NVIDIA vGPU Software for VMware vSphere Hypervisor

Deployment Guide

Document History

DU-10915-001_v03

Version	Date	Authors	Description of Change
01	2018	AJ	Initial Release
02	September 2020	AFS	Reflect changes to the products featured in this guide
03	April 28, 2021	AFS, AH, UH	Update to VMware 7U2 and Horizon 8

Table of Contents

Chapter 1. Getting Started

NVIDIA virtual GPU (vGPU) software enables multiple virtual machines (VMs) to have simultaneous, direct access to a single physical GPU, using the same NVIDIA graphics drivers deployed on non-virtualized operating systems. This gives VMs unparalleled graphics performance application compatibility due to the cost-effectiveness of sharing a GPU among multiple scaled workloads.

This chapter covers how NVIDIA vGPU solutions fundamentally alter the landscape of desktop virtualization and GPU accelerated servers. NVIDIA vGPU enables users to execute these solutions with various workloads of all levels of complexity and graphics requirements. This chapter also describes the NVIDIA vGPU architecture, the NVIDIA GPUs recommended for virtualization, the NVIDIA vGPU software licensed products for desktop virtualization, as well as key standards supported by NVIDIA virtual GPU technology.

1.1 Why NVIDIA vGPU?

The promise of desktop and data center virtualization is flexibility and manageability. Initially, desktop and data center virtualization was used, as flexibility and security were the primary drivers of cost considerations. The democratization of technology has reduced the total cost of desktop virtualization, thereby expanding market accessibility and driving growth with NVIDIA as a key facilitator. This, along with advances in storage and multi-core processors, make for a good and/or competitive advantage regarding the ownership cost.

The biggest challenge for desktop virtualization is providing a cost-effective yet rich user experience. There have been attempts to solve this problem with shared GPU technologies like vSGA that are cost-effective. Still, those technologies do not provide the rich application support needed to succeed and ensure end-user adoption. Dedicated GPU pass-through with vDGA provides 100% application compatibility but is cost-effective only for the highest end-use cases due to the high cost and limited density of virtual machines per host server.

Due to the lack of scalable, sharable, and cost-effective per-user GPUs that provide 100% application compatibility, providing a cost-effective rich user experience provided a challenge for broad use cases in desktop virtualization. Meanwhile, high-end 3D applications did not work in a virtualized environment or were so expensive to implement with vDGA that it was reserved for only the most limited circumstances.

Today this is no longer true. Thanks to NVIDIA vGPU technology combined with VMware Horizon, NVIDIA vGPU allows flexibility where multiple virtual desktops share a single physical GPU. This breakthrough provides the 100% application compatibility of vDGA pass-through graphics. Still, the lower cost of multiple desktops sharing a single graphics card gives a rich yet more cost-effective user

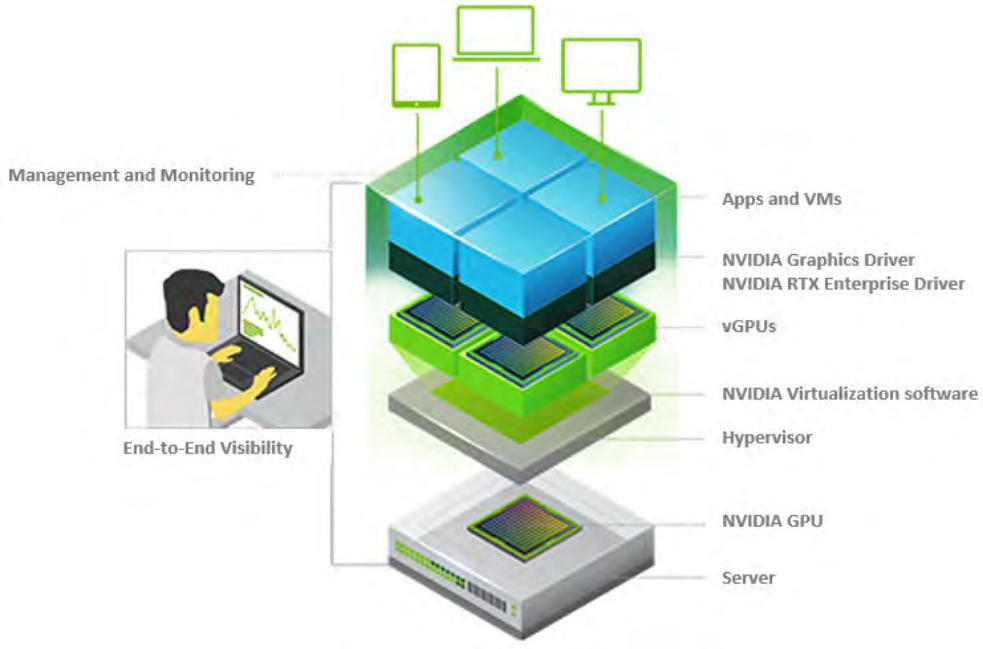
experience. With VMware Horizon, you can centralize, pool, and manage traditionally complex and expensive distributed workstations and desktops more easily. Now all your user groups can take advantage of the promise of virtualization.

1.2 NVIDIA vGPU Architecture

The high-level architecture of an NVIDIA virtual GPU-enabled VDI environment is illustrated in Figure 1.1. Here, we have GPUs in the server, and the NVIDIA vGPU manager software (VIB) is installed on the host server. This software enables multiple VMs to share a single GPU, or if there are multiple GPUs in the server, they can be aggregated so that a single VM can access multiple GPUs. This GPU-enabled environment provides unprecedented performance and enables support for more users on a server. Work typically done by the CPU can now be offloaded to the GPU. Physical NVIDIA GPUs can support multiple virtual GPUs (vGPUs) and be assigned directly to guest VMs under the control of NVIDIA's Virtual GPU Manager running in a hypervisor.

Guest VMs use the NVIDIA GPUs in the same manner as a physical GPU passed through by the hypervisor. In the VM itself, vGPU drivers are installed, which support the available different license levels.

Figure 1.1 NVIDIA vGPU Platform Solution Architecture



NVIDIA vGPUs are comparable to conventional GPUs in that they have a fixed amount of GPU memory and one or more virtual display outputs or *head.* Managed by the NVIDIA vGPU Manager installed in the hypervisor, the vGPU memory is allocated out of the physical GPU frame buffer when the vGPU is created. The vGPU retains exclusive use of that GPU Memory until it is destroyed.

Note: These are virtual heads, meaning that there is no physical connection point for external physical displays on GPUs.

All vGPUs that reside on a physical GPU share GPU engines, including the graphics (3D) and video decode and encode engines. The right side of Figure 1.1 shows the vGPU internal architecture. The VM's guest OS leverages direct access to the GPU for performance and fast critical paths. Noncritical performance management operations use a para-virtualization interface to the NVIDIA Virtual GPU Manager.

1.3 Supported NVIDIA GPUs

NVIDIA virtual GPU software is supported with NVIDIA GPUs and supported on vSphere with a vSphere/ESXi Enterprise Plus License. Determine the GPU best suited for your environment dependent upon application use, optimization for performance or density, and professional visualization via GPU acceleration.

Please refer to NVIDIA GPUs for Virtualization for a complete list of recommended and supported GPUs. For a list of certified servers with NVIDIA GPUs, consult the NVIDIA vGPU Certified Servers page. Cross-reference the NVIDIA certified server list with the VMware HCL to find servers best suited for your NVIDIA vGPU and VMware vSphere environment. Each card requires auxiliary power cables connected to it (except NVIDIA T4). Most industry-standard servers require an enablement kit for the proper mounting of NVIDIA cards. Check with your server OEM of choice for more specific requirements.

1.4 NVIDIA vGPU Software Licensed Products

NVIDIA virtual GPU software divides NVIDIA GPU resources so the GPU can be shared across multiple virtual machines running any application.

The portfolio of NVIDIA virtual GPU software products for desktop virtualization is as follows:

- NVIDIA Virtual Applications (NVIDIA vApps)
- NVIDIA Virtual PC (NVIDIA vPC)
- NVIDIA RTX® Virtual Workstation (RTX vWS)

> **!** CAUTION: To run these software products, you need an NVIDIA GPU supported by vGPU software and a license that addresses your specific use case bundled with a vSphere/ESXi Enterprise Plus License.

NVIDIA vGPU software allows you to partition or fractionalize an NVIDIA data center GPU. These virtual GPU resources are then assigned to VMs in the hypervisor management console using vGPU profiles. Virtual GPU profiles determine the amount of GPU frame buffer allocated to your virtual machines (VMs). Selecting the correct vGPU profile will improve your total cost of ownership, scalability, stability, and performance of your VDI environment.

The NVIDIA vGPU software solution offers unmatched flexibility and performance when correctly paired with the proper vGPU software Licenses and NVIDIA GPU combination. These vGPU software solutions are designed to meet today's modern enterprises' ever-shifting workloads and organizational needs. Refer to the NVIDIA Virtual GPU Positioning Guide to select the best vGPU software license and GPU combination based on your workload.

- Please refer to NVIDIA GPUs for Virtualization for a product overview of each NVIDIA GPU.
- The NVIDIA Virtual GPU Software User Guide contains Information on Supported Graphics APIs and Support for OpenCL and NVIDIA® CUDA® applications.

1.4.1 NVIDIA AI Enterprise licensed software

The NVIDIA AI Enterprise software suite of licensed products is an end-to-end, cloud-native suite of AI and data analytics software that includes AI frameworks and tools for performance-optimized deep learning, machine learning, and data science workloads that simplify building, sharing, and deploying AI software. NVIDIA AI Enterprise's end-to-end software solution is for every phase of AI, from data prep (NVIDIA RAPIDS™) to AI training (TensorFlow, PyTorch) to inference (NVIDIA® TensorRT™), and scale (NVIDIA Triton™ Inference Server).

The NVIDIA AI Enterprise software suite is optimized, certified, and supported by NVIDIA to run on VMware vSphere with NVIDIA-Certified Systems™. It includes key enabling technologies from NVIDIA for rapid deployment, management, and scaling AI workloads in the modern hybrid cloud.

Note: As of NVIDIA vGPU software release 13.0, VMware compute-intensive workloads will be supported through the NVIDIA AI Enterprise software suite of licensed products; These AI and data science applications and frameworks are optimized and exclusively certified by NVIDIA to run on VMware vSphere with NVIDIA-Certified Systems. Refer to the NVIDIA AI Enterprise deployment guide for more information.

1.5 Before You Begin

This section describes building a Proof of Concept (POC), sizing your VDI environment, general prerequisites, and some general preparatory steps that must be addressed before deployment.

1.5.1 Building a Proof of Concept

It is recommended that you test your unique workloads to determine the best NVIDIA virtual GPU solution to meet your organizational needs and goals. The most successful customer deployments start with a proof of concept (POC) and are "tuned" throughout the lifecycle of the deployment. Beginning with a POC enables customers to understand the expectations and behavior

of their users and optimize their deployment for the best user density while maintaining required performance levels. Continued monitoring is essential because user behavior can change throughout a project or an individual change within the organization. A user that was once a light graphics user (vApps, vPC) might become a heavy graphics user via professional visualization (RTX vWS) when they change teams and/or projects.

Consider the following during your POC:

- A comprehensive review of all user groups, their workloads, applications utilized, and current and future projections should be considered
- A vision for balancing user density with end-user experience measurements and analysis
- Gather feedback from IT and end-users regarding infrastructure and productivity needs.

1.5.2 Sizing Your Environment

Based on your Proof of Concept (POC), we recommend sizing an appropriate environment for each user group you are trying to reach with your evaluation. NVIDIA provides in-depth sizing guides to assist with the process of optimally scaling your organization's workloads.

Please refer to the appropriate sizing guides below to build your NVIDIA vGPU environment:

- NVIDIA vPC Sizing Guide
- NVIDIA Virtual Apps on Citrix Virtual Apps with VMware ESXI
- NVIDIA RTX Virtual Workstation (vWS) Sizing Guide

As an overview:

- Scope your environment for the needs of each end-user
- Run a proof of concept for each deployment type.
- Implement the NVIDIA recommended sizing methodology
- Utilize benchmark testing to help validate your deployment.
- Utilize NVIDIA-specific and industry-wide performance tools for monitoring.
- Ensure performance and experience metrics are within acceptable thresholds.

Note: It is essential to resize your environment when switching from Maxwell GPUs to newer GPUs like the Pascal, Turing, and Ampere GPUs. NVIDIA GPU Architectures that came after the Maxwell Generation of GPUs may support ECC memory which incurs a 1/15 overhead cost on the GPU frame buffer. Additional information can be found here.

1.5.3 Choosing Your Hardware

The following elements are required to install and configure vGPU software on VMware vSphere:

- NVIDIA certified servers with NVIDIA cards (see the web page NVIDIA vGPU Partners for a list of certified NVIDIA servers. It is helpful to cross-check this list with the VMware HCL to ensure compatibility for your deployment. The following specifications are recommended:
  - CPU for vPC/vApps:
    - Intel Xeon Gold 6338 @2.0 GHz or faster
    - AMD EPYC 7713 @2.0 GHz or faster
  - CPU for RTX vWS:
    - Intel Xeon Gold 6354 @3.0 GHz or faster
    - AMD EPYC 7763 @2.4 GHz or faster

    Note: When considering CPUs for your vGPU deployment, NVIDIA recommends the following:

    vPC deployments should have a higher core count and lower clock speeds to prioritize density.

    RTX vWS deployments should have a higher clock speed priority core count to prioprioritypperformance.
  - High-speed RAM
  - Fast networking, e.g., best-in-class Mellanox ConnectX certified by NVIDIA
- High-performance storage with Virtual SAN or high IOPS storage system
- High-performance endpoints for testing access

Note: If you use local storage, input/output operations per second (IOPS) play a significant role in performance. If you are using VMware for a Virtual SAN, see the VMware Virtual SAN requirements website for more details.

1.5.4 General Prerequisites

The following elements are required to install and configure vGPU software on VMware vSphere. For demonstration purposes, this guide uses ESXi release 7.0.2.

Review the Getting Started page of the VMware Horizon documentation. It provides a roadmap for implementing Horizon as a server with Citrix clients and links to a list of VMware education courses and other resources.

Note: The requirements to install and configure vGPU software on VMware vSphere are the same for implementing VMware Horizon as a Server. Refer to the Choosing Your Hardware section for the requirements.

- Ensure to use the appropriate NVIDIA GPU for your use case. Refer to the NVIDIA Virtual GPU Positioning Guide to better understand which GPU suits your deployment requirements. For additional guidance, contact your NVIDIA and VMware sales representatives.

VMware vSphere 7.0.2 build:

- vSphere and vCenter Server is available from the VMware website at Product Evaluation Center for VMware vSphere 7.0.
- As of vSphere 7.0, deploying a new vCenter Server or upgrading to vCenter Server 7.0 requires using the vCenter Server Appliance (VCSA).

VMware Horizon software:

- If needed, you may register for a trial to obtain the license keys required for various elements to deploy and manage Horizon at the web page Welcome to My VMware.

NVIDIA vGPU software:

- NVIDIA vGPU manager vSphere Installation Bundle (VIB)

Note: The vGPU Manager VIB is loaded similarly to a driver in the vSphere hypervisor, and the vCenter Server then manages it.

- For NVIDIA vGPU software builds latest releases, please refer to your NVIDIA Application Hub.
- For additional information on these current releases, please consult NVIDIA vGPU product documentation for VMware vSphere.

Microsoft software:

- Refer to the vGPU Software documentation for a supported Windows Guest OS list.

Your choice of one of the following CLI/SSH/SCP tools installed on your Windows-based toolbox PC:

- MobaXterm (SSH and SPC) is available from the mobaxterm website download; This is the recommended tool.
- Putty (SSH) and WinSPC (SPC), available from www.putty.org and www.winscp.net.
- Tight VNC Viewer (SSH) is available from www.tightvnc.com/download.php.

Licenses:

- From the VMWare Horizon website:
  - vSphere/ESXi Enterprise Plus is required to use vGPU on vSphere
  - VM vSphere
  - VMware Horizon
- Microsoft licenses can be found at the service center here (volume licenses recommended)

Testing and Benchmarking tools (Optional but recommended):

- NVIDIA System Management interface (NVSMI)
- GPU Profiler - GPUProfiler download
- VMware vRealize Operations for Horizon (V4H)
- Lakeside Systrack with GPU monitoring
- End-User satisfaction survey

A Virtual GPU Certified Server on which to install VMWare Horizon

- VMware vSphere ESXi
- VMware vCenter
- Must be joined to a domain
- Must be assigned a static IP address

Note: VMware Horizon consists of many components working together in concert. Many environments are built on the vSphere virtual infrastructure. Refer to the VMware Horizon Documentation page for a complete list of components and features.

The versions of vCenter and ESXi supported by Horizon 8 can be found in the VMware Product Interoperability Matrix.

1.5.5 Preparation for Pre-Installation

Before you install NVIDIA vGPU software:

1. Determine how vSphere will run on the physical hosts. Consider booting from a thumb drive, as this is an Early Access build.
2. Download and install any of the following for SSH and SCP:
   - An SSH tool (such as PuTTY)
   - WinSCP, which handles both SSH and SCP functions
   - MobaXterm, which handles both SSH and SCP functions
   - Tight VNCViewer (Remote console)

1.5.6 Server BIOS Settings

Configure the BIOS for your physical hosts, as described below:

- Hyperthreading – Enabled
- Power Setting or System Profile– High Performance
- CPU Performance (if applicable) – Enterprise or High Throughput
- Memory Mapped I/O above 4-GB - Enabled (if applicable)
- SR-IOV enabled
- VT-d/IOMMU - Enabled

Note: NVIDIA GPU architectures after the Maxwell architecture (Pascal, Turing, & Ampere) support VT-d and IOMMU.

Chapter 2. Installing VMware ESXi

This chapter covers the following ESXi installation topics:

- Choosing an Install Method
- Preparing the USB boot media
- Installing VMware ESXi
- Initial host configuration

Note: This deployment guide assumes you are building an environment as a proof of concept, not as a production deployment. Consequently, it recommends settings and other choices that make the process faster and easier. Before you build your production environment, consult the corresponding guides for each technology and make choices appropriate for your needs.

2.1 Choosing an Install Method

VMware vSphere offers several methods to install the ESXi hypervisor. These installation methods include CD/DVD (burning the ESXi Installer ISO), USB Flash drive (formatting a USB flash drive to Boot the ESXi Installation), and PXE Boot Installation (Booting from a location on the network). The ESXi Installer must be accessible to the system you are installing ESXi. See the vSphere documentation regarding best practices for logs when booting from a USB flash drive or similar device. This guide uses an IPMI and virtual media to boot from an ISO file and install local storage.

2.2 Preparing USB Boot Media

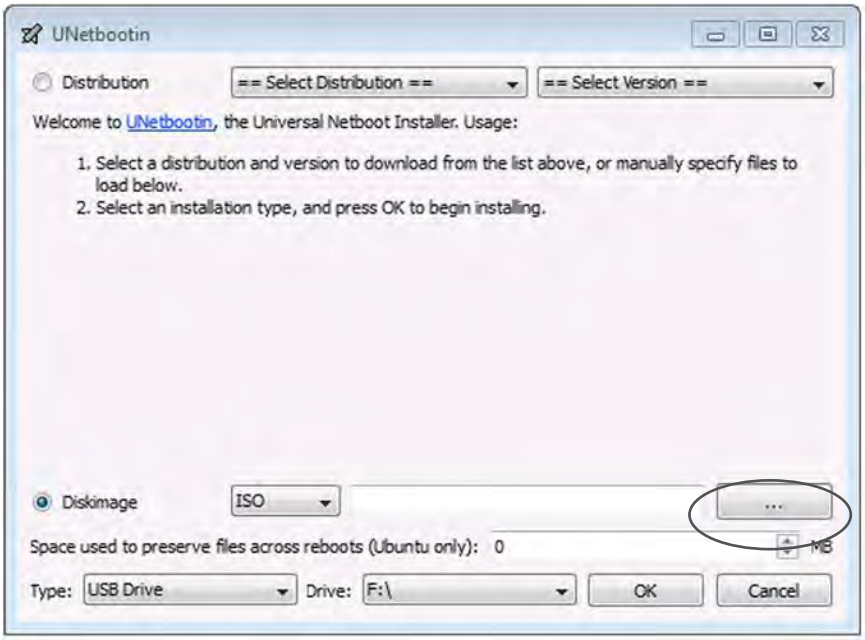
Note: Booting vSphere from a USB flash drive is useful if your host has an existing vSphere Version or earlier installation that you want to retain. Installing ESXi on a supported USB flash drive or SD flash card (2004784).

Use the following procedure to prepare a USB flash drive for booting:

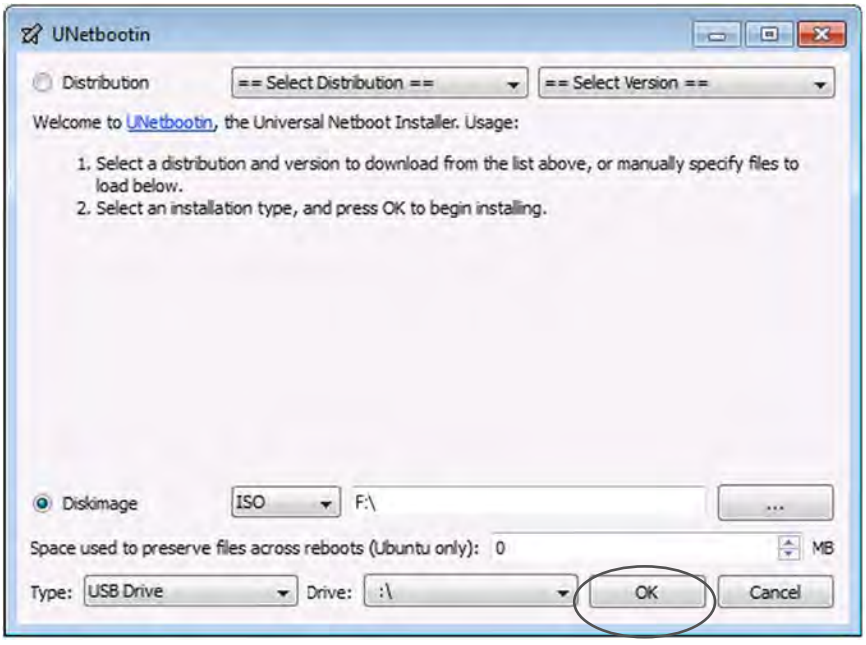
1. Download **UNetbootin** from the GitHub UNetbootin page.
   - The Windows version does not include an installer; however, the OSX version is packaged in a `.DMGizing` file that you must mount. You must also copy the application to the `Applications` folder before launching.

- Alternatively, you can use YUMI, which allows booting from multiple installation images on one USB device, and lets you load the entire installation into RAM. You can download YUMI from Pendrivelinux.com.

2. Start the application, select **Diskimage**, and click the "**...**" button in the bottom right corner to browse for the installation `.ISO` file:



3. Browse to the location that contains the installation `.ISO` file, then select **Open.**
4. Select the mounted USB flash drive location where you perform the installation, then click **OK**. The copying process begins, and UNetbootin displays a series of progress bars.



5. When the copying process is complete, click **Exit**, then **remove the USB flash drive**.

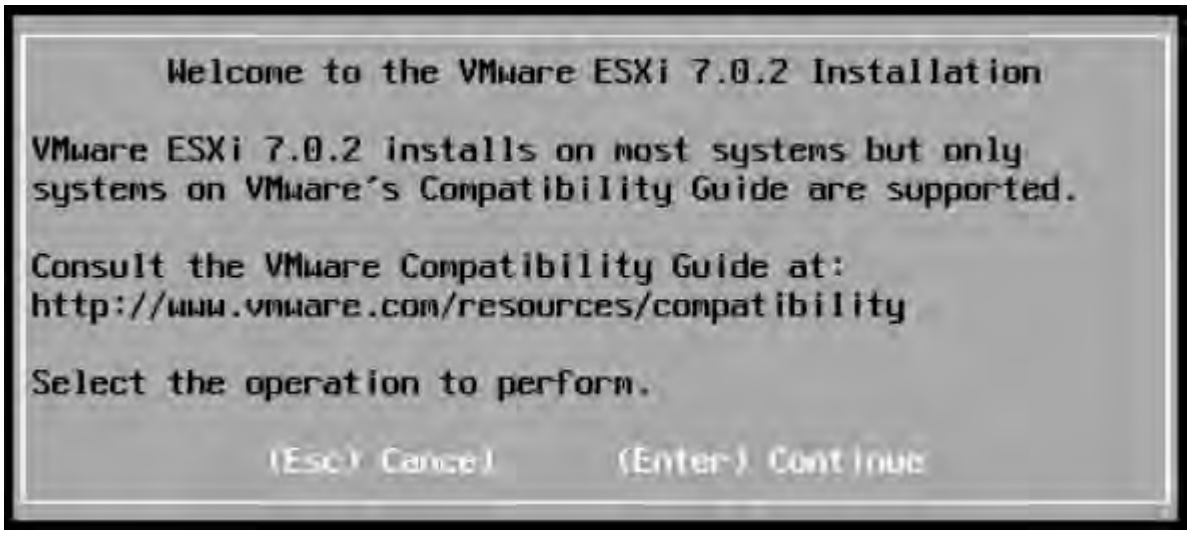
6. To install from this USB flash drive, insert it into the host using either an internal USB port or an external USB port, then set the port as the primary boot source or select from the boot menu on power-up.

2.3 Installing VMWare ESXi

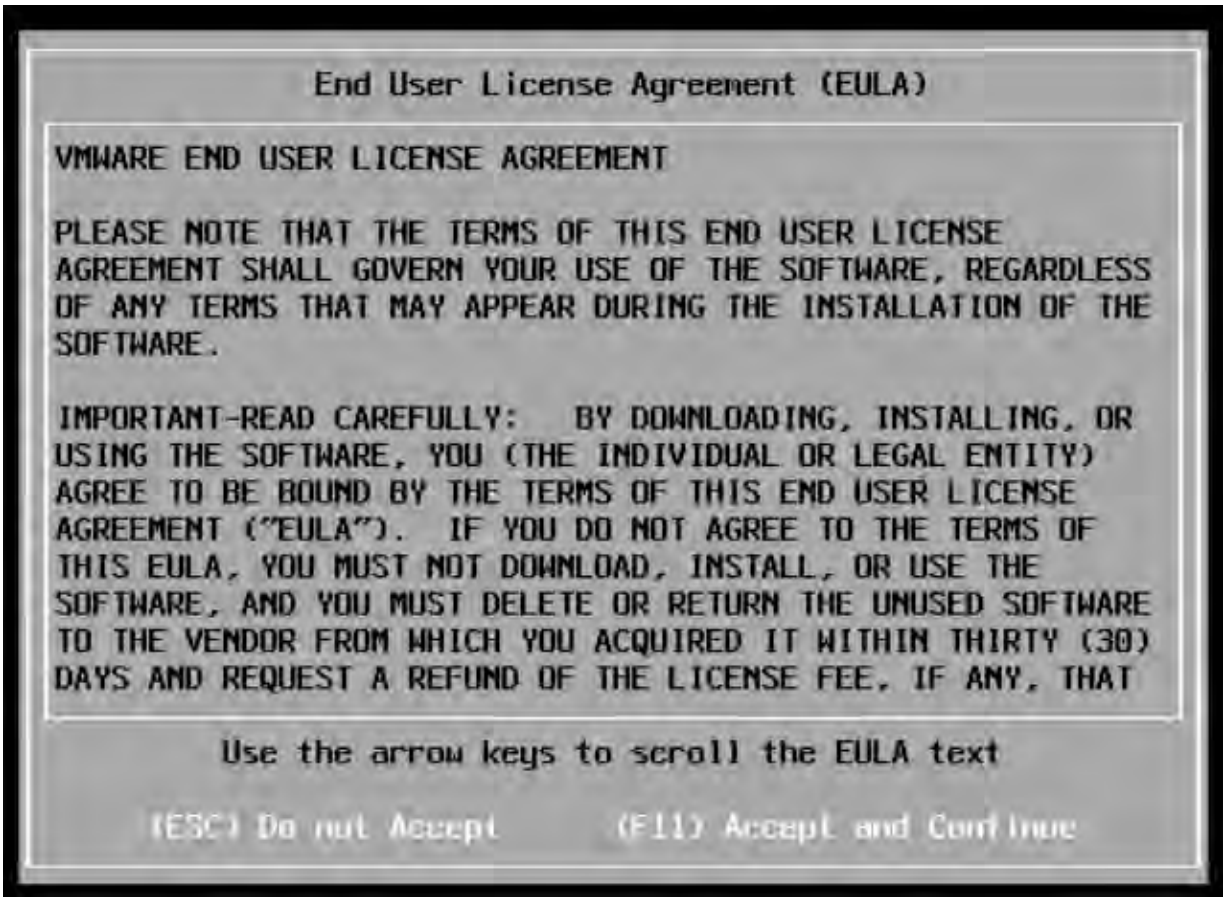
Use the following procedure to install ESXi regardless of boot source. Select the boot medium that holds the vSphere ISO from your host's boot menu.

> **!** CAUTION: VMware ESXi 7.0 Update 2 is used to capture content for this guide. Text, images, and screen layout may differ from version to version.

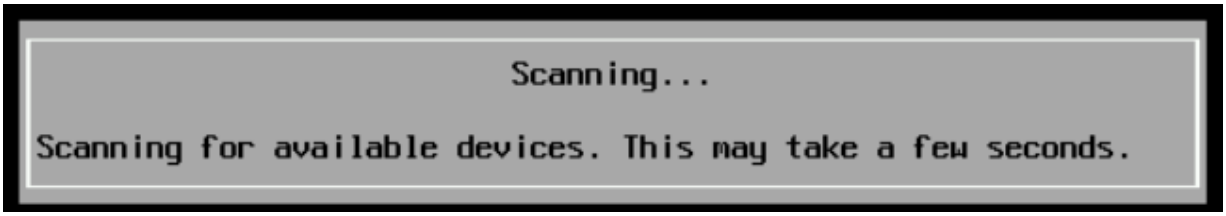
1. Apply power to start the host.
2. Select the installer using the arrow keys and press **[ENTER]** to begin booting the ESXi installer. A compatibility warning is displayed.



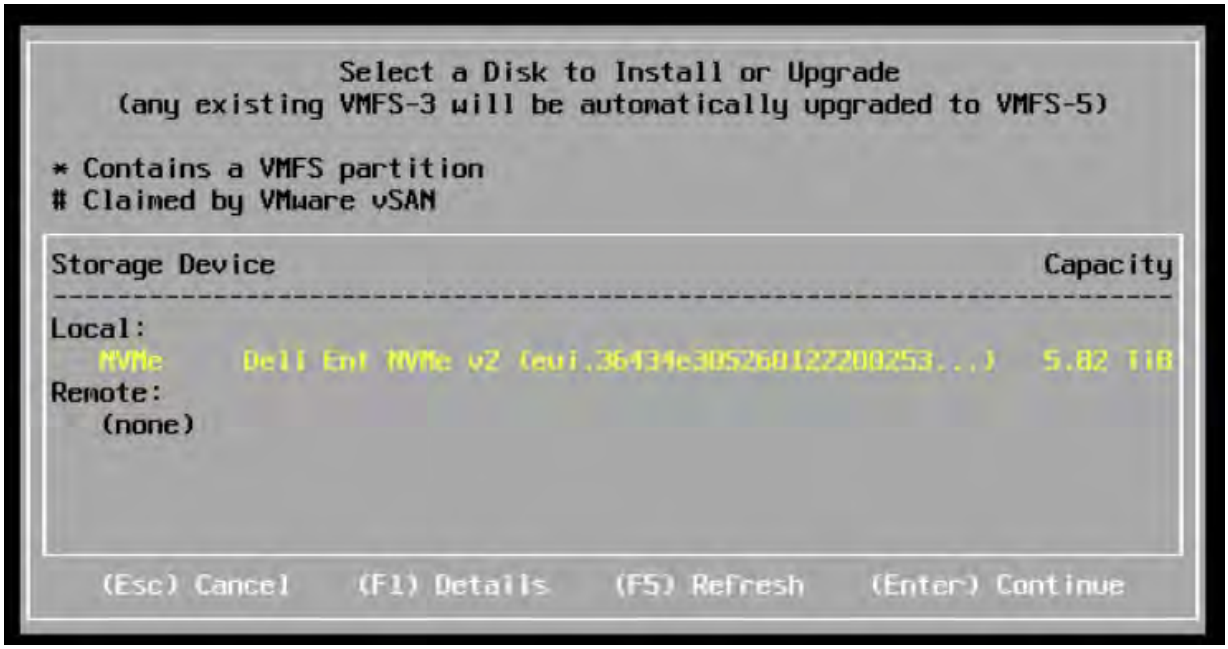
3. Press **[ENTER]** to proceed. The End User License Agreement (EULA) displays.



4. Read the EULA and then press **[F11]** to accept it and continue the installation. The installer scans the host to locate a suitable installation drive:



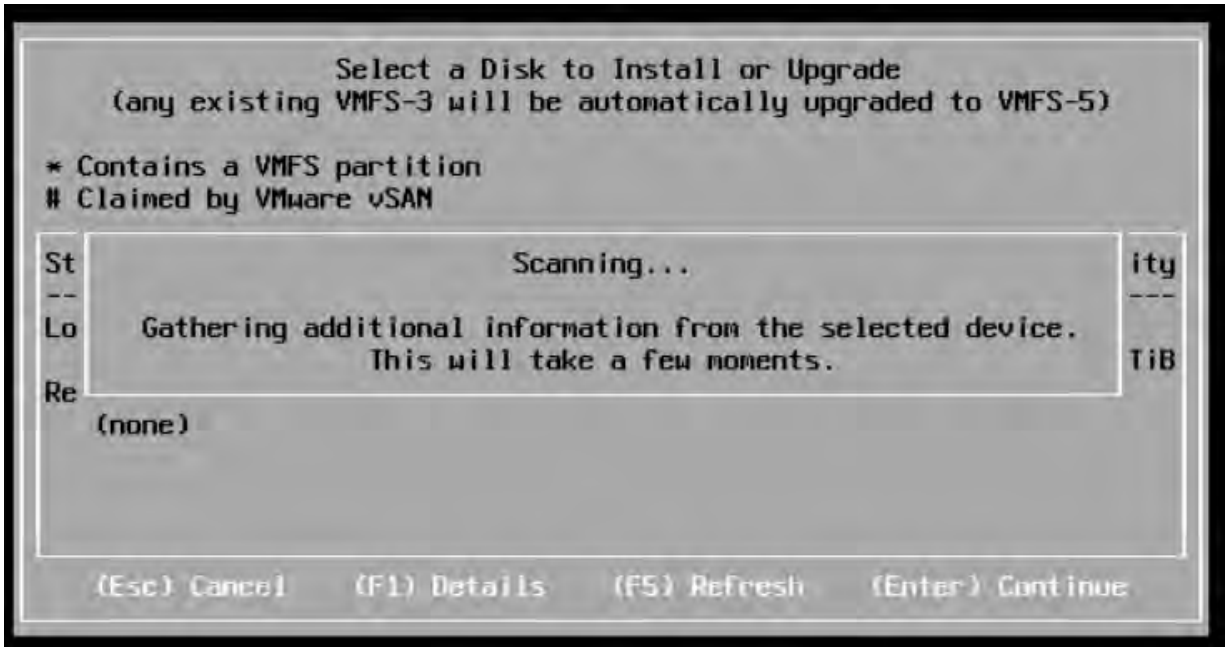
5. It should display all drives available for install.



6. Use the arrow keys to select the drive you want to install ESXi, and then press **[ENTER]** to continue.

Note: You can install ESXi to a USB flash drive, then boot and run the system from that USB flash drive. This sample installation shows ESXi being installed on a local hard drive.

7. The installer scans the chosen drive to determine suitability for install



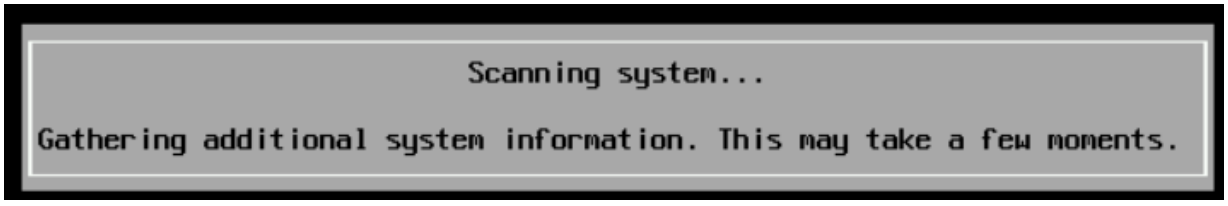
8. The Confirm Disk Selection window displays.

9. Press **[ENTER]** to accept your selection and continue.
10. **Please select a keyboard layout** window displays. Select your desired keyboard layout using the arrow keys and press **[ENTER]**.

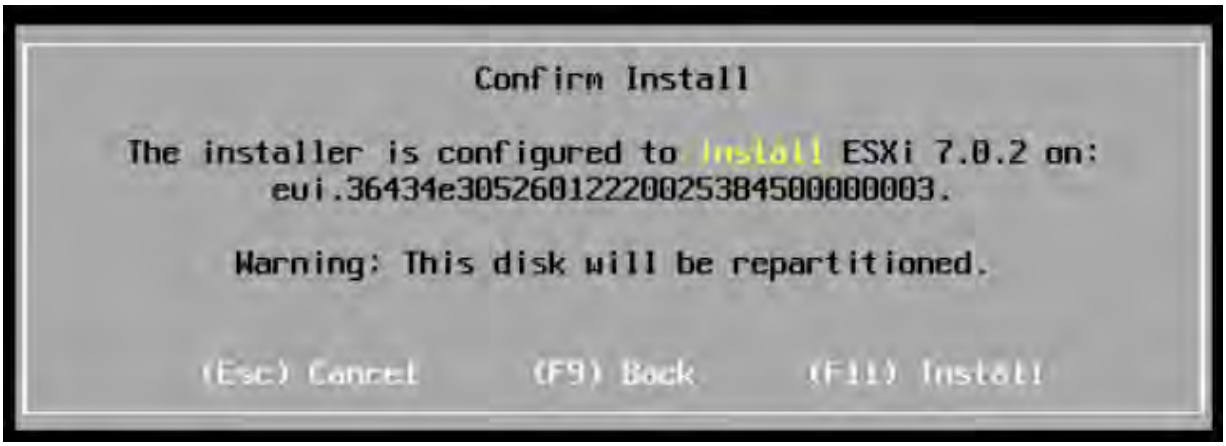
11. The Enter a root password window displays. Enter a **root password** in the Root password field. Confirm the password in the **Confirm password field** and press **[ENTER]** to proceed.

CAUTION: To prevent unauthorized access, your selected root password should contain at least eight (8) characters and a mix of lowercase and capital letters, digits, and special characters.

12. The installer rescans the system.

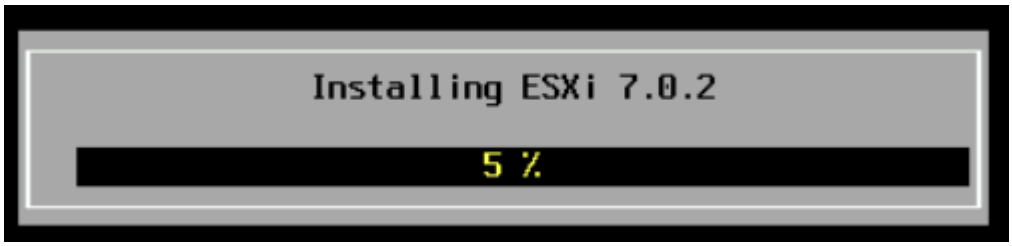


13. When the installer finishes rescanning, it displays the **Confirm Install** window. Press **F11** to proceed with the installation.

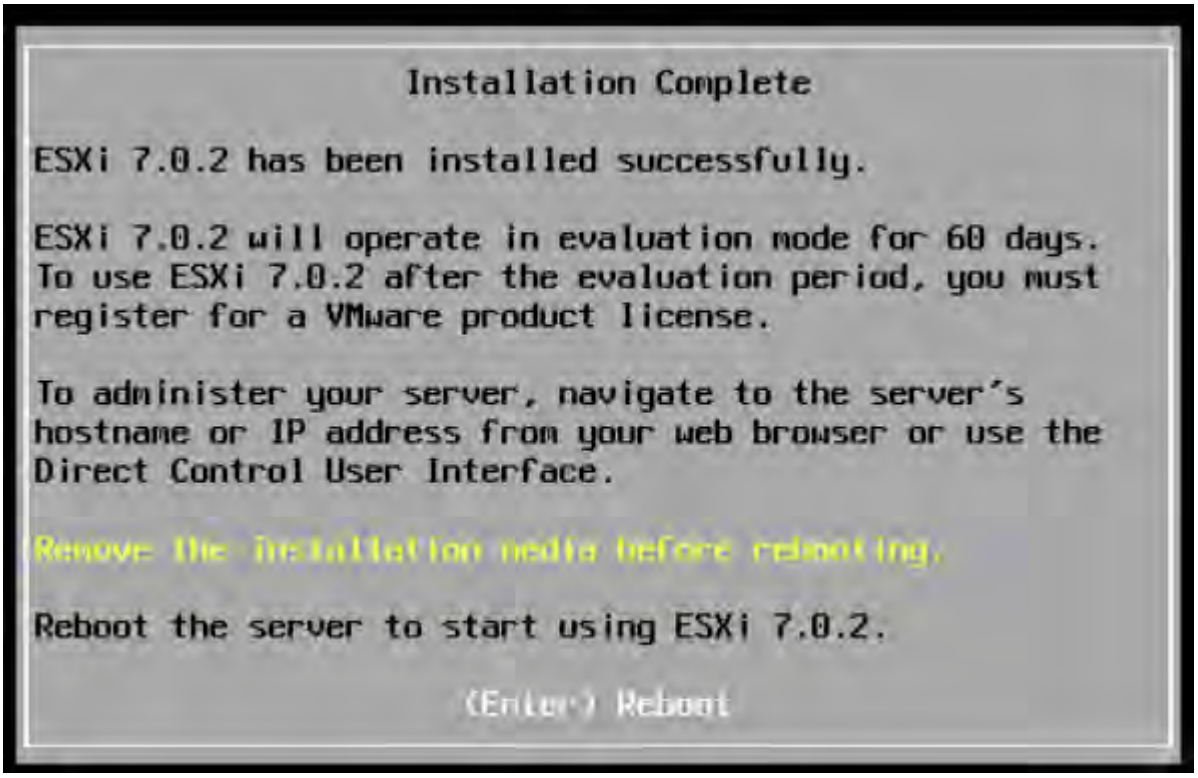


CAUTION: The installer will repartition the selected disk. All data on the selected disk will be destroyed.

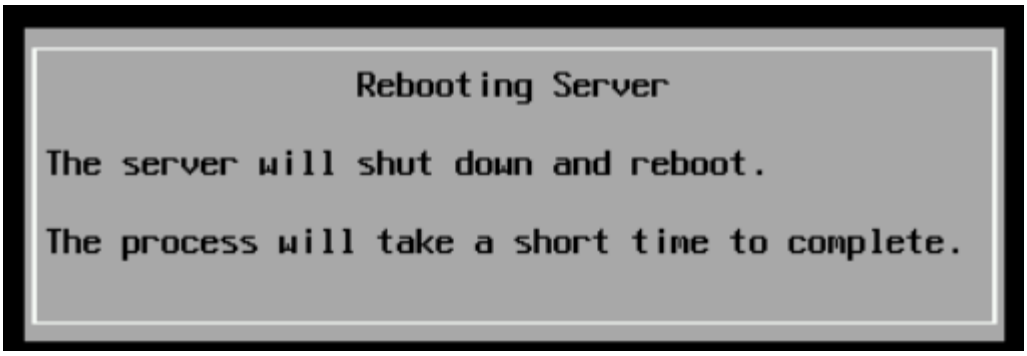
- The ESXi installation proceeds.



14. The installer displays the **Installation Complete** window. Confirm that your installation medium has been unmounted and your system is set to the boot disk. Then press **Enter** to reboot the system.



15. The server will shut down and reboot.

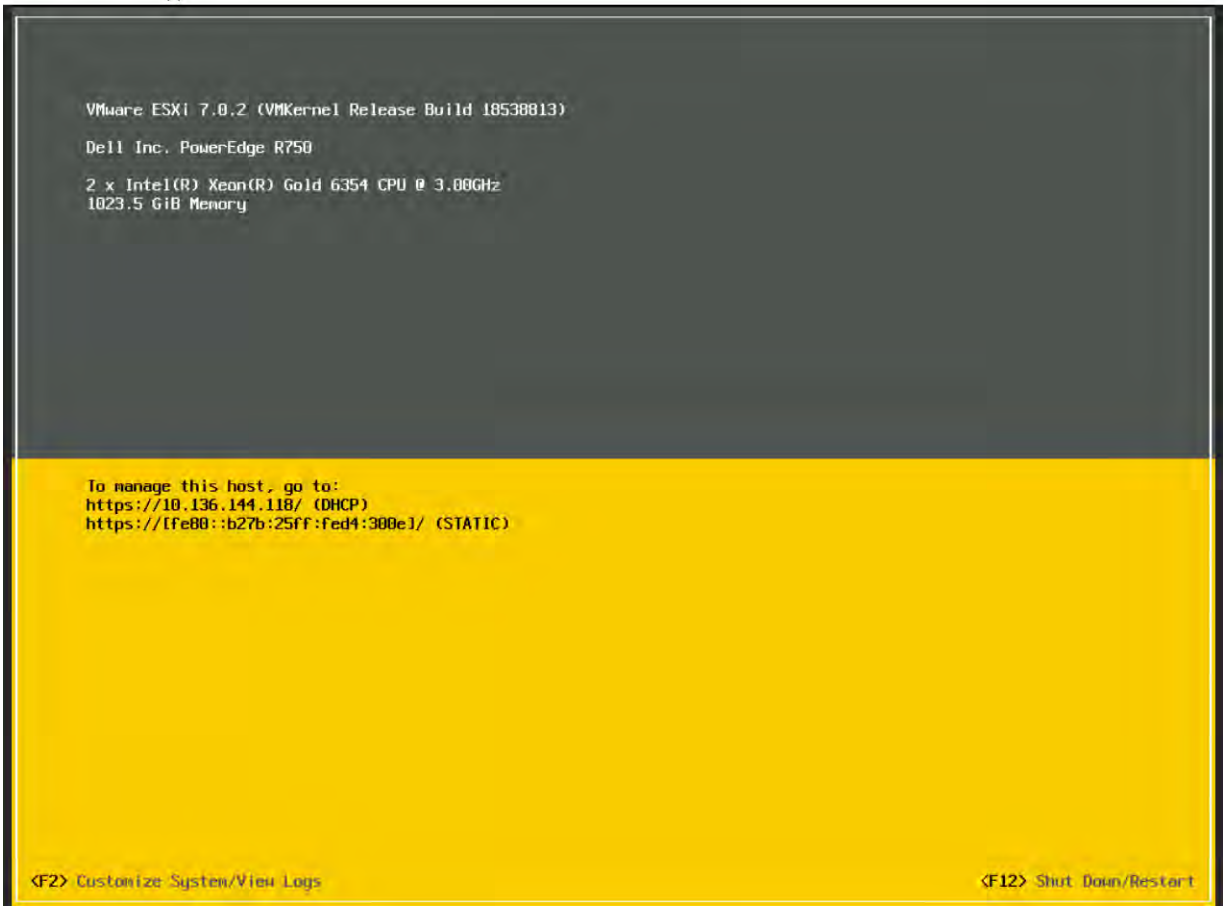


16. VMWare ESXi has been installed successfully.

2.4 Initial Host Configuration

A countdown timer displays when you first boot ESXi. You can wait for the countdown to expire or press **[ENTER]** to proceed with booting. A series of notifications displays during boot, taking several

minutes to complete. When the boot process completes, VMware ESXi displays the DCUI Splash screen.

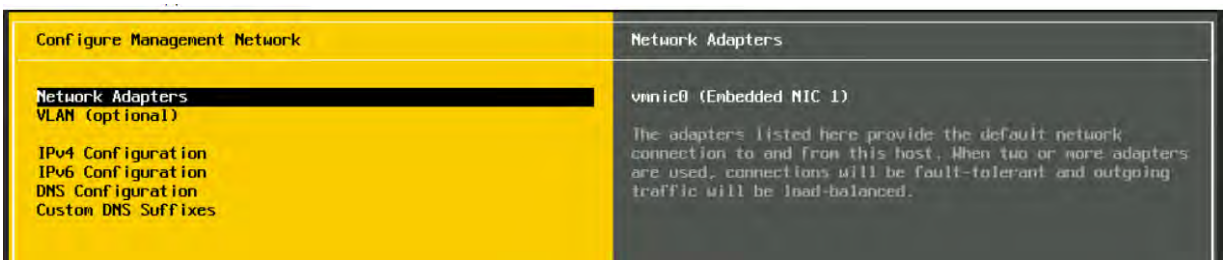


Use the following procedure to configure the host:

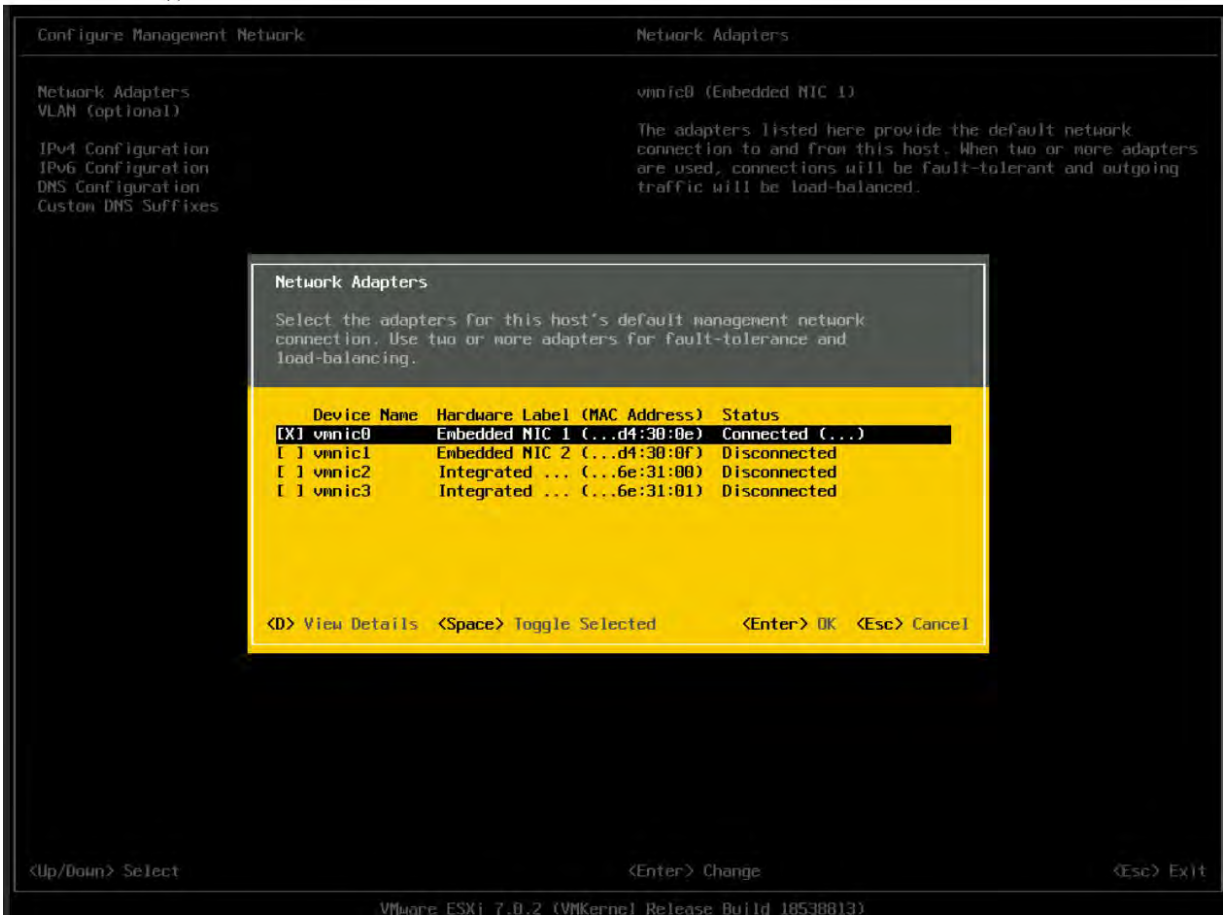
1. Press **F2** to Customize the System.
2. ESXi displays the **Authentication Required** window.
   - Enter the root account credentials you created during the installation process, and then press Enter.

3. ESXi displays the **System Customization** screen. Select the **Configure Management Network** tab using the down-arrow key, then press **Enter**.

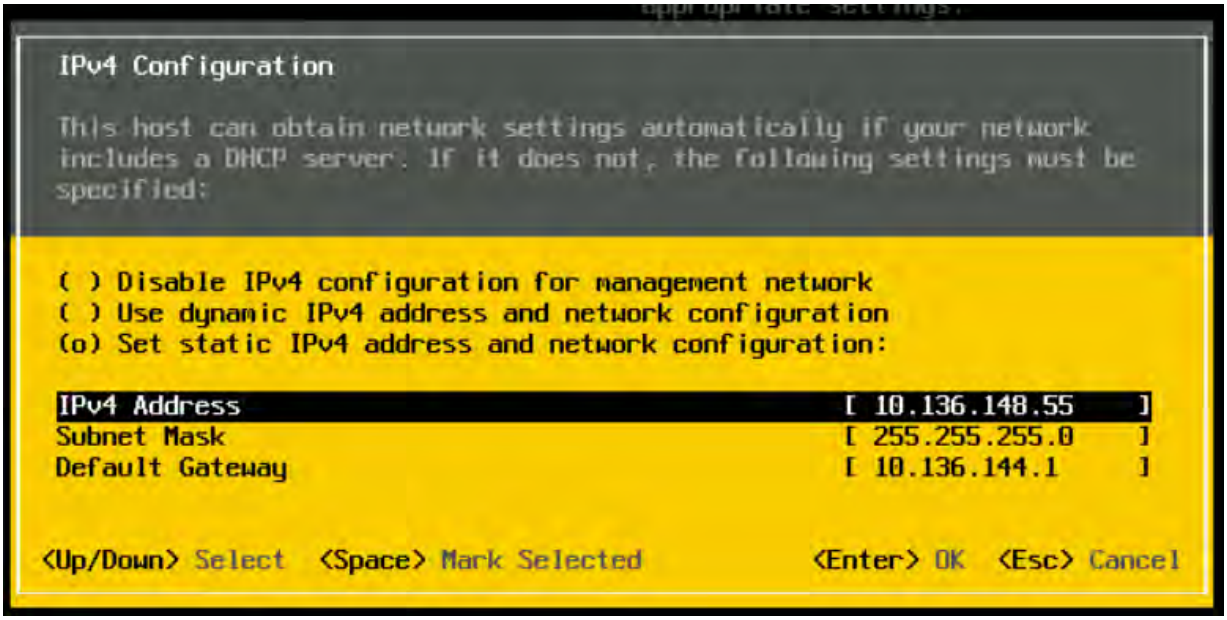
4. ESXi displays the **Configure Management Network** window. Press **Enter** to select the **Network Adapters**.



5. In the **Network adapters** window, select the adapter as the default management network with the arrow keys, then press **Enter**.



6. Select **IPv4 Configuration** from the **Configure Management Network** window and Press **Enter**.
7. In the **IPv4 Configuration** window. Select **Set static IPv4 address and network configuration** with the arrow keys:
   - Enter the IPv4 address, subnet mask, and default gateway in their respective fields.
   - Press **Enter** when finished reviewing to apply the new management network settings.

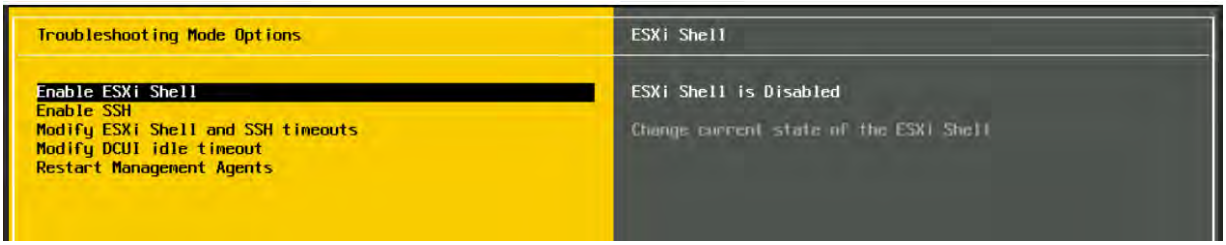


8. In the **Confirm Management Network** window, Press the **Y** key to confirm your selection.
9. ESXi displays the **DNS Configuration** window.
   - Add the primary and (if available) secondary DNS server IP address(es) in their respective fields.
   - Enter the vSphere host's hostname in the Hostname field, then press **Enter.**
10. Select the **Test Management Network** tab on the main ESXi screen to open the **Test Management Network** window.

    Perform the following tests:
    - Ping the default gateway.
    - Ping the DNS server.
    - Resolve any connectivity issues.
11. When you have completed testing, return to the main ESXi screen and select **Troubleshooting Options.**

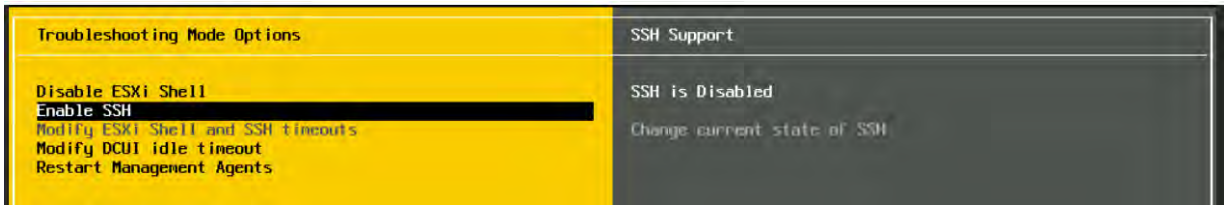
    ESXi displays the **Troubleshooting Mode Options** window:
    - Enable the ESXi Shell by selecting **Enable ESXi Shell** and pressing **Enter**.

      The window on the right displays the shell's status.



12. Enable SSH by selecting **Enable SSH** and pressing **Enter**.

    The window on the right displays the SSH status.



13. This concludes the ESXi configuration. The ESXi host is accessible by the IP set in the Management Network in this section.

Chapter 3. Installing VMware vCenter Server

This chapter covers installing VMware vCenter Server, including:

- Installing VCenter Server Appliance
- Adding Licenses to Your vCenter Server
- Adding a Host
- Setting the NTP Service on a Host
- Setting a vCenter Appliance to Auto-Start
- Mounting an NFS ISO Data Store

Review the General Prerequisites in section 1.5.4 before proceeding with these installations.

Note: This deployment guide assumes you are building an environment for a proof of concept. Refer to VMware best practice guides before building your production environment.

3.1 Installing vCenter Server Appliance

3.1.1 About VCSA

The VCSA is a preconfigured virtual appliance built on Project Photon OS that allows you to manage multiple ESXi hosts and perform configuration changes from a single pane of glass. Since the OS was developed by VMware and accelerated by NVIDIA vGPUs, it offers better performance and boot times than the previous Linux-based appliance. Furthermore, it uses an embedded vPostgres database, giving VMware full control of the software stack in tandem with the performance of NVIDIA certified servers. This results in significant optimization for vSphere environments and quicker release of security patches and bug fixes, enabling IT admins to focus on organizational goals and strategic initiatives.

The VCSA scales up to 2500 hosts and 45,000 virtual machines. Features such as Update Manager are bundled into the VCSA, file-based backups and restores, and vCenter High Availability. The appliance also saves operating system license costs and is quicker and easier to deploy and patch.

Note: A couple of releases ago, the VCSA reached feature parity with its Microsoft Windows counterpart and is now the preferred deployment method for vCenter Server.

- **Software Considerations:**
  - As of vSphere 7.0, deploying a new vCenter Server or upgrading to vCenter Server 7.0 requires using the vCenter Server Appliance (VCSA). The new vCenter contains all Platform Controller services. Review the vCenter Server Components and Services guide for more details about the VCSA authentication services and other installed services with the new vCenter appliance.
  - VCSA must be deployed to an ESXi host running version 6.5 or above. In addition, all the hosts you intend to connect to VCSA should be running ESXi 6.5 or above. Hosts running v6.0 and earlier cannot be managed by the version 7 vCenter Server Appliance, and there is no direct upgrade path.
  - You must check the compatibility of any third-party products and plugins used for backup, virus protection, monitoring, etc., as they may need upgrading for ESXi compatibility.
  - To check version compatibility with other VMware products, see the Product Interoperability Matrix.
  - For more information, please refer to vSphere Software Requirements

- **Architectural Considerations:**
  - When implementing a new vSphere 7 environment, you must plan its topology by the vSphere Installation and Setup guide.
  - This guide uses the new vSphere 7 VCSA, which deploys the vCenter Server, vCenter Server components, and the authentication services on one system.

Note: All Platform Services Controller services are consolidated into vCenter Server, simplifying deployment and administration.

- **Hardware and Storage Requirements:**
  - See the document Hardware Requirements for specifications. During installation, the corresponding size you select will determine the number of CPUs and the amount of memory (disk can be thin provisioned).
  - See the document Storage Requirements for specifications. Storage requirements for the smallest environments start at 250 GB and increase depending on your specific database.
  - The ESXi host you deploy the VCSA on must not be in lockdown or Maintenance Mode.
  - All vSphere components must be configured to use an NTP server. The installation may fail, or the vCenter Server Appliance VPXD service may not start if the clocks are not synchronized.
  - FQDN resolution must be enabled when you deploy the vCenter Server.
  - Required Ports for vCenter Server.
  - vSphere VMware Configuration Maximum tool.

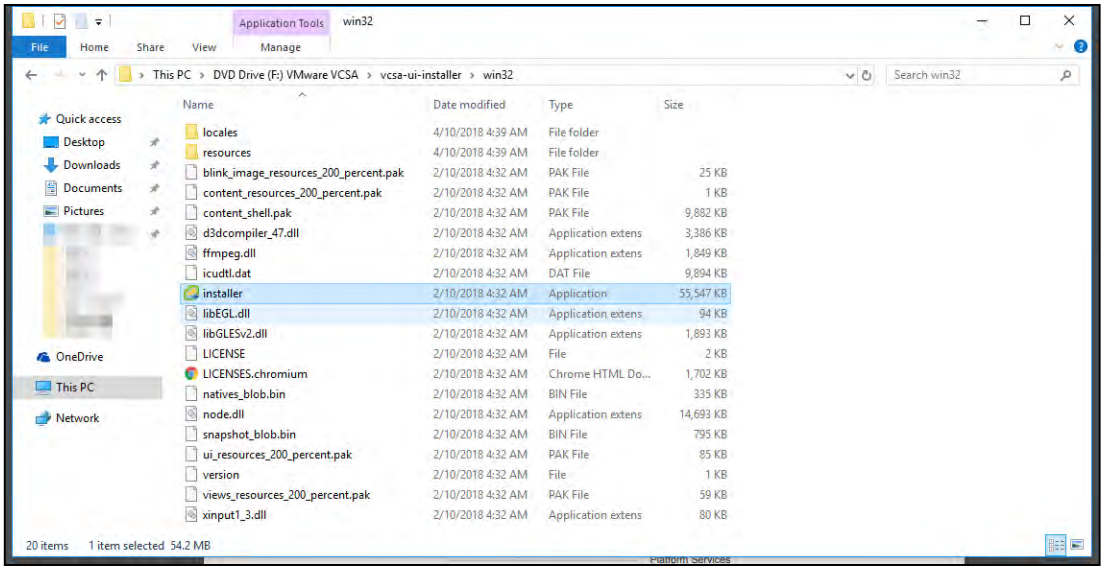
3.1.2 vCenter Server Appliance (VCSA) Installation

Download the latest VMware vCenter Server Appliance ISO from VMWare downloads.

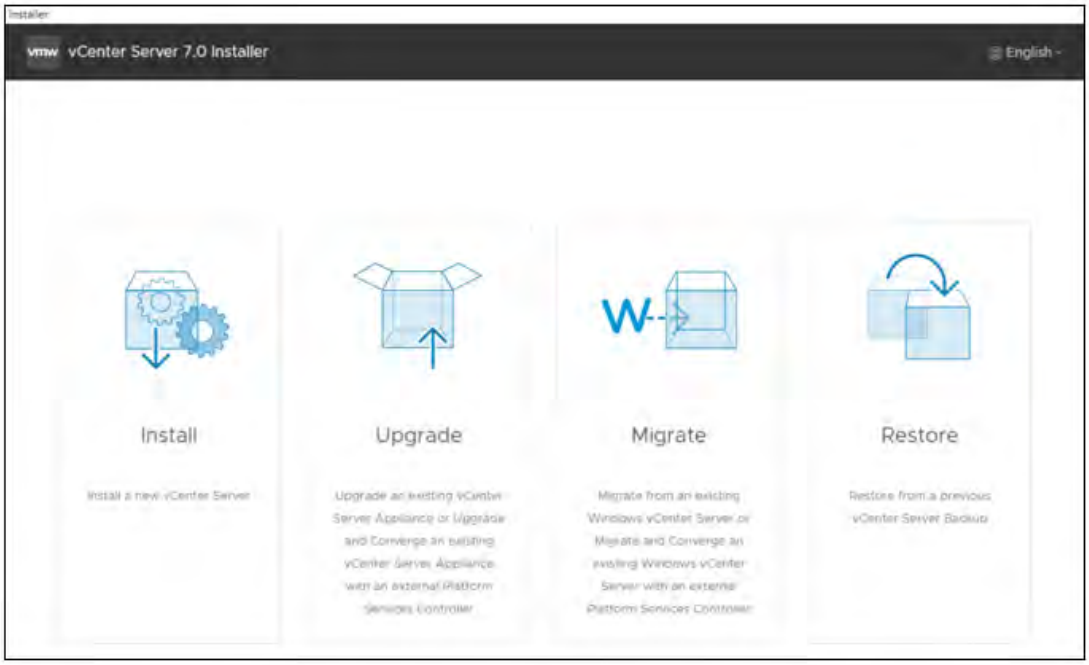
1. Mount the ISO on your computer.

   The VCSA installer is compatible with Mac, Linux, and Windows.

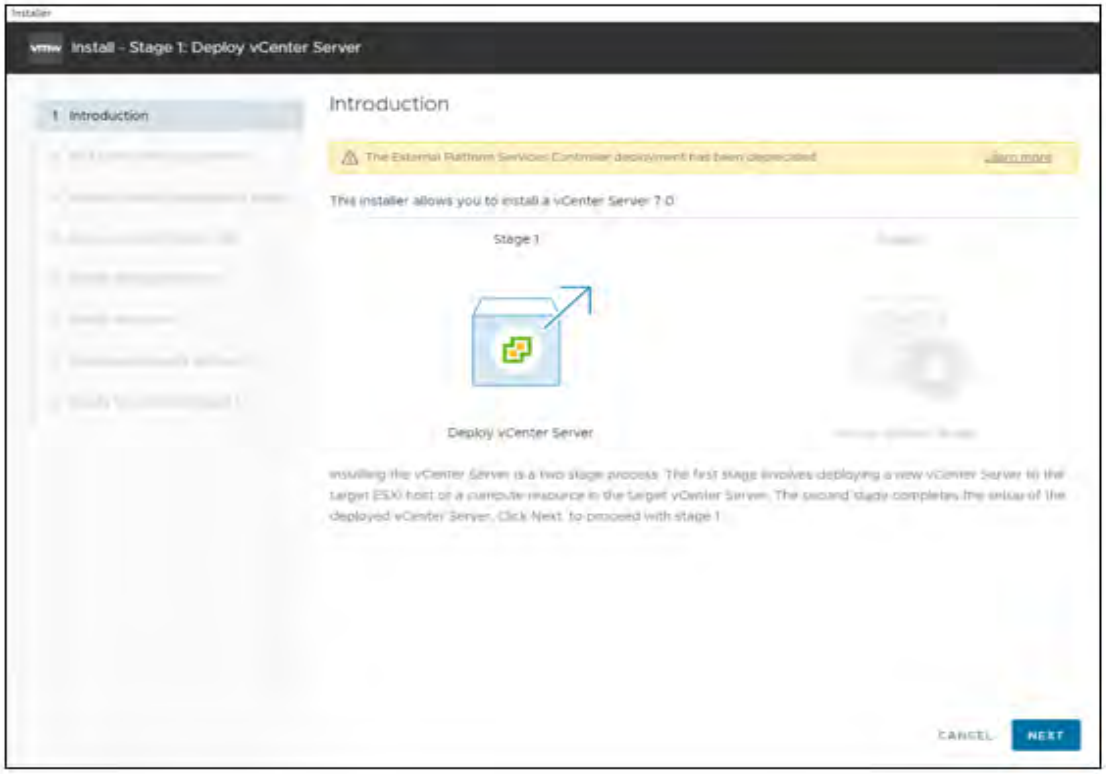
2. Browse to the corresponding directory for your operating system, e.g., \vcsa-ui-installer\win32: Right-click **Installer** and select **Run as administrator.**



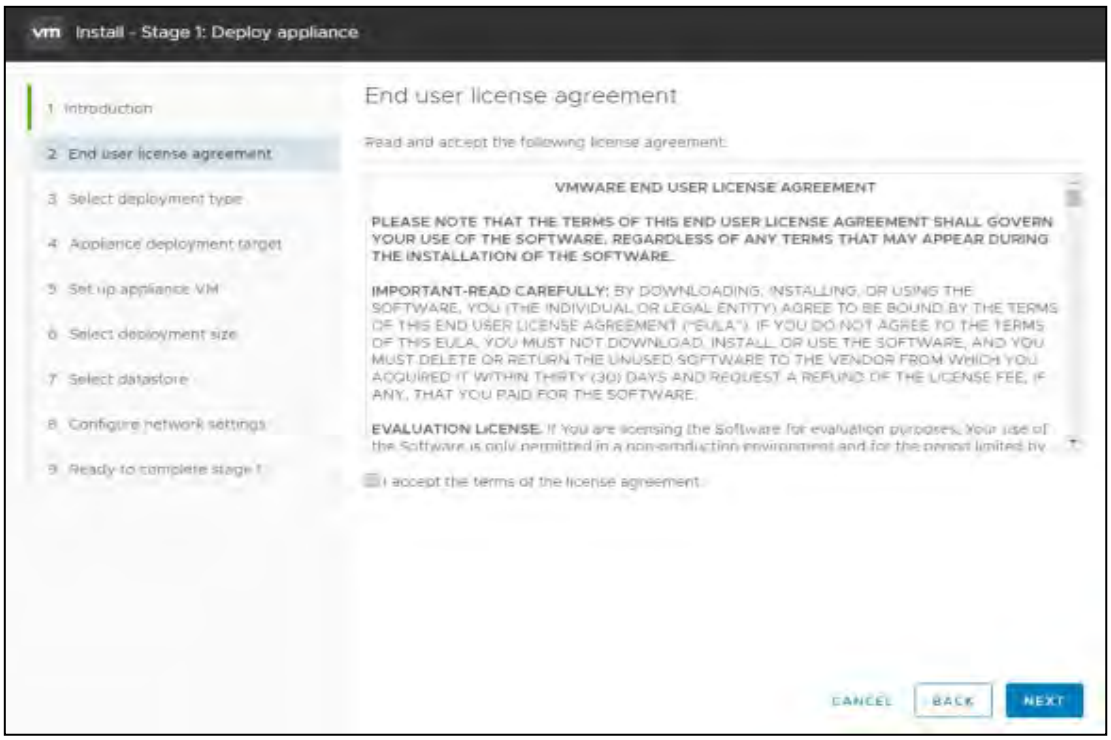
3. As we are installing a new instance, click **Install.**



4. The installation is split into two stages, and we will begin with deploying the appliance. Click Next to begin.

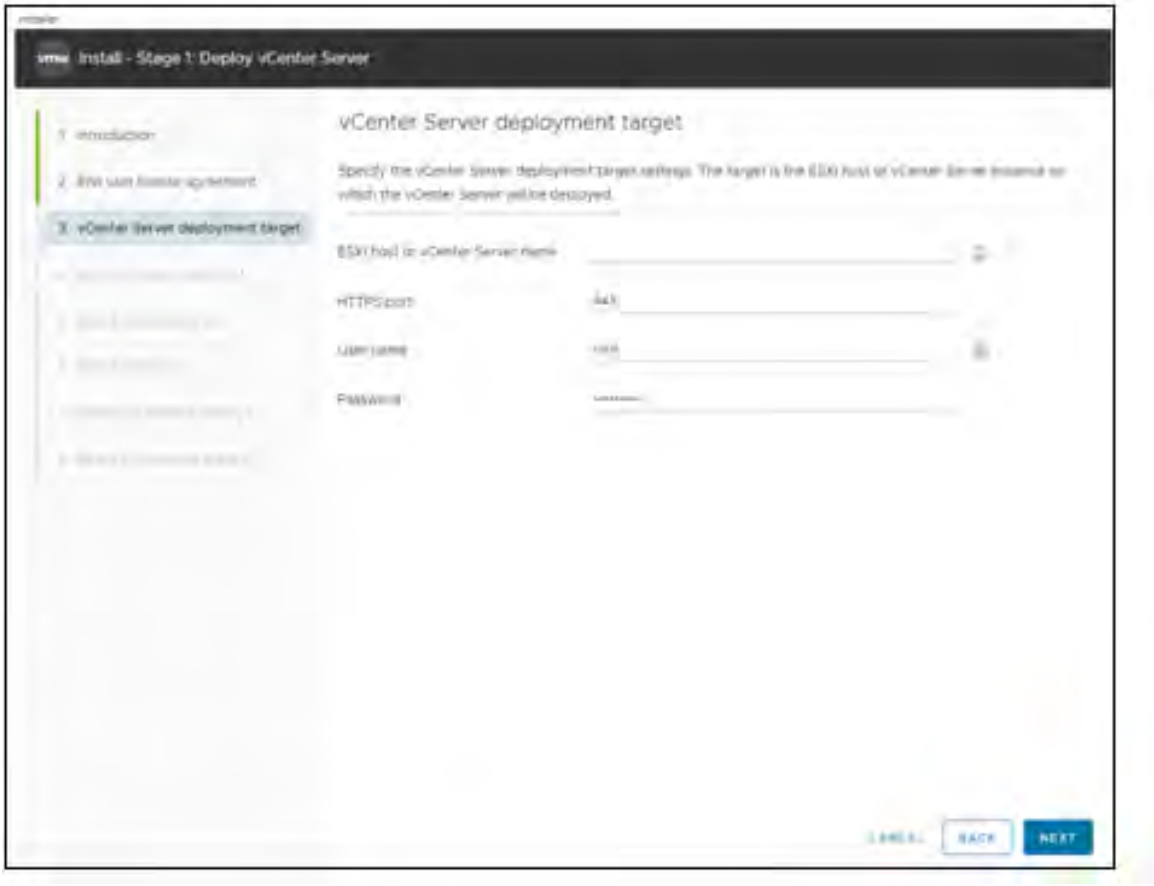


5. Read and accept the EULA, then click **Next** to continue.

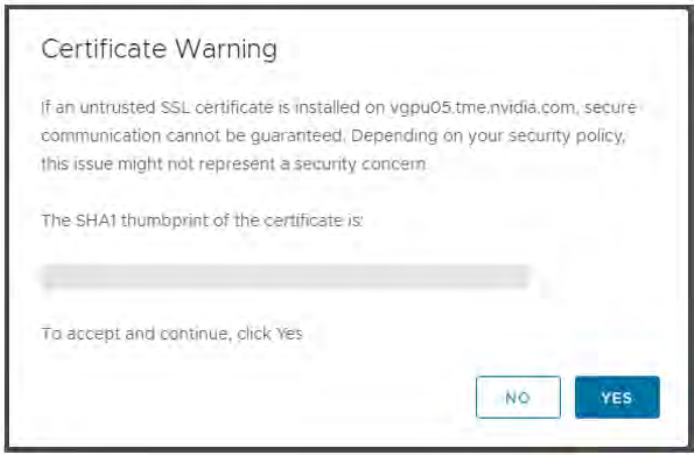


6. Select the ESXi host to install the VCSA as a guest. This must be a host that runs ESXi 6.5 or later. NVIDIA recommends that the vCenter Server Appliance run on a separate management cluster from the one designated for VDI workloads.

Enter the IP address or fully qualified domain name (FQDN) of the chosen host, then its root username and password, and click Next:

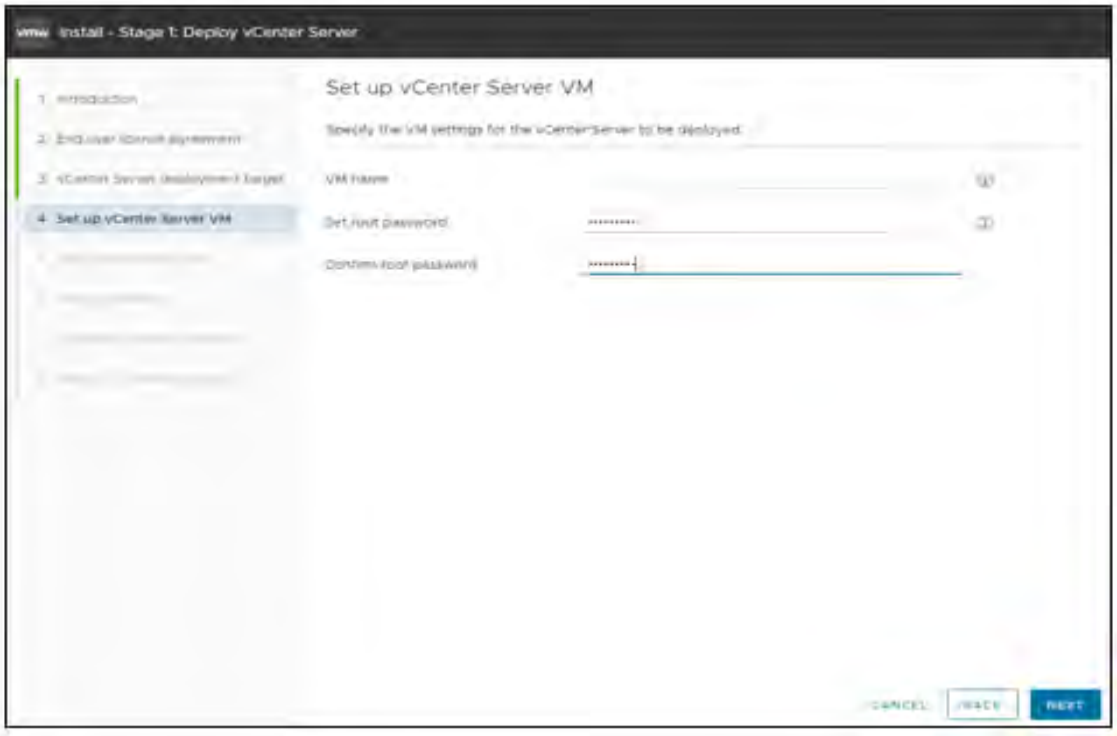


7. If your desktop can reach the host, you should see a certificate warning as it connects. This warning is due to the use of a self-signed certificate. If you use a signed certificate, you will not see this warning. Click **Yes** to continue.

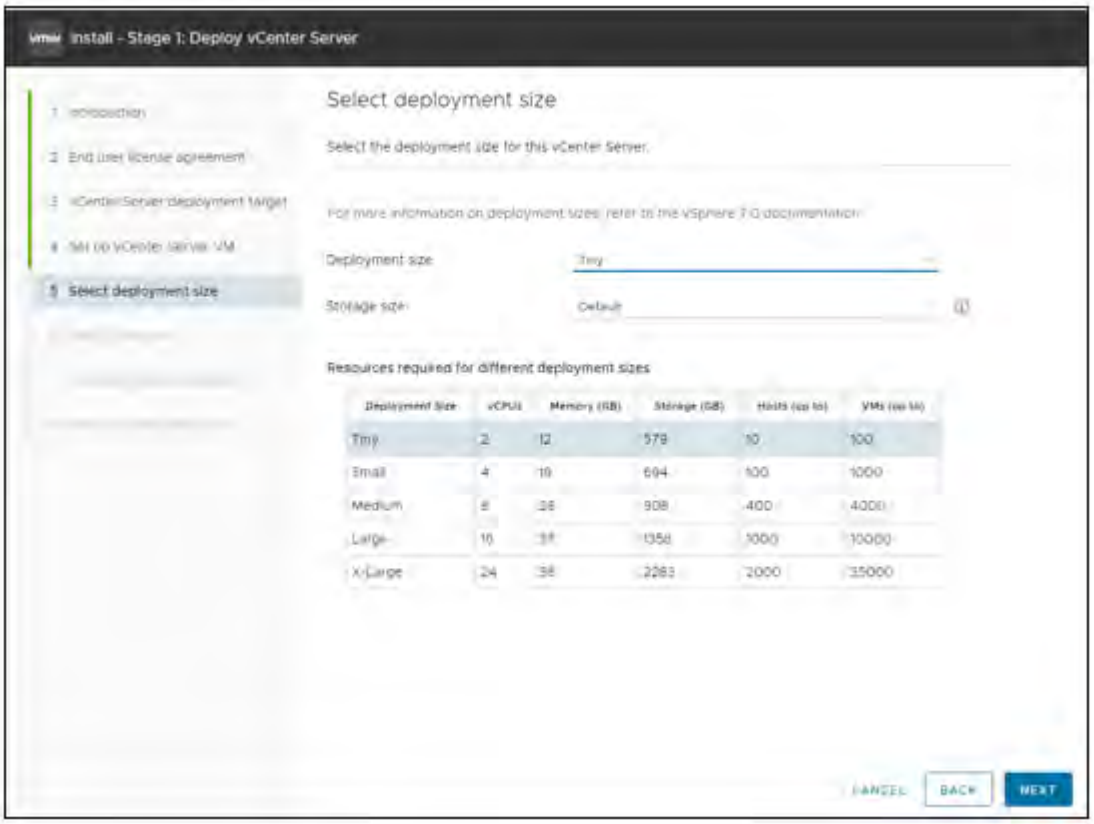


The credentials you provided are validated:

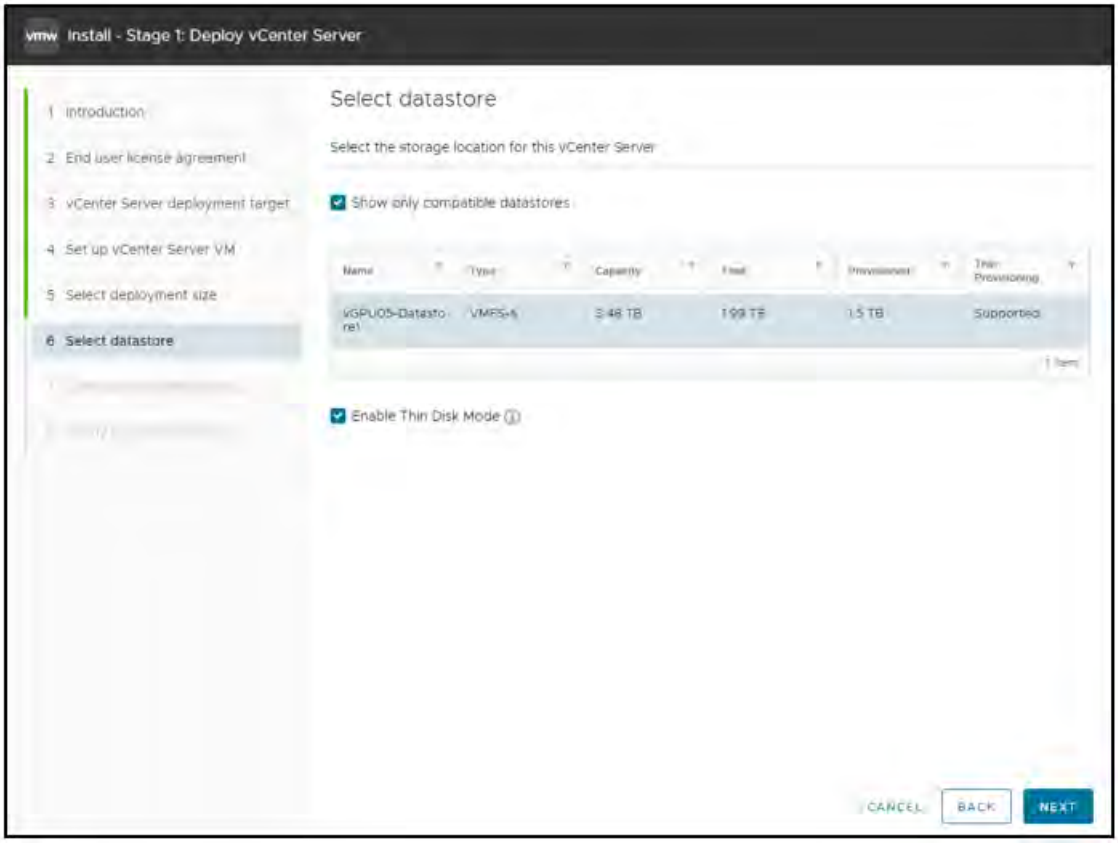
8. When prompted after a successful connection, name the appliance, enter a root password for the appliance, confirm the root password, and click **Next**.



9. Select a deployment size appropriate to the number of hosts and virtual machines that the vCenter Server will manage, then click **Next**.

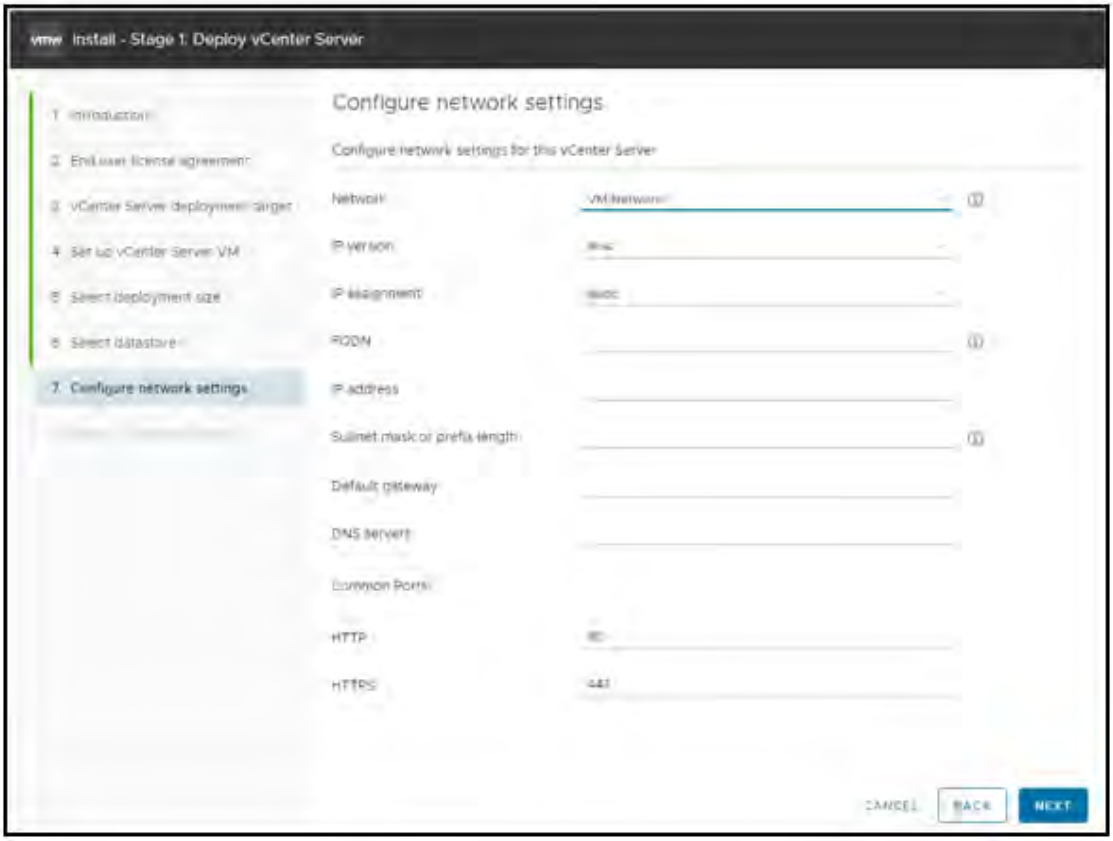


10. Select the datastore where the VCSA will be deployed, select thin provisioning if needed, then click **Next**.

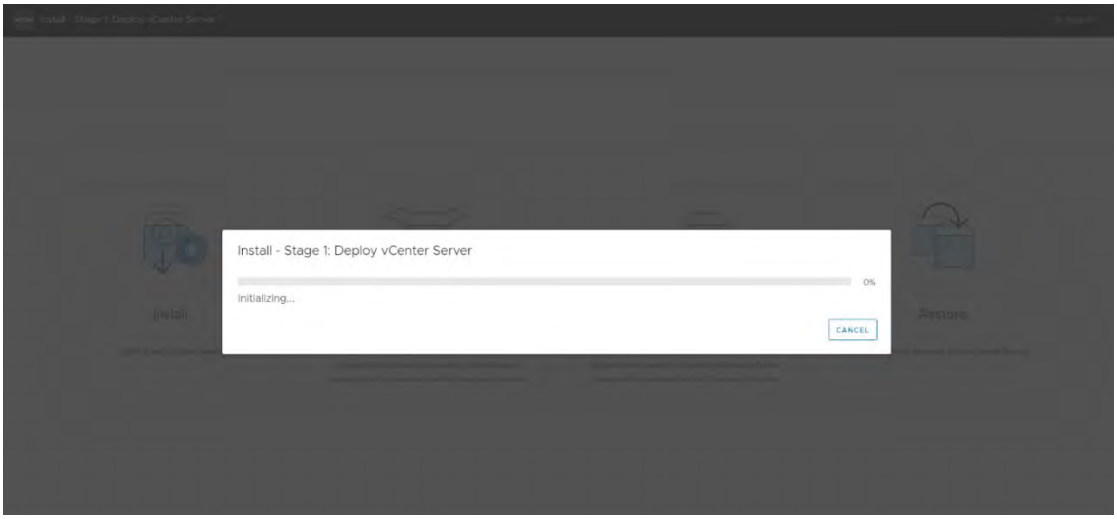


11. The installer displays the *Configure network settings*. Before configuring these settings, please choose an appropriate static IP address and enter it into local DNS (e.g., on the Domain Controller). Once you can resolve the address, enter the IP address hostname on the network setting page, then scroll down and enter the remaining items.

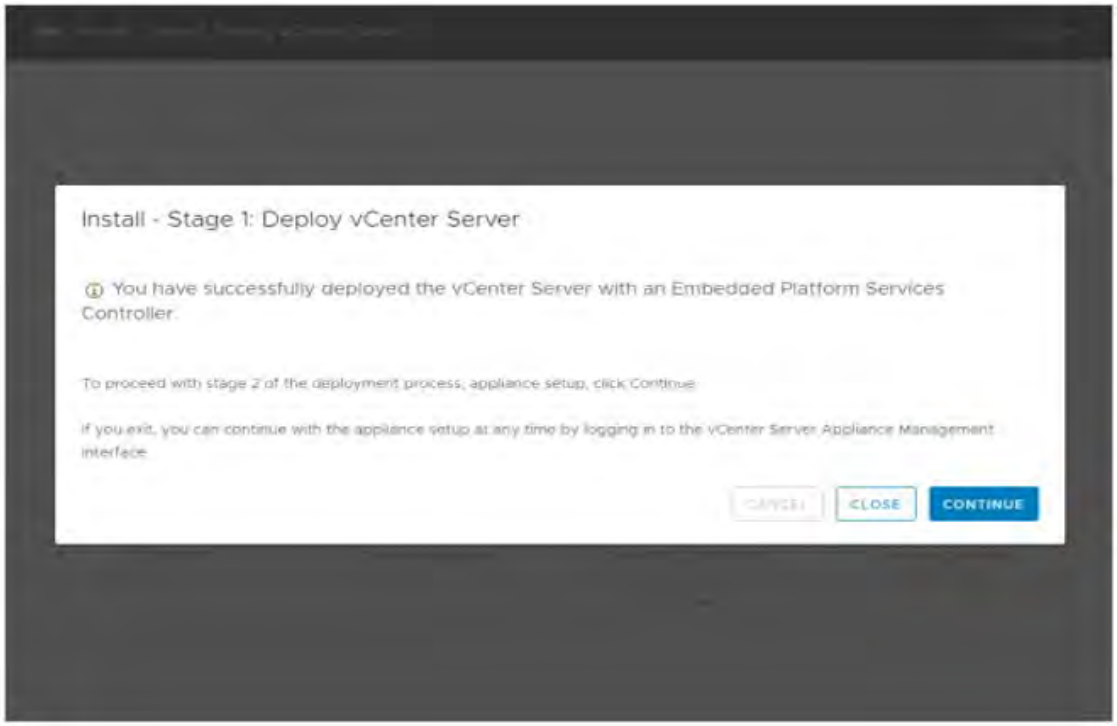
    When all desired settings are complete, select Next.



12. The installer displays a summary page. Click **Finish**. The installer deploys the appliance.

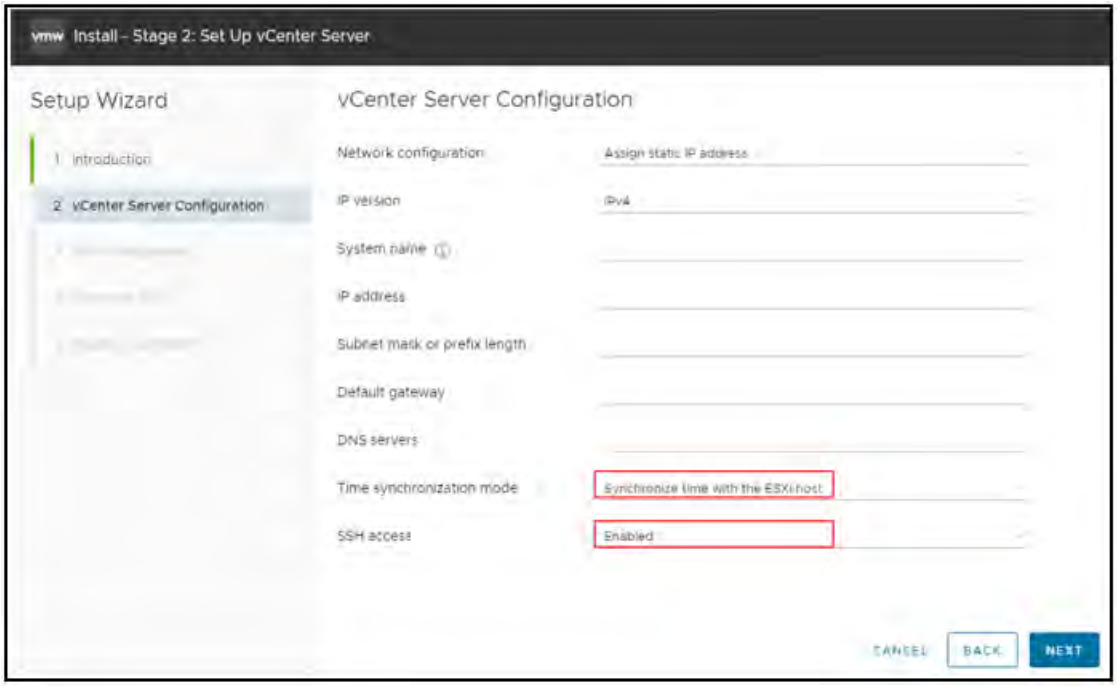


13. With the VCSA now deployed, move on to stage 2 by clicking **Continue**.



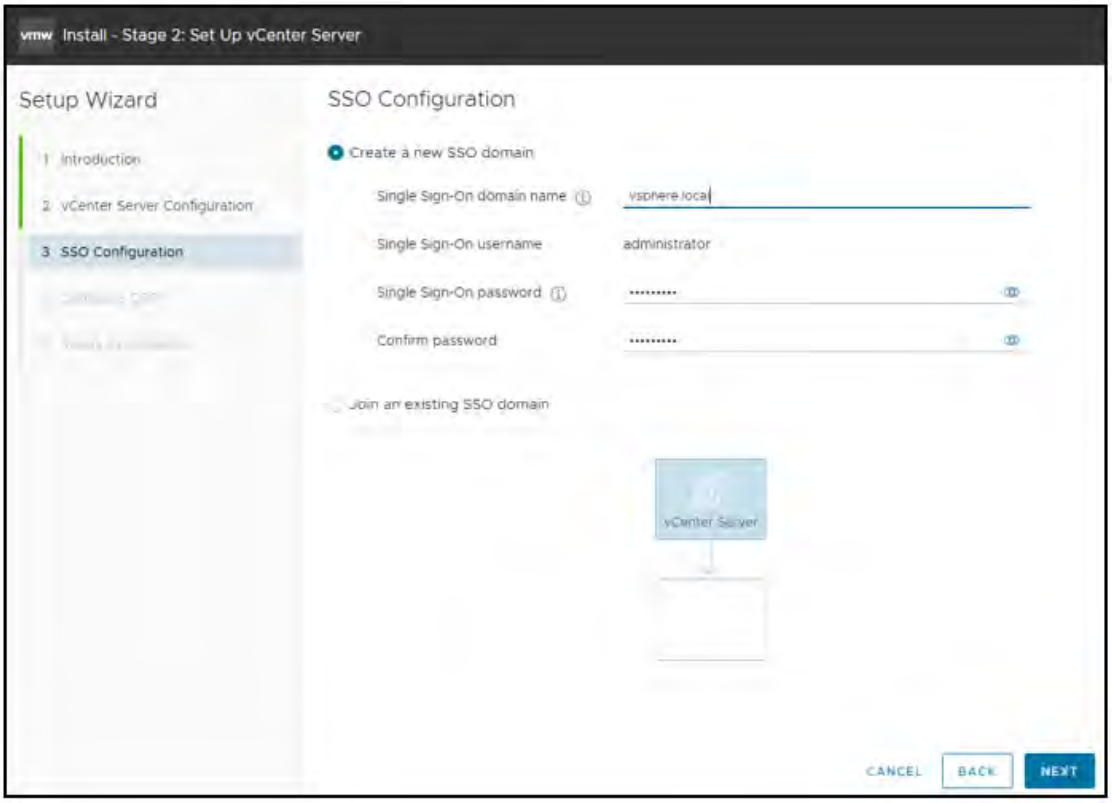
14. Select **Next** to begin Stage 2 of the VCSA setup.

15. Configure the NTP servers, enable SSH access if required, and click **Next**.

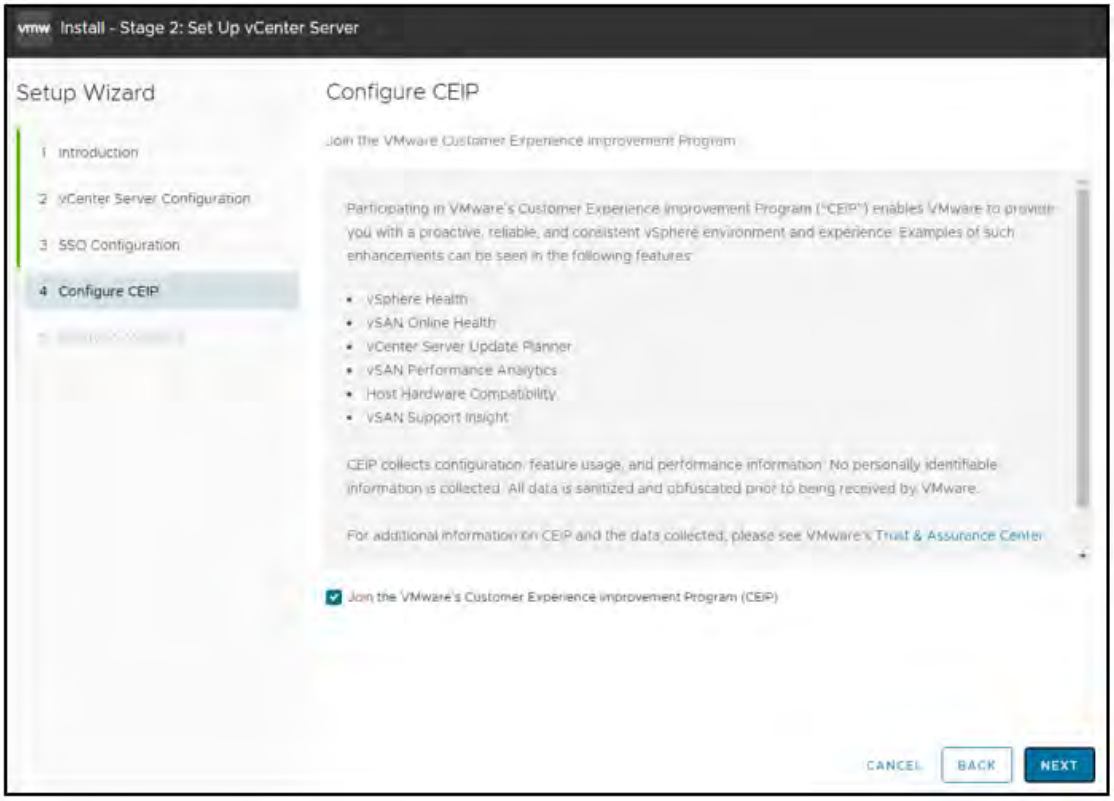


16. Enter a unique SSO domain name, configure a password for the SSO administrator, click **Next**.

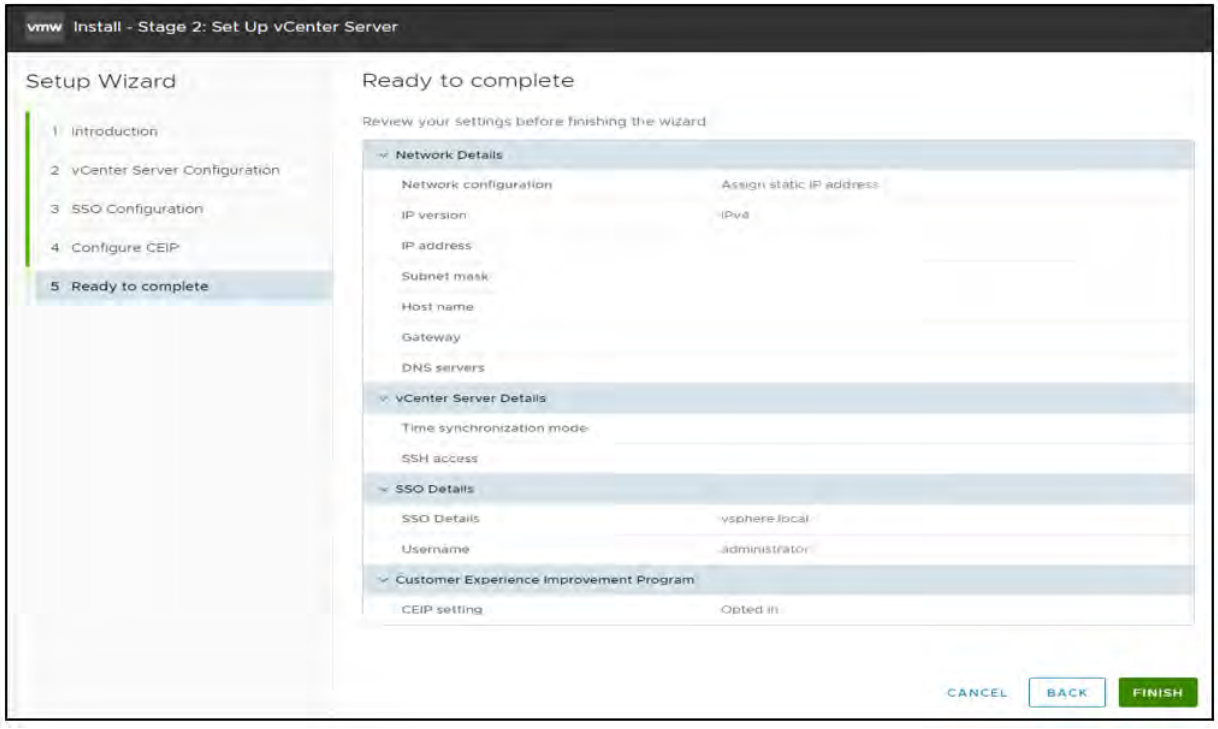
    The default SSO domain name is vSphere.local. The SSO domain name should not be the same as your Active Directory Domain.



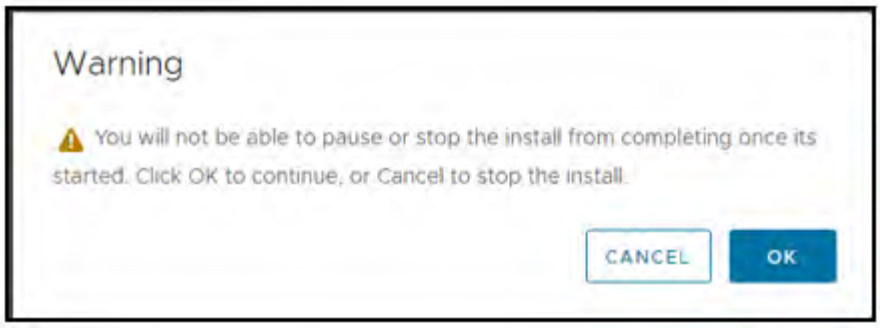
17. Select or deselect the customer experience improvement program box and click **Next**.



18. Review the details on the summary page and click **Finish**.



19. The installer displays a warning that you cannot pause or stop the install once you start. Click **OK** to acknowledge the warning and start the install.



20. When the install process is complete, click **Close** to exit the installer and complete Stage 2 of the VCSA setup.

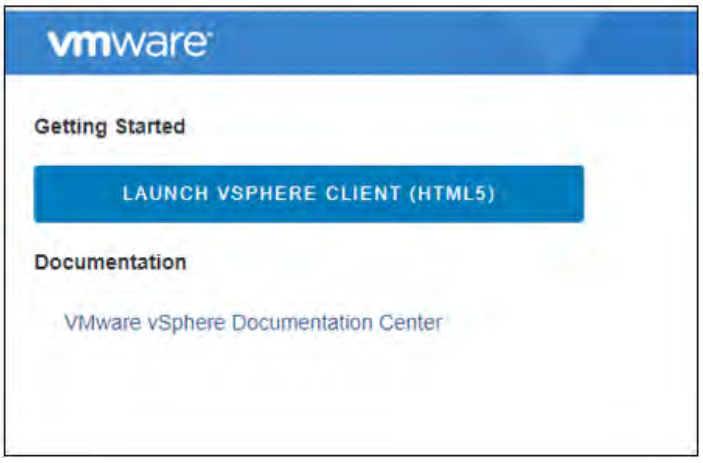
3.2 Post Installation

This section describes the post-install and configuration of VMware's vSphere VCSA.

3.2.1 Adding Licenses to Your vCenter Server

Use the following procedure to configure vCenter:

1. Connect to the vCenter post install using the IP or FQDN of the vCenter. Access vSphere by clicking **Launch vSphere Client (HTML5)**.



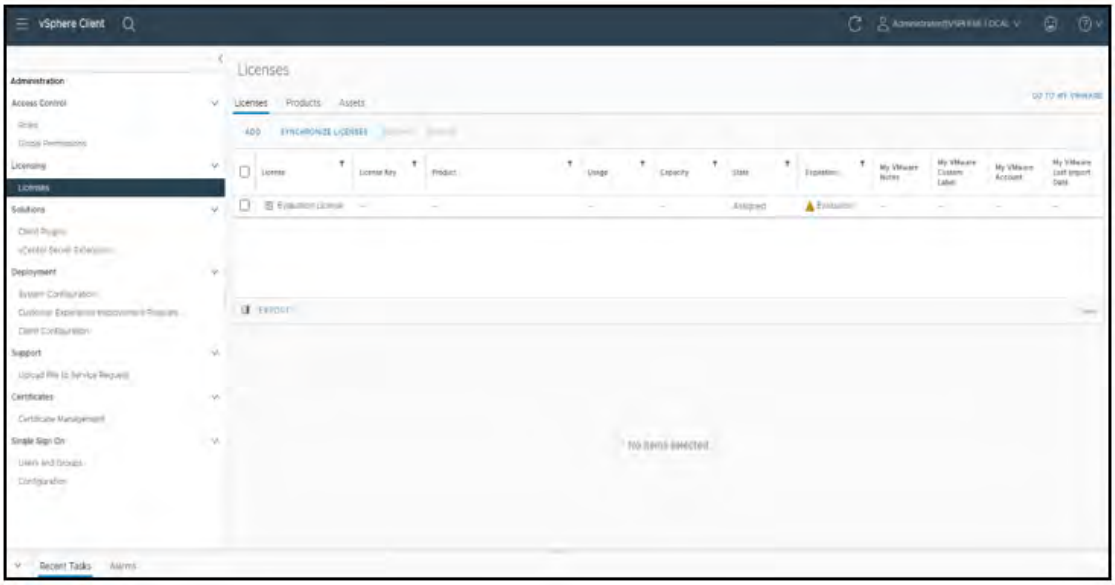
2. The **VMware Single Single-On** page displays:

Enter the username and password you specified during installation. Click the **Login** button to continue.

3. The **VMware vSphere Web Client** page displays:
   You must apply for a new vCenter license key within 60 days (about two months). If you have purchased vCenter Server, log in to your VMware vSphere License portal. Select your license and log in to the vSphere Web Client using the SSO administrator login (If the license key does not appear, check with your VMware account manager).

14. Click the **Menu drop-down**, then click **Administration**.

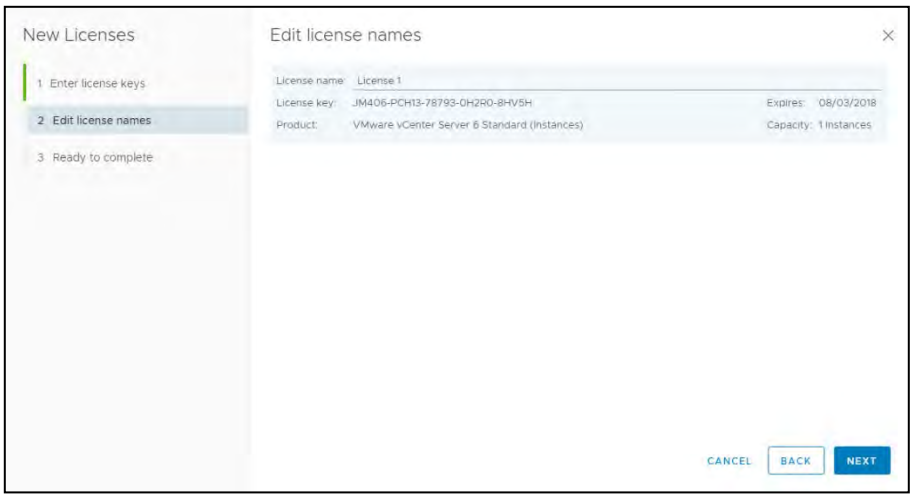
    Select Licenses from the left-hand menu, then select the Licenses tab.



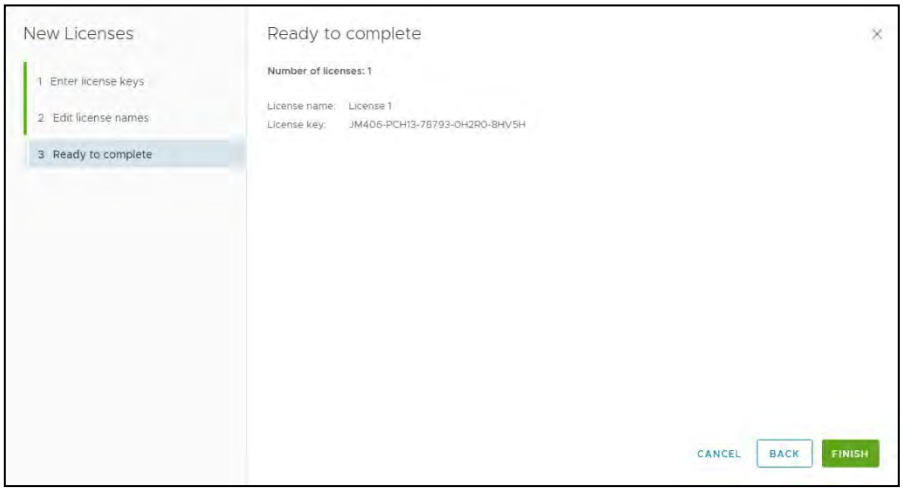
4. Click **Add New Licenses** to open the **New Licenses** window.

5. Enter your vCenter Server Standard license key from the VMware vSphere License portal. Login required.

6. Enter a unique name for the license in the **License Name** field. Click **Next.**

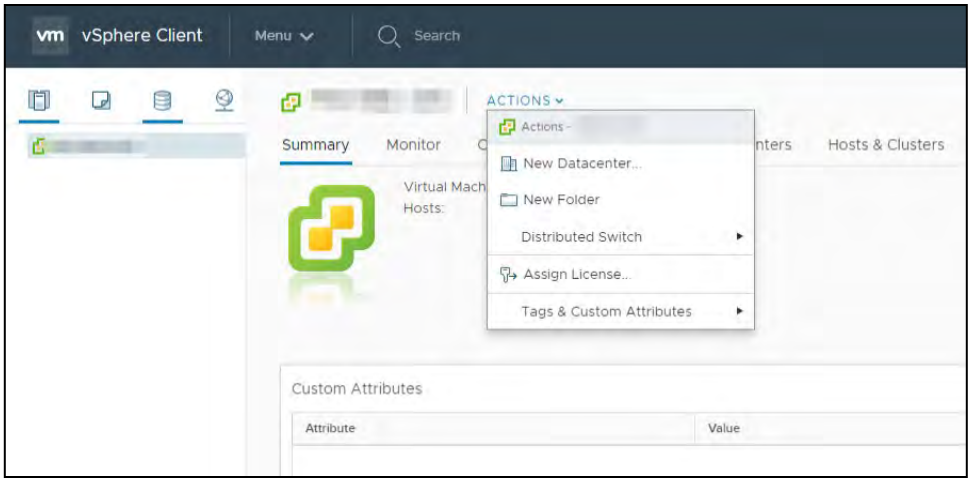


7. Review your selections and click **Finish** to close the **Enter New License** window and return to the **VMware vSphere Web Client** page.



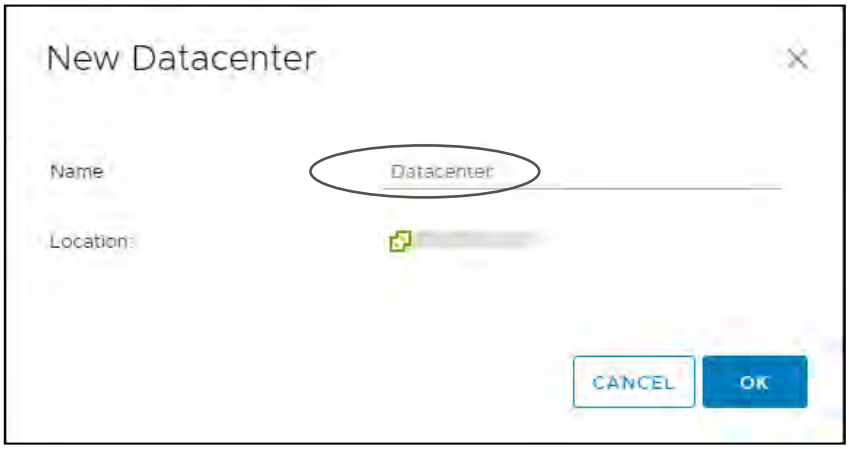
3.2.2 Adding a Host

1. Use the following procedure to add a host in vCenter:
   Select the **Home** icon (house) on the *VMware vSphere Web Client* page.
   Select **Hosts and Clusters**.
   Click the **ACTIONS** drop-down list and select **New Datacenter.**



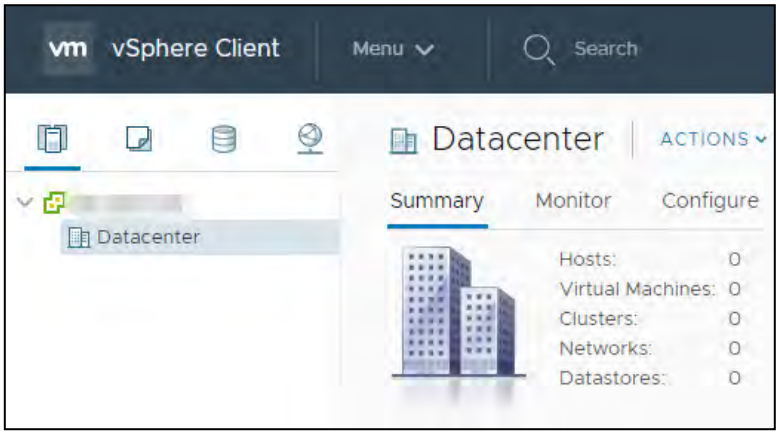
2. The **New Datacenter** window displays.

   Enter a name for the data center in the Datacenter Name field and click OK.



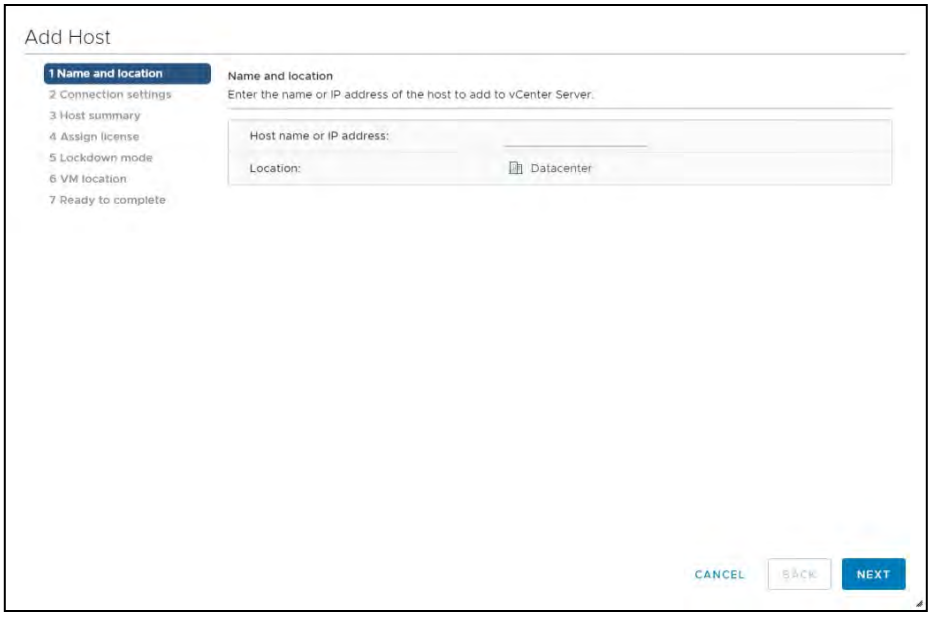
3. The new data center is visible in the left panel of the *vSphere Web Client*.

   Click the ACTIONS drop-down and select Add a Host.



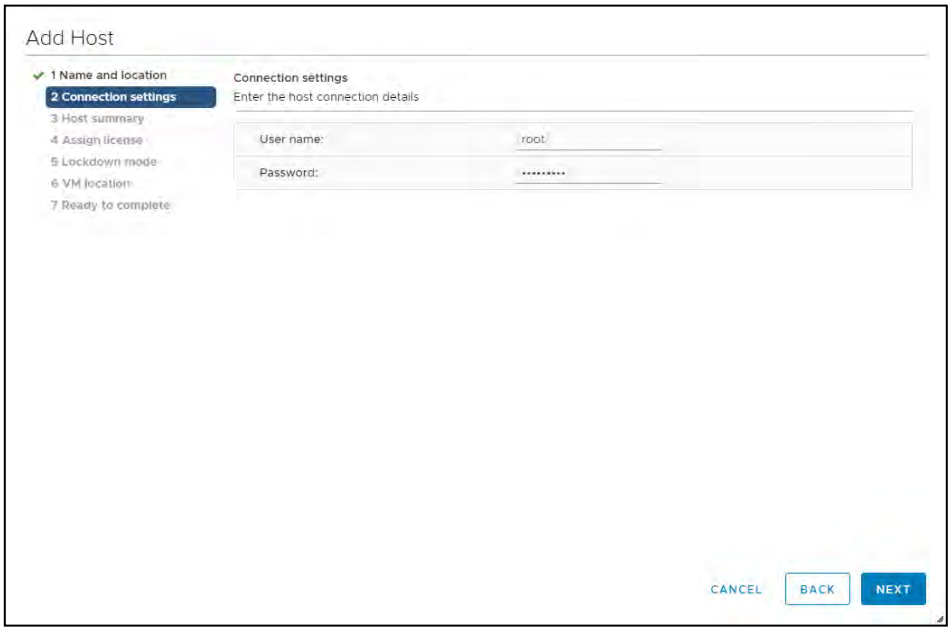
4. The **Add Host** window box opens.

   Enter the hostname or IP address of the vSphere host and click Next.



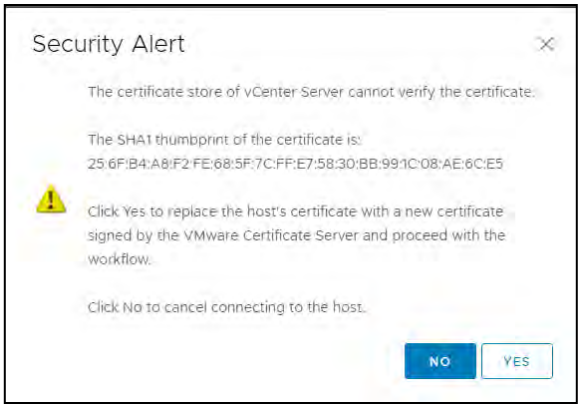
5. The **Connection settings** window box displays.

   Enter the administrator account credentials in the Username and Password fields and click Next.



6. The **Security Alert** window displays.

   Click Yes to replace the host certificate.



7. The **Host summary** window displays.

   Review the settings and click Next to proceed.

8. The **Assign license** window displays.

   Confirm the license selection and click Next.

8. The **Lockdown mode** window displays.

   Accept the default setting (Disabled) and click Next.

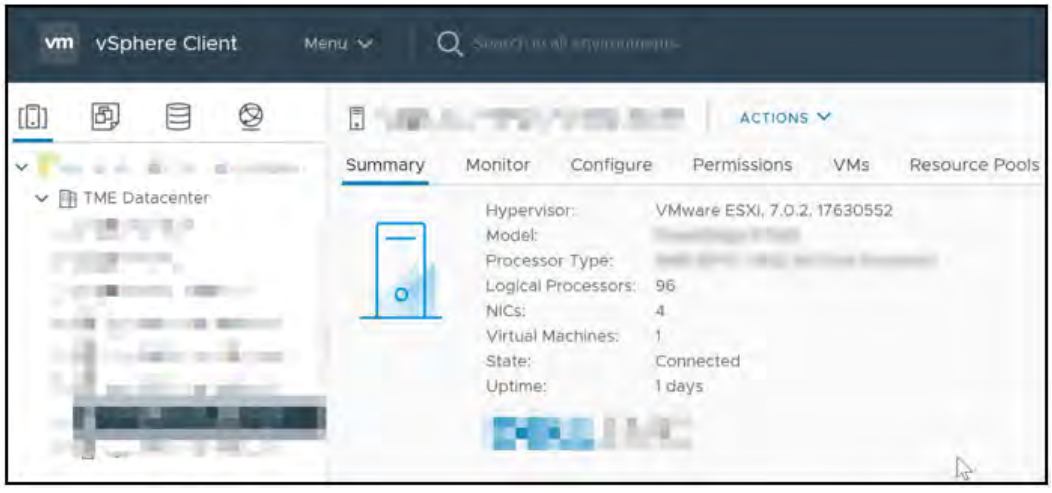
9. The **VM location** window displays.

   Select a cluster or accept the default option and click Next to continue.

10. The **Ready complete** window displays.

    Click Finish to complete adding the new host.

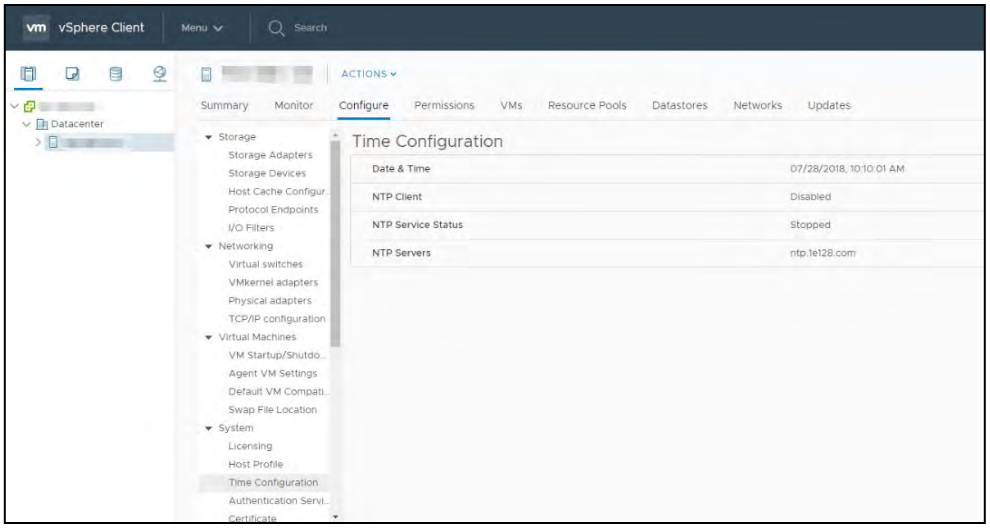
11. The new host is now visible in the left-hand panel when clicking the datacenter name.



3.2.3 Setting the NTP Service on All Hosts

Set the NTP (Network Time Protocol) service on each host to ensure that all guests' time synchronization is accurate.

1. Click a host object in the menu on the left, click **Configure**> **System**> **Time Configuration**> **Edit**.

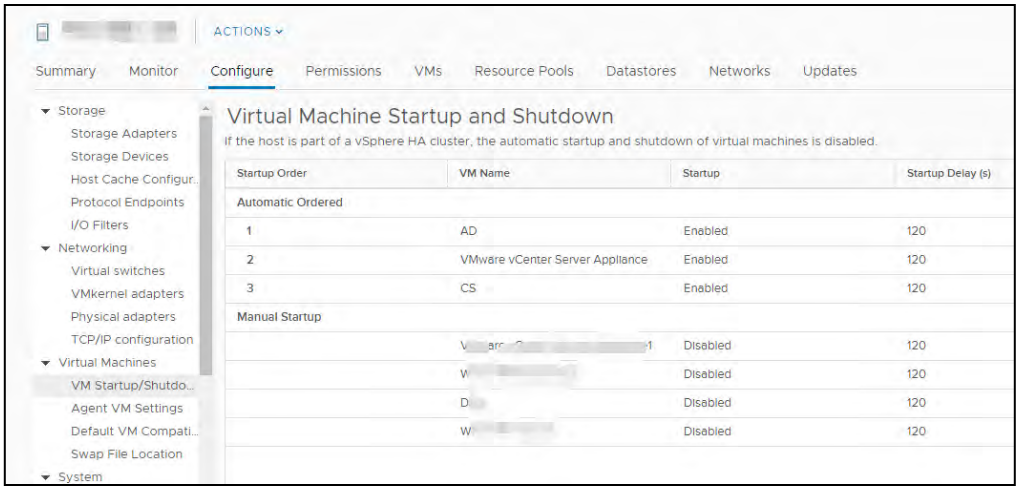


2. Enter a valid NTP time server and click **OK.**

3.2.4 Setting a vCenter Appliance to Auto-Start

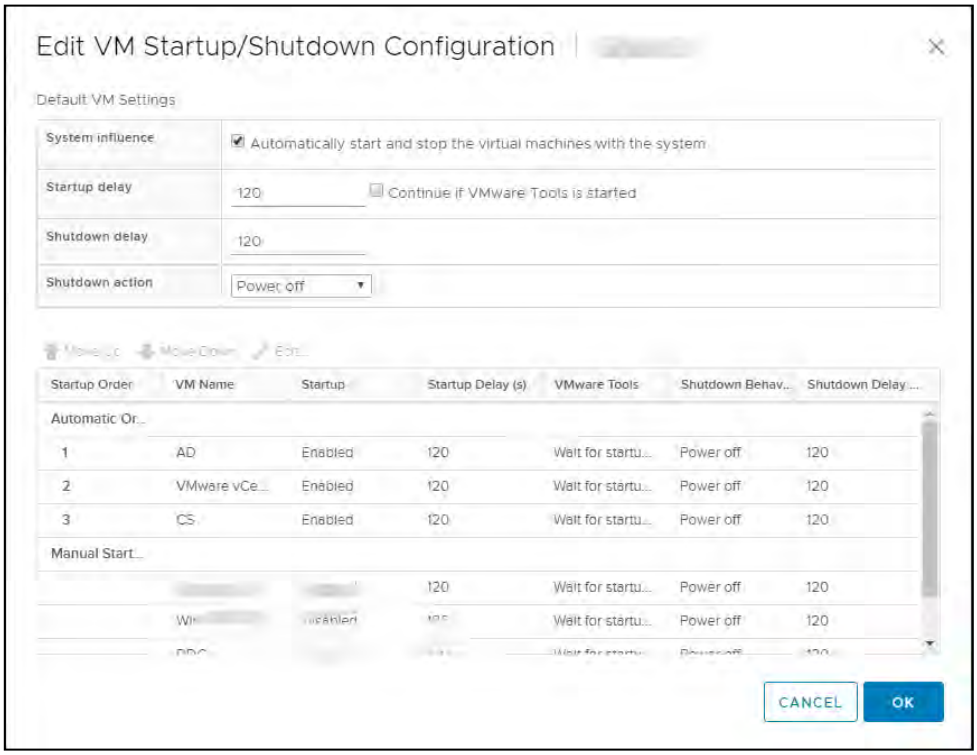
Use the following procedure to set a vCenter Appliance to start automatically:

1. Select the host in the vSphere Web Client and then select **Configure**> **Virtual Machines**> **VM Startup/Shutdown.**



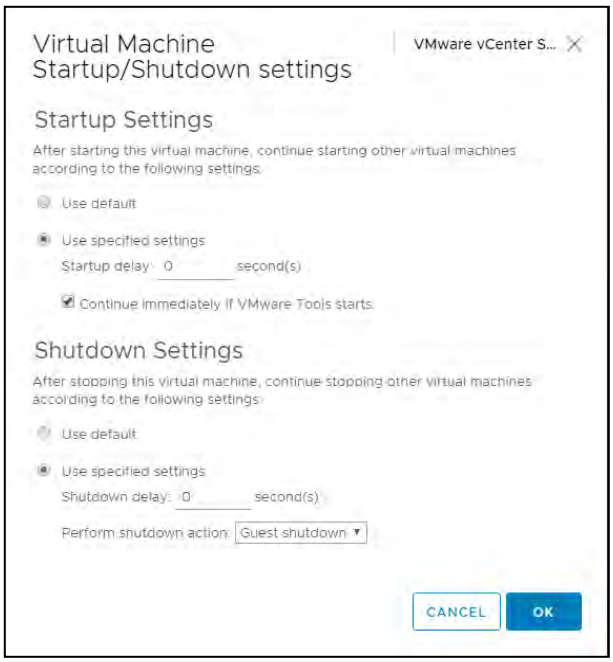
2. Click the **Edit** button.

   The **Edit VM Startup and Shutdown** window displays.



3. Select **vCenter Appliance**, then press the **Up Arrow key** to move that virtual machine up to the **Automatic Startup** section of the appliance table. Then click the **Edit** button.
4. Select and set the following options:
   - Set Startup Behavior to Use specified settings
   - Select Continue immediately if VMware Tools starts
   - Set Startup Delay to 0
   - Set Shutdown Behavior to Use specified settings
   - Set Shutdown Delay to 0

- Set Perform shutdown action to Guest shutdown



5. Click **OK** to apply the configuration settings.

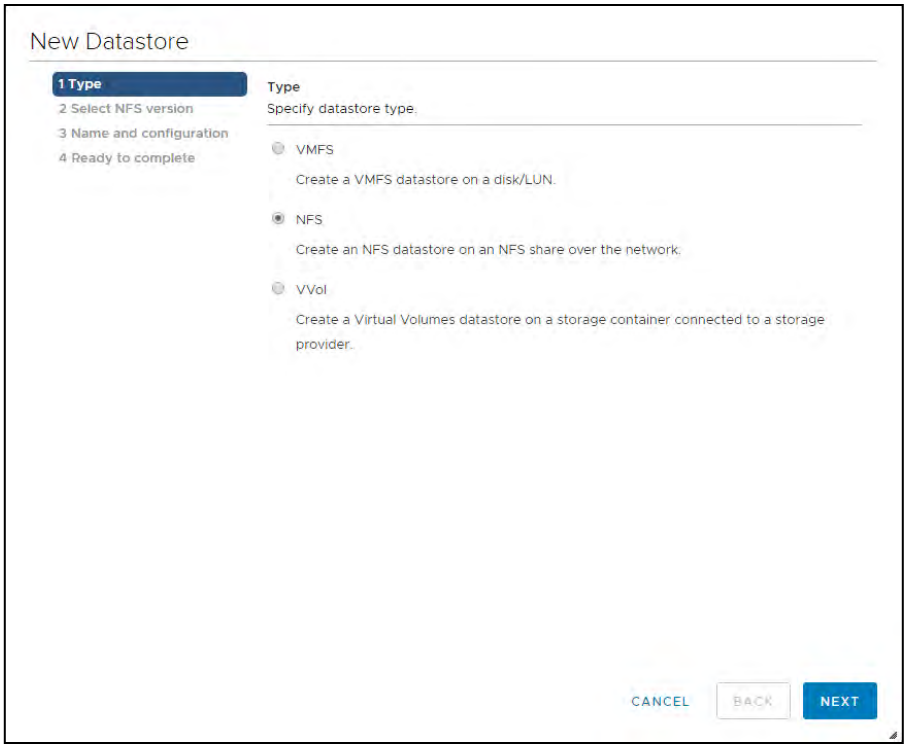
> Note: the vCenter Web Client may not reflect these configuration changes immediately. Either click the Refresh icon or a different configuration group and return to the current setting.

3.2.5 Mounting an NFS ISO Datastore

Use the following procedure to mount an NFS ISO datastore:

15. In the main *vSphere Web Client* window, select **Hosts and Clusters** and select the host.
16. Select **Storage** -> **New Datastore** from the **Actions** drop-down menu.

The *New Datastore* window displays with the Type tab selected.



3. Select **NFS** and click **Next** to proceed.

   The *Select NFS version* tab displays.

4. Select the correct NFS version and click **Next** to proceed.

   The *Name and configuration* tab displays.

5. Enter the NFS exported folder path and the NFS server address in the **Folder** and **Address** fields.

   a) Because the data store is an ISO data store, consider mounting it as read-only by checking the **Mount NFS** as read-only checkbox.

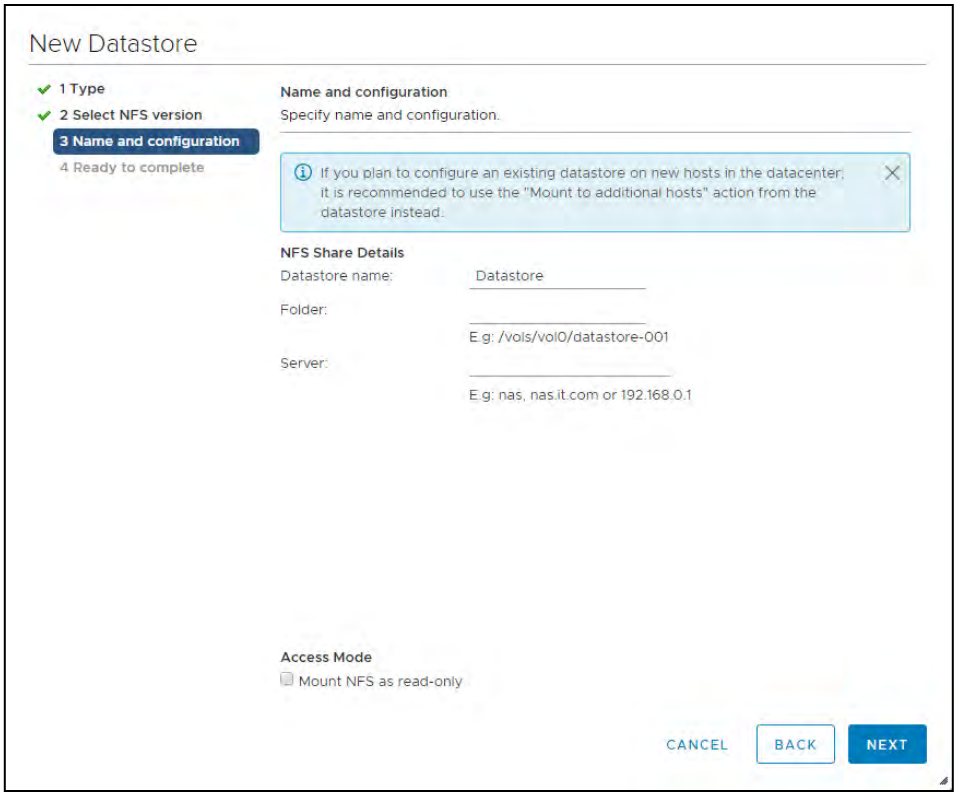
6. Click **Next** to proceed.

   The *Host accessibility* tab displays.

7. Select the host that will use the new data store.

8. Select **Next** to proceed.

   The *Ready to complete* tab displays.



17. **Review** the settings.
18. Click **Finish** to complete the procedure for adding the NFS ISO datastore.
0. This datastore is now accessible as an installation source for virtual machine CD drives.

Chapter 4. Building VMware Horizon

VMware Horizon® is a platform for managing and delivering virtualized or hosted desktops and applications to end-users. Horizon allows you to create and broker connections to Windows virtual desktops, Linux virtual desktops, and Remote Desktop Server (RDS) – enabling hosted applications and desktops.

This chapter covers installing VMware Horizon Connection Server and its supporting components, including:

- Prerequisites for VMWare Horizon Connection Server
- Installing Horizon Connection Server
- Registering the license
- Registering vCenter Server

4.1 Prerequisites for VMware Horizon Connection Server

Note: The Horizon Connection Server is not required for connections to NVIDIA vGPU VMs. The Horizon client can connect directly to VM's, desktops, and vApps using the View Agent Direct-Connection (VADC) plugin.

The following elements are required to install and configure VMWare Horizon Connection Server. For demonstration purposes, this guide will use the Horizon 8 2111 release.

- A valid license for VMWare Horizon, which can be registered here.

- Hardware Considerations:
  - Minimum: A 2.0 GHz processor or higher, 100 Mbps NIC, and 4GB RAM or higher
  - Recommended: 4 CPU's, 1 Gbps NICs, and at least 10 GB RAM for deployments of 50 or more desktops

- You must install all Horizon Connection Server installation types, including standard, replica, and enrollment server installations, on a dedicated physical or virtual machine that meets these specific hardware requirements.
- **Important:** The virtual machine that hosts Horizon Connection Server must have an IP address that does not change. In an IPv4 environment, configure a static IP address.
- **See also:** VMware Connection Server PreRequisites

- Architectural Considerations:
  - A VMWare Horizon supported operating system and Active Directory. The Connection Server host must not be a domain controller. Please refer to the VMWare knowledge base article for more information.
  - A compatible version of Horizon with versions of vCenter and ESXi. Please refer to the VMWare Product Interoperability Matrix for details.
  - When installing replicated Horizon Connector Server instances, you must configure the instances in the same physical location and connect them over a high-performance LAN.
  - A compatible web browser
  - A Domain User Account with Administrator Privileges
  - It is recommended that you prepare a data recovery password

- Software Considerations:
  - Remote Display Protocol and Software enables access to remote desktops and applications.
  - VMware's Blast Extreme is a feature-rich display protocol that enables clients to connect to remote desktops and published applications. The Blast Extreme display protocol connects to a wide range of end-client devices and maintains network continuity in cases of network hiccups on windows clients. Refer to the VMware Blast Extreme page for a complete list of VMware's Blast display protocol advantages.
  - Before creating the Connection Servers virtual machine template, run Sysprep to generalize the VM; This ensures the VMs will have a unique SID when you clone it later to install primary and secondary Connection Servers.
  - Additional Considerations - First, clone a virtual machine from a VM template that does not have a Connection Server installed, run Sysprep on each cloned VM, and install the Connection Server on each virtual machine separately.
  - The Connection Server software cannot coexist on the same virtual or physical machine with any other VMware Horizon software components, including a replica server, Horizon Agent, or Horizon Client.

- Review the Getting Started page of VMware Horizon documentation. It provides a roadmap for implementing Horizon as a server and links to a list of helpful VMware education courses.

Note: Please consult the requirements listed in VMware Connection Server Prerequisites.

4.2 Installing Horizon Connection Server

To find the most recent version of Horizon Connection Server, please refer to VMware Horizon Documentation for release information.

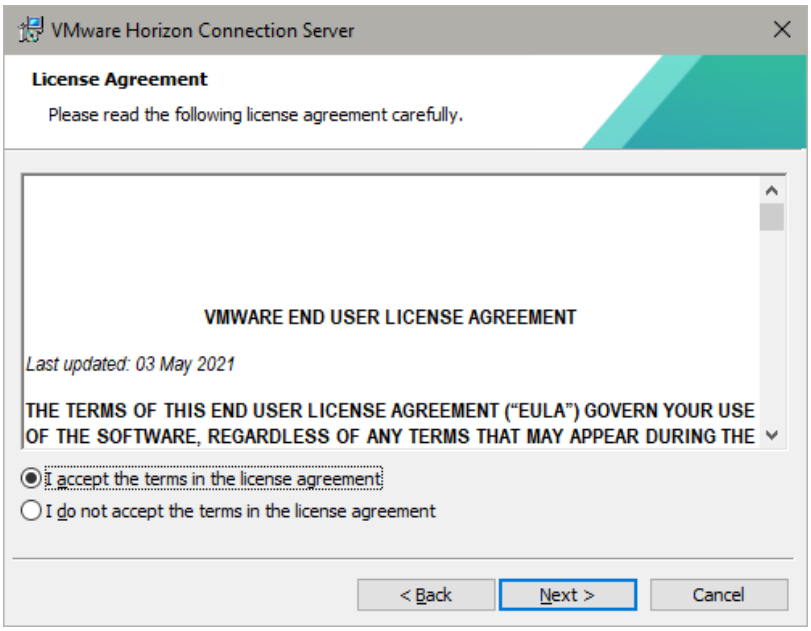
> **!** CAUTION: VMware Horizon 8 2111 is used to capture content for this guide. Text, images, and screen layout may differ from version to version.

1. Download the Connection Server file from the VMWare download site.
2. Navigate to the Connection Server installer and double-click the file to start the installation wizard. The installer filename is VMware-Horizon-Connection-Server-x86_64-*y.y.y-xxxxxx*.exe, where *xxxxxx* is the build number, and *y.y.y* is the version number.

   Ex. VMware-Horizon-Connection-Server-x86_64-8.4.0-18964782.exe

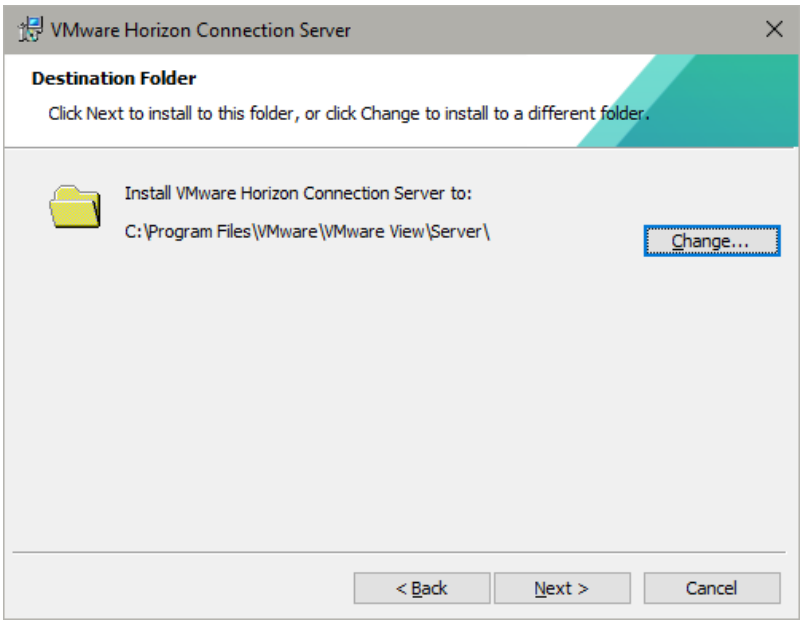
3. If prompted to allow the app to change your device, click **Yes**.
   The *Welcome page* of the wizard appears.
4. Click **Next** to display the *End User License Agreement* screen.

5. **Accept** the license agreement terms, then select **Next** to continue.

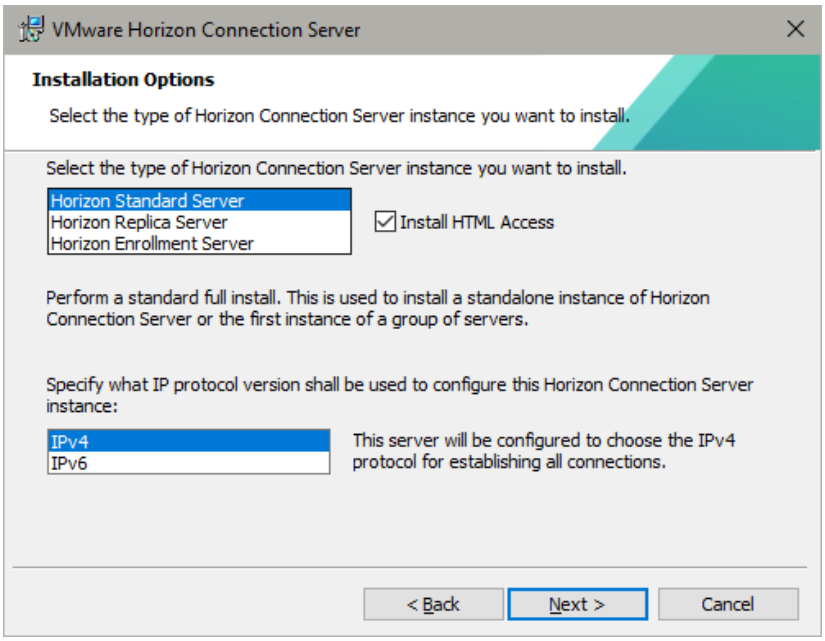


6. The Horizon installer displays the *Destination Folder* window.

   Choose an install location and select Next.

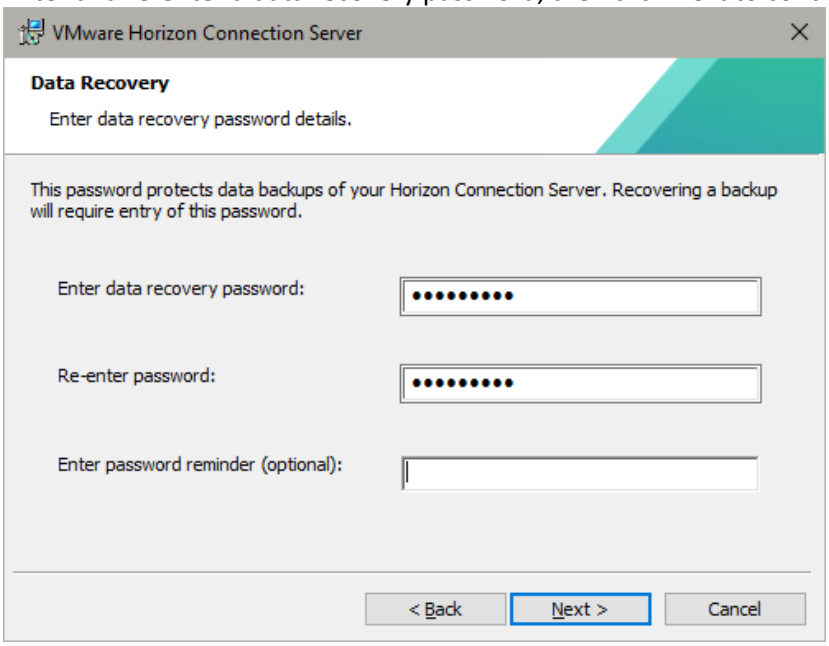


7. The Horizon installer displays the *Installation Options* screen. Configure the following:
   - Set the type of Horizon Connection Server instance to **Horizon Standard Server**.
   - Check the **Install HTML Access** checkbox
   - Select IP protocol version **IPv4.**
   - Click **Next** to continue.



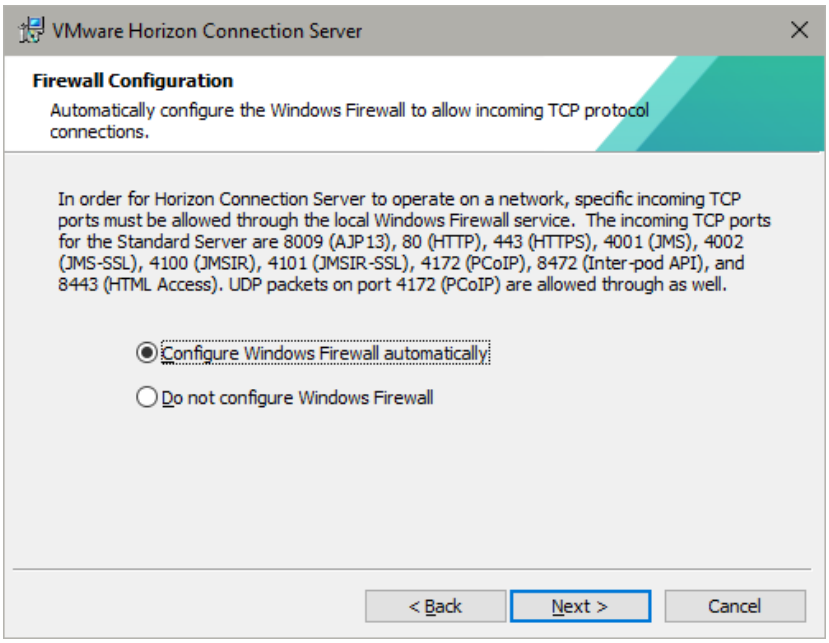
8. The Horizon installer displays the *Data Recovery* screen.

   Enter and re-enter a data recovery password, then click Next to continue.



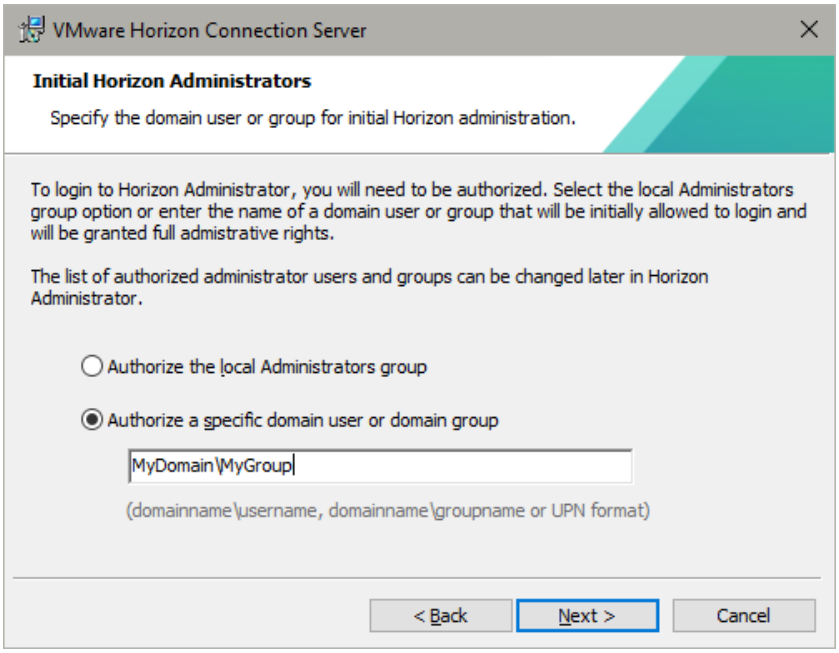
9. The Horizon installer displays the *Firewall Configuration* screen:

   Check the Configure Windows Firewall automatically checkbox. Click Next.



10. The Horizon installer displays the *Initial Horizon Administrators* screen.

    NVIDIA recommends authorizing an Active Directory domain group.

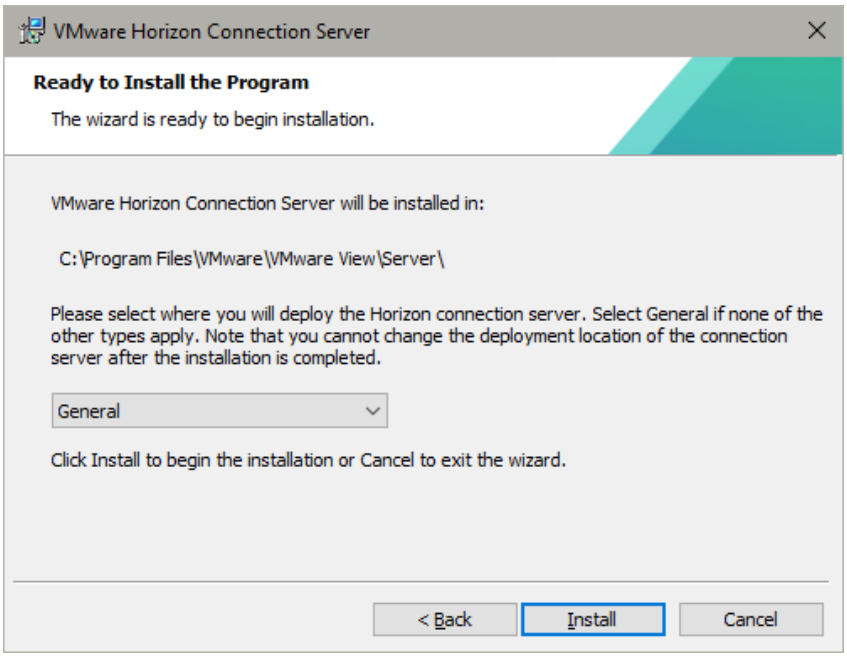


11. The Horizon install displays the *User Experience Improvement Program* screen:

    Check or clear the Join the VMware Customer Experience Improvement Program based on preference. Select the appropriate values in the remaining fields, then click Next to continue.

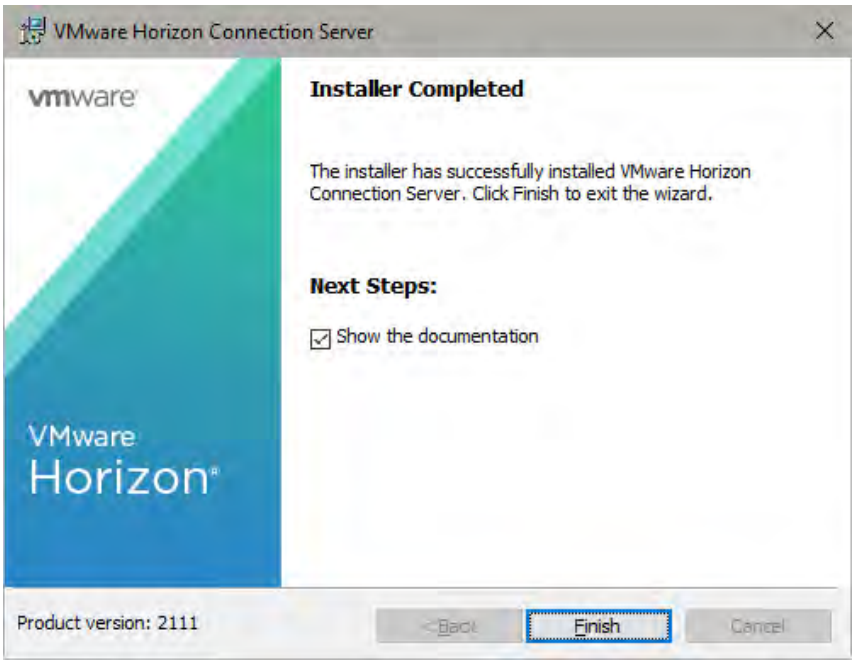
12. On the *Ready to Install* page, leave **General** as the default to indicate that you are deploying the Connection Server as an on-premises environment.

    Click **Install** to complete the installation.



13. The Horizon installer displays the *Installer Completed* screen.

    Click **Finish** to exit the installer.



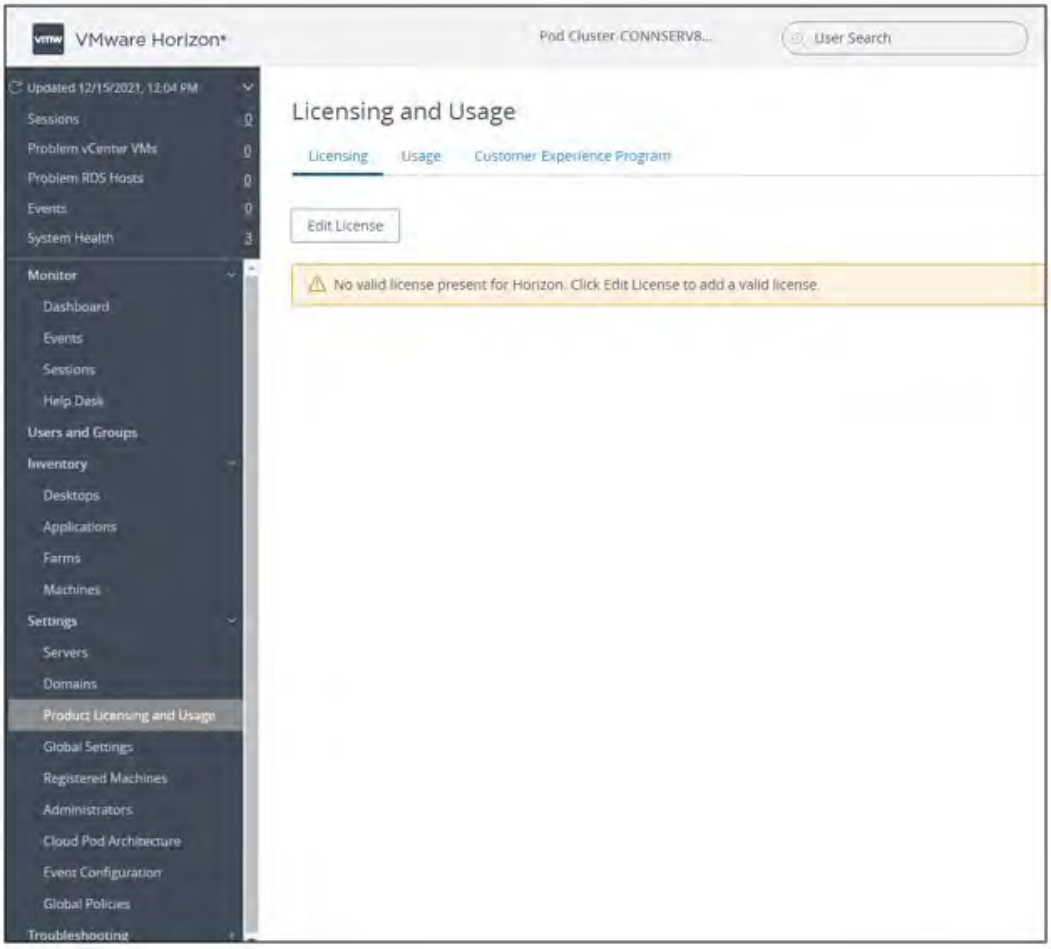
14. Open the browser page at *https://<host>/admin,* where **<host>** is the **FQDN hostname** of the server you installed Horizon.

4.2.1 Registering the Horizon License

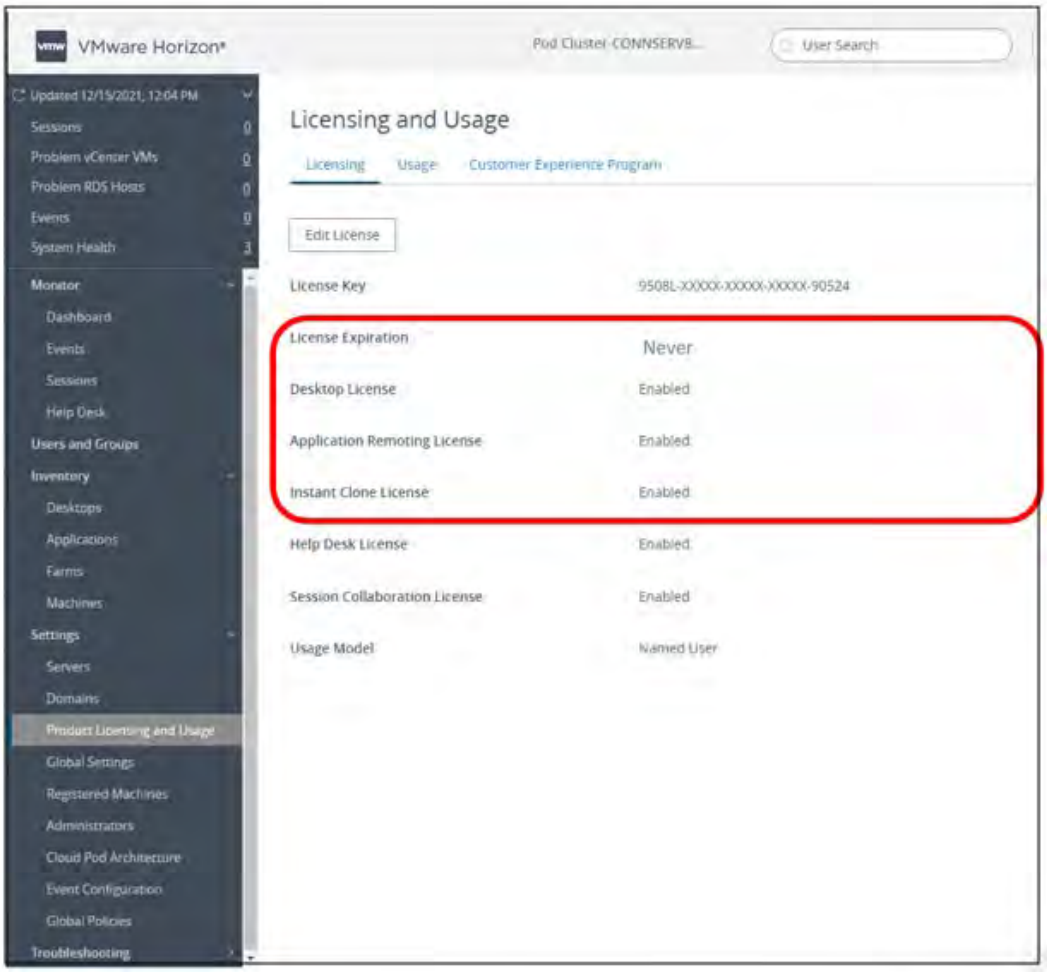
Generate a license key from VMWare Product Registration if you have not already done so.

1. Connect to the Horizon web console to open the Horizon Administrator:

2. Horizon defaults to the **Product Licensing and Usage** page. In the Right-hand pane, Horizon pre-selects the **Licensing and Usage** tab. Select the **Edit License** button – *Horizon Administrator* displays the *Edit License* window:



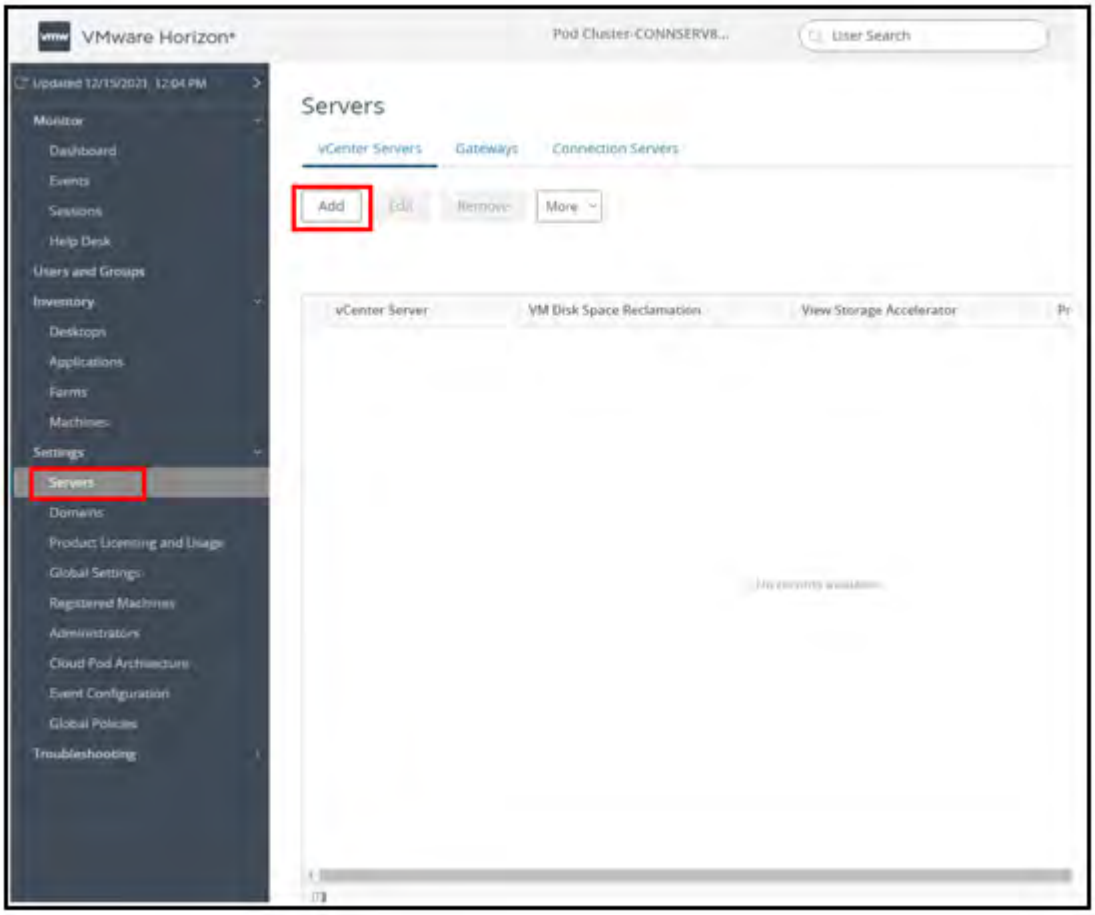
3. Enter your **license key** in the License serial number field. Click **OK** to decode and apply for the license.
4. Verify that the license is enabled by looking for the **Enabled** messages in the **License Expiration, Desktop License, Application Remoting License**, and **Instant Clone License** part of the *Licensing and Usage* window.



4.3 Registering vCenter Server

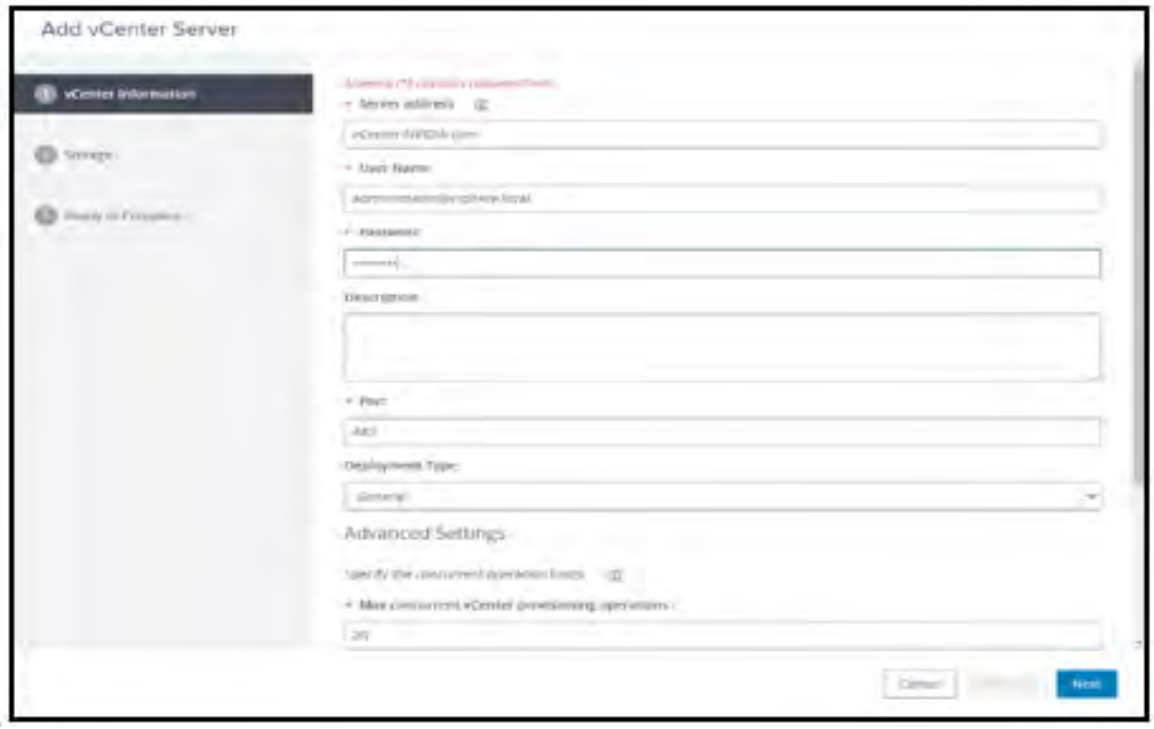
vCenter Server creates and manages the virtual machines used in Horizon desktop pools. The Connection Server uses a secure channel (TLS/SSL) to connect to the vCenter Server instance.

1. In the left-hand pane of the Horizon Administrator window, select **Settings > Server,** which takes you to the vCenter servers tab, then click the **Add** button.



2. On the **Add vCenter Server** page, complete the following text boxes before clicking **Next**:
   - **Server address** – Enter the fully qualified domain name (FQDN) of the vCenter Server instance.
   - **Username and Password** – Use the format name@domain.com for the name of the vCenter Server user account.

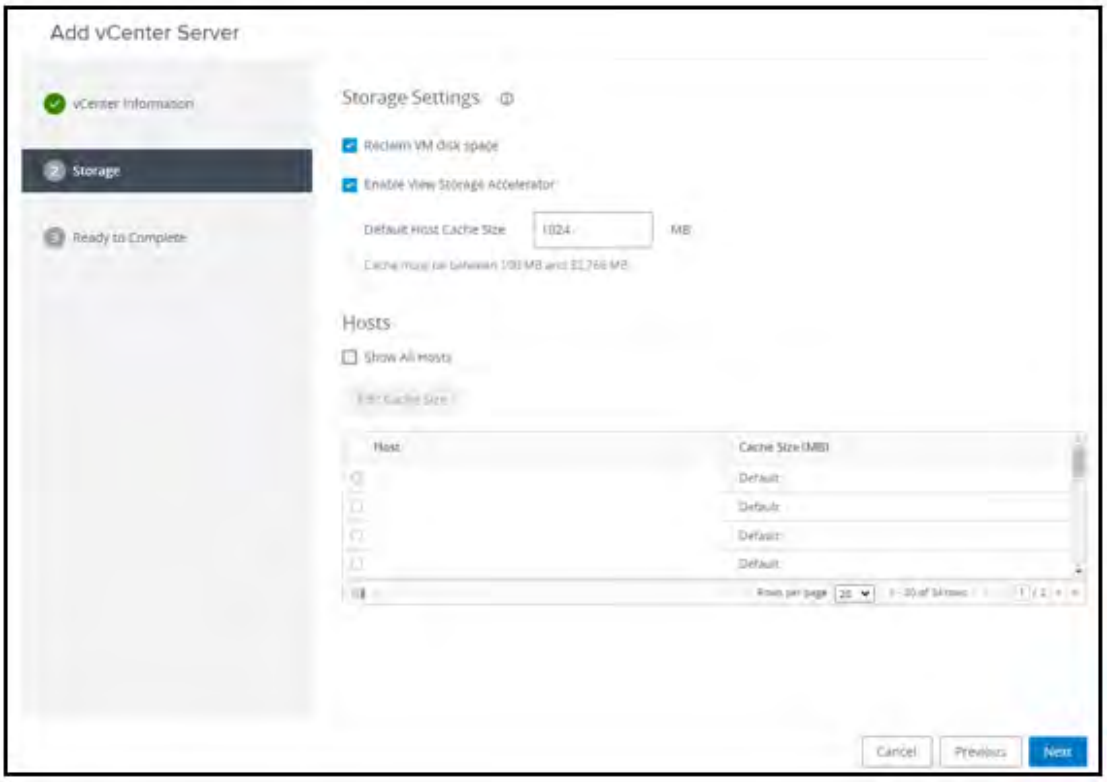
   You can leave the **default settings** for the other text boxes. **(Do Not Use View Composer)**



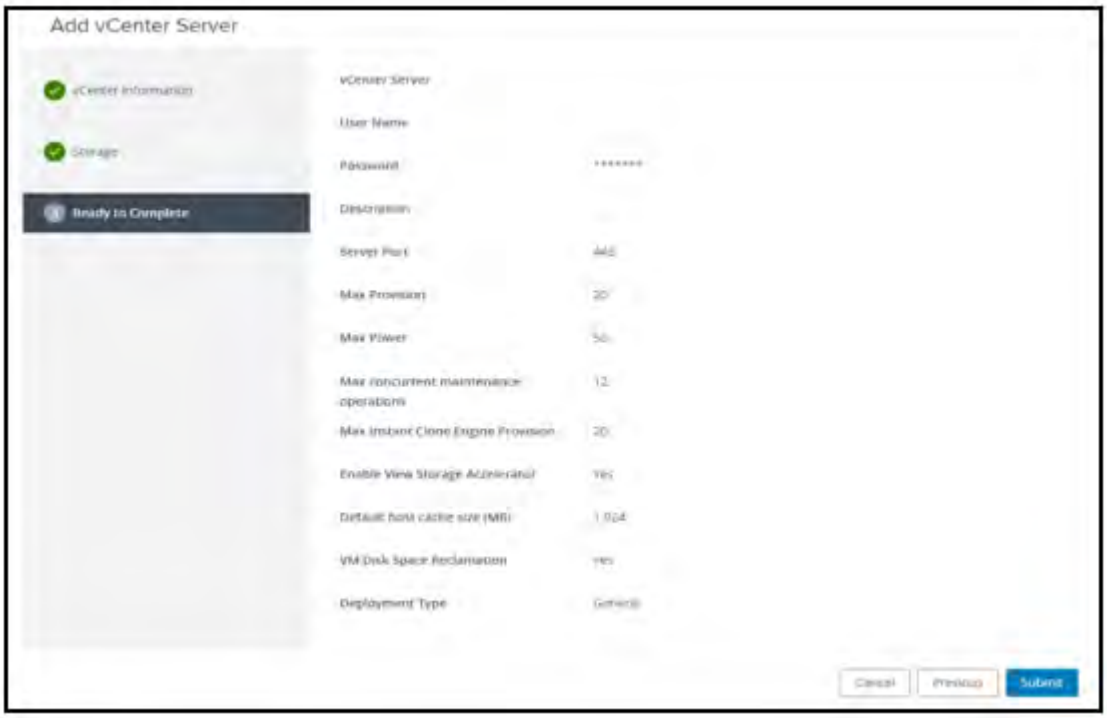
3. If an Invalid Certificate Detected prompt is displayed:

   - Click **View Certificate**.
   - In the **Certificate Information** window, review the thumbprint of the default self-signed certificate that was generated during installation, then click **Accept.**

4. Click **Next** on the rest of the wizard pages to accept the defaults.



5. On the **Ready to Complete** page, select **Submit.**



6. You are returned to the **Servers** > **vCenter Servers** tab, and the server you just added will appear in the list.

Chapter 5. Installing and configuring the NVIDIA vGPU Manager VIB

This chapter covers installing and configuring the NVIDIA vGPU Manager:

- Preparing the VIB file for Install
- Uploading VIB using WinSCP
- Installing vGPU Manager with the VIB
- Updating the VIB
- Verifying the Installation of the VIB
- Uninstalling the VIB
- Changing the Default Graphics Type in VMWare vSphere 6.5 and Later
- Changing the vGPU Scheduling Policy
- Disabling and Enabling ECC memory

5.1 Preparing the VIB file for Install

Before you begin, download the archive containing the VIB file from the NVIDIA Enterprise Application Hub login page and extract the archived contents to a folder. The file ends with .VIB is the file you must upload to the host datastore for installation.

Note: For demonstration purposes, these steps use WinSCP to upload the .VIB file to the ESXi host.WinSCP can be downloaded here.

5.1.1 Upload the .VIB file using WinSCP

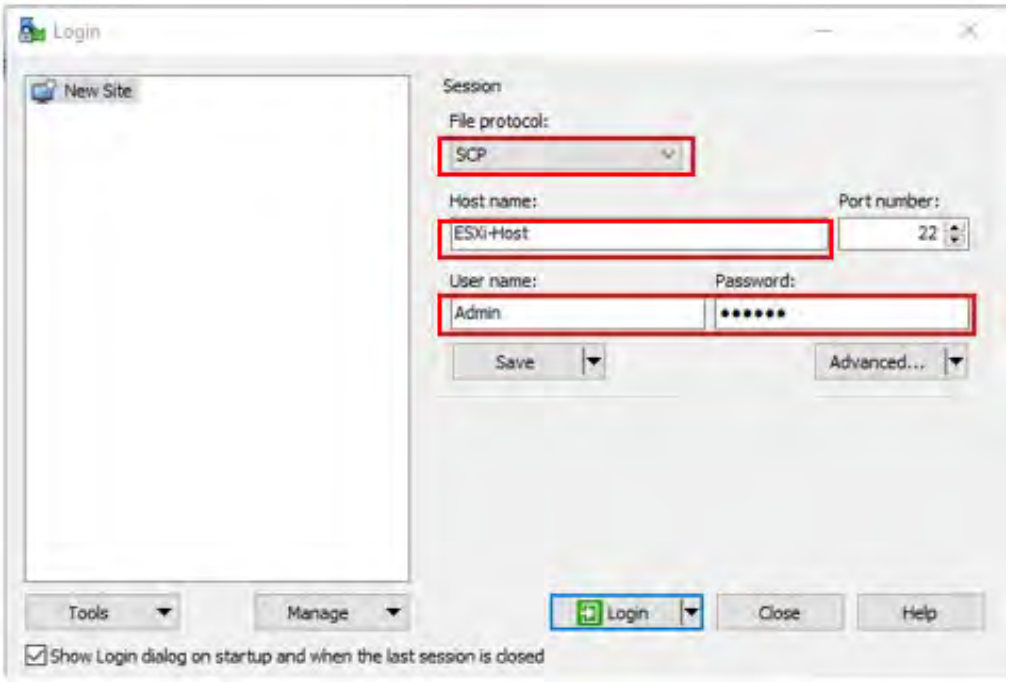
WinSCP is a Secure Copy (SCP) protocol based on SSH (Secure Shell) that enables file transfers between hosts on a network—uploading the**. VIB** file using WinSCP is the quickest way to transfer the file from the Source location to the destination location, the ESXi datastore. Refer to the WinSCP documentation page for detailed information.

1. To upload the **.VIB** file to the ESXi datastore using WinSCP. Start WinSCP.

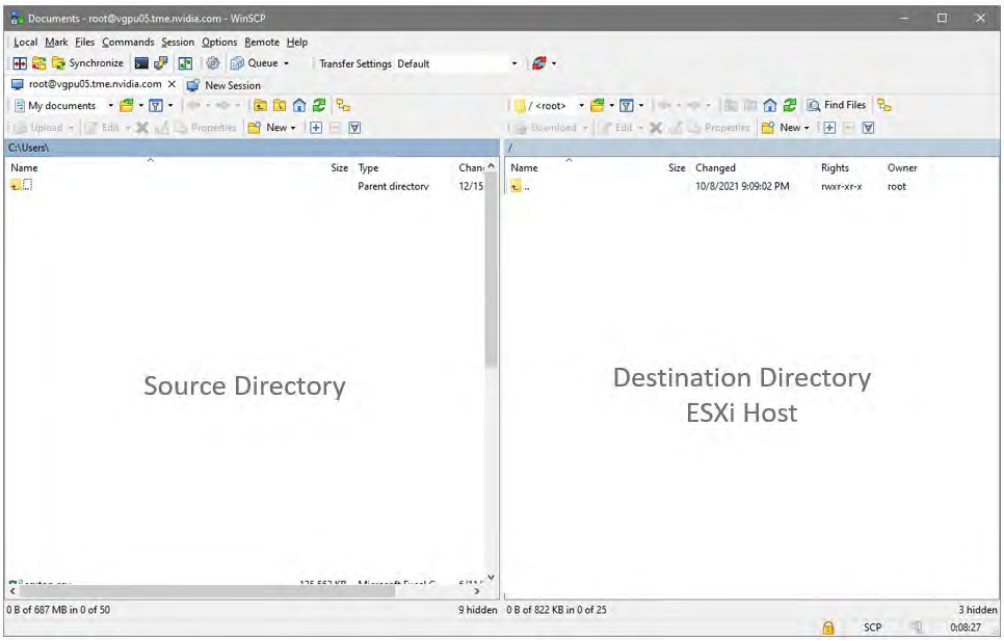
2. WinSCP opens to the **Login** window. In the Login window:
   - Select **SCP** as the file transfer protocol from the **File Protocol** dropdown menu.
   - Enter your ESXi hosts' name in the **Hostname** field.
   - Enter the hosts' username in the **User name** field.
   - Enter the hosts' password in the **Password** field.

   Note: Optional - You may want to save your session details to a site, so you do not need to type them in each time you want to connect—Press the Save button and type the site name.

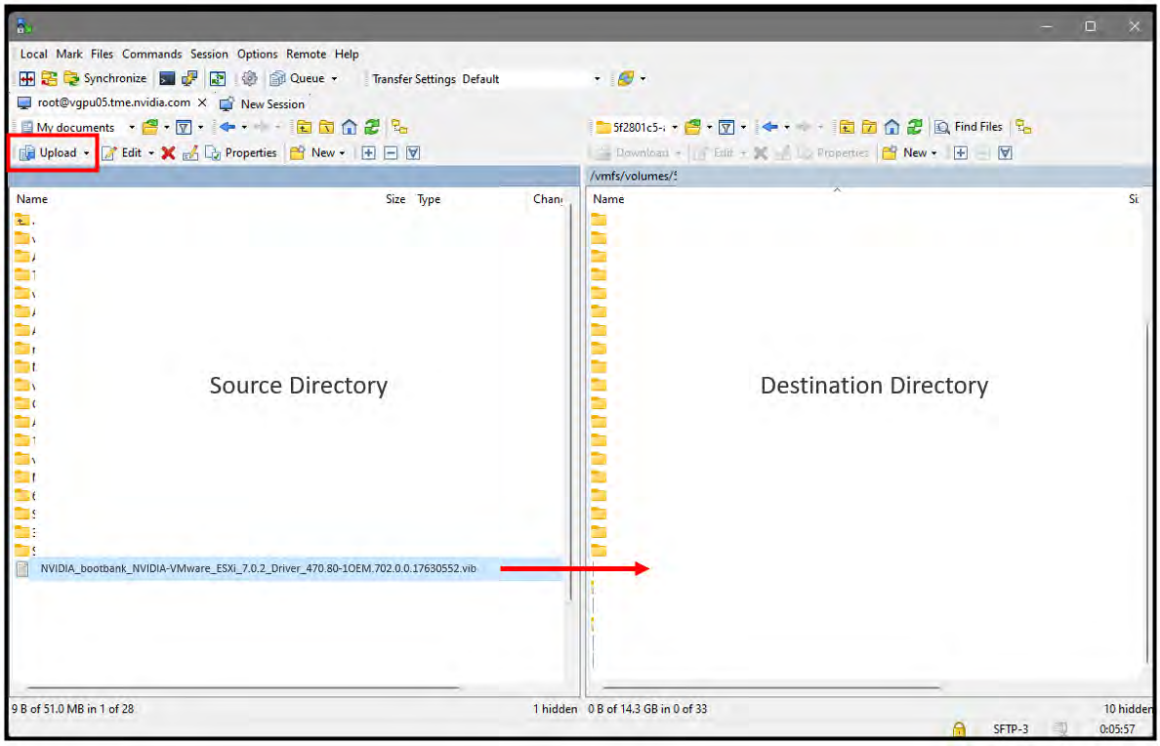
   - Select the **Login** button to connect to the ESXi host using the credentials you provided.



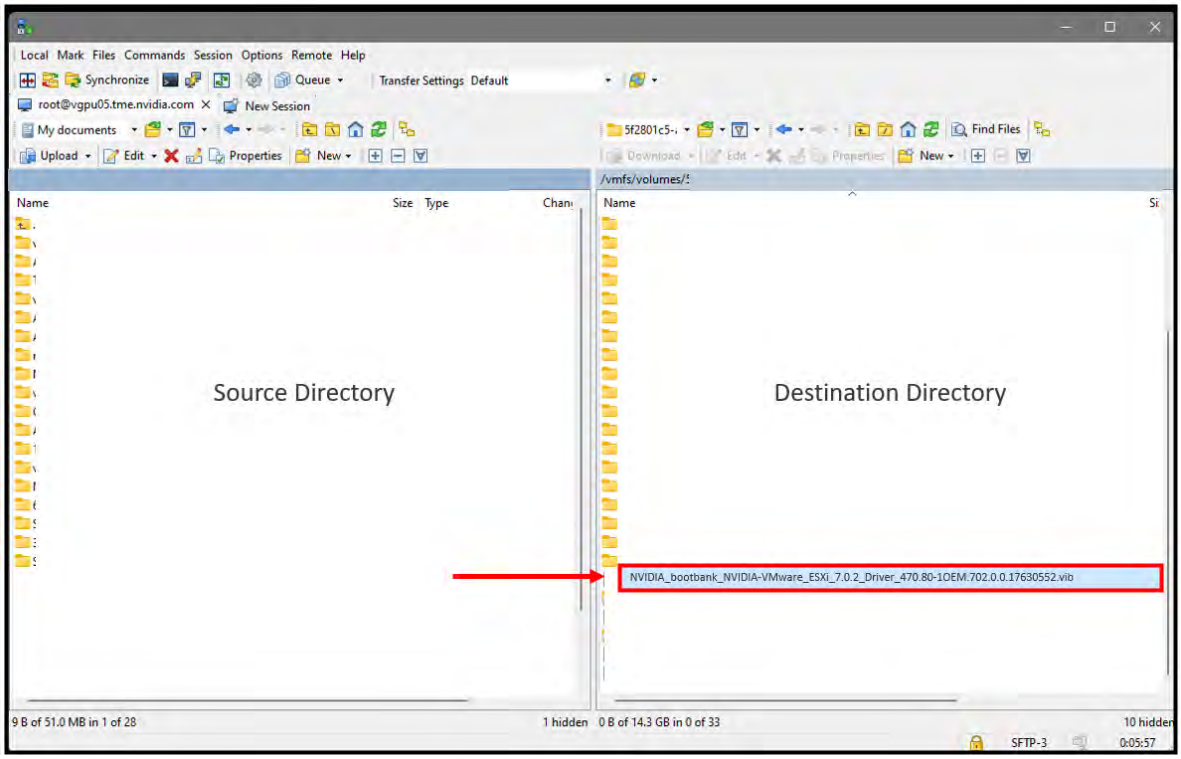
3. Select **Yes** to the Host **Certificate** warning.
4. Once you are connected to the ESXi host, you will see the contents of the default remote directory (typically, this is the ESXi hosts user's home directory) on the remote file panel.



5. Upload the .VIB file from the Source location to the Destination location (the DataStore on the ESXi host).
   - In the left panel (Source Directory), navigate to the location of the .VIB file. Select the **.VIB** file.
   - Navigate to the destination location within the DataStore on the ESXi host in the right panel.
   - Select the **Upload** button over the left panel to start the .VIB file download.



6. The **.VIB** file is uploaded to the datastore on the ESXi host.



5.2 Installing vGPU Manager with the VIB

The NVIDIA Virtual GPU Manager runs on the ESXi host. It is provided in the following formats:

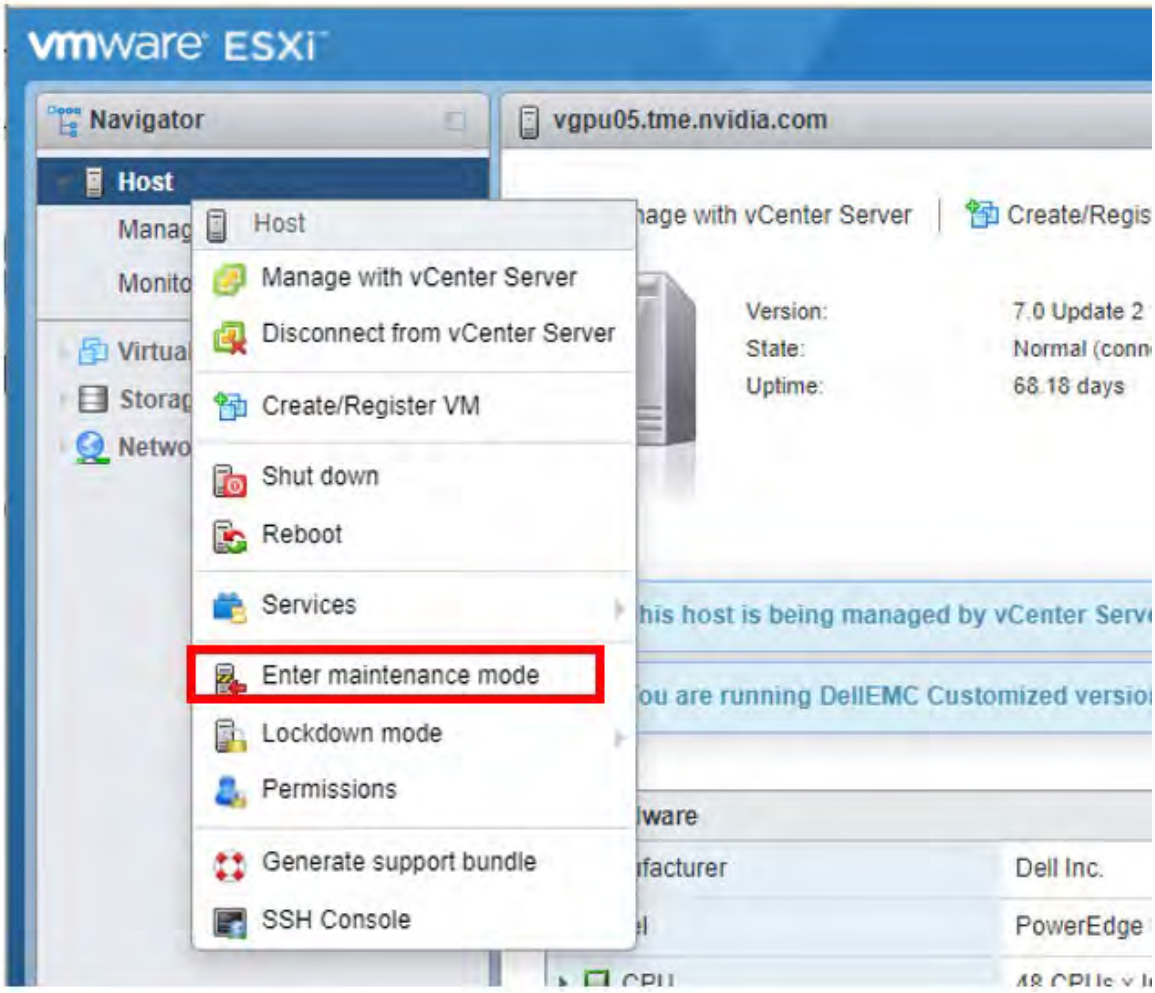
- As a VIB file, which must be copied to the ESXi host and then installed
- As an offline bundle that you can import manually as explained in Import Patches Manually in the VMware vSphere documentation.

> **!** CAUTION: Before vGPU release 11, NVIDIA Virtual GPU Manager and Guest VM drivers must be matched from the same main driver branch. If you update vGPU Manager to a release from another driver branch, guest VMs will boot with vGPU disabled until their guest vGPU driver is updated to match the vGPU Manager version. Consult Virtual GPU Software for VMware vSphere Release Notes for further details.

To install the vGPU Manager .VIB, you need to access the ESXi host via the ESXi Shell or SSH. Refer to VMware's documentation on how to enable ESXi Shell or SSH for an ESXi host

> Note: Before proceeding with the vGPU Manager installation, make sure that all VMs are powered off, and the ESXi host is in Maintenance Mode. Refer to VMware's documentation on how to place an ESXi host in maintenance mode.

1. Place the host into Maintenance mode by right-clicking it and then selecting **Maintenance Mode** - **Enter Maintenance Mode.**



Note: Alternatively, you can place the host into Maintenance mode using the command prompt by entering:

$ esxcli system maintenanceMode set --enable=true

This command will not return a response. Making this change using the command prompt will not refresh the vSphere Web Client UI. Click the Refresh icon in the upper right corner of the vSphere Web Client window.

CAUTION: Placing the host in Maintenance Mode disables any vCenter appliance running on this host until you exit Maintenance Mode, then restart that vCenter appliance.

2. Click **OK** to confirm your selection. This place's the ESXi host in Maintenance Mode.
3. Enter the `esxcli` command to install the vGPU Manager package:

```
[root@esxi:~] esxcli software vib install -v directory/NVIDIA_bootbank_NVIDIA-
VMware_ESXi_7.0.2_Driver_470.80-1OEM.702.0.0.17630552.vib

Installation Result
   Message: Operation finished successfully.
   Reboot Required: false
   VIBs Installed: NVIDIA_bootbank_NVIDIA-VMware_ESXi_7.0.2_Driver_470.80-
1OEM.702.0.0.17630552
   VIBs Removed:
   VIBs Skipped:
```

Note: The directory is the absolute path to the directory that contains the VIB file. You must specify the absolute path even if the VIB file is in the current working directory.

4. Reboot the ESXi host and remove it from Maintenance Mode.

Note: Although the display states "**Reboot Required: false**," a reboot is necessary for the vib to load and Xorg to start.

5. From the vSphere Web Client, exit Maintenance Mode by right-clicking the host and selecting **Exit Maintenance Mode.**

Note: Alternatively, you may exit from Maintenance mode via the command prompt by entering:

$ esxcli system maintenanceMode set --enable=false

This command will not return a response.

Making this change via the command prompt will not refresh the vSphere Web Client UI. Click the **Refresh** icon in the upper right corner of the vSphere Web Client window.

6. Reboot the host from the vSphere Web Client by right-clicking the host and selecting **Reboot**.

Note: You can reboot the host by entering the following at the command prompt:

$ reboot

This command will not return a response—the Reboot Host window displays.

7. When rebooting from the vSphere Web Client, enter a descriptive reason for the reboot in the **Log a reason for this reboot operation** field, and click **OK** to proceed.

5.3 Updating the VIB

Update the vGPU Manager VIB package if you want to install a new version of NVIDIA Virtual GPU Manager on a system where an existing version is already installed.

- To update the vGPU Manager VIB you need to access the ESXi host via the ESXi Shell or SSH. Refer to VMware's documentation on enabling ESXi Shell or SSH for an ESXi host.
- The driver version seen within this document is for demonstration purposes. There will be similarities, albeit minor differences, within your local environment.

Note: Before proceeding with the vGPU Manager update, make sure that all VMs are powered off, and the ESXi host is placed in maintenance mode. Refer to VMware's documentation on putting an ESXi host in maintenance mode.

1. Use the `esxcli` command to update the vGPU Manager package:

```
[root@esxi:~] esxcli software vib update -v directory/NVIDIA_bootbank_NVIDIA-
VMware_ESXi_7.0.2_Driver_470.80-1OEM.702.0.0.17630552.vib

Installation Result
Message: Operation finished successfully.
   Reboot Required: false
   VIBs Installed: NVIDIA_bootbank_NVIDIA-VMware_ESXi_7.0.2_Driver_470.80-
1OEM.702.0.0.17630552
   VIBs Removed: NVIDIA_bootbank_NVIDIA-
VMware_ESXi_7.0_Host_Driver_460.73.021OEM.700.0.0.15525992
   VIBs Skipped:
```

Note: *directory* is the path to the directory that contains the VIB file.

2. Reboot the ESXi host and remove it from maintenance mode.

5.4 Verifying the Installation of the VIB

After the ESXi host has rebooted, verify the installation of the NVIDIA vGPU software package.

1. Verify that the NVIDIA vGPU software package is installed and loaded correctly by checking for the NVIDIA kernel driver in the list of kernels-loaded modules.

```
[root@esxi:~] vmkload_mod -l | grep nvidia
nvidia                  5    8420
```

2. If the NVIDIA driver is not listed in the output, check dmesg for any load-time errors reported by the driver.
3. Verify that the NVIDIA kernel driver can successfully communicate with the NVIDIA physical GPUs in your system by running the nvidia-smi command.

- The nvidia-smi command is described in more detail in NVIDIA System Management Interface nvidia-smi.

Running the nvidia-smi command should produce a listing of the GPUs in your platform.

```
Tue Jan  4 20:48:42 2022
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 470.80    Driver Version: 470.80    CUDA Version: N/A            |
|-------------------------------+----------------------+----------------------+
| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
|                               |                      |               MIG M. |
|===============================+======================+======================|
|   0  NVIDIA A16       On      | 00000000:3F:00.0 Off |                    0 |
|  0%   33C    P8    15W /  62W |    896MiB / 15105MiB |      0%      Default |
|                               |                      |                  N/A |
+-------------------------------+----------------------+----------------------+
|   0  NVIDIA A16       On      | 00000000:41:00.0 Off |                    0 |
|  0%   34C    P8    15W /  62W |   3648MiB / 15105MiB |      0%      Default |
|                               |                      |                  N/A |
+-------------------------------+----------------------+----------------------+
|   0  NVIDIA A16       On      | 00000000:43:00.0 Off |                    0 |
|  0%   29C    P8    15W /  62W |      0MiB / 15105MiB |      0%      Default |
|                               |                      |                  N/A |
+-------------------------------+----------------------+----------------------+
|   0  NVIDIA A16       On      | 00000000:45:00.0 Off |                    0 |
|  0%   26C    P8    15W /  62W |      0MiB / 15105MiB |      0%      Default |
|                               |                      |                  N/A |
+-------------------------------+----------------------+----------------------+
|   0  NVIDIA A40       On      | 00000000:D8:00.0 Off |                    0 |
|  0%   30C    P8    30W / 300W |      0MiB / 45634MiB |      0%      Default |
|                               |                      |                  N/A |
+-------------------------------+----------------------+----------------------+

+-----------------------------------------------------------------------------+
| Processes:                                                                  |
|  GPU   GI   CI        PID   Type   Process name                  GPU Memory |
|        ID   ID                                                   Usage      |
|=============================================================================|
|  No running processes found                                                 |
+-----------------------------------------------------------------------------+
```

- If nvidia-smi fails to report the expected output for all the NVIDIA GPUs in your system, see *NVIDIA Virtual GPU Software User Guide* for troubleshooting steps.

The NVIDIA System Management Interface `nvidia-smi` also allows GPU monitoring using the following command:

```
$ nvidia-smi -l
```

This command switch adds a loop, automatically refreshing the display. The default refresh interval is 1 second

5.5 Uninstalling the VIB

To uninstall vGPU Manager:

1. Run `esxcli` to determine the name of the vGPU driver bundle.

```
$ esxcli software vib list | grep -i nvidia


NVIDIA-VMware_ESXi_7.0.2_Driver 470.80-1OEM.702.0.0.17630552
```

2. Run the following command to uninstall the driver package.

```
$ esxcli software vib remove -n NVIDIA-VMware_ESXi_7.0_Host_Driver --
maintenance-mode
```

3. The following message displays if the uninstall process is successful.

```
Removal Result
   Message: Operation finished successfully.
   Reboot Required: false
   VIBs Installed:
   VIBs Removed: NVIDIA-VMware_ESXi_7.0_Host_Driver-460.73.02-
1OEM.700.0.0.15525992
   VIBs Skipped:
```

4. Reboot the host to complete the uninstall of the vGPU Manager.

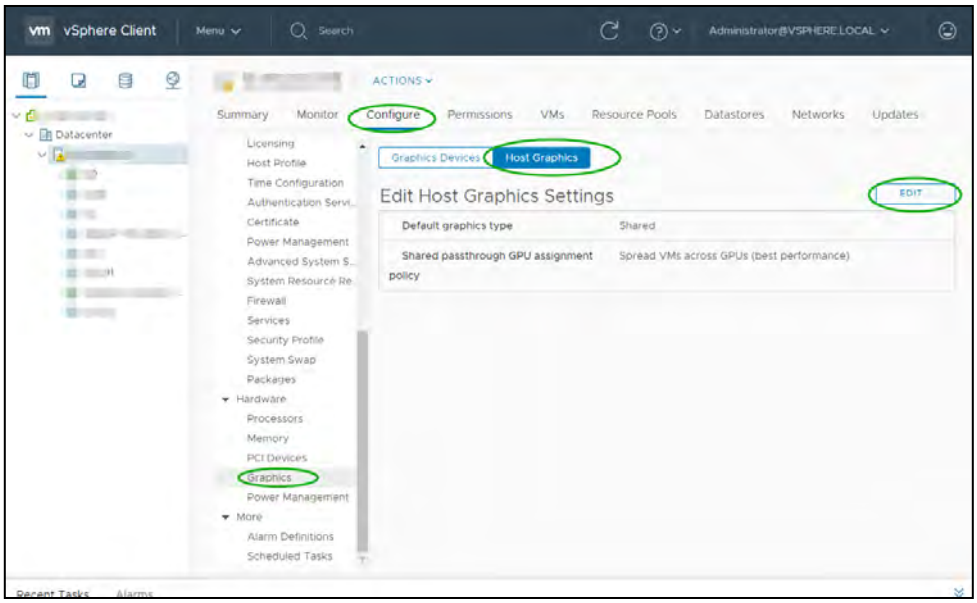
5.6 Changing the Default Graphics Type in VMware vSphere 6.5 and Later

The vGPU Manager VIBs for VMware vSphere 6.5 later provides vSGA and vGPU functionality in a single VIB. After the VIB is installed, the default graphics type is Shared, which provides vSGA functionality. To enable vGPU support for VMs in VMware vSphere 6.5, you must change the default graphics type to Shared Direct. If you do not modify the default graphics type, VMs to which a vGPU is assigned fail to start, and the following error message is displayed:

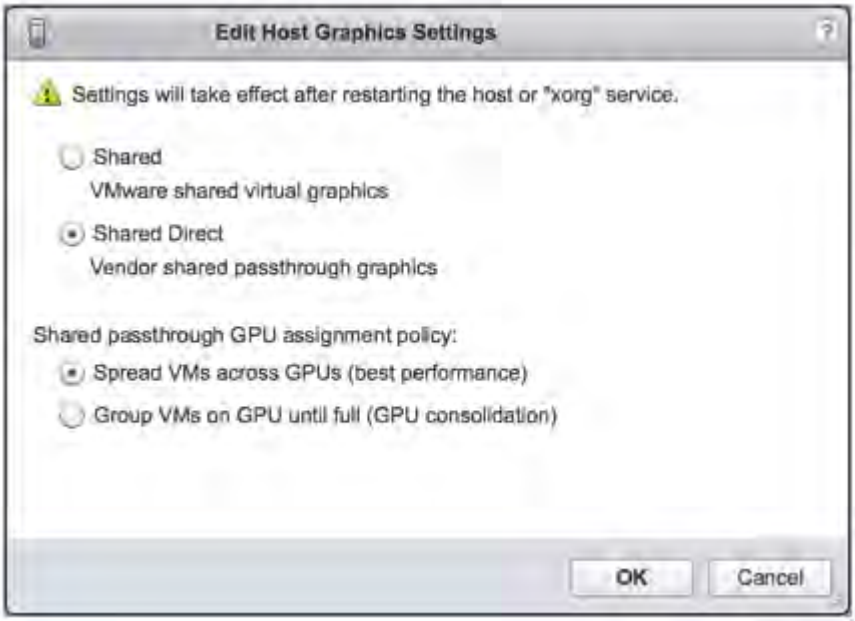
Note: If you are using a supported version of VMware vSphere earlier than 6.5, or are configuring a VM to use vSGA, omit this task.

Change the default graphics type before configuring vGPU. Output from the VM console in the VMware vSphere Web Client is unavailable for VMs running vGPU. Before changing the default graphics type, ensure that the ESXi host is running and that all VMs on the host is powered off.

1. Log in to vCenter Server by using the vSphere Web Client.
2. In the navigation tree, select your ESXi host and click the **Configure** tab.
3. From the menu, choose **Graphics** and then click the **Host Graphics** tab.
4. On the **Host Graphics** tab, click **Edit**.



5. In the **Edit Host Graphics Settings** window box that opens, select **Shared Direct** and click **OK**.



Note: This dialog box also lets you change the allocation scheme for vGPU-enabled VMs. For more information, see Modifying GPU Allocation Policy on VMware vSphere.

After you click **OK**, the default graphics type changes to Shared Direct.

6. Restart the ESXi host ***or*** stop and restart the Xorg service and nv-hostengine on the ESXi host.

   To stop and restart the Xorg service and nv-hostengine, perform these steps:

   a) Stop the Xorg service.
   ```
   [root@esxi:~] /etc/init.d/xorg stop
   ```
   b) Stop nv-hostengine.
   ```
   [root@esxi:~] nv-hostengine -t
   ```
   c) Wait for 1 second to allow nv-hostengine to stop.

   d) Start nv-hostengine.
   ```
   [root@esxi:~] nv-hostengine -d
   ```
   e) Start the Xorg service.
   ```
   [root@esxi:~] /etc/init.d/xorg start
   ```

After changing the default graphics type, configure vGPU as needed in Configuring a vSphere VM with Virtual GPU.

See also the following topics in VMware vSphere documentation:

- Log in to vCenter Server by Using the vSphere Web Client
- Configuring Host Graphics

5.7 Changing the vGPU Scheduling Policy

GPUs starting with the NVIDIA Maxwell™ graphic architecture implement a best-effort vGPU scheduler that aims to balance performance across vGPUs. The best effort scheduler allows a vGPU to use GPU processing cycles that are not being used by other vGPUs. Under some circumstances, a VM running a graphics-intensive application may adversely affect graphics-light applic performance in other VMs.

GPUs, starting with the NVIDIA Pascal™ architecture, also support equal share and fixed share vGPU schedulers. These schedulers limit GPU processing cycles used by a vGPU which prevents graphics-intensive applications running in one VM from affecting the performance of graphics-light applications running in other VMs. The best effort scheduler is the default scheduler for all supported GPU architectures.

5.7.1 vGPU Scheduling Policies

In this section, the three NVIDIA vGPU scheduling policies are defined. The vGPU scheduling policy is designed to fully use the GPU by balancing processes during GPU availability and unavailability. The overall intent of the three vGPU scheduling policies is to keep the GPU efficient, fast, and fair.

- **Best effort scheduling** provides consistent performance at a larger scale and reduces the TCO per user. The best effort scheduler leverages a round-robin scheduling algorithm, which shares GPU resources based on actual demand, optimally utilizing resources. This results in consistent performance with optimized user density. The best effort scheduling policy best uses the GPU during idle and not fully utilized times, allowing for optimized density and a good QoS.

- **Fixed share scheduling** always guarantees the same dedicated quality of service. The fixed share scheduling policies guarantee equal GPU performance across all vGPUs sharing the same physical GPU. Dedicated quality of service simplifies a POC. It also common benchmarks to measure physical workstation performance, such as SPECviewperf, to compare the performance with current physical or virtual workstations.

- **Equal share scheduling** provides equal GPU resources to each running VM. As vGPUs are added or removed, the share of GPU processing cycles allocated changes, accordingly, resulting in performance to increase when utilization is low and decrease when utilization is high.

Organizations typically leverage the best effort GPU scheduler policy for their deployment to achieve better utilization of the GPU, which usually results in supporting more users per server with a lower quality of service (QoS) and better TCO per user. Additional information regarding GPU scheduling can be found here.

5.7.2 RmPVMRL Registry Key

The `RmPVMRL` registry key sets the scheduling policy for NVIDIA vGPUs.

Note: You can change the vGPU scheduling policy only on GPUs based on the Pascal, Volta, Turing, and Ampere architectures.

Type

Dword

Contents

Value	Meaning
`0x00` (default)	Best effort scheduler
`0x01`	Equal share scheduler with the default time slice length
`0x00TT0001`	Equal share scheduler with a user-defined time slice length TT
`0x11`	Fixed share scheduler with the default time slice length
`0x00TT0011`	Fixed share scheduler with a user-defined time slice length TT

Examples

The default time slice length depends on the maximum number of vGPUs per physical GPU allowed for the vGPU type.

Maximum Number of vGPUs	Default Time Slice Length
Less than or equal to 8	2 ms
Greater than 8	1 ms

TT

- Two hexadecimal digits in the range 01 to 1E that set the length of the time slice in milliseconds (ms) for the equal share and fixed share schedulers. The minimum length is 1 ms, and the maximum length is 30 ms.
- If *TT* is 00, the length is set to the default length for the vGPU type.
- If *TT* is greater than 1E, the length is set to 30 ms.

Examples

This example sets the vGPU scheduler to equal share scheduler with the default time slice length.

```
RmPVMRL=0x01
```

This example sets the vGPU scheduler to equal share scheduler with a time slice that is 3 ms long.

```
RmPVMRL=0x00030001
```

This example sets the vGPU scheduler to a fixed share scheduler with the default time slice length.

```
RmPVMRL=0x11
```

This example sets the vGPU scheduler to a fixed share scheduler with a time slice that is 24 (0x18) ms long.

```
RmPVMRL=0x00180011
```

5.7.3 Changing the vGPU Scheduling Policy for All GPUs

Perform this task in your hypervisor command shell.

1. Open a command shell as the root user on your hypervisor host machine. You can use a secure shell (SSH) on all supported hypervisors for this purpose.
2. Set the `RmPVMRL` registry key to the value that sets the GPU scheduling policy that you want.
3. Use the `esxcli` command:

```
# esxcli system module parameters set -m nvidia -p
"NVreg_RegistryDwords=RmPVMRL=value"
```

Where <value> ***is*** the value that sets the vGPU scheduling policy you want, for example:

- **0x00 (default)** - Best Effort Scheduler
- **0x01 -** Equal Share Scheduler with the default time slice length
- **0x00030001** - Equal Share Scheduler with a time slice of 3 ms

- **0x11** - Fixed Share Scheduler with the default time slice length
- **0x00180011**- Fixed Share Scheduler with a time slice of 24 ms (0x18)

  - To review: the default time slice length depends on the maximum number of vGPUs per physical GPU allowed for the vGPU type:

Maximum Number of vGPUs	Default Time Slice Length
Less than or equal to 8	2 ms
Greater than 8	1 ms

For all supported values, see RmPVMRL Registry Key.

4. Reboot your hypervisor host machine.

Note: Confirm that the scheduling behavior was changed as explained in Getting the Current Time-Sliced vGPU Scheduling Behavior for All GPUs.

5.7.4 Changing the vGPU Scheduling Policy for Select GPUs

Perform this task in your hypervisor command shell:

1. Open a command shell as the root user on your hypervisor host machine. On all supported hypervisors, you can use the secure shell (SSH) for this purpose.

2. Use the lspci command to obtain the PCI domain and bus/device/function (BDF) of each GPU for which you want to change the scheduling behavior.

3. Pipe the output of lspci to the grep command to display information only for NVIDIA GPUs.

```
# lspci | grep NVIDIA
```

The NVIDIA GPU listed in this example has the PCI domain `0000` and BDF `3f:00.0`.

```
0000:3f:00.0 3D controller: NVIDIA Corporation NVIDIANVIDIA A16 [vmgfx0]
```

4. On VMware vSphere, use the `esxcli` set command.

```
# esxcli system module parameters set -m nvidia \
-p "NVreg_RegistryDwordsPerDevice=pci=pci-domain:pci-bdf;RmPVMRL=value\
[;pci=pci-domain:pci-bdf;RmPVMRL=value...]"
```

For each GPU, provide the following information:

- **pci-domain**
  - The PCI domain of the GPU.
- **pci-bdf**

- The PCI device BDF of the GPU.

- **value**
  - The value that sets the GPU scheduling policy and the length of the tie slice you want. For example:
    - **0x01** - Sets the vGPU scheduling policy to Equal Share Scheduler with the default time slice length.
    - **0x00030001** - Sets the vGPU scheduling policy to Equal Share Scheduler with a time slice that is 3ms long.
    - **0x11** - Sets the vGPU scheduling policy to Fixed Share Scheduler with the default time slice length.
    - **0x00180011** - Sets the vGPU scheduling policy to Fixed Share Scheduler with a time slice that is 24 ms (0x18) long.
      - For all supported values, see RmPVMRL Registry Key.

  This example adds an entry to the /etc/modprobe.d/nvidia.conf file to change the scheduling behavior of two GPUs as follows:

- For the GPU at PCI domain 0000 and BDF 85:00.0, the vGPU scheduling policy is set to Equal Share Scheduler.
- For the GPU at PCI domain 0000 and BDF 86:00.0, the vGPU scheduling policy is set to Fixed Share Scheduler.

```
options nvidia NVreg_RegistryDwordsPerDevice=
"pci=0000:85:00.0;RmPVMRL=0x01;pci=0000:86:00.0;RmPVMRL=0x11"
```

5. Reboot your hypervisor host machine.

Note: Confirm that the scheduling behavior was changed as explained in Getting the Current Time-Sliced vGPU Scheduling Behavior for All GPUs.

5.7.5 Restoring Default vGPU Scheduling Policies

Perform this task in your hypervisor command shell.

1. Open a command shell as the root user on your hypervisor host machine. You can use a secure shell (SSH) on all supported hypervisors for this purpose.
2. Unset the `RmPVMRL` registry key.
3. Set the module parameter to an empty string.

```
# esxcli system module parameters set -m nvidia -p "module-parameter="
```

module-parameter

The module parameter to set, which depends on whether the scheduling behavior was changed for all GPUs or select GPUs:

- For all GPUs, set the `NVreg_RegistryDwords` module parameter.
- For select GPUs, set the `NVreg_RegistryDwordsPerDevice` module parameter.

For example, to restore default vGPU scheduler settings after they were changed for all GPUs, enter this command:

```
# esxcli system module parameters set -m nvidia -p "NVreg_RegistryDwords="
```

4. Reboot your hypervisor host machine.

5.8 Disabling and Enabling ECC Memory

Some NVIDIA GPUs support error-correcting code (ECC) memory with NVIDIA vGPU software. ECC memory improves data integrity by detecting and handling double-bit errors. However, not all GPUs, vGPU types, and hypervisor software versions support ECC memory with NVIDIA vGPU. Refer to the NVIDIA Virtual GPU Software Documentation for detailed information about ECC Memory.

Note: Enabling ECC memory has a 1/15 overhead cost because it uses the GPU VRAM to store the ECC bits —resulting in a less usable frame buffer for the vGPU.

On GPUs that support ECC memory with NVIDIA vGPU, ECC memory is supported with C-series and Q-series vGPUs, but not with A-series and B-series vGPUs. Although A-series and B-series vGPUs start on physical GPUs on which ECC memory is enabled, enabling ECC with vGPUs that do not support it might incur some costs.

On physical GPUs that do not have HBM2 memory, the amount of frame buffer usable by vGPUs is reduced. All types of vGPU are affected, not just vGPUs that support ECC memory.

The effects of enabling ECC memory on a physical GPU are as follows:

- ECC memory is exposed as a feature on all supported vGPUs on the physical GPU.
- In VMs that support ECC memory, ECC memory is enabled, with the option to disable ECC in the VM.
- ECC memory can be enabled or disabled for individual VMs. Enabling or disabling ECC memory in a VM does not affect the amount of frame buffer usable by the vGPUs.

GPUs based on the Pascal GPU architecture and later GPU architectures support ECC memory with NVIDIA vGPU. These GPUs are supplied with ECC memory enabled.

Some hypervisor software versions do not support ECC memory with NVIDIA vGPU.

If you use a hypervisor software version or GPU that does not support ECC memory with NVIDIA vGPU and ECC memory is enabled, NVIDIA vGPU fails to start. In this situation, you must ensure that ECC memory is disabled on all GPUs using NVIDIA vGPU.

5.8.1 Disabling ECC Memory

If ECC memory is unsuitable for your workloads but is enabled on your GPUs, disable it. You must also ensure that ECC memory is disabled on all GPUs if you use NVIDIA vGPU with a hypervisor software

version or a GPU that does not support ECC memory with NVIDIA vGPU. If your hypervisor software version or GPU does not support ECC memory and ECC memory is enabled, NVIDIA vGPU fails to start.

Where to perform this task depends on whether you are changing ECC memory settings for a physical GPU or a vGPU.

- For a physical GPU, perform this task from the hypervisor host.
- For a vGPU, perform this task from the VM to which the vGPU is assigned.

Note: ECC memory must be enabled on the physical GPU where the vGPUs reside.

Before you begin, ensure that NVIDIA Virtual GPU Manager is installed on your hypervisor. If you are changing ECC memory settings for a vGPU, ensure that the NVIDIA vGPU software graphics driver is installed in the VM to which the vGPU is assigned.

1. Use nvidia-smi to list the status of all physical GPUs or vGPUs, and check for ECC noted as enabled.

```
# nvidia-smi -q

==============NVSMI LOG==============

Timestamp                                 : Fri Dec  3 20:33:42 2021
Driver Version                            : 470.80

Attached GPUs                             : 5
GPU 00000000:3F:00.0

[...]

   Ecc Mode
        Current                           : Enabled
        Pending                           : Enabled

[...]
```

2. Change the ECC status to off for each GPU for which ECC is enabled.
   a) If you want to change the ECC status to off for all GPUs on your host machine or vGPUs assigned to the VM, run this command:

```
# nvidia-smi -e 0
```

   b) If you want to change the ECC status to off for a specific GPU or vGPU, run this command:

```
# nvidia-smi -i id -e 0
```

id is the index of the GPU or vGPU as reported by nvidia-smi.

This example disables ECC for the GPU with index 0000:02:00.0.

```
# nvidia-smi -i 0000:02:00.0 -e 0
```

3. Reboot the host or restart the VM.
4. Confirm that ECC is now disabled for the GPU or vGPU.

```
# nvidia-smi -q

==============NVSMI LOG==============

Timestamp                                 : Fri Dec  3 20:38:42 2021
Driver Version                            : 470.80

Attached GPUs                             : 5
GPU 00000000:3F:00.0


[...]

   Ecc Mode
        Current                           : Disabled
        Pending                           : Disabled

[...]
```

5.8.2 Enabling ECC Memory

If ECC memory is suitable for your workloads and is supported by your hypervisor software and GPUs, but is disabled on your GPUs or vGPUs, enable it.

Where this task is performed depends on whether you are changing ECC memory settings for a physical GPU or a vGPU.

- For a physical GPU, perform this task from the hypervisor host.
- For a vGPU, perform this task from the VM to which the vGPU is assigned.

Note: ECC memory must be enabled on the physical GPU where the vGPUs reside.

Before you begin, ensure that NVIDIA Virtual GPU Manager is installed on your hypervisor. If you are changing ECC memory settings for a vGPU, ensure that the NVIDIA vGPU software graphics driver is installed in the VM to which the vGPU is assigned.

1. Use nvidia-smi to list the status of all physical GPUs or vGPUs and check for ECC noted as disabled.

```
# nvidia-smi -q

==============NVSMI LOG==============

Timestamp                                 : Fri Dec  3 20:45:42 2021
Driver Version                            : 470.80
```

```
CUDA Version                                : Not Found

Attached GPUs                               : 5
GPU 00000000:3F:00.0

[...]

   Ecc Mode
        Current                             : Disabled
        Pending                             : Disabled

[...]
```

2. Change the ECC status to on for each GPU or vGPU for which ECC is enabled.

   - If you want to change the ECC status to on for all GPUs on your host machine or vGPUs assigned to the VM, run this command:

```
# nvidia-smi -e 1
```

   - If you want to change the ECC status to on for a specific GPU or vGPU, run this command:

```
# nvidia-smi -i id -e 1
```

   id is the index of the GPU or vGPU as reported by nvidia-smi.

   This example enables ECC for the GPU with index 0000:02:00.0.

```
# nvidia-smi -i 0000:02:00.0 -e 1
```

3. Reboot the host or restart the VM.

4. Confirm that ECC is now enabled for the GPU or vGPU

```
# nvidia-smi -q

==============NVSMI LOG==============

Timestamp                                   : Fri Dec  3 20:50:42 2021
Driver Version                              : 470.80

Attached GPUs                               : 5
GPU 00000000:3F:00.0

[...]

   Ecc Mode
        Current                             : Enabled
        Pending                             : Enabled

[...]
```

Chapter 6. NVIDIA License Server System

This chapter covers The NVIDIA Licensing System, including:

- NVIDIA License Server Documentation.
- License Purchasing
- Server Instance types

6.1 NVIDIA License Server Documentation

The NVIDIA License System serves licenses to NVIDIA software products. To activate licensed functionalities, a licensed client leases a software license served over the network from an NVIDIA License System service instance. The NVIDIA License System Documentation explains in full detail how to install, configure, and manage licenses for virtual GPU software.

NVIDI RTX Virtual Workstation (vWS), NVIDIA Virtual PC (vPC), and NVIDIA Virtual Applications (vApps) are available as licensed products on NVIDIA GPUs. To enable the full features of the vGPU, configure the licensing for these products. NVIDIA vGPUs that require licensing run at a reduced capability until a license is acquired.

6.2 License Purchasing

NVIDIA vGPU software products can be purchased as either a perpetual license with a Support Updates and Maintenance Subscription (SUMS) or an annual subscription. The perpetual license gives the user the right to use the software indefinitely, with no expiration. All NVIDIA vGPU software products with perpetual licenses must be purchased in conjunction with a four-year or five-year SUMS. A one-year SUMS is available only for renewals. For more information, refer to the Virtual GPU Packaging and Licensing Guide.

6.3 Server Instance Types

The role of the service instance is to distribute licenses to client systems from the license server to which it is linked. Summarizing the process; A client system passes its client configuration token to the server instance. Information in the client configuration token enables the service instance to identify the licensed client, the License Server, and the entitlements (or software functionality) to distribute with the requested license.

Note: A Service Instance is required to serve licenses to licensed clients. You must choose a service instance type for your environment before proceeding. Refer to the NVIDIA License System Documentation for more information.

The NVIDIA License System has two types of Service Instances:

1. **Cloud License Service (CLS) instance** - **(Recommended)** – NVIDIA CLS Instance is a cloud-based service hosted on the NVIDIA Licensing Portal and managed by NVIDIA and the cloud service provider.
   - Quick setup: The process for configuring a CLS instance is designed for ease of use and simplicity.
   - You do not need dedicated resources; NVIDIA manages the infrastructure entirely.
   - The CLS Instance provides robustness and dynamic scalability, and maintenance updates are generally transparent.
2. **Delegated License Service (DLS) instance -** NVIDIA DLS instance is hosted on-premises at a location accessible from your private network, such as inside your data center. The DLS instance is installed as a virtual appliance.
   - The DLS virtual appliance is on-premises and is entirely disconnected from the NVIDIA Licensing Portal.
   - You can choose which data Centers to deploy your DLS instances and how many DLS instances you need.
   - Review the DLS Virtual Appliance Platform Requirements documentation for configuration and installation information.

Note: Refer here for NVIDIA Virtual GPU Software License Server Documentation.

Chapter 7. Selecting the Correct vGPU Profiles

This chapter covers selecting the correct vGPU Profiles:

- Creative and Technical Professionals
- Knowledge Worker profiles
- Frame buffer Utilization and vGPU Profile selection

Selecting the right vGPU profile based on the users' tasks and workloads maximizes the whole virtualization experience. Virtual GPU profiles determine the amount of frame buffer allocated to your virtual machine and which license is used for the virtual instances. This section provides vGPU profile guidance and additional references for these NVIDIA licensed software products - NVIDIA RTX Virtual Workstation (RTX vWS), NVIDIA Virtual PC (vPC), and NVIDIA Virtual Applications (vApps).

7.1 Creative and Technical Professionals

Previously, creative and technical professionals were limited to physical devices like laptops or desktop workstations due to their demanding tasks and heavy workloads. But now, the NVIDIA Virtual GPU (vGPU) paired with the NVIDIA RTX Virtual Workstation (RTX vWS) software solution enables creative and technical professionals the ability to access their most demanding applications from anywhere with performance that rivals physical workstations. NVIDIA RTX vWS software accelerates professional design and visualization applications, including Autodesk Revit, Maya, Dassault Systèmes CATIA, Solidworks, Esri ArcGIS Pro, Petrel, and more.

Considerations for selecting the right vGPU profile for creative and technical professionals are compatibility and performance. The Q-series vGPU profiles undergo the same rigorous application certification process as the NVIDIA RTX™ Enterprise platform drivers for professional graphics applications. RTX vWS software supports RTX Enterprise drivers, allowing users to benefit from the acceleration and stability that RTX brings to professional applications used by the most demanding customers today. As a result, you can expect 100% compatibility and performance within your applications using the NVIDIA RTX vWS software licensed product.

Begin your profile selection by considering the requirements of your users' primary applications. Professional application software vendors certify their products to run with NVIDIA vGPU software, ensuring that the performance is tuned for maximum efficiency. Most vendors' websites have a dedicated page indicating the proper GPU hardware; use those recommendations to select the

appropriate vGPU profile to meet your end users' needs. To understand more about the graphics requirements of your users' applications, consult the application vendors.

7.1.1 Matching Profiles to User Needs

As stated earlier in the Size Your Environment section, you must define your users' needs and match them to the NVIDIA vGPU profiles that provide the right amount of frame buffer and the correct software licenses. By categorizing a user's workload as Light, Medium, and Heavy, we can illustrate a base VM configuration for an NVIDIA RTX vWS deployment.

- Light user
  - 8 GB RAM
  - Four vCPUs (2.4 GHz)
  - A40-8Q vGPU Profile
- Medium
  - 16 GB RAM
  - Eight vCPUs (2.6 GHz)
  - A40-12Q
- Heavy user
  - 32 GB RAM
  - 12 vCPUs (3.2 GHz)
  - A40-24Q

Note: These VM configurations are benchmark testing for the RTX vWS “Dedicated Performance” user type quality of service (QoS). For more RTX vWS Benchmarking results and vGPU sizing guidance, refer to the NVIDIA RTX vWS Workstation Sizing Guide.

7.2 Knowledge Worker Profiles

Today's digital workplace has become a hybrid workforce that is more now than ever dependent on desktop virtualization with increased graphics requirements for productivity applications. The workload of the digital worker (aka Knowledge Worker) includes a graphics-rich experience with immersive visual quality and processing speed. Selecting the right NVIDIA virtual GPU (vGPU) profile can meet the increasing demands of the Knowledge worker in a virtual desktop (VDI) environment. NVIDIA vGPU profiles determine the amount of frame buffer allocated to your virtual machine. The vGPU profiles supported on NVIDIA GPUs with NVIDIA software are the 1B (with 1024 MB of frame buffer) and 2B (with 2048 MB of frame buffer).

The NVIDIA vPC licensed software is selected for knowledge worker workloads within a virtual desktop infrastructure (VDI) environment. Workloads configured with the NVIDIA vPC software accelerates office productivity applications, streaming video, Windows, RDSH, multiple and high-resolution monitors, and 2D electric design automation (EDA).

7.2.1 Matching Profiles to User Needs

As stated earlier in the section Size Your Environment, you must define your users' needs and match them to the NVIDIA vGPU profiles that provide the right amount of resources. Here is a recommended knowledge worker's virtual machine configuration.

- vPC-1B vGPU profile for a Horizon VDI Desktop
  - 6 GB RAM
  - Four vCPUs @ 2.4 GHz or faster (single-socket)
  - Dual HD Monitor - (1920x1080)
- vPC-2B vGPU profile for a Horizon VDI Desktop
  - 6 GB RAM
  - Four vCPUs @ 2.4 GHz or faster (single-socket)
  - Dual Quad Monitor (2560x1440) or Single 4K Monitor (4096x2160)

Note: For more information on Knowledge Worker sizing guidance, refer to the NVIDIA vPC Sizing Guide.

7.3 Frame Buffer Utilization and vGPU Profile selection

As stated earlier, NVIDIA vGPU profiles determine the amount of frame buffer allocated to your virtual machine, thereby rendering frame buffer utilization a critical metric when considering vGPU Profiles for your VDI deployment. Frame buffer utilization within the VM can be affected by the application load, monitor configurations, and screen resolution.

Multiple monitors and higher screen resolutions scenarios are also impactful considerations when deciding your deployment's most optimal vGPU profile. When the number of monitors is increased, more pixels are being delivered to the screen. The NVIDIA Engineering team conducted benchmark testing to illustrate the increased advantages of VDI desktops configured with NVIDIA's vPC software over CPU-Only configured VDI desktops..

Chapter 8. Creating Your First vGPU Virtual Desktop

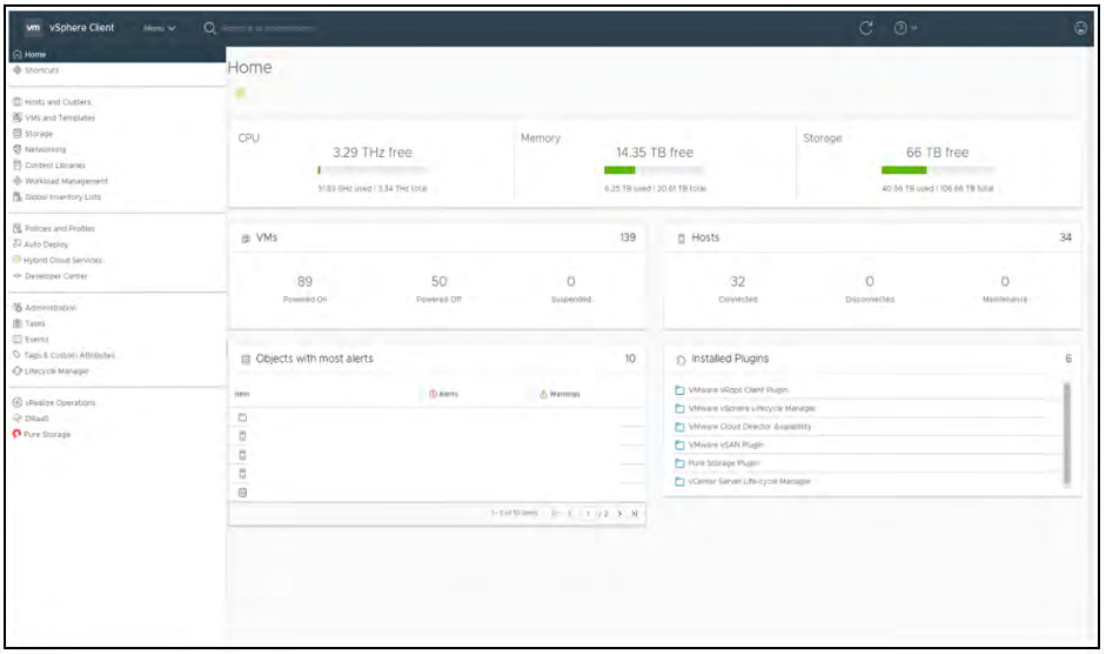
This chapter describes how to:

- Creating a Virtual Machine in vSphere
- Installing Microsoft Windows
- Installing VMware Tools
- Customize Windows settings
- Install Horizon Agent and Horizon Direct Connection on the VM
- Adjust additional VM settings and enable VM console access
- Enable the NVIDIA vGPU and finalize the installation

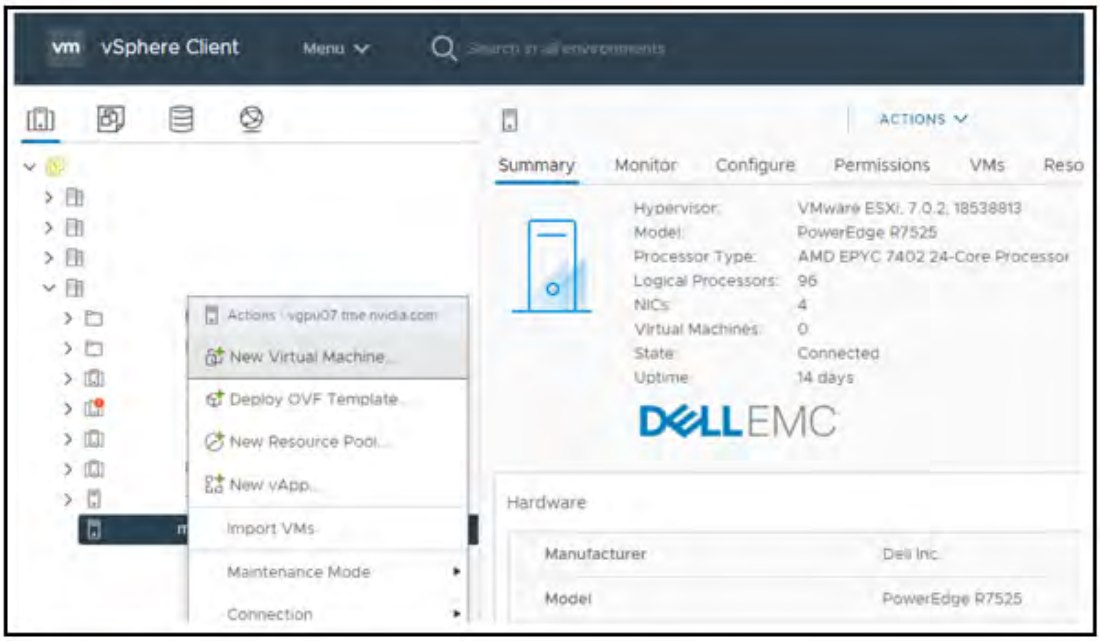
8.1 Creating a Virtual Machine in vSphere

These instructions assist in creating a VM from scratch to support NVIDIA vGPUs. This VM may be used as a gold master image to create additional VMs. Use the following procedure to configure a vGPU VM for a single guest desktop:

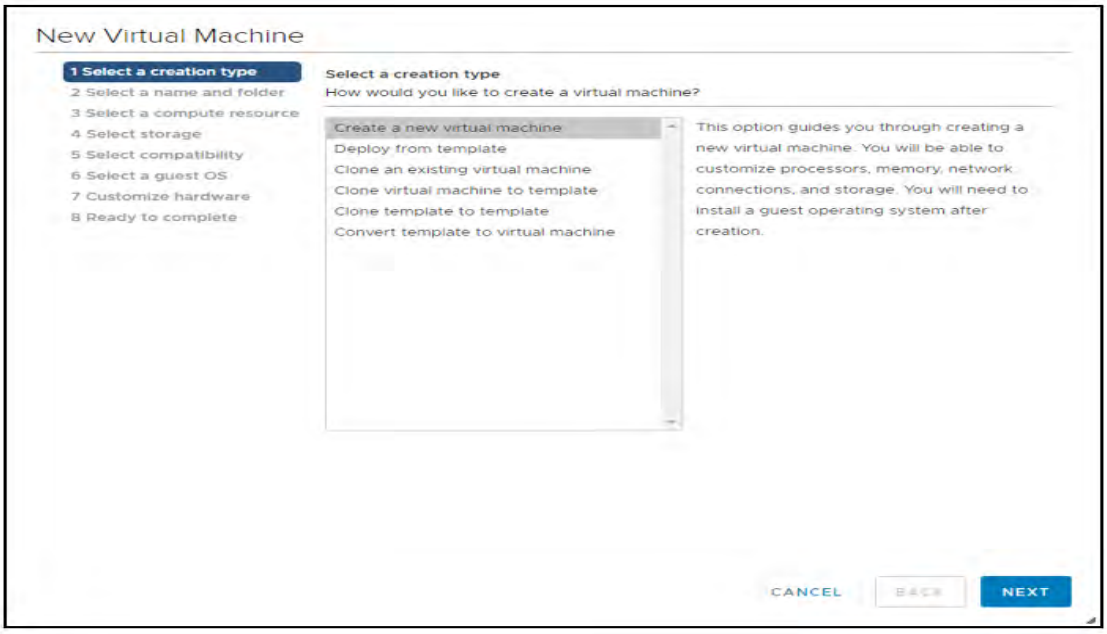
1. Open the vSphere Web Client:



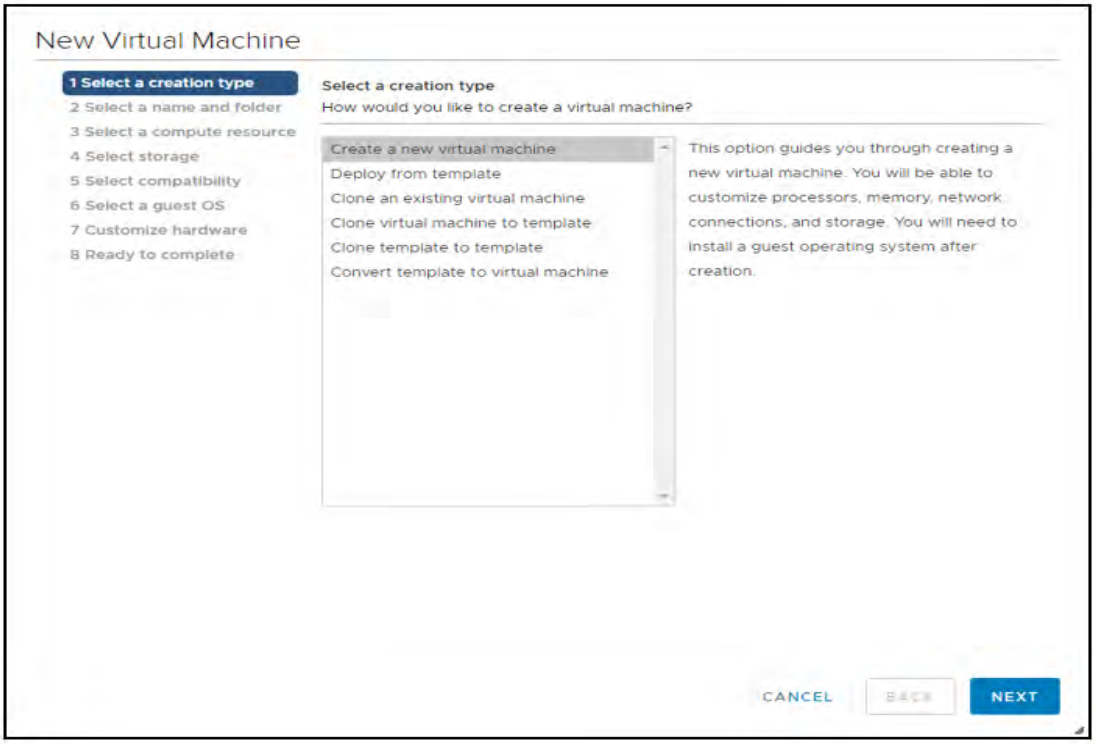
2. From the vSphere Web Client's **Home** page, select **Hosts and Clusters.** Right-click the host or cluster location the VM will be created. Select **New Virtual Machine**.



3. From the **Select a creation type** tab, select **Create a new virtual machine** and click Next.

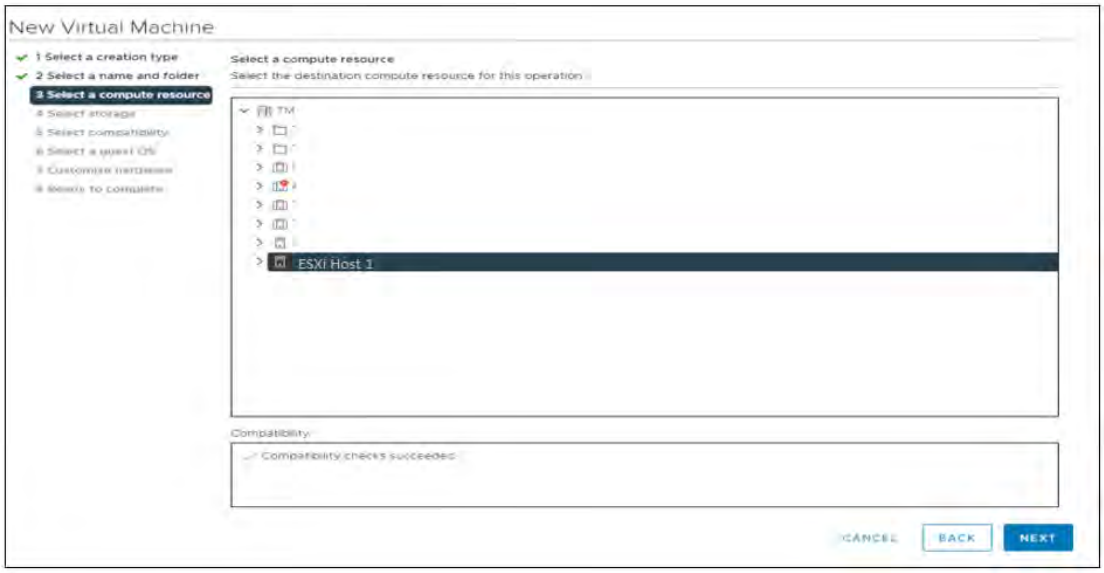


4. Enter a name for the virtual machine. Choose the location to host the virtual machine using the collapsible tree under the label. **Select a location for the virtual machine**, then click **Next** to continue.

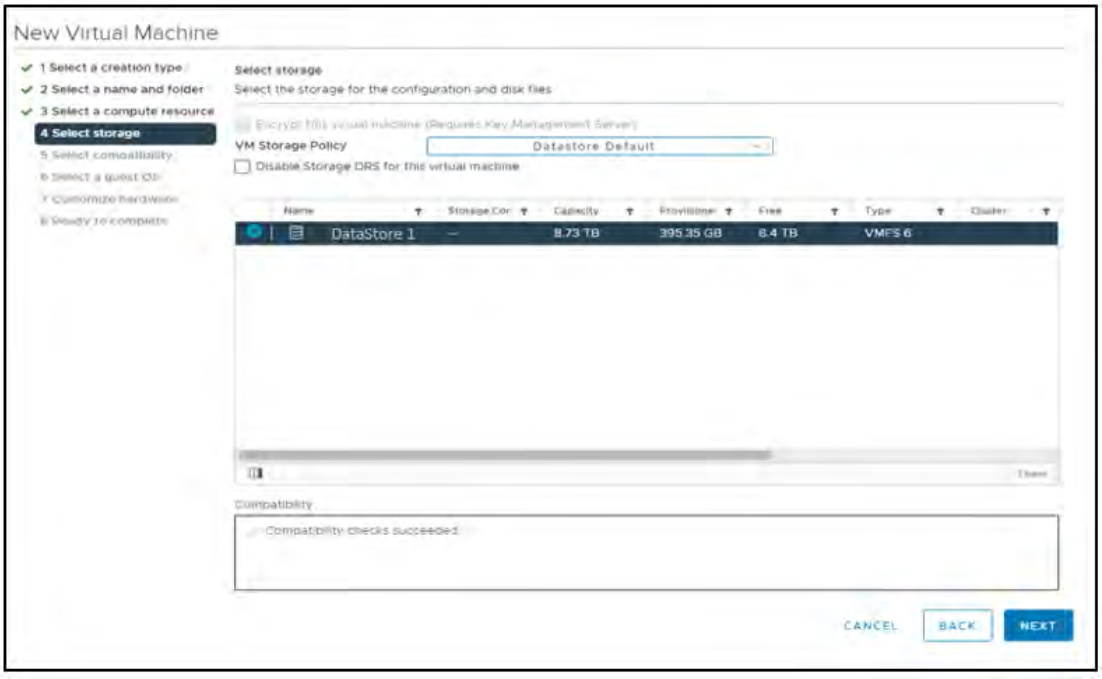


5. Select a compute resource to run the VM.

   The compute resource you select must include an installed, correctly configured NVIDIA adapter that supports vGPU operations. Click **Next** to continue. vSphere prompts you to select a storage resource:

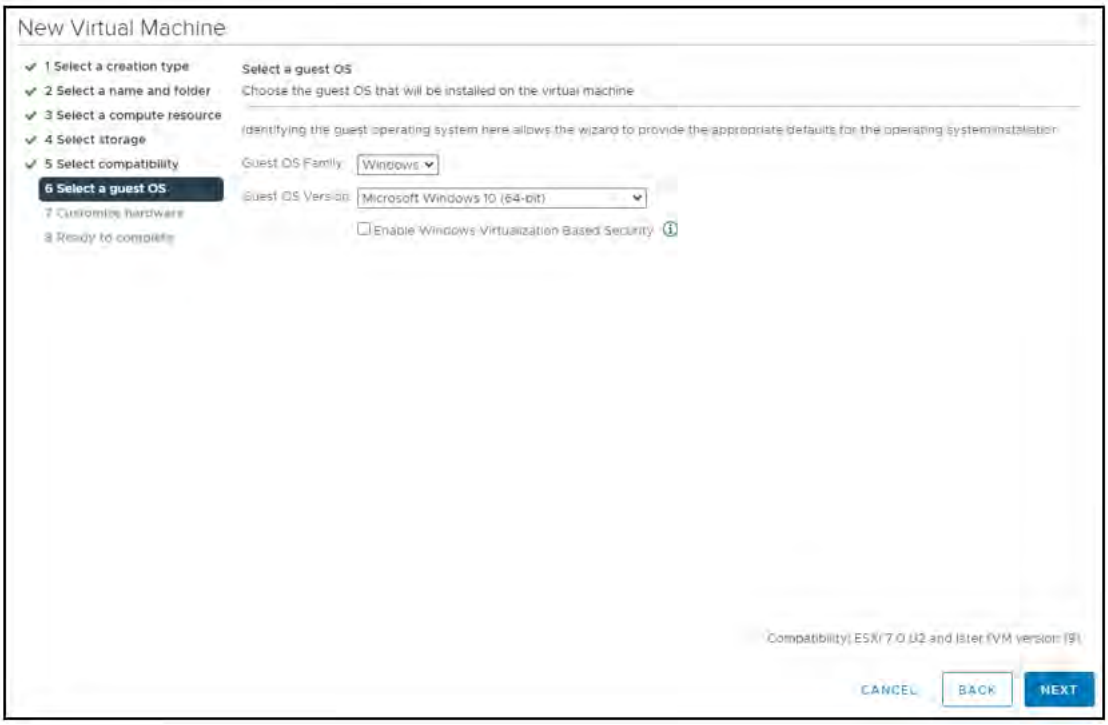


6. Select the datastore to host the virtual machine, then click **Next** to continue. vSphere prompts you to select compatibility for the vGPU:

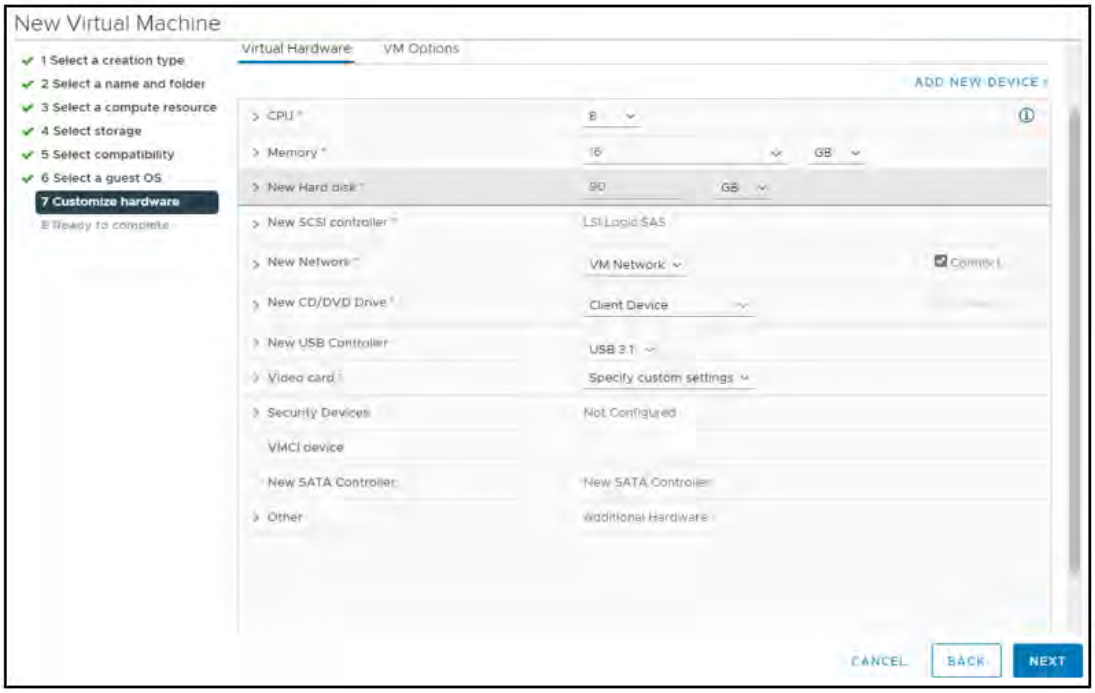


7. Compatibility settings allow the VM to run on different versions of vSphere. To run vGPUs, select ESXi 7.0 U2, and later, click **Next** to continue. vSphere prompts you to select a guest OS:

8. Select the appropriate Windows OS from the Guest OS Family and Guest OS Version dropdowns, then click **Next** to continue. vSphere prompts you to customize the vGPU's virtual hardware:



9. Set the virtual hardware based on your desktop workload requirements, then click **Next** to continue. vSphere displays a summary of the settings you have selected:



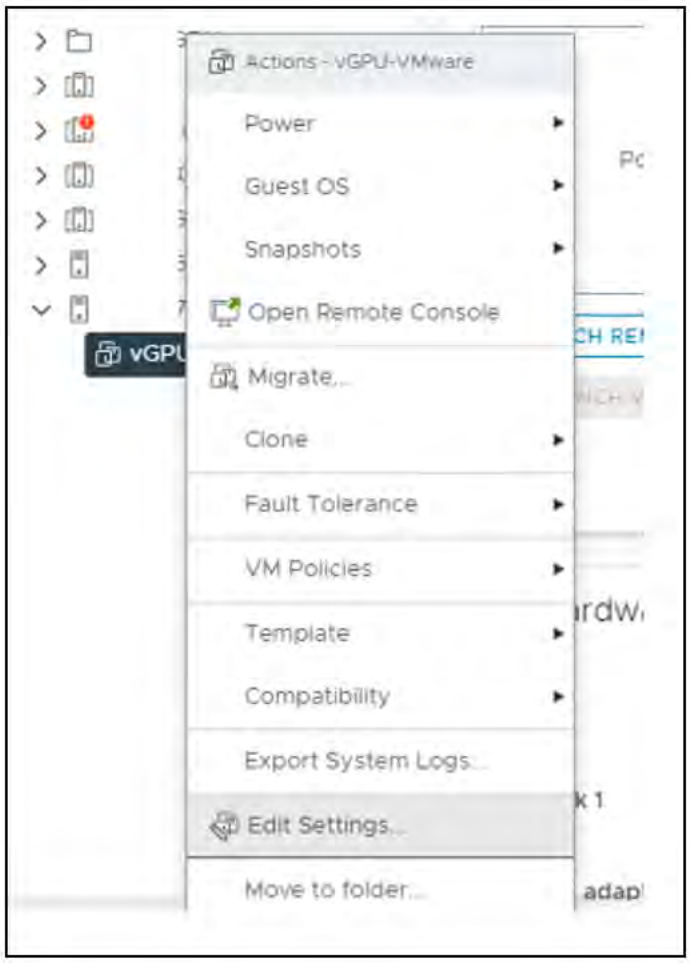
10. Review the new virtual machine's configuration. If any of the settings are wrong, click Back and correct them. When all of the settings are right, click Finish.

8.2 Installing Microsoft Windows

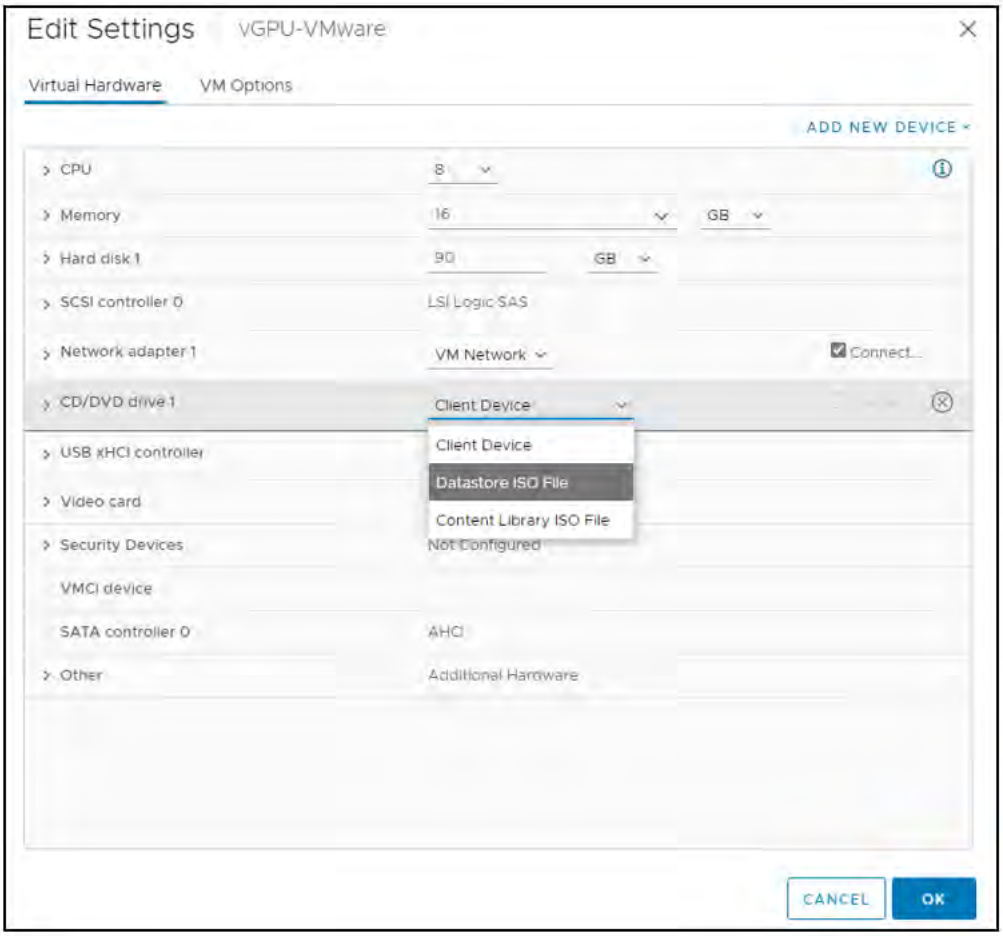
> **!** CAUTION: Windows 11 is used to capture content for this section in the guide. Text, images, and screen layout may differ from version to version. Refer here for the Windows 11 requirements.

To install Microsoft Windows on the virtual machine:

1. Select the virtual machine, right-click it, and select **Edit Settings**:

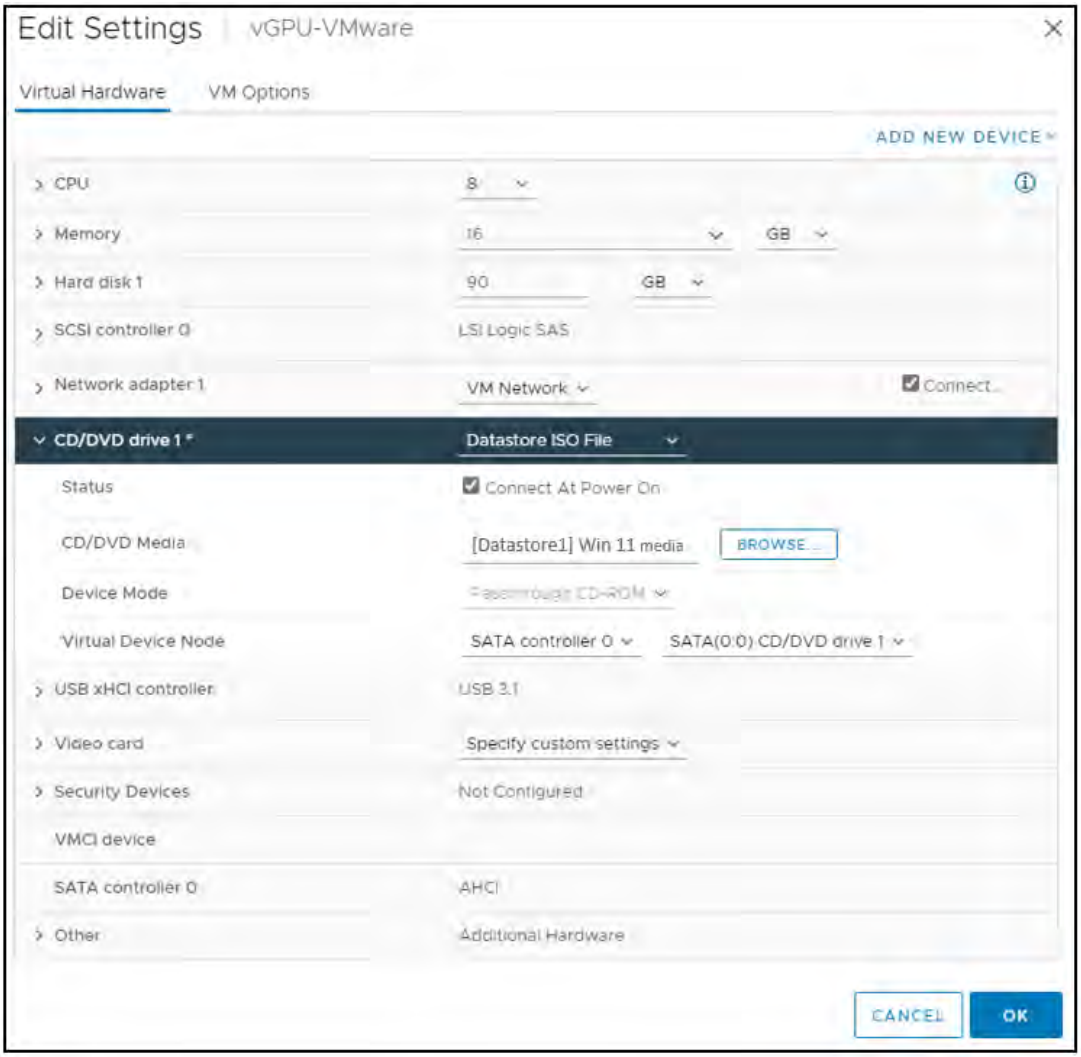


2. vSphere displays the **Edit Settings** window:

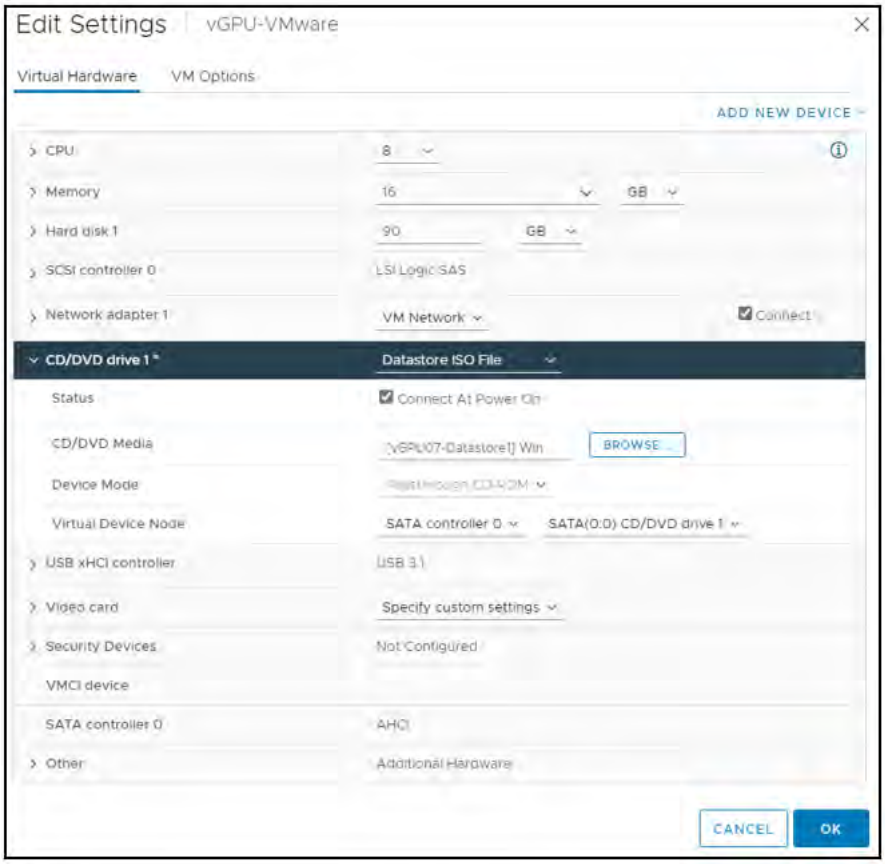


3. Click the dropdown list opposite the label “CD/DVD drive 1” and select an appropriate data source for CD/DVD media. (This example uses a Datastore ISO file.) Check the Connect checkbox to the right of the CD/DVD drive 1 dropdown to connect the ISO file to the VM’s virtual CD/DVD drive:

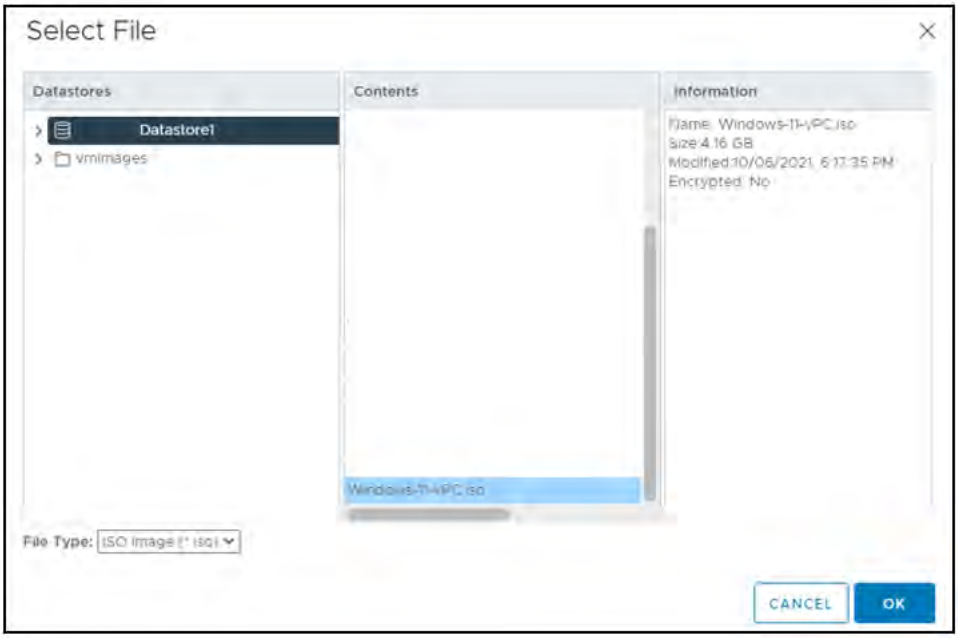
   Click the caret next to **CD/DVD drive 1**. vSphere expands the tree entry to reveal the details of the virtual device. Check the **Connect At Power On** checkbox to connect the ISO file to the VM’s virtual CD/DVD drive at boot time, then click the **Browse** button to the right of “CD/DVD Media.”



4. Check the **Connect At Power On** checkbox to connect the ISO file to the VM's virtual CD/DVD drive at boot time, then click the **Browse** button to the right of "CD/DVD Media." vSphere opens a **Select File** window:



5. Navigate to and select the OS ISO file for installation. Click **OK** to select the file.



6. Right-click the virtual machine, select Power>Power On to start the virtual machine, boot the ISO file, and install the operating system.

   The virtual machine boots from the selected ISO file.

> Suppose you create a new virtual machine and use vSphere Web Client's VM console functionality. In that case, the mouse may not be usable in the virtual machine until after both the operating system and the VMware tools have been installed.

7. Perform a Custom (fresh) installation of Microsoft Windows 11 on the virtual machine.

   During installation, Windows reboots the VM several times.

8. When Windows is installed, disconnect the ISO from the VM.
9. Go through the initial Windows setup wizard to name the computer, create a local account, set the time zone, choose the update installation policy, etc.

Windows 11 is now installed on the virtual machine.

8.3 Installing VMware Tools

After installing the Microsoft Windows OS on the virtual machine, you must install VMware Tools. VMware Tools is a suite of utilities that enhances the performance of virtual machines and their guest operating systems and improves the management of virtual machines. VMware tools provide many benefits for the VM and the Guest OS; Benefits include improved network adapter, Smoother mouse experience, copying and pasting, drag and drop files, Sound quality, and the ability to take quiesced snapshots of the Guest OS. Refer to the VMware Tools Services page for more information.

The VMware Tools Administration document describes the necessary steps for Manually Installing VMware Tools on Windows.

After going through the manual installation process in the provided link, VMware Tools is installed on the virtual machine.

8.4 Adding the Golden Master to the Domain

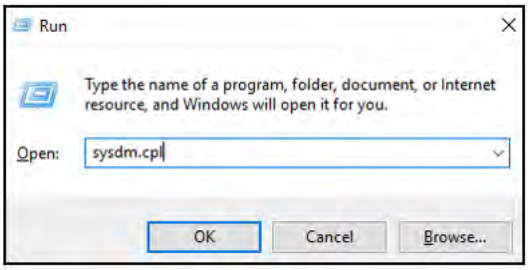
If you join a VM to a Windows Active Directory domain, you can manage it as you would any physical desktop in the domain.

Customize Windows on the virtual machine as follows:

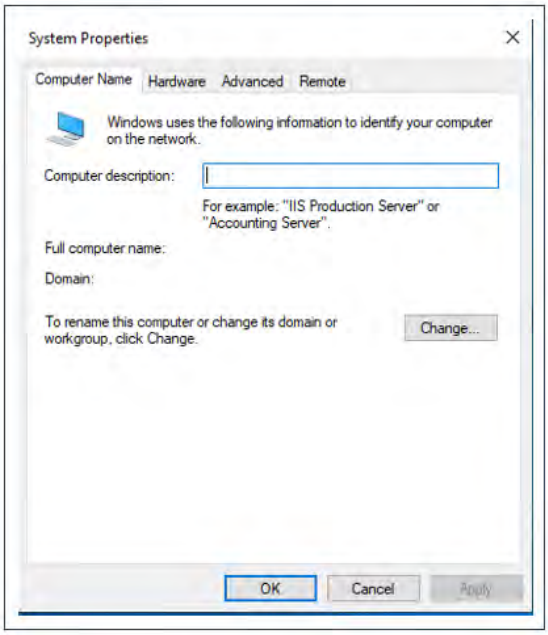
- Join the domain
- Add appropriate Domain groups to Local Administrators

Adding a VM to the domain:

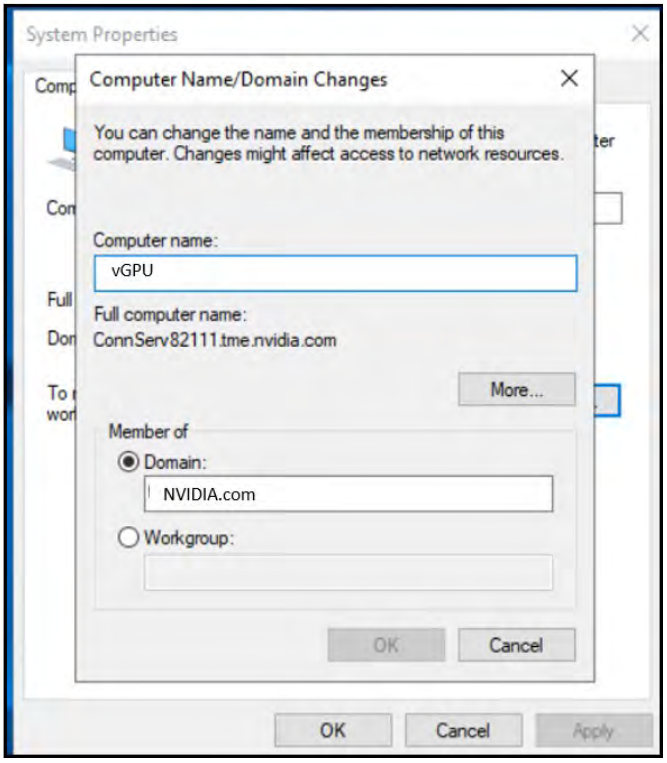
2. In the Windows Desktop. Open the Run dialog box by pressing the Windows-Logo Key + R. Type “sysdm.cpl” and press enter in the command box. The **System Properties** window opens.



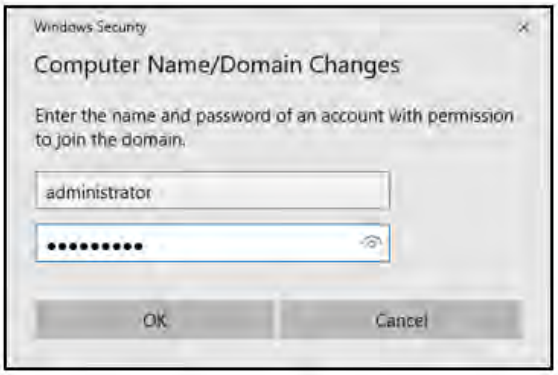
3. Enter a Computer description in the **System Properties** window (Optional). Select the change button.



4. The **Computer Name/Domain Changes** window opens. Enter an appropriate name in the Computer name field. Select Domain and enter a proper name in the Domain name field. (The names in the illustration are examples and are not necessarily appropriate for your VM.)

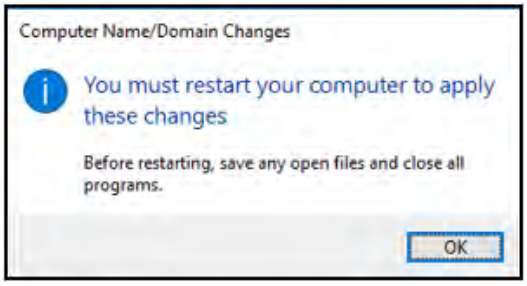


5. Enter your domain administrator credentials in the **Windows Security** window, click OK. If the credentials are valid, Windows sets the domain name as you have specified it and opens a "Welcome…" window.



6. Click OK to close the Domain Welcome window.

7. Windows displays a "You must restart your computer…" window.



8. Click OK. The VM reboots when you close the **System Properties** window.

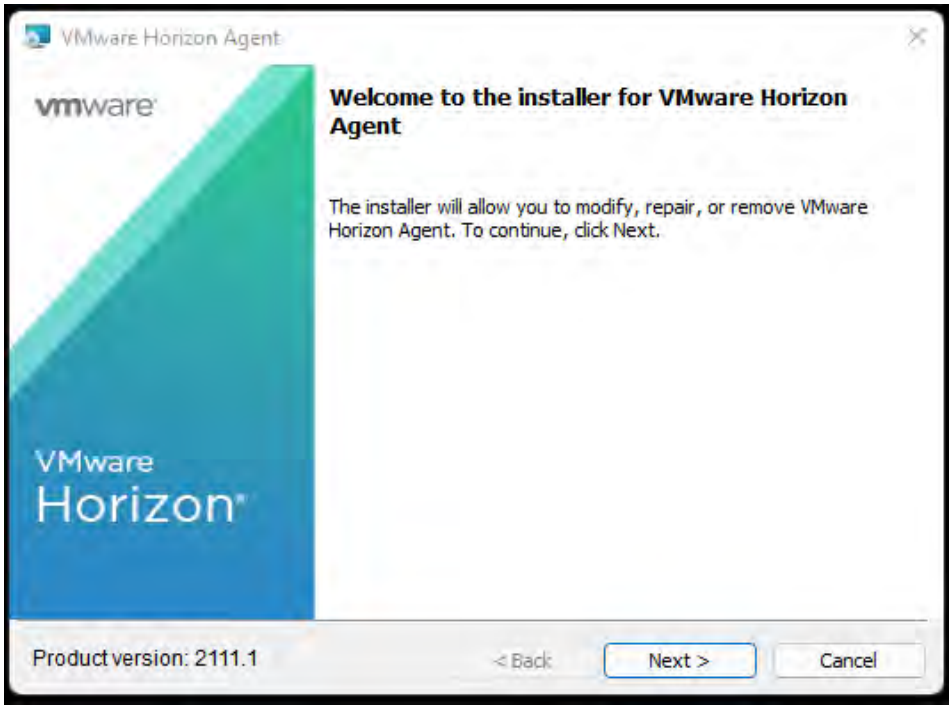
8.5 Installing Horizon Agent

You must install the correct version of the Horizon Agent for your virtual machine.

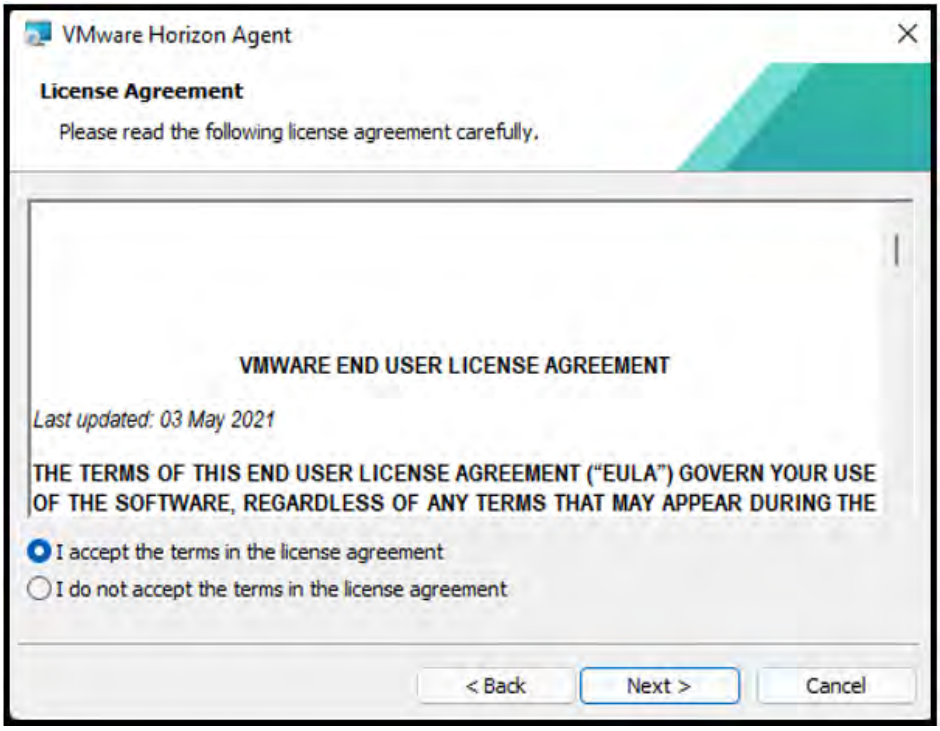
> **!** CAUTION: VMware Horizon 8 2111 is used to capture content for this guide. Text, images, and screen layout may differ from version to version.

To install Horizon Agent on your VM:

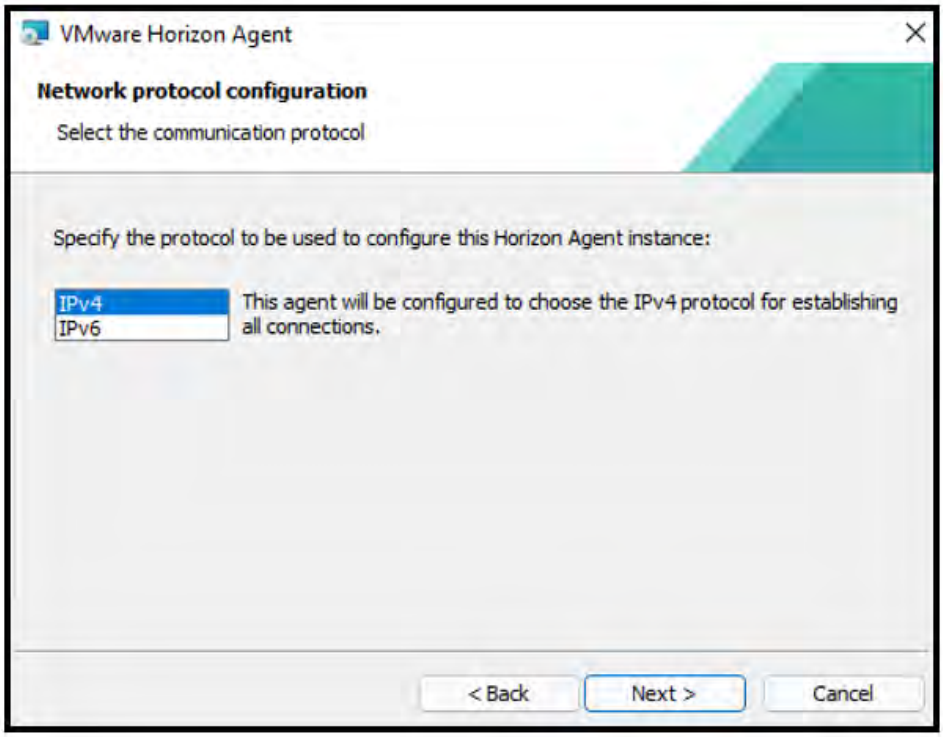
1. On the virtual machine where the software will be installed, download the VMware Horizon 8 2111 software and extract the installers.
    - For 64-bit virtual machines, install VMware-Horizon-Agent-x86_64-2111.1-8.4.0-19066669.
2. Launch the installer. It displays the “VMware Horizon Agent” Welcome window. Select *Next*.



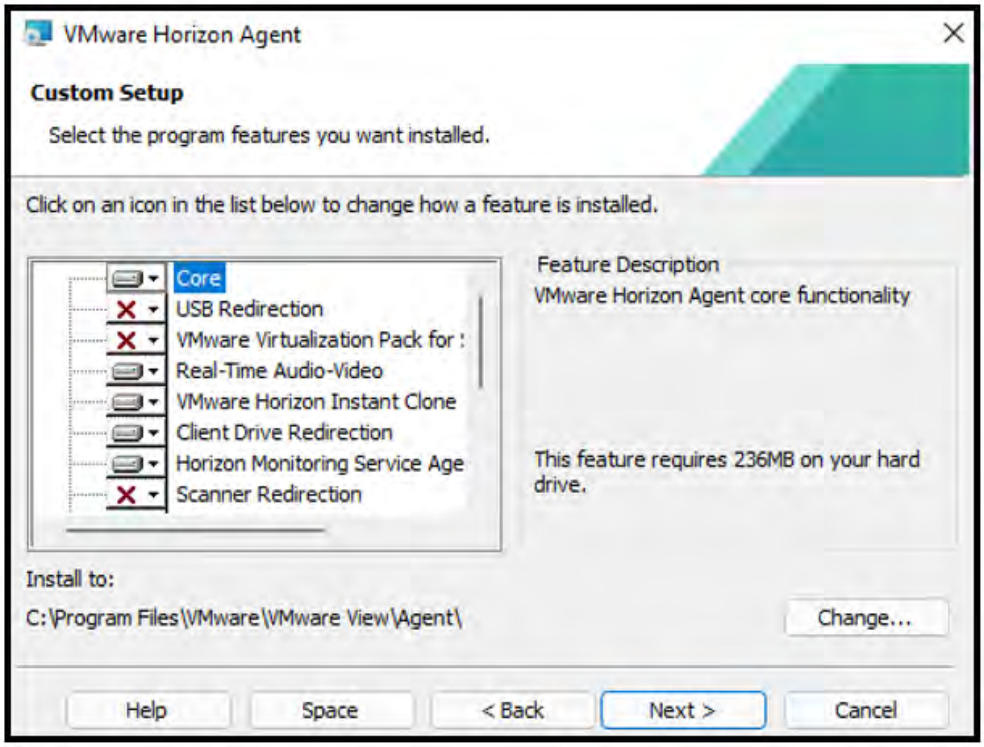
3. The installer displays the **License Agreement** screen. Select the "*I accept the terms…*" radio button. Then click next.



4. The installer displays the Network protocol configuration screen. Choose the appropriate network protocol (typically IPv4), then click Next to continue.

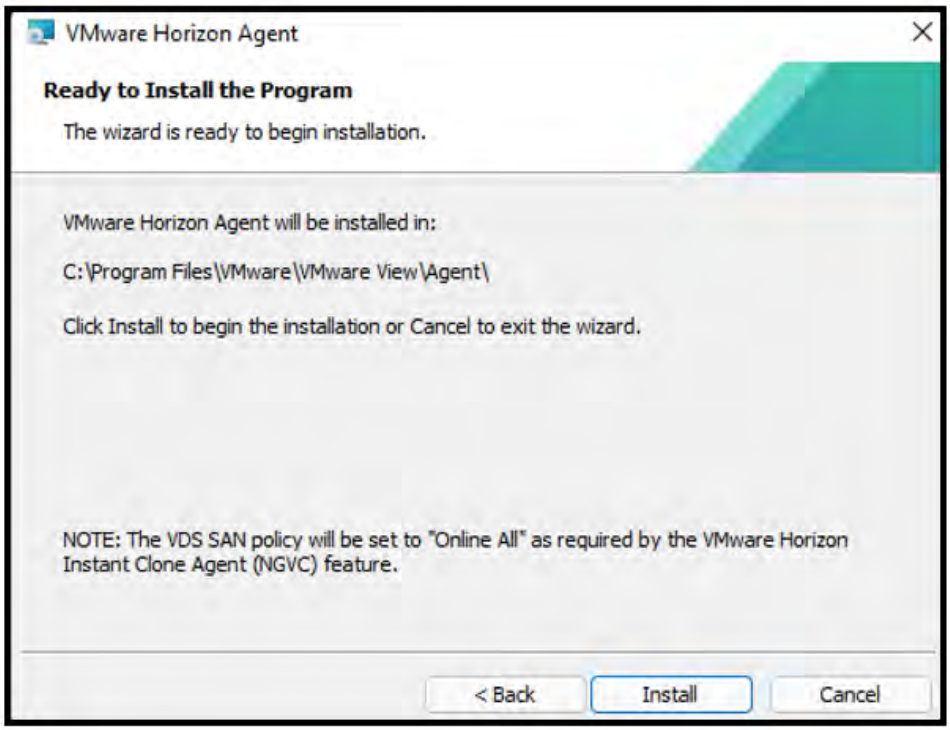


5. The installer displays the **Custom Setup** window. Accept all default options; click Next.

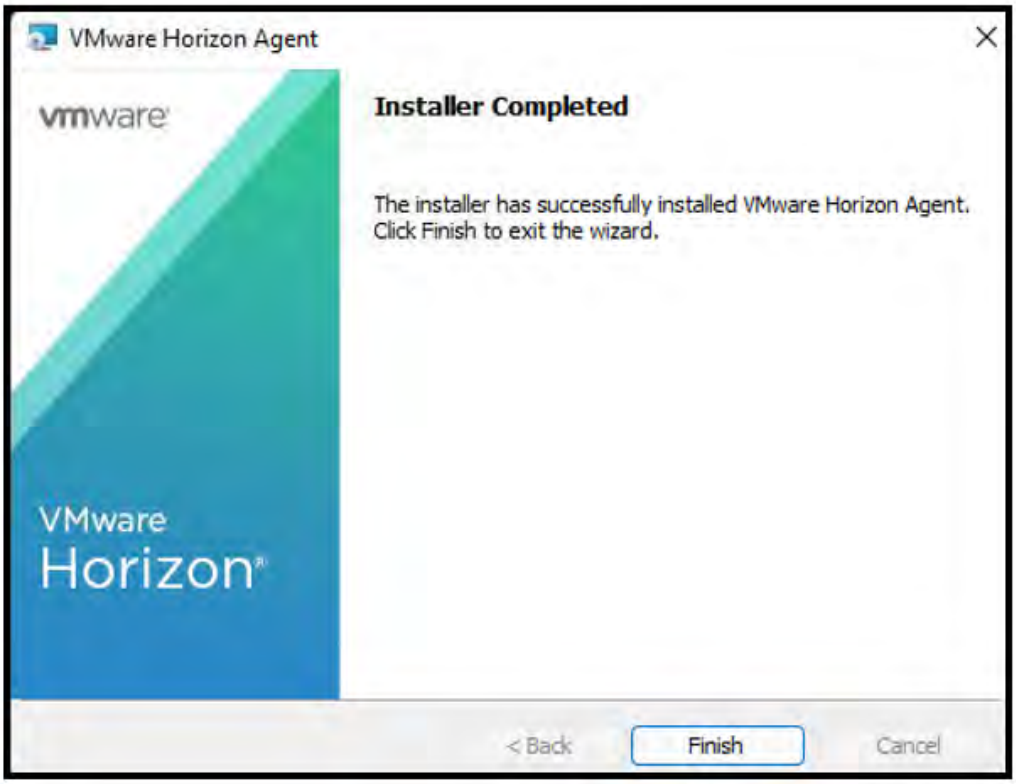


Note: The Horizon Agent Custom Setup Options article should be used to learn more about the default options and additional custom options.

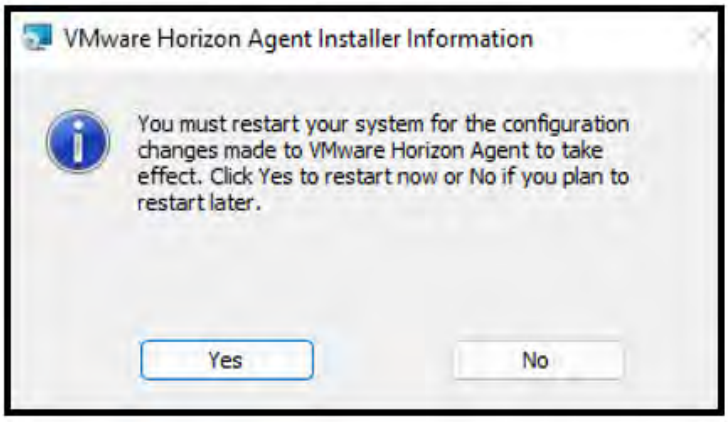
6. The installer displays the **Ready to install the program** window. Select *Install* to continue. The installer installs Horizon Agent on the VM.



7. The **Installer Completed** window shows the installation completed successfully. Select **Finish** to close the installer. The installer prompts you to reboot:



8. The installer prompts you to reboot. Select **Yes** to reboot the VM.



9. VMware Horizon Agent is installed on the virtual machine.

8.6 Installing Horizon Direct Connection

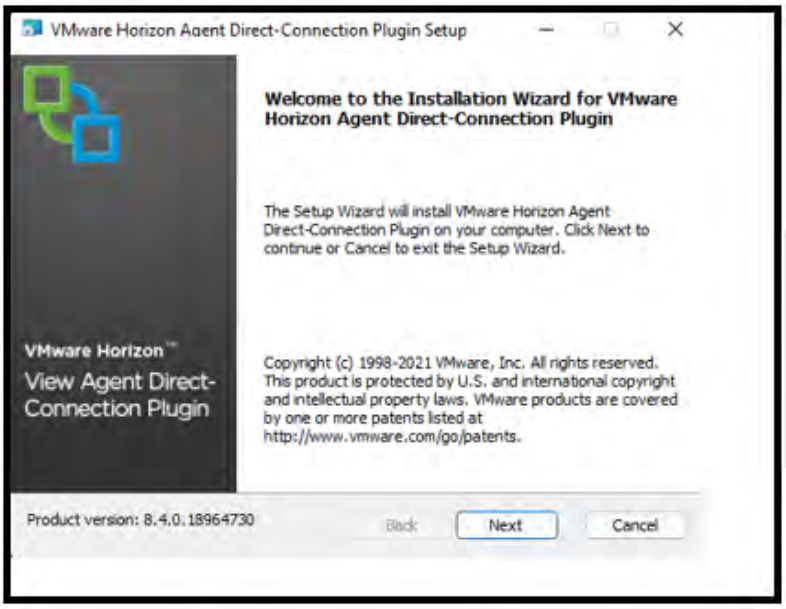
The Horizon Direct Connection agent is a helpful tool for debugging when you configure the virtual machines with a vGPU. Install the correct version for your VM.

You must install the correct version of the Horizon Agent Direct-Connect for your virtual machine.

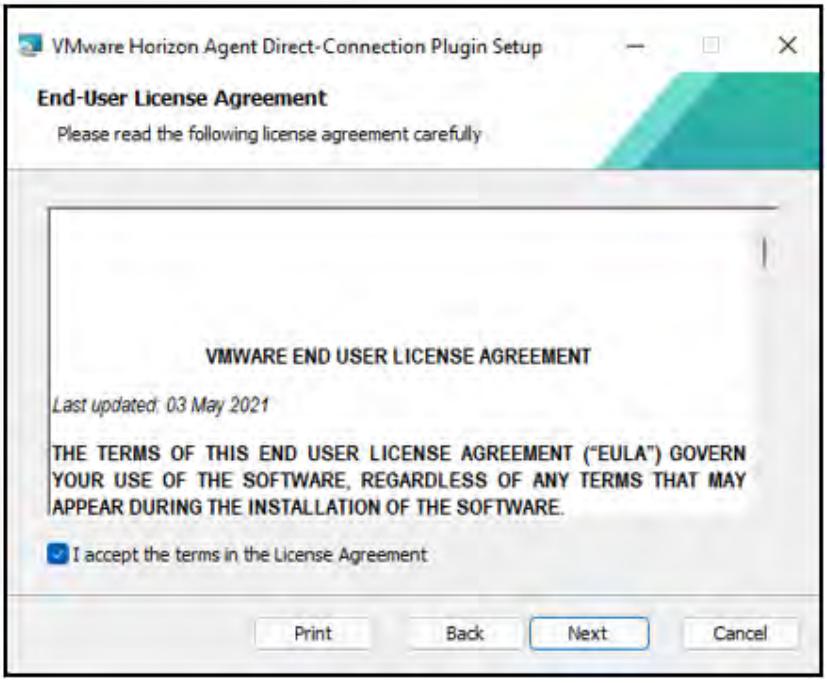
!	CAUTION: VMware Horizon 8 2111 is used to capture content for this guide. Text, images, and screen layout may differ from version to version.

To install the Horizon Agent Direct-Connect on your VM:

1. On the virtual machine, download the VMware Horizon 8 2111 software and extract the installers.
   - For 64-bit virtual machines, install `VMware–Horizon-Agent-Direct-Connet-x86-8-4-0-18964730`
2. Launch the installer. It displays the "**VMware Horizon Agent Direct-Connect Plugin Setup**" window. Select *Next*.

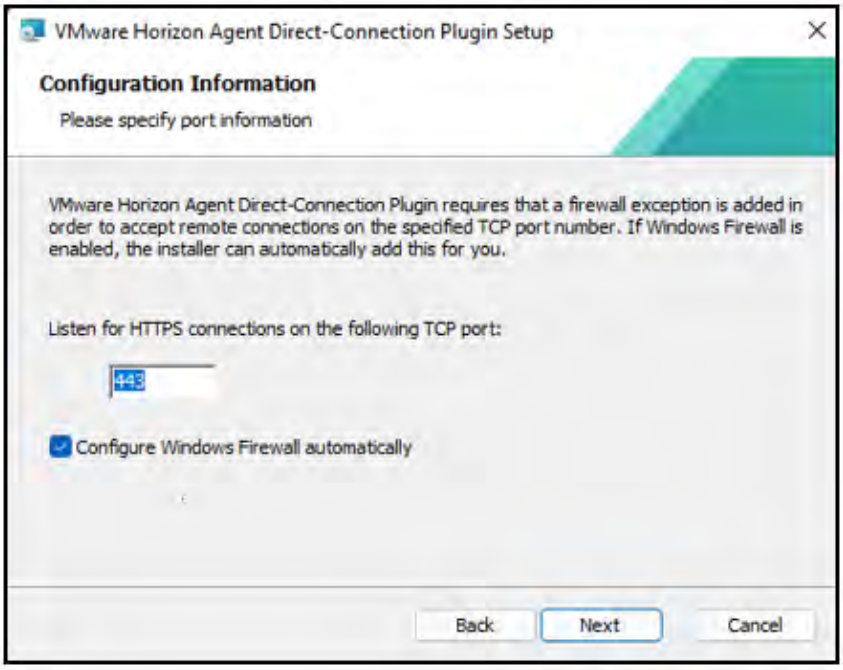


3. The installer displays the **License Agreement** screen. Select the "*I accept the terms…*" radio button. Then click next.

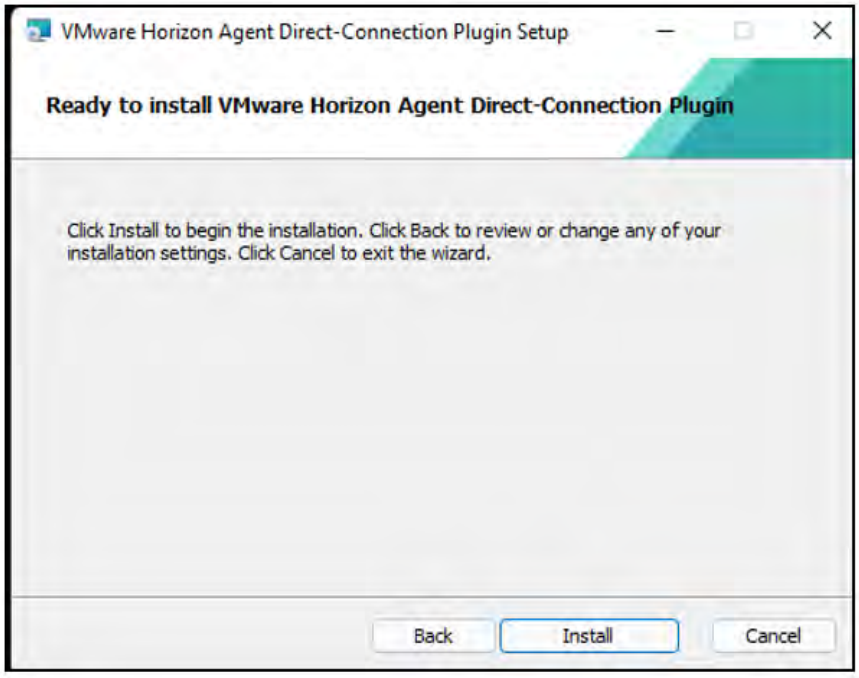


4. Enter an appropriate port number in the Listen for HTTPS connections field. The default HTTPS port number is 443; however, you can use a different port.

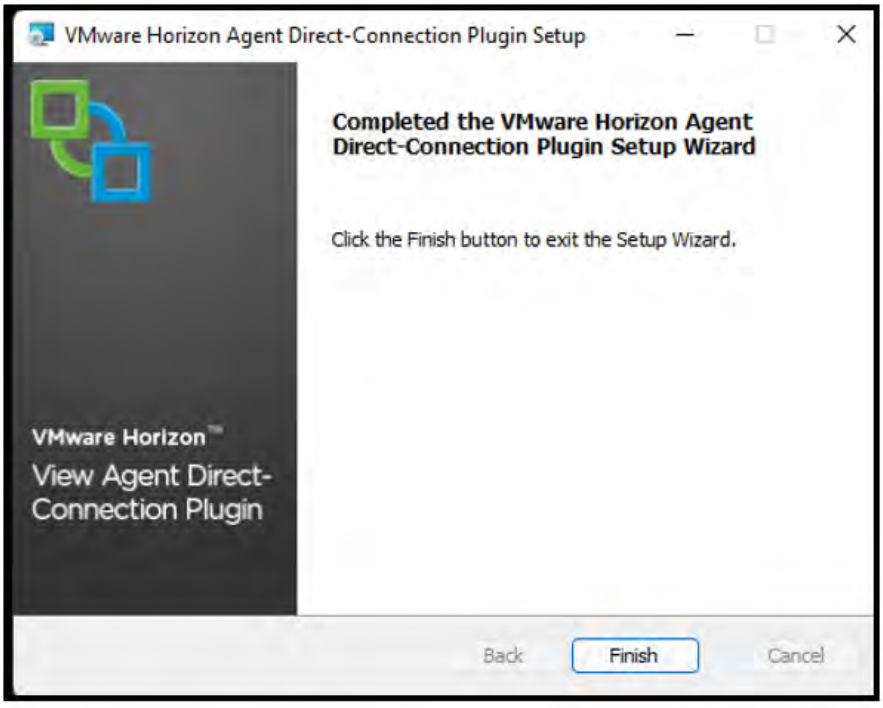
   Check the Configure Windows Firewall automatically checkbox. Then click Next to continue.



5. The installer displays the **Ready to install VMware Horizon Agent Direct-Connection Plugin** window. Select **Install** to continue. The installer installs Horizon Agent Direct-Connect on the VM.



6. The **Installer Completed** window shows the installation completed successfully. Select **Finish** to complete the install.



7. When Windows has fully booted, test the Direct Connection using the installed VMware Horizon client by adding a server, entering the IP address of the VM, and logging in to the VM with the appropriate credentials when you are prompted to do so.

Horizon Direct Connection is installed on the virtual machine.

8.7 Optimizing Windows

Windows OS Optimization Tool for VMware Horizon is VMWare's official release of the Optimization Tool. The Windows OS Optimization Tool provides the easiest way to optimize windows desktops and server master images for VMware Horizon. The optimization tool includes customizable templates to enable or disable Windows system services and features across multiple systems, per VMware recommendations and best practices. Since most Windows system services are enabled by default, the optimization tool can easily disable unnecessary services and allow features to improve performance.

You can perform the following actions using the VMware OS Optimization Tool:

- Local analysis and optimization
- Remote analysis
- Optimization history recording and rollback
- Template management

Review the VMware Horizon Optimization Guide for Windows for OS-specific tweaks.

With the release of Windows OS Optimization Tool for Horizon version 1.0 (2111), a single template is built that includes support for all versions of Windows 10, Windows 11, Windows Server 2019, and Windows Server 2022.

Note: VMware recommends using this tool in a development lab before running in a production environment. It would be best to understand the recommended settings before applying any changes, as this could have adverse effects and can cause damage to your deployment.

8.8 Additional Virtual Machine Settings

Perform the following additional tasks on the virtual machine as required in preparation for configuring its vGPU:

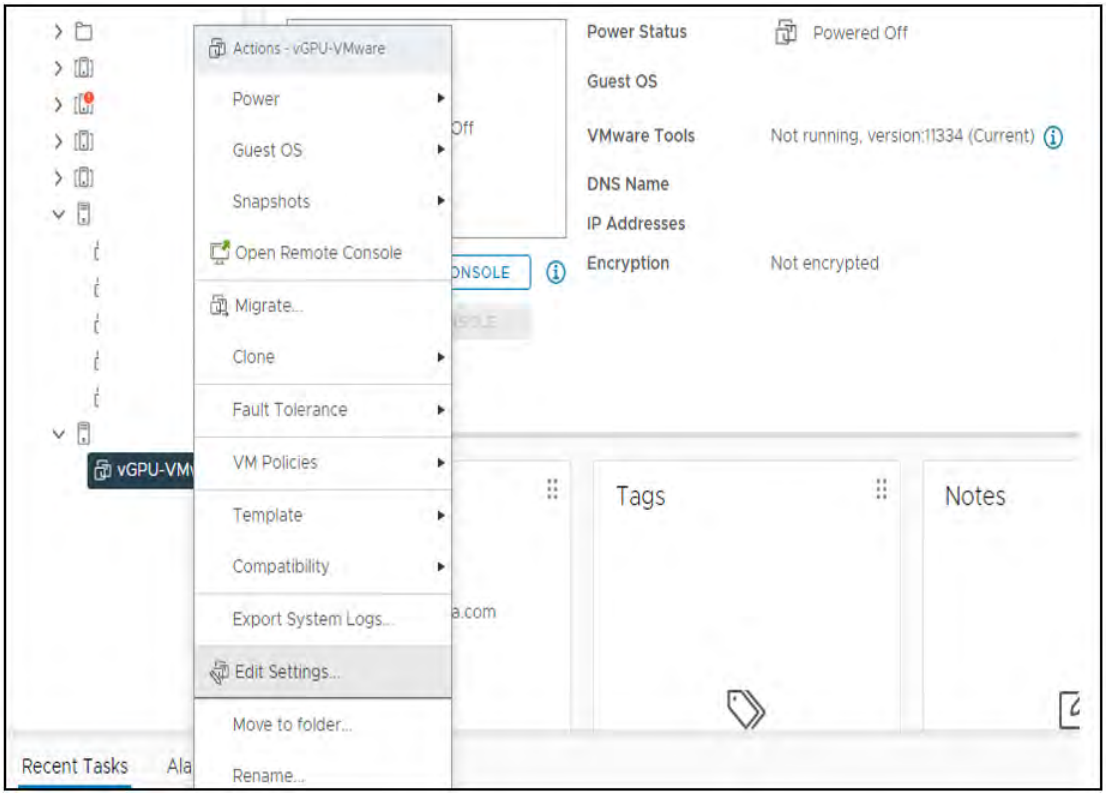
1. . Turn Off Windows Firewall for all network types.

   CAUTION: These instructions assume that the VM is for proof-of-concept only and that disabling the firewall poses only a minimal security breach. Always follow your established security procedures and best practices when setting up security for a production machine or for any environment that can be accessed from outside your network.

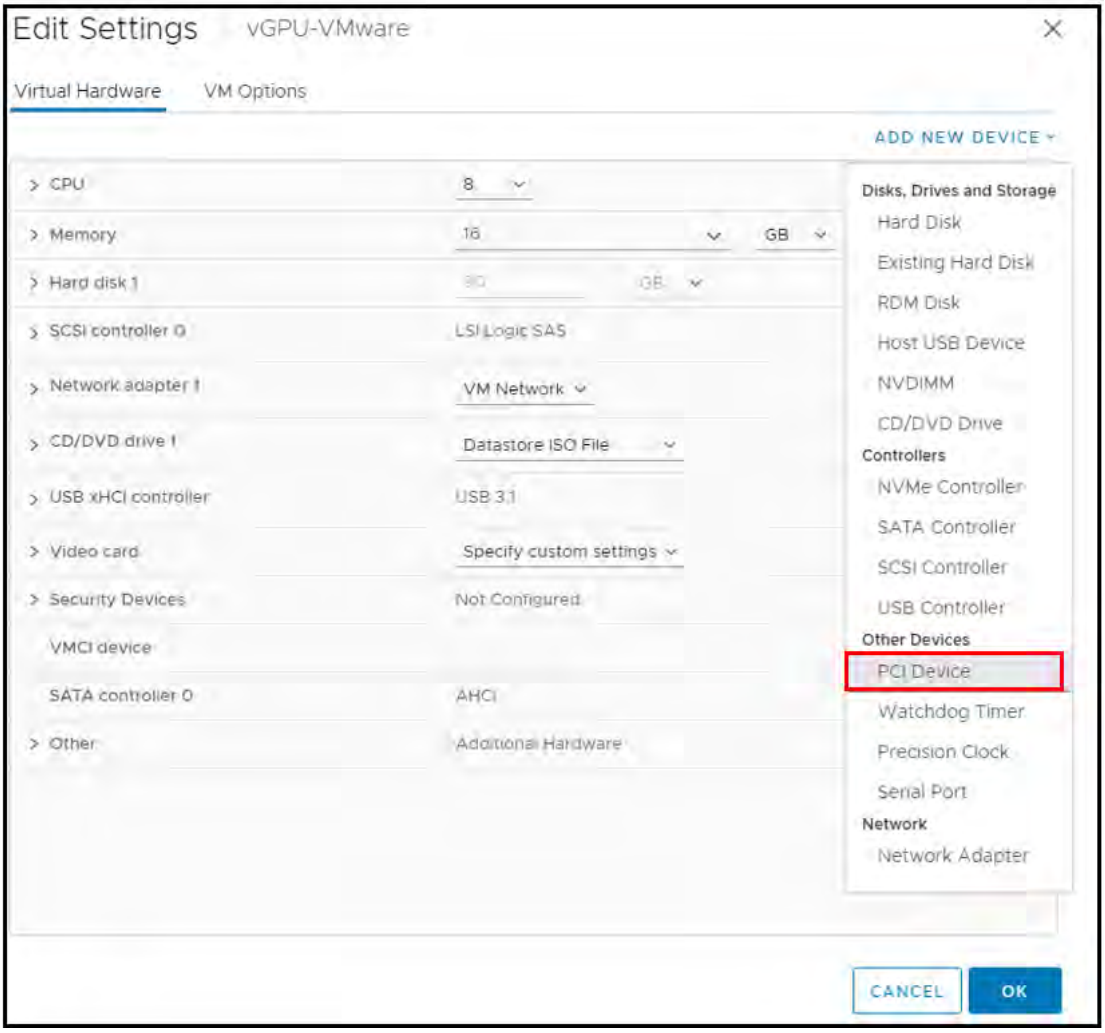
8.9 Enabling the NVIDIA vGPU

The following steps will enable vGPU support for Windows OS virtual machines and Linux OS virtual machines. The virtual machine settings must be edited.

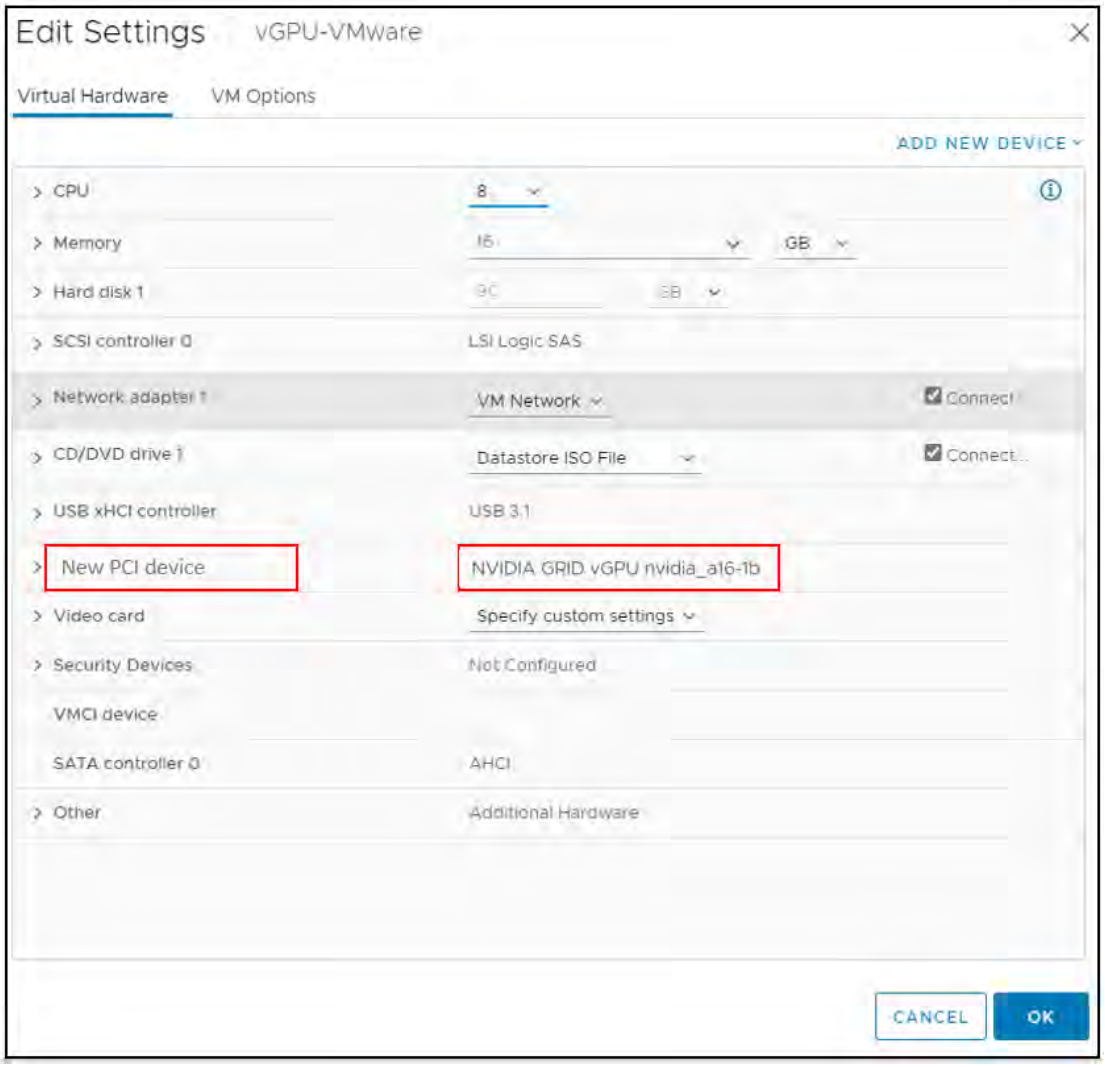
1. Power down the VM.
2. Click on your VM in the inventory window: Right-click your VM and select **Edit Settings**.



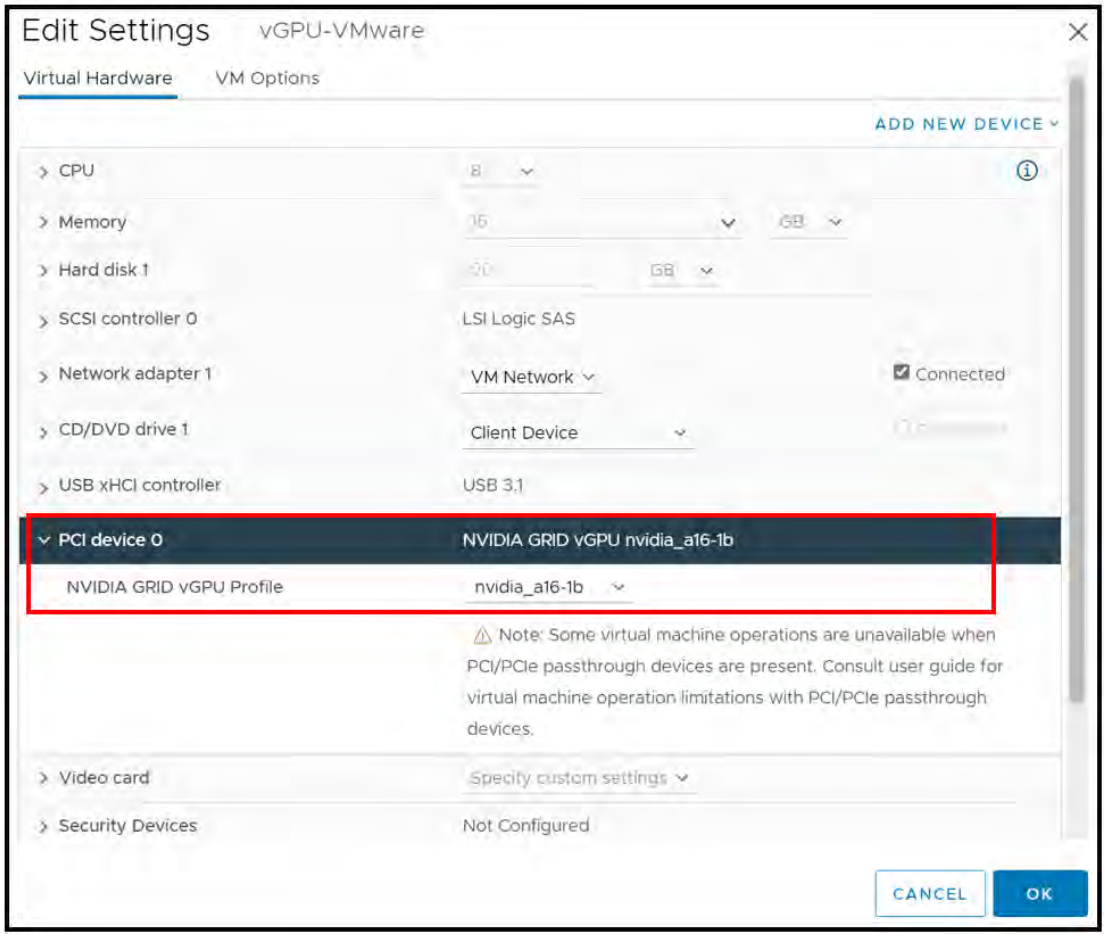
3. Select **Add New Device**, then Select **PCI device** from the dropdown menu.



4. The new PCI device shows that an NVIDIA vGPU device has been added.



5. Expand the **New PCI device**, expand the **NVIDIA GRID VGPU Profile** dropdown and select your vGPU Profile.



6. Click **OK** to complete the configuration.

8.10 Installing the NVIDIA vGPU Driver: Microsoft Windows

After you create a Microsoft Windows virtual machine on the hypervisor and boot the VM, you must install the NVIDIA vGPU software display driver to enable GPU operation fully.

To install the NVIDIA driver in Microsoft Windows:

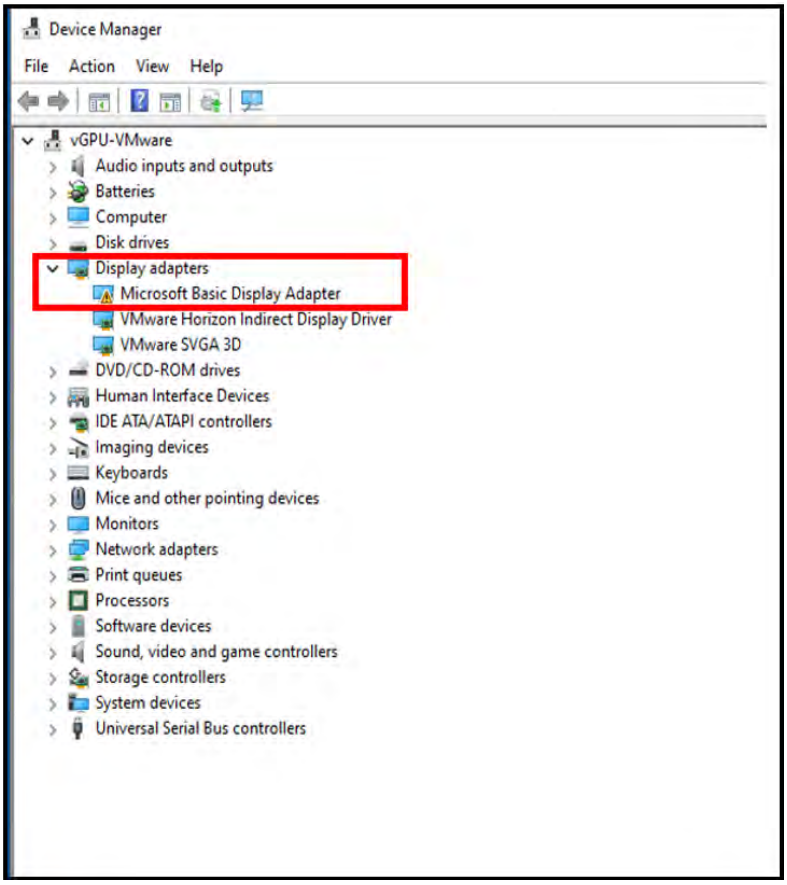
1. Start the virtual machine, then connect to it using either VMware Remote Console through the vSphere Web Client or VMware Horizon Client (via Direct Connection).

   The first time you boot the VM after enabling an NVIDIA vGPU, it displays a dialog warning requesting that you restart the computer to apply changes. Click Restart Later to continue booting the VM.

   CAUTION: Do not reboot the VM if older NVIDIA drivers are installed. Doing so would produce a blue screen.

2. Log in to Windows and open Device Manager.

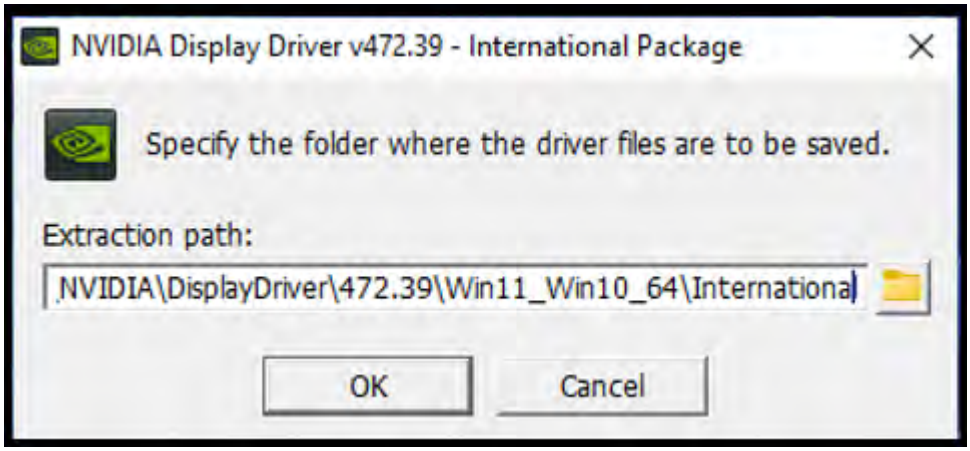
   The window's "Display adapters" section shows a "Microsoft Basic Display Adapter" with an exclamation point on its icon to indicate a driver problem. This is normal.



3. Locate the NVIDIA driver and double-click its Setup icon to launch it.

NVIDIA recommends that the installer share volume that the VM can mount for quick access.

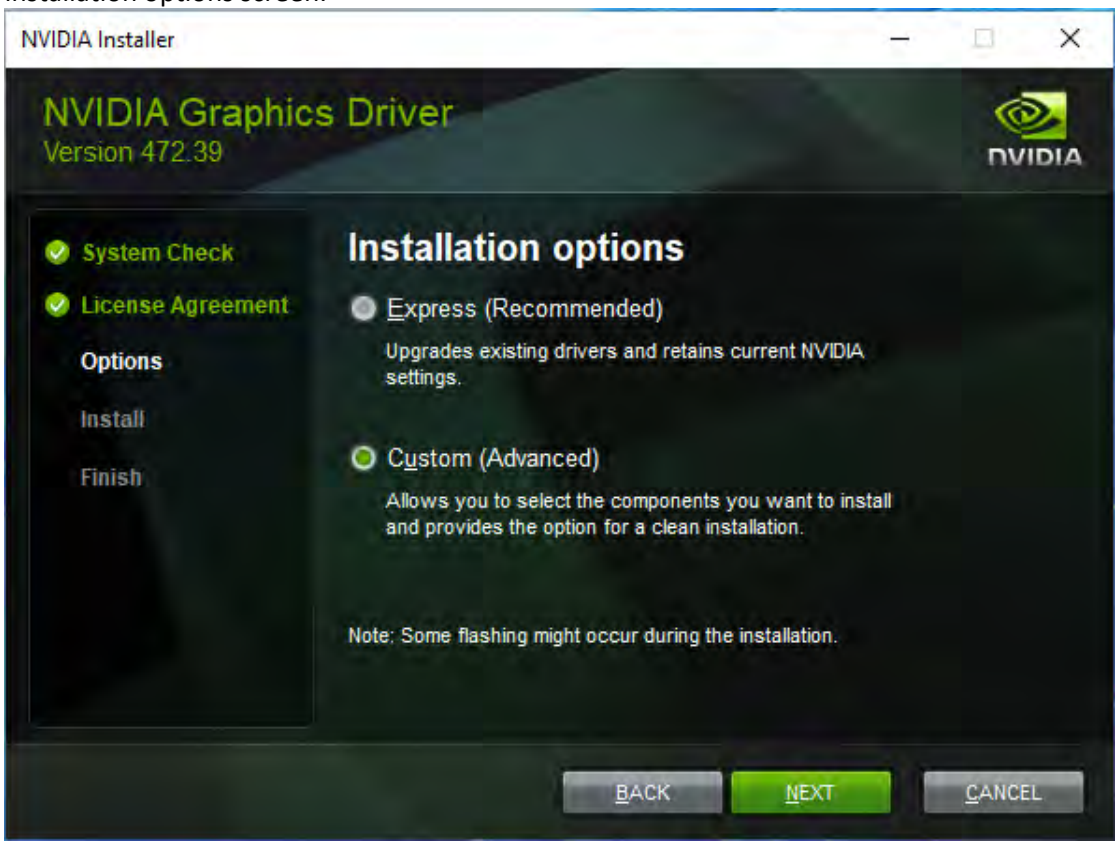
4. Click OK to accept the default driver directory.



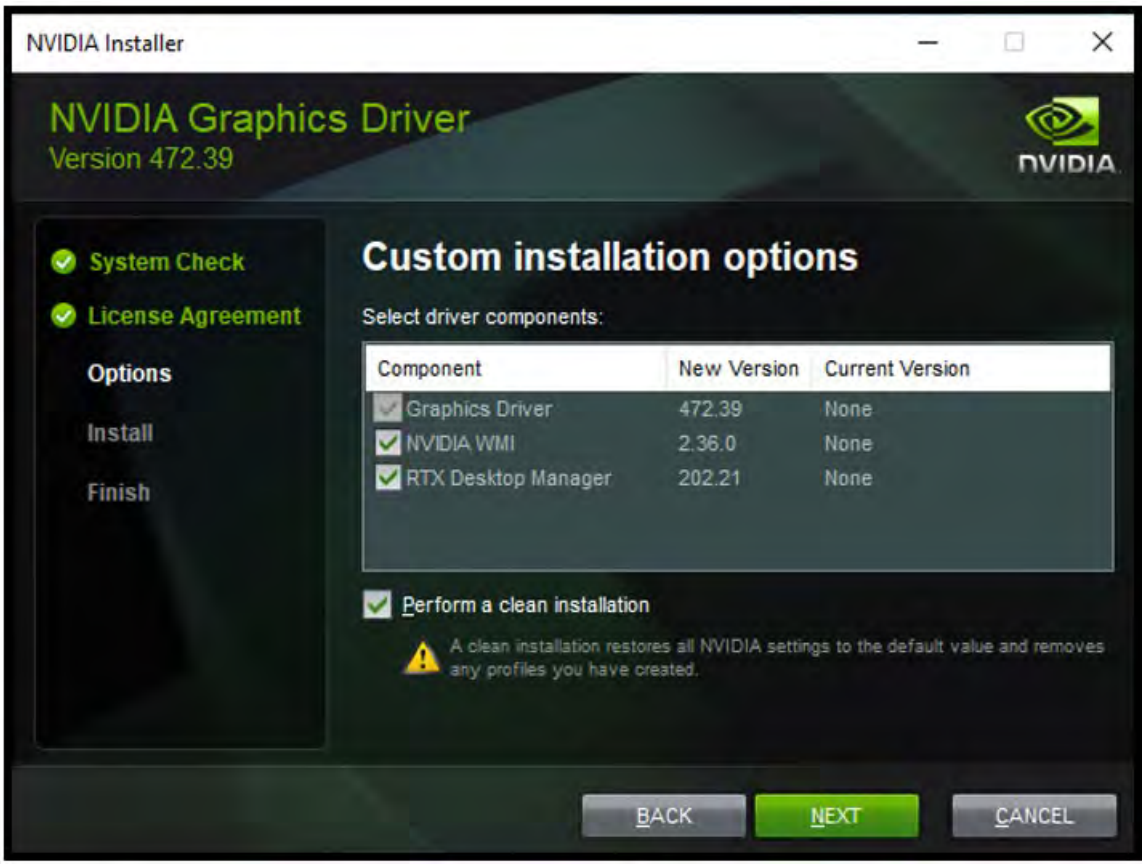
5. Read through the **NVIDIA software license agreement**. Click **OK** to **Agree and Continue**.


NVIDIA Installer
NVIDIA Graphics Driver
Version 472.39
NVIDIA
System Check
License Agreement
Options
Install
Finish
NVIDIA software license agreement
Please read the following NVIDIA software license agreement carefully.
END USER LICENSE AGREEMENT Release Date: September 4, 2015
NVIDIA GRID SOFTWARE END-USER LICENSE AGREEMENT
IMPORTANT - READ BEFORE DOWNLOADING, INSTALLING, COPYING OR USING THE LICENSED SOFTWARE
READ CAREFULLY: This Enterprise End User License Agreement ("EULA"), made and
Click Agree and Continue if you accept the terms of the agreement.
AGREE AND CONTINUE
CANCEL

6. Click the Custom (Advanced) radio button, then click Next. The installer displays the Custom installation options screen:



7. Check the Perform a clean installation checkbox, then click Next. The installer begins installing the driver.



8. Click **Restart Now** to restart the VM and complete the install.

After restarting the VM, the mouse cursor may not track properly using VNC or the VMware Remote Console (VMRC). In that case, use the View Agent Direct-Connect (VADC) to connect directly to the VM.

8.11 Installing the NVIDIA vGPU Driver: Linux

Create a Linux VM configured with a virtual GPU. This section will install the vGPU driver and license the vGPU software for full functionality within the VM.

In this guide, the Ubuntu 20.04 LTS operating system is used. It is important to note there are two Ubuntu ISO types: Desktop and Live Server. The Desktop version includes a graphical user interface (GUI), while the Live Server version only operates via a command line. This document uses the Live Server version 20.04 (amd64 architecture) of Ubuntu. If needed, a GUI can be installed later.

8.11.1 Installing the vGPU Driver in the Ubuntu VM

After you have created the Linux VM on the hypervisor and have booted the VM, install the NVIDIA vGPU software graphics driver in the VM to enable GPU operations fully.

Note: 64-bit Linux guest VMs are supported only on Q-series, C-series, and B-series NVIDIA vGPU types. They are not supported on A-series NVIDIA vGPU types.

Note: The procedure for installing the driver is the same in a VM and bare metal.

Prerequisites

Installing the NVIDIA vGPU software display driver for Linux requires:

- Compiler toolchain
- Kernel headers

To install the NVIDIA vGPU driver for Linux

1. Use WinScp to copy the NVIDIA vGPU software Linux driver package to the Ubuntu vGPU VM.

   The NVIDIA vGPU driver for Linux uses this naming convention or a similar one: NVIDIA-Linux_x86_64-470.82.01-grid.run

2. Log in to the VM and check for updates.

   ```
   $ sudo apt-get update
   ```

3. Install the GCC compiler and make the tool in the terminal.

   ```
   $ sudo apt-get install build-essential
   ```

4. Navigate to the directory containing the NVIDIA Driver .run file. Then, add the Executable permission to the NVIDIA Driver file using the chmod command.

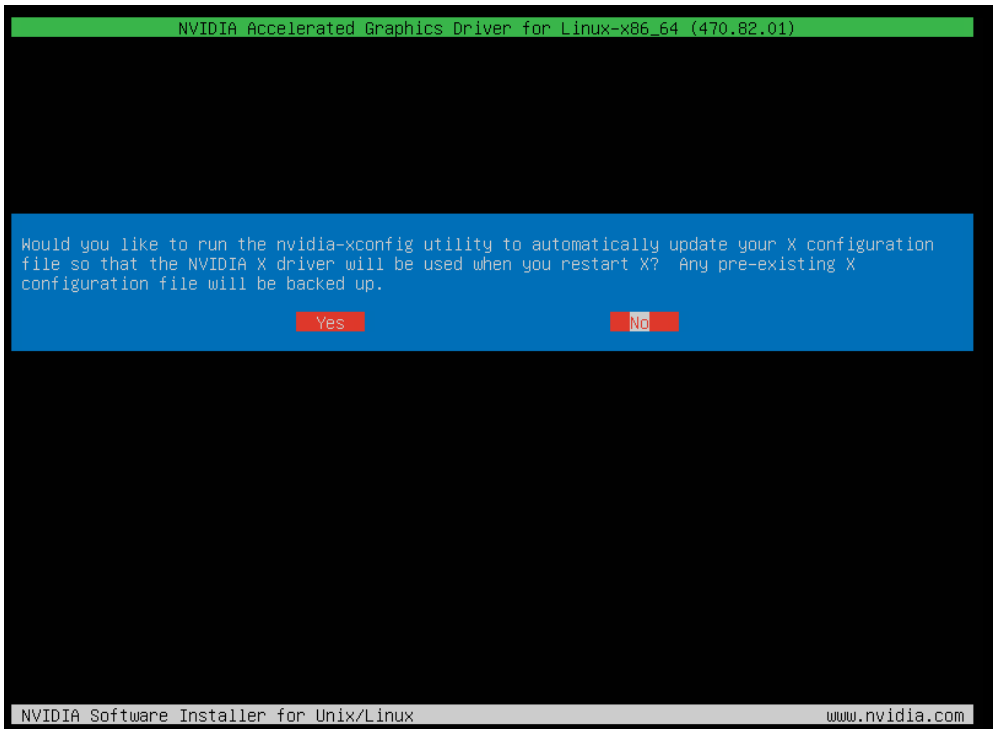
   ```
   $ cd /vgpu-driver/ NVIDIA-Linux_x86_64-470.82.01-grid.run
   $ sudo chmod +x NVIDIA-Linux_x86_64-470.82.01-grid.run
   ```

5. Run the driver installer as the root user from a console shell and accept defaults.

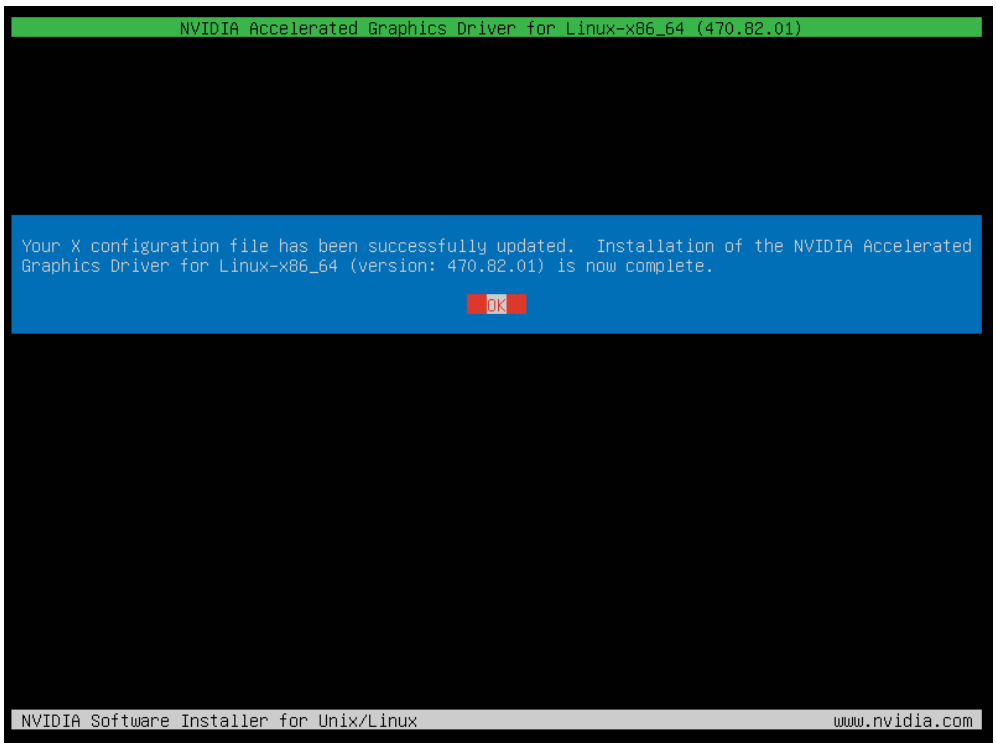
   ```
   $ sudo sh ./NVIDIA-Linux_x86_64-470.82.01-grid.run
   ```

6. When the installer prompts you, accept the option to update the X configuration file `Xorg.conf`:

   Note: Screenshots are based on the X OS.



7. Select OK to close the installer when the install is finished.



8. Reboot the system and log in.

```
$ sudo reboot
```

9. Verify that the NVIDIA vGPU driver is operational by running the `nvidia-smi command.`

```
$ nvidia-smi
```

```
Thur Dec  16 12:34:21 2021
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 470.82.01    Driver Version: 470.82.01    CUDA Version: 11.4     |
|-------------------------------+----------------------+----------------------+
| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
|                               |                      |               MIG M. |
|===============================+======================+======================|
|   0  NVIDIA A16-4Q       On   | 00000000:02:03.0 Off |                    0 |
| N/A   N/A    P8    N/A /  N/A |     448MiB / 3932MiB |     0%      Default |
|                                                                      N/A |
+-------------------------------+----------------------+----------------------+

+-----------------------------------------------------------------------------+
| Processes:                                                                  |
|  GPU   GI   CI  PID   Type   Process name                  GPU Memory       |
|        ID   ID                                             Usage            |
|=============================================================================|
|  No running processes found                                                 |
+-----------------------------------------------------------------------------+
```

8.12 Licensing NVIDIA vGPU Software (Update 13.1)

To use an NVIDIA vGPU software licensed product, each client system to which a physical or virtual GPU is assigned must obtain a license from the NVIDIA License System. A client system can be a VM configured with an NVIDIA vGPU, a VM configured for GPU pass through or a physical host to which a physical GPU is assigned in a bare-metal deployment.

Client Configuration Token

The client config token is a file that must be copied to the licensed client's default location when generated. The client system requests a license from the service instance using the Client Configuration token. Information within the client configuration token identifies the service instance, license server, and fulfillment conditions used to serve a license in response to the licensed client request.

- Generating a Client Configuration Token
- Configuring a Licensed Client on Windows
- Configuring a Licensed Client on Linux

8.13 Finalizing the Installation

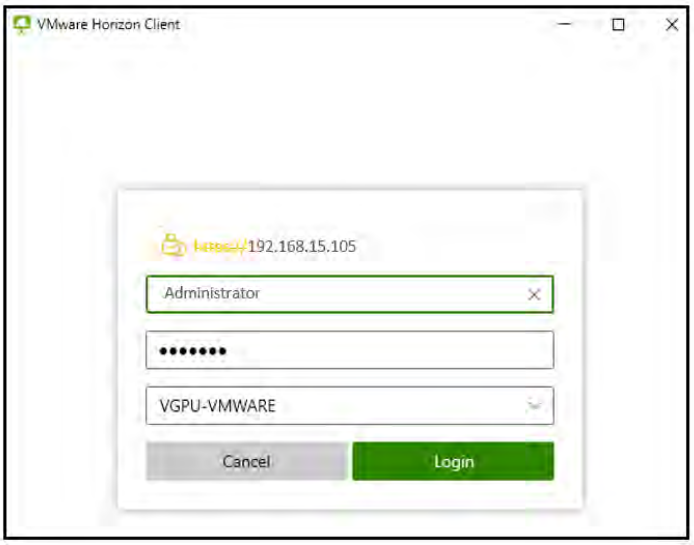
The final phase of the NVIDIA vGPU configuration uses the Horizon Client to directly connect to the VM with the View Direct-Connect Agent (VADC) and verify the VMs settings.

To finalize the installation:

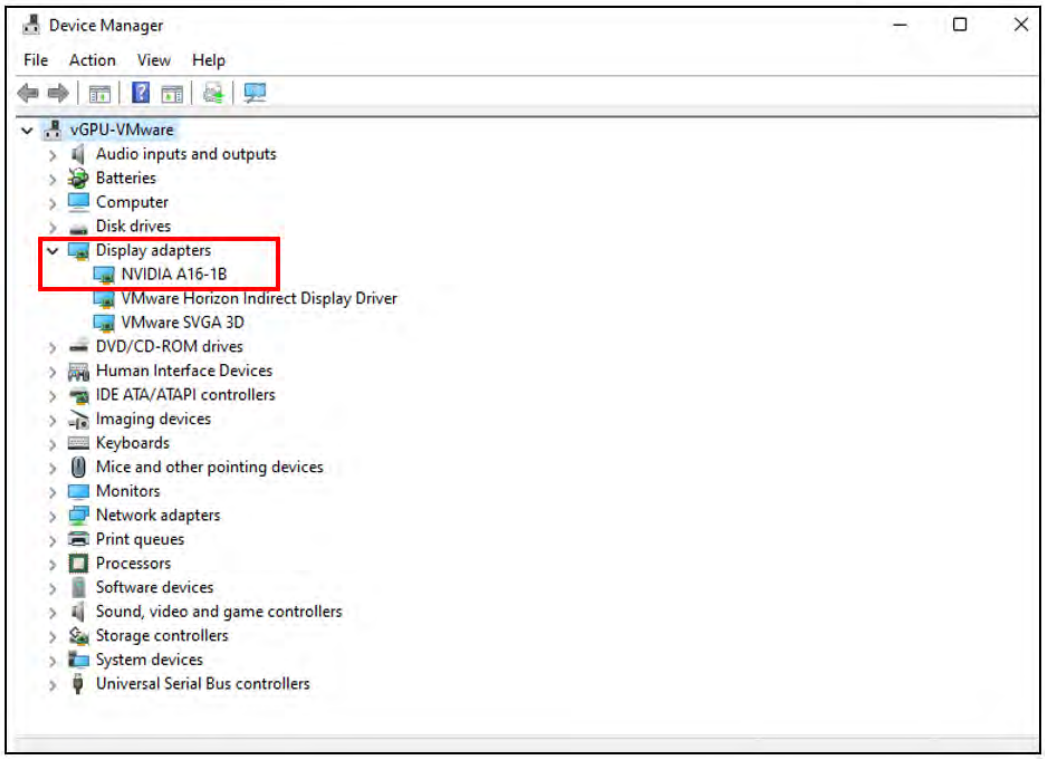
1. Start the VMware Horizon Client. Click **Add Server** to register a new virtual machine connection.

2. Enter the virtual machine's IP address in the Connection Server field, click Continue. (You are connecting directly to a desktop, so do not enter the Horizon Connection Broker address at this time.)

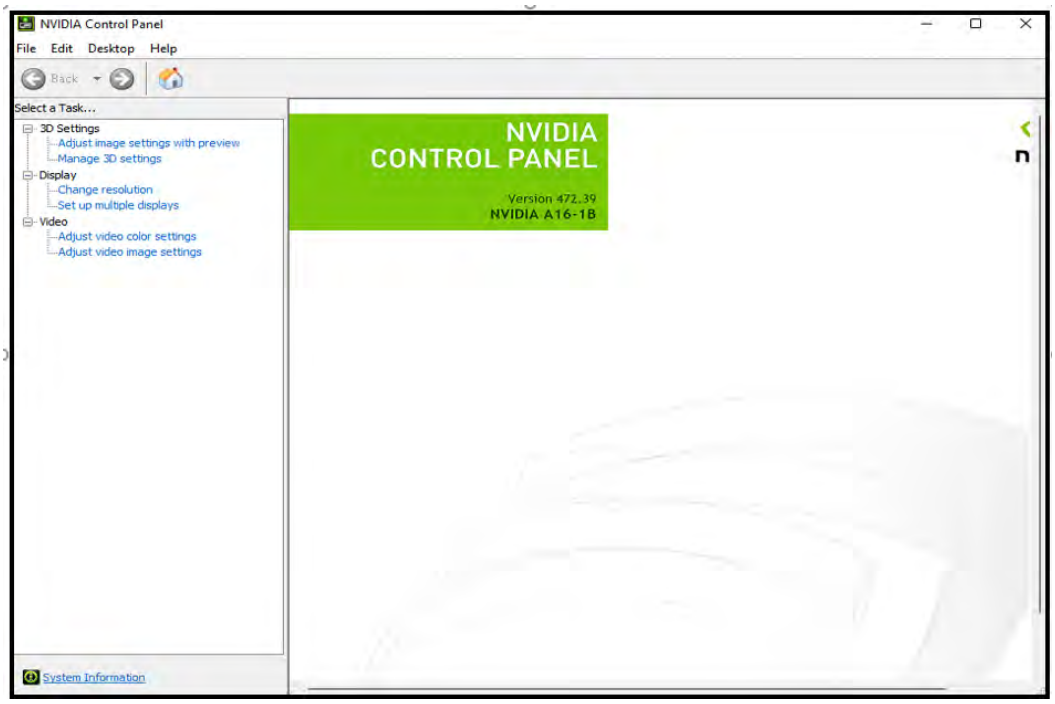
3. Enter the local username and password (or the domain user and password if the virtual machine is a domain member), then click **Login**.



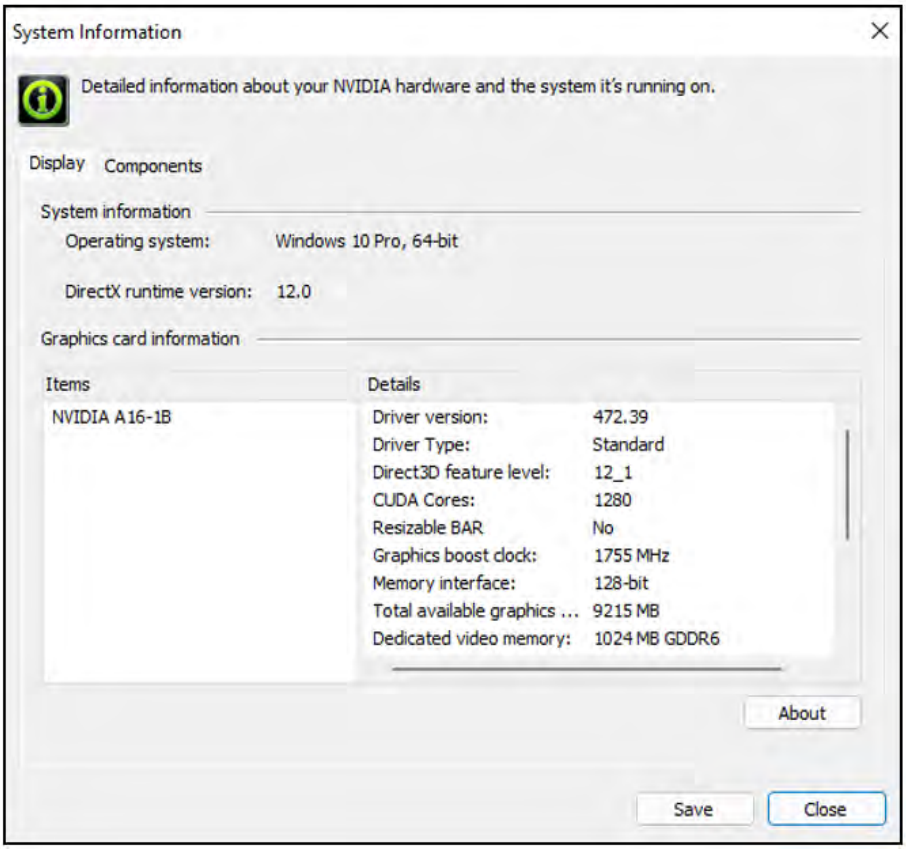
4. Open the **Device Manager** on the virtual machine and expand the **Display Adapters**. Confirm that the display adapter is now the NVIDIA vGPU.



5. Right-click the desktop and select NVIDIA Control Panel. Click the System Information link in the bottom left corner of the window



6. Confirm that the DirectX and graphics card driver versions are correct in the System Information window, then click Close.



7. If you plan to use this VM as a gold master image, release the DHCP address before logging out and shutting down. Open a command prompt and enter:

```
C...> ipconfig /release
```

8. At this point, you lose connectivity to the VM. Use the vSphere Web Client to shut down the guest OS.
9. The vGPU enabled virtual machine is ready to be the image for the desktop pools. The next step is to build and deploy a Horizon vGPU Desktop Pool.

Chapter 9. Creating a Horizon vGPU Pool

VMware Horizon uses desktop pools for centralized desktop management and distribution. In Horizon, you create a pool of virtual machines and select settings that give all the machines in the pool a standard desktop definition

This chapter describes the following:

- Creating a template from an existing virtual machine
- Creating a Customization Specification
- Provisioning a single vGPU-enabled virtual machine from a template
- Creating Full and Linked Clones Horizon pools
- Enabling User Access to pools

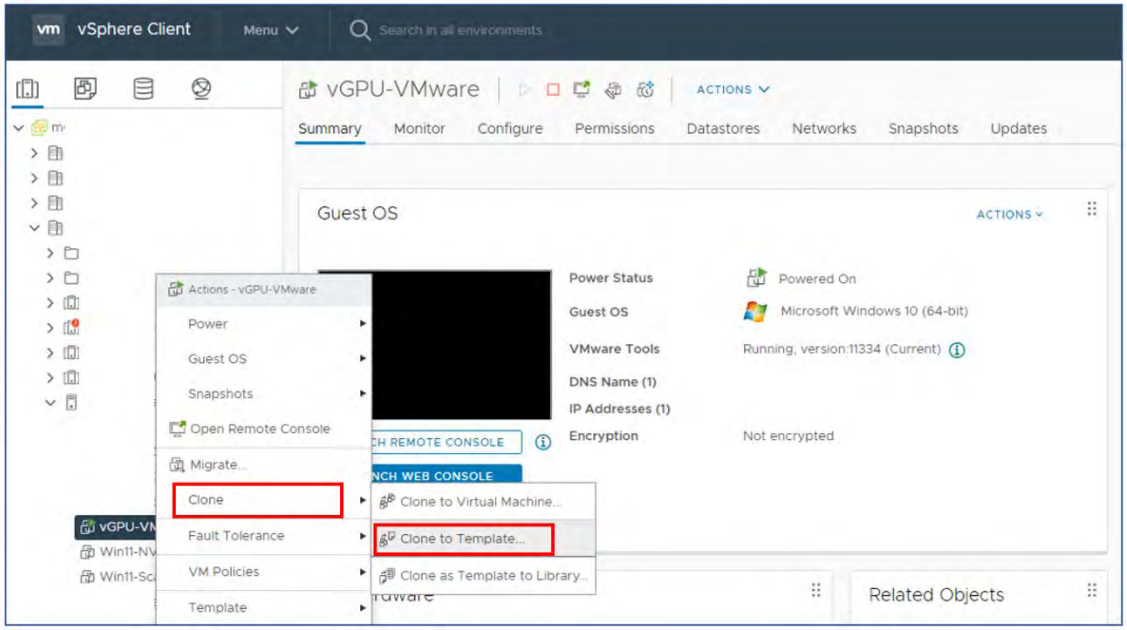
To create a pool, you must first convert an existing virtual machine into a template, which you can create a single virtual machine or virtual machines on demand.

A complete demonstration of these features is beyond the scope of this document; however, converting a virtual machine to a template and then deploying virtual machines from that template is a fundamental operation that can reduce evaluation time.

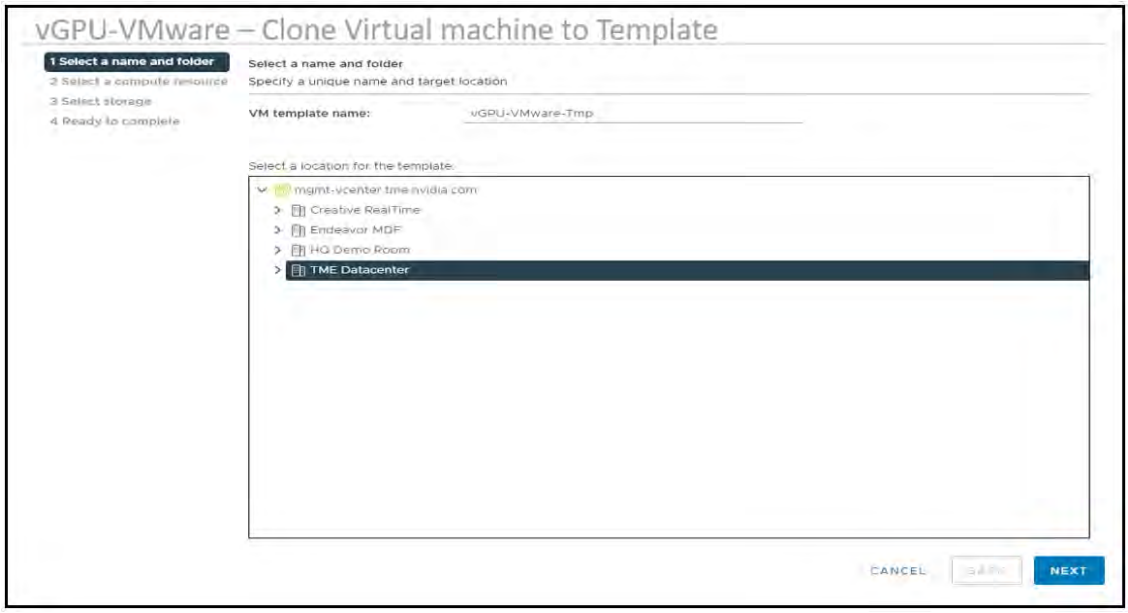
9.1 Creating a Template

To create a template from an existing virtual machine:

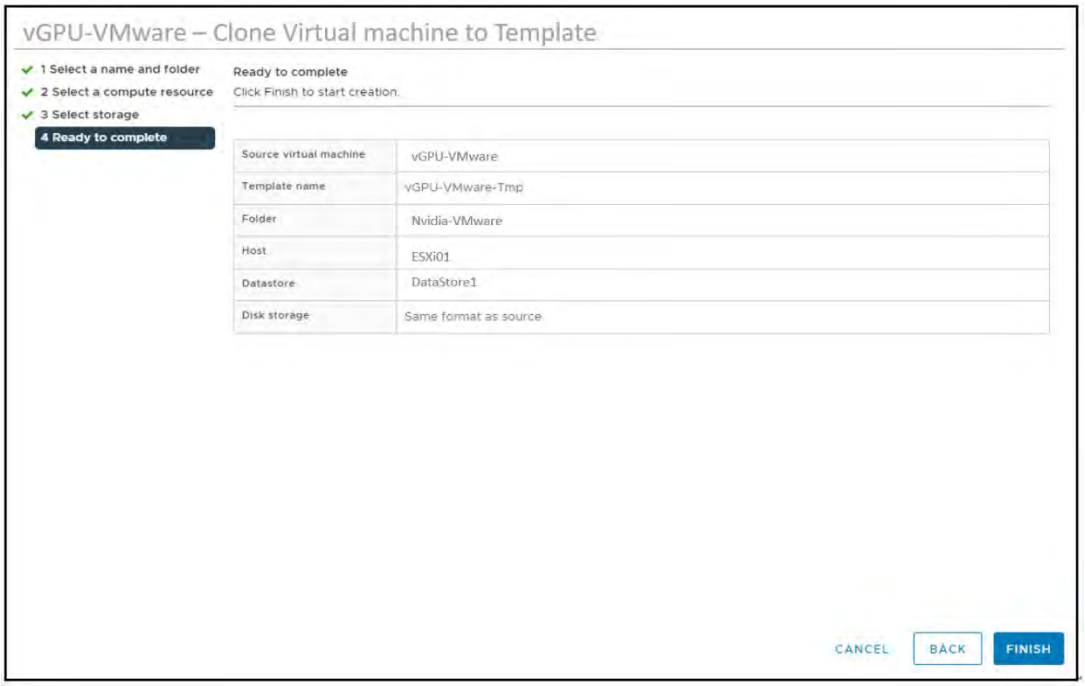
1. In vSphere Web Client, right-click the Golden Image virtual machine and select Clone - > Clone to Template.



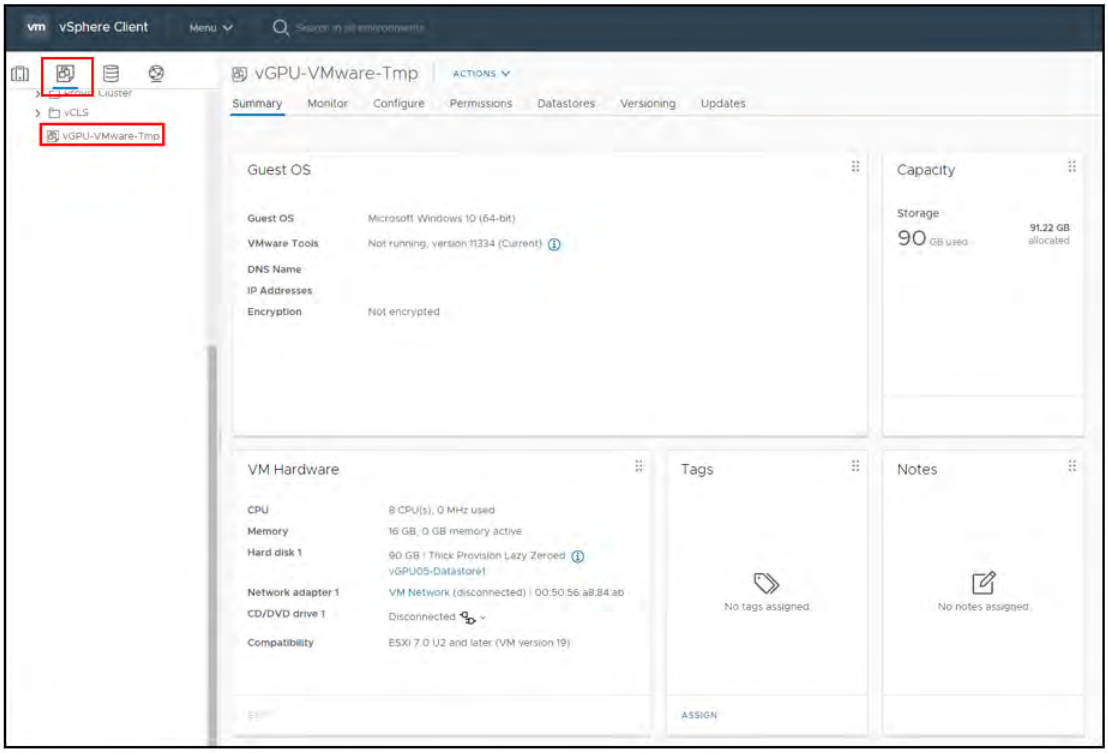
2. vSphere displays the **Clone Virtual Machine To Template** window. With the **Select a name and folder** tab selected.



2. In the **VM template name** field, enter a name for the template.
3. Select a location to create the template. Click **Next**.
4. Select a compute resource for the template. Click **Next**.
5. Select the target datastore. Click **Next**.
6. Review the Ready to Complete window settings and click **Finish** to start cloning the virtual machine to a template.



7. When the cloning process is complete, the VM template is displayed at the end of the list in the **VMs and Templates** section.

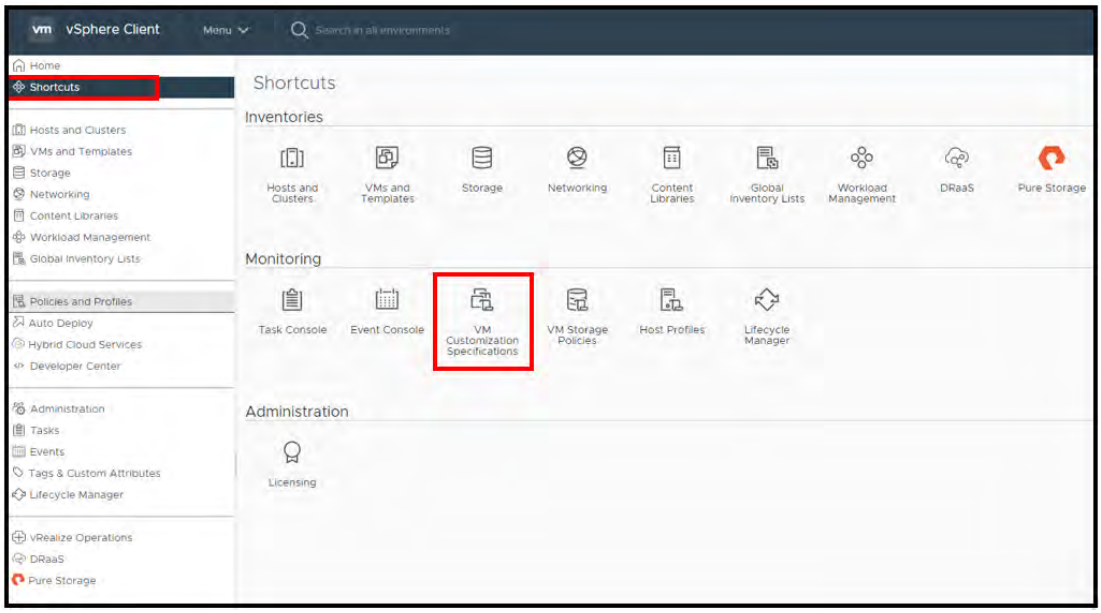


9.2 Creating a Customization Specification

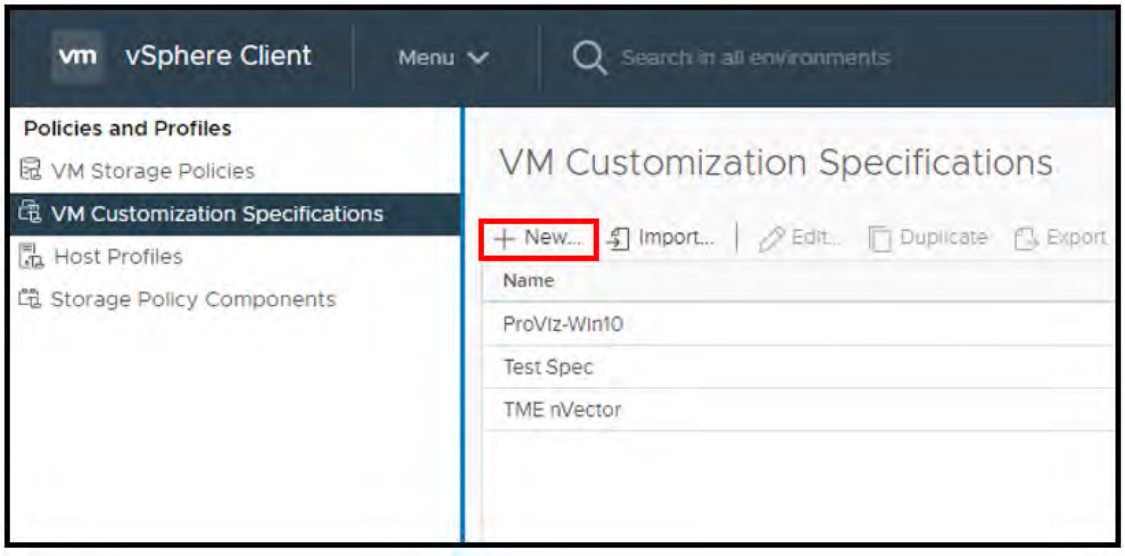
Before you can provision a virtual machine from a template, you must create a **VM Customization Specification.** This will handle actions such as joining the domain and renaming the machine.

To create a VM Customization Specification:

1. From the left-hand pane of the vSphere Client Shortcuts view, select **VM Customization Specifications** in the **Monitoring** section.



2. In the **VM Customization Specifications** window, click **New**.

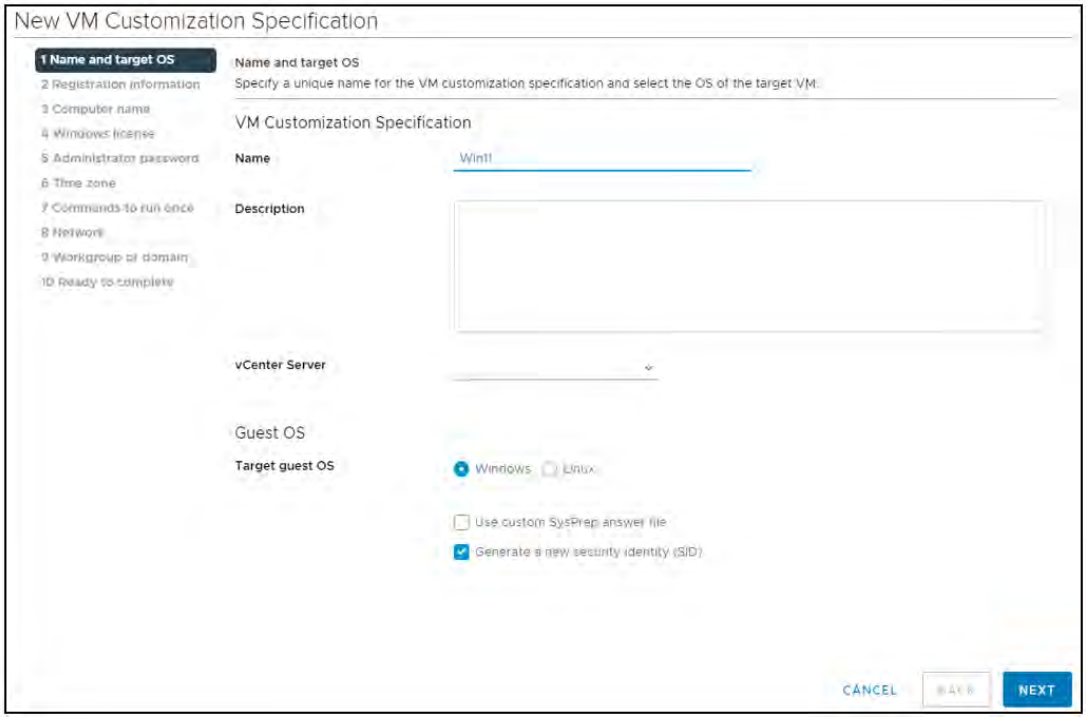


2. For Target guest OS, select Windows.

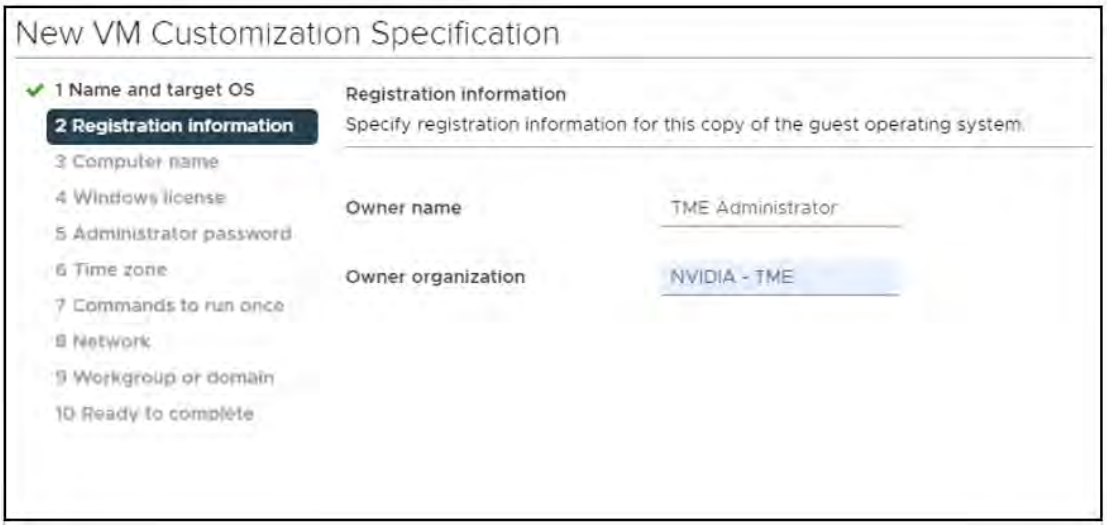
   Check the Generate New Security ID (SID) checkbox.

If your organization uses an answer file, you can load that file by checking the **Use custom SysPrep answer file** checkbox.

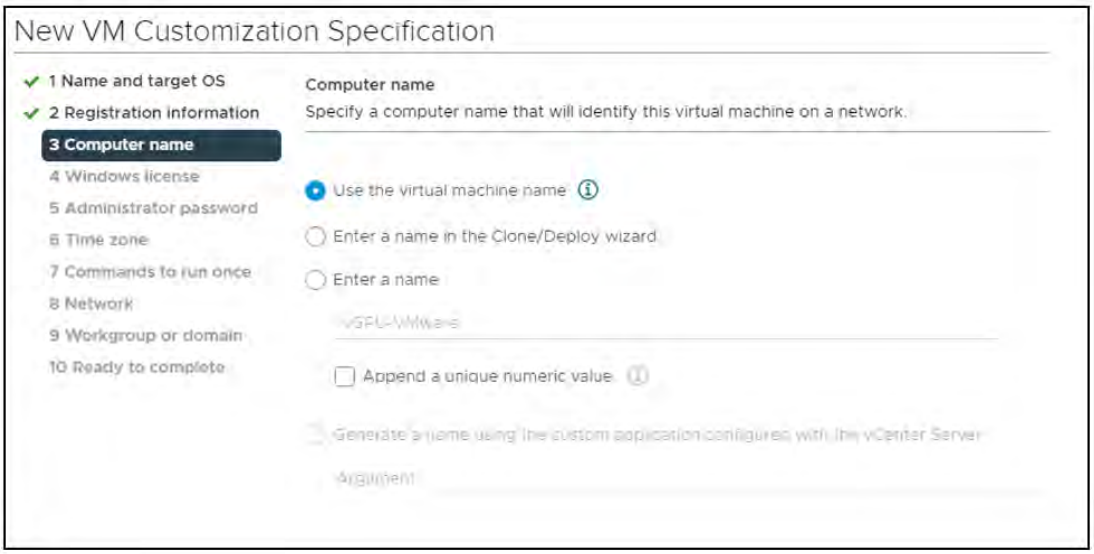
If your organization does not use this functionality, leave the checkbox cleared. Enter a name for the new specification in the Customization Specification Name field. Click **Next**.



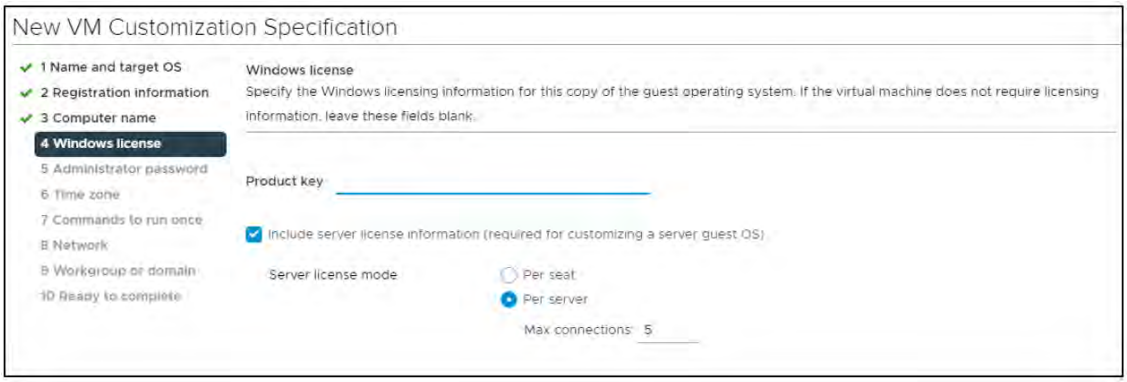
3. Enter the VM owner's name in the Name field and the owner's organization's name in the Organization field, then click **Next**.



4. Enter a computer name for the VM, then click Next.

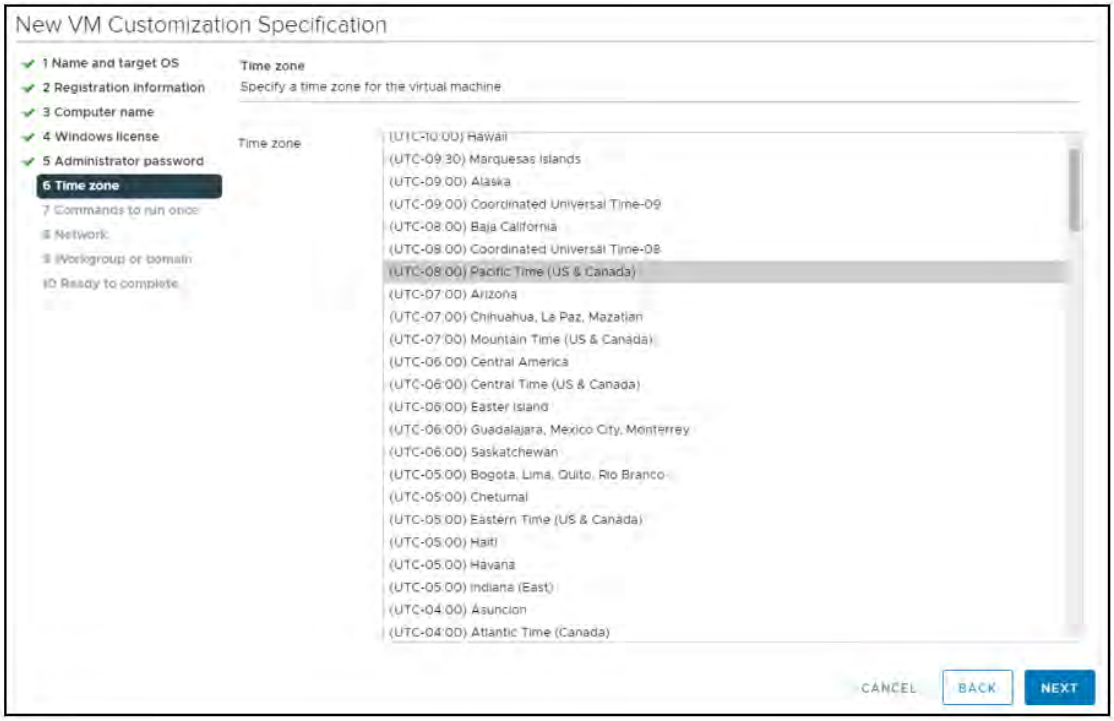


5. Enter your Windows product key information or leave this field blank. Then click Next

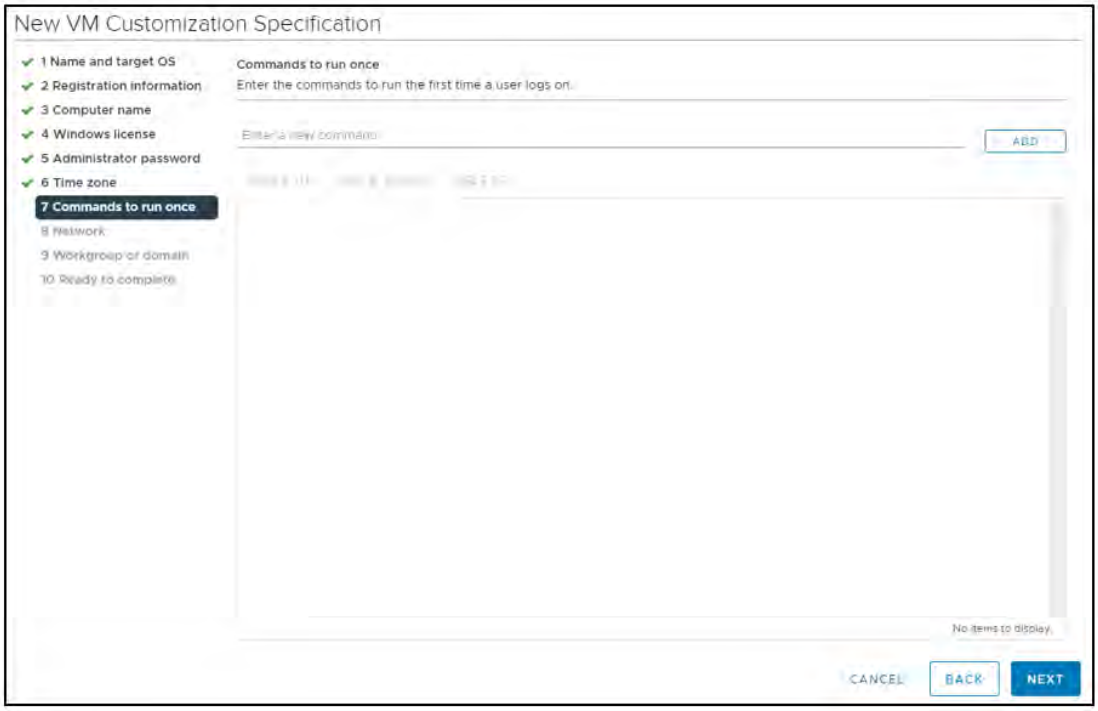


6. Enter and confirm the administrator password (it is case sensitive) in the Password and Confirm Password fields, then click Next

7. Select your time zone from the Time Zone pulldown menu, then click Next



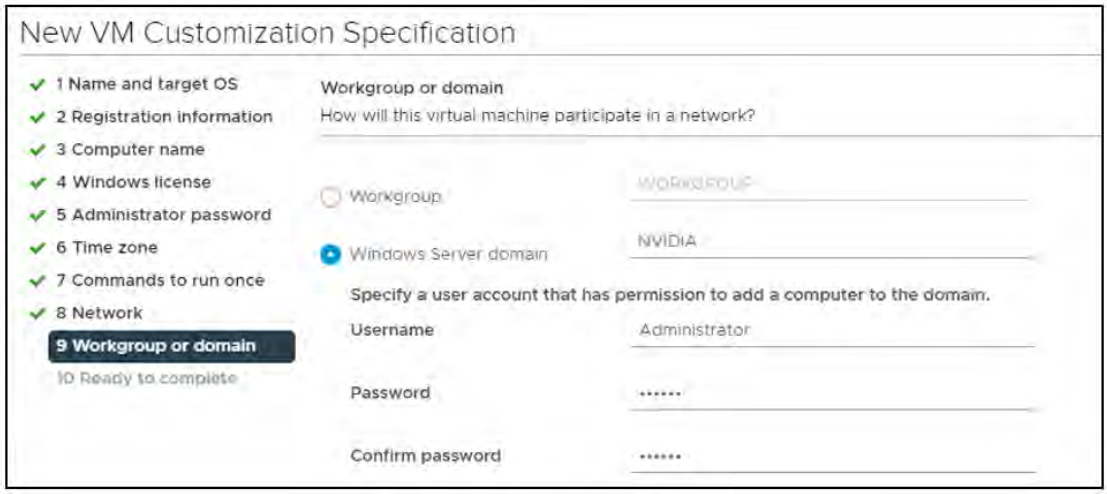
8. Enter any required one-time commands in this tab, then click Next



9. Click the appropriate radio button to either:
   - Use standard network settings: Automatically selects network settings.
   - Use manual network settings: Enter the network description, IPv4 address, and/or IPv6 address in the respective fields.

Click Next.

10. Select the appropriate radio button to make the VM a member of either a workgroup or a domain. Enter the VM's workgroup or domain information in the appropriate fields. Then click. Next



11. Review your settings. If any are incorrect, click Back to correct them. When all of the settings are correct, click Finish to view the new customization specification from the template.

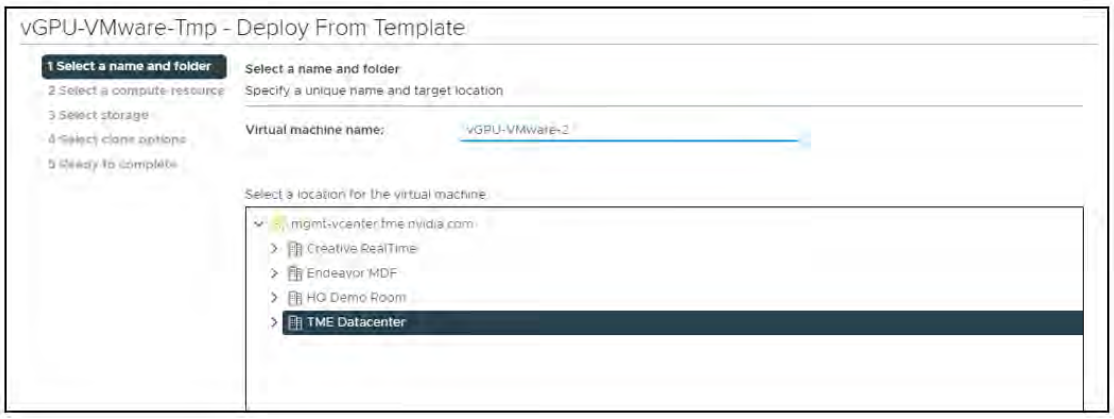
You do not need to create a virtual machine guest customization specification every time you clone a virtual machine from a template.

9.3 Provisioning a Single Virtual Machine

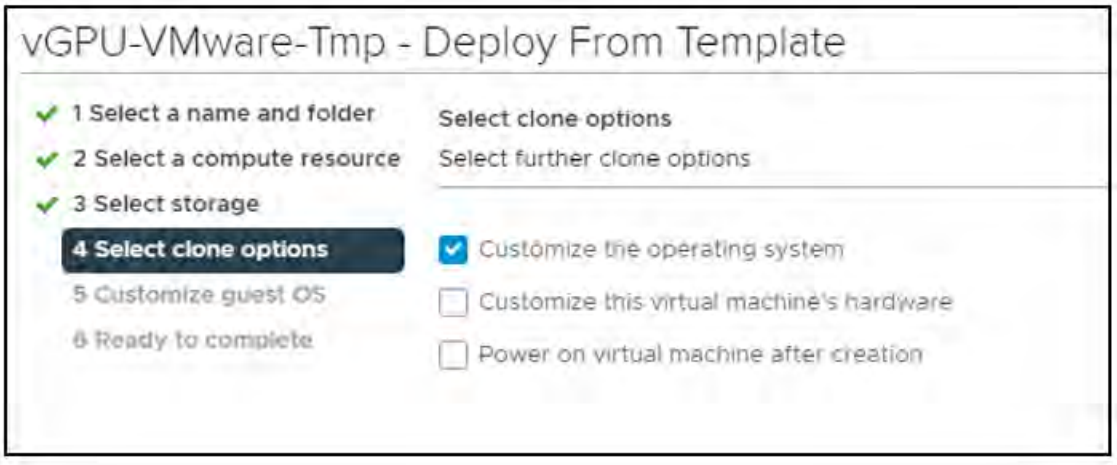
To provision a single vGPU-enabled virtual machine from a template:

1. From vSphere Client, right-click the template and select **New VM from this Template**. vSphere displays the **Deploy From Template** window with the Select a name and folder tab selected.

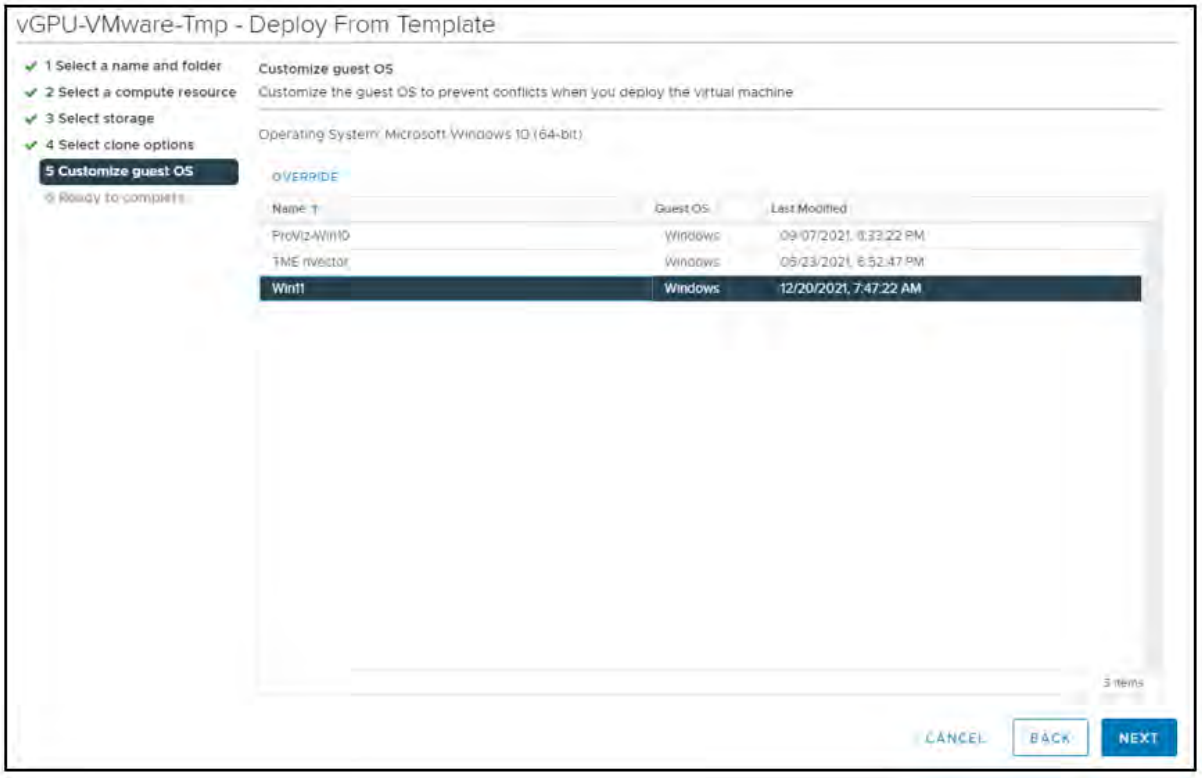
   Enter a name for the new VM in the Virtual machine name field and select a location. Click **Next**.



2. Select the **Compute Resource**, click **Next**.
3. Select the **target storage type and datastore**, click **Next**.
4. Check the **Customize the operating system** checkbox In the **Select clone options** window. Click N**ext**.

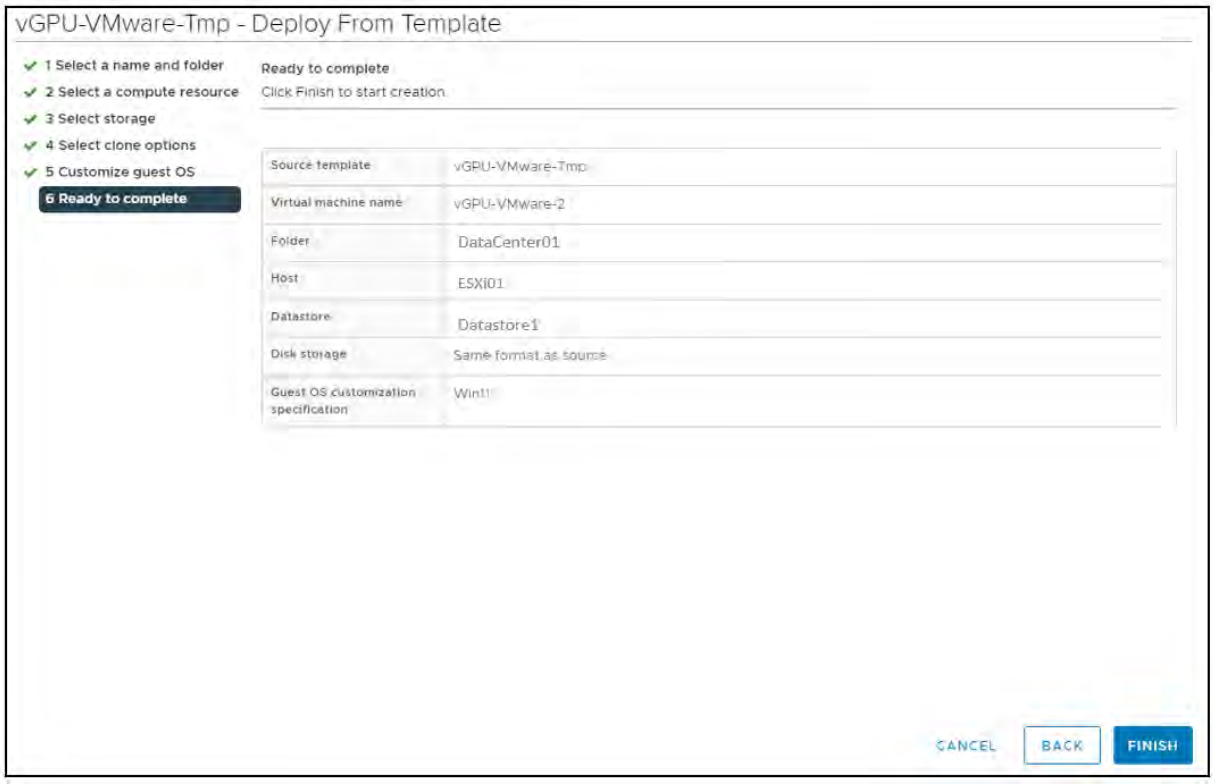


5. Select the Guest OS used to create the VM. Click **Next.**



6. vSphere displays the **Ready to complete** tab of the **Deploy From Template** window, displaying the "Win11" Guest OS Customization specification properties.

   Review the settings. If any are incorrect, click the Back button to make corrections. Click **Finish** to begin creating the virtual machine from the template.



7. Once the VM is created, it becomes visible in the data center on its cloned host.

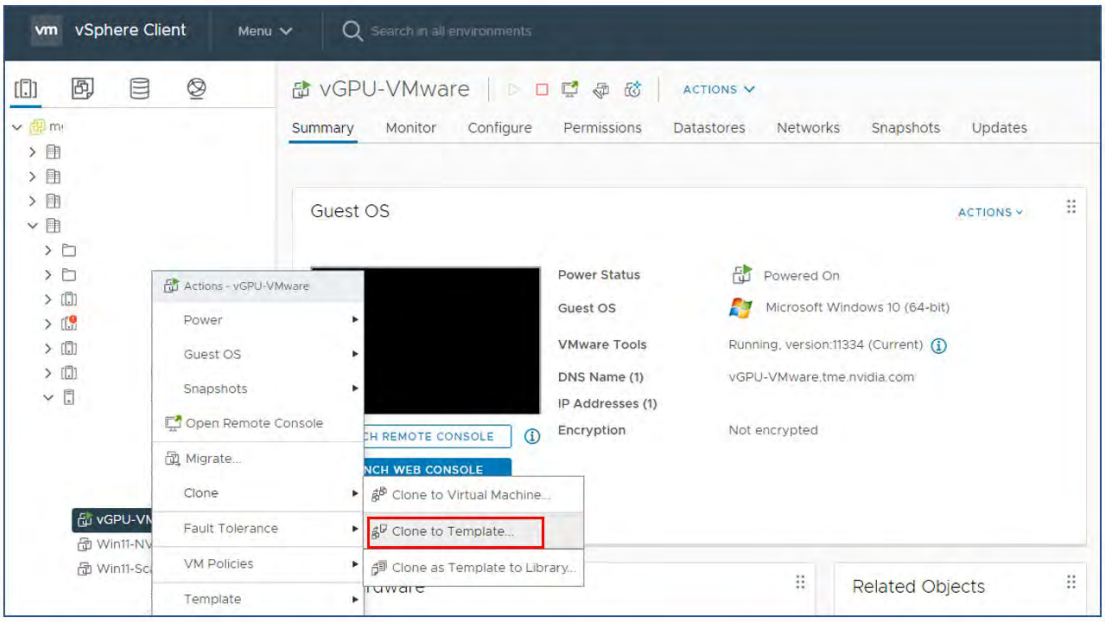
   We will add the newly cloned VM to a Horizon Desktop Pool, then authorize user/groups to use it. These steps are described later in Section 9.6 - Enabling User access to Desktop Pools.

Horizon can leverage templates to create virtual machines automatically and on-demand to save time and resources. See the Horizon 8 Documentation for instructions on using this functionality.

The following sections describe how to create a desktop pool and grant entitlements to users and groups who use the pool. In Horizon 8 2111, you can create desktop pools from a gold master image with Full Clones or Linked Clones. Both use cases are explained in the following sections.

9.4 Creating Full Clone Desktop Pools

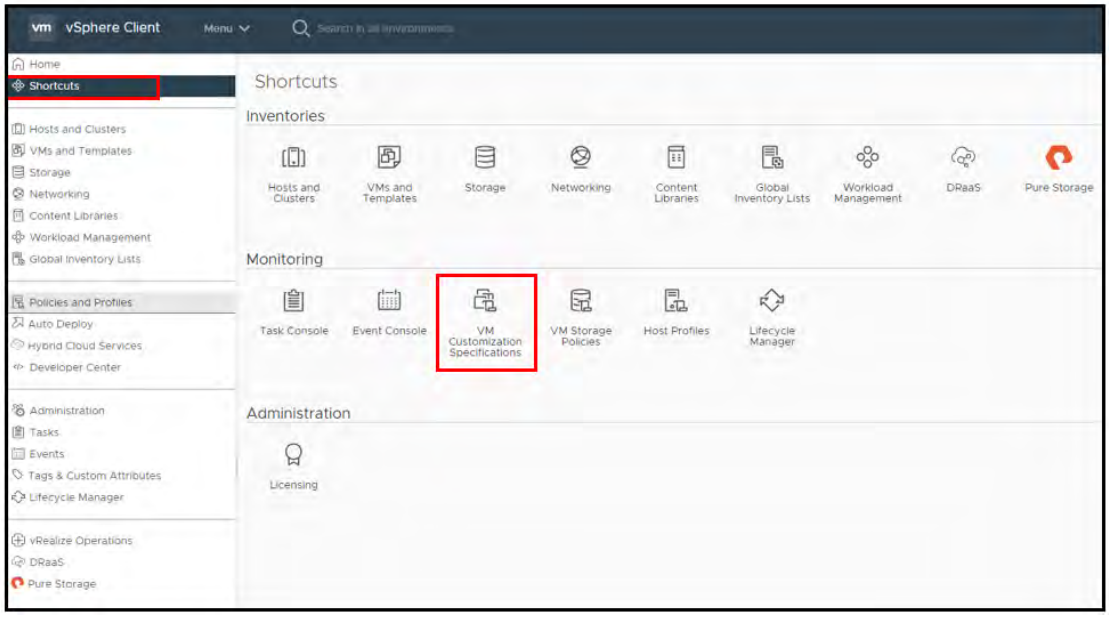
1. To create full clone desktop pools, you must have a template of the golden master image. Find the reference VM in the Navigator and right-click it. Select Clone > Clone to Template.



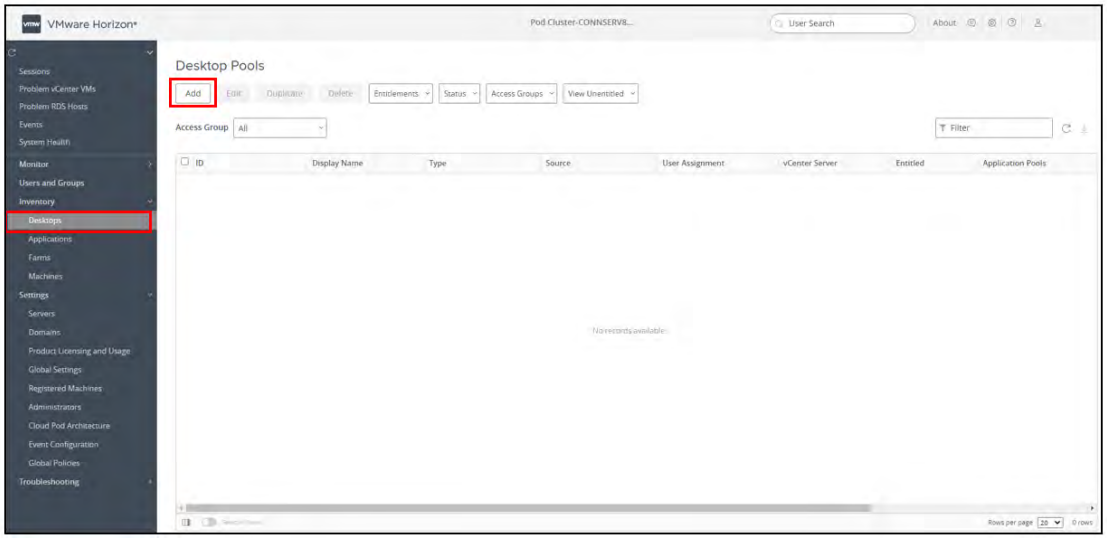
Alternatively, you can create a VM template using instructions from Section 9.1 - Creating a Template.

The Clone Virtual Machine To Template wizard walks you through the steps. At the end of the process, click **Finish** to create the template.

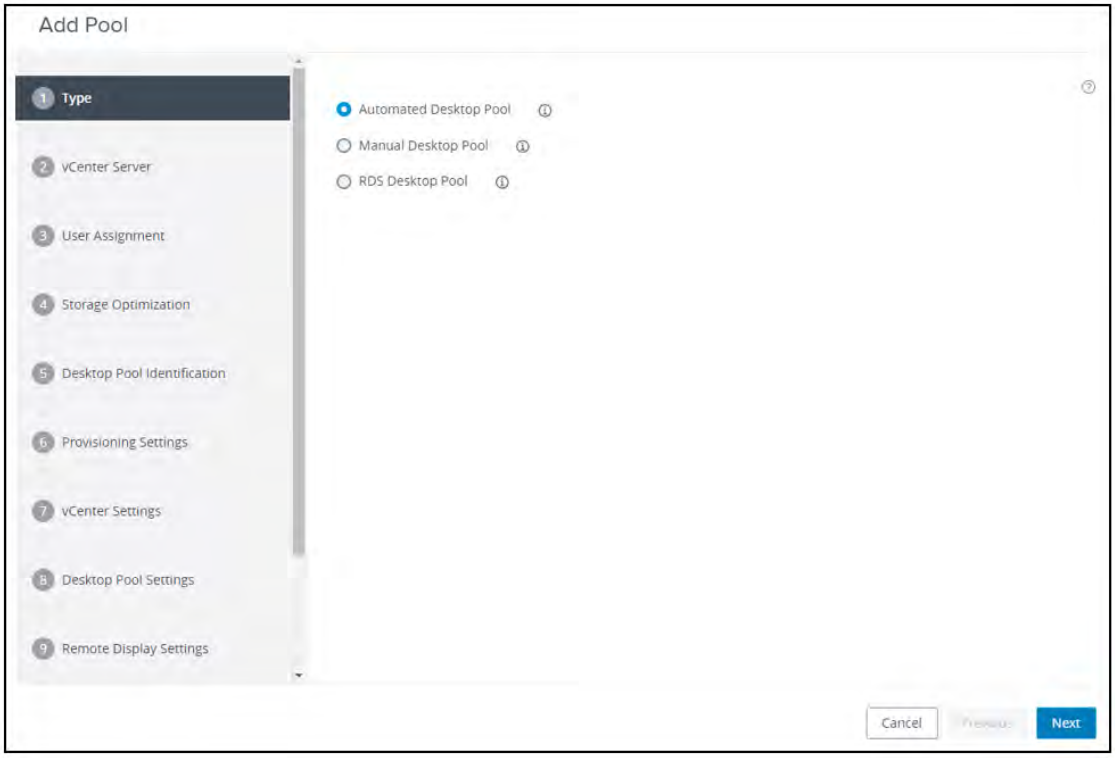
2. Create a Customization Specification file to deploy the full clone pool. The **VM Customization Specification** wizard walks you through these steps, as described in Section 9.2 - Creating a Customization Specification.



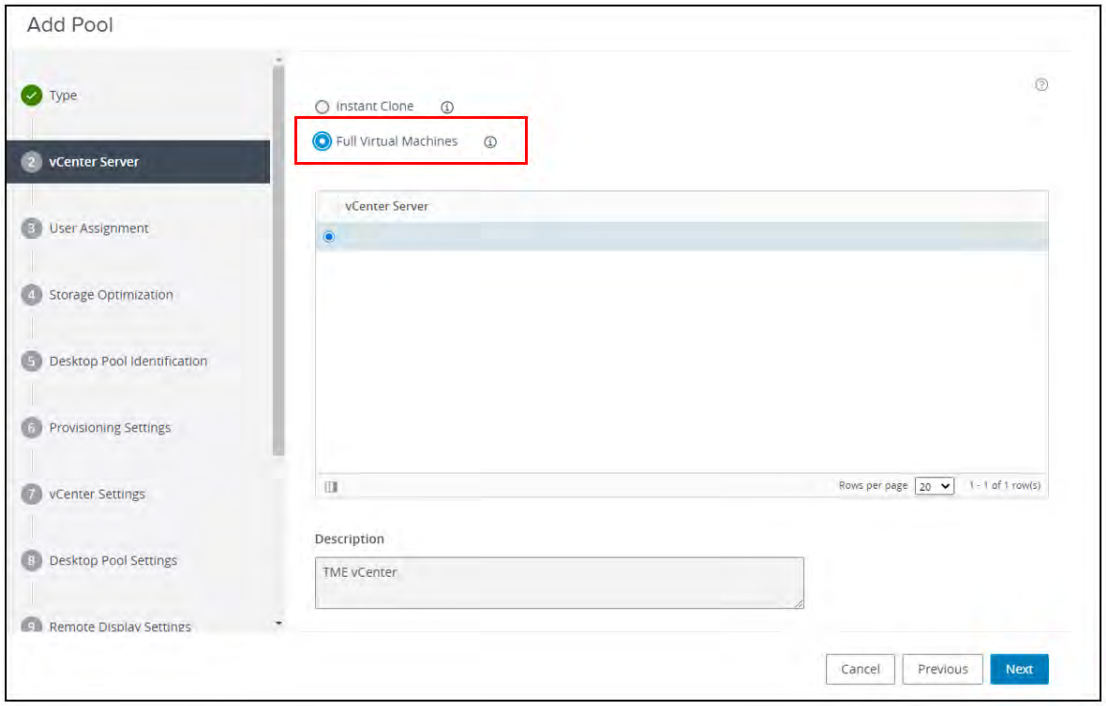
3. log in to the Horizon Administrator. Select **Desktops** in the Inventory pane, then click Add.



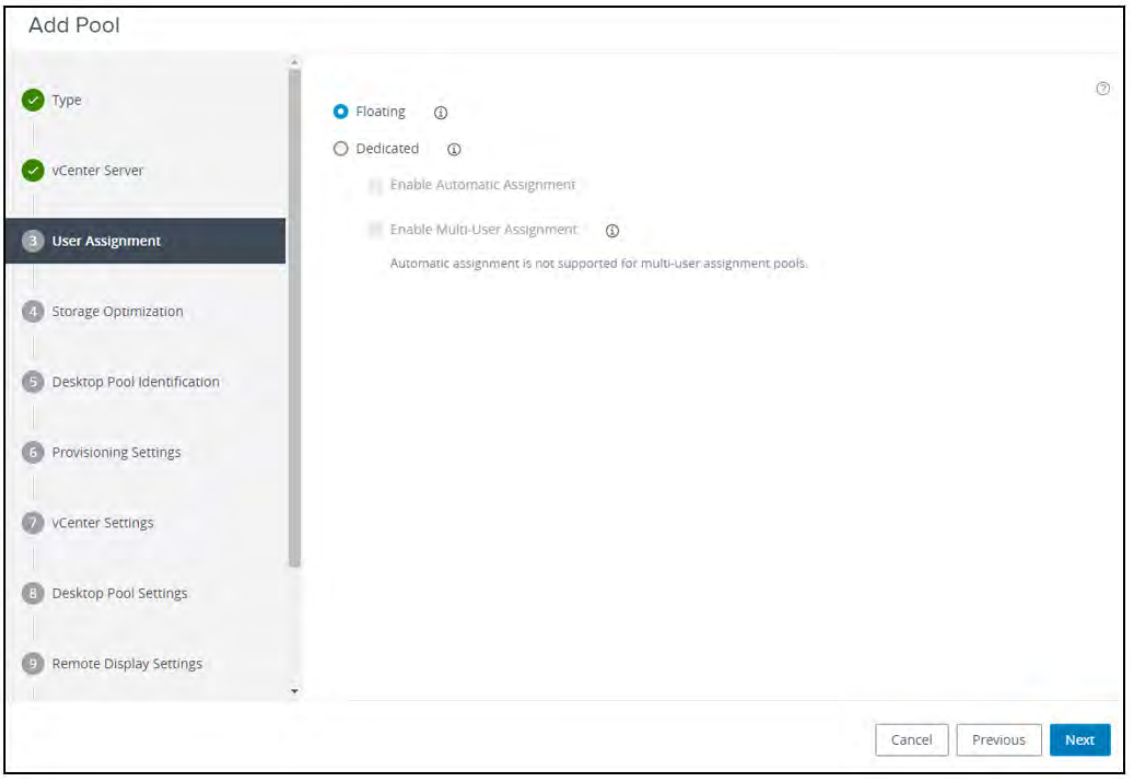
4. Select **Automated Desktop Pool**, then click Next.



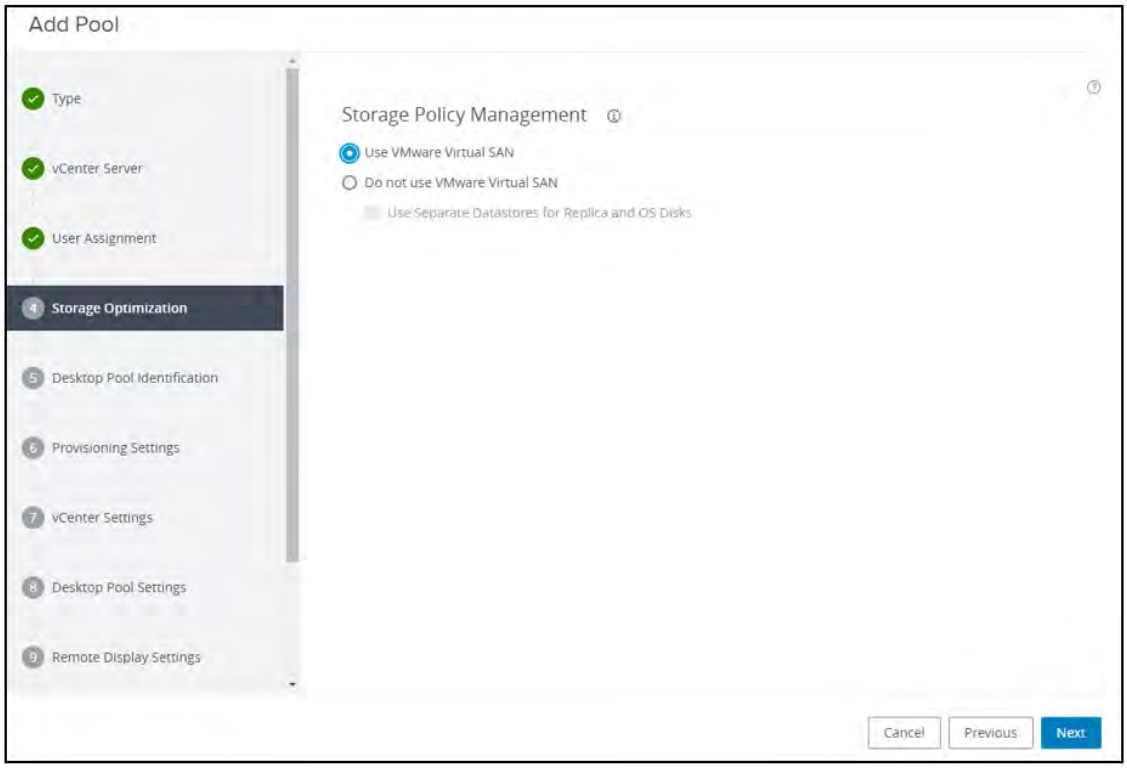
5. Select the **vCenter Server** instance. Select the **Full Virtual Machine** radio button to create a full clone desktop pool. Click **Next**.



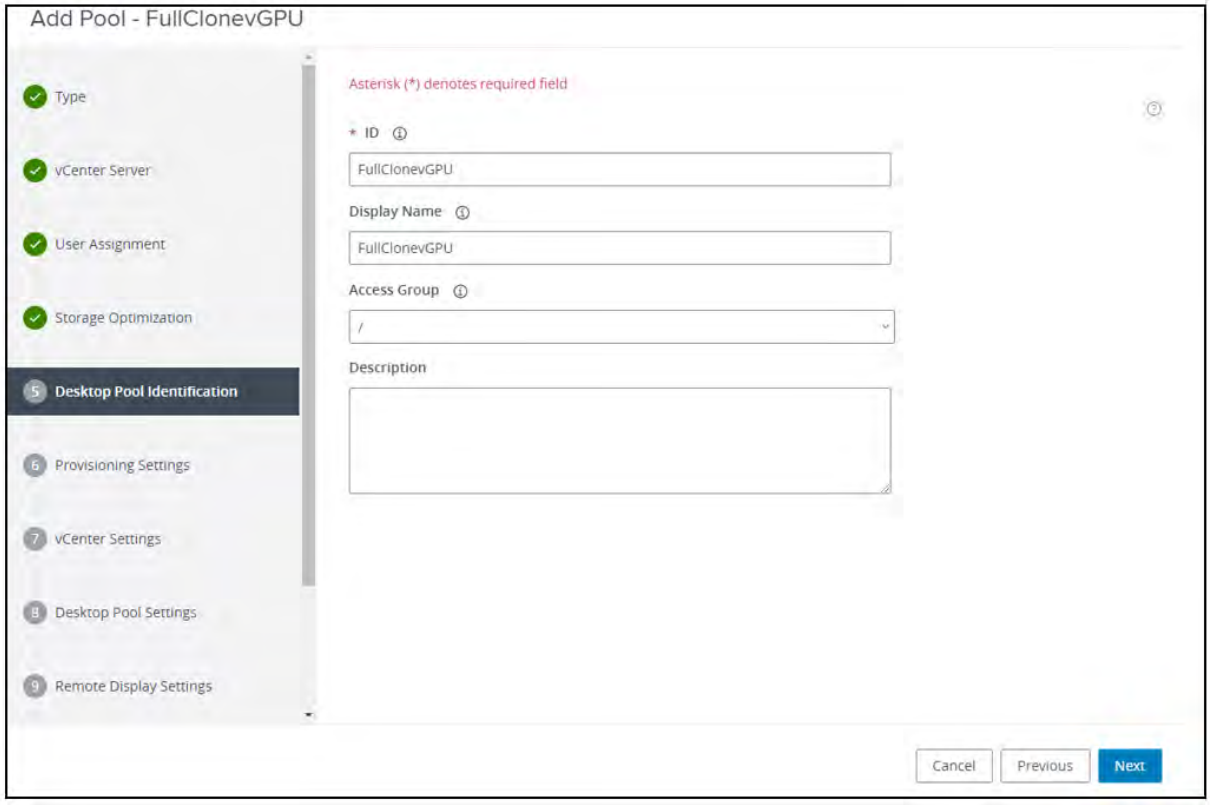
6. Select Floating as the User Assignment type. Click **Next**.



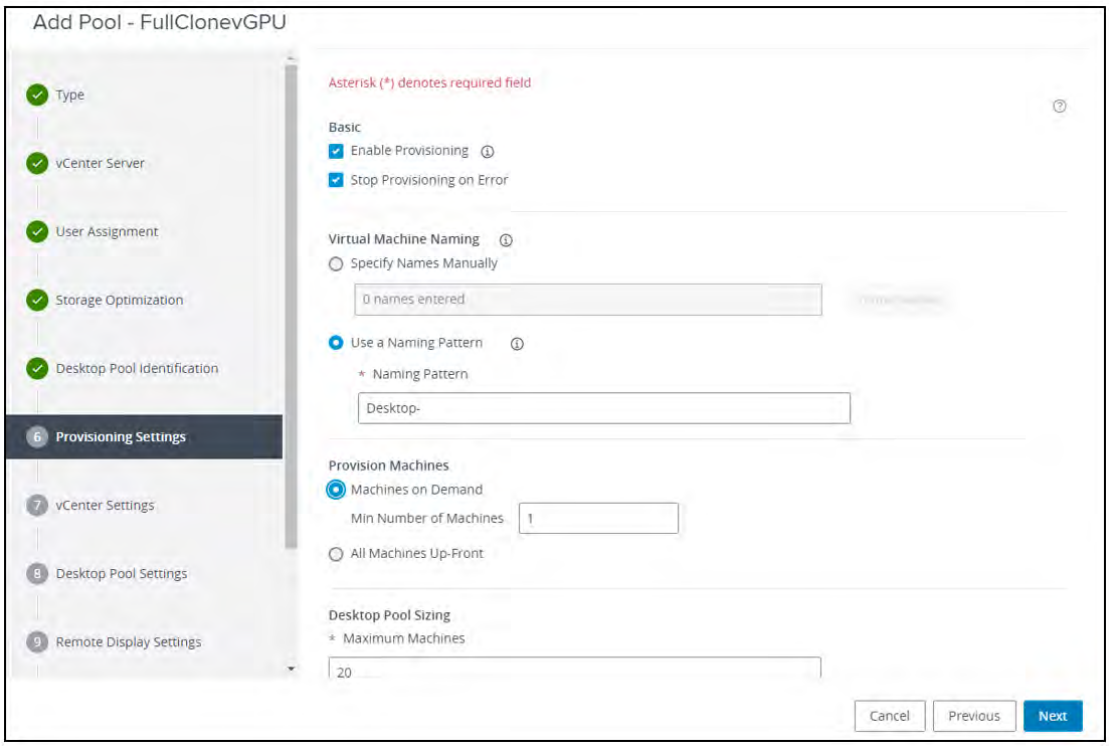
7. Select the **Use VMware virtual SAN** Radio button for the **Storage Optimization**. Click Next.



8. Enter a unique ID and Display Name for the desktop pool. Click Next.

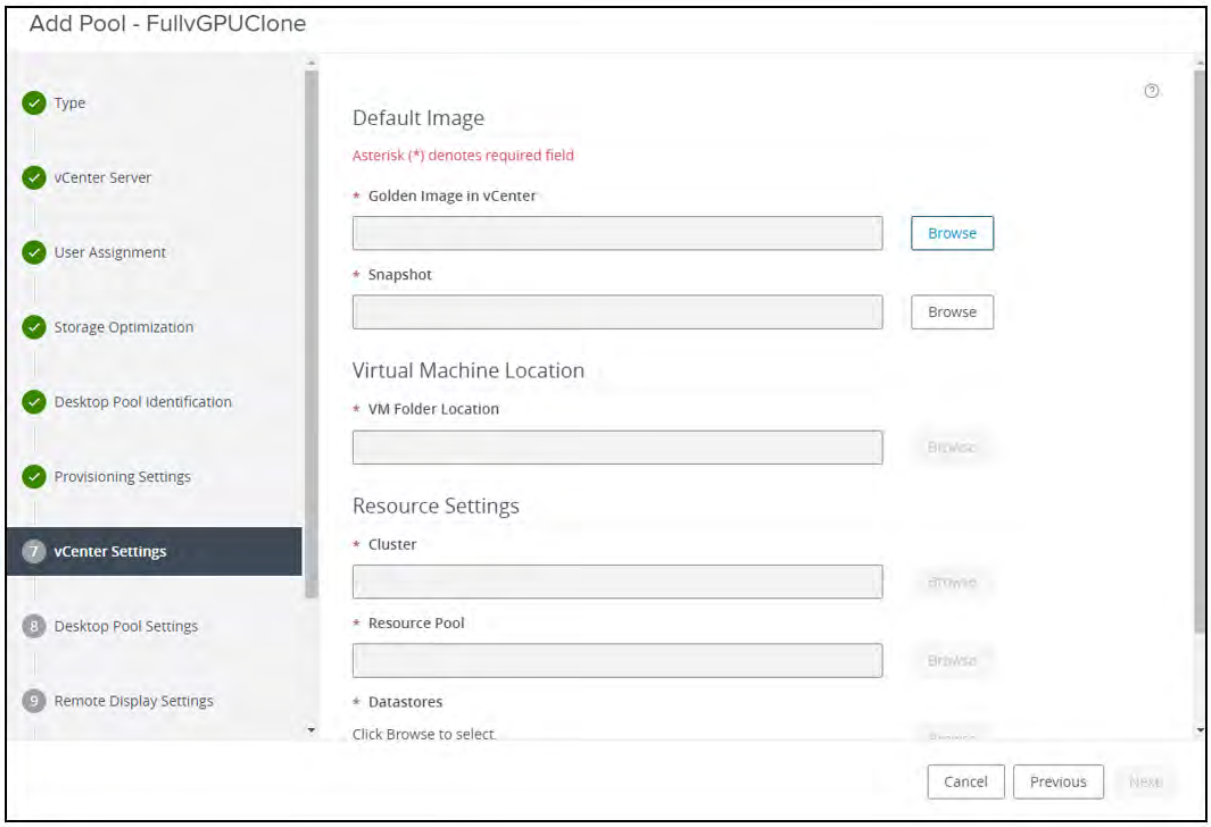


9. Enter the Provisioning Settings for your environment. Click Next.



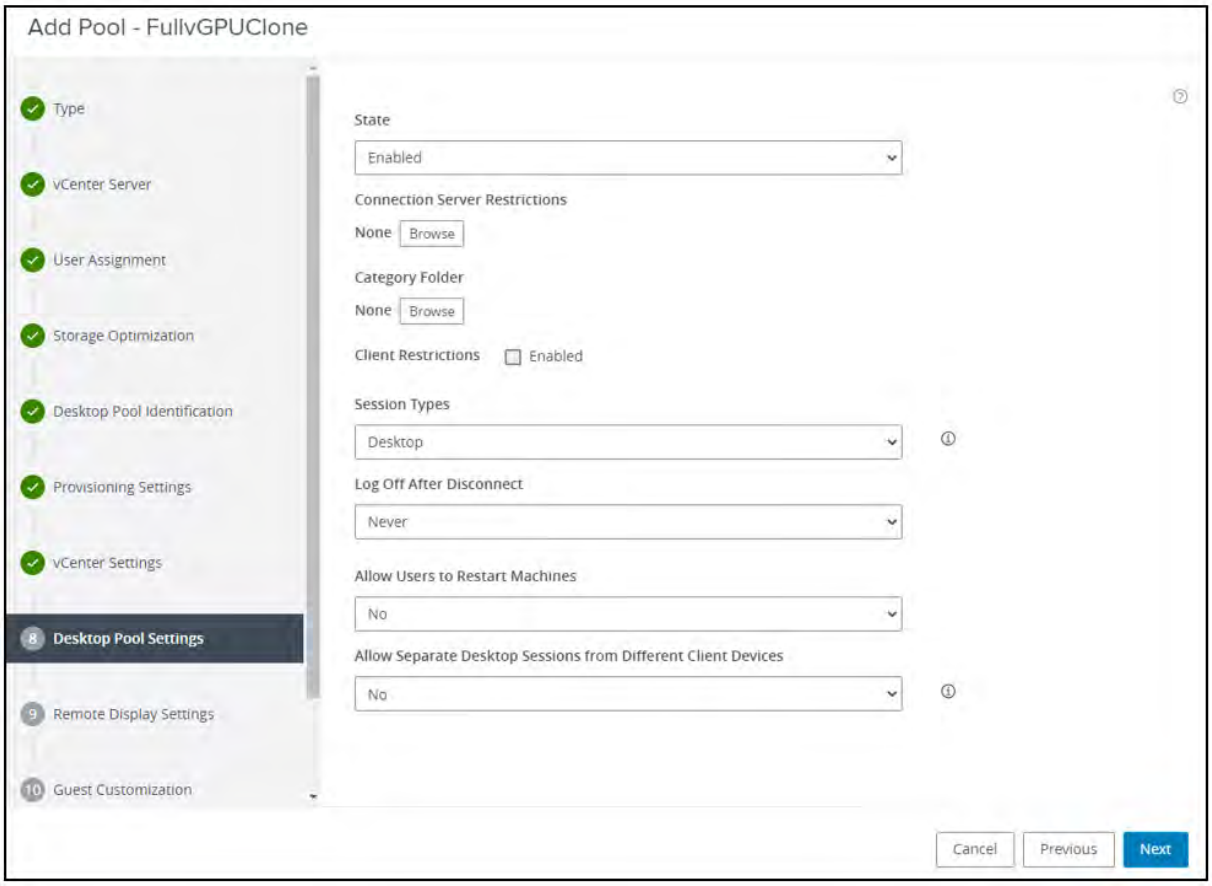
10. In the vCenter Settings tab. Fill in the details for your Horizon Pool.

Vmware suggests using the Worksheet to Create an Automated Full-Clone Desktop Pool to prepare your configuration options before creating the desktop pool.

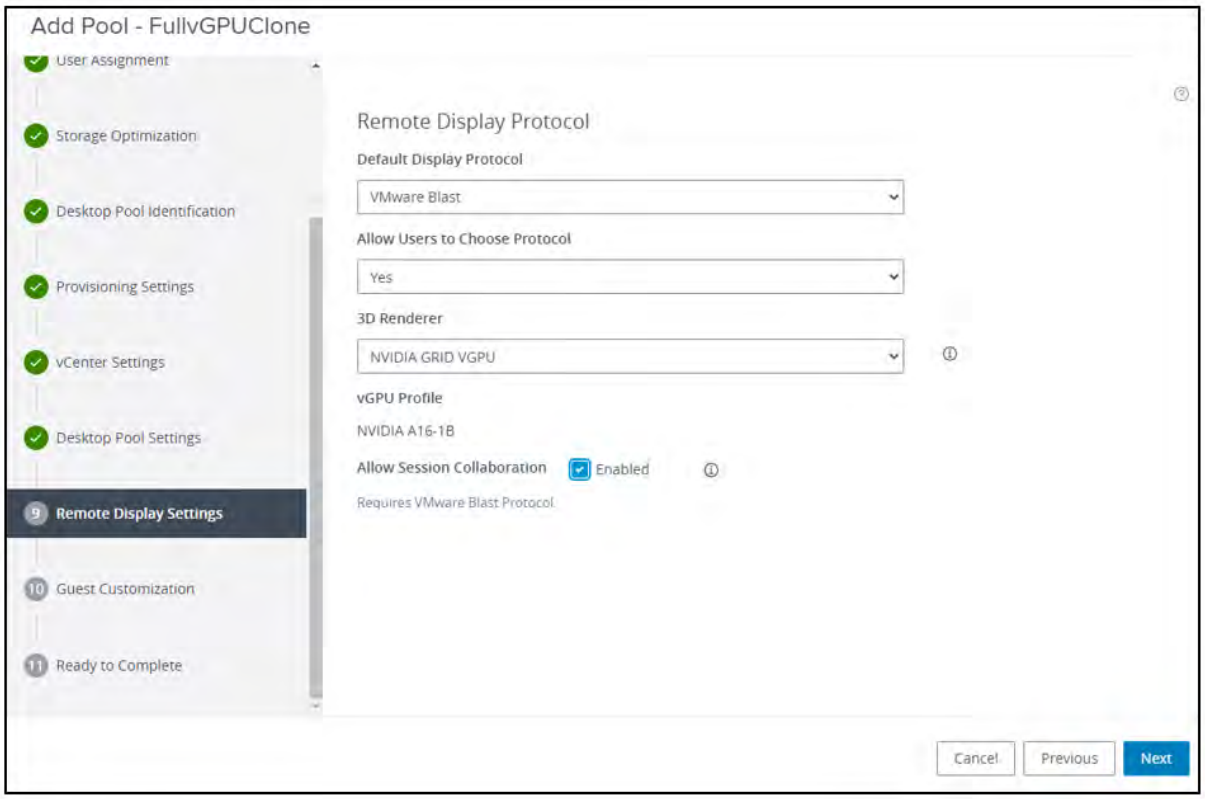


11. In the Desktop Pool Settings tab. Fill in the details for your Horizon Desktop Pool.

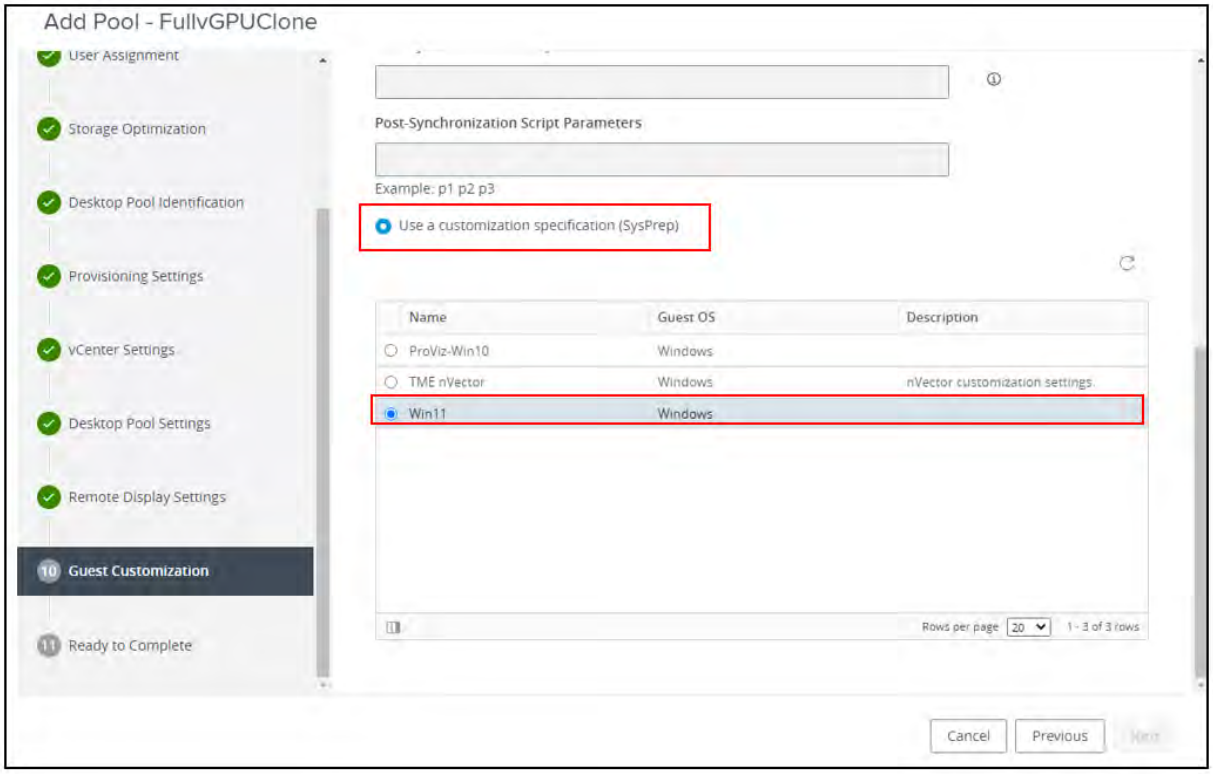
    Vmware suggests using the Worksheet to Create an Automated Full-Clone Desktop Pool to prepare your configuration options before creating the desktop pool.



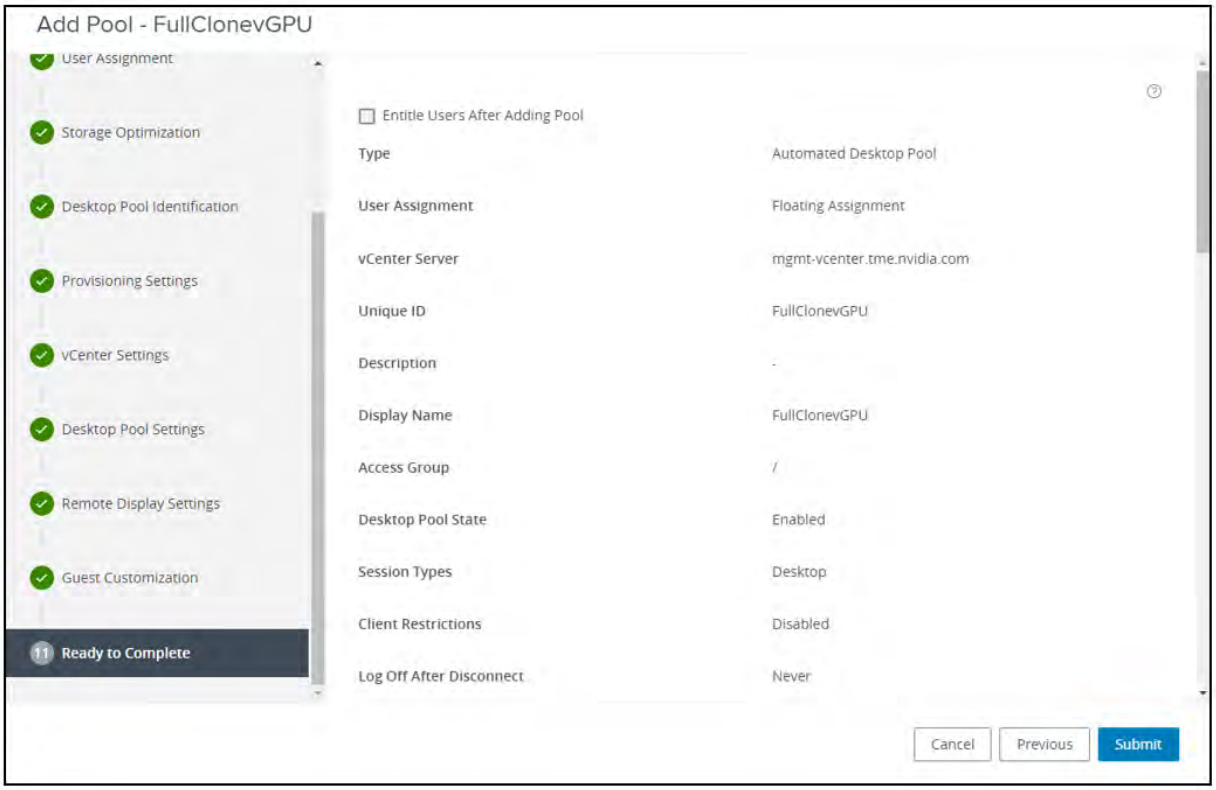
12. In the Remote Display Settings tab. Fill in the details for your Horizon Desktop Pool.



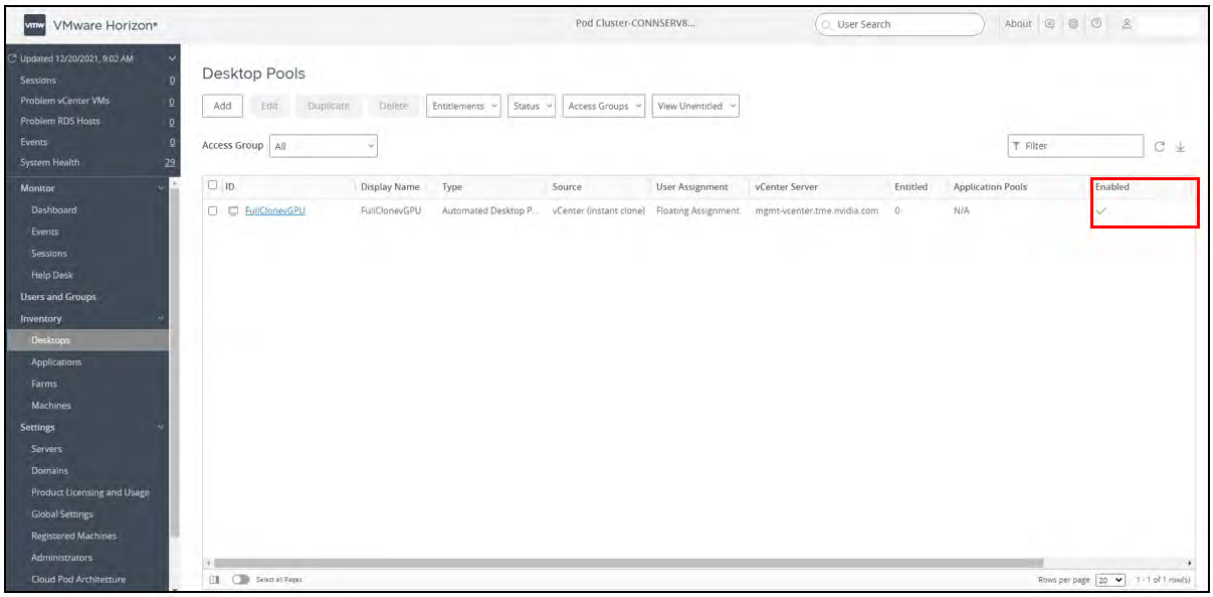
13. Under the **Guest Customization** tab. Scroll towards the bottom and select the **Use a Customization Specification (Sysprep)**, then select the customization specification you created for the Horizon Pool..



14. Review your selections. If any are incorrect, click **Previous** to correct them. When your selections are correct, click **Finish** to deploy the Desktop Pool.



15. Click the Status button to display the pools' status.



9.5 Creating Linked Clone Desktop Pools

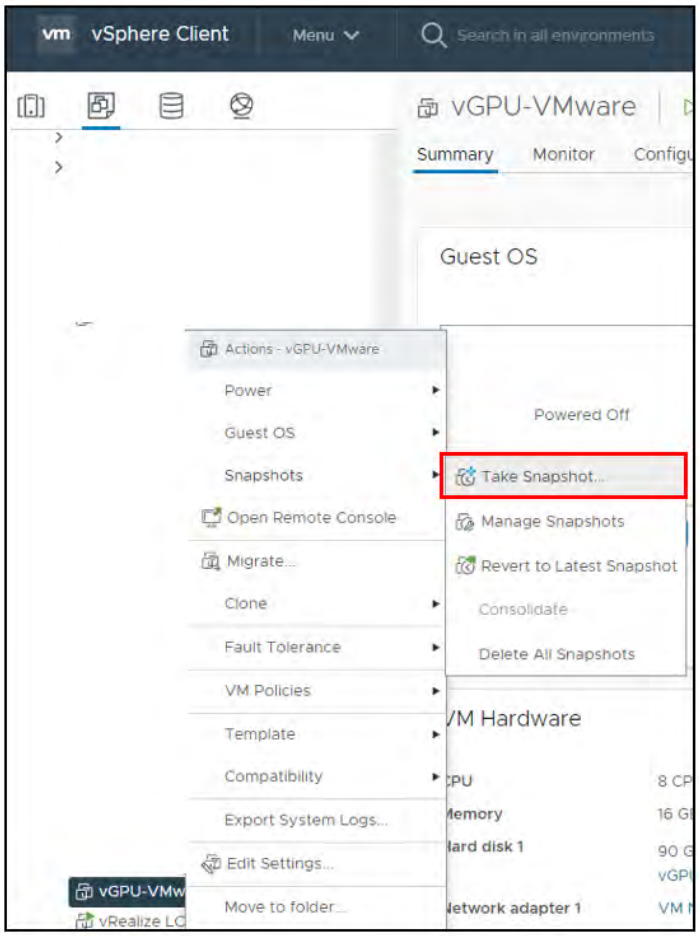
Creating a linked pool of VDI desktops is similar to creating a full clone pool. A linked pool leverages Horizon 8 to deploy multiple desktops from one master image. Linked clones consume less disk space than full clones and are easier to manage, upgrade, and deploy with little impact on the end-user.

As for a clone pool, you begin by selecting a VM as the golden master image. Instead of using a template of the VM, though, you take a snapshot of it.

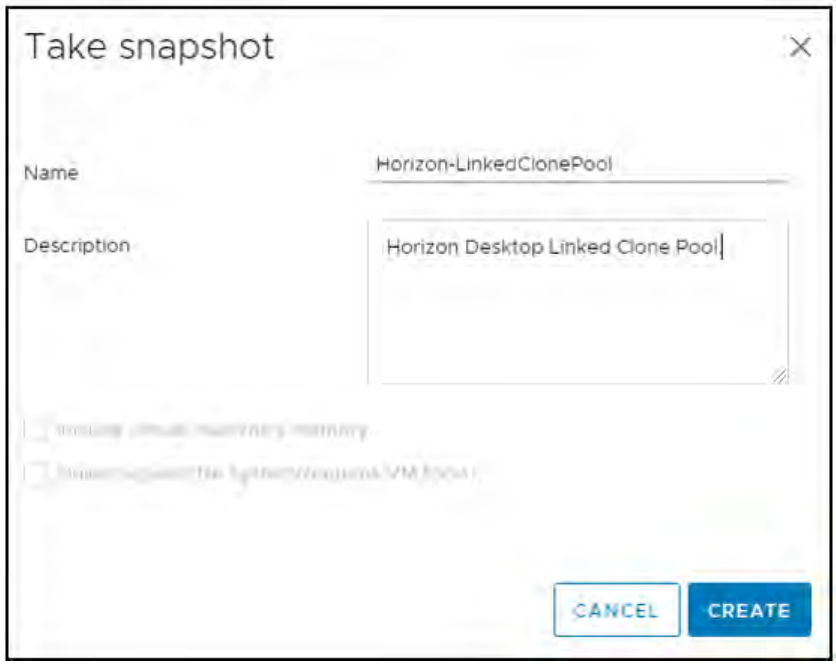
To create a linked-clone desktop pool:

1. Start vSphere Web Client. Right-click the gold master image VM.

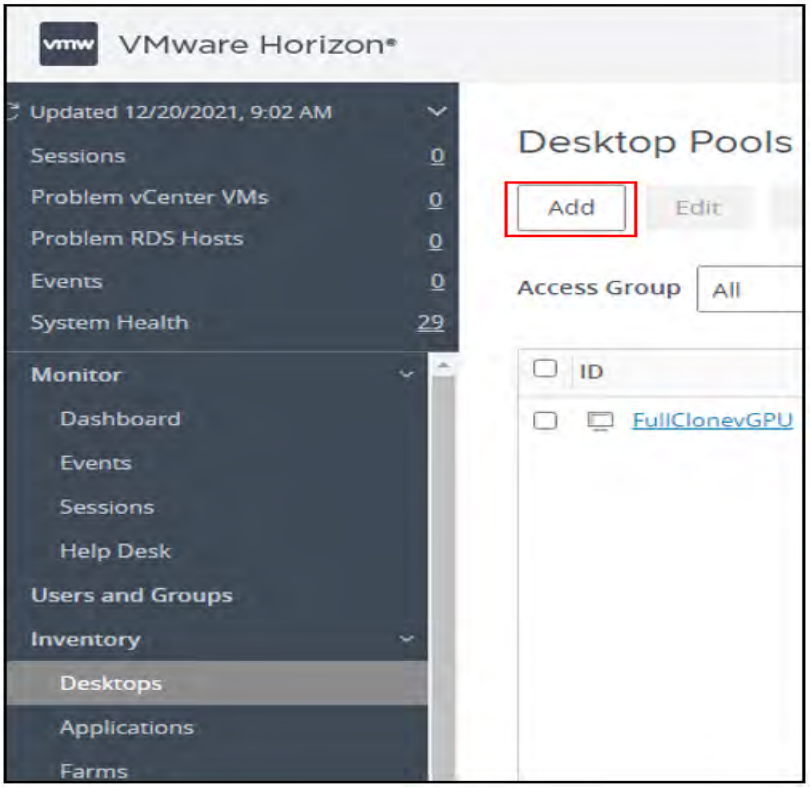
   Select **Snapshots** > **Take Snapshot** to open the Take Snapshot window.



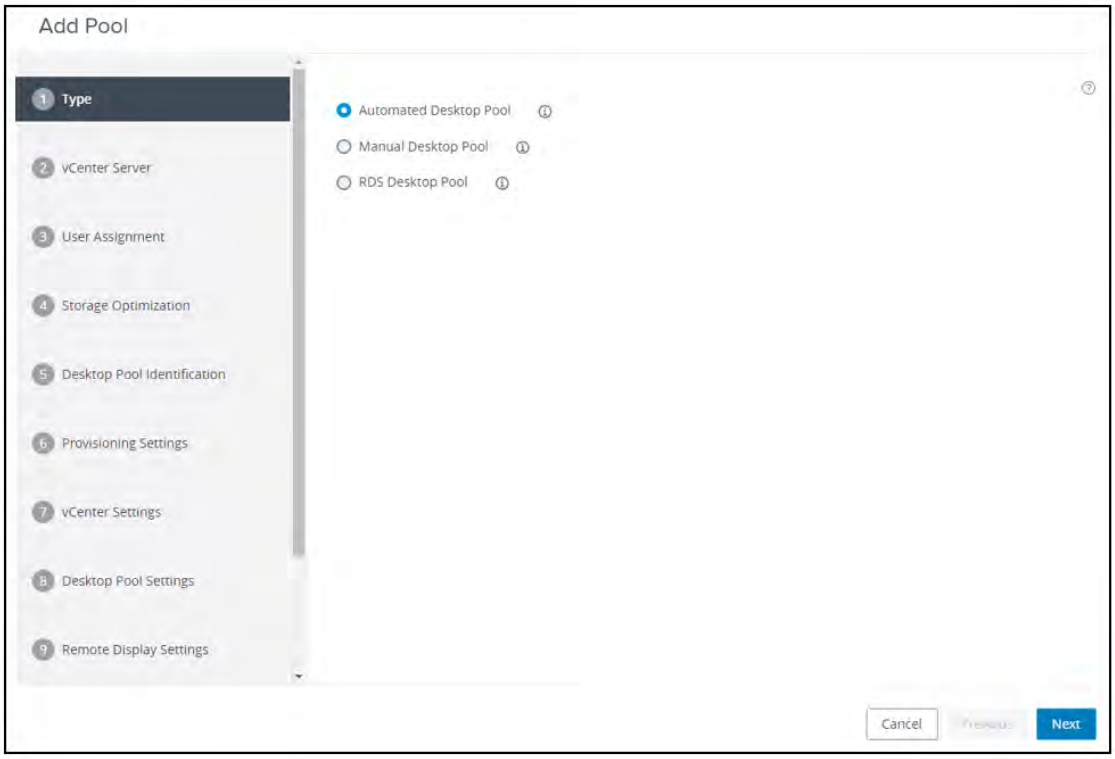
2. Name the snapshot to conform to your VM naming standards. Give the snapshot a meaningful description, then click OK to create the snapshot.



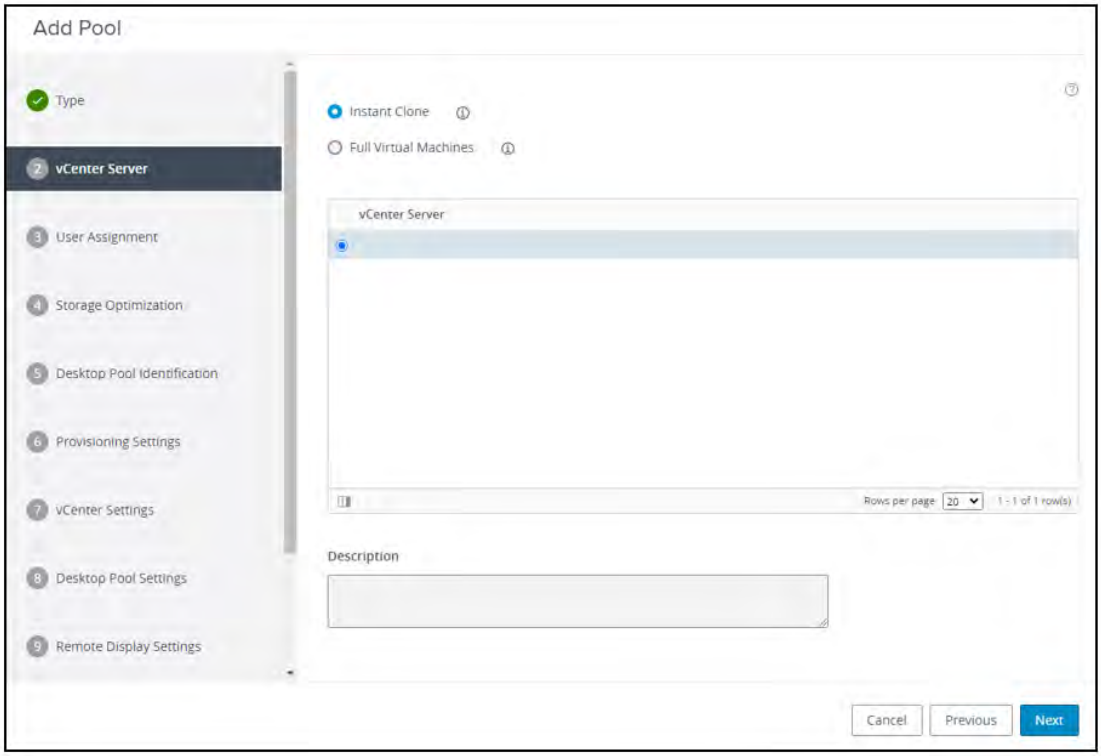
3. Start the VMware Horizon Administrator. Select Desktops in the Inventory pane, then click the Add button. Horizon Administrator starts the Desktop Pool wizard.



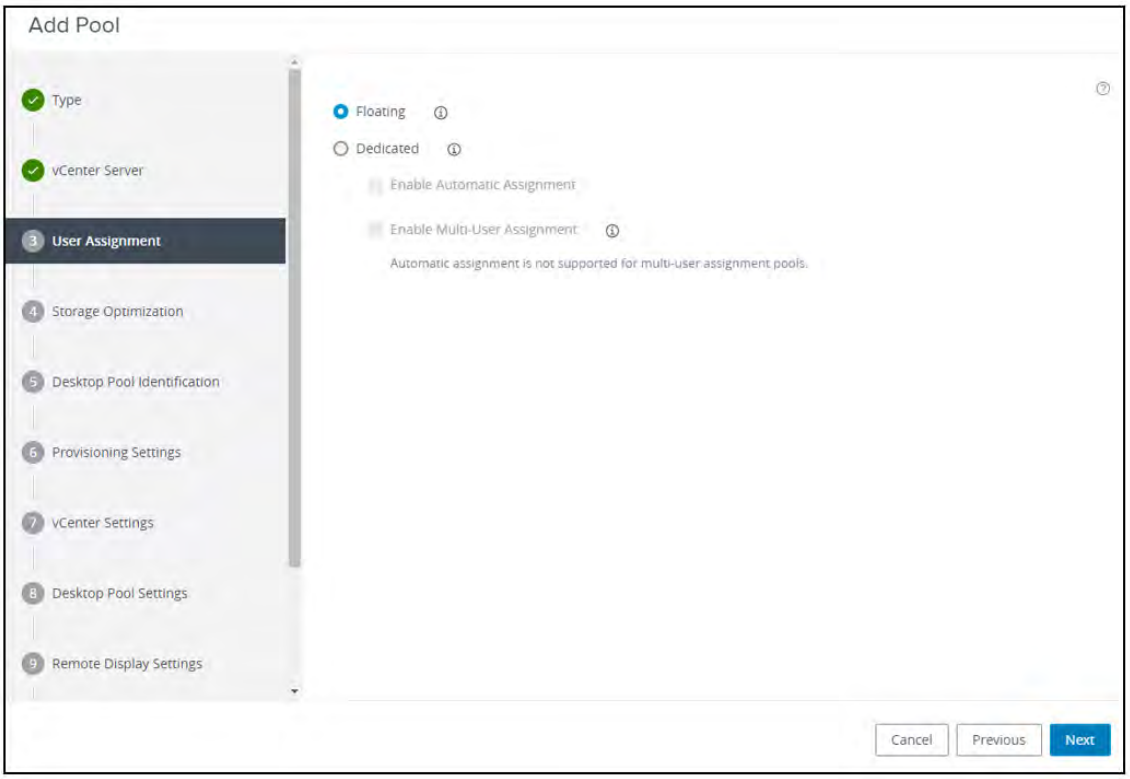
4. Select Automated Desktop Pool, then click Next.



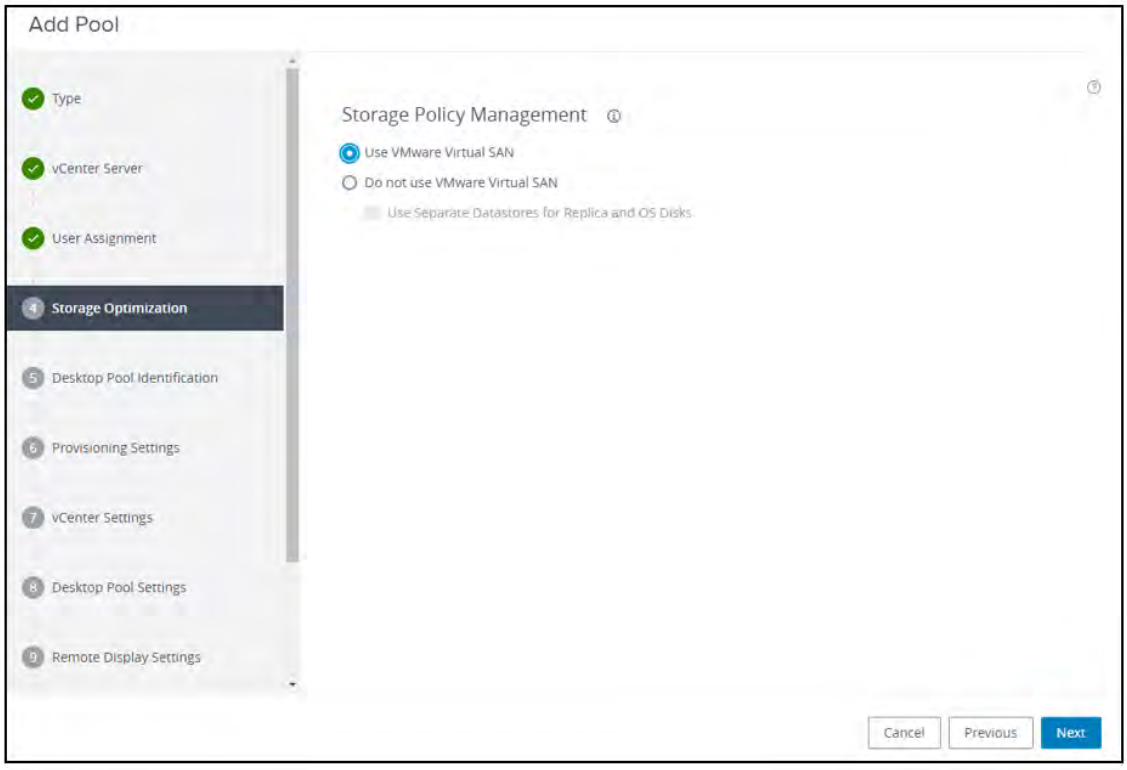
5. Select the vCenter Server instance. You must select the Instant Clone Radio button for the Linked Clones pool. Click **Next**.



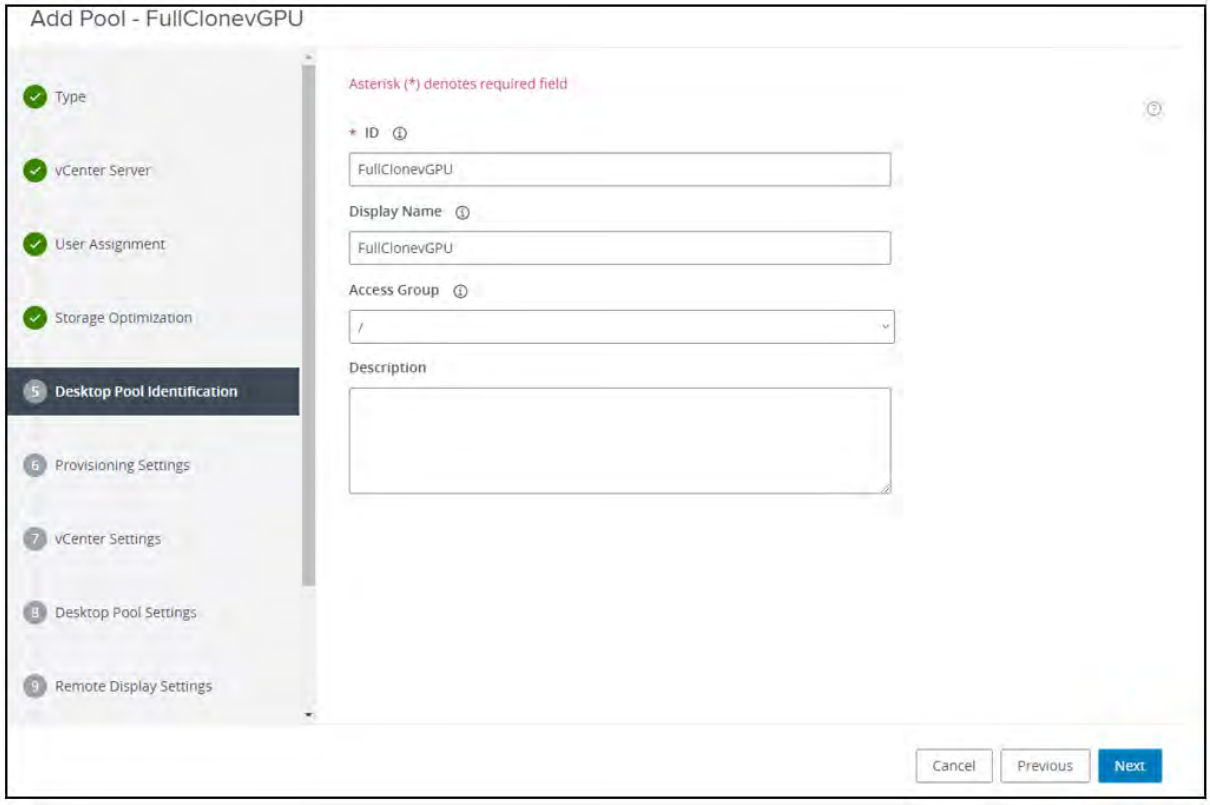
6. Select Floating as the User Assignment type. Click **Next**.



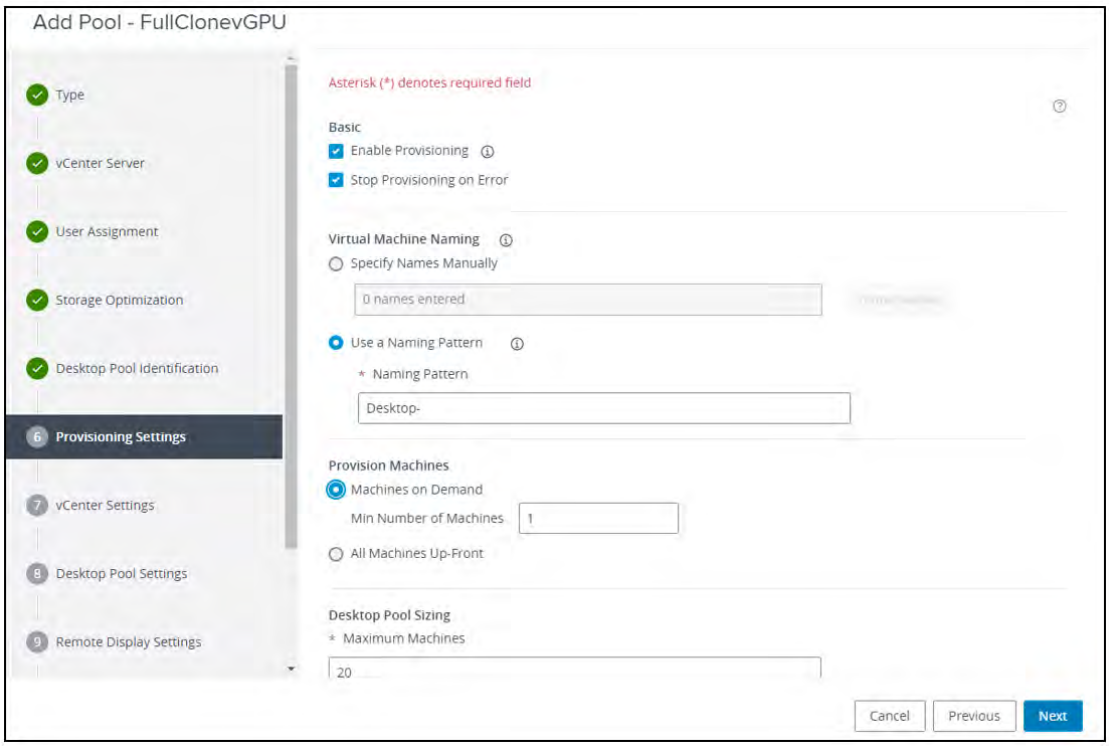
7. Select the **Use VMware virtual SAN** Radio button for the **Storage Optimization**. Click Next.



8. Enter a unique ID and a display name for the desktop pool. Click Next.

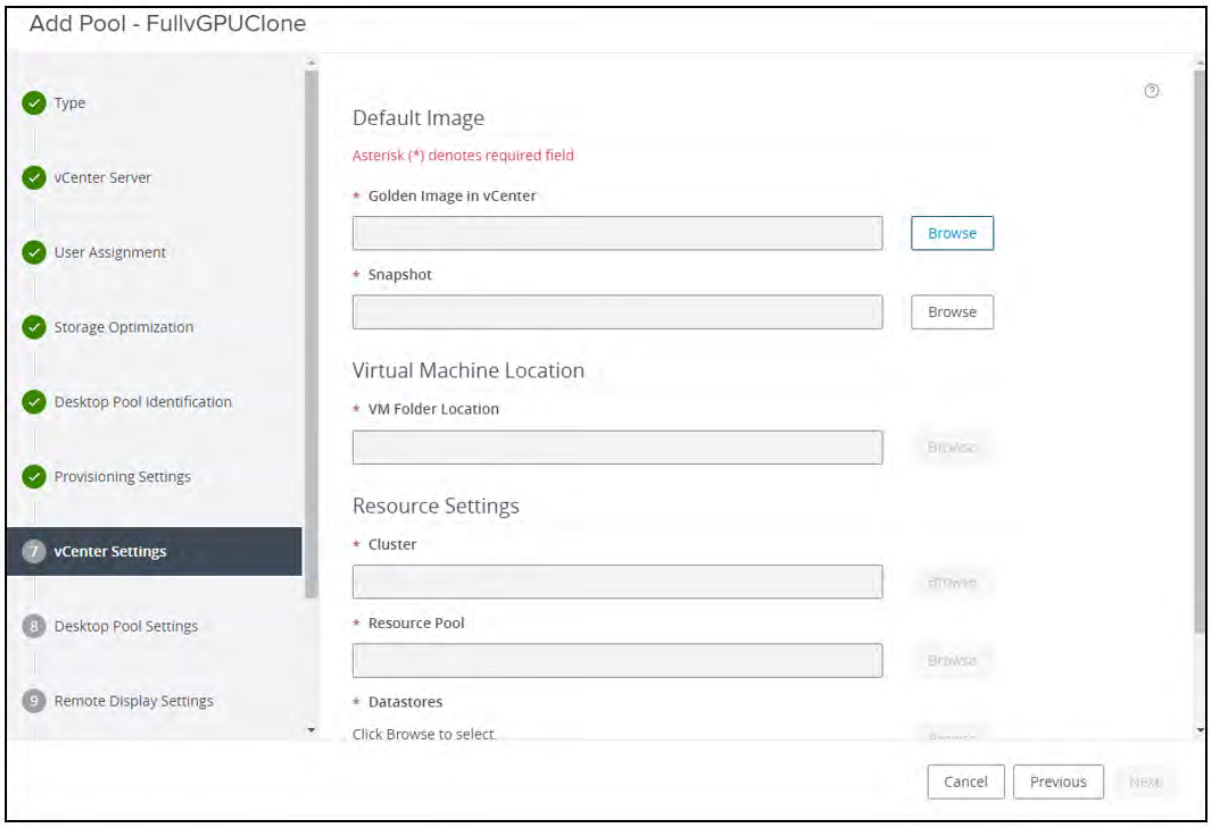


9. Enter the Provisioning Settings for your environment. Click Next.



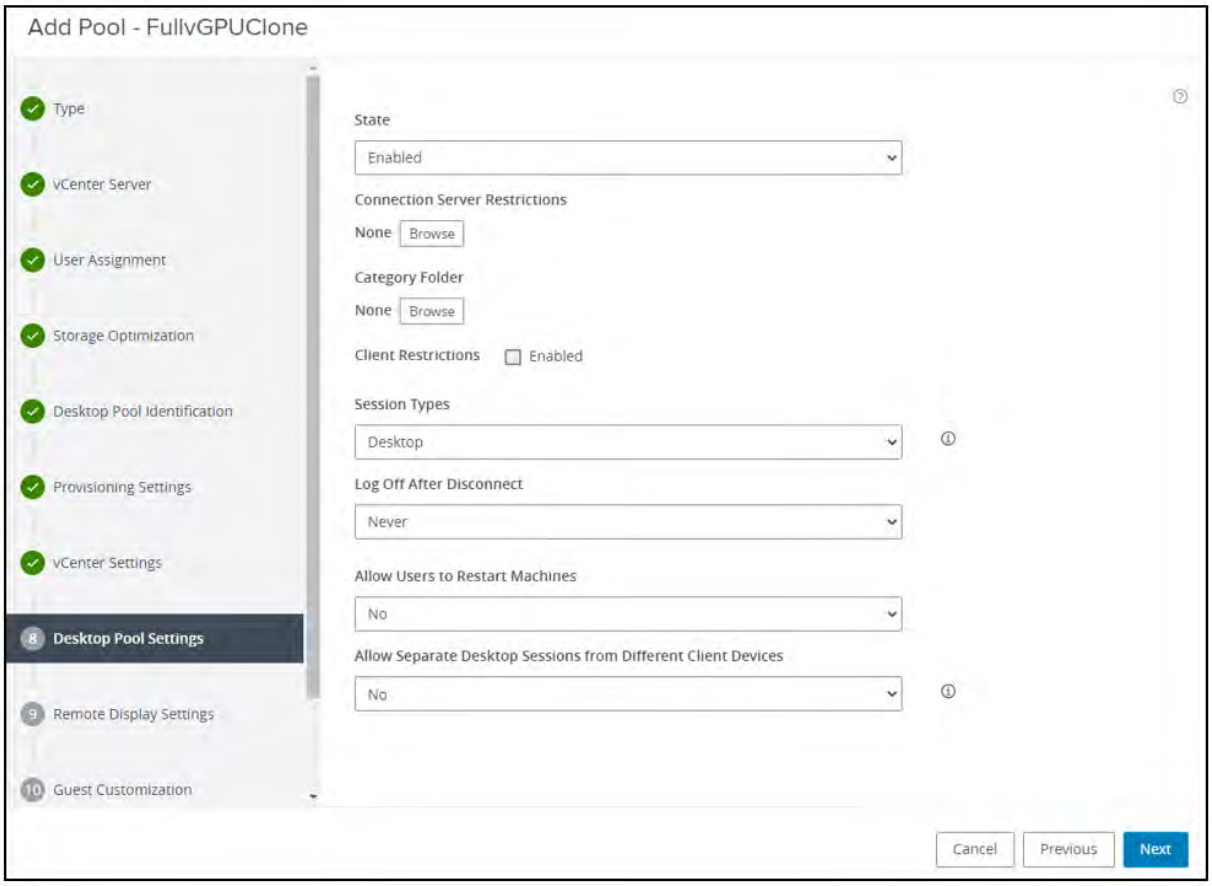
10. In the vCenter Settings tab. Fill in the details for your Horizon Pool.

Vmware suggests using the Worksheet to Create an Automated Full-Clone Desktop Pool to prepare your configuration options before creating the desktop pool.

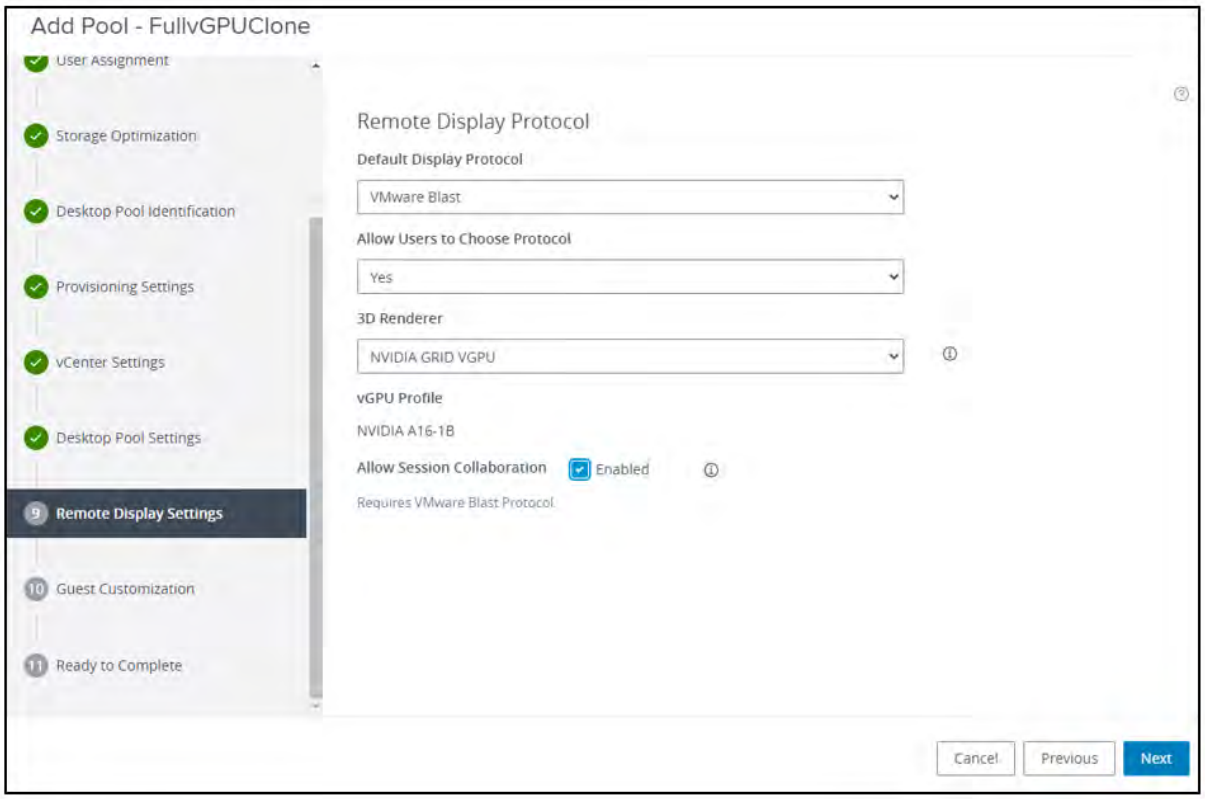


11. In the Desktop Pool Settings tab. Fill in the details for your Horizon Desktop Pool.

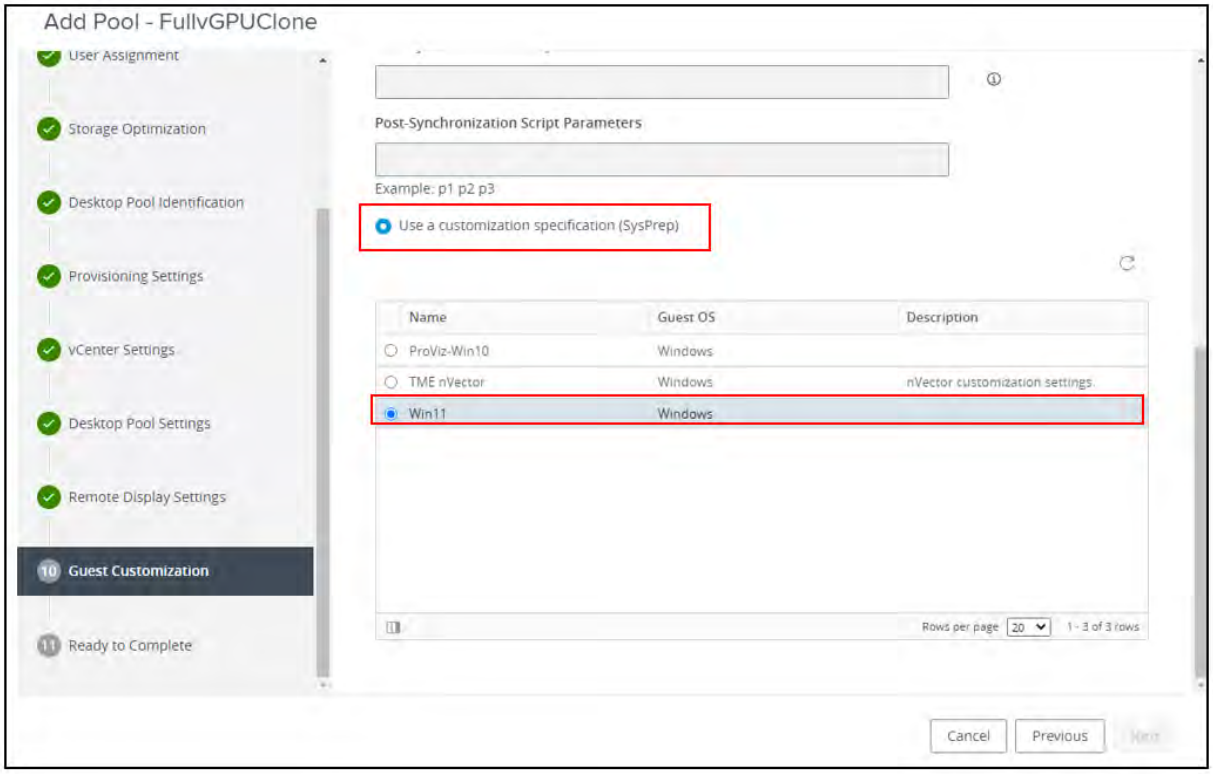
    Vmware suggests using the Worksheet to Create an Automated Full-Clone Desktop Pool to prepare your configuration options before creating the desktop pool.



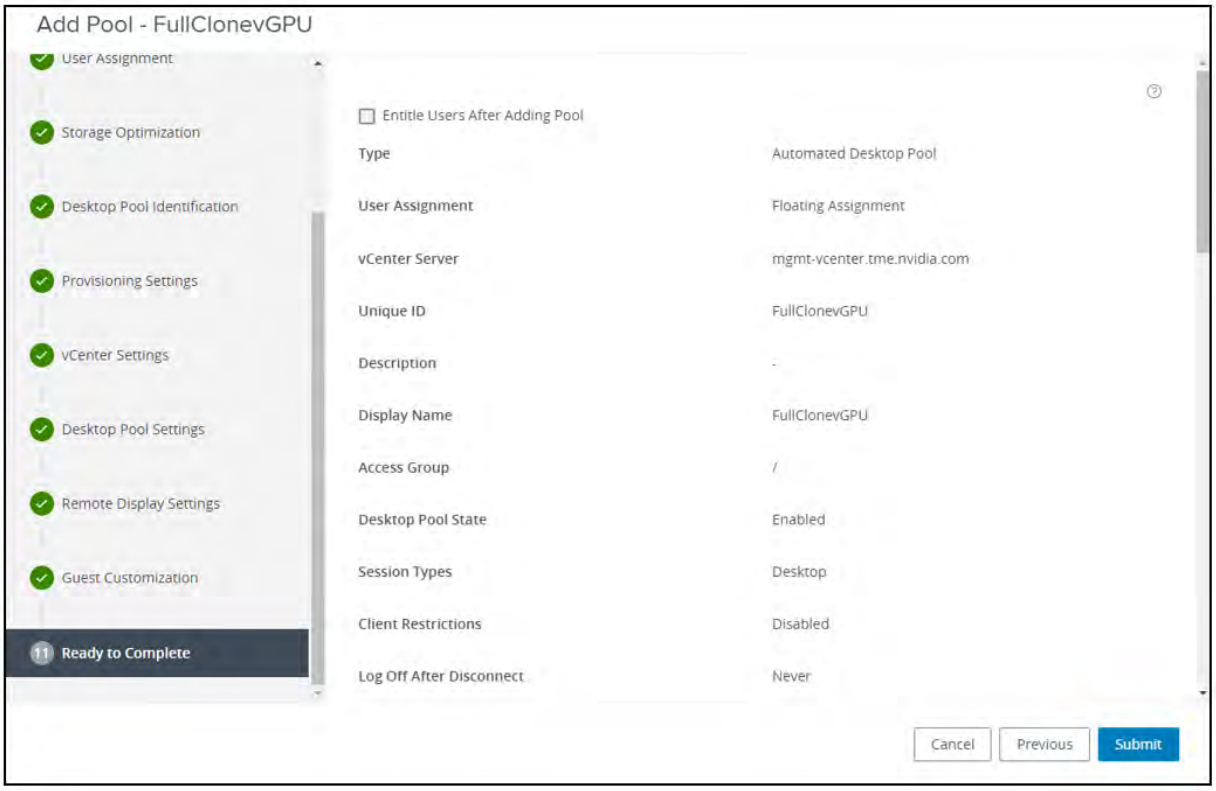
12. In the Remote Display Settings tab. Fill in the details for your Horizon Desktop Pool.



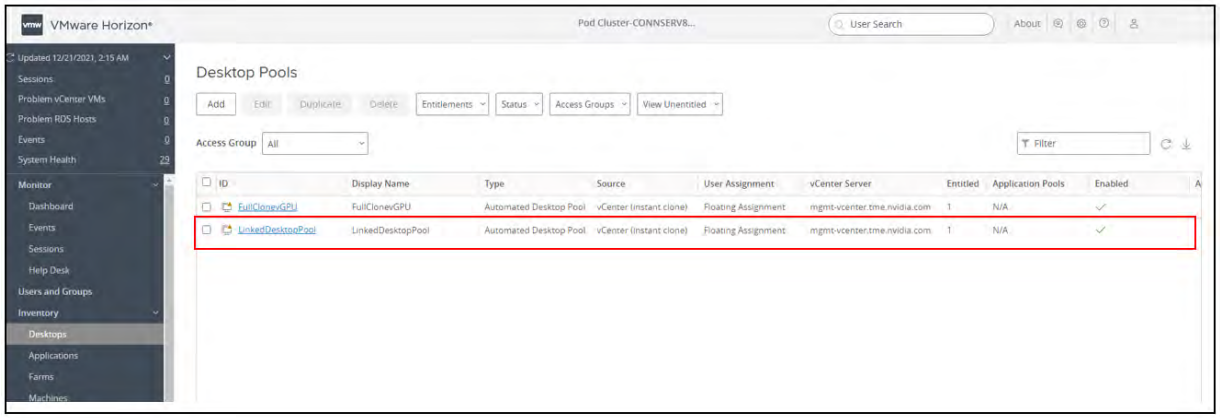
13. In the **Guest Customization** tab. Scroll towards the bottom and Select the customization specification (Sysprep), then select the customization specification you created for the Horizon Pool..



14. Review your selections. If any are incorrect, click Back to correct them. When your selections are correct, click Finish to deploy the Desktop Pool.



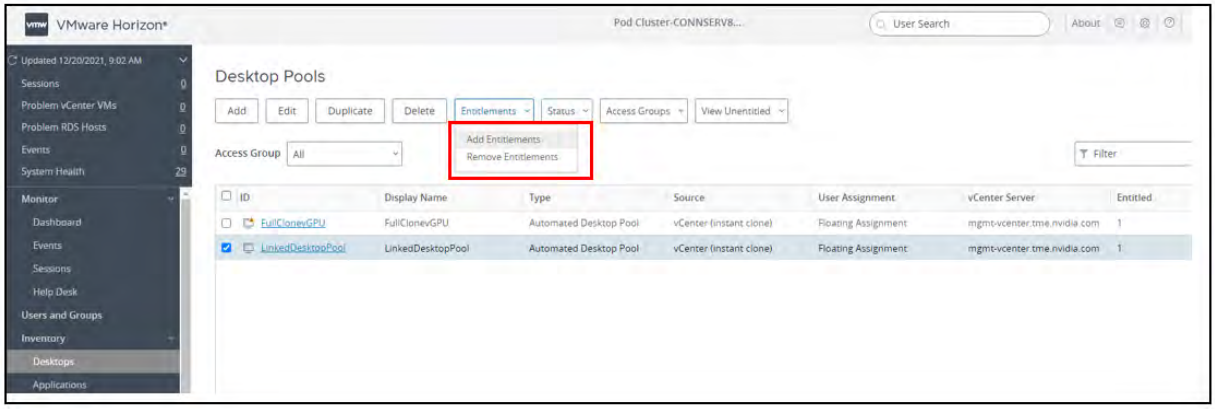
15. Click the Status button to display the pools' status.



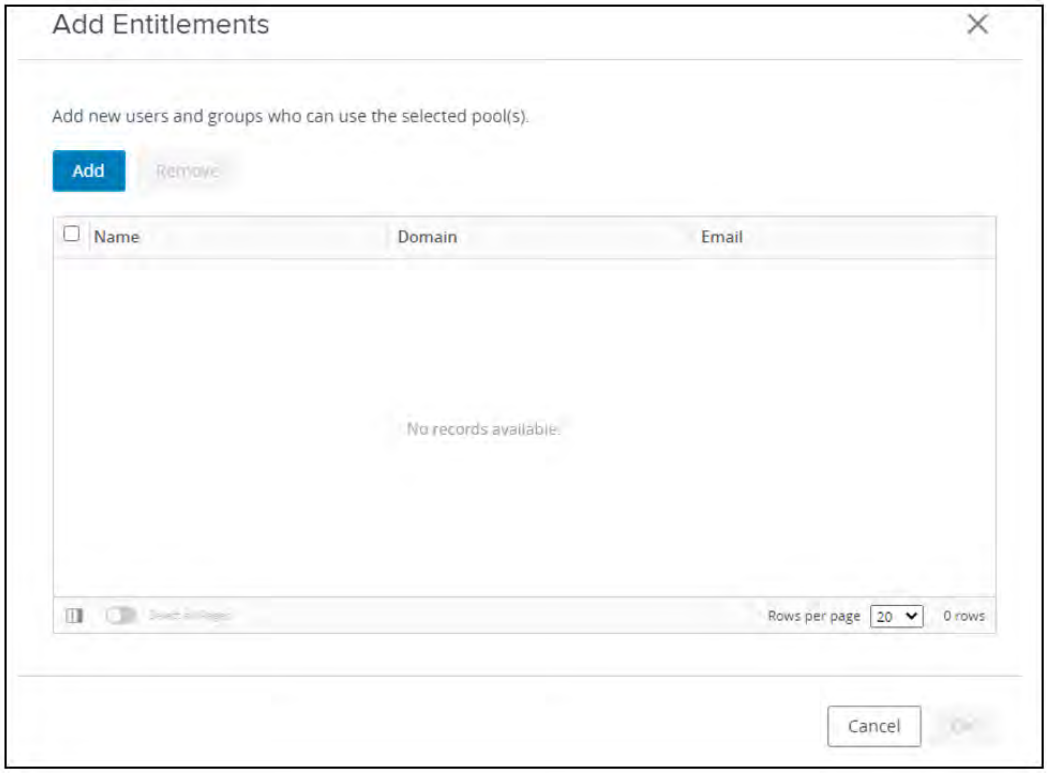
9.6 Enabling User Access to Desktop Pools

Entitle users to access the VDI desktops in a desktop pool.

1. Start or return to VMware Horizon Administrator and click Desktop Pools in the left-hand pane.



2. Click the Add button to enter your entitlements.

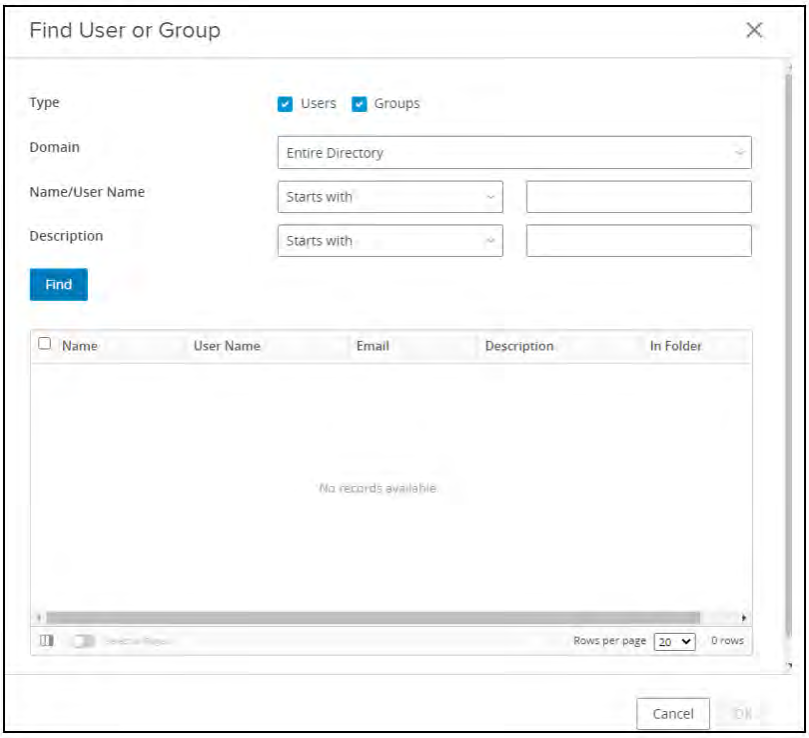


3. In the **Find User or Group** window.

   Check the Users checkbox or the Groups checkbox (or both) to display the defined users or groups.

   Enter a string in the **Name/Username** field or the **Description** field (or both) to limit the display to users and/or groups whose name or description (or both) match that string.

   By default, “matching the string” means “containing the string.” Possible matching criteria are “Contains,” “Begins with,” etc. You can change the criteria for a match on either field by selecting a different entry in the dropdown between the field’s label and the field’s matching string.



4. When the **Find User** or **Group** contains the exact set of users and/or groups that you want to entitle, click OK to entitle them and close the window.

Chapter 10. VMware Horizon Client

Before connecting to a virtual desktop over a Blast/PCoIP connection, VMware Horizon Client must be installed on a desktop or a device from which the virtual desktop will be accessed.

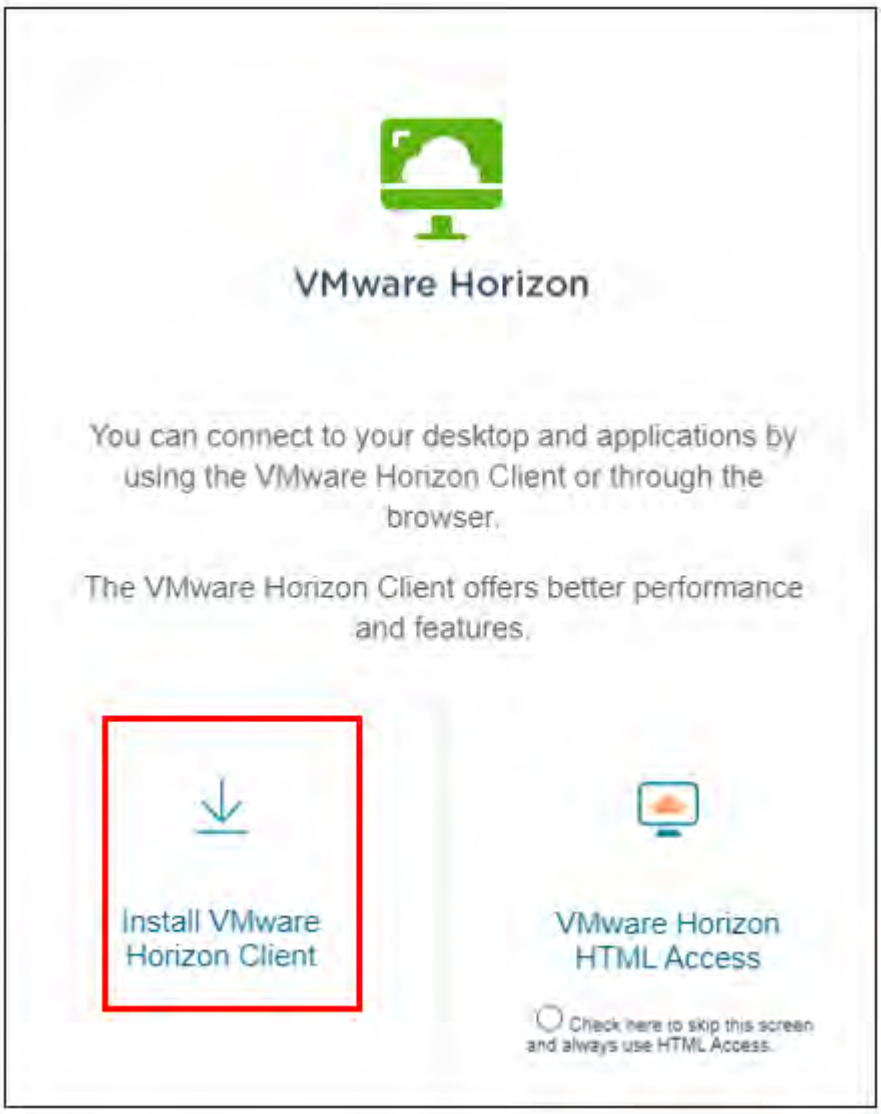
This Chapter covers The VMware Horizon Client:

- Installing the VMware Horizon Client
- Configuring the VMware Horizon Client Connection

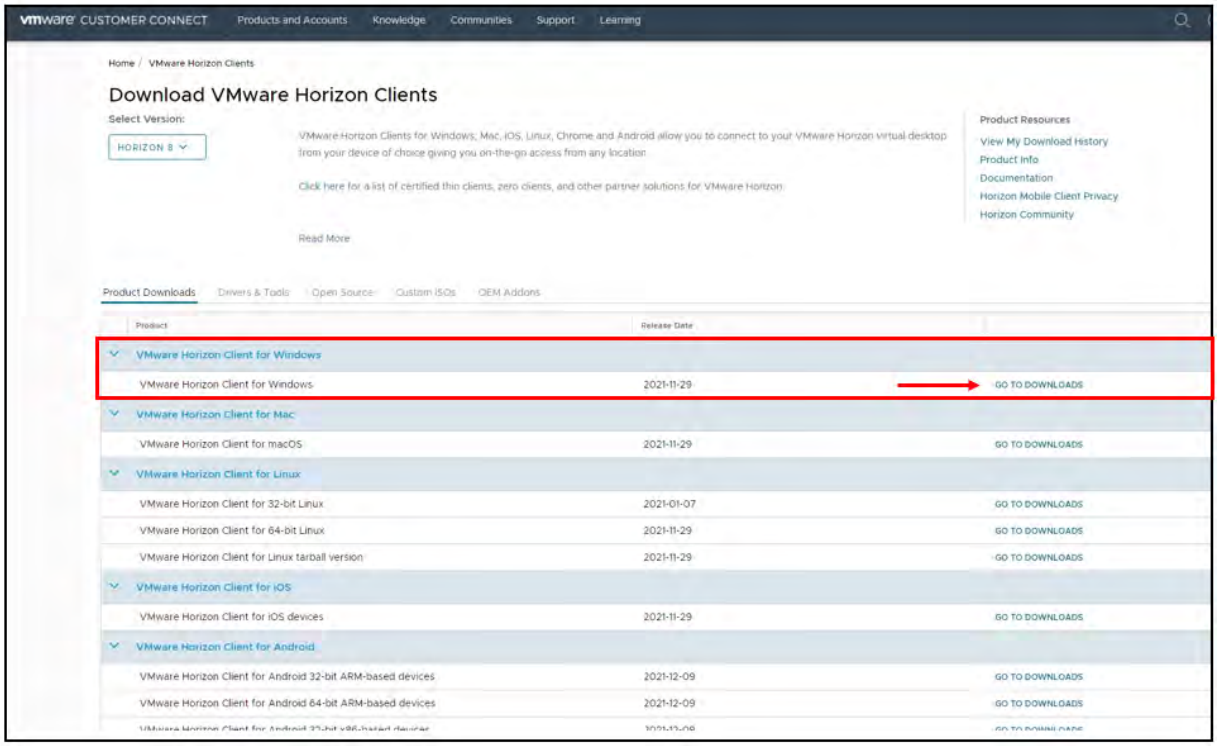
10.1 Installing VMware Horizon Client

To install VMware Horizon Client:

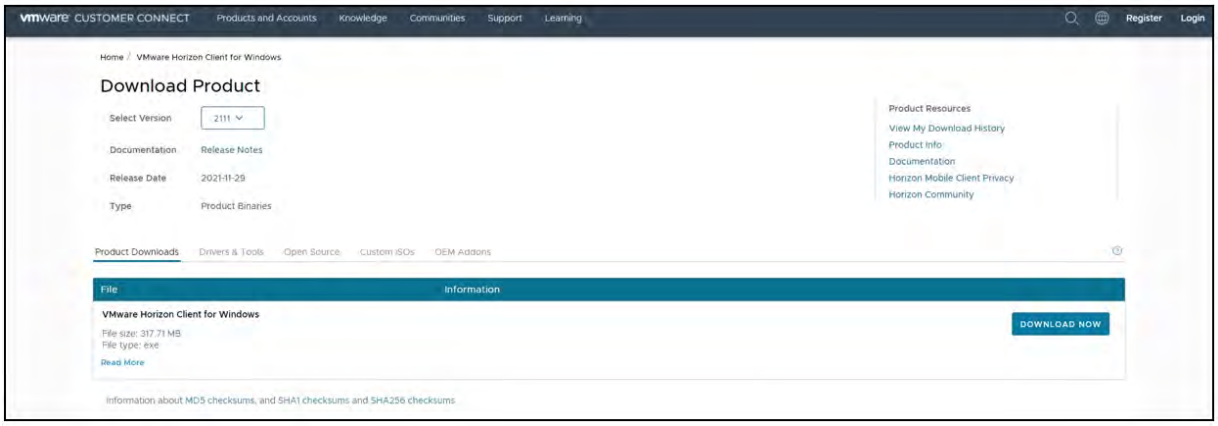
1. Log into the physical device, then open an internet browser. Navigate to the Horizon Connection Server URL installed earlier in this guide. See Section 4.2 - Installing the Horizon Connection Server. Click Install VMware Horizon Client.



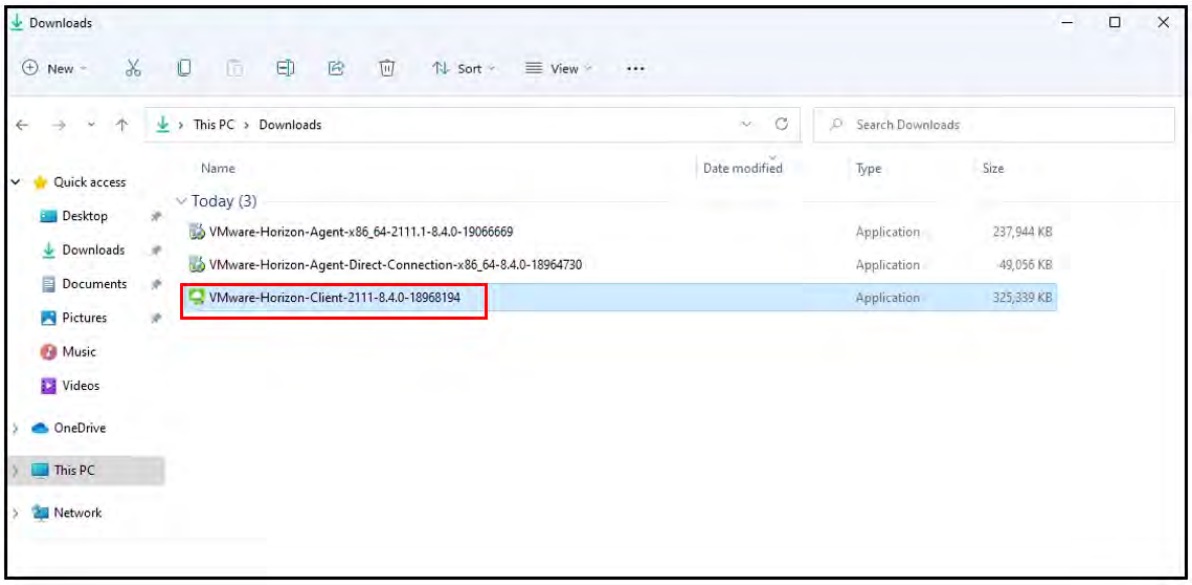
2. The browser opens the **Download VMware Horizon Clients** page. Locate the download for the appropriate OS and click **Go to Downloads** to its right.



3. The browser opens the **Download Product** page. Click **Download Now** to begin downloading.

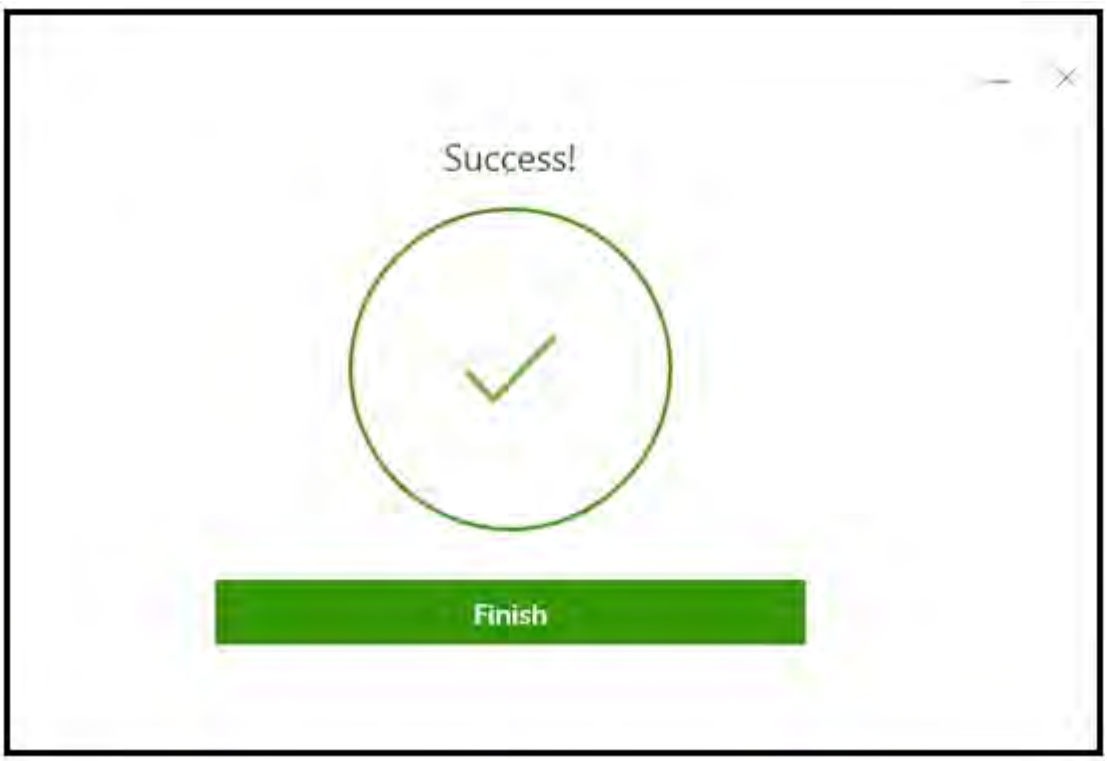


4. When the download is complete, locate the downloaded installer and double-click it to begin the installation. The installer prompts you to accept the privacy agreement and license.



5. Click **Agree & Install** to continue the installation.

6. The installer displays the Success window when the installation is done. Click Finish to complete the installation.

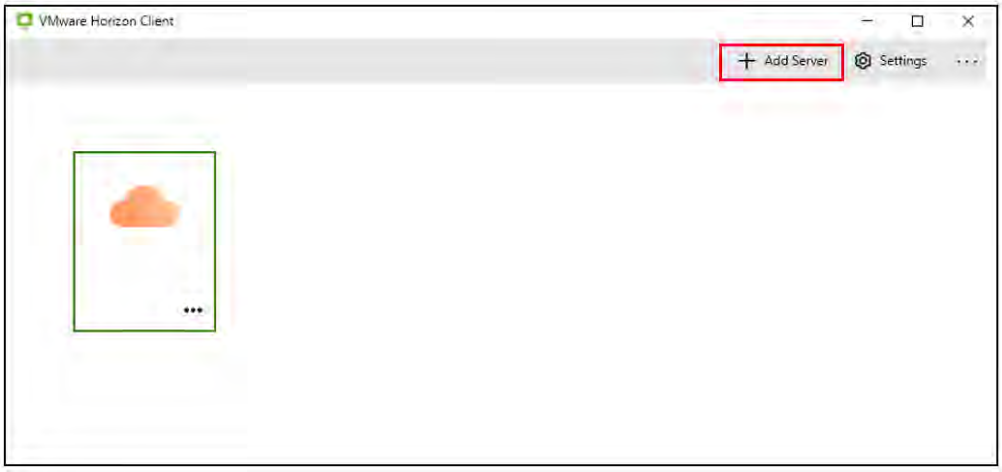


7. The installer displays the Restart window. Click **Restart Now** to restart the physical desktop. When the restart is complete, log in to see that Horizon Client installed.

10.2 Configuring the VMware Horizon Client Connection

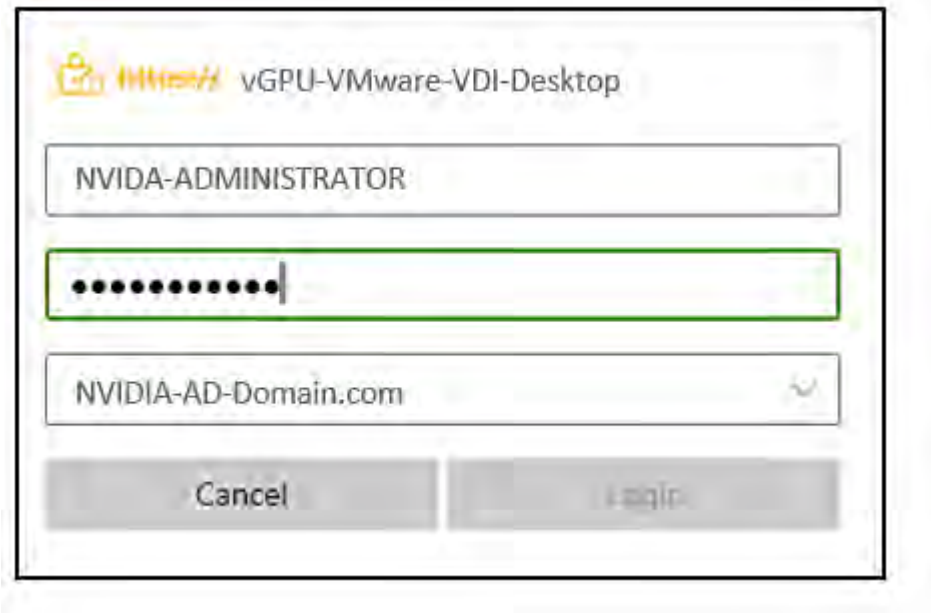
To configure the VMware Horizon Client:

1. Start VMware Horizon Client on the desktop from which you want to connect. The application displays a splash screen, then the home window. Click **Add Server** to register a new virtual machine connection.



2. VMware Horizon Client prompts you for the **Name of the Connection Server**. Enter the IP address or the FQDN of the VDI desktop you want to connect to, then click **Continue**. (You are connecting directly to a desktop, so do not enter the Horizon Connection Broker address at this time.)

3. VMware Horizon Client displays a log-in window. Enter the local username and password (or the domain user's username and password if the virtual machine is a domain member) and click **Login**.


vGPU-VMware-VDI-Desktop
NVIDA-ADMINISTRATOR
NVIDIA-AD-Domain.com
Cancel

Appendix A. About This Document

A.1 Related Documentation

NVIDIA publishes several other documents that are helpful to users of VMware Hypervisor with NVIDIA vGPU software. See the NVIDIA Virtual GPU (vGPU) resources page for additional information about NVIDIA vGPU technology, including:

- NVIDIA Virtual GPU Technology
- Purchasing Guide for NVIDIA vGPU Solutions
- Relevant White Papers
- Relevant videos

A.2 Support Contact Information

NVIDIA and other vendors provide several technical support resources to assist you:
Contact NVIDIA Sales Representatives | NVIDIA

The NVIDIA vGPU resources page describes additional contact methods to help you get the answers you need as soon as possible.

Notice

This document is provided for information purposes only and shall not be regarded as a warranty of a certain functionality, condition, or quality of a product. NVIDIA Corporation (“NVIDIA”) makes no representations or warranties, expressed or implied, as to the accuracy or completeness of the information contained in this document and assumes no responsibility for any errors contained herein. NVIDIA shall have no liability for the consequences or use of such information or for any infringement of patents or other rights of third parties that may result from its use. This document is not a commitment to develop, release, or deliver any Material (defined below), code, or functionality.

NVIDIA reserves the right to make corrections, modifications, enhancements, improvements, and any other changes to this document, at any time without notice.

Customer should obtain the latest relevant information before placing orders and should verify that such information is current and complete.

NVIDIA products are sold subject to the NVIDIA standard terms and conditions of sale supplied at the time of order acknowledgement, unless otherwise agreed in an individual sales agreement signed by authorized representatives of NVIDIA and customer (“Terms of Sale”). NVIDIA hereby expressly objects to applying any customer general terms and conditions with regards to the purchase of the NVIDIA product referenced in this document. No contractual obligations are formed either directly or indirectly by this document.

NVIDIA makes no representation or warranty that products based on this document will be suitable for any specified use. Testing of all parameters of each product is not necessarily performed by NVIDIA. It is customer’s sole responsibility to evaluate and determine the applicability of any information contained in this document, ensure the product is suitable and fit for the application planned by customer, and perform the necessary testing for the application in order to avoid a default of the application or the product. Weaknesses in customer’s product designs may affect the quality and reliability of the NVIDIA product and may result in additional or different conditions and/or requirements beyond those contained in this document. NVIDIA accepts no liability related to any default, damage, costs, or problem which may be based on or attributable to: (i) the use of the NVIDIA product in any manner that is contrary to this document or (ii) customer product designs.

No license, either expressed or implied, is granted under any NVIDIA patent right, copyright, or other NVIDIA intellectual property right under this document. Information published by NVIDIA regarding third-party products or services does not constitute a license from NVIDIA to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property rights of the third party, or a license from NVIDIA under the patents or other intellectual property rights of NVIDIA.

Reproduction of information in this document is permissible only if approved in advance by NVIDIA in writing, reproduced without alteration and in full compliance with all applicable export laws and regulations, and accompanied by all associated conditions, limitations, and notices.

THIS DOCUMENT AND ALL NVIDIA DESIGN SPECIFICATIONS, REFERENCE BOARDS, FILES, DRAWINGS, DIAGNOSTICS, LISTS, AND OTHER DOCUMENTS (TOGETHER AND SEPARATELY, “MATERIALS”) ARE BEING PROVIDED “AS IS.” NVIDIA MAKES NO WARRANTIES, EXPRESSED, IMPLIED, STATUTORY, OR OTHERWISE WITH RESPECT TO THE MATERIALS, AND EXPRESSLY DISCLAIMS ALL IMPLIED WARRANTIES OF NONINFRINGEMENT, MERCHANTABILITY, AND FITNESS FOR A PARTICULAR PURPOSE. TO THE EXTENT NOT PROHIBITED BY LAW, IN NO EVENT WILL NVIDIA BE LIABLE FOR ANY DAMAGES, INCLUDING WITHOUT LIMITATION ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL, PUNITIVE, OR CONSEQUENTIAL DAMAGES, HOWEVER CAUSED AND REGARDLESS OF THE THEORY OF LIABILITY, ARISING OUT OF ANY USE OF THIS DOCUMENT, EVEN IF NVIDIA HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. Notwithstanding any damages that customer might incur for any reason whatsoever, NVIDIA’s aggregate and cumulative liability towards customer for the products described herein shall be limited in accordance with the Terms of Sale for the product.

Trademarks

NVIDIA and the NVIDIA logo are trademarks and/or registered trademarks of NVIDIA Corporation in the U.S. and other countries. Other company and product names may be trademarks of the respective companies with which they are associated.

VESA DisplayPort

DisplayPort and DisplayPort Compliance Logo, DisplayPort Compliance Logo for Dual-mode Sources, and DisplayPort Compliance Logo for Active Cables are trademarks owned by the Video Electronics Standards Association in the United States and other countries.

HDMI

HDMI, the HDMI logo, and High-Definition Multimedia Interface are trademarks or registered trademarks of HDMI Licensing LLC.

OpenCL

OpenCL is a trademark of Apple Inc. used under license to the Khronos Group Inc.

Copyright

NVIDIA Corporation | 2788 San Tomas Expressway, Santa Clara, CA 95051
http://www.nvidia.com