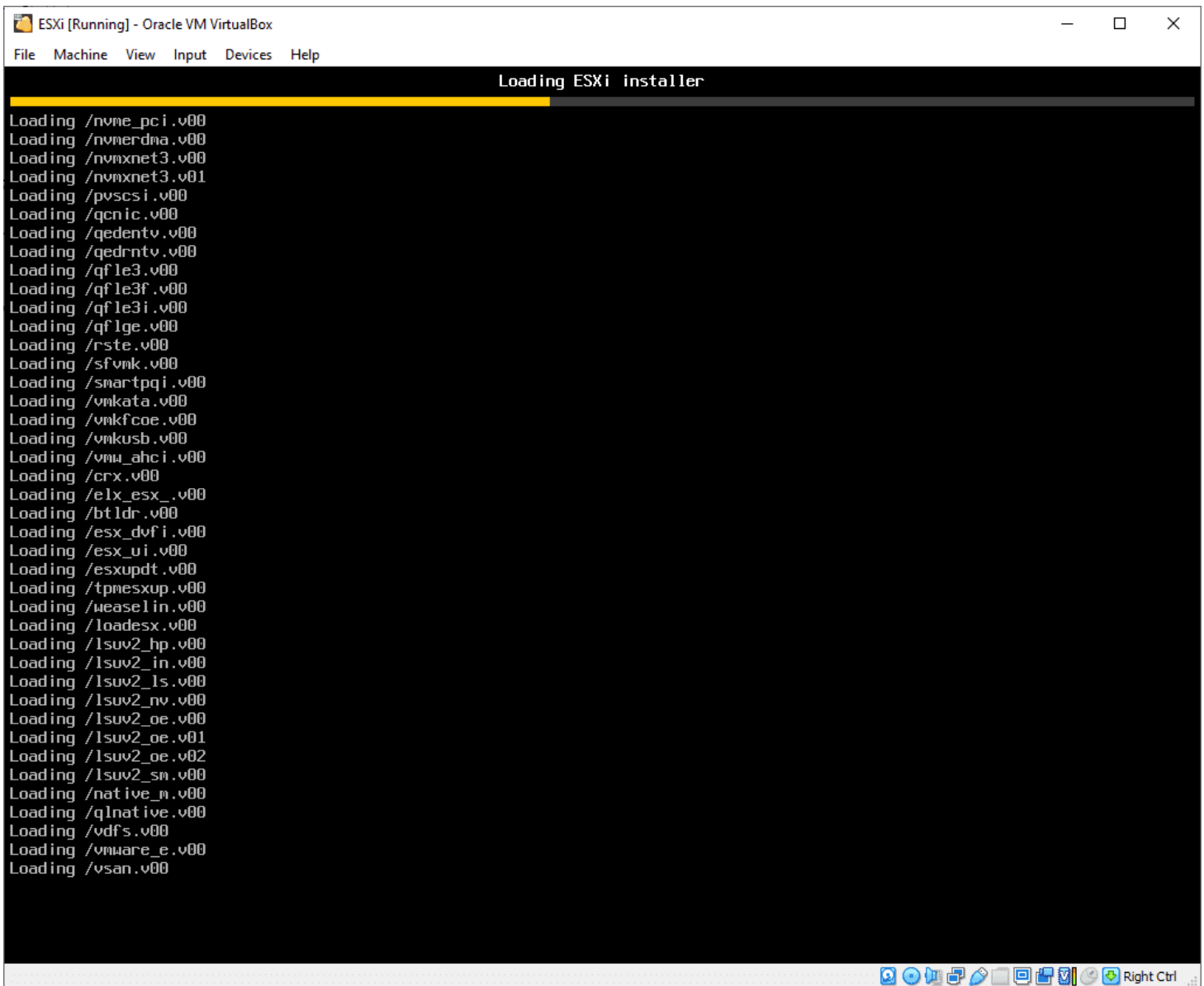
Choose optical drive for adding the esxi iso

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](#)

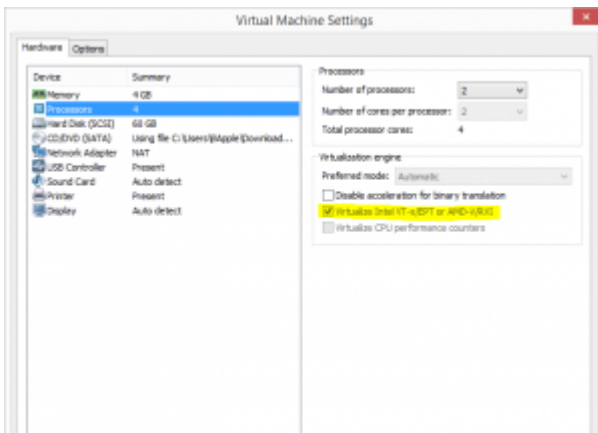
```
-----@ubuntu:~$ sudo apt-get install cpu-checker
[sudo] password for zerouser:
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following extra packages will be installed:
  msr-tools
The following NEW packages will be installed:
  cpu-checker msr-tools
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.
  tu5 [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) ...
```

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
```

```
zerouser@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 1.3-2ubuntu5 [6,868 B]
Get:2 http://us.archive.ubuntu.com/ubuntu/ wily/main cpu-checker amd64 0.7-0ubuntu5 [10,632 B]
Fetched 17.5 kB in 0s (85.0 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:

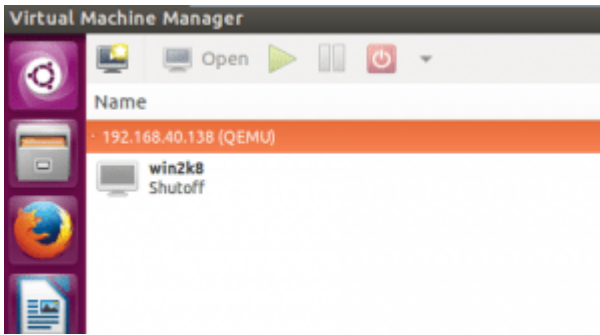
To manage my virtual machines, I used another Ubuntu 15.10 desktop machine to install the **virt-manager** package on. To install virt-manager, issue this command:

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

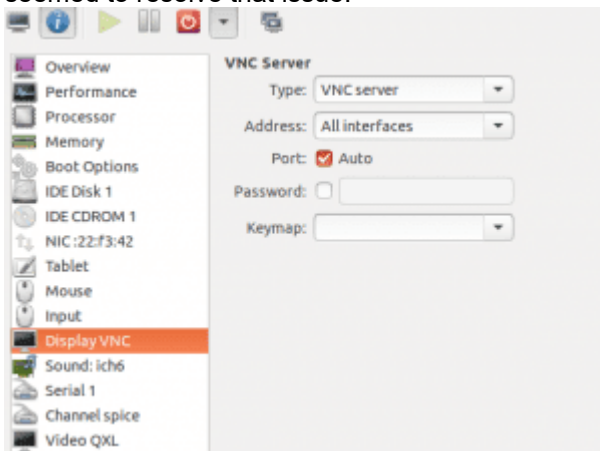
Now you are ready to start playing around with creating some Virtual machines.

Windows VM

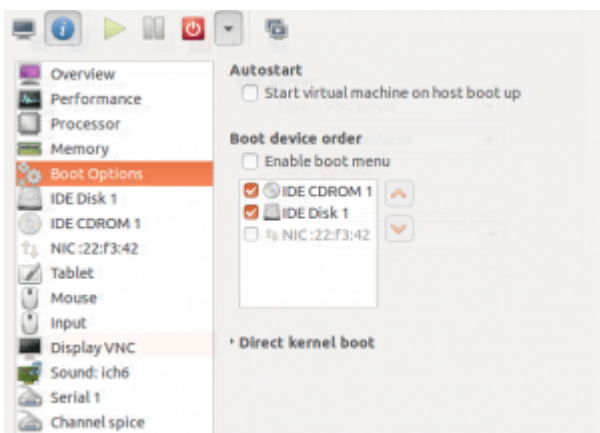
I played around with a Windows 2012 R2 VM as the first VM that I created in the KVM environment. This was fairly straightforward. After you launch Virtual Machine Manager, just click the button at the top left to “create a new virtual machine” and follow the prompts to assign storage, networking, and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



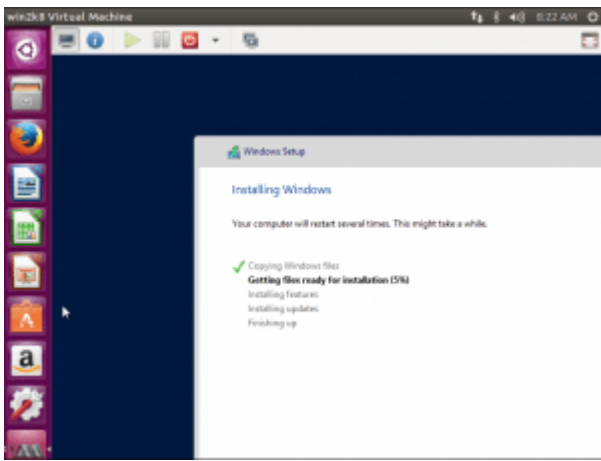
I had an issue right at first where I couldn't see the console. I changed the Console display type to VNC Server and this seemed to resolve that issue.



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.



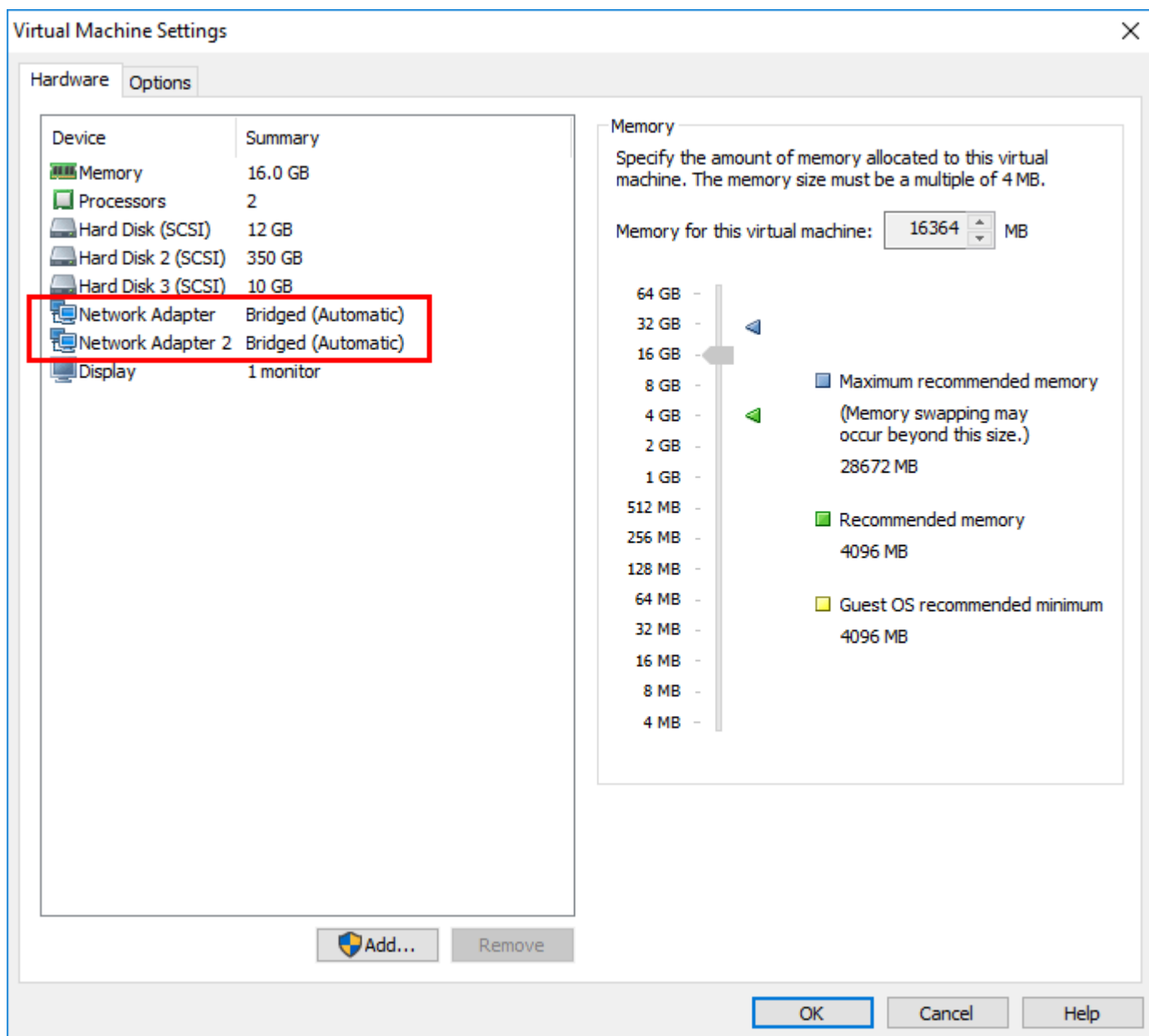
Final Thoughts

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.

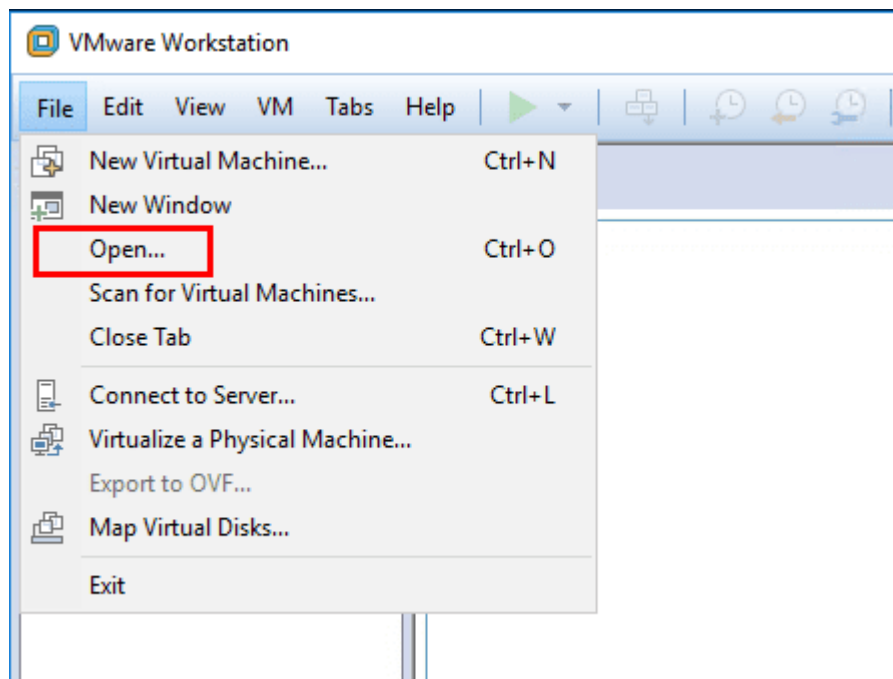
I recommend reading the following Cormac Hogan's resources on VMware vSAN Witness Appliance:

- [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

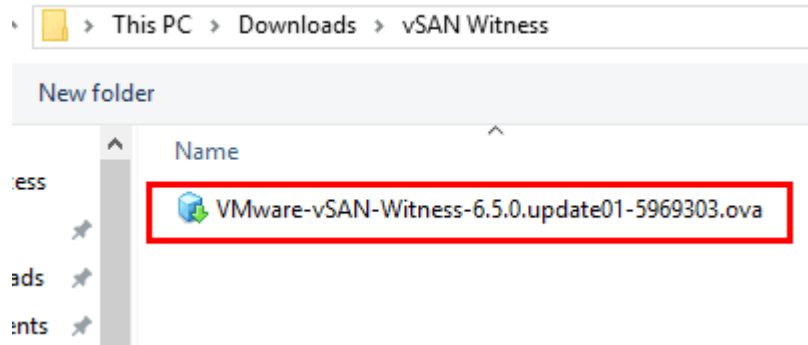
In my home lab environment, I have a standalone server outside of my 2-node ESXi cluster that serves other purposes for the lab environment. I wanted to utilize this server for the vSAN Witness Appliance version 6.5 U1 by using VMware Workstation 14. The steps to get up and running with the VMware vSAN 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.



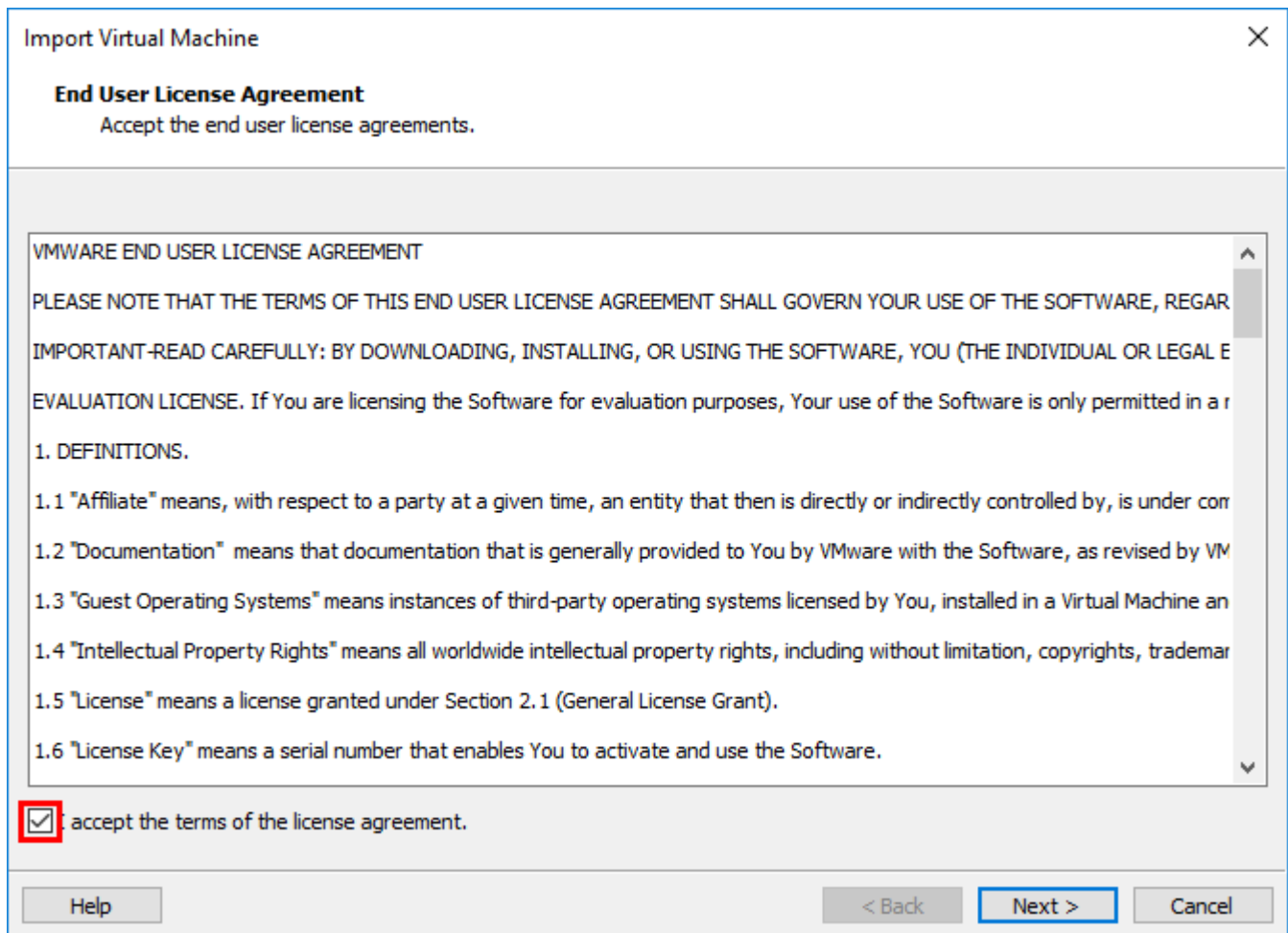
Open the VMware vSAN Witness Appliance in VMware Workstation

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.

Import Virtual Machine ×

Store the new Virtual Machine
Provide a name and local storage path for the new virtual machine.

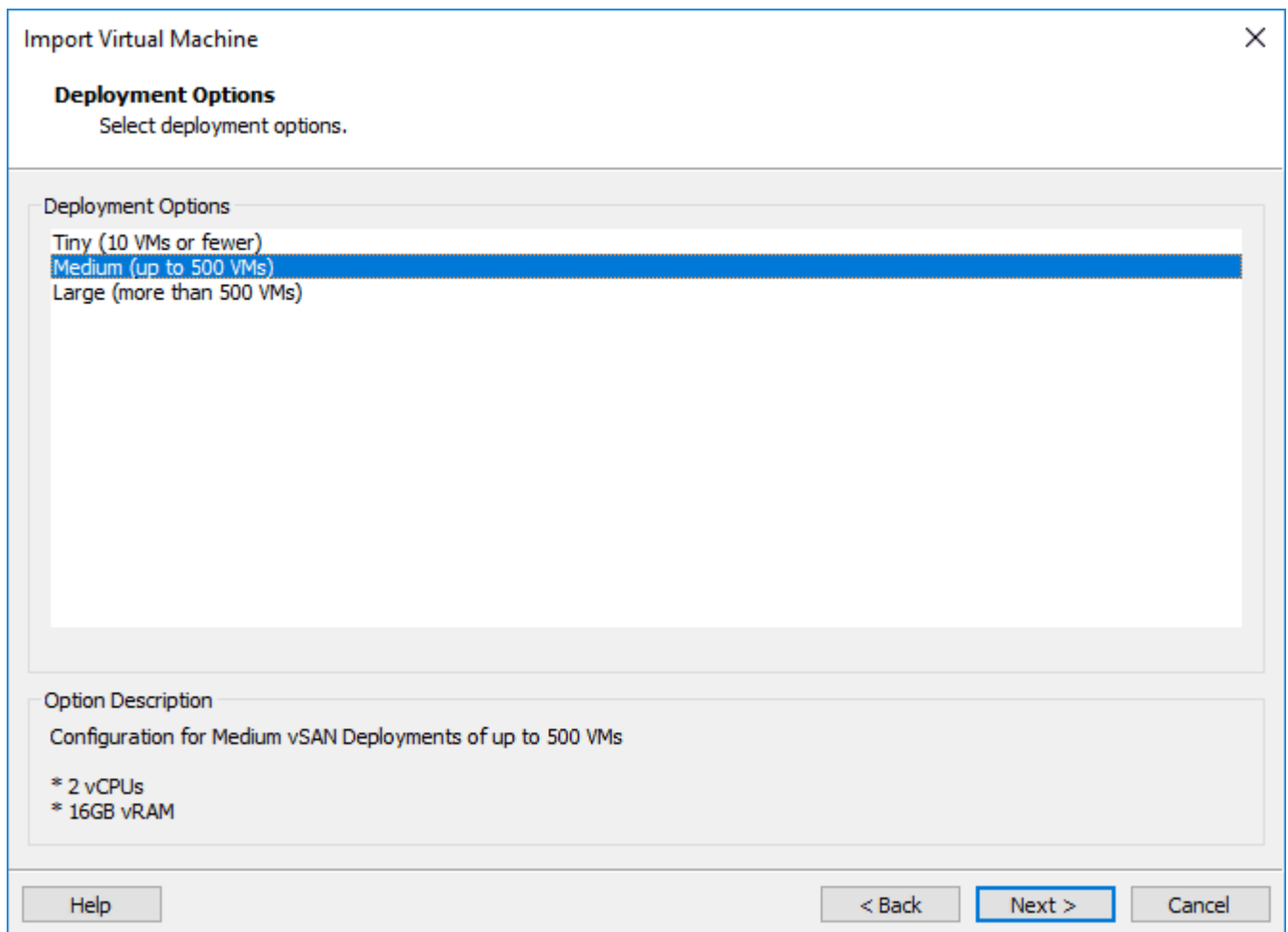
Name for the new virtual machine:

Storage path for the new virtual machine:
 Browse...

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

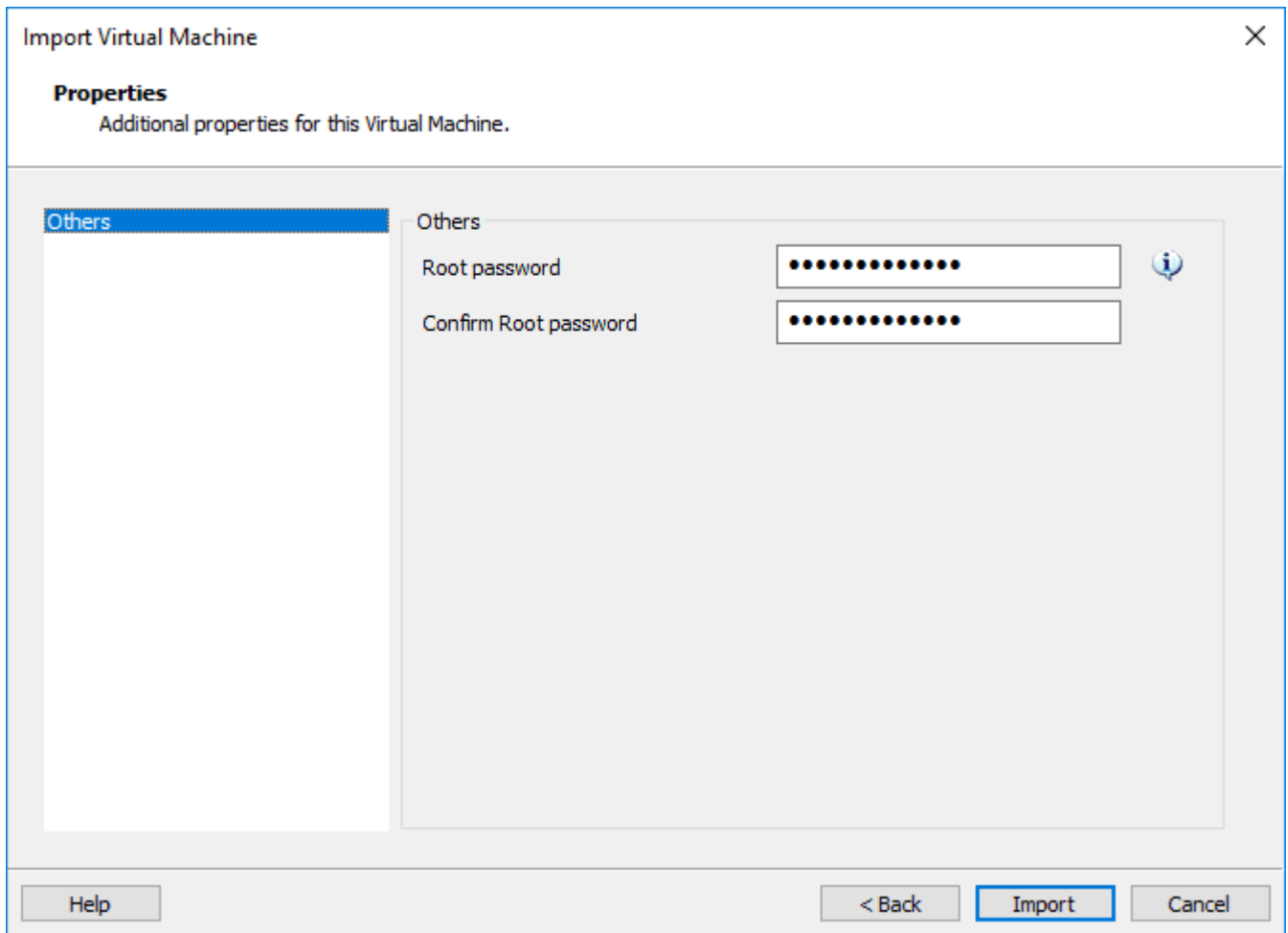
By default, on the **Deployment Options** screen, you will see the sizing options for the appliance. A brief sizing guideline is given beside each sizing option by way of “number of VMs”. A further breakdown of the sizing options is as follows:

- **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



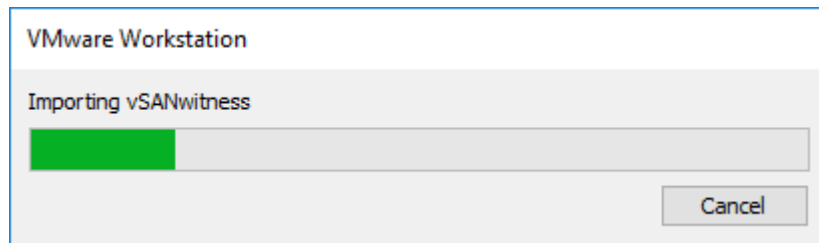
Choose the VMware vSAN Witness Appliance sizing in deployment options

Next, we set the root password for the appliance.



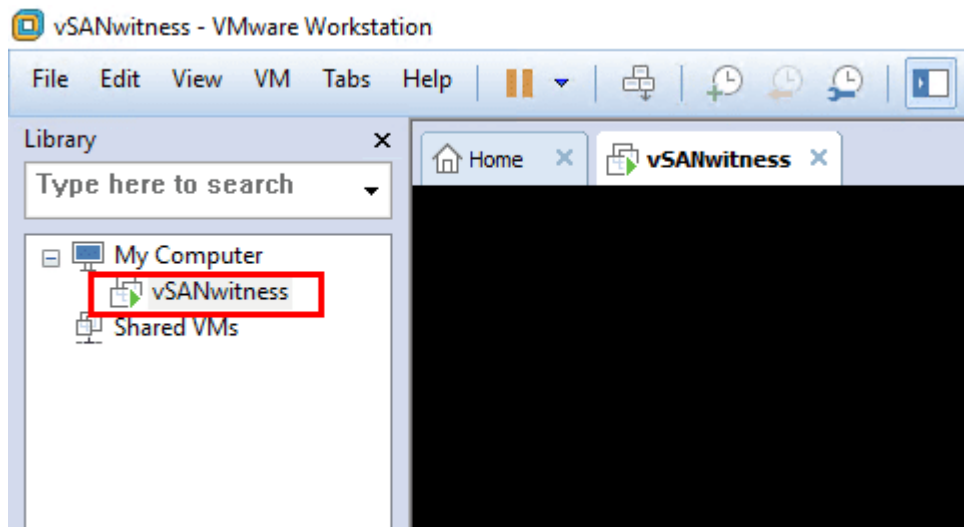
Set the VMware vSAN Witness Appliance root password

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



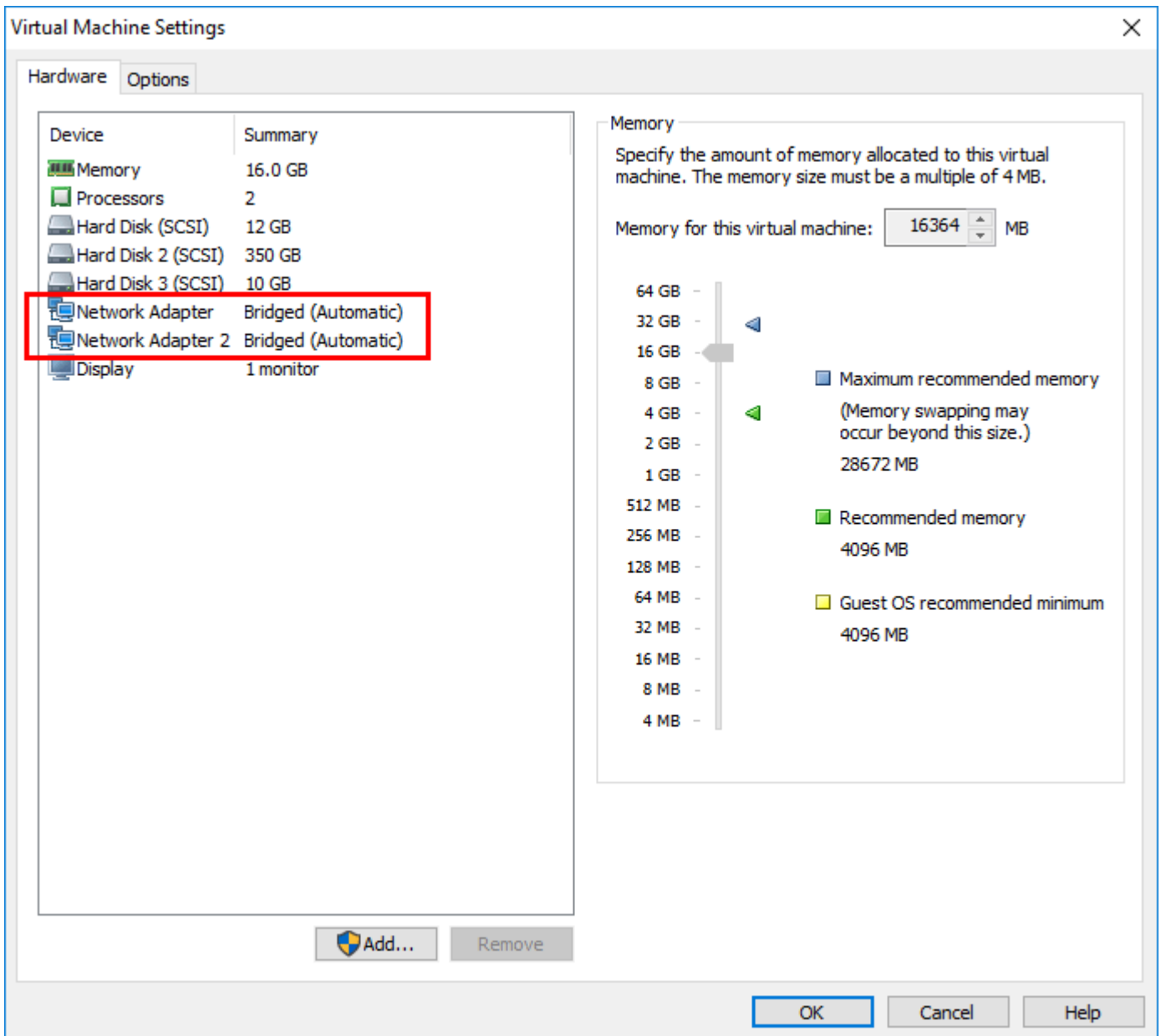
The import of the VMware vSAN Witness Appliance OVA appliance into VMware Workstation begins

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



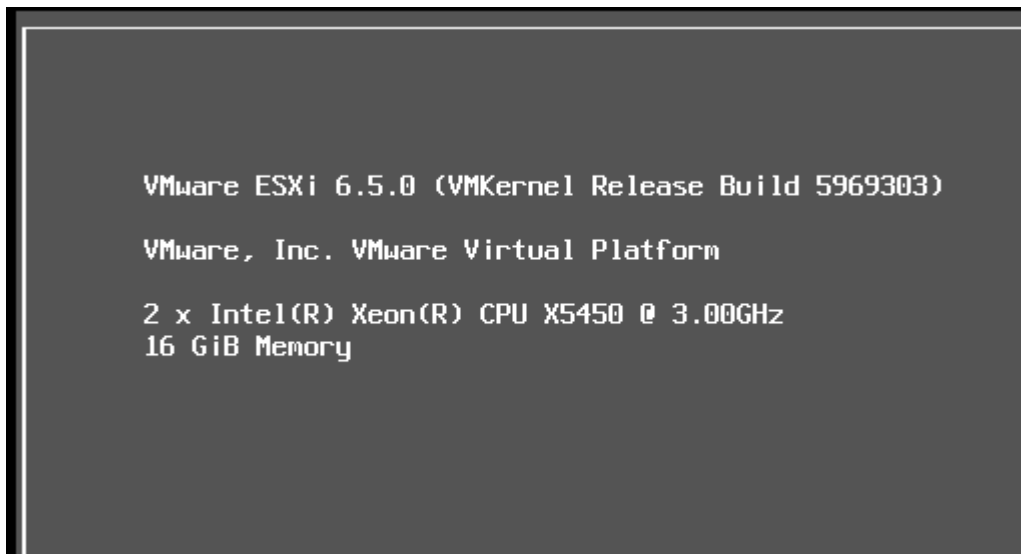
After importing the VMware vSAN Witness Appliance we power it on

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-vsant-6-1-witness-virtual-appliance-on-vmware-fusion-workstation.html>

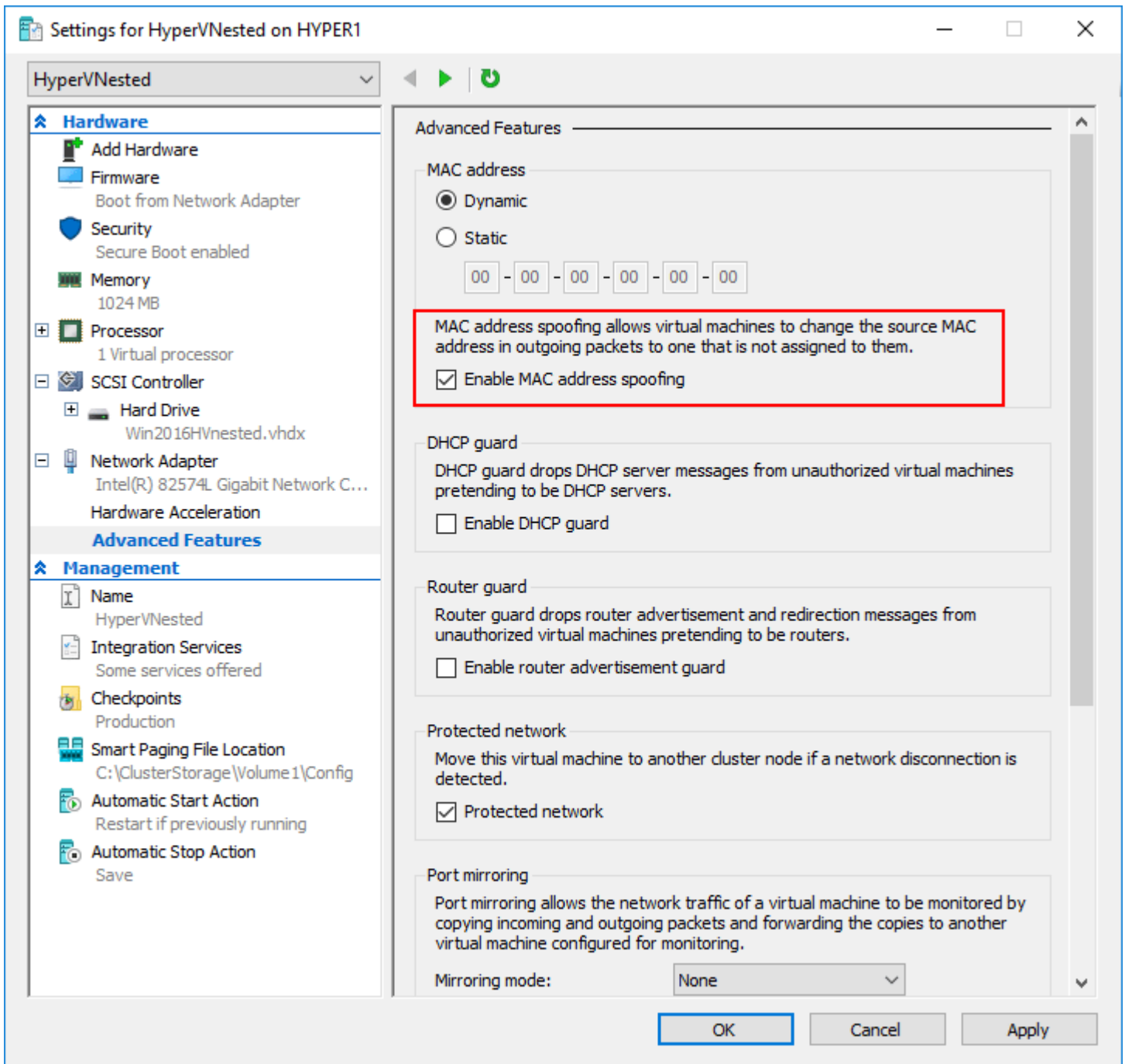
Thoughts

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

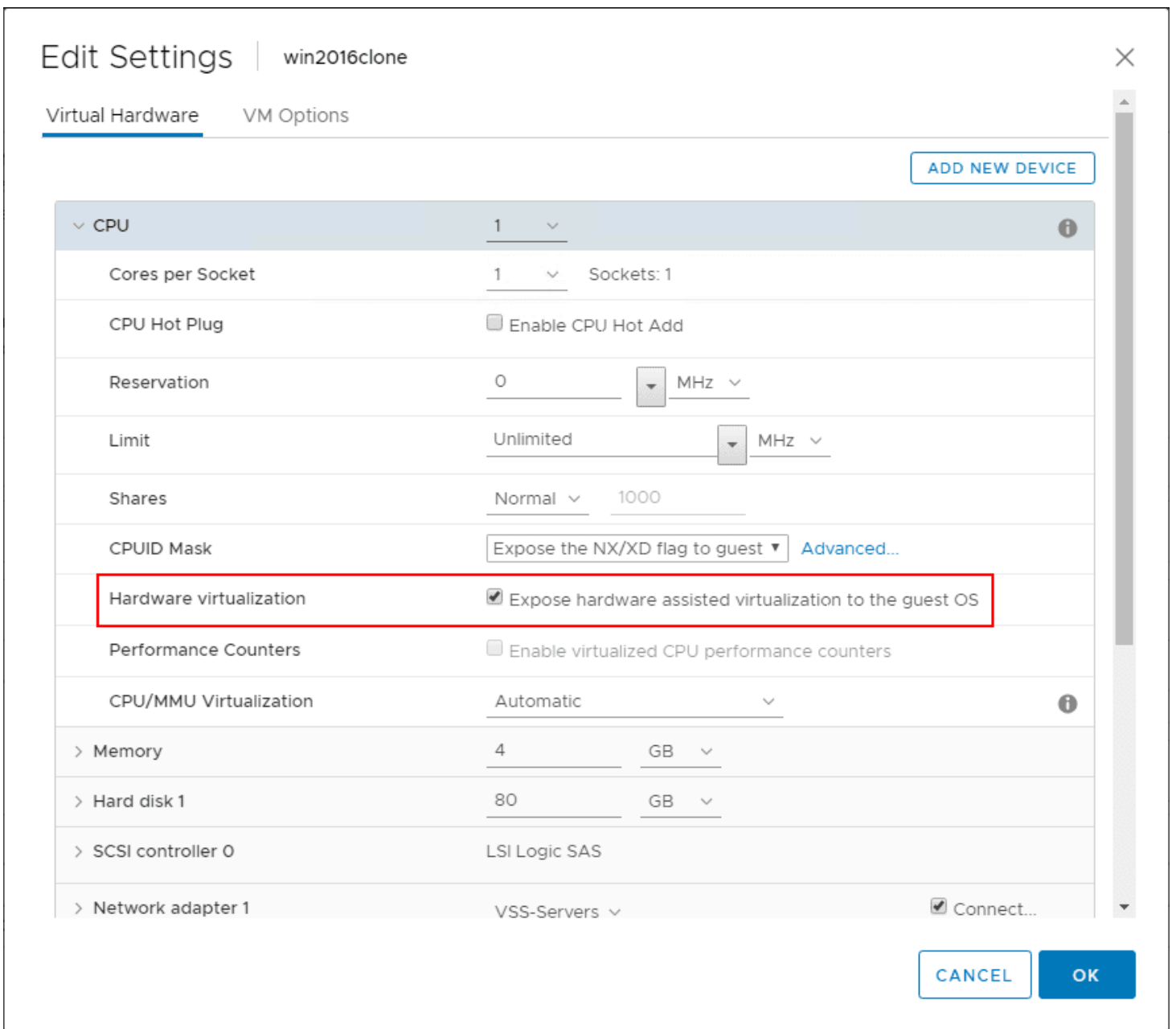
VMware has certainly been in the nested virtualization game for much longer than Hyper-V as Microsoft has only now released nested virtualization as a “thing” in [Windows Server 2016](#) with Hyper-V. There are certainly similarities between the VMware and Hyper-V implementation of nested virtualization 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](#))

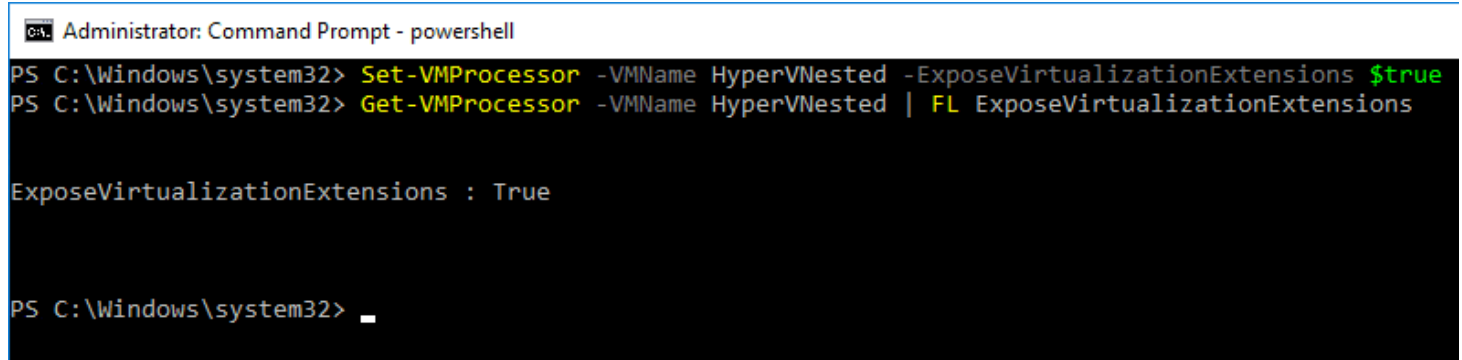
```
$vmName = 'MyVM'  
$vm = Get-VM -Name $vmName  
  
$spec = New-Object VMware.Vim.VirtualMachineConfigSpec  
$spec.nestedHVEEnabled = $true  
  
$vm.ExtensionData.ReconfigVM($spec)
```

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



```
Administrator: Command Prompt - powershell
PS C:\Windows\system32> Set-VMProcessor -VMName HyperVNested -ExposeVirtualizationExtensions $true
PS C:\Windows\system32> Get-VMProcessor -VMName HyperVNested | FL ExposeVirtualizationExtensions

ExposeVirtualizationExtensions : True

PS C:\Windows\system32> _
```

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 packets 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**.

vSwitch0 - Edit Settings

Properties

Security

Promiscuous mode Accept ▼

MAC address changes Accept ▼

Traffic shaping

Forged transmits Accept ▼

Teaming and failover

CANCEL

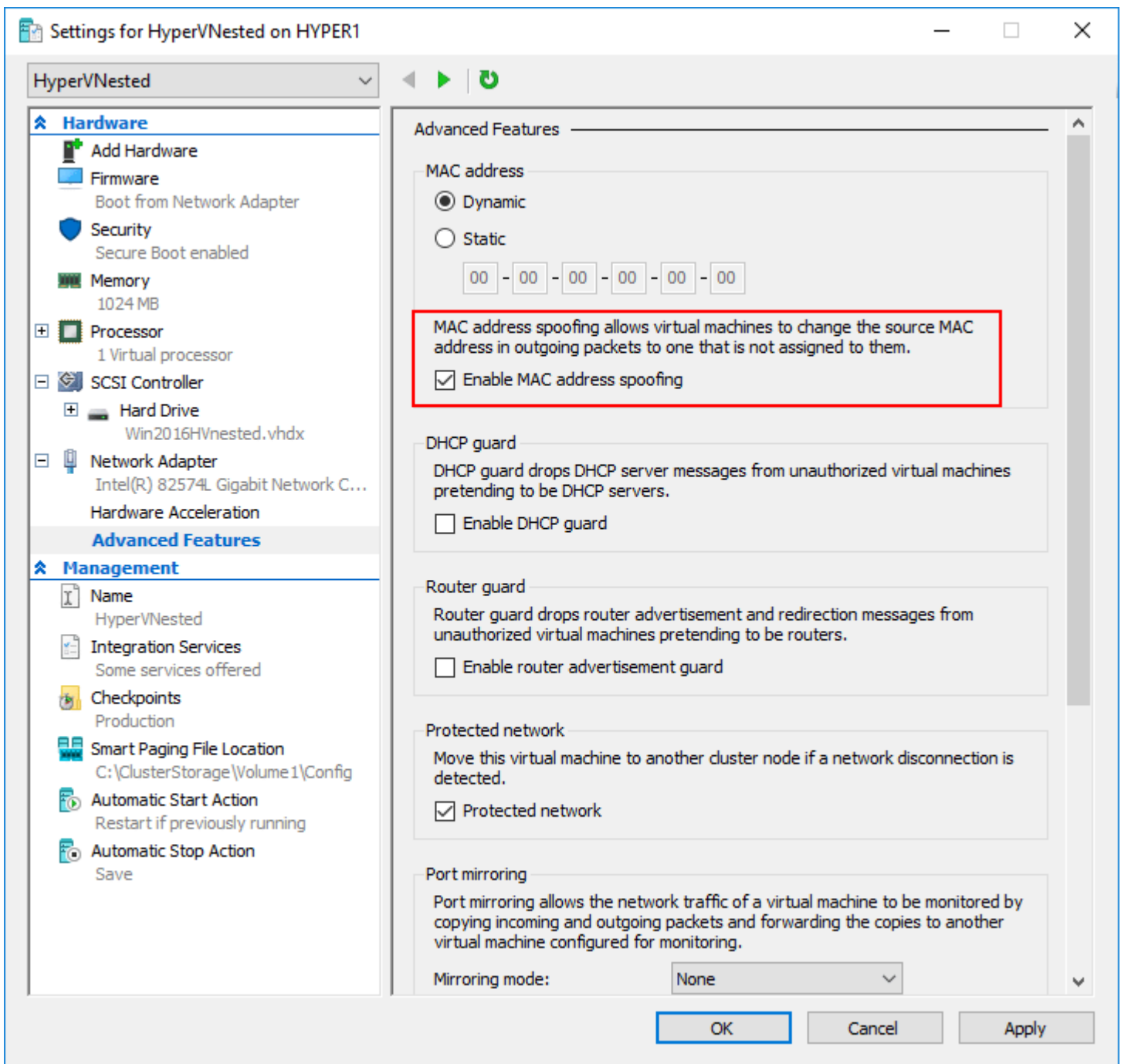
OK

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 virtualization 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](#)

```
Nutanix Community Edition AHV (4.4.77-1.el7.nutanix.20190211.279.x86_64) 7 (Core)
Nutanix Community Edition AHV (0-rescue-e7d17aaf2faf440ca6af4b49163191a8) 7 (Core)
```

The Nutanix logo is displayed in the center of the screen. It consists of the word "NUTANIX" in a bold, white, sans-serif font. The letter "X" is stylized with a green diagonal bar crossing it from the bottom-left to the top-right.

```
Use the ↑ and ↓ keys to change the selection.
Press 'e' to edit the selected item, or 'c' for a command prompt.
The selected entry will be started automatically in 3s.
```

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 visiting 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.

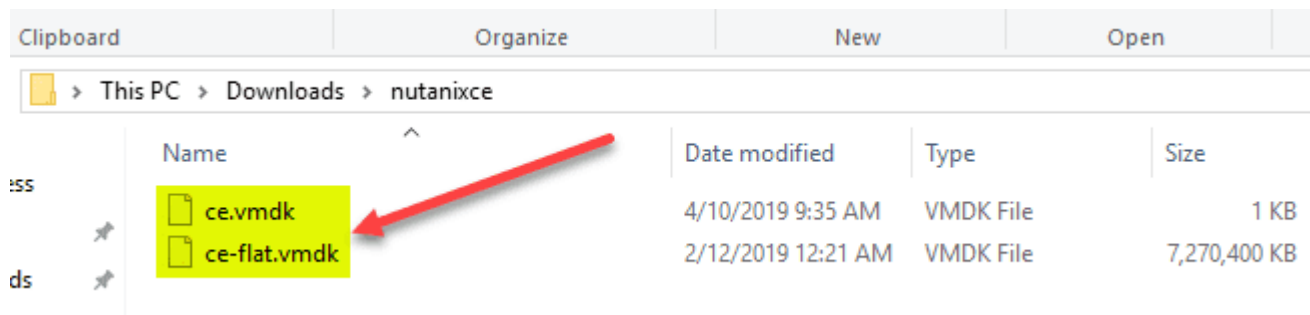
Getting the Nutanix VMDK Files Ready

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.



The screenshot shows a Windows File Explorer window with the address bar set to 'This PC > Downloads > nutanixce'. The main pane displays a table of files:

Name	Date modified	Type	Size
ce.vmdk	4/10/2019 9:35 AM	VMDK File	1 KB
ce-flat.vmdk	2/12/2019 12:21 AM	VMDK File	7,270,400 KB

A red arrow points to the 'ce-flat.vmdk' file, and both files are highlighted in yellow.

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.

New Virtual Machine

- ✓ 1 Select a creation type
- ✓ 2 Select a name and folder
- ✓ 3 Select a compute resource
- ✓ 4 Select storage
- ✓ 5 Select compatibility
- 6 Select a guest OS
- 7 Customize hardware
- 8 Ready to complete

Select a creation type

How would you like to create a virtual machine?

- Create a new virtual machine
- Deploy from template
- Clone an existing virtual machine
- Clone virtual machine to template
- Clone template to template
- Convert template to virtual machine

This option guides you through creating a new virtual machine. You will be able to customize processors, memory, network connections, and storage. You will need to install a guest operating system after creation.

CANCEL

BACK

NEXT

Creating a new VMware ESXi virtual machine for installing Nutanix CE

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

New Virtual Machine

- ✓ 1 Select a creation type
- ✓ **2 Select a name and folder**
- ✓ 3 Select a compute resource
- ✓ 4 Select storage
- ✓ 5 Select compatibility
- 6 Select a guest OS
- 7 Customize hardware
- 8 Ready to complete

Select a name and folder

Specify a unique name and target location

Virtual machine name: NCE

Select a location for the virtual machine.

- ✓ vcsa.cloud.local
 - > CloudLocal

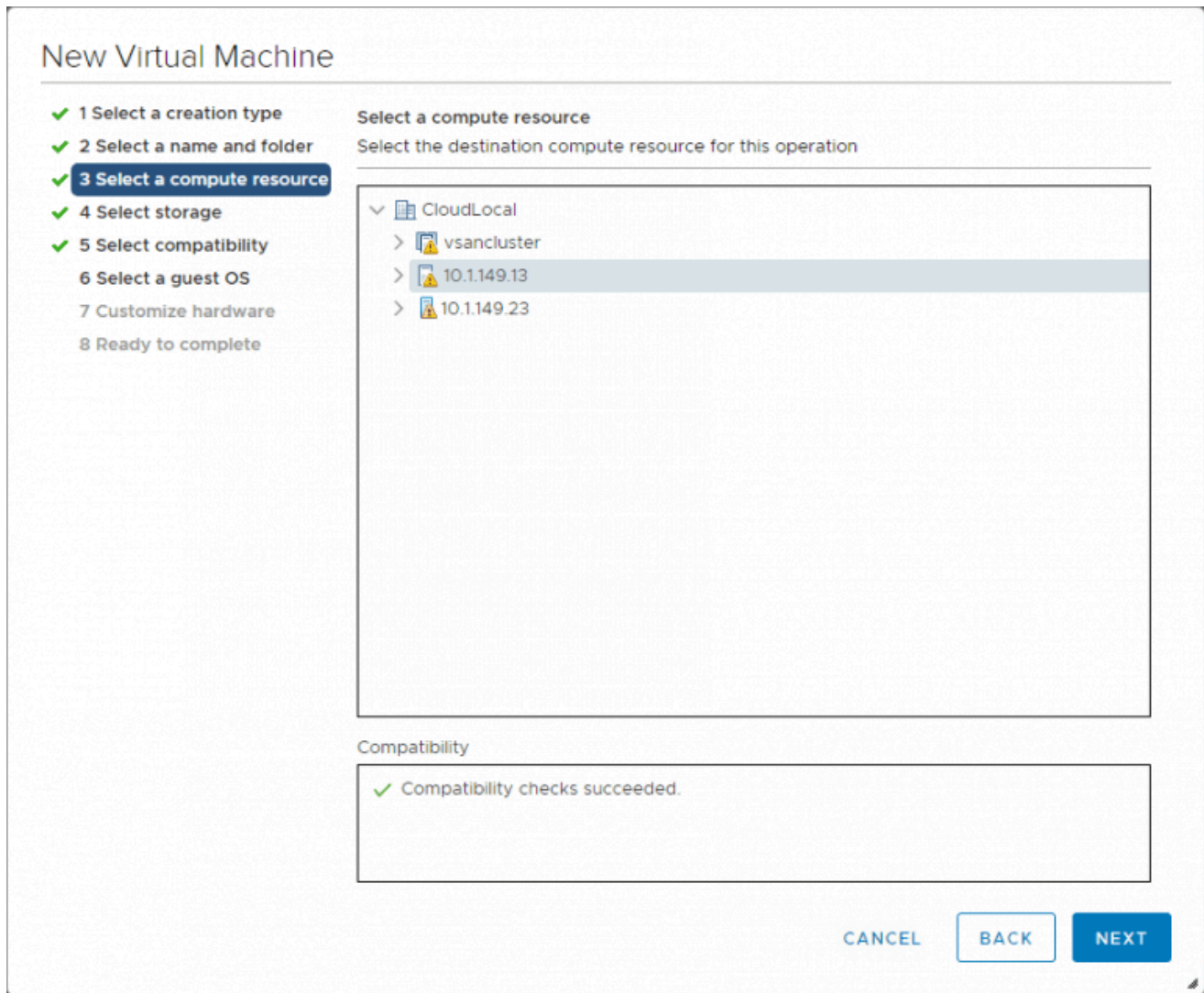
CANCEL

BACK

NEXT

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.

New Virtual Machine

- ✓ 1 Select a creation type
- ✓ 2 Select a name and folder
- ✓ 3 Select a compute resource
- ✓ 4 Select storage
- ✓ 5 Select compatibility
- 6 Select a guest OS
- 7 Customize hardware
- 8 Ready to complete

Select storage

Select the storage for the configuration and disk files

Encrypt this virtual machine

VM Storage Policy: Datastore Default ▾

Name	Capacity	Provisioned	Free	Type
ESX3DS01	931.25 GB	504.64 GB	797.36 GB	VM
ESX3DS02	5.46 TB	3 TB	2.46 TB	VM
ESX3DS03	3.64 TB	3 TB	652.53 GB	VM

Compatibility

✓ Compatibility checks succeeded.

CANCEL

BACK

NEXT

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.

New Virtual Machine

- ✓ 1 Select a creation type
- ✓ 2 Select a name and folder
- ✓ 3 Select a compute resource
- ✓ 4 Select storage
- ✓ 5 Select compatibility
- 6 Select a guest OS
- 7 Customize hardware
- 8 Ready to complete

Select compatibility

Select compatibility for this virtual machine depending on the hosts in your environment

The host or cluster supports more than one VMware virtual machine version. Select a compatibility for the virtual machine.

Compatible with: ⓘ

This virtual machine uses hardware version 11, which is compatible with ESXi 6.0 and later. Some virtual machine hardware features are unavailable with this option.

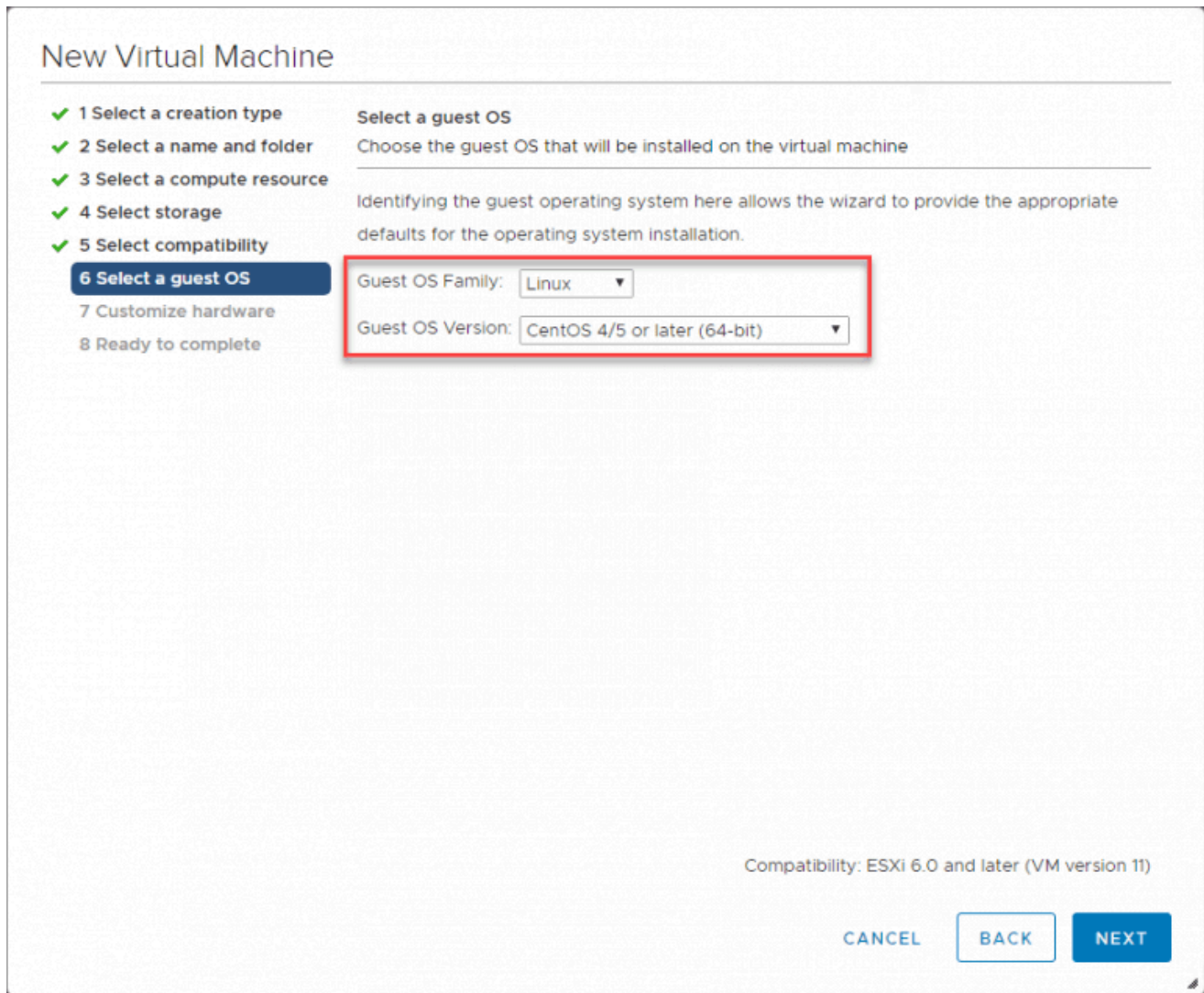
CANCEL

BACK

NEXT

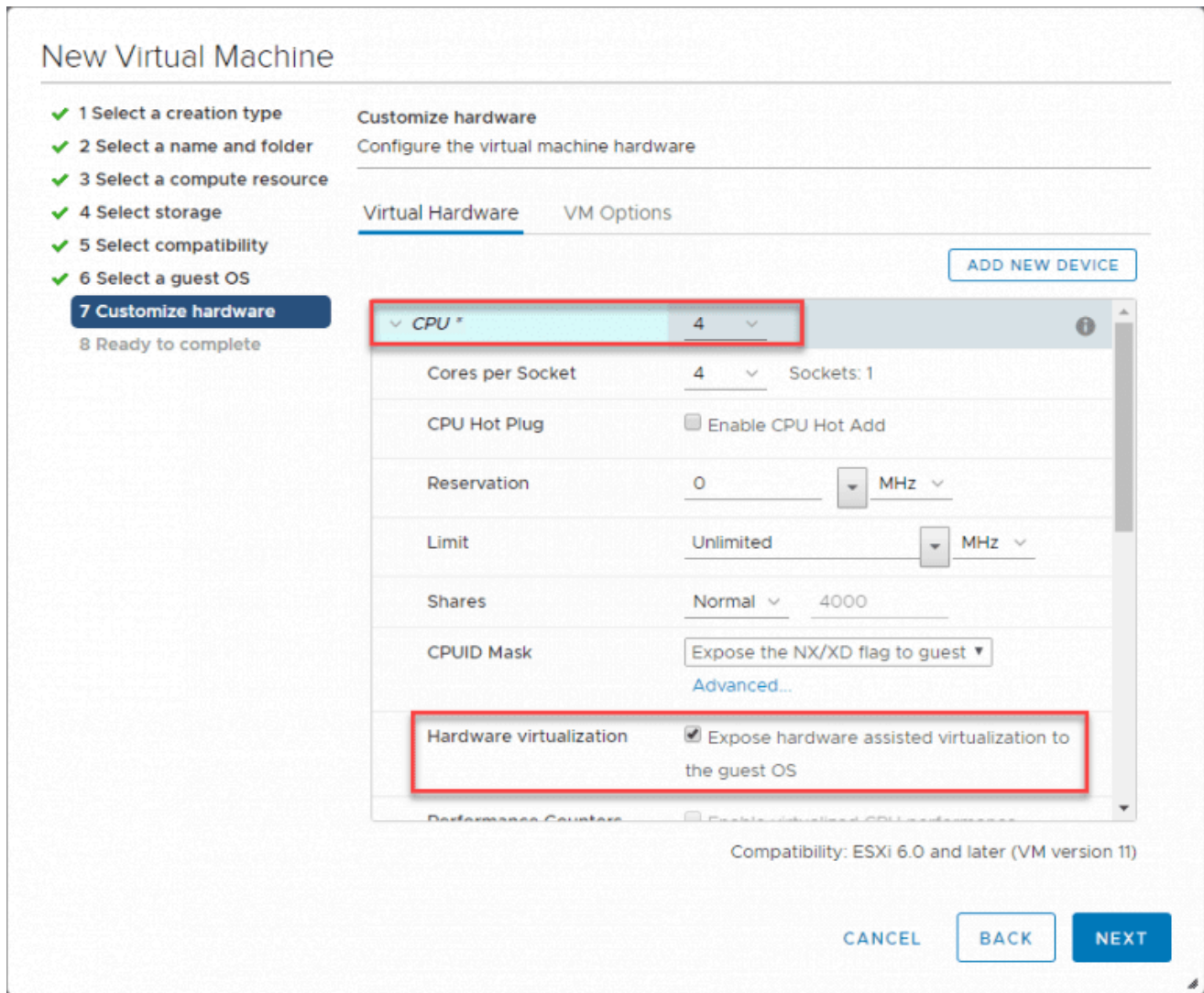
Select the compatibility mode for the Nutanix CE nested installation

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.

New Virtual Machine

- ✓ 1 Select a creation type
- ✓ 2 Select a name and folder
- ✓ 3 Select a compute resource
- ✓ 4 Select storage
- ✓ 5 Select compatibility
- ✓ 6 Select a guest OS
- 7 Customize hardware**
- 8 Ready to complete

Customize hardware
Configure the virtual machine hardware

Virtual Hardware VM Options

ADD NEW DEVICE

> CPU *	4	▼	
> Memory *	24		GB ▼
> New Hard disk *	500		GB ▼
▼ New Hard disk *	200		GB ▼
Maximum Size	797.36 GB		
VM storage policy	Datastore Default ▼		
Location	Store with the virtual machine ▼		
Disk Provisioning	Thin Provision ▼		
Sharing	Unspecified ▼		
Shares	Normal ▼	1000	

Compatibility: ESXi 6.0 and later (VM version 11)

CANCEL

BACK

NEXT

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

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

New Virtual Machine

- ✓ 1 Select a creation type
- ✓ 2 Select a name and folder
- ✓ 3 Select a compute resource
- ✓ 4 Select storage
- ✓ 5 Select compatibility
- ✓ 6 Select a guest OS
- 7 Customize hardware**
- 8 Ready to complete

Customize hardware

Configure the virtual machine hardware

Virtual Hardware VM Options

ADD NEW DEVICE

> CPU *	4	▼	
> Memory *	24	GB	▼
> New Hard disk *	500	GB	▼
> New Hard disk *	200	GB	▼
> New SCSI controller *	LSI Logic Parallel		
▼ New Network *	VSS-Servers ▼		
Status	<input checked="" type="checkbox"/> Connect At Power On		
Adapter Type	E1000 ▼		
MAC Address		Automatic ▼	
> New CD/DVD Drive *	Client Device	▼	<input type="checkbox"/> Connect...

Compatibility: ESXi 6.0 and later (VM version 11)

CANCEL

BACK

NEXT

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.

New Virtual Machine

- ✓ 1 Select a creation type
- ✓ 2 Select a name and folder
- ✓ 3 Select a compute resource
- ✓ 4 Select storage
- ✓ 5 Select compatibility
- ✓ 6 Select a guest OS
- 7 Customize hardware**
- 8 Ready to complete

Customize hardware

Configure the virtual machine hardware

Virtual Hardware VM Options

ADD NEW DEVICE

▼ New Network	v33-Servers
Status	<input checked="" type="checkbox"/> Connect At Power On
Adapter Type	E1000
MAC Address	Automatic
> New CD/DVD Drive *	Client Device <input type="checkbox"/> Connect...
> Video card *	Specify custom settings
VMCI device	Device on the virtual machine PCI bus that provides support for the virtual machine communication interface
New SATA Controller	New SATA Controller
> Other	Additional Hardware

Compatibility: ESXi 6.0 and later (VM version 11)

CANCEL

BACK

NEXT

Finish out the wizard.

New Virtual Machine

- ✓ 1 Select a creation type
- ✓ 2 Select a name and folder
- ✓ 3 Select a compute resource
- ✓ 4 Select storage
- ✓ 5 Select compatibility
- ✓ 6 Select a guest OS
- ✓ 7 Customize hardware
- 8 Ready to complete**

Ready to complete

Click Finish to start creation.

Provisioning type	Create a new virtual machine
Virtual machine name	NCE
Folder	CloudLocal
Host	10.1.149.13
Datastore	ESX3DS01
Guest OS name	CentOS 4/5 or later (64-bit)
Virtualization Based Security	Disabled
CPUs	4
Memory	24 GB
NICs	1
NIC 1 network	VSS-Servers
NIC 1 type	E1000
SCSI controller 1	LSI Logic Parallel

Compatibility: ESXi 6.0 and later (VM version 11)

CANCEL

BACK

FINISH

Finishing the initial VM creation wizard for Nutanix CE

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

Edit Settings | NCE

Virtual Hardware | VM Options

ADD NEW DEVICE

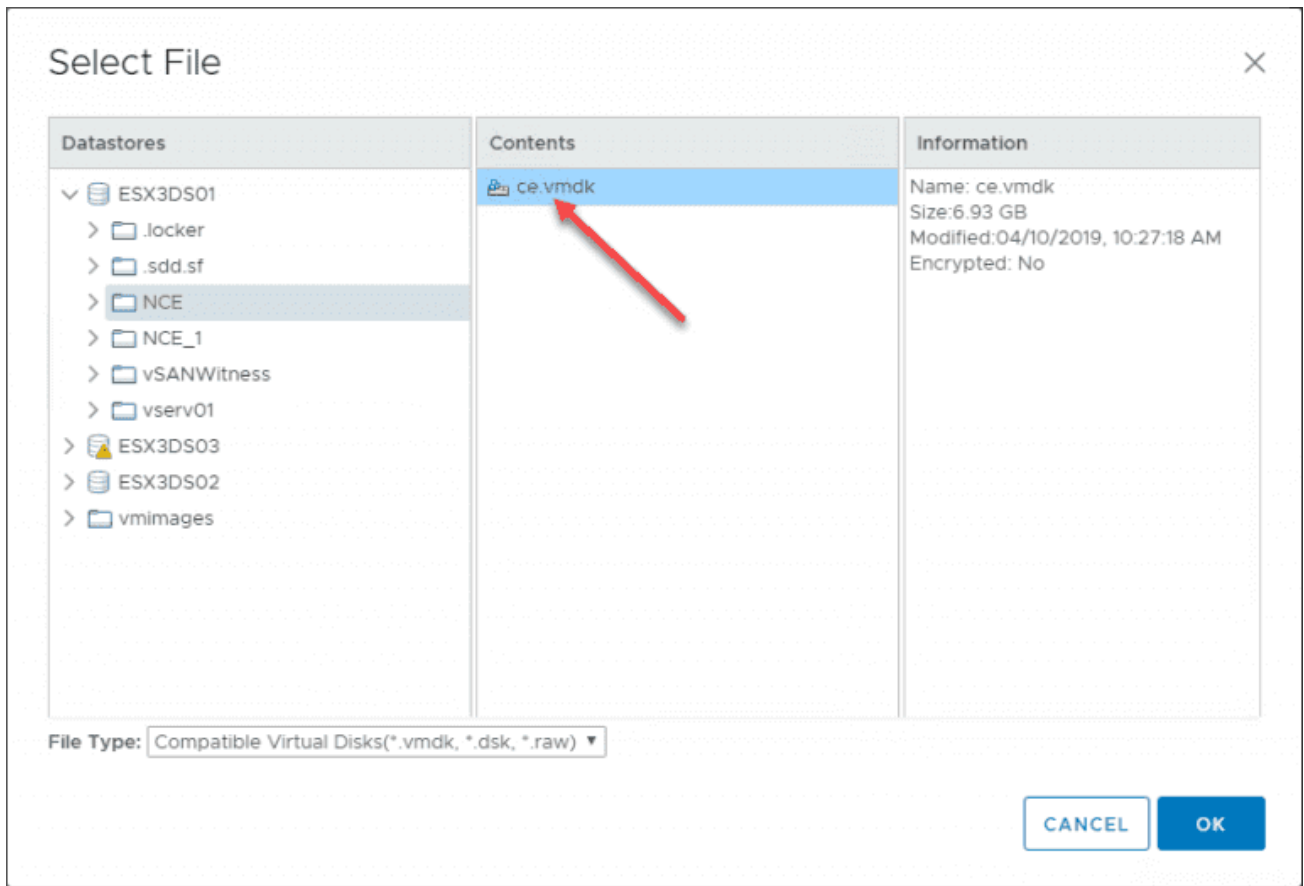
> CPU	4	▼
> Memory	24	GB ▼
> Hard disk 1	500	GB ▼
> Hard disk 2	200	GB ▼
> SCSI controller 0	LSI Logic Parallel	
> Network adapter 1	VSS-Servers ▼	
> CD/DVD drive 1	Client Device ▼	
> Video card	Specify custom settings ▼	
VMCI device	Device on the virtual machine PCI bus that provides support for the virtual machine communication interface	
SATA controller 0	AHCI	
> Other	Additional Hardware	

- CD/DVD Drive
- Host USB Device
- Hard Disk
- RDM Disk
- Existing Hard Disk
- Network Adapter
- SCSI Controller
- USB Controller
- SATA Controller
- NVMe Controller
- Shared PCI Device
- PCI Device
- Serial Port

CANCEL OK

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.

Edit Settings | NCE

Virtual Hardware | VM Options

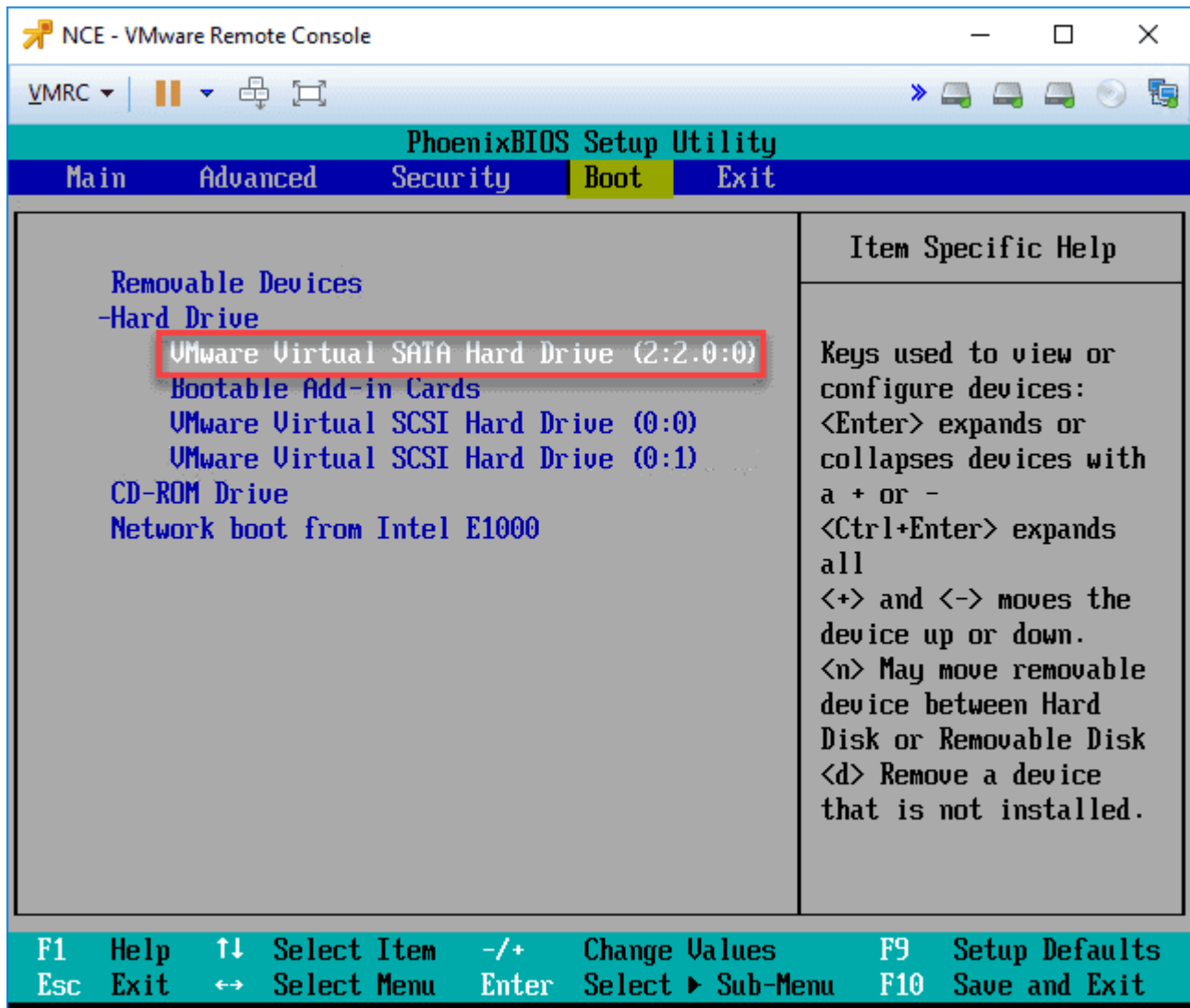
ADD NEW DEVICE

Sharing	Unspecified	
Disk File	[ESX3DS01] NCE/ce.vmdk	
Shares	Normal	1000
Limit - IOPs	Unlimited	
Virtual flash read cache	0	MB
Disk Mode	Dependent	
Virtual Device Node	SATA controller 0	SATA(0:0) New Hard disk
> SCSI controller 0	LSI Logic Parallel	
> Network adapter 1	VSS-Servers	<input checked="" type="checkbox"/> Connect...
> CD/DVD drive 1	Client Device	<input type="checkbox"/> Connect...
> Video card	Specify custom settings	
VMCI device	Device on the virtual machine PCI bus that provides support for the virtual machine communication interface	

CANCEL OK

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.

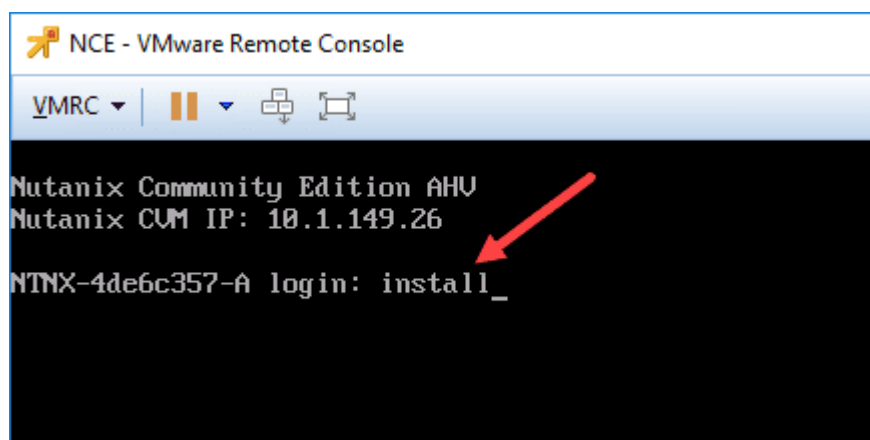
```
Nutanix Community Edition AHV (4.4.77-1.el7.nutanix.20190211.279.x86_64) 7 (Core)
Nutanix Community Edition AHV (0-rescue-e7d17aaf2faf440ca6af4b49163191a8) 7 (Core)
```

NUTANIX

Use the ↑ and ↓ keys to change the selection.
Press 'e' to edit the selected item, or 'c' for a command prompt.
The selected entry will be started automatically in 3s.

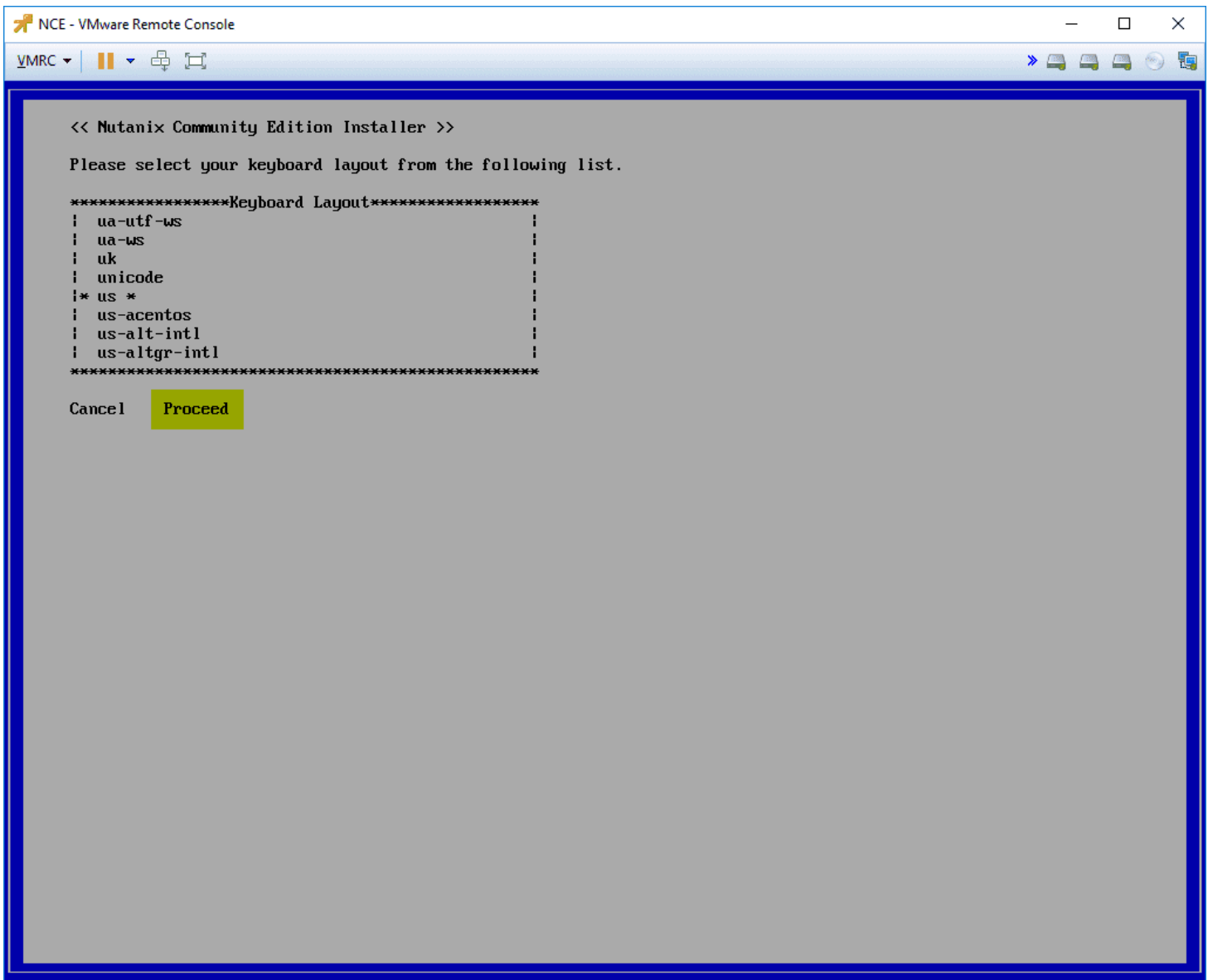
Booting the Nutanix CE VM to begin the installation

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



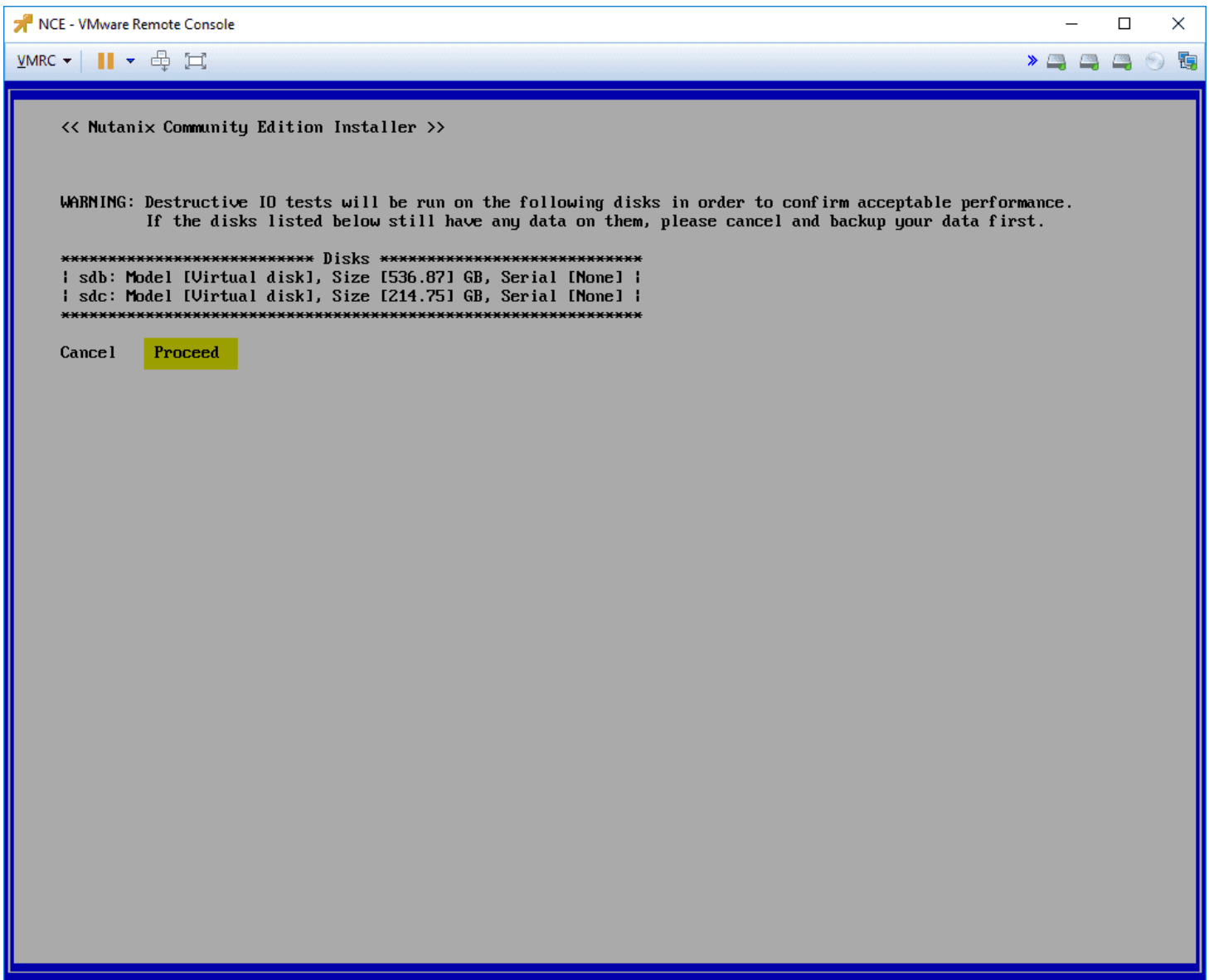
Login as the install user to begin installation

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



Nutanix Community Edition installer begins

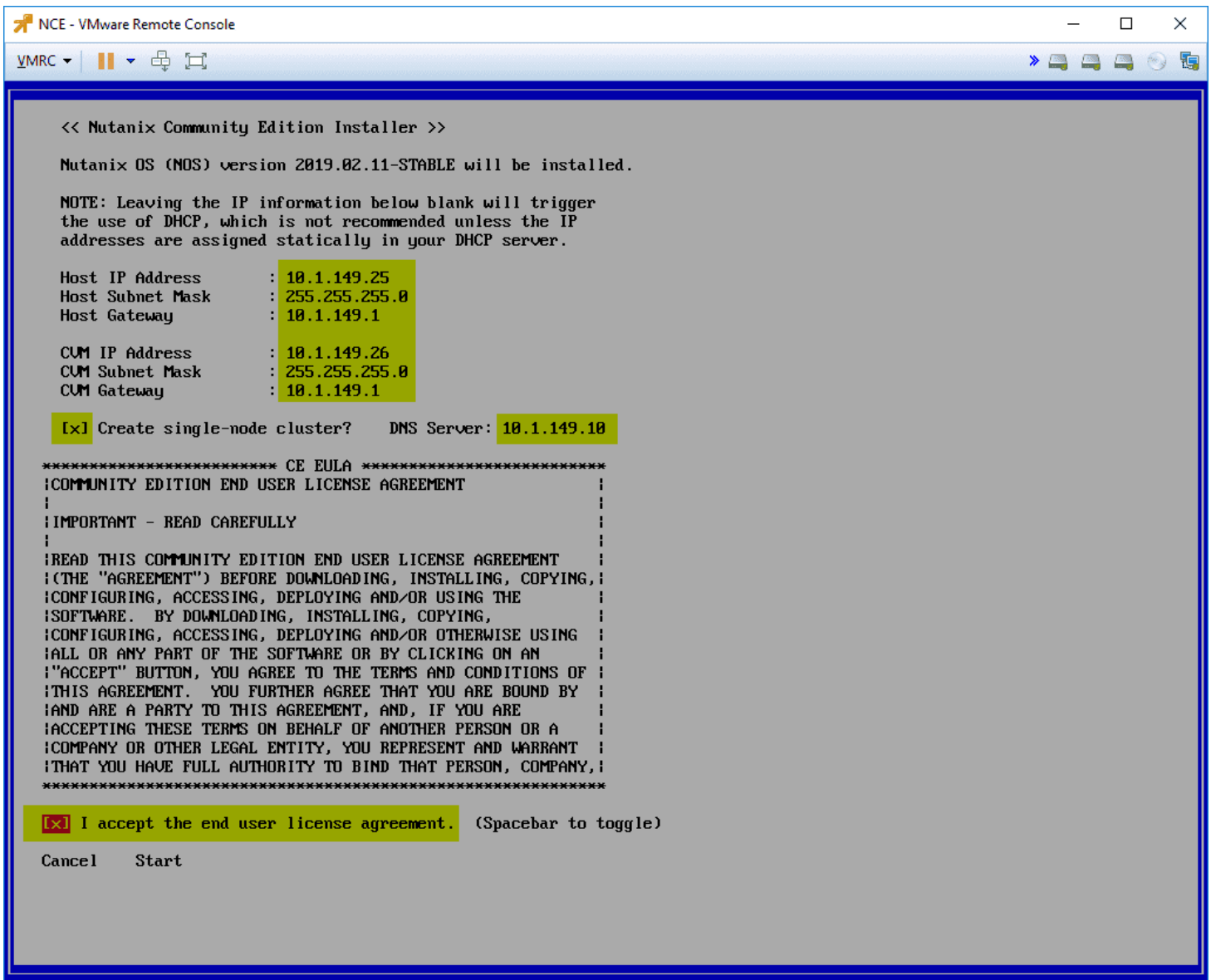
Next, you will be asked to confirm the destructive [IO tests](#) on your hard disks.



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.

```
NCE - VMware Remote Console
VMRC | 100%
Hypervisor : [
SUM (% n/a): /
Total imaging time: 00:00:19
install/lib/sg3utils/bin/sg_read_long
install/lib/sg3utils/bin/sg_referrals
install/lib/sreinstall.sh
install/lib/intel_ssd/
install/lib/intel_ssd/isdct
install/lib/storcli/
install/lib/storcli/storcli
install/lib/storcli/storcli64
install/lib/py27/
install/lib/py27/cross-e17.3-x86_64-sysroot-runtime-20170519.180215-1.noarch.rpm
install/lib/py27/cross-e17.3-x86_64-sysroot-runtime-python-20170519.180215-1.noarch.rpm
install/lib/py27/cross-e17.3-x86_64-rtld-2.17-3.x86_64.rpm
install/ce_vibs/
install/ce_vibs/5.0.0/
install/ce_vibs/5.0.0/nfs-vaai-plugin.vib
install/ce_vibs/5.1.0/
install/ce_vibs/5.1.0/nfs-vaai-plugin.vib
install/ce_vibs/5.5.0/
install/ce_vibs/5.5.0/nfs-vaai-plugin.vib
install/ce_vibs/6.0.0/
install/ce_vibs/6.0.0/nfs-vaai-plugin.vib
install/nutanix-packages.json
Total bytes read: 3483627520 (3.3GiB, 509MiB/s)
INFO: Injecting SSH keys into SUM installer.
INFO: Chose boot disk: /dev/sdb
INFO: Customizing KVM instance
INFO: Imaging the SUM
INFO: Setting hostname to 'NTNX-4de6c357-A'
INFO: Formatting all data disks ['sdb', 'sdc']
INFO: Setting IP address information to 10.1.149.25/255.255.255.0
INFO: Executing /mnt/sum_installer/install/bin/sum_rescue with arg_list ['-i', '/mnt/sum_installer/install', '--factory_deploy', '--node_name=4de6c357-A', '--node_serial=f8f87539-9918-48cb-851d-03bd35008c0c', '--node_model=USE_LAYOUT', '--cluster_id=8952030504355473166', '--ipconfig=10.1.149.26/255.255.255.0/10.1.149.1', '--node_uuid=ef95ca57-5ccf-4ce3-aaf0-c011dac66bd2']
INFO: Copying SUM template files
INFO: Configuring SUM resources
INFO: Setting CVM memory to 12GB
INFO: Setting vCPUs to 2
INFO: CE is using LUN pass-through instead of HBA PCI pass-through.
INFO: Using PCI passthrough for NUMe devices.
INFO: Setting up authorized_keys
INFO: Copying firstboot scripts into /mnt/stage/root/firstboot
INFO: Copying SSH keys
INFO: Installing firstboot marker file
INFO: Imaging thread 'hypervisor' has completed successfully
```

Installation of the Nutanix CE in the VM begins

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

```
NCE - VMware Remote Console
VMRC | [Icons] [System Tray]
install/lib/py27/
install/lib/py27/cross-e17.3-x86_64-sysroot-runtime-20170519.180215-1.noarch.rpm
install/lib/py27/cross-e17.3-x86_64-sysroot-runtime-python-20170519.180215-1.noarch.rpm
install/lib/py27/cross-e17.3-x86_64-rtld-2.17-3.x86_64.rpm
install/ce_vibs/
install/ce_vibs/5.0.0/
install/ce_vibs/5.0.0/nfs-vaai-plugin.vib
install/ce_vibs/5.1.0/
install/ce_vibs/5.1.0/nfs-vaai-plugin.vib
install/ce_vibs/5.5.0/
install/ce_vibs/5.5.0/nfs-vaai-plugin.vib
install/ce_vibs/6.0.0/
install/ce_vibs/6.0.0/nfs-vaai-plugin.vib
install/nutanix-packages.json
Total bytes read: 3483627520 (3.3GiB, 509MiB/s)
INFO: Injecting SSH keys into SUM installer.
INFO: Chose boot disk: /dev/sdb
INFO: Customizing KUM instance
INFO: Imaging the SUM
INFO: Setting hostname to 'NTNX-4de6c357-A'
INFO: Formatting all data disks ['sdb', 'sdc']
INFO: Setting IP address information to 10.1.149.25/255.255.255.0
INFO: Executing /mnt/sum_installer/install/bin/sum_rescue with arg_list ['-i', '/mnt/sum_installer/install', '--factory_deploy', '--node_name=4de6c357-A', '--node_serial=f8f87539-9918-48cb-851d-03bd35008c0c', '--node_model=USE_LAYOUT', '--cluster_id=8952030504355473166', '--ipconfig=10.1.149.26/255.255.255.0/10.1.149.1', '--node_uuid=ef95ca57-5ccf-4ce3-aaf0-c011dac66bd2']
INFO: Copying SUM template files
INFO: Configuring SUM resources
INFO: Setting CUM memory to 12GB
INFO: Setting vCPUs to 2
INFO: CE is using LUN pass-through instead of HBA PCI pass-through.
INFO: Using PCI passthrough for NUMe devices.
INFO: Setting up authorized keys
INFO: Copying firstboot scripts into /mnt/stage/root/firstboot
INFO: Copying SSH keys
INFO: Installing firstboot marker file
INFO: Imaging thread 'hypervisor' has completed successfully

INFO: Creating layout file for CommunityEdition in position A
INFO: Injecting post-cluster create settings into CUM
INFO: Copying diagnostic UM into SUM
INFO: Imaging of SUM has completed successfully!
INFO: Imaging thread 'svm' has completed successfully
INFO: Cleaning up
INFO: Imaging process completed successfully!

Updating the initramfs... done.

Waiting for the Nutanix Controller VM to start up.....
```

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.

PRISM

Create a new password for the cluster admin.

admin

.....

.....

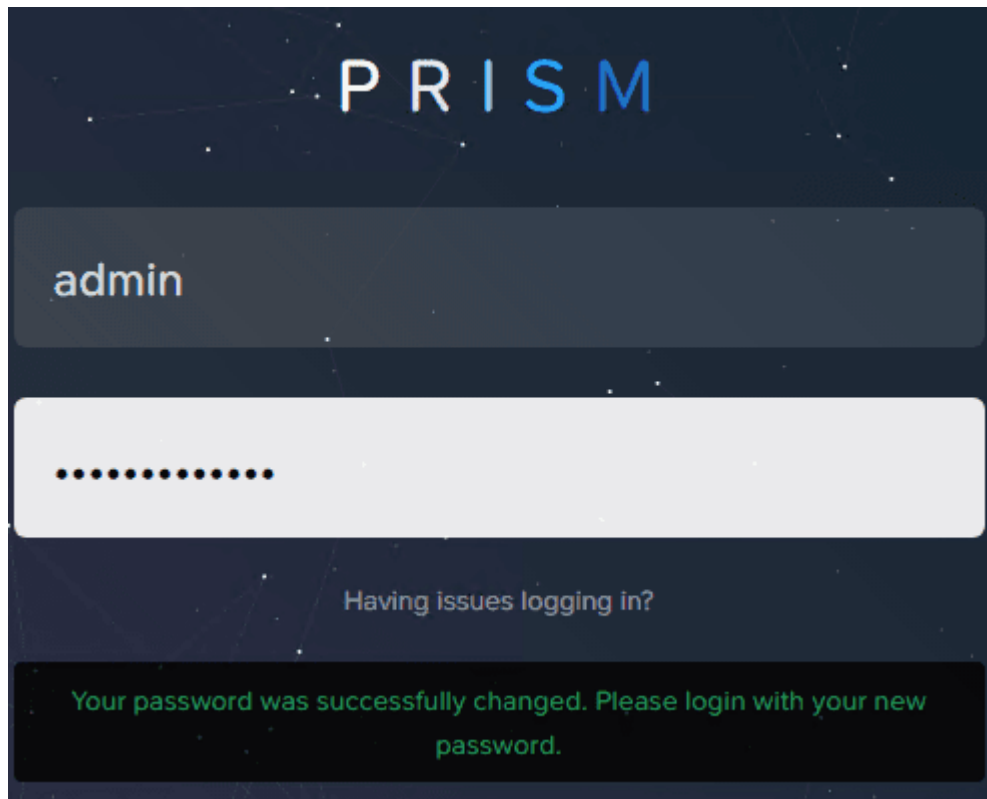


Note: When you change the admin user password, update any applications and scripts using the admin user credentials for authentication. Nutanix recommends that you create a user assigned with the admin role instead of using the admin user for authentication. The Prism Web Console Guide describes authentication and roles.

[Having issues logging in?](#)

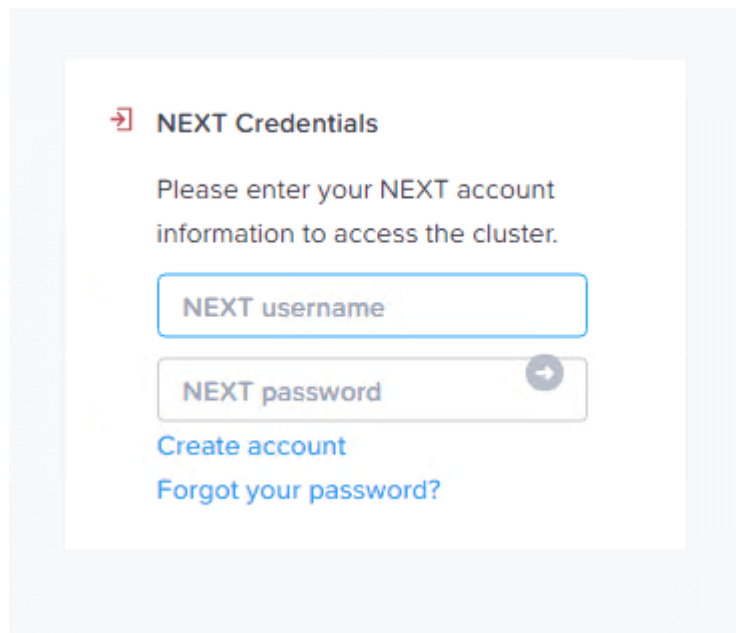
Logging into the Nutanix CE interface with the default credentials

After changing your admin password, login to the interface.



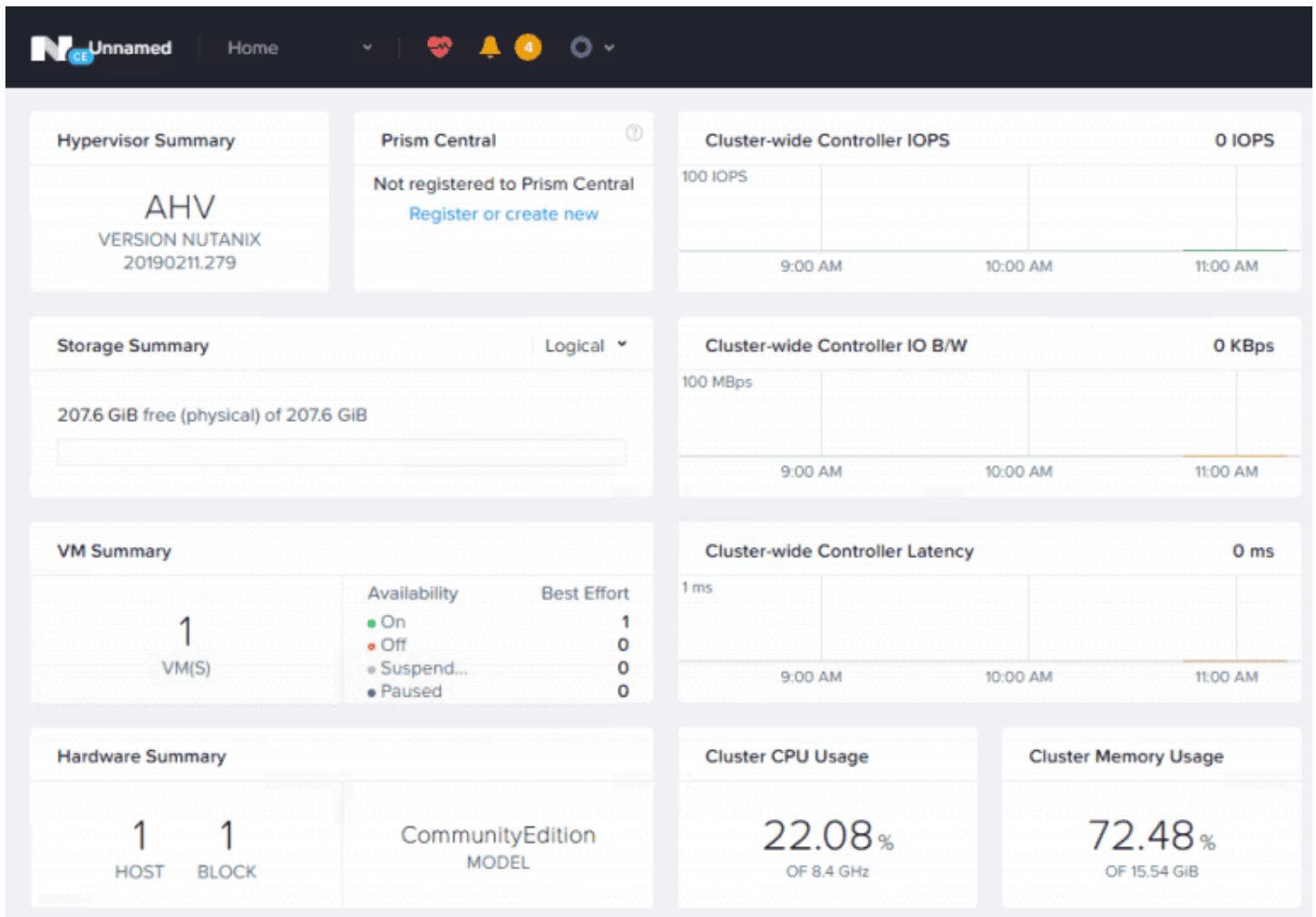
Prompted to change the default password for Nutanix CE

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.

Property	Value
Hypervisor:	VMware ESXi, 6.7.0, 16316930
Model:	SYS-E301-9D-8CN8TP
Processor Type:	Intel(R) Xeon(R) D-2146NT CPU @ 2.30GHz
Logical Processors:	16
NICs:	8
Virtual Machines:	5
State:	Connected
Uptime:	46 minutes

Manufacturer	Supermicro
--------------	------------

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)

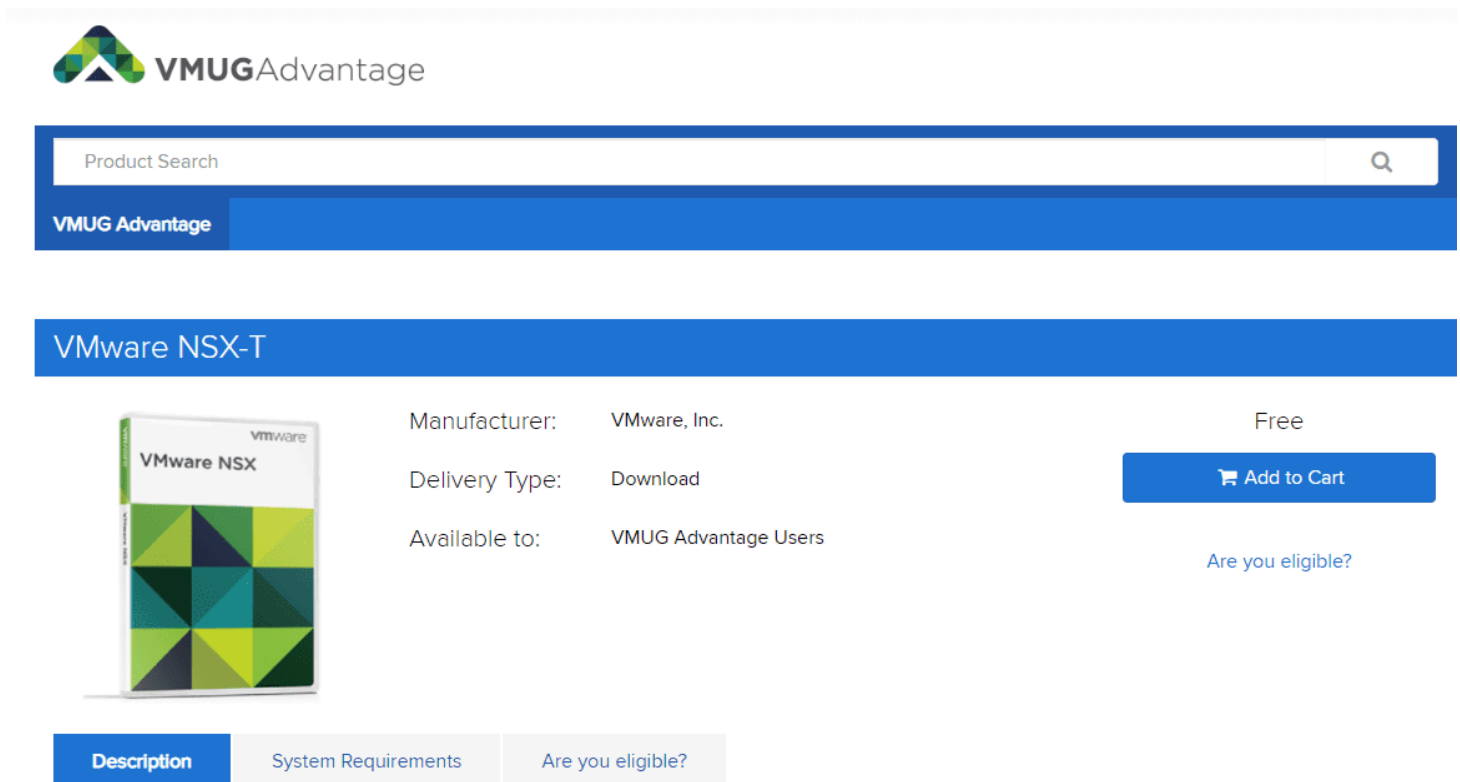
- 32 x IP Addresses (/27) for Ingress CIDR range is the minimum (must not overlap with Egress CIDR)
- All IP Addresses should be able to communicate with each other
- Storage
 - Ability to provision up to **1TB of storage**

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.



The screenshot shows the VMUG Advantage website interface. At the top, there is a search bar with the text 'Product Search' and a magnifying glass icon. Below the search bar is a blue navigation bar with the text 'VMUG Advantage'. The main content area features a blue header for 'VMware NSX-T'. On the left, there is a product image of the VMware NSX software box. To the right of the image, the following details are listed:

- Manufacturer: VMware, Inc.
- Delivery Type: Download
- Available to: VMUG Advantage Users

 The price is listed as 'Free'. There is a blue 'Add to Cart' button with a shopping cart icon. Below the button is a link that says 'Are you eligible?'. At the bottom of the product page, there are three tabs: 'Description' (which is active), 'System Requirements', and 'Are you eligible?'.

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.

HOW TO BUILD A NESTED NSX-T 2.3 LAB

Explore the features and capabilities of VMware NSX-T

Jim Streit, VCIX-NV
NSX Senior Technical Account Specialist (TAS)
VMware Professional Services

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](#)

The screenshot displays the vSphere Client interface for a cluster named 'cluster01'. The interface includes a top navigation bar with the vSphere Client logo, a search bar, and the user 'Administrator@VSPHERE.LOCAL'. The left sidebar shows a tree view of the environment, including 'pacific-vcsa.cloud.local', 'CloudPacific', and 'cluster01'. The main content area shows the 'cluster01' configuration page with tabs for Summary, Monitor, Configure, Permissions, Hosts, VMs, Datastores, and Networks. The Summary tab is active, displaying various cluster statistics and resource usage.

cluster01 ACTIONS

Summary Monitor Configure Permissions Hosts VMs Datastores Networks

Total Processors: 24
Total vMotion Migrations: 7
Fault Domains: Secondary, Preferred

CPU Free: 54.2 GHz
Used: 878 MHz Capacity: 55.08 GHz

Memory Free: 72.99 GB
Used: 34.98 GB Capacity: 107.97 GB

Storage Free: 512.93 GB
Used: 12.02 GB Capacity: 524.95 GB

Related Objects: Datacenter CloudPacific

vSphere HA

Tags

Assigned Tag	Category	Description
--------------	----------	-------------

Custom Attributes

Attribute	Value
-----------	-------

No items to display

Recent Tasks Alarms

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 24x7. 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

The screenshot shows the 'Edit Settings' window for a virtual machine named 'pacific-esx1'. The 'Virtual Hardware' tab is active. The 'CPU' section is expanded, and the 'Hardware virtualization' setting is highlighted with a red box. The checkbox 'Expose hardware assisted virtualization to the guest OS' is checked. Other settings include 4 CPUs, 1 core per socket, 4 sockets, 0 MHz reservation, unlimited MHz limit, normal shares with 4000 shares, performance counters disabled, and I/O MMU disabled. Memory is set to 12 GB and hard disk 1 is 4 GB.

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.

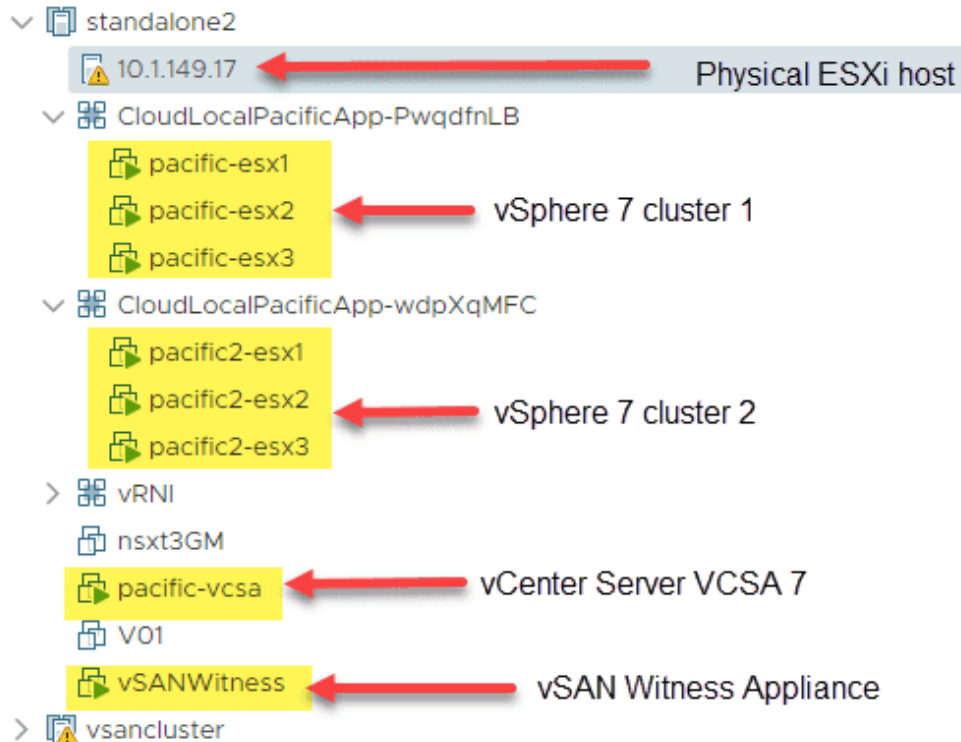
Check out the nested deployment here:

- <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.

The screenshot displays the vSphere Client interface for a vSAN cluster named 'cluster01'. The left-hand navigation pane shows a tree view of the environment, including 'pacific-vcsa.cloud.local', 'CloudPacific', and 'cluster01'. The main content area shows the 'Summary' tab for the cluster, which includes hardware and resource statistics. Below the statistics are several expandable panels: 'Related Objects' (showing Datacenter: CloudPacific), 'vSphere HA', 'Tags' (with an empty table), 'vSphere DRS', 'Cluster Consumers', and 'Custom Attributes' (with an empty table).

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.

DPG-Servers-LAG - Edit Settings

General

Advanced

VLAN

Security

Teaming and failover

Traffic shaping

Monitoring

Miscellaneous

Promiscuous mode	Accept	▼
MAC address changes	Accept	▼
Forged transmits	Accept	▼

CANCEL

OK

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**.

10.1.149.17 | ACTIONS

Summary Monitor **Configure** Permissions VMs Datastores Networks Updates

Storage

- Storage Adapters
- Storage Devices
- Host Cache Configuration
- Protocol Endpoints
- I/O Filters

Networking

- Virtual switches**
- VMkernel adapters
- Physical adapters
- TCP/IP configuration

Virtual Machines

- VM Startup/Shutdown
- Agent VM Settings
- Default VM Compatibility
- Swap File Location

System

- Licensing
- Host Profile
- Time Configuration

Virtual switches

ADD NETWORKING... REFRESH

▼ Distributed Switch: VDS-Mgmt-LAG | MANAGE PHYSICAL ADAPTERS ...

- DPG-Mgmt-LAG ...
 - VMkernel Ports (1)
 - Virtual Machines (0)
- DPG-Servers-LAG ...
 - VLAN ID: 149
 - Virtual Machines (11)

VDS-Mgmt-LAG-DVUplinks-44... ...

- Uplink 1 (0 NIC Adapters)
- Uplink 2 (0 NIC Adapters)

LAG01 ...

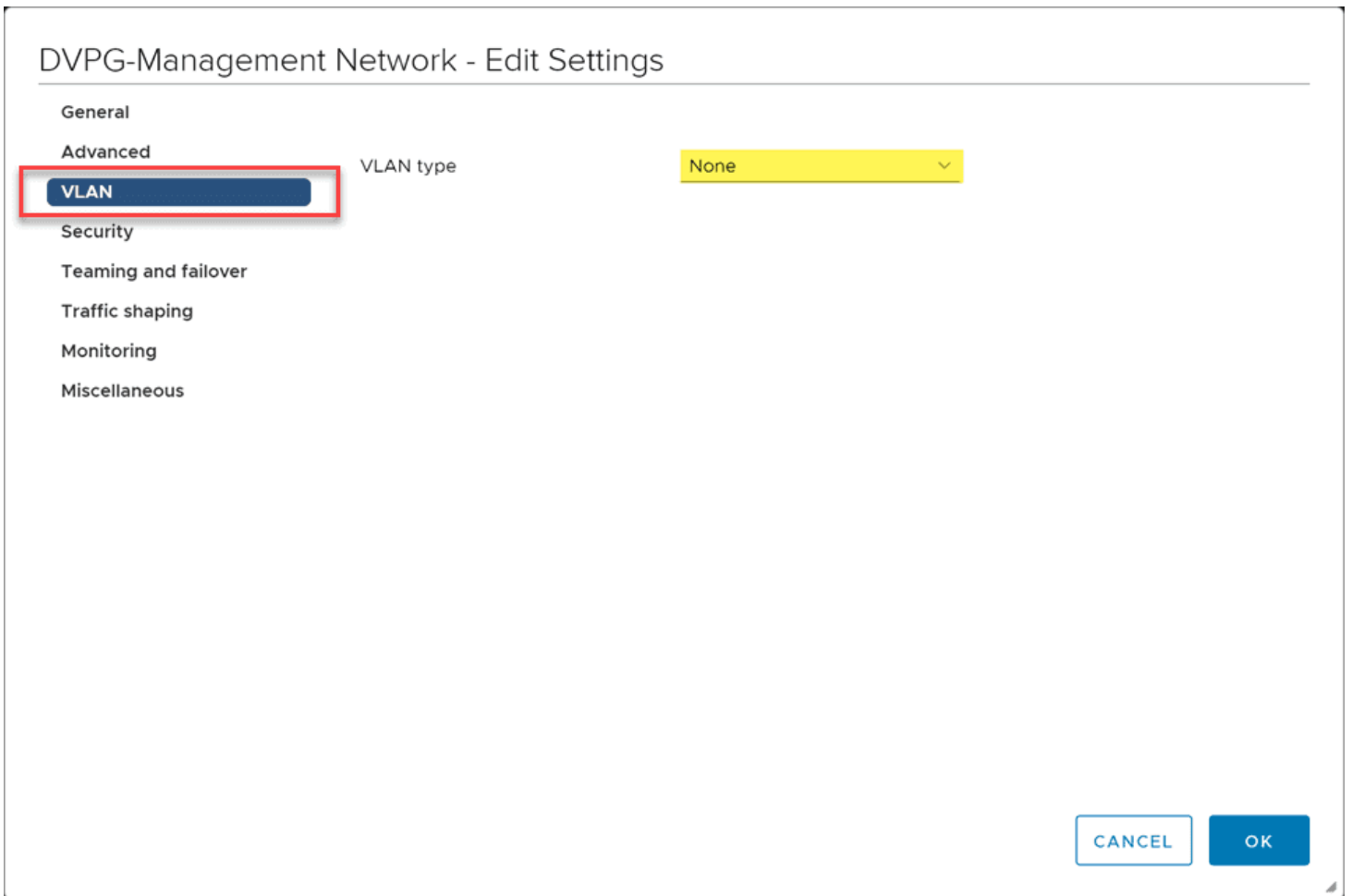
> Distributed Switch: VDS-Mgmt

> Standard Switch: vSwitch0

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.

