NVIDIA vGPU Software for Citrix Virtual Apps & Desktops on VMware vSphere

Deployment Guide

Document History

nv-quadro-vgpu-deployment-guide-citrixonvmware-v2-082020

Version	Date	Authors	Description of Change
01	August 17, 2020	AFS, JJC, EA	Initial Release
02	January 20, 2021	AFS	Marketing Update

Table of Contents

Chapter 1. Getting Started

NVIDIA vGPU allows multiple virtual machines (VMs) to have simultaneous, direct access to a single physical GPU, using the same NVIDIA graphics drivers that are deployed on non-virtualized operating systems. This gives VMs unparalleled graphics performance and application compatibility, together with cost-effectiveness and scalability brought about by sharing a GPU among multiple workloads.

This chapter covers how NVIDIA vGPU solutions fundamentally alters the landscape of desktop virtualization and enables users and applications of all levels of complexity and graphics requirements to utilize said solutions. It also describes the NVIDIA vGPU architecture, the GPUs recommended for virtualization, the three virtual GPU software editions, and key standards supported by NVIDIA virtual GPU technology.

1.1 Why NVIDIA vGPU?

The promise of desktop virtualization, realized for server workloads years ago, is flexibility and manageability. Initially, desktop virtualization was used where flexibility and security were the primary drivers due to cost considerations. The democratization of technology over the years has reduced the total cost of ownership of desktop virtualization. This, along with advances in storage and multi-core processors make for a reasonable and/or advantageous cost to ownership.

The big remaining 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, but those technologies do not support the rich application support needed to be successful and ensure end user adoption. This compares to dedicated GPU pass-through with vDGA that provides 100% application compatibility, but only for the highest-end use cases due to the high cost with 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 has been challenging for broad use cases in desktop virtualization. Meanwhile, high-end 3D applications simply did not work in a virtualized environment or were so expensive to implement with vDGA it was reserved for only the most limited of circumstances.

Today, this is no longer true to thanks to NVIDIA vGPU technology combined with Citrix Virtual Apps & Desktops. NVIDIA vGPU technology allows the flexibility where multiple virtual desktops share a single physical GPU that may reside on a single physical PCI card. This provides the 100% application compatibility of vDGA pass-through graphics, but with lower cost of multiple desktops sharing a single graphics card to provide a rich, yet more cost-effective user experience. With Citrix Virtual Apps & Desktops you can centralize, pool, and more easily manage traditionally complex and expensive,

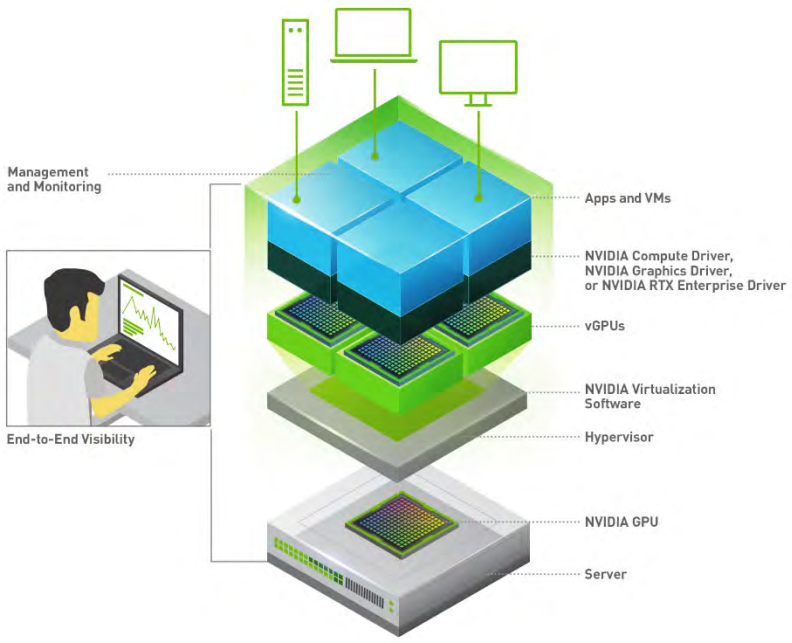
distributed workstations and desktops. Now all your user groups can take advantage of the promise of virtualization.

1.2 NVIDIA vGPU Architecture

A high-level architecture of NVIDIA vGPU is illustrated below. The NVIDIA virtual GPU enabled VDI environment is illustrated below 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 allows multiple VMs to share a single GPU or if there are multiple GPU's in the server, they can be aggregated so that a single VM can access multiple GPUs. This GPU enabled environment, provides not only unprecedented performance, it also enables support for more users on a server because work that was typically done by the CPU, can 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 vGPUs in the same manner as a physical GPU that has been passed through by the hypervisor. In the VM itself, vGPU drivers are installed which pertain to the different license levels that are available. The NVIDIA Virtual Compute Server license pertains to the NVIDIA compute driver.

Figure 2.1 NVIDIA vGPU Platform Solution Architecture



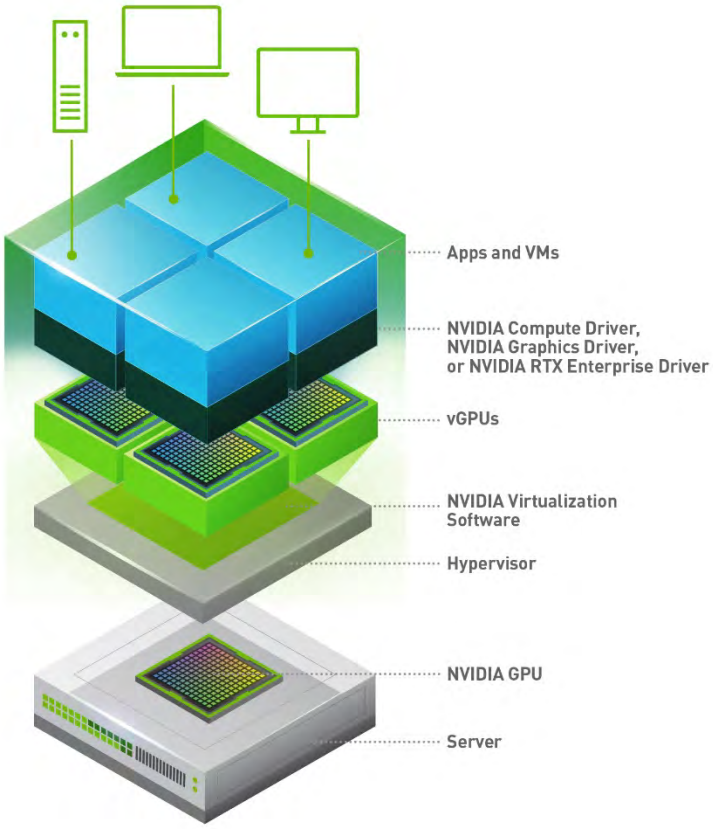
NVIDIA vGPUs are comparable to conventional GPUs in that they have a fixed amount of GPU-Memory and one or more virtual display outputs or *heads*. Multiple heads support multiple displays. Managed by the NVIDIA vGPU Manager installed in the hypervisor, the vGPU Memory is allocated out

of the physical GPU frame buffer at the time the vGPU is created. The vGPU retains exclusive use of that GPU Memory until it is destroyed.

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

All vGPUs resident on a physical GPU share access to the GPU's engines, including the graphics (3D) and video decode and encode engines. Figure 1-2 shows the vGPU internal architecture. VM's guest OS leverages direct access to the GPU for performance and critical fast paths. Non-critical performance management operations use a para-virtualized interface to the NVIDIA Virtual GPU Manager.

Figure 3.2 NVIDIA vGPU Internal Architecture



1.3 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 RTX Virtual Workstation (vWS)
- NVIDIA Virtual PC (NVIDIA vPC)
- NVIDIA Virtual Apps (NVIDIA vApps)

To run these software products, you need an NVIDIA GPU and software license that addresses your specific use case. For NVIDIA Virtual Applications (NVIDIA vApps) you can use Citrix Virtual Apps and for NVIDIA Virtual PC (NVIDIA vPC) and NVIDIA RTX Virtual Workstation (vWS) you can use Citrix Virtual Desktop.

For further details on vGPU licensing, please refer to the following guide:

NVIDIA Virtual GPU Software Packaging, Pricing, and Licensing Guide

1.4 Supported Graphics Protocols

This version of NVIDIA vGPU software includes support for:

- Full DirectX 12, Direct2D, and DirectX Video Acceleration (DXVA)
- OpenGL 4.6
- NVIDIA vGPU SDK (remote graphics acceleration)
- Vulkan 1.1
- OpenCL and CUDA applications **WITHOUT** Unified Memory are supported on supported GPUs.
  - https://docs.nvidia.com/grid/11.0/grid-vgpu-user-guide/index.html#cuda-open-cl-support-vgpu

Note: Unified Memory and CUDA tools are NOT supported on NVIDIA vGPU

1.5 Before You Begin

This section describes the general prerequisites and some general preparatory steps that must be addressed before proceeding with the deployment.

Note: This deployment guide assumes you are building an environment as a proof of concept and is not meant to be a production deployment. As a result, choices made are meant to speed up and ease the process. See the corresponding guides for each technology, and make choices appropriate for your needs, before building your production environment.

1.5.1 Server BIOS Settings

Configure the BIOS as appropriate 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)
- VT-d or AMD IOMMU – Enabled

1.5.2 Citrix GPU Utilization Patch

KB4586830 & KB458639 address an issue with incorrect Canonical Display Driver (CDD) buffer flushing, which degrades performance in Remote Desktop Protocol (RDP) Windows 2000 Display Driver Model (XDDM) scenarios. This issue affects applications that use graphics processing units (GPU) to operate, such as Microsoft Teams, Microsoft Office, and web browsers.

- Server 2016 – KB4586830
- Server 2019 – KB4586839

Please follow the steps below to enable KB4586830 on Server 2016. It is not enabled by default post installation.

- To Enable the fix - reg add HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Policies\Microsoft\FeatureManagement\Overrides /v 1826589834 /t REG_DWORD /d 1 /f
- To Disable the fix - reg add HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Policies\Microsoft\FeatureManagement\Overrides /v 1826589834 /t REG_DWORD /d 0 /f

1.5.3 Citrix Virtual Machine Requirements

1.5.3.1 Delivery Controller

Supported operating systems:

- Windows Server 2019, Standard and Datacenter Editions, and with the Server Core option
- Windows Server 2016, Standard and Datacenter Editions, and with the Server Core option

Requirements:

- Microsoft .NET Framework 4.8 is installed automatically if it (or a later version) is not already installed.
- Microsoft Internet Information Services (installed automatically and used by a feature currently in development that is installed by the Citrix Orchestration Service).

- Windows PowerShell 3.0 or later.
- Microsoft Visual C++ 2017 Runtime, 32-bit and 64-bit.

1.5.3.2 Virtual Delivery Agent (VDA) for Desktop OS

Supported operating systems:

- Windows 10, (see edition support in the *Introduction* section. The following features are not supported on Windows 10: desktop composition redirection and legacy graphics mode.
- Windows 8.1, Professional and Enterprise Editions
- Windows 7 SP1, Professional, Enterprise, and Ultimate Editions

Requirements:

- Microsoft .NET Framework 4.5.2 (4.6 through 4.8 are also supported)
- Microsoft .NET Framework 3.5.1 (Windows 7 only)
- Microsoft Visual C++ 2013 and 2015 Runtimes, 32- and 64-bit
- PowerShell 3.0 or later

1.5.3.3 Virtual Delivery Agent (VDA) for Server OS

Supported operating systems:

- Windows Server 2016, Standard and Datacenter Editions
- Windows Server 2012 R2, Standard and Datacenter Editions
- Windows Server 2012, Standard and Datacenter Editions
- Windows Server 2008 R2 SP1, Standard, Enterprise, and Datacenter Editions

The installer automatically deploys the following requirements, which are also available on the Citrix installation media in the Support folders:

- Microsoft .NET Framework 4.5.2 (4.6 through 4.8 are also supported)
- Microsoft .NET Framework 3.5.1 (Windows Server 2008 R2 only)
- Microsoft Visual C++ 2013 and 2015 Runtimes, 32- and 64-bit
- PowerShell 3.0 or later

1.5.3.4 Additional Sizing Resources for Citrix

- Database sizing tool for XenDesktop 7
- Citrix VDI Best Practices for XenApp and XenDesktop 7.15 LTSR
- Citrix Virtual Apps and Desktops Single-Server Scalability
- Complete system requirements

1.5.4 Virtual GPU Evaluation licenses

In order to run a PoC/trial of NVIDIA Virtual GPU, a vGPU license is required. An evaluation license is available at the following address:

https://www.nvidia.com/object/vgpu-evaluation.html

Chapter 2. Installing VMware ESXi

This chapter covers the following VMware ESXi installation topics:

- Choosing an install method
- Preparing the USB boot media
- Installing ESXi from the USB media
- Initial host configuration
- Assigning a host license

Note: This deployment guide assumes you are building an environment as a proof of concept and is not meant to be a production deployment, as a result, choices made are meant to speed up and ease the process. See the corresponding guides for each technology, and make choices appropriate for your needs, before building your production environment.

For the purpose of this guide, ESXi 6.7 U3 is used as the hypervisor version.

2.1 Choosing the Installation method

With the ability to install from and onto a SD card or USB flash drive, ESXi offers flexibility verses local hard drive install. Please see vSphere documentation regarding best practices for logs when booting from USB or similar. In our main lab we used Supermicro's IPMI and virtual media to boot from ISO file and install on local storage.

2.2 Preparing USB Boot Media

For more information, see the VMware knowledgebase article Installing ESXi on a supported USB flash drive or SD flash card (2004784).

Booting ESXi from a USB flash drive is useful if your host has an existing ESXi Version 6.X or earlier installation that you want to retain.

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

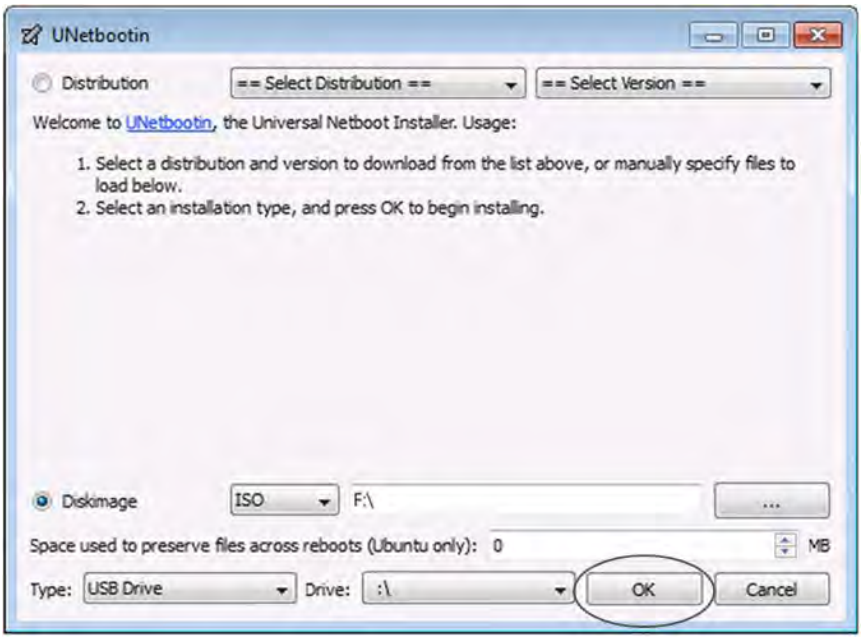
1. Download **UNetbootin** from http://unetbootin.sourceforge.net/.

The Windows version of the application does not include an installer; however, the OSX version is packaged in a **.DMG** file that you must mount. You must also copy the application to the **Applications** folder before launching.

2. Start the application, select **Diskimage**, and then click the **...** icon to browse for the installation **.ISO** file.

3. Navigate to the location that contains the installation **.ISO** file and then select **Open**.
4. Select the mounted USB drive on which to perform the installation and then select **OK.**

   The copying process begins, and a series of progress bars are displayed.



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

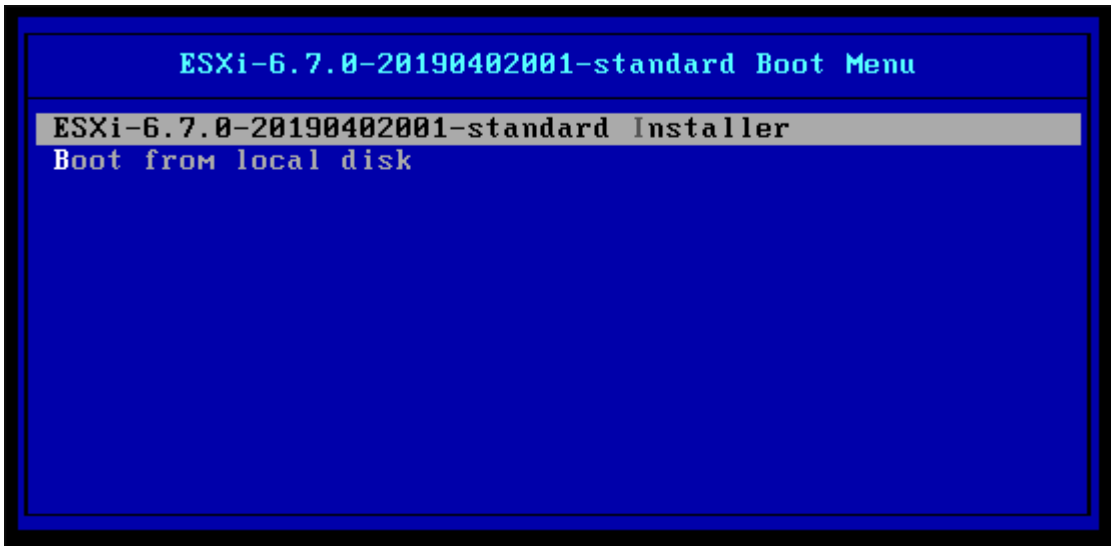
6. To install from this USB flash drive, insert it into the host using either an internal USB port or on an external USB port, then set that as the primary boot source or select from the boot menu on power up.

2.3 Installing VMware ESXi

Use the following procedure to install VMWare ESXi regardless of boot source. Select the boot media with the ESXi ISO on your host's boot menu.

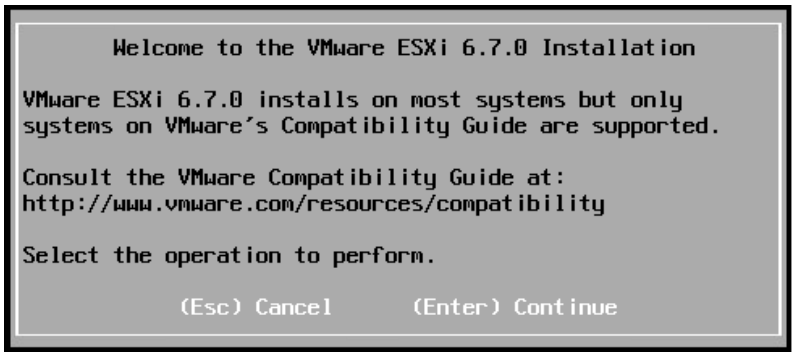
1. Apply power to start the host.

   The following menu displays when the host starts up.



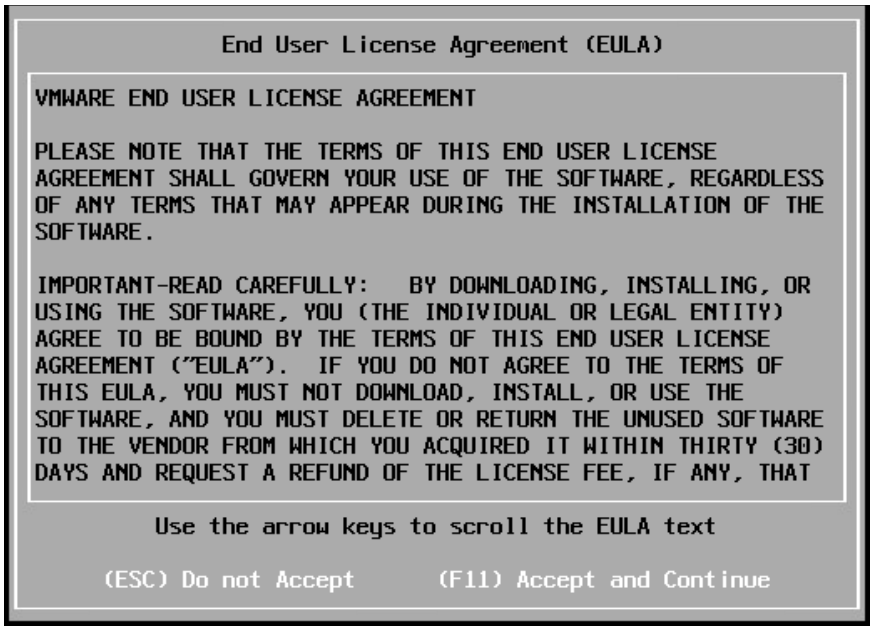
2. Select the installer using the arrow keys and then press **[ENTER]** to begin booting the ESXi installer.

   A compatibility warning is displayed.



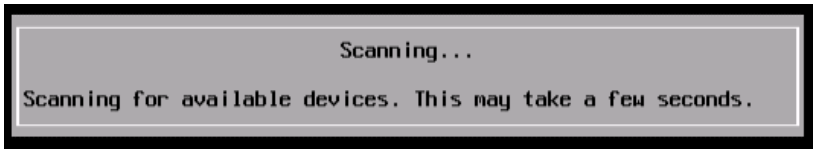
3. Press **[ENTER]** to proceed.

   The End User License Agreement (EULA) displays.

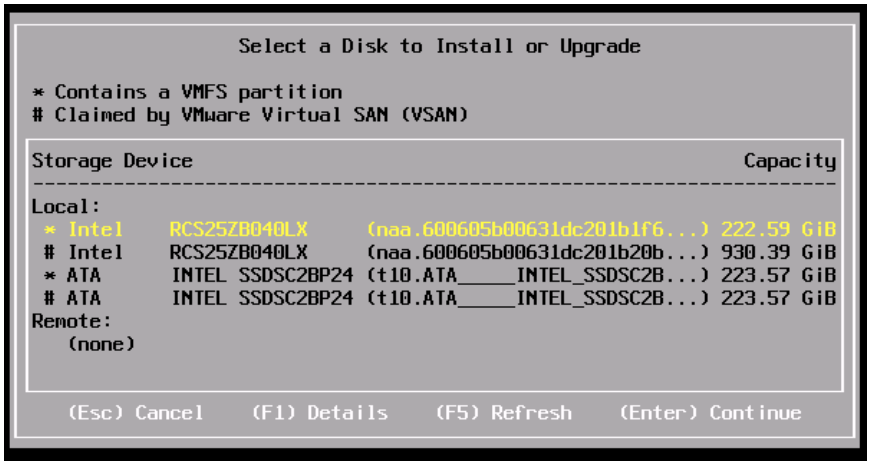


4. Read the EULA and then press **[F11]** to accept it and continue the installation.

   The installer scans the host to locate a suitable installation drive.



   It should display all drives available for install.

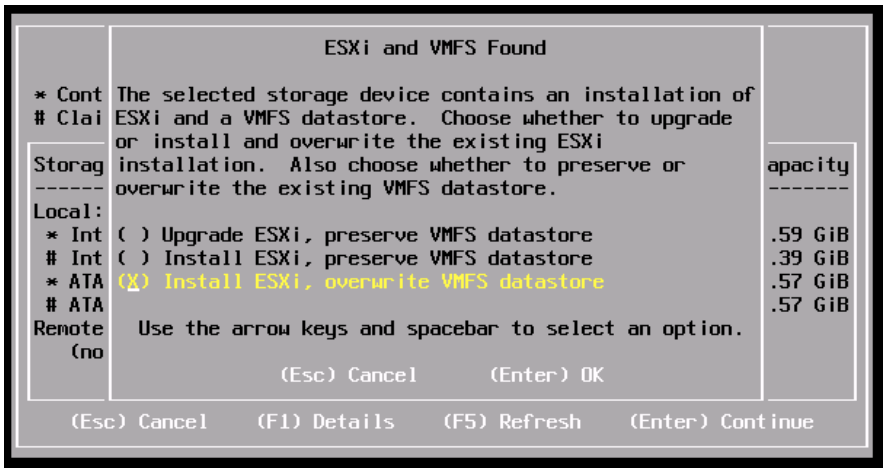


5. Use the arrow keys to select the drive you want to install ESXi and then press **[ENTER]** to continue.

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

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

The Confirm Disk Selection window displays.



6. Press **[ENTER]** to accept your selection and continue. (For this EA2 release, Upgrade ESXi is not a supported selection.)

   The **Please select a keyboard layout** window displays.

7. Select your desired keyboard layout using the arrow keys and then press **[ENTER].**

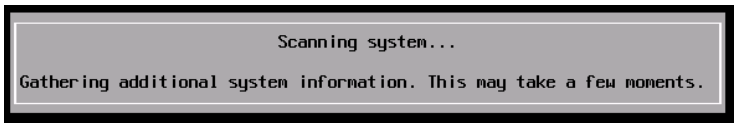
   The **Enter a root password** window displays.

8. Enter a root password in the Root password field.

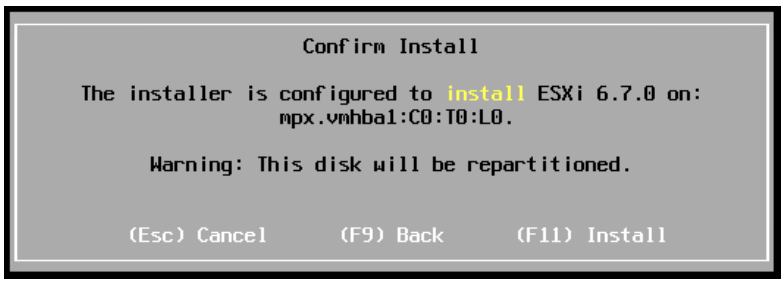
> **!** CAUTION: To prevent unauthorized access, your selected root password should contain at least eight (8) characters and consist of a mix of lowercase and capital letters, digits, and special characters.

9. Confirm the password in the **Confirm password** field and then press **[ENTER]** to proceed.

   The installer rescans the system.



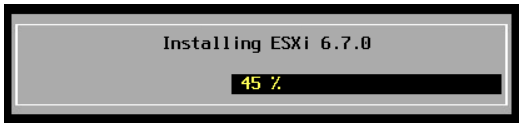
   It then displays the **Confirm Install** window.



10. Press **[F11]** to proceed with the installation.

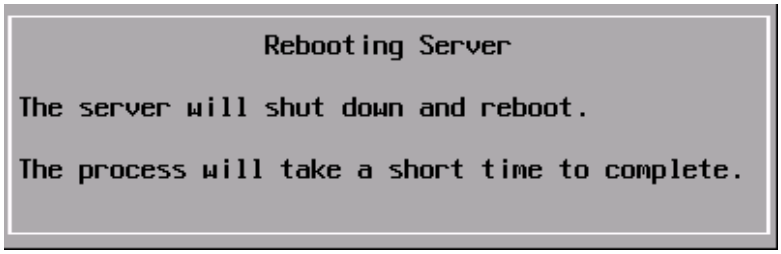
> **!** CAUTION: The installer will repartition the selected disk. All data on the selected disk will be destroyed.

   The ESXi installation proceeds.



   The **Installation Complete** window displays when the installation process is completed.

11. Press **[ENTER]** to reboot the system. (Make sure your installation media has been ejected and your BIOS set to the boot disk.)



12. The installation is now complete.

2.4 Initial Host Configuration

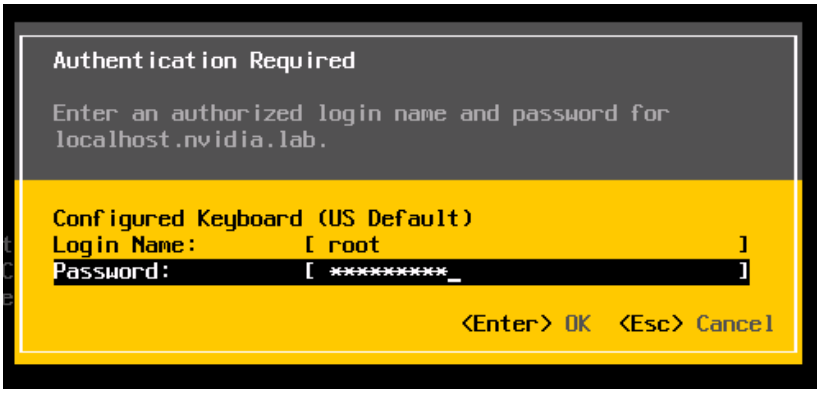
A countdown timer displays when you first boot ESXi. You can wait for the countdown to expire or press **[ENTER]** to proceed with booting. A series of notifications displays during boot.

The VMware ESXi screen displays when the boot completes.

Use the following procedure to configure the host:

1. Press **[F2]**.

   The **Authentication Required** window displays.

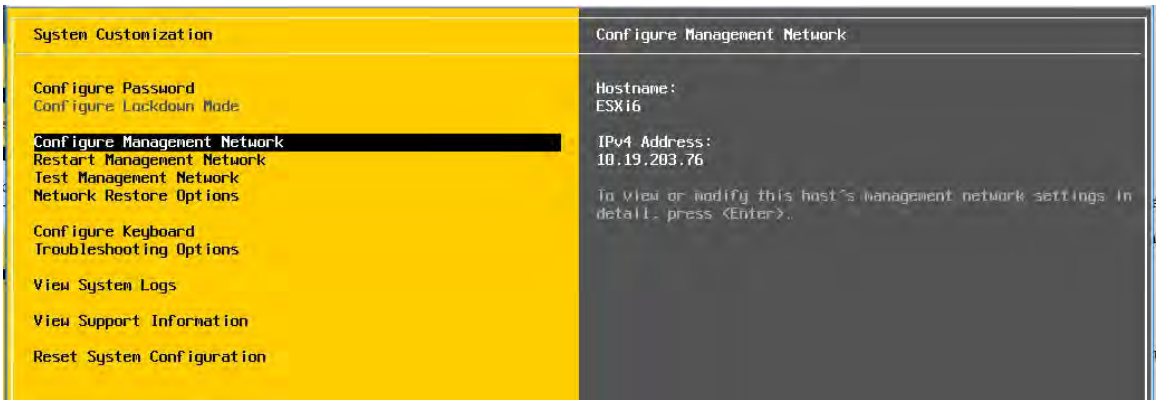


2. Enter the root account credentials that you created during the installation process and then press **[ENTER].**

   The **System Customization** screen displays.

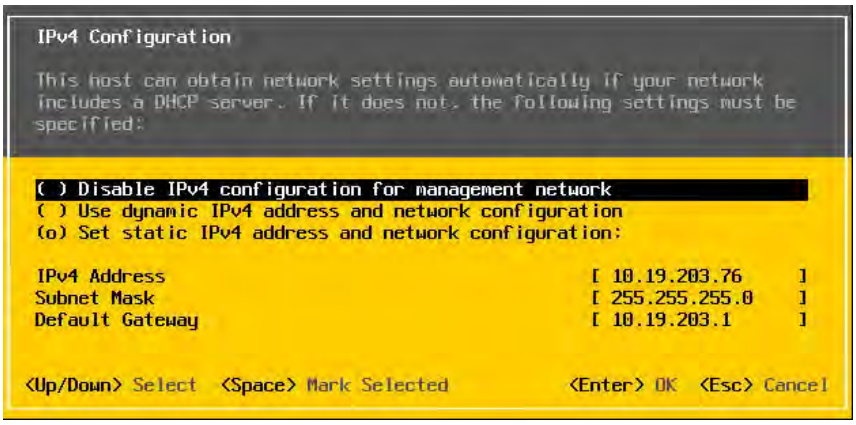
3. Scroll down to select **Configure Management Network** and then press **[ENTER].**

   The **Network Adapters** window appears.



4. Use the arrow keys to select the adapter to use as the default management network and then press **[ENTER].**

   The IPv4 Configuration window displays.



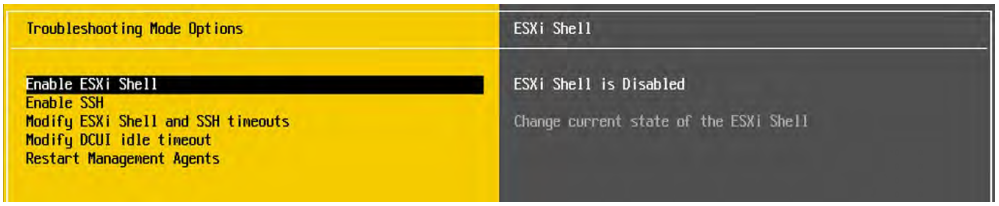
5. Use the arrow keys to select **Set static IPv4 address and network configuration** and then enter the IPv4 address, subnet mask, and default gateway in the respective fields.
6. Press **[ENTER]** when finished to apply the new management network settings.

   The Confirm Management Network popup displays.
7. Press **[Y]** to confirm your selection.

The **DNS Configuration** window displays.

8. Add the primary and (if available) secondary DNS server address(es) in the respective fields.
9. Set the host name for this ESXi host in the **Hostname** field.
10. Press **[ENTER]** when finished.
11. Select **Test Management Network** on the main ESXi screen to open the **Test Management Network** window.
12. Perform the following tests:
    a) Ping the default gateway.
    b) Ping the DNS server.
    c) Resolve a known address.
13. Return to the main ESXi screen when you have completed testing, and then select **Troubleshooting Options**.

    The Troubleshooting Mode Options window displays.



To install the NVDIA VIB in a later step, you will need to enable the ESXi shell. This can be accomplished by selecting **Enable ESXi Shell**.

14. Press **[ENTER]** to toggle **Enable ESXi Shell** on.

    The window on the right displays the status: **Enable ESXi Shell Disabled.**

15. Enable SSH by selecting **Enable SSH** and press **[ENTER]** to toggle this option on.

    The window on the right displays the status: **SSH is Enabled.**

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 prerequisites in General Prerequisites 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 6.7 host and perform configuration changes from a single pane of glass. Since the OS was developed by VMware, 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, and resulting in significant optimization for vSphere environments, and quicker release of security patches and bug fixes.

The VCSA scales up to 2000 hosts and 35,000 virtual machines. 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. Features such as Update Manager are bundled into the VCSA, as are file-based backup and restore and vCenter High Availability. The appliance also saves operating system license costs and is quicker and easier to deploy and patch.

- Software Considerations:
  - VCSA must be deployed to an ESXi host running version 5.5 or above. However, all hosts you intend to connect to VCSA should be running ESXi 6.0 or above. Hosts running v5.5 and earlier cannot be managed by the VCSA and do not have a direct upgrade path.
  - You must check compatibility of any third-party products and plugins that may be used for backup, virus protection, monitoring, etc., as they may need upgrading for ESXi compatibility.

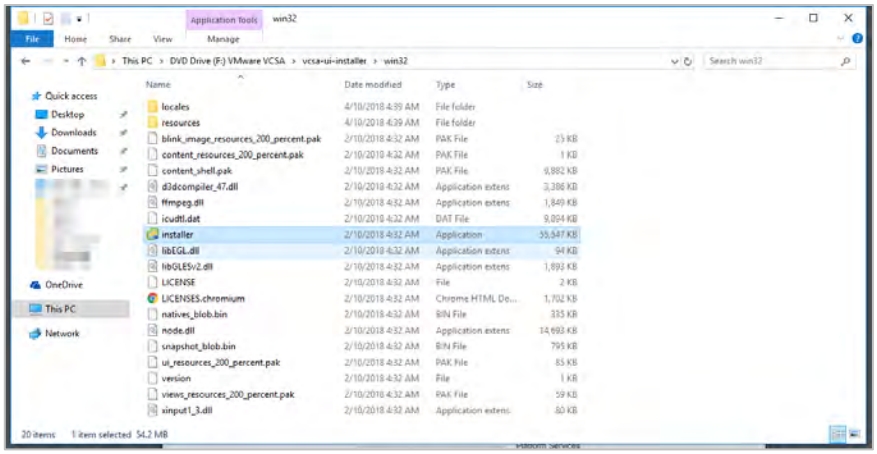
- To check version compatibility with other VMware products, see the Product Interoperability Matrix.

- Architectural Considerations:
  - When you implement a new vSphere environment, you must plan its topology in accordance with the VMware vCenter Server and Platform Services Controller Deployment Types.
  - Most deployments include vCenter Server and Platform Service Controller in one appliance, following the embedded deployment model, which is used in this guide.
- Other Considerations:
  - The VCSA with embedded PSC requires the following hardware resources (disk can be thin provisioned):
    - Tiny (up to 10 hosts, 100 VMs): 2 CPUs, 10 GB RAM.
    - Small (up to 100 hosts, 1000 VMs): 4 CPUs, 16 GB RAM.
    - Medium (up to 400 hosts, 4000 VMs): 8 CPUs, 24 GB RAM.
    - Large (up to 1000 hosts, 10,000 VMs): 16 CPUs, 32 GB RAM.
    - X-Large (up to 2000 hosts, 35,000 VMs): 24 CPUs, 48 GB RAM; new in v6.5.
  - Storage requirements for the smallest environments start at 250 GB and increase depending on your specific database requirements. See the document Storage Requirements for further details.
  - If the PSC is deployed as a separate appliance it requires two CPUs, 4 GB of RAM, and 60 GB of disk storage.
  - Environments with ESXi host(s) that have more than 512 LUNs and 2048 paths should be sized large or x-large.
  - The ESXi host on which you deploy the VCSA 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 be unable to start if the clocks are not synchronized.
  - FQDN resolution must be enabled when you deploy vCenter Server.
  - Required Ports for vCenter Server and Platform Services Controller.
  - vSphere VMware Configuration Maximums.

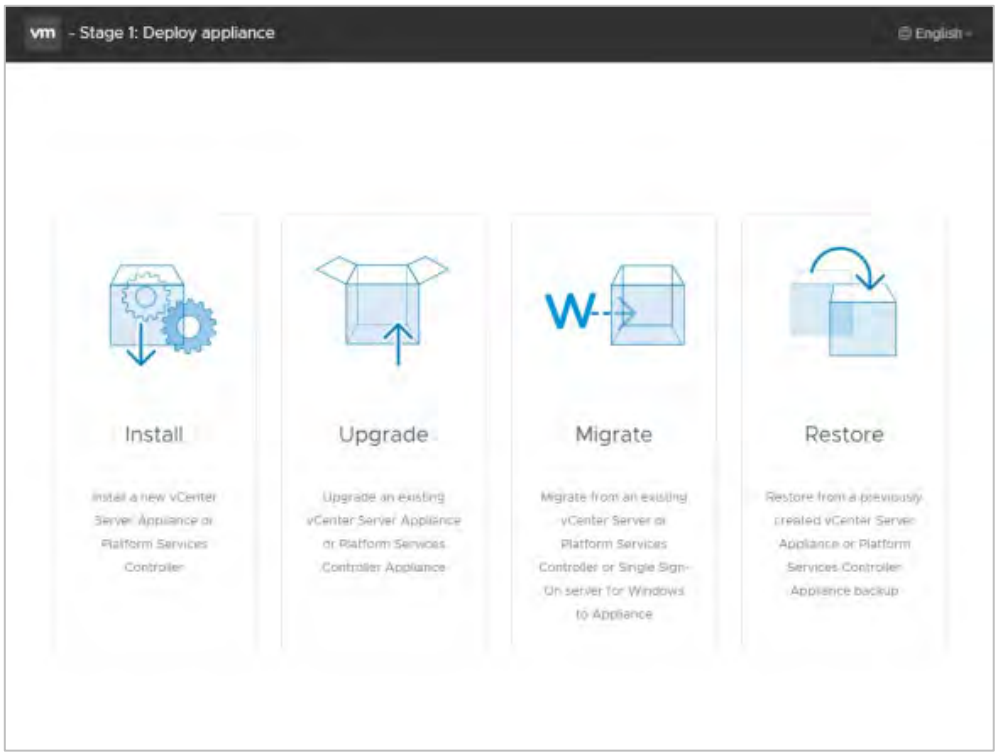
3.1.2 vCenter Server Appliance (VCSA) Installation

Download the VMware vCenter Server Appliance ISO from VMware downloads: v6.7.0.

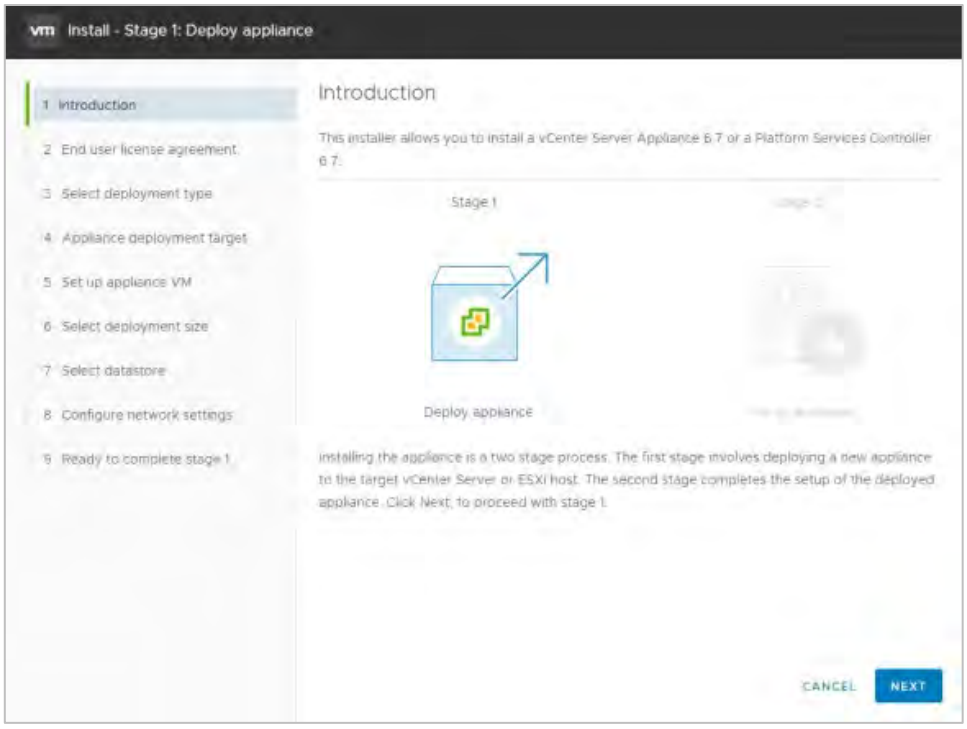
1. Mount the ISO on your computer. The VCSA installer is compatible with Mac, Linux, and Windows.
2. Browse to the corresponding directory for your operating system, e.g. `\vcsa-ui-installer\win32`. Right click Installer and select Run as administrator.



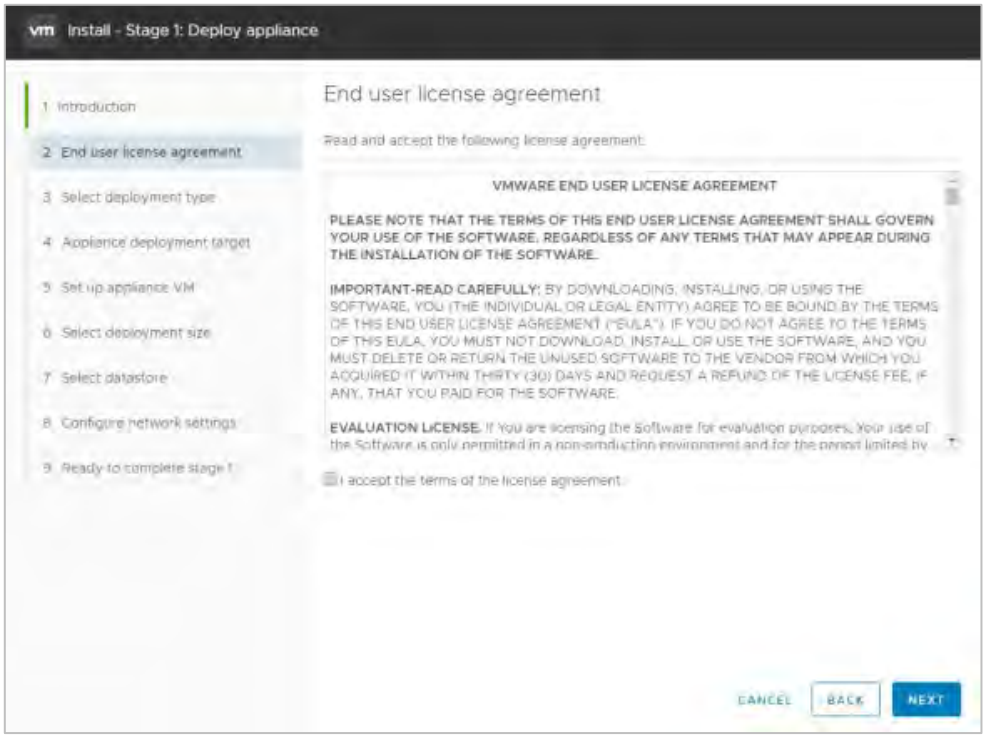
3. As you are installing a new instance, click Install.



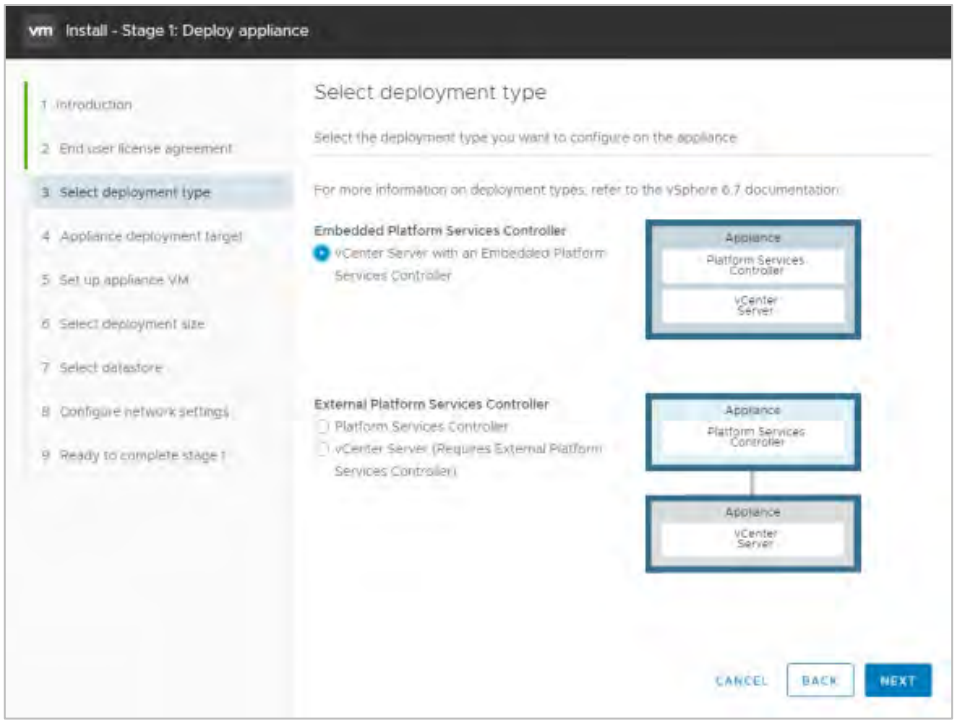
4. In Stage 1 of the install process, the installer deploys the appliance. Click Next to begin:



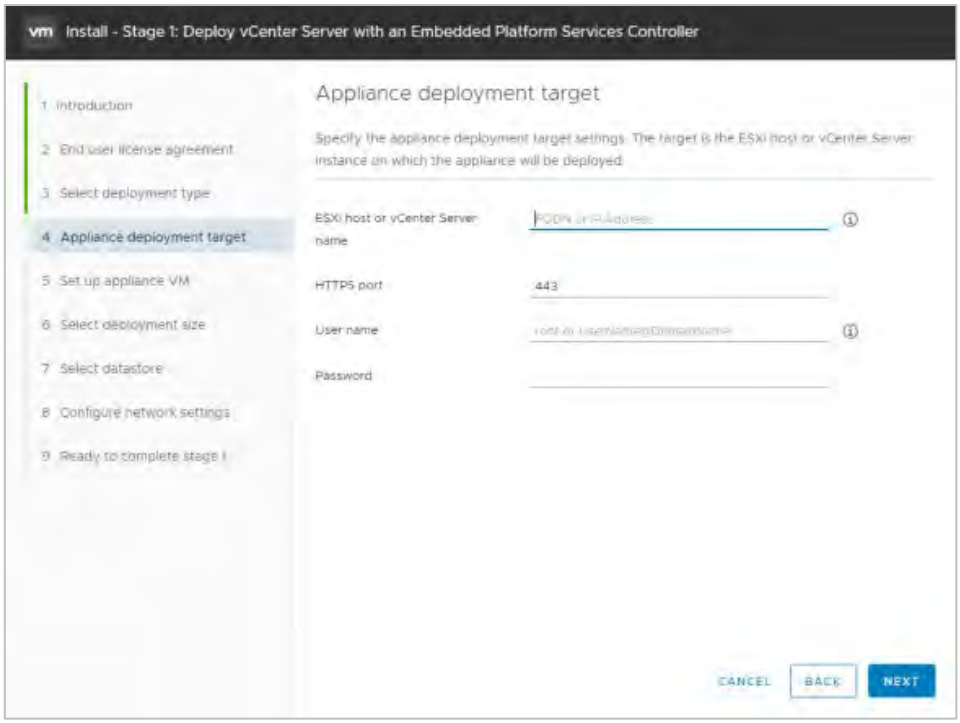
5. Read the EULA, then click Next to continue:



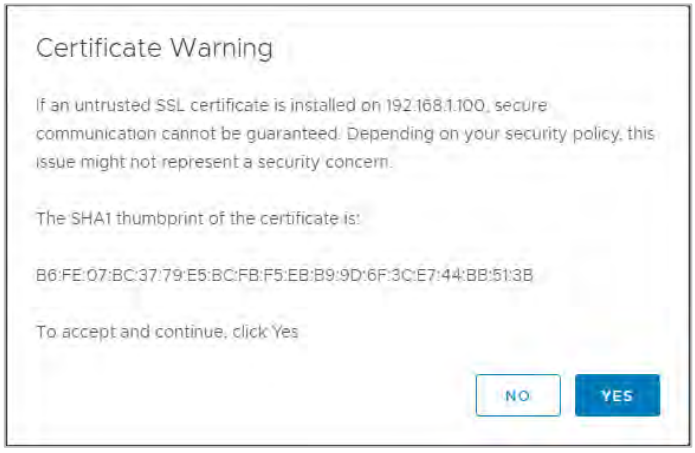
6. Select a deployment type. This example uses an embedded deployment combining the vCenter Server and Platform Services Controller in one appliance. Select Embedded Platform Services Controller, then click Next:



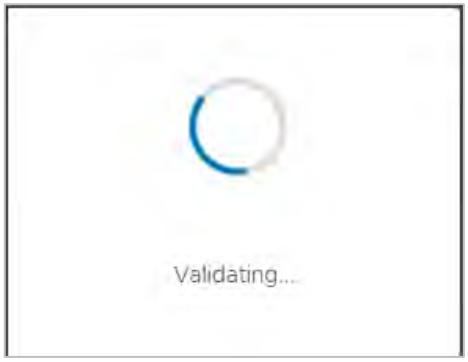
7. Select the ESXi host on which to install the VCSA as a guest. This must be a host that runs ESXi 5.5 or later. NVIDIA recommends that the vCenter server (Windows or appliance-based) run on a separate management cluster from the one designated for VDI workloads. Enter the IP address or fully qualified domain name (FQDN) of the chosen host, then its root username and password., and click Next:



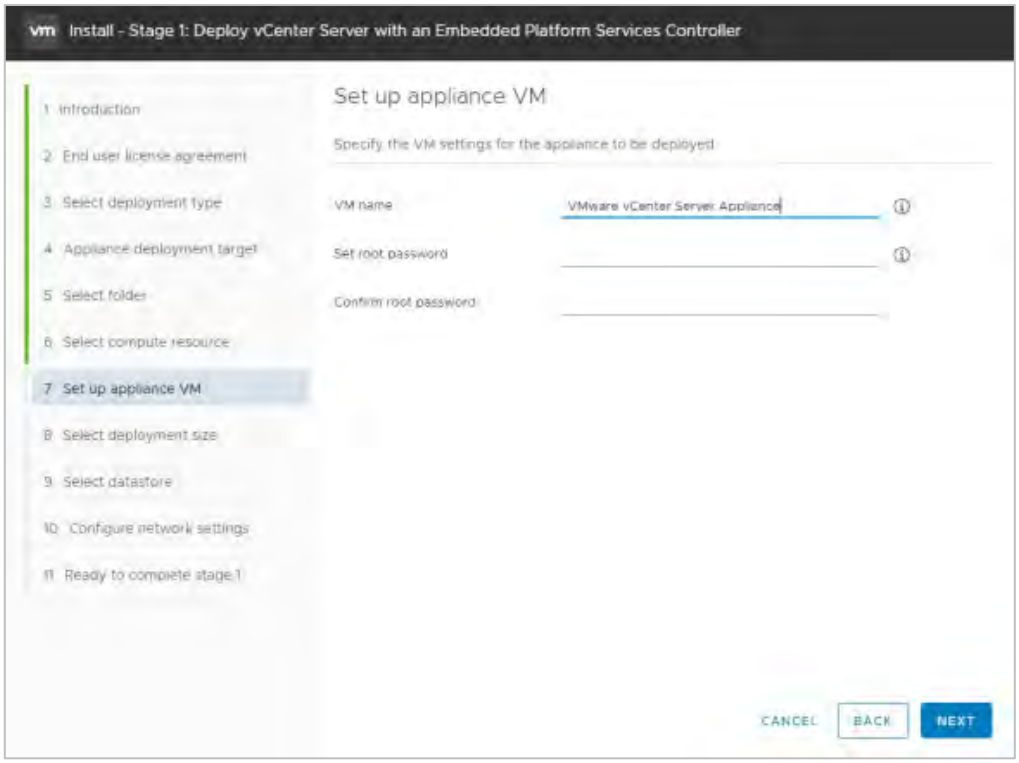
8. If your computer running the installer can access the host, the installer may display a certificate warning as it connects. This happens if the host is using a self-signed cert. If the host uses a signed certificate, this warning does not appear. If you get the warning, click Yes to continue:



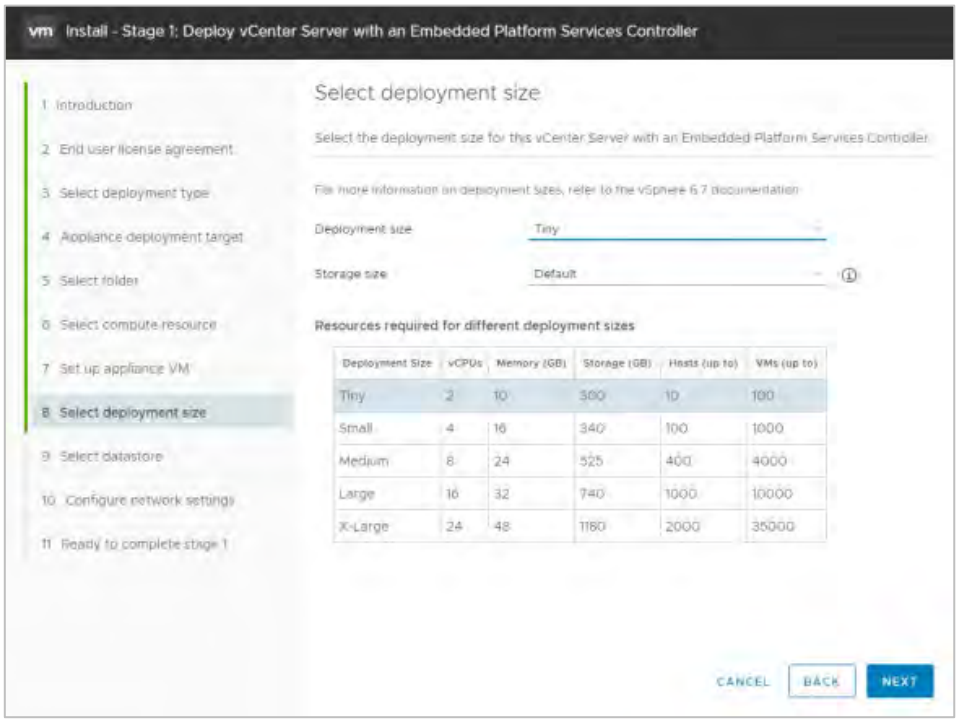
9. The installer validates the credentials you provided:



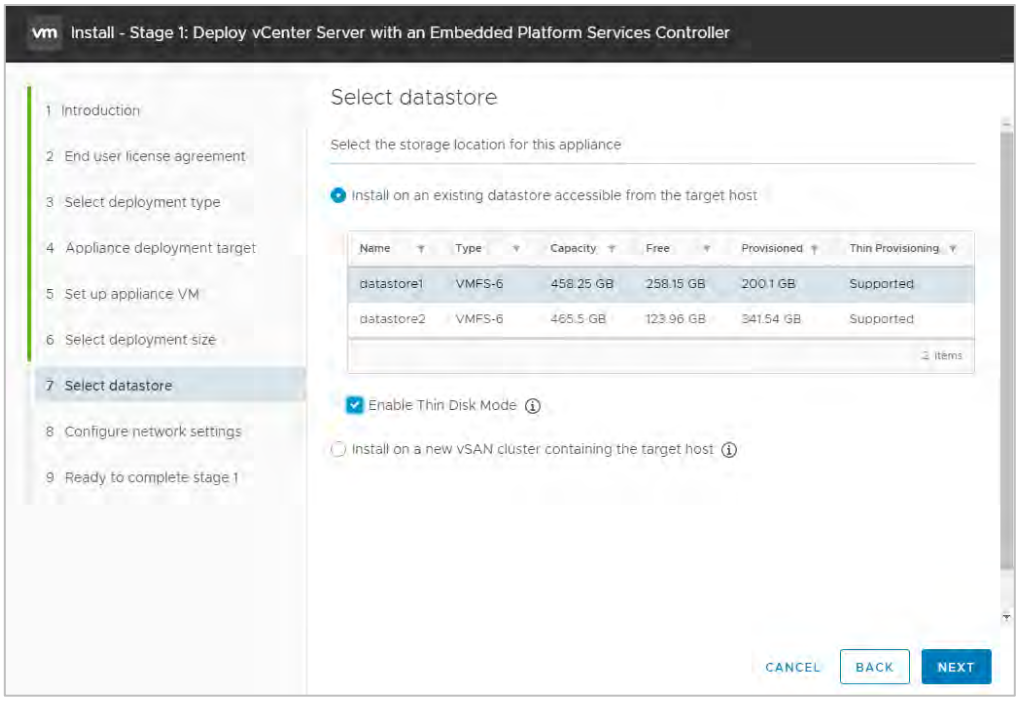
10. If the installer connects to the host successfully it prompts you to name the appliance, enter a root password for the appliance (twice, to check for typing errors), and click Next:



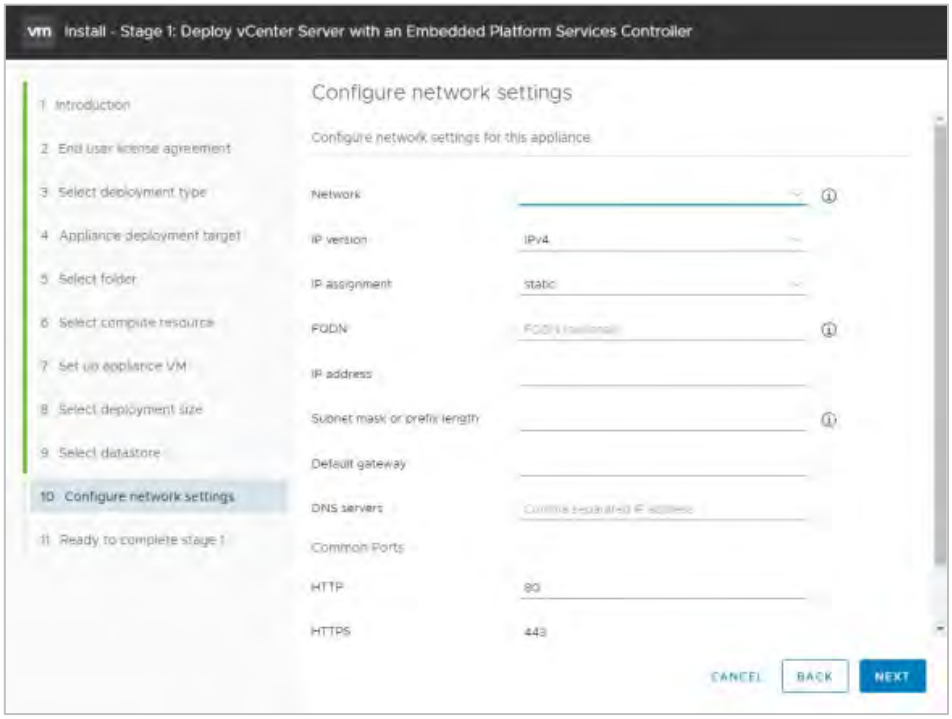
11. Select a deployment size appropriate to the number of hosts and virtual machines that that vCenter Server will manage, then click Next:



12. Select the datastore where the VCSA is to be deployed’; select thin provisioning if required, then click Next. Configure the network settings for the appliance and click Next.

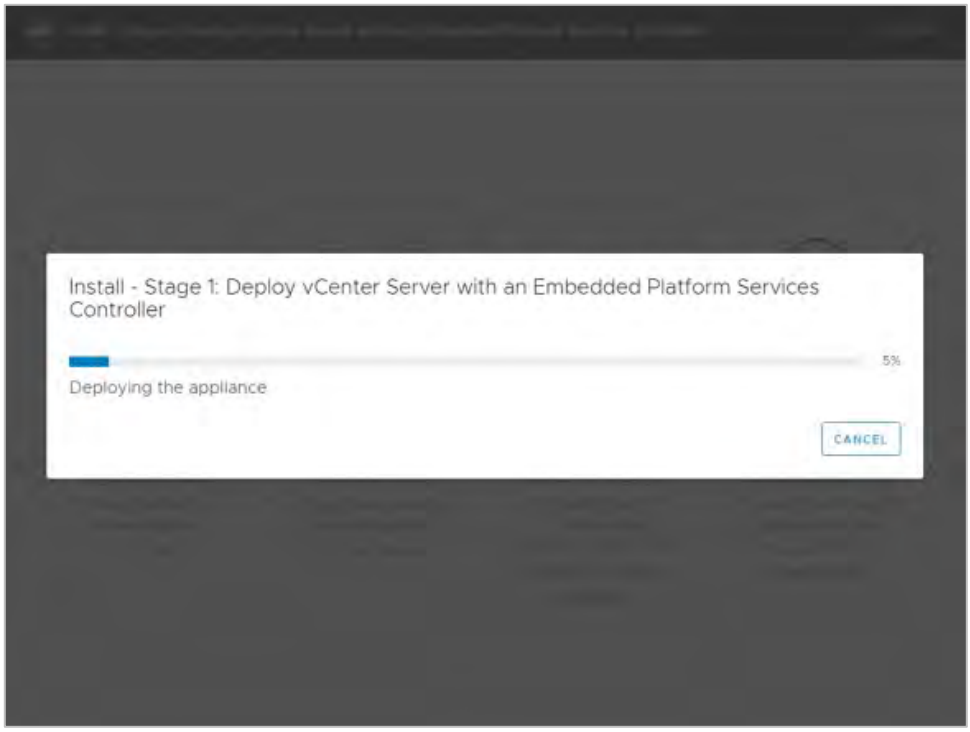


13. The installer displays a page for configuring network settings. This is a long page which you must scroll to display all of the settings.

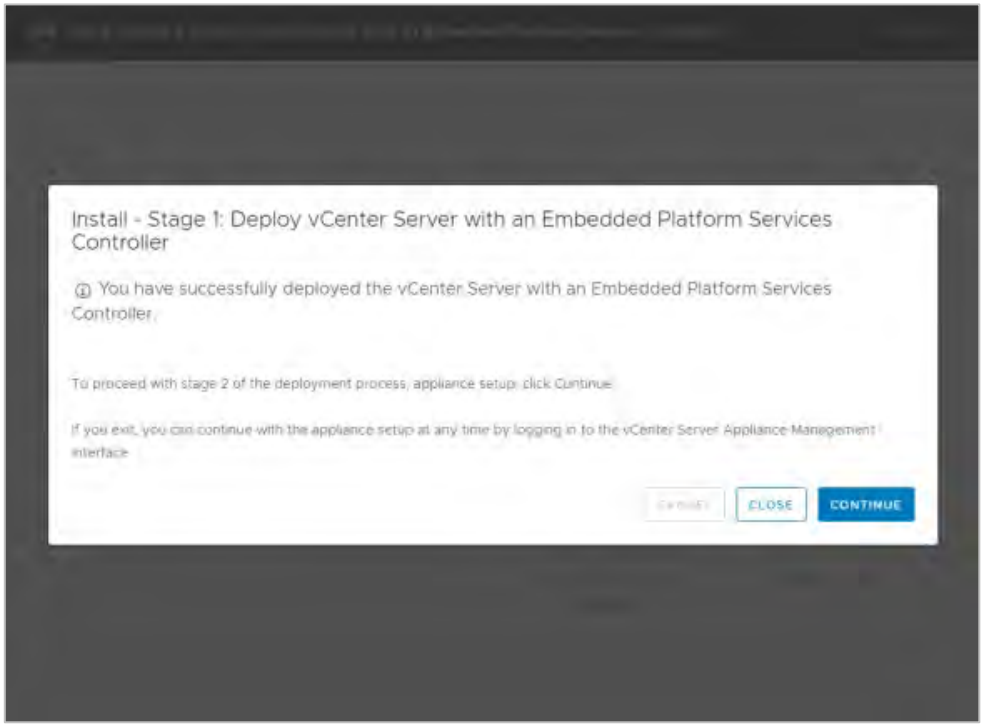


Before you configure these settings, choose an appropriate static IP address and enter it into local DNS (e.g. on the Domain Controller). Once you can resolve the address, enter the IP address its host name on the network setting page, then scroll down and enter remaining items. When the settings are complete, click Next.

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

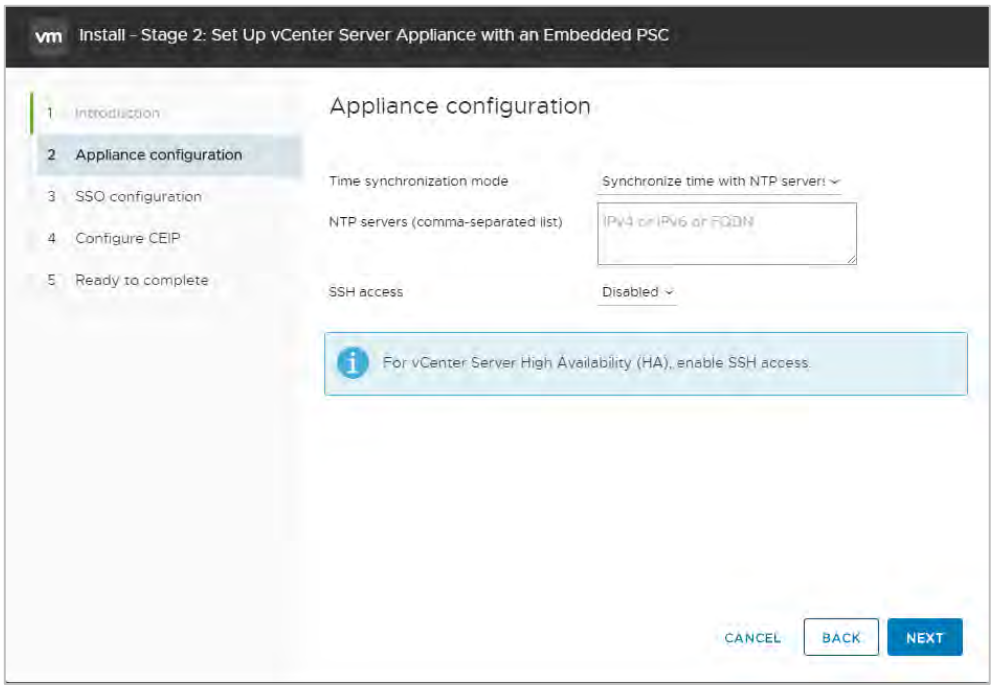


15. The VCSA is now deployed. Click Continue to proceed to the install process's Stage 2, setting up the VCSA.



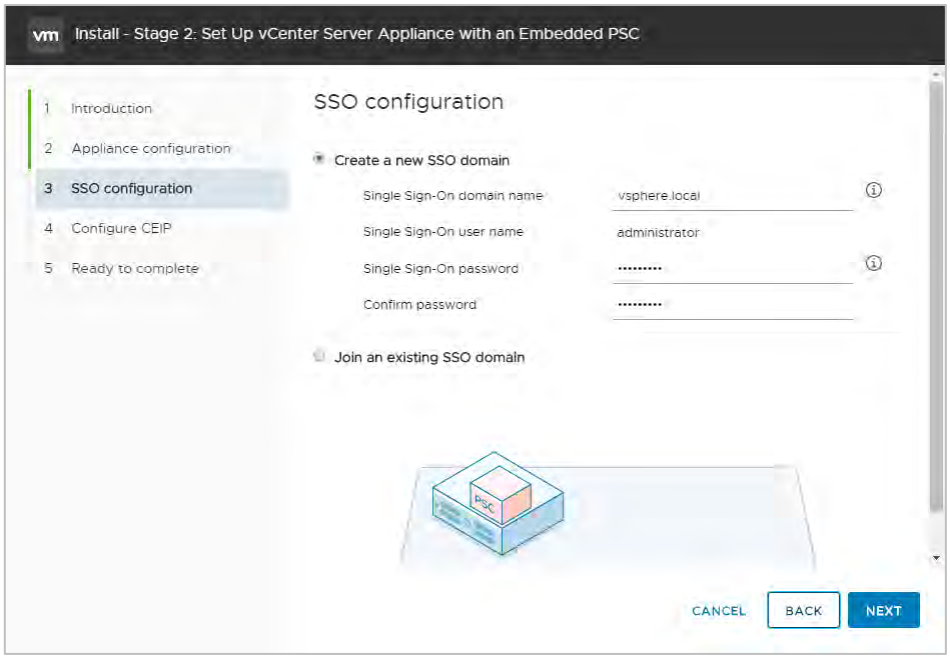
16. Click Next to begin the VCSA setup.

17. Configure the NTP servers and enable SSH access if required, then click Next.

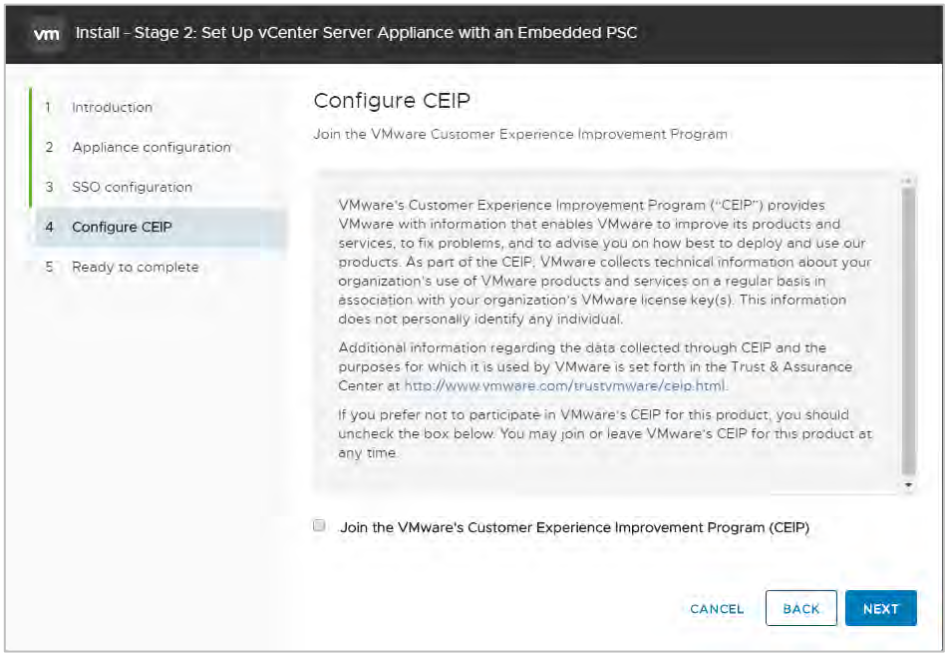


18. Enter a unique SSO domain name. The default name is `vSphere.local`. Configure a password for the SSO administrator, then click Next.

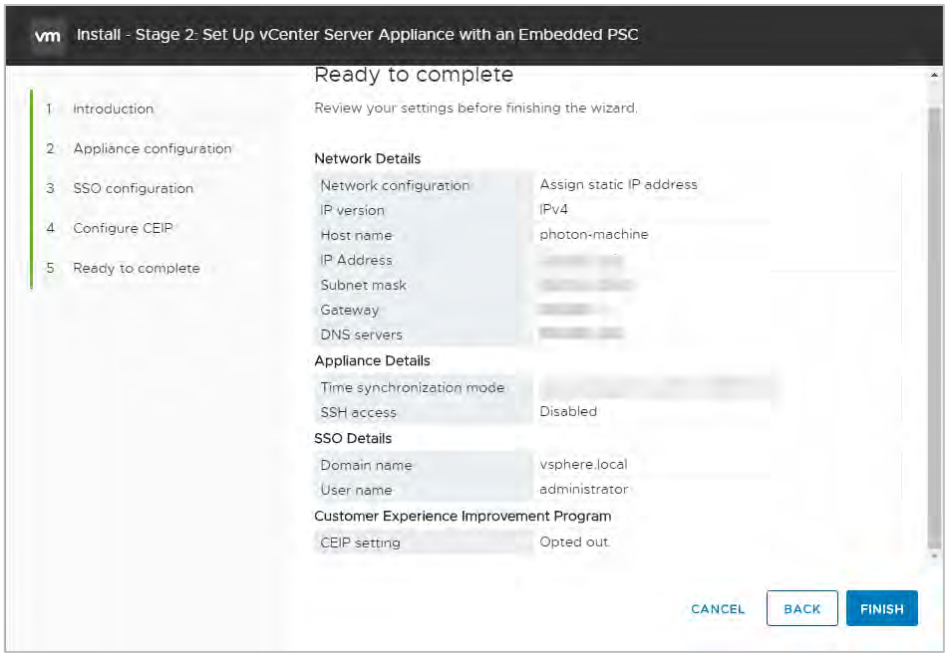
Note: Do not make the SSO domain name the same as your Active Directory Domain.



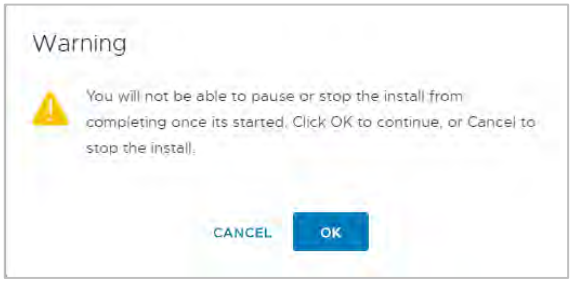
19. Check or clear the checkbox that opts into the VMware Customer Experience Improvement Program, then click Next:



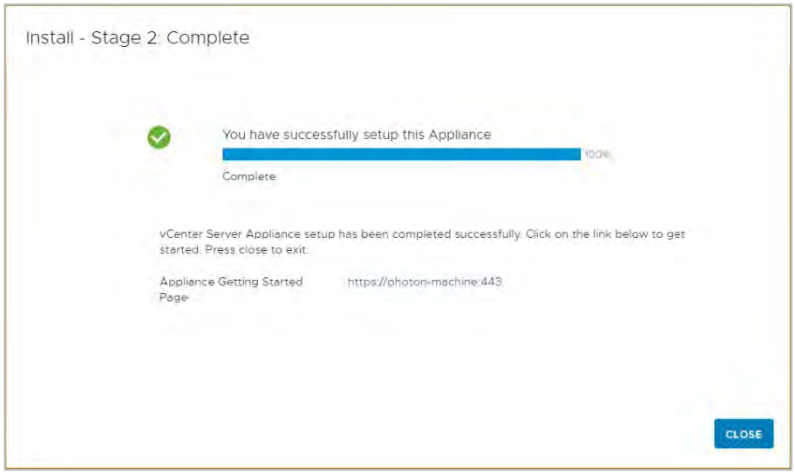
20. The installer displays a summary page. Review the details on this page, then click Finish.



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



22. When the install process is complete, click Close to close the installer:



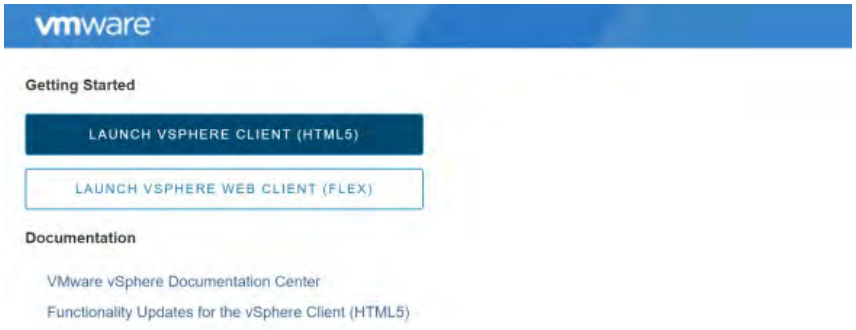
3.2 Post Installation

This section describes post install and configure vCenter Server.

3.2.1 Adding Licenses to Your vCenter Server

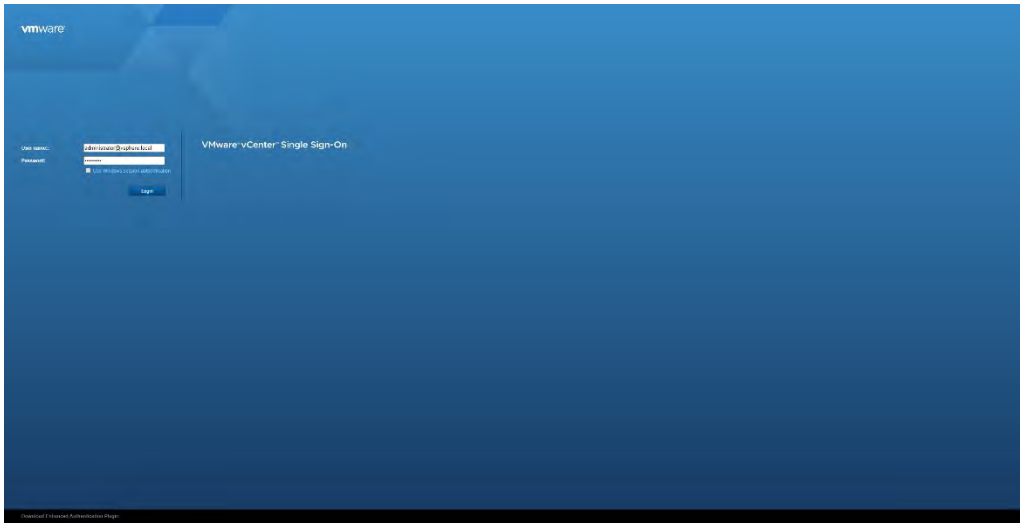
Use the following procedure to configure vCenter:

1. Connect to the vCenter post install using the IP or FQDN of the vCenter. Access vSphere by clicking either **Launch vSphere Client (HTML5)** or **Launch vSphere Web Client (FLEX)**. As the web client will be depreciated in future versions, and the HTML5 client is now nearly at full feature parity, we will use the HTML5 vSphere client.



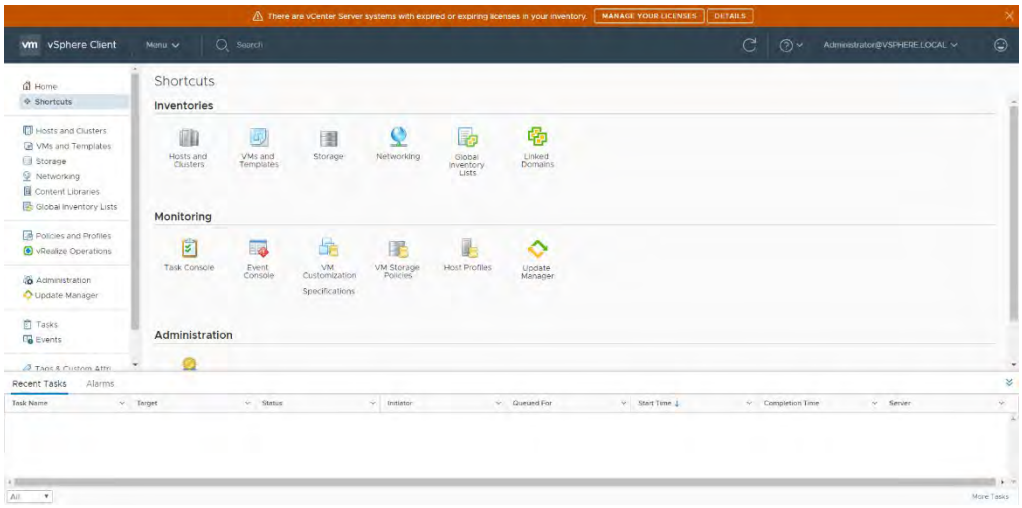
Note: NVIDIA recommends selecting the HTML5 client. The web client will be depreciated in a future version, and the HTML5 client is at nearly full feature parity.

The *VMware Single Single-On* page displays.

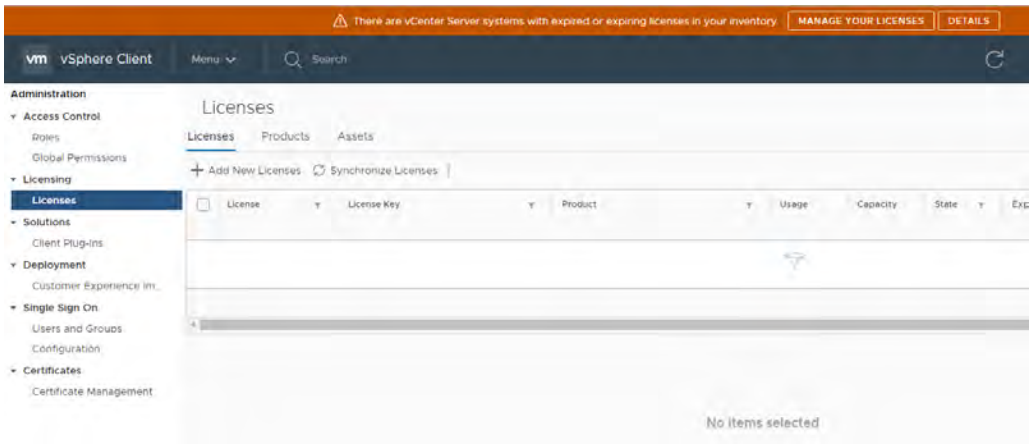


2. Enter the username and password that you specified during installation, and then click the **Login** button.

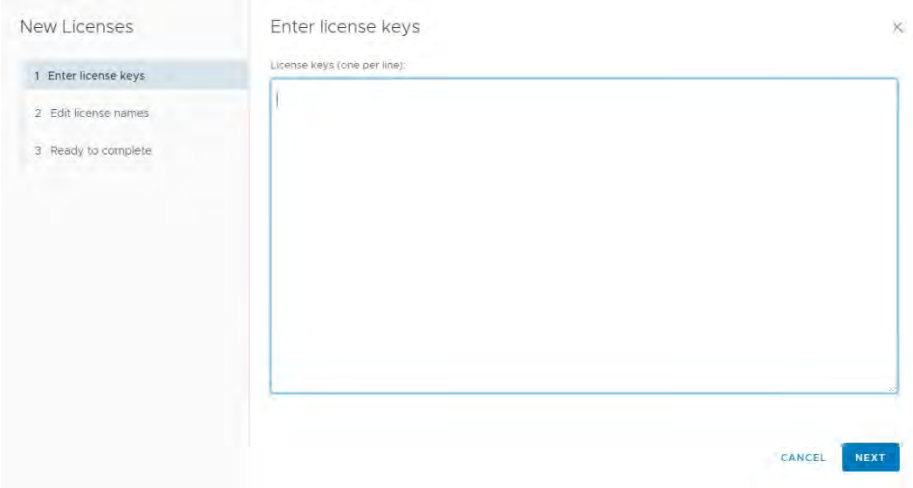
The VMware vSphere Web Client page displays.



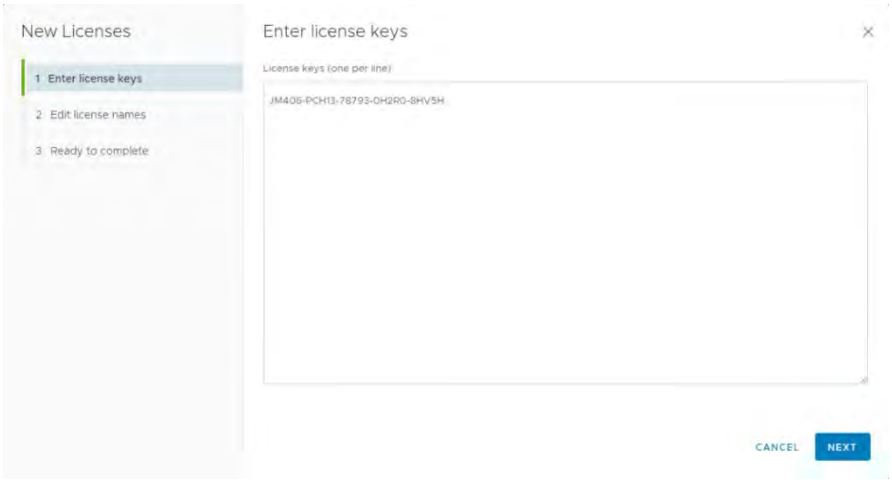
3. You must apply a new vCenter license key within 60 days. If you have purchased vCenter Server then log into your licensing portal here. If the license key does not appear then check with your VMware account manager. Log in to the vSphere Web Client using the SSO administrator login. From the **Menu** drop-down click **Administration**.



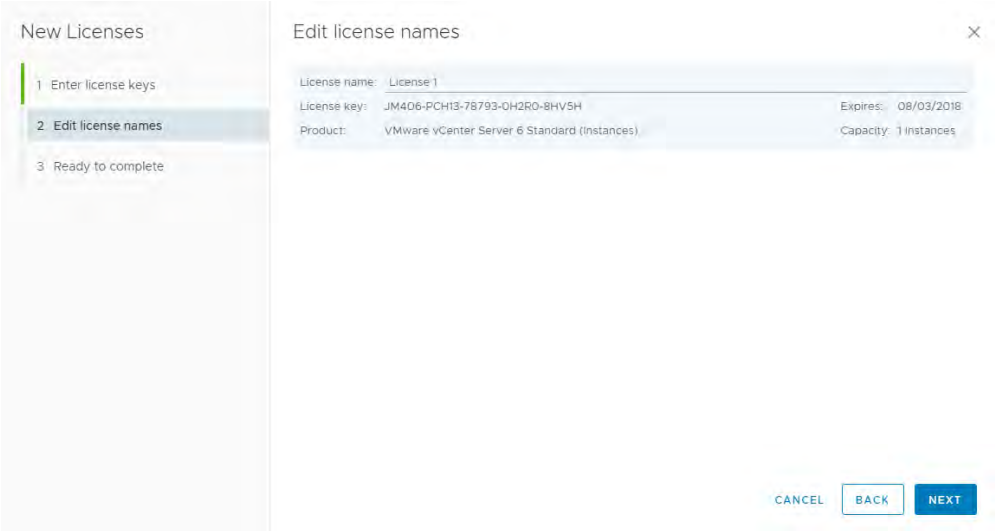
4. Select Licenses from the left-hand menu and then select the Licenses tab to open the Licenses tab. Click **Add New Licenses** to open the New Licenses popup.



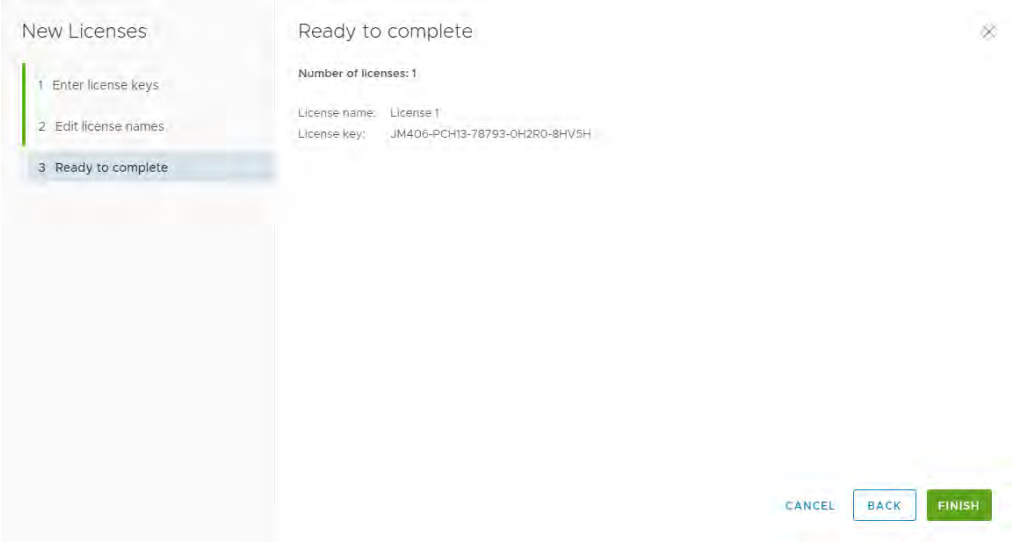
5. Enter the vCenter Server Standard license key provided at the vSphere beta program website.



6. Enter a unique name for the license in the License Name field and then click **Next**.



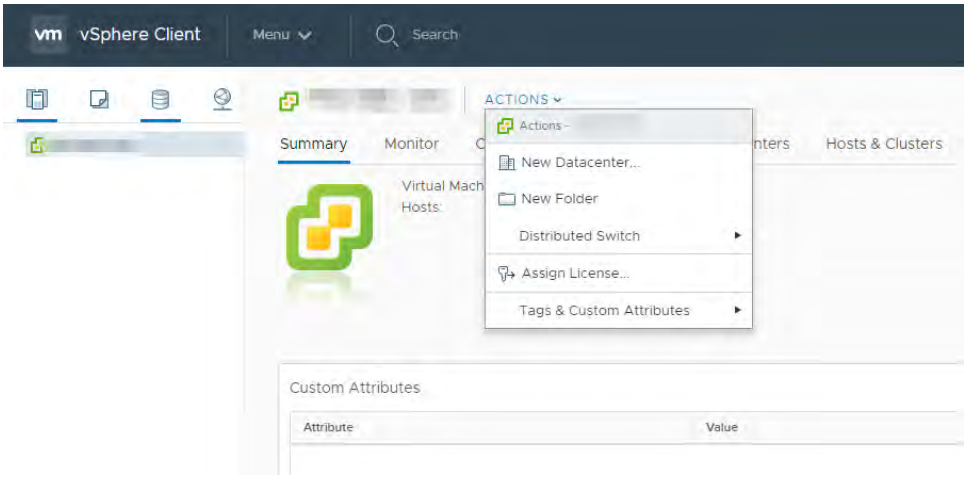
7. Review your selections and then click **Finish** to close the Enter New License popup and return to the VMware vSphere Web Client page.



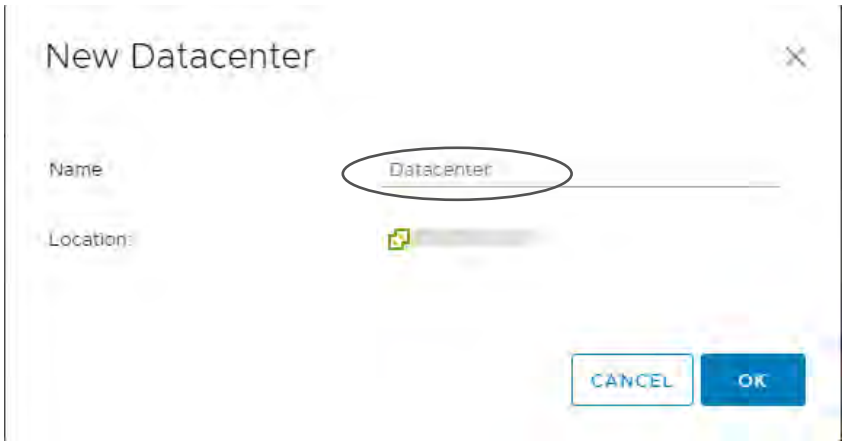
3.2.2 Adding a Host

Use the following procedure to add a host in vCenter:

1. Select the **Home** icon (house) on the *VMware vSphere Web Client* page.
2. Select **Hosts** and **Clusters**.
3. From the **ACTIONS** drop-down list, select **New Datacenter.**

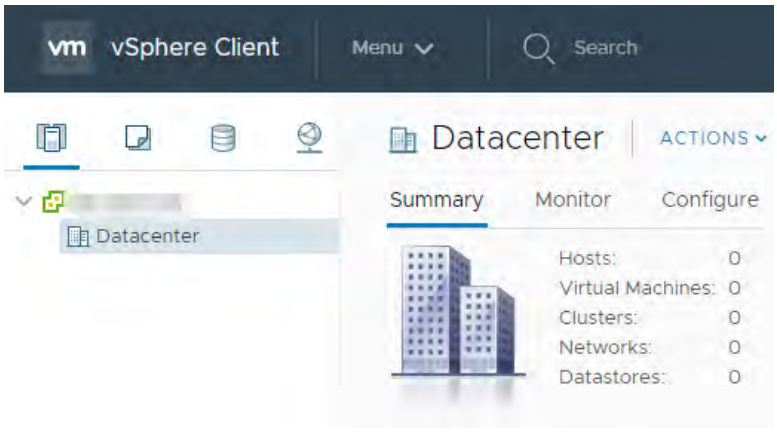


The New Datacenter popup displays.



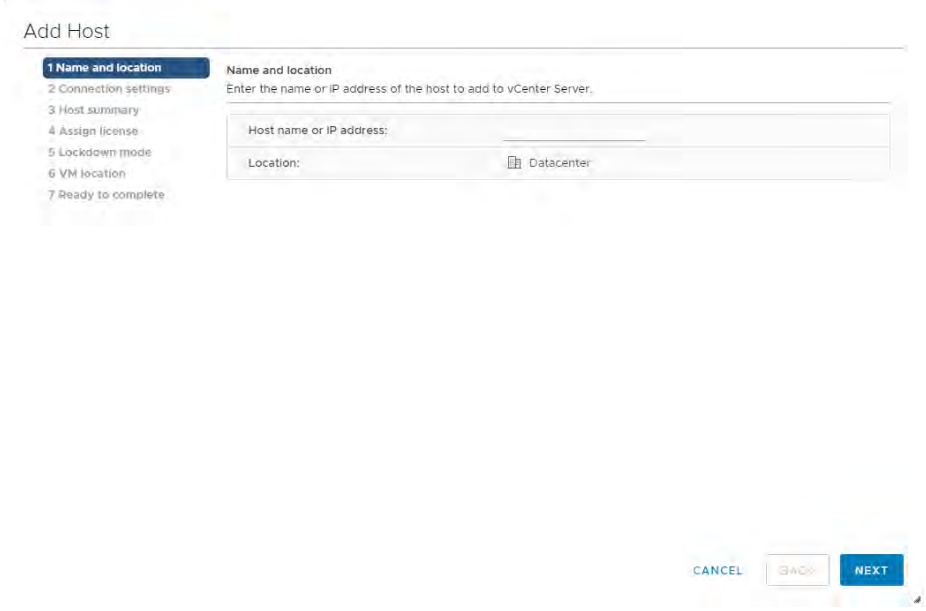
4. Enter a name for the datacenter in the **Datacenter name** field and click **OK**.

   The new datacenter is visible in the left panel of the *vSphere Web Client*.



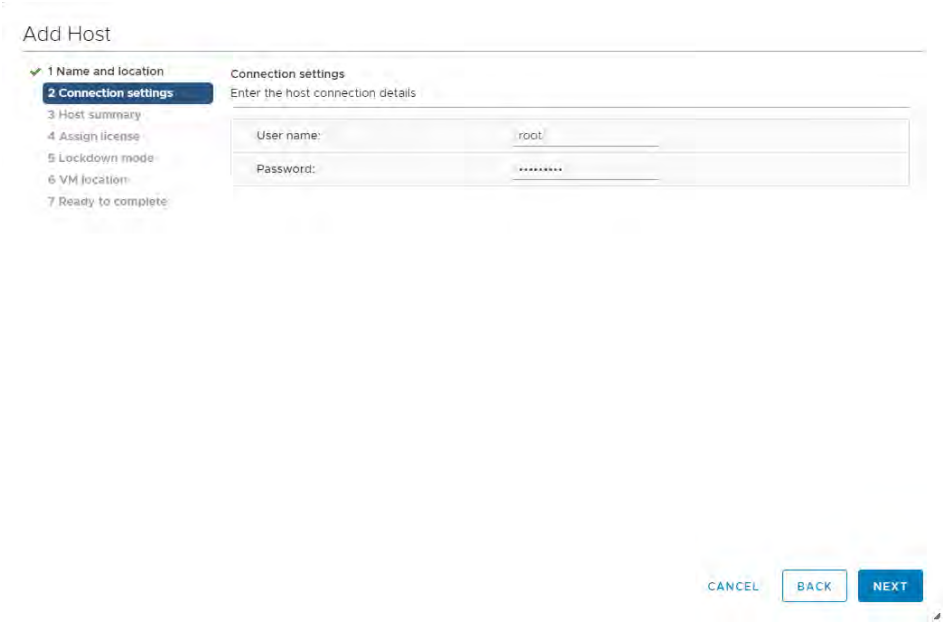
5. Select the **Datacenter**, go to drop down **Actions**, and select **Add a Host**.

   The *Name and location* dialog box opens.



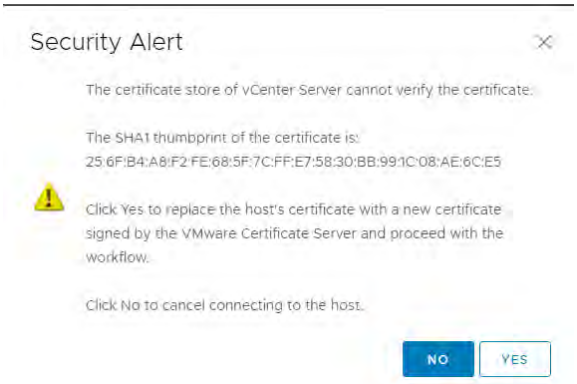
6. Enter the host name or IP address of the vSphere host and click **Next**.

   The ***Connection settings*** dialog box displays.



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

   The *Security Alert* popup displays.



8. Click **Yes** to replace the host certificate.

   The *Host summary* dialog displays.

9. Review the settings and click **Next** to proceed.

   The *Assign license* dialog displays.

10. Confirm the license selection and click **Next**.

    The *Lockdown mode* dialog displays.

11. Accept the default setting (Disabled) and click **Next**.

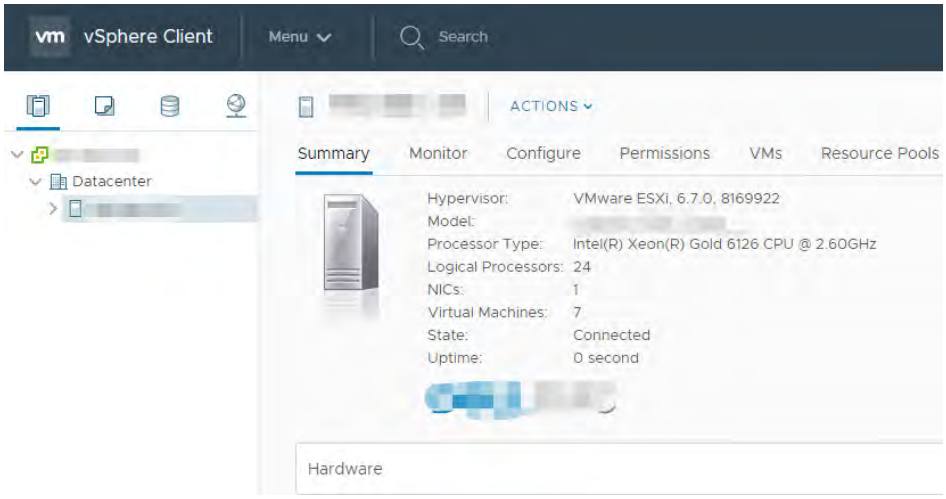
    The *VM location* dialog displays.

12. Select a cluster or accept the default option and click **Next** to proceed.

The *Ready to complete* dialog displays.

13. Click **Finish** to complete adding the new host.

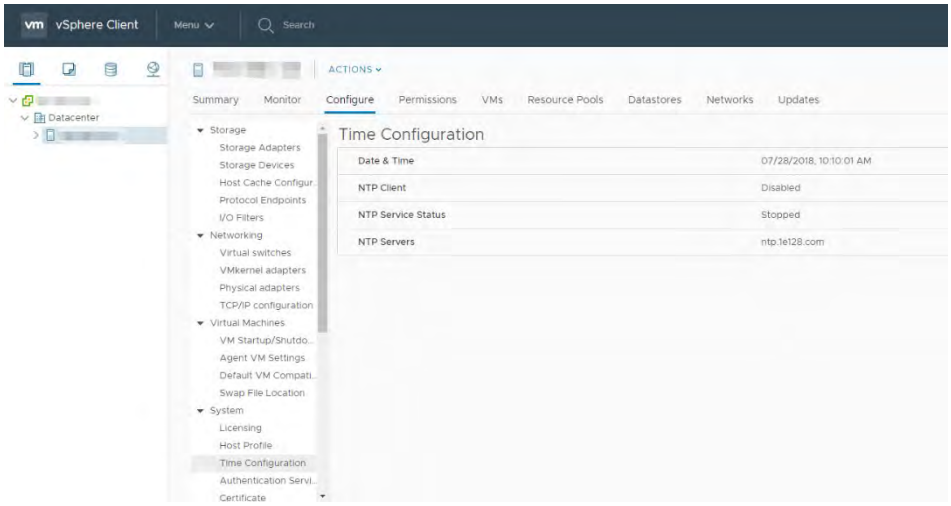
The new host is now visible in the left panel when you click the datacenter name.



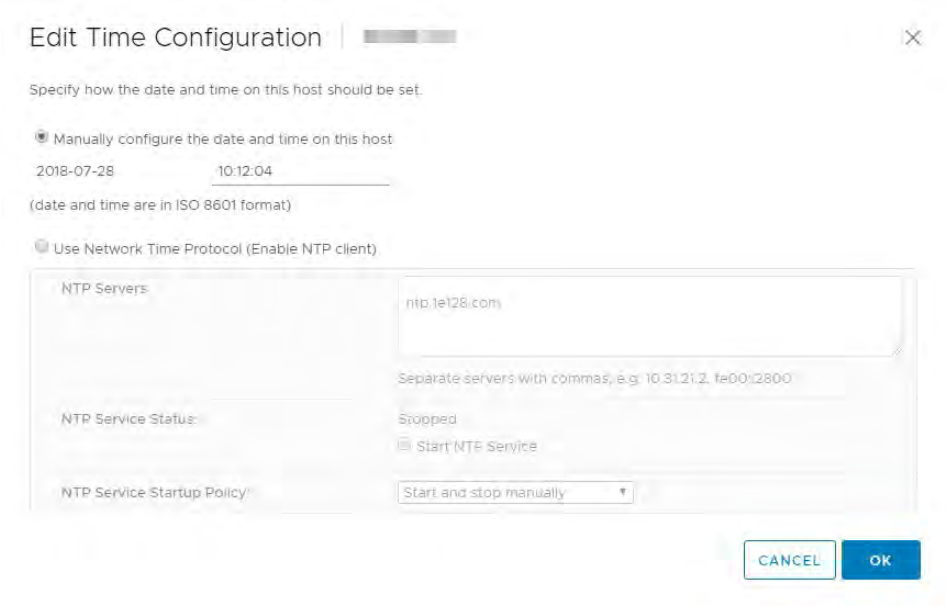
3.2.3 Setting the NTP Service on a Host

Set the NTP service on each host to ensure time is accurate for all guests.

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



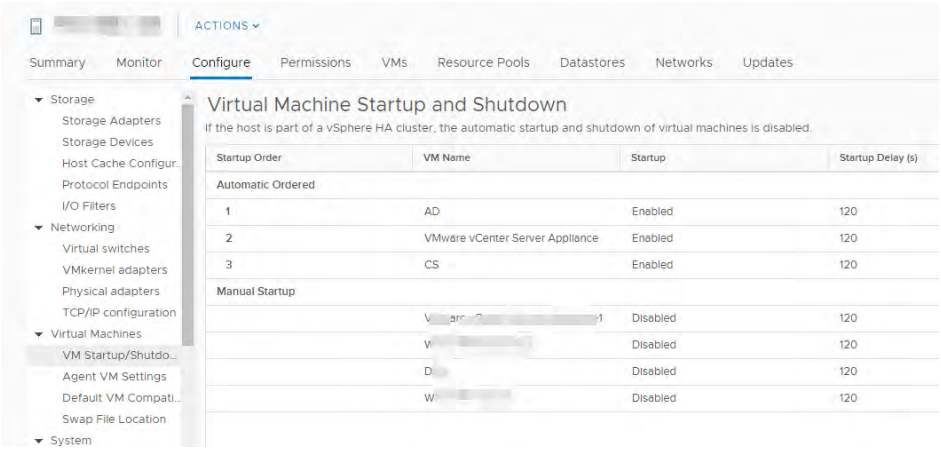
2. Enter a valid time server and click **OK**.



3.2.4 Setting a vCenter Appliance to Auto-Start

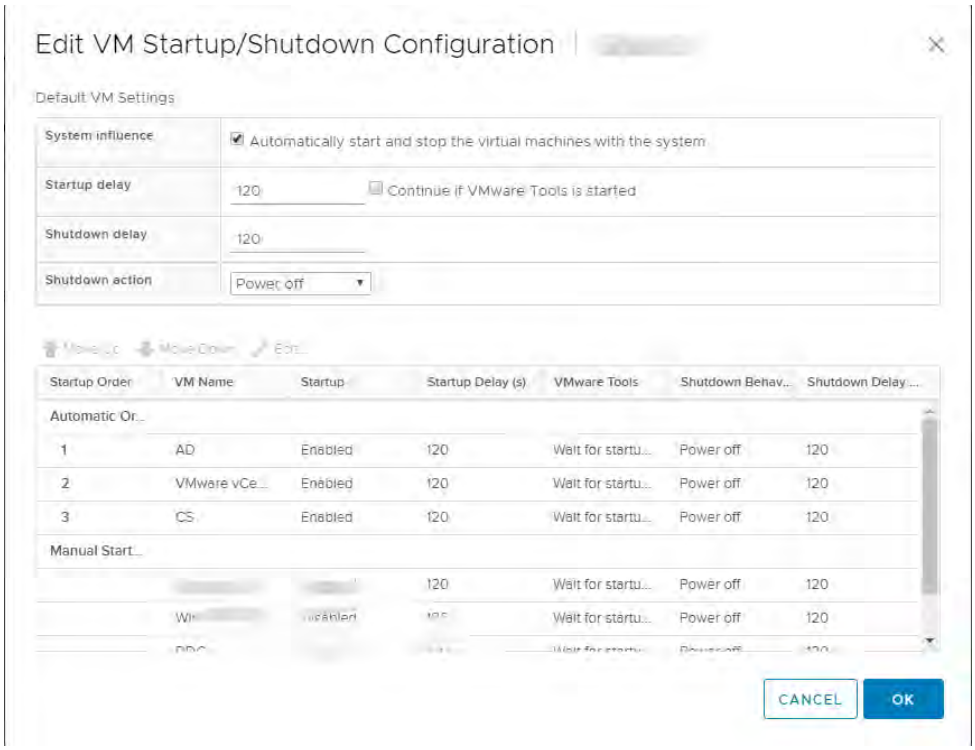
Use the following procedure to set a vCenter Appliance to start automatically:

1. In the vSphere Web Client, select the host then select **Configure**> **Virtual Machines**> **VM Startup/Shutdown.**



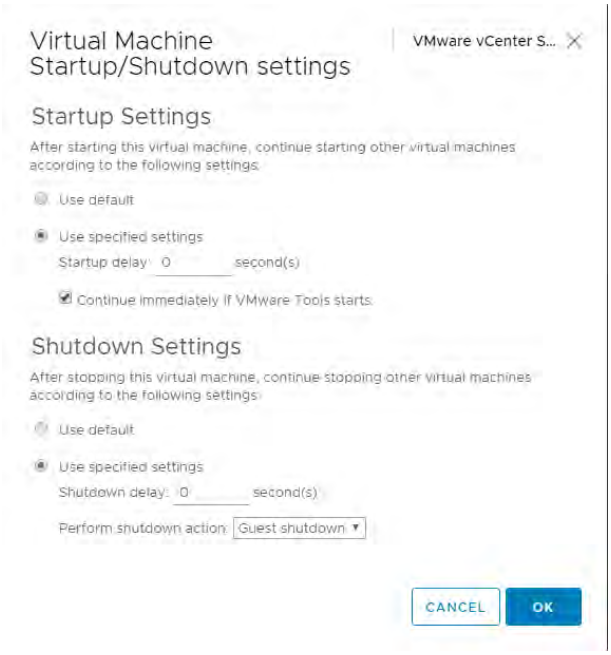
2. Click the **Edit** button.

   The Edit VM Startup and Shutdown window displays.



3. Select the **vCenter Appliance** and click the **Up** arrow to move that virtual machine up to the **Automatic Startup** section. Click the **Edit** button.

4. Continue if VMware Tools is started and select the following options:
   a) Set Startup Behavior to Use specified settings and select Continue immediately if VMware Tools starts
   b) Set Startup Delay to 0
   c) Set Shutdown Behavior to Use specified settings
   d) Set Shutdown Delay to 0
   e) Select Perform Shutdown Action and select Guest Shutdown



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

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

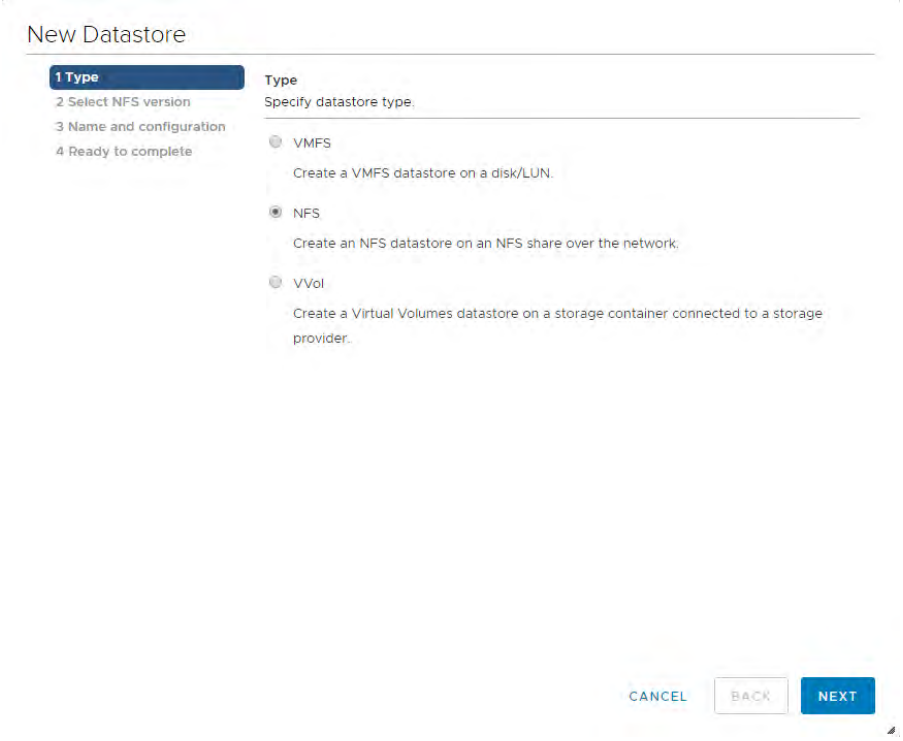
3.2.5 Mounting an NFS ISO Data Store

Use the following procedure to mount an NFS ISO data store:

1. In the main *vSphere Web Client* window, select **Hosts and Clusters** and select the host.

Select **Storage** -> **New Datastore** from the **Actions** drop-down menu.

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



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

   The **Select NFS version** tab displays.

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

   The Name and configuration tab displays.

4. Enter the NFS exported folder path and the NFS server address in the **Folder** and **Address** fields, respectively.

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

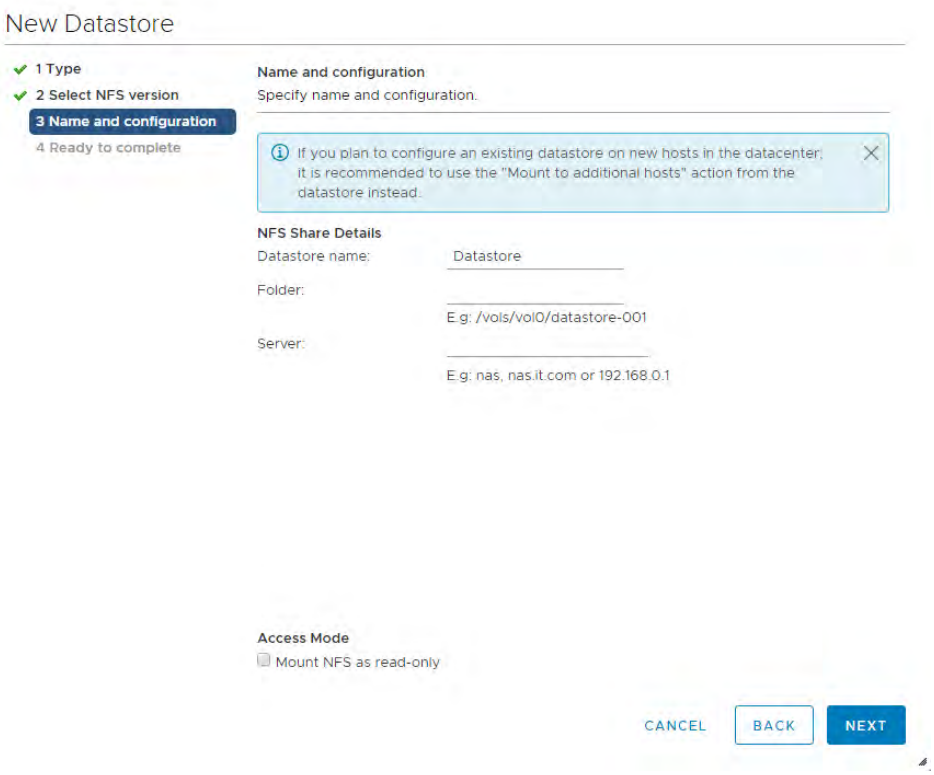
5. Click **Next** to proceed.

   The **Host accessibility** tab displays.

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

7. Select **Next** to proceed.

   The **Ready to complete** tab displays.



8. Review the settings.
9. Click **Finish** to complete adding the NFS ISO data store.

This data store is now accessible as an installation source for virtual machine CD drives.

Chapter 4. Building Citrix Virtual Apps & Desktops

This chapter covers building the core components for a Citrix Virtual Apps & Desktops environment, including:

- Installing the Citrix Delivery Controller, Citrix Studio, Citrix License Server
- Configuring the Citrix Delivery Controller, Citrix License Server

For the latest install and configure information refer to Citrix's Install Core Components Section of the Citrix Virtual Apps and Desktops Product Documentation. Additionally, Citrix offers solutions for deploying Delivery Controllers to the Cloud. Citrix Cloud is outside the scope of this document, so please consult Citrix and refer to the Citrix Cloud Product Documentation if you choose to have any components of your Citrix infrastructure in the Cloud.

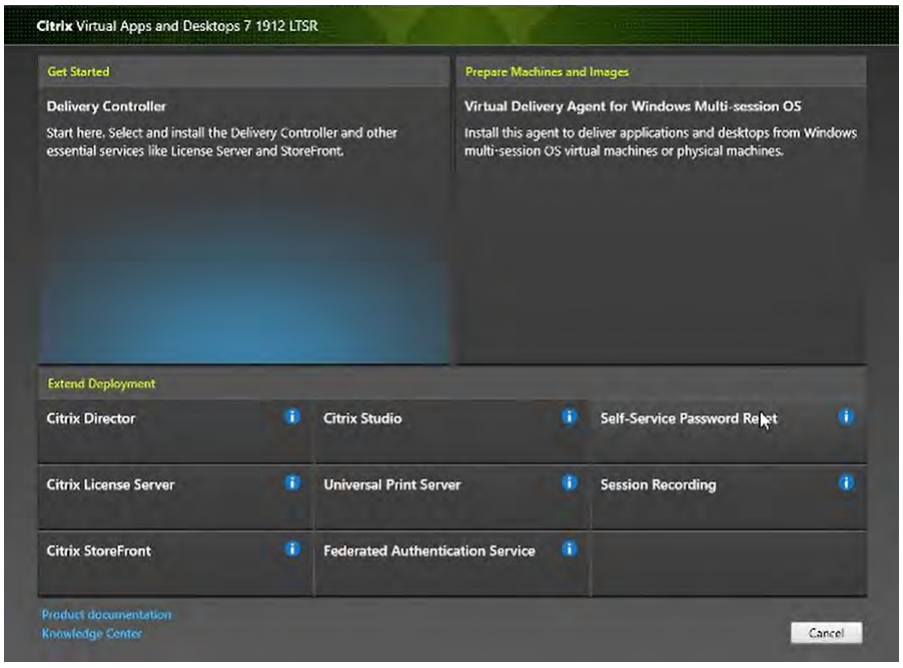
4.1 Installing the Citrix Delivery Controller

The Citrix Delivery Controller Server must meet the requirements listed in General Prerequisites..

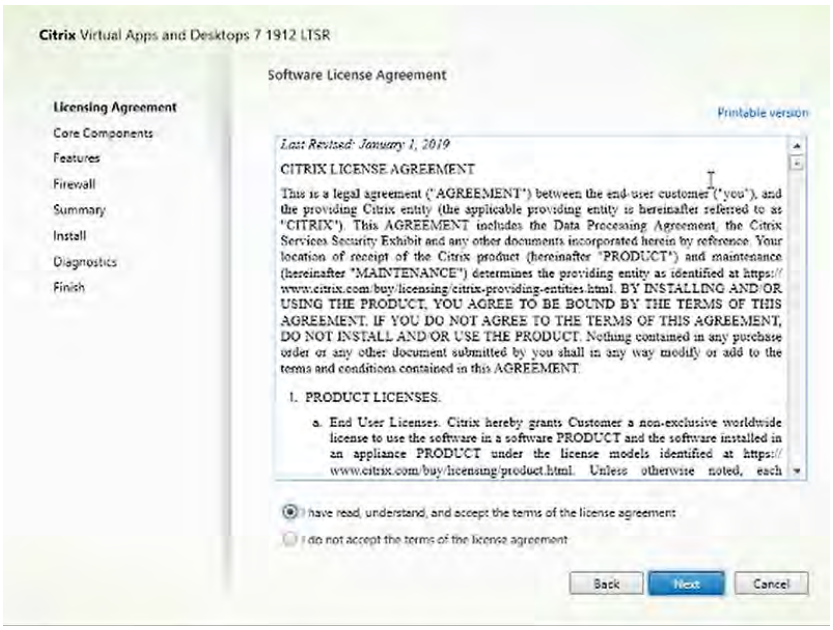
Use the following procedure to install Citrix Delivery Controller:

1. Attach the iso file to the server OS and open it via File Explorer.
2. Launch the ***Auto Select*** Application and accept the Windows User Account Control Popup.
3. Click **Start** for ***Virtual Apps and Desktops*** section.

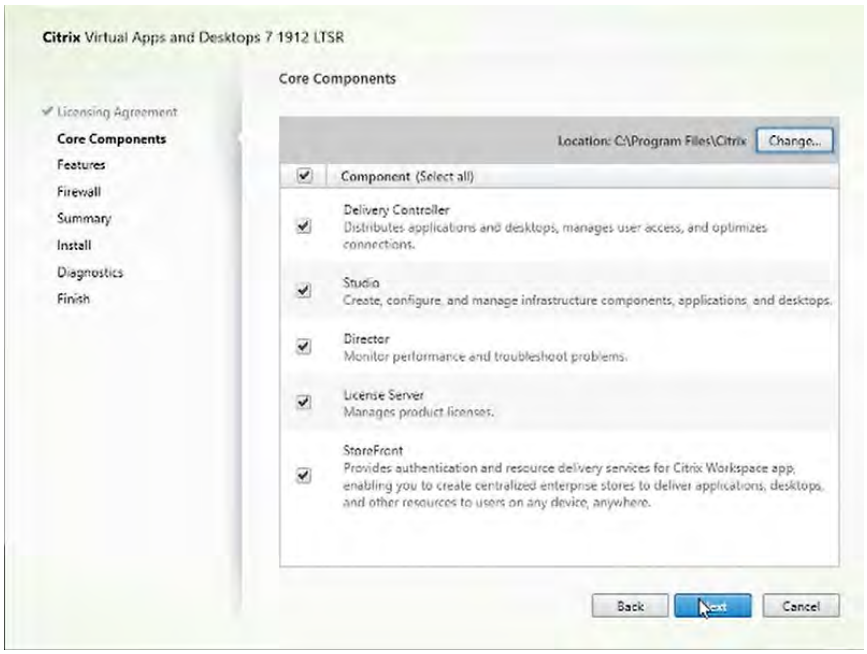
4. Select **Delivery Controller** to launch the Citrix Virtual Apps and Desktops installer.



5. Scroll and read and through the Software License Agreement.

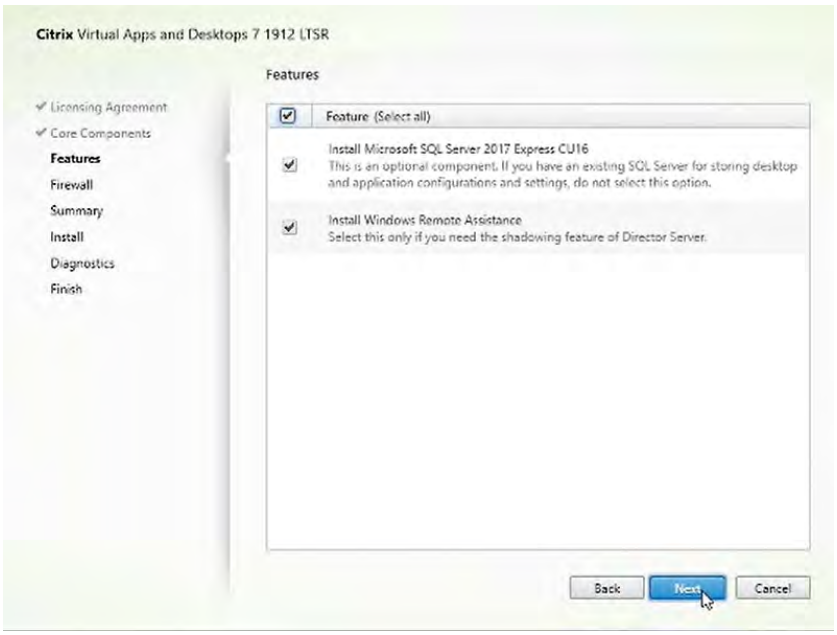


6. Check the **I have read, understand, and accept the terms of the license agreement** radio button to accept the agreement. Select **Next** to continue.



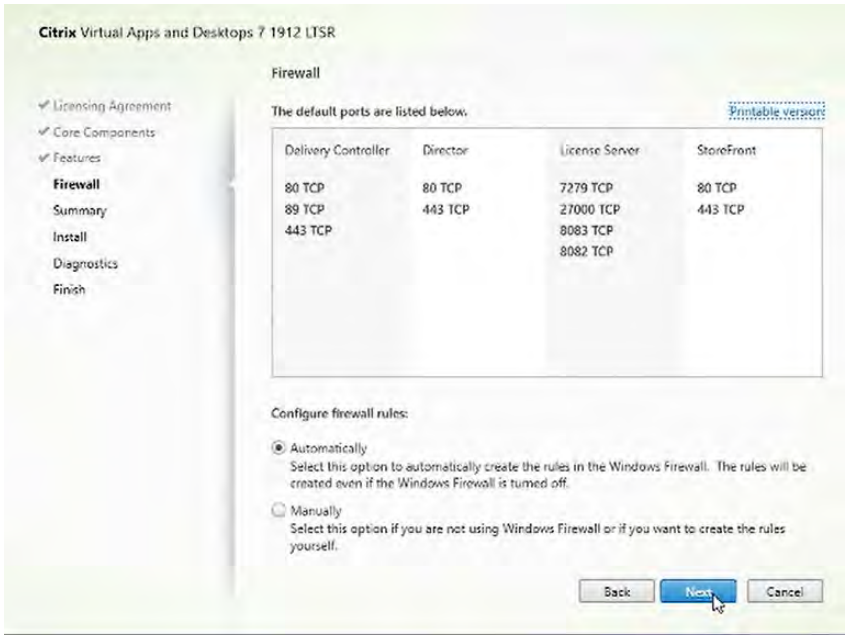
7. The Core Components window allows you to choose an install location and which components to install. For the purposes of POC/trial, ensure all components are selected and click **Next**.

Note: In a production environment, only Delivery Controller and Studio should be checked. Director, License Server, & StoreFront, should all reside on their own isolated servers. See the Citrix Virtual Apps & Desktops Install & Configure Product Documentation for production deployment instructions.

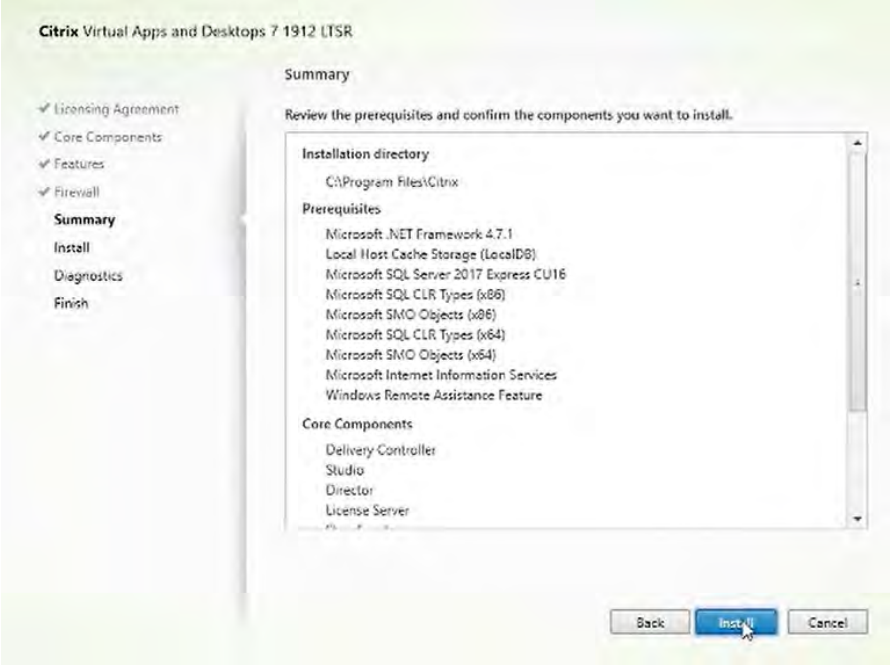


8. The Features window allows you to choose which Features to install. Ensure all features are selected and click **Next**.

9. The Firewall window allows you to configure Windows Firewall. Select the **Automatically** radio button and click **Next**.



10. On the Summary window select **Install**.



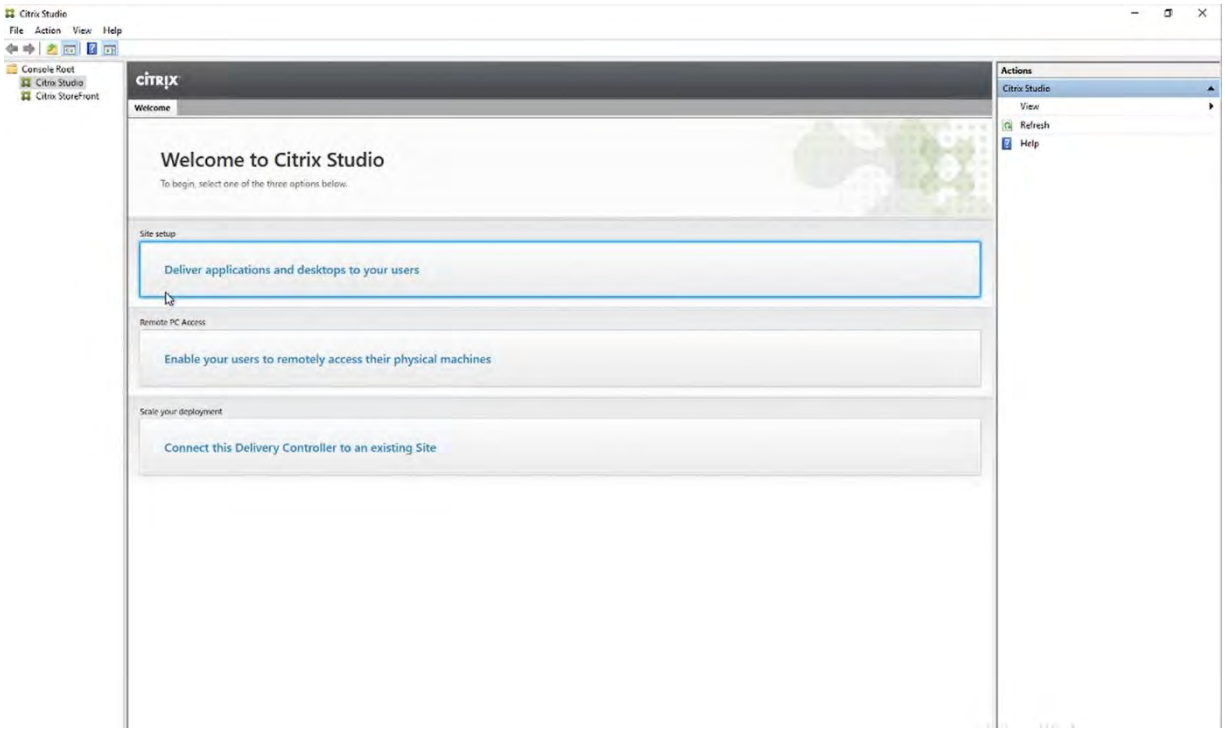
Accept any ***Reboot Prompts*** and reconnect to the server.

11. In the Diagnostics window, select the appropriate option to ***Collect diagnostic information*** **according to your organization policies**.

12. Select **Next** to continue.
13. Select **Finish** to complete the install.

4.2 Configuring the Citrix Delivery Controller

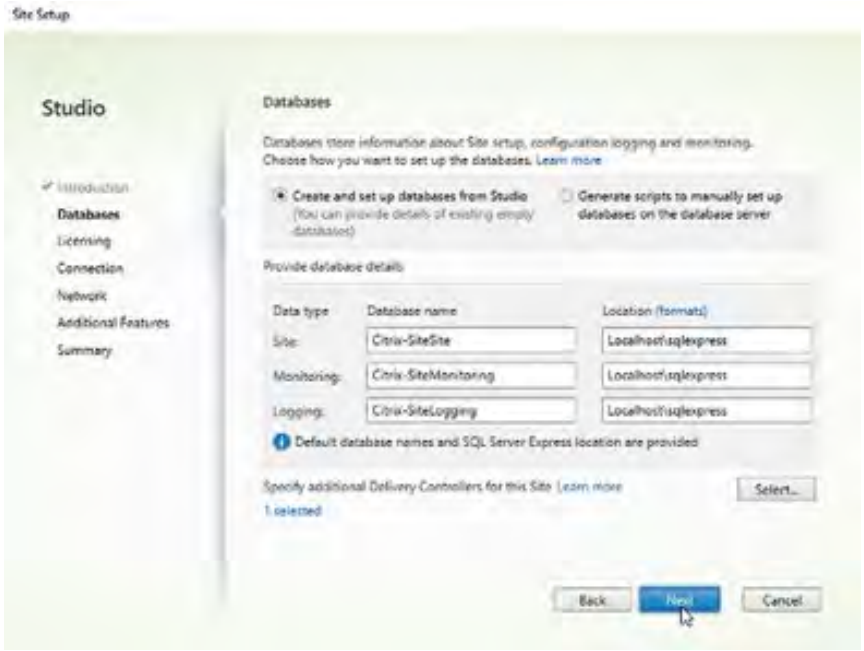
Use the following procedure to configure the Citrix Delivery Controller:



1. If Citrix Studio was not launched automatically after the install, launch **Citrix Studio** from the Windows Start Menu.

2. Select **Deliver applications and desktops to your users**.
3. On the Introduction window, ensure the "**A fully configured, production-ready Site (recommended for new users)**" radio button is selected.

4. In the **Site name:** type in a Site name.

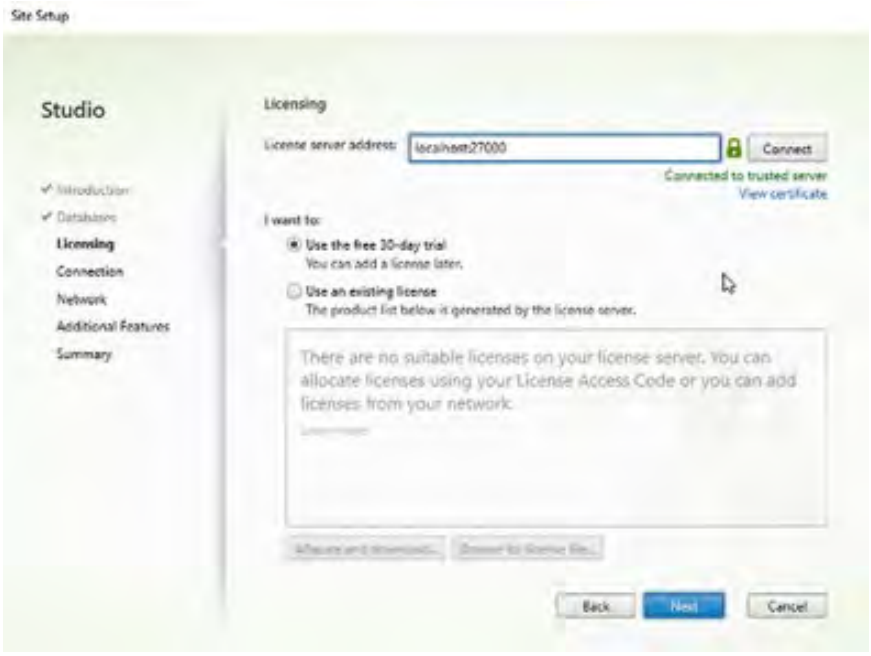


5. On the Licensing window in the "**License server address:**" text field, ensure **localhost:27000** is specified.

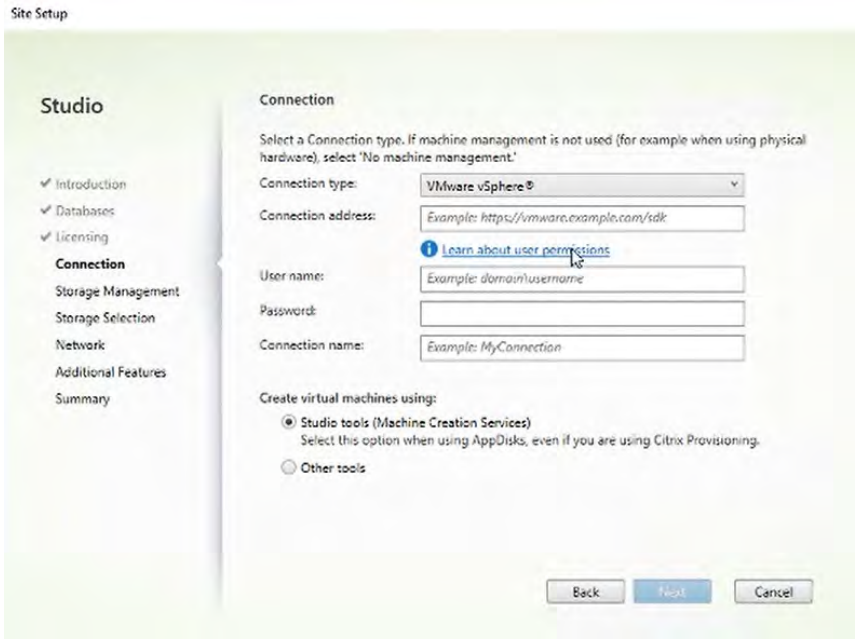
Note: In a production environment, you should provide the FQDN of your separate License Server, & correct port number.

6. Next, ensure the "**Use the free 30-day trial**" radio button is selected and click **Next**.

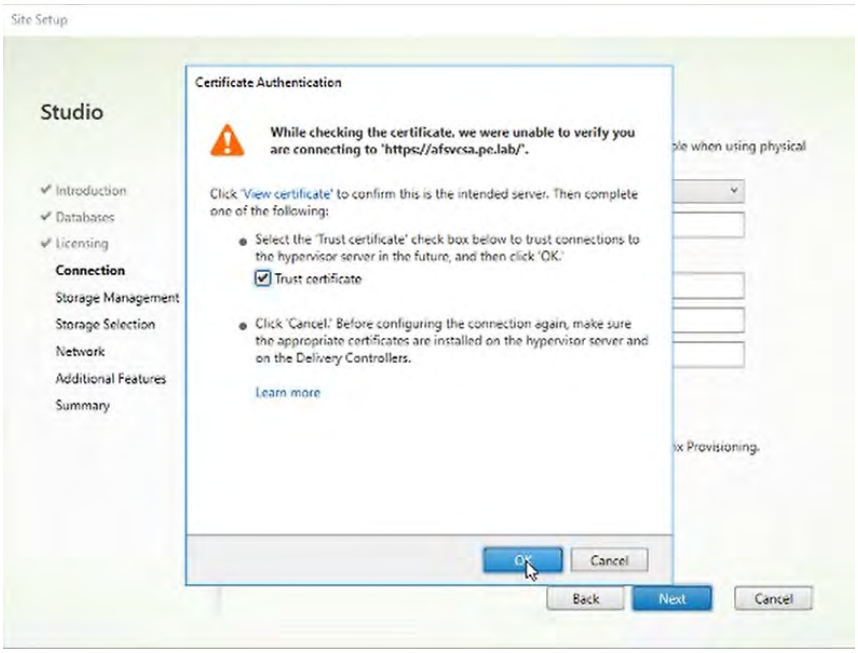
Note: In a production environment, you should use the existing license from the FQDN License Sever.



7. On the Connection Window perform the following tasks.
   a) Under the **Connection type:** drop down menu, select **VMware vSphere®**.
   b) In the **Connection address:** text filed, enter the URL for your vSphere management server.
      i. Note that the connection must use SSL.
   c) In the **User name:** text field, enter your VMware vSphere® administrator's username.
   d) In the **Password:** text field, enter the VMware vSphere® administrator's password.
   e) In the **Connection name:** text field, enter a name for this connection.



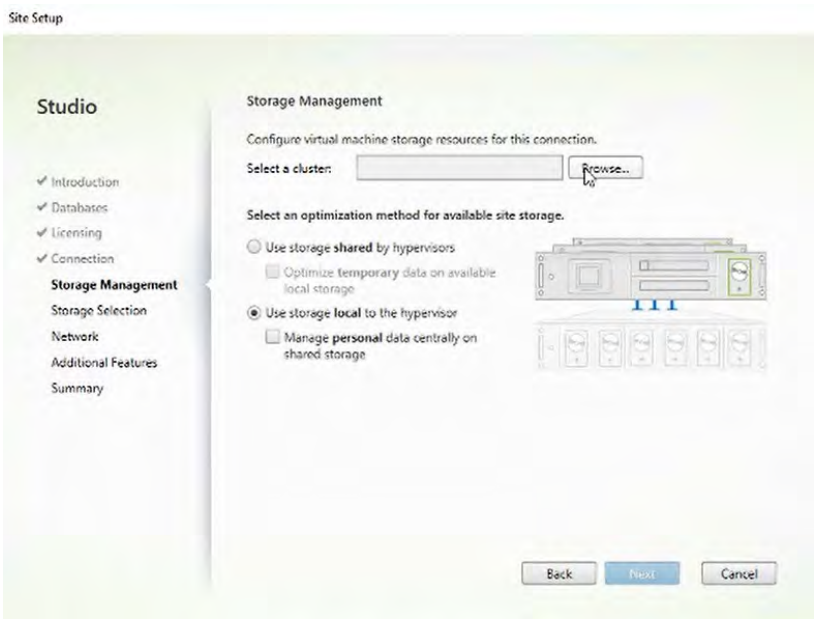
8. Ensure the **Studio tools (Machine Creation Services)** radio button is selected and click **Next**.
9. If the Delivery Controller does not already trust the vSphere Management server certificate, the Certificate Authentication window will appear.
   a) Select the **Trust certificate** checkbox and click **OK**.



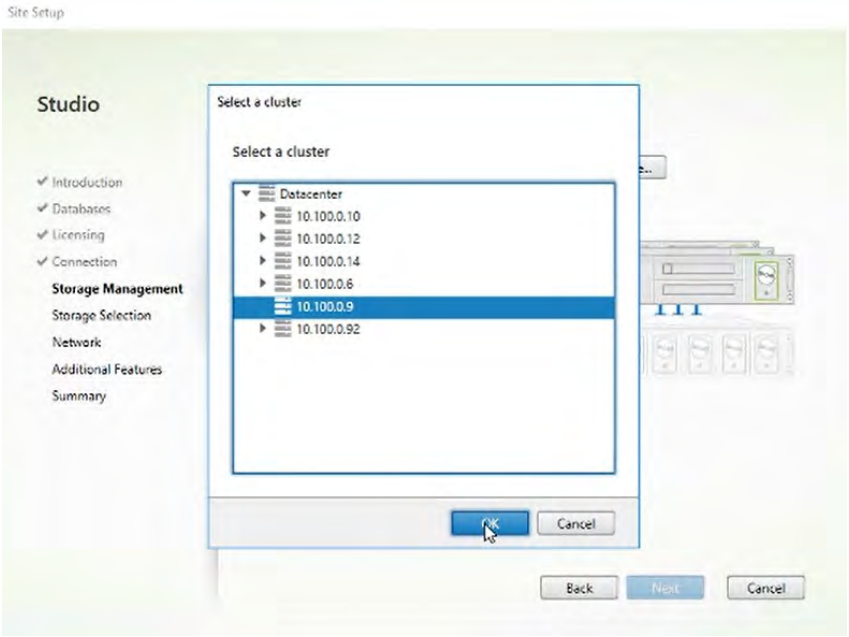
10. The Storage Management window allows you to choose a cluster for machine creation. For purposes of POC/trial, we select a single host with local storage.

Note: In a production environment, you may need to select a cluster with multiple hosts and shared storage.

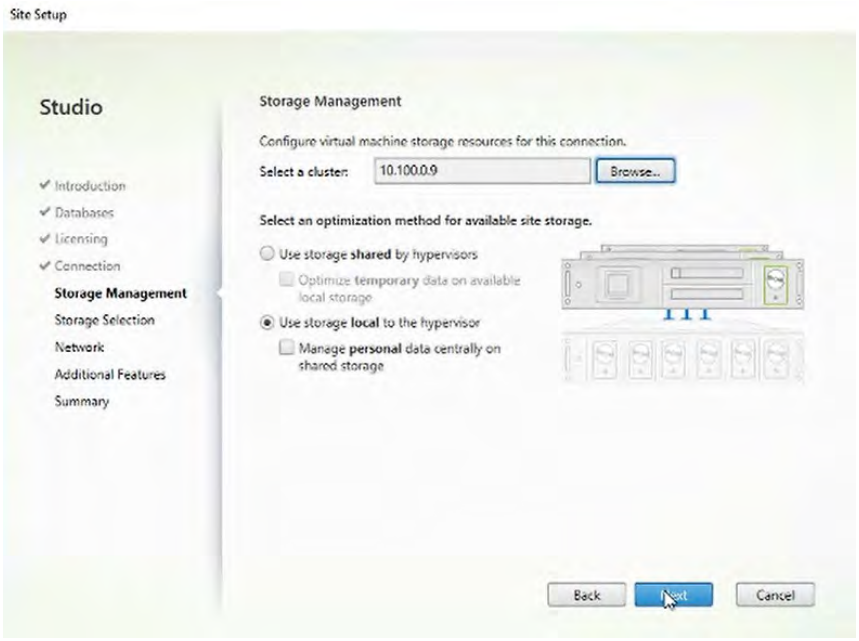
11. To select your cluster, click **Browse…**



12. Select the cluster from the dropdown menu and click **OK**.

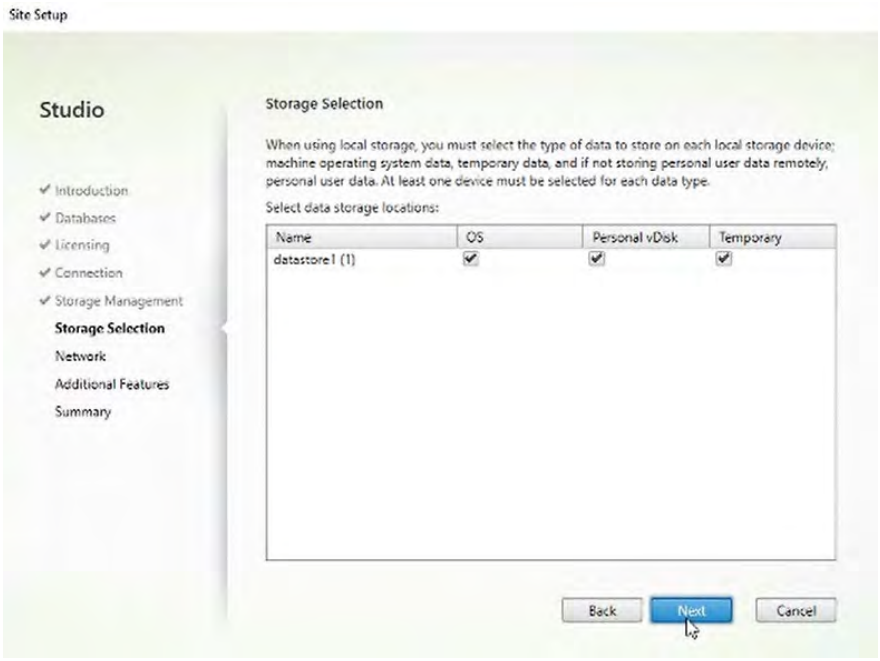


13. Select the appropriate Storage Management option. Since we are using local storage, the “Use storage **local** to the hypervisor” radio button is selected and click **Next**.

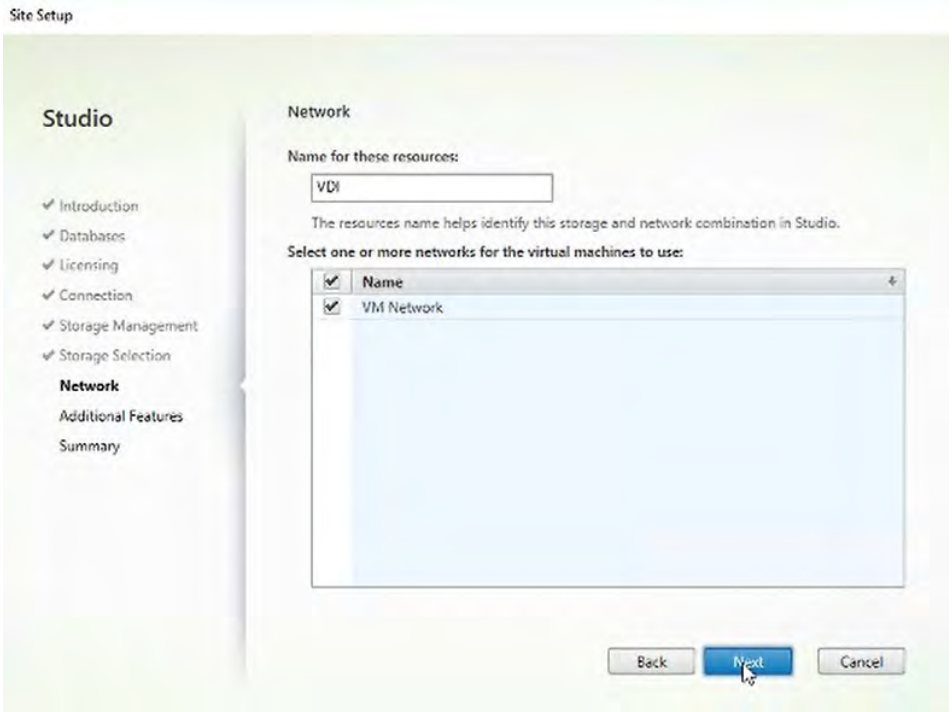


14. On the Storage Selection window, ensure the OS, Personal vDisk, & Temporary, check boxes are selected for your local storage and click **Next**.

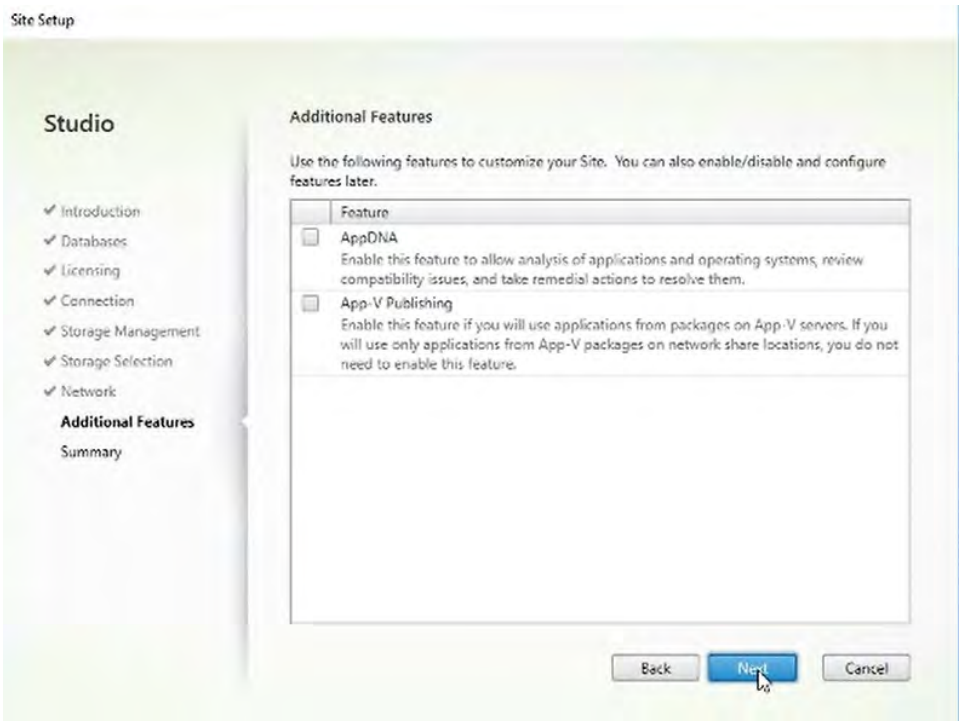
Note: In a production environment you may need to place the OS, Personal vDisk, & Temporary files on separate datastores.



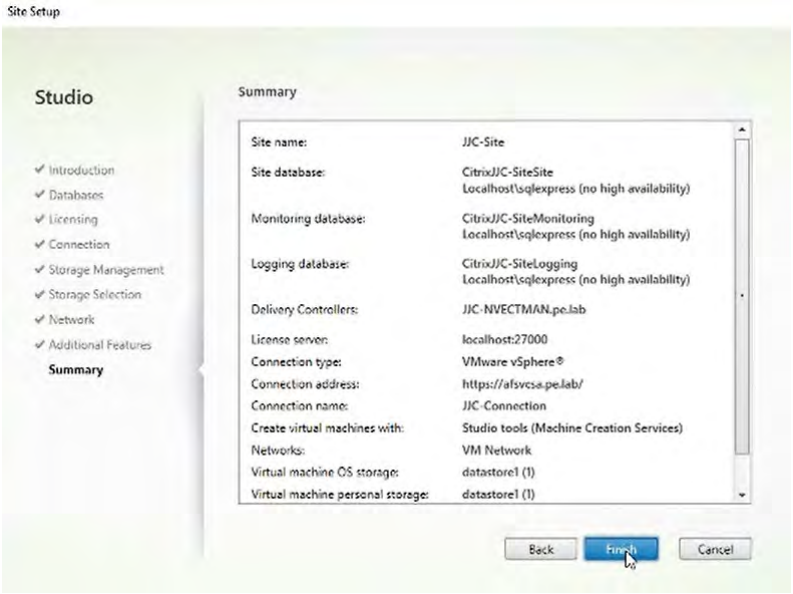
15. On the Network window, type a name for the resources in the **Name for these resources:** text field.



16. On the Additional Features window, click **Next**



17. On the Summary window, click **Finish**.



Chapter 5. NVIDIA vGPU Manager Installation

This chapter covers installing and configuring the NVIDIA vGPU Manager:

- Uploading VIB in vSphere Web Client
- Installing the VIB
- Updating the VIB
- Verifying the Installation of the VIB
- Uninstalling VIB
- Changing the Default Graphics Type in VMware vSphere 6.5 and Later
- Changing the vGPU Scheduling Policy

5.1 Uploading VIB in vSphere Web Client

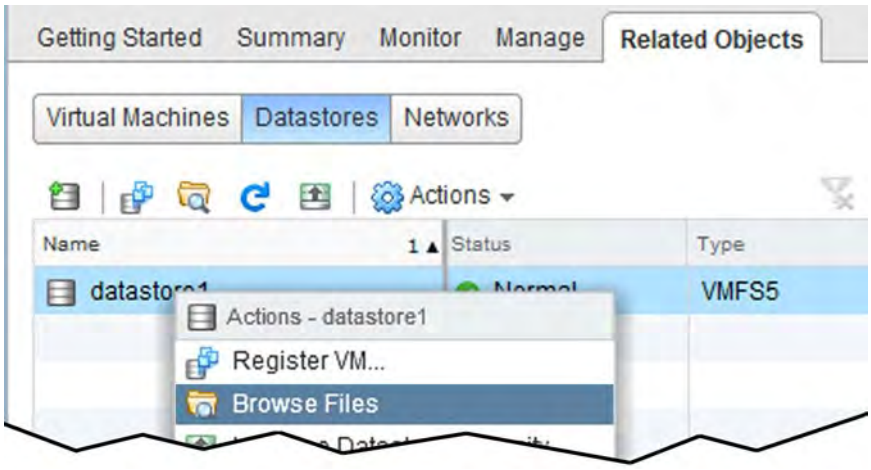
For demonstration purposes, these steps use the VMWare vSphere web interface for uploading the VIB to the server host.

Before you begin, download the archive containing the VIB file and extract the contents of the archive to a folder. The file ending with VIB is the file that you must upload to the host data store for installation.

To upload the file to the data store using vSphere Web Client:

1. Click the **Related Objects** tab for the desired server.
2. Select **Datastores**.
3. Either right click the data store and then select **Browse Files** or click the icon in the toolbar.

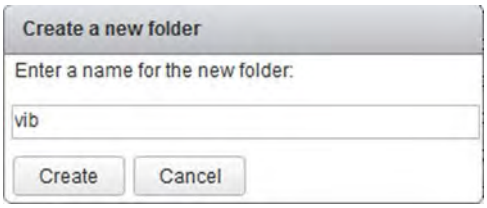
   The *Datastore Browser* window displays.



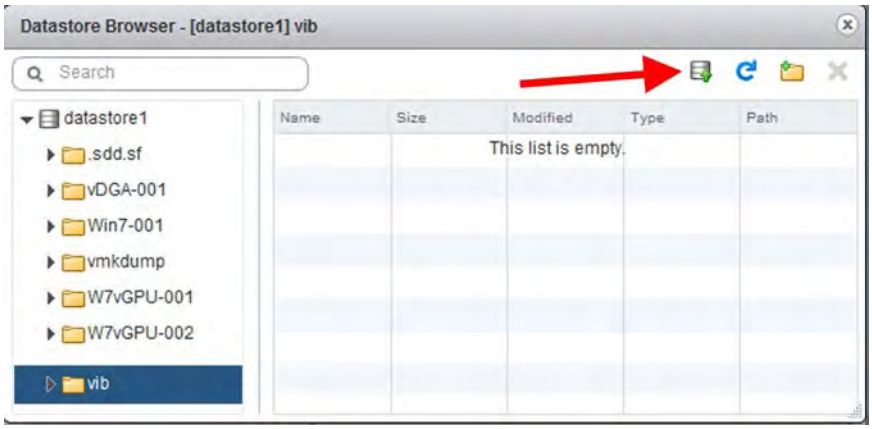
4. Click the **New Folder** icon.

   The *Create a new folder* window displays.

5. Name the new folder **vib** and then click **Create**.



6. Select the **vib** folder in the *Datastore Browser* window.
7. Click the **Upload** icon.



   The *Client Integration Access Control* window displays.

8. Select **Allow**.

   The **.VIB** file is uploaded to the data store on the host.

> Note: If you do not click Allow before the timer runs out, then further attempts to upload a file will silently fail. If this happens, exit and restart vSphere Web Client., Repeat this procedure and be sure to click Allow before the timer runs out.

5.2 Installing vGPU Manager with the .vib File

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

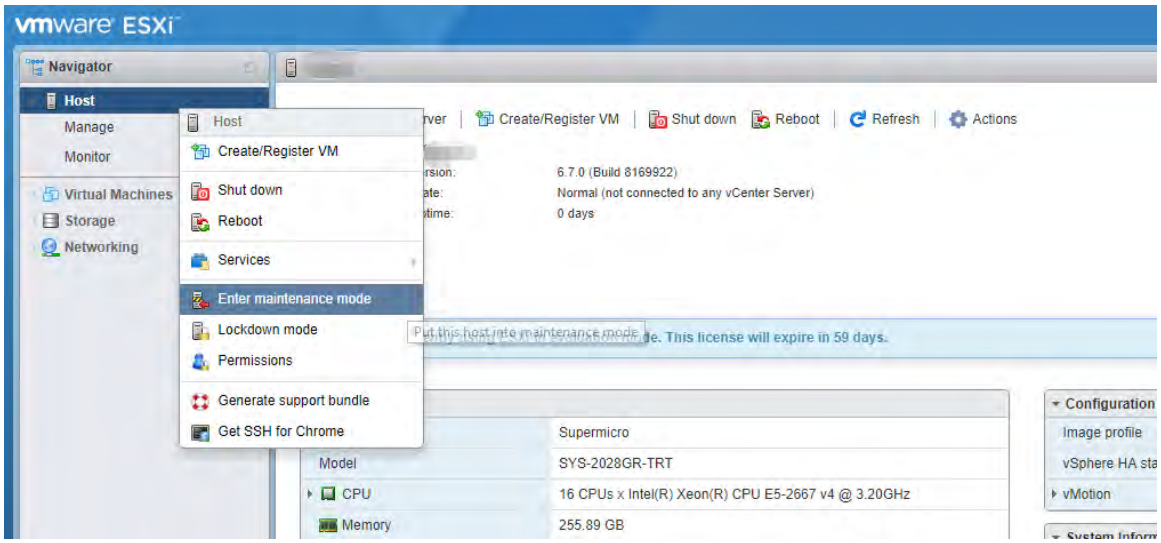
- As a VIB file, which must be copied to the ESXi host and then installed
- As an offline bundle that you can import manually as explained in Import Patches Manually in the VMware vSphere documentation

CAUTION: Prior to vGPU software 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 placed 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 into maintenance mode disables any vCenter appliance running on this host until you exit maintenance mode and then restart that vCenter appliance.

2. Click **OK** to confirm your selection.
3. Use the esxcli command to install the vGPU Manager package:

```
[root@esxi:~] esxcli software vib install -v directory/NVIDIA-vGPU-
VMware_ESXi_6.0_Host_Driver_390.72-1OEM.600.0.0.2159203.vib
Installation Result    Message: Operation finished successfully.
   Reboot Required: false
   VIBs Installed: NVIDIA-vGPU-VMware_ESXi_6.0_Host_Driver_390.72-
1OEM.600.0.0.2159203
   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 then 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 then click **OK** to proceed.

5.3 Updating vGPU Manager with the .vib File

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 how to enable ESXi Shell or SSH for an ESXi host.

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 how to place 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-vGPU-
VMware_ESXi_6.0_Host_Driver_390.72-1OEM.600.0.0.2159203.vib
Installation Result    Message: Operation finished successfully.
   Reboot Required: false
   VIBs Installed: NVIDIA-vGPU-
VMware_ESXi_6.0_Host_Driver_390.72-1OEM.600.0.0.2159203
   VIBs Removed: NVIDIA-vGPU-
VMware_ESXi_6.0_Host_Driver_390.57-1OEM.600.0.0.2159203
    VIBs Skipped:
```

   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 vGPU Manager

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

1. Verify that the NVIDIA vGPU software package 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.

```
[root@esxi:~] nvidia-smi
Fri Jul 20 17:56:22 2018
+-----------------------------------------------------+
| NVIDIA-SMI 390.72     Driver Version: 390.75        |
|-------------------------------+----------------------+----------------------+
| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
|===============================+======================+======================|
|   0  M60           On         | 0000:85:00.0     Off |                  Off |
| N/A   23C    P8    23W / 150W |     13MiB /  8191MiB |      0%      Default |
+-------------------------------+----------------------+----------------------+
|   1  M60           On         | 0000:86:00.0     Off |                  Off |
| N/A   29C    P8    23W / 150W |     13MiB /  8191MiB |      0%      Default |
```

```
+-------------------------------+----------------------+----------------------+
|   2  P40           On         | 0000:87:00.0     Off |                  Off |
| N/A   21C    P8    18W / 250W |     53MiB / 24575MiB |      0%      Default |
+-------------------------------+----------------------+----------------------+
+-----------------------------------------------------------------------------+
| Processes:                                                       GPU Memory |
|  GPU       PID  Type  Process name                               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.

Example:

```
nvidia-smi –query gpu=timestamp,name,utilization.gpu,memory.free,memory.used --
format=csv -l 5
```

5.5 Uninstalling vGPU Manager

1. Determine the name of the vGPU driver bundle.

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

   This command returns output similar to the following:

   ```
   NVIDIA-VMware_ESXi_6.7_Host_Driver 390.72-1OEM.600.0.0.2159203
   NVIDIA VMwareAccepted    2018-07-20
   ```

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

   ```
   $ esxcli software vib remove -n NVIDIA-VMware_ESXi_6.7_Host_Driver
   --maintenance-mode
   ```

   The following message displays when installation is successful:

   ```
   Removal Result
      Message: Operation finished successfully.
      Reboot Required: false
      VIBs Installed:
      VIBs Removed: NVIDIA_bootbank_NVIDIA-
   VMware_ESXi_6.7_Host_Driver_390.72-1OEM.600.0.0.2159203
      VIBs Skipped:
   ```

3. Reboot the host to complete the uninstallation process.

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

The vGPU Manager VIBs for VMware vSphere 6.5 and later provide vSGA and vGPU functionality in a single VIB. After this VIB is installed, the default graphics type is set to **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 change the default graphics type, VMs to which a vGPU is assigned fail to start and the following error message is displayed:

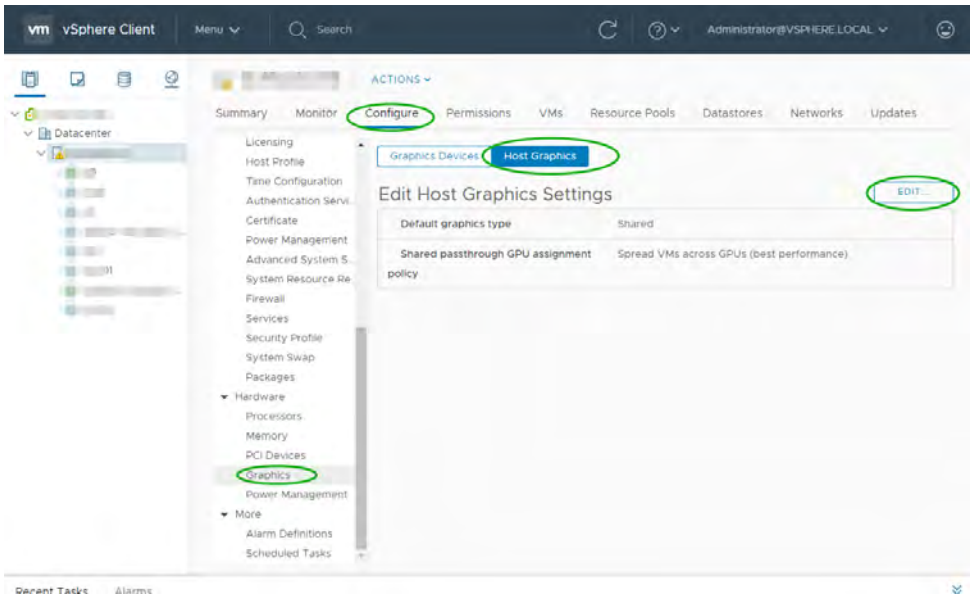
```
The amount of graphics resource available in the parent resource pool is
insufficient for the operation.
```

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.

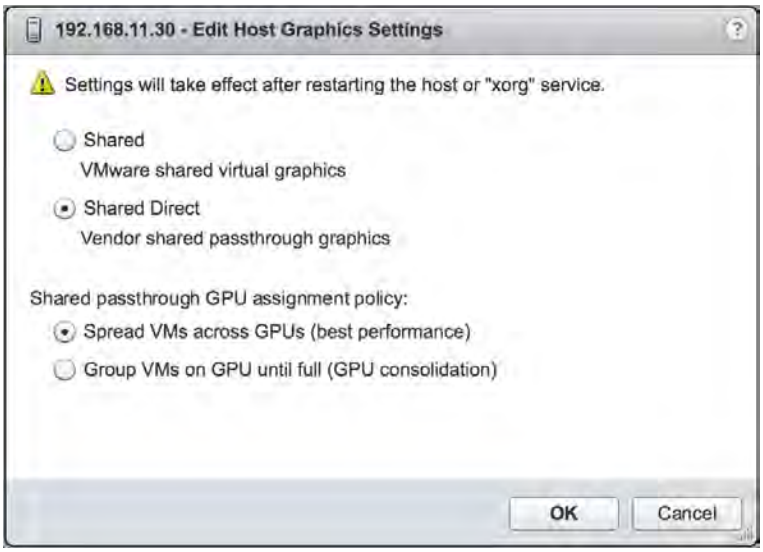
Change the default graphics type before configuring vGPU. Output from the VM console in the VMware vSphere Web Client is not available for VMs that are running vGPU.

Before changing the default graphics type, ensure that the ESXi host is running and that all VMs on the host are 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 dialog box that opens, select **Shared Direct** and click **OK**.



Note: In this dialog box, you can also 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 explained in Configuring a vSphere VM with Virtual GPU.

See also the following topics in the 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 the performance of graphics-light applications running in other VMs.

GPUs, starting with the NVIDIA Pascal™ architecture, also supports equal share and fixed share vGPU schedulers. These schedulers impose a limit on 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.

The GPUs that are based on the Pascal architecture are the NVIDIA P4, NVIDIA P6, NVIDIA P40, and NVIDIA P100.

The GPUs that are based on the Volta™ architecture are the NVIDIA V100 SXM2, NVIDIA V100 PCIe, NVIDIA V100 FHHL, and NVIDIA V100s.

The GPUs that are based on the Turing™ architecture are the NVIDIA T4, RTX6000 and RTX8000.

The GPU that is based on the Ampere™ architecture is the NVIDIA A100 & A40.

5.7.1 vGPU Scheduling Policies

NVIDIA RTX vWS provides three GPU scheduling options to accommodate a variety of QoS requirements of customers. Additional information regarding GPU scheduling can be found here.

- **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 since it allows the use of common benchmarks used to measure physical workstation performance such as SPECviewperf, to compare the performance with current physical or virtual workstations.
- **Best effort scheduling** provides consistent performance at a higher scale and therefore reduces the TCO per user. The best effort scheduler leverages a round-robin scheduling algorithm which shares GPU resources based on actual demand which results in optimal utilization of resources. This results in consistent performance with optimized user density. The best effort scheduling policy best utilizes the GPU during idle and not fully utilized times, allowing for optimized density and a good QoS.
- **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.

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 fixed share scheduler with the default time slice length.

```
RmPVMRL=0x11
```

This example sets the vGPU scheduler to 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

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

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 secure shell (SSH) for this purpose. Set the RmPVMRL registry key to the value that sets the GPU scheduling policy that you want.

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

Value **-** The value that sets the vGPU scheduling policy that you want, for example:

   a) **0x01** **-** Sets the vGPU scheduling policy to Equal Share Scheduler.
   b) **0x11** **-** Sets the vGPU scheduling policy to Fixed Share Scheduler.
   c) For all supported values, see RmPVMRL Registry Key.

2. Reboot your hypervisor host machine.

5.7.4 Changing the vGPU Scheduling Policy for Select GPUs

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

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 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. Add the `-D` option to display the PCI domain and the `-d 10de:` option to display information only for NVIDIA GPUs.

```
# lspci -D -d 10de:
```

The NVIDIA GPUs listed in this example have the PCI domain 0000 and BDFs 85:00.0 and 86:00.0.

```
0000:85:00.0 VGA compatible controller: NVIDIA Corporation GM204GL [M60] (rev a1)
0000:86:00.0 VGA compatible controller: NVIDIA Corporation GM204GL [M60] (rev a1)
```

3. Use the module parameter `NVreg_RegistryDwordsPerDevice` to set the `pci` and `RmPVMRL` registry keys for each GPU.
4. Add the following entry to the `/etc/modprobe.d/nvidia.conf` file.

```
options nvidia 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`
  - **0x00** - 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.
  - **0x011** - 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.

5.7.5 Restoring Default vGPU Scheduler Settings

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 secure shell (SSH) 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="
```

4. Reboot your hypervisor host machine

5.8 Disabling and Enabling ECC Memory

Some GPUs that support NVIDIA vGPU software support error correcting code (ECC) memory with NVIDIA vGPU. 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.

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 that is 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 that is usable by 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. M60 and M6 GPUs support ECC memory when used without GPU virtualization, but NVIDIA vGPU does not support ECC memory with these GPUs. In graphics mode, these GPUs are supplied with ECC memory disabled by default.

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

If you are using 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 if you are 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 are using 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 from 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 on which 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, also 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                           : Mon Jul 13 18:36:45 2020
Driver Version                      : 450.55

Attached GPUs                       : 1
GPU 0000:02:00.0

[...]

    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                                 : Mon Jul 13 18:37:53 2020
Driver Version                            : 450.55

Attached GPUs                             : 1
GPU 0000:02:00.0
[...]

    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 to perform this task from 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 on which 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, also 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                                 : Mon Jul 13 18:36:45 2020
Driver Version                            : 450.55

Attached GPUs                             : 1
GPU 0000:02:00.0

[...]
```

```
    Ecc Mode
        Current                     : Disabled
        Pending                     : Disabled

[...]
```

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

   a) 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
```

   b) 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                           : Mon Jul 13 18:37:53 2020
Driver Version                      : 450.55

Attached GPUs                       : 1
GPU 0000:02:00.0
[...]

    Ecc Mode
        Current                     : Enabled
        Pending                     : Enabled

[...]
```

Chapter 6. Deploying the NVIDIA vGPU Software License Server

This chapter covers deployment of the NVIDIA vGPU software license server, including:

- Platform Requirements
- Installing the Java Runtime Environment on Windows
- Installing the License Server Software on Windows

6.1 Platform Requirements

Before proceeding, ensure that you have a platform suitable for hosting the license server

6.1.1 Hardware and Software Requirements

- The hosting platform may be a physical machine, an on-premises virtual machine (VM), or a VM on a supported cloud service. NVIDIA recommends using a host that is dedicated solely to running the license server.
- The recommended minimum configuration is 2 CPU cores and 4 GB of RAM. A high-end configuration of 4 or more CPU cores with 16 GB of RAM is suitable for handling up to 150,000 licensed clients.
- At least 1 GB of hard drive space is required.
- The hosting platform must run a supported operating system.
- On Window platforms, .NET Framework 4.5 or later is required.

6.1.2 Platform Configuration Requirements

- The platform must have a fixed (unchanging) IP address. The IP address may be assigned dynamically by DHCP or statically configured but must be constant.
- The platform must have at least one unchanging Ethernet MAC address, to be used as a unique identifier when registering the server and generating licenses in the NVIDIA Licensing Portal.
- The platform's date and time must be set accurately. NTP is recommended.

6.1.3 Network Ports and Management Interface

The license server requires TCP port 7070 to be open in the platform's firewall, to serve licenses to clients. By default, the installer will automatically open this port. The license server's management interface is web-based and uses TCP port 8080. The management interface itself does not implement access control; instead, the installer does not open port 8080 by default, so that the management interface is only available to web browsers running locally on the license server host. Access to the management interface is therefore controlled by limiting remote access (via VNC, RDP, etc.) to the license server platform.

Note: If you choose to open port 8080 during license server installation, or at any time afterwards, the license server's management interface is unprotected.

6.2 Installing the NVIDIA vGPU Software License Server on Windows

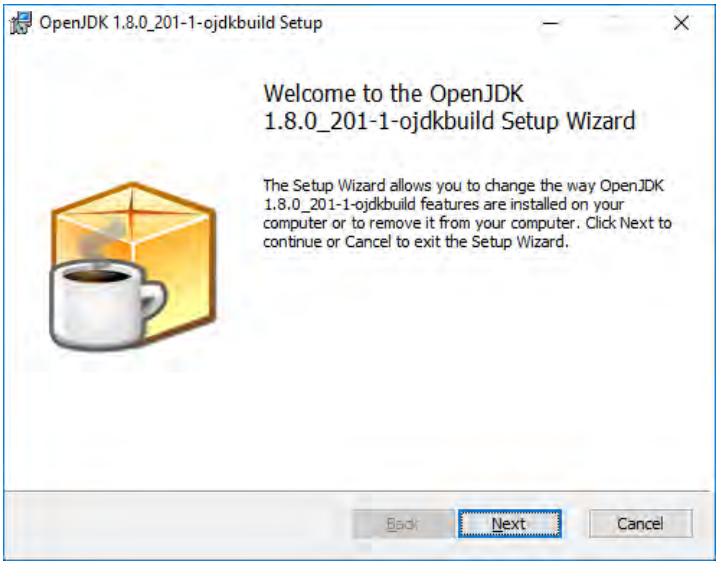
The license server requires a Java runtime environment, which must be installed separately before you install the license server.

6.2.1 Installing the Java Runtime Environment on Windows

If a suitable Java runtime environment (JRE) version is not already installed on your system install a supported JRE before running the NVIDIA license server installer.

1. Download a supported 64-bit Oracle Java SE JRE or OpenJDK JRE.
   a) Download Oracle Java SE JRE from the Java Downloads for All Operating Systems page.
      i. Download Oracle Java SE JRE from the java.com: Java + You page
   b) Download OpenJDK JRE from the Community builds using source code from OpenJDK project on GitHub.
2. Install the JRE that you downloaded.
   a) Oracle Java SE JRE installation:

b) OpenJDK JRE installation:



3. Set the JAVA_HOME system variable to the full path to the jre folder of your JRE installation.
   a) **For 64-bit Oracle Java SE JRE**: C:\Program Files\Java\jre1.8.0_191
   b) **For 64-bit OpenJDK JRE**: C:\Program Files\ojdkbuild\java-1.8.0-openjdk-1.8.0.201-1\jre

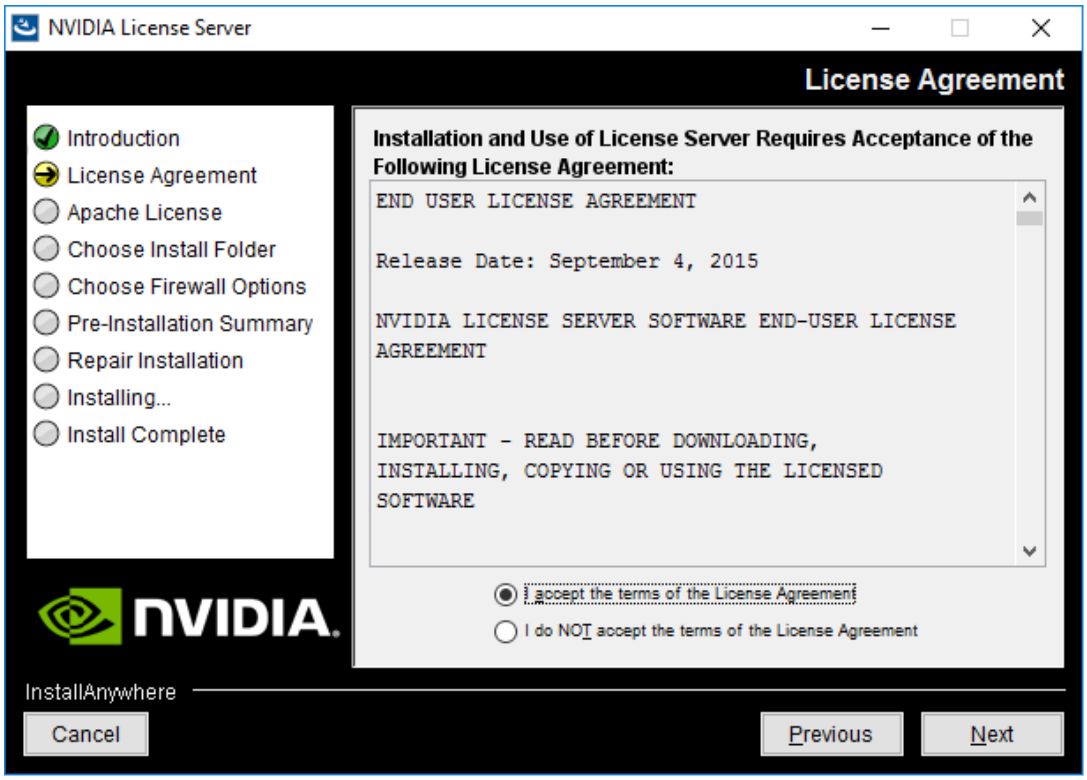
   Ensure that the path does not include any trailing characters, such as a slash or a space.

   If you are upgrading to a new version of the JRE, update the value of the JAVA_HOME system variable to the full path to the jre folder of your new JRE version.
4. Ensure that the Path system variable contains the path to the java.exe executable file.
   a) **For 64-bit Oracle Java SE JRE**: C:\Program Files\Java\jre1.8.0_191\bin

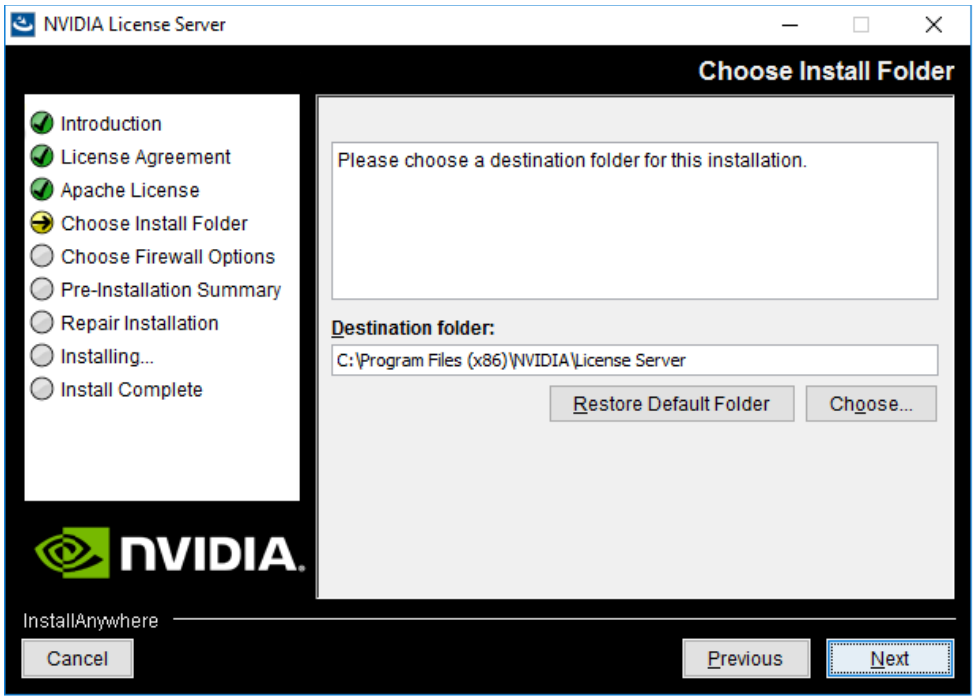
b) **For 64-bit OpenJDK JRE**: C:\Program Files\ojdkbuild\java-1.8.0-openjdk-1.8.0.201-1\bin

6.2.2 Installing the License Server Software on Windows

1. Unzip the license server installer and run setup.exe.
2. Accept the EULA for the license server software and the Apache Tomcat software used to support the license server's management interface.



3. Choose the destination folder where you want the license server software to be installed.

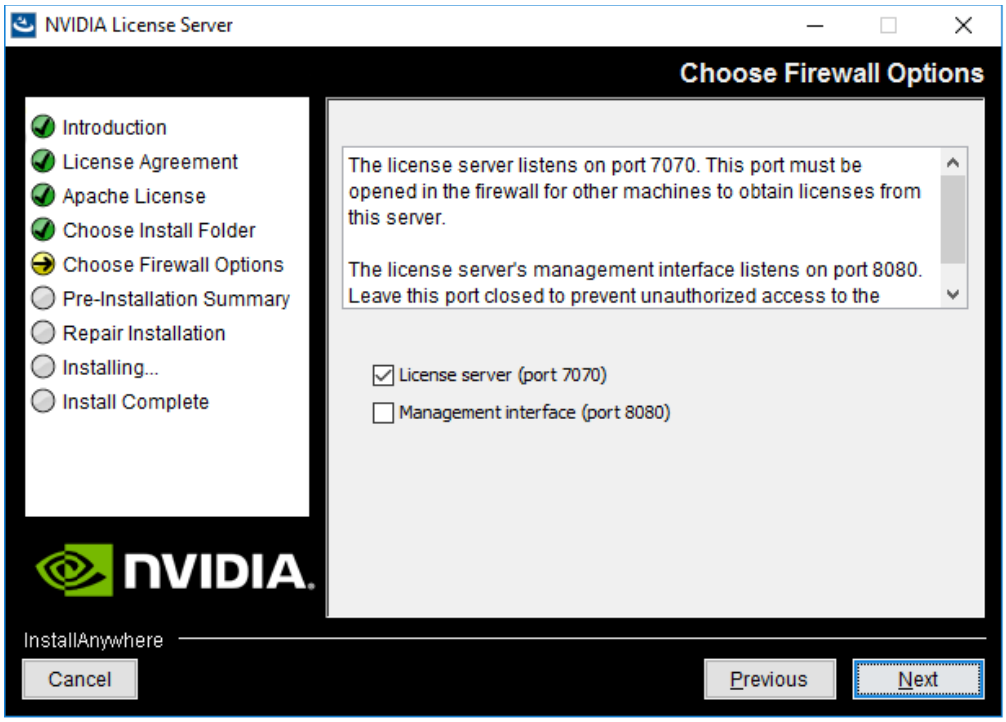


4. In the Choose Firewall Options dialog box, select the ports to be opened in the firewall.

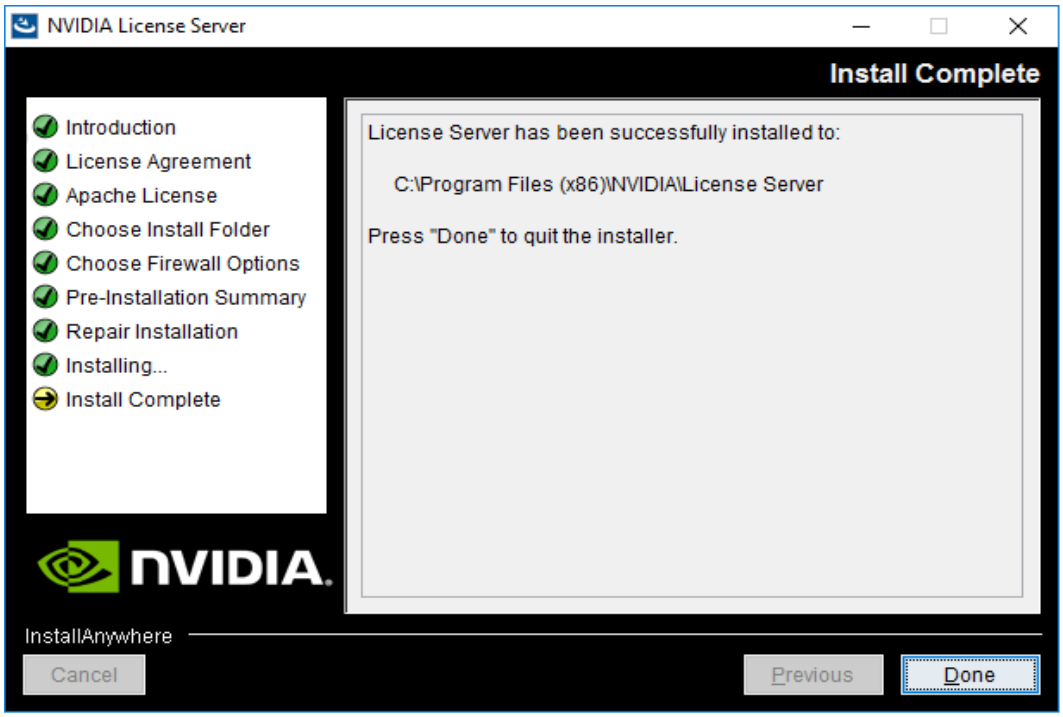
   To enable remote clients to access licenses from the server and prevent remote access to the management interface, use the default setting, which sets ports as follows:

   a) Port 7070 is open to enable remote clients to access licenses from the server.

   b) Port 8080 is closed to ensure that the management interface is available only through a web browser running locally on the license server host.



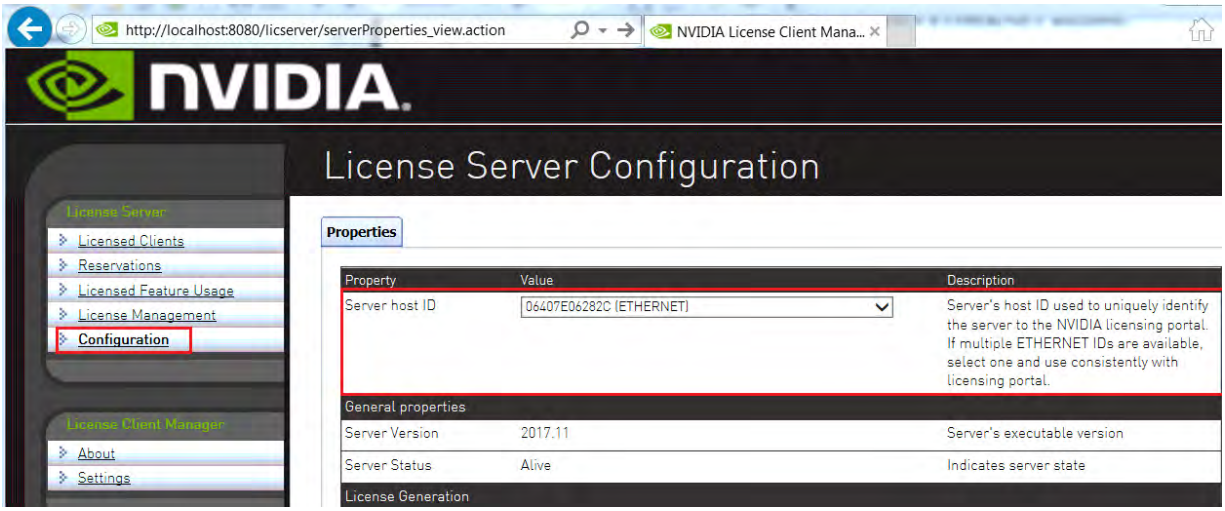
5. After installation has completed successfully, click Done to exit the installer.



6.2.3 Obtaining the License Server's MAC Address

The license server's Ethernet MAC address uniquely identifies your server to the NVIDIA Licensing Portal. You will need this address to register your license server with the NVIDIA Licensing Portal to generate license files.

1. Open a web browser on the license server host and connect to the URL `http://localhost:8080/licserver`.
2. In the license server management interface, select **Configuration**.
3. On the License Server Configuration page that opens, in the **Server host ID** drop-down list, select the platform's ETHERNET address.



6.2.4 Managing your License Server and Getting your License Files

To be able to download NVIDIA vGPU software licenses, you must create at least one license server on the NVIDIA Licensing Portal and allocate licenses to the server. After creating a license server and allocating licenses to it, you can download your license file.

6.2.4.1 Creating a License Server on the NVIDIA Licensing Portal

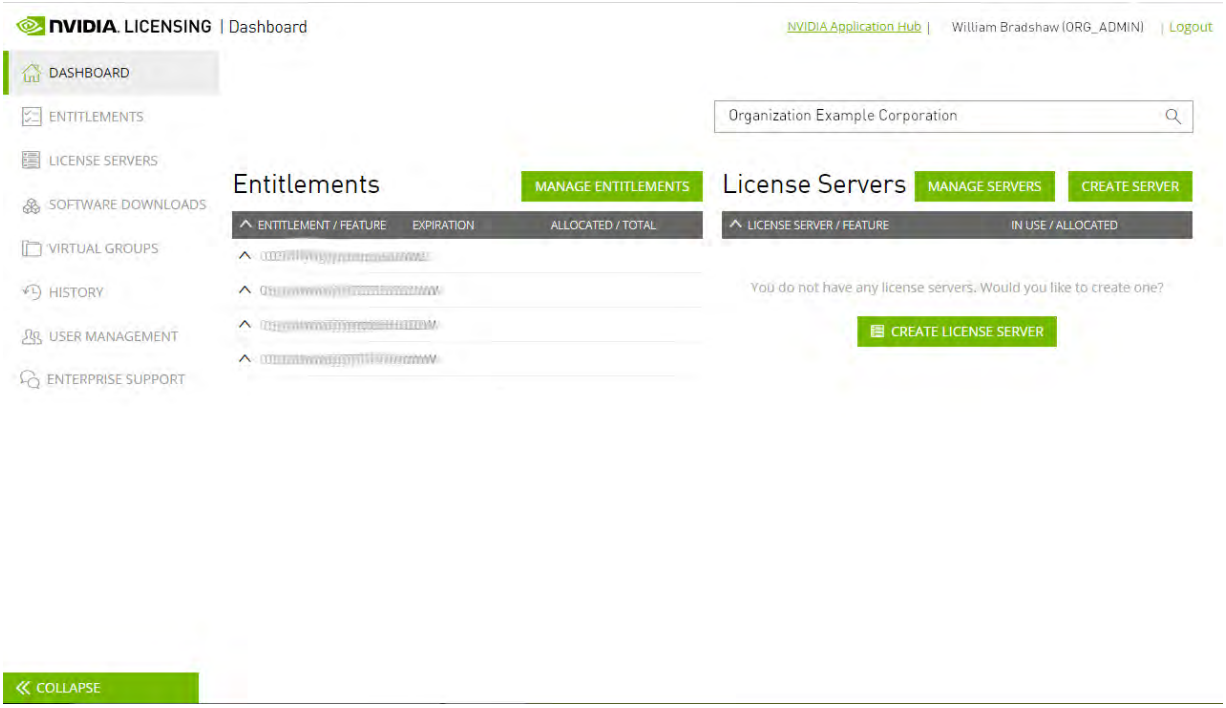
To be able to download NVIDIA vGPU software licenses, you must create at least one license server on the NVIDIA Licensing Portal. Creating a license server on the NVIDIA Licensing Portal registers your license server host with the NVIDIA Licensing Portal through the MAC address of the host.

1. In the NVIDIA Licensing Portal, navigate to the organization or virtual group for which you want to create the license server.

a) If you are not already logged in, log in to the NVIDIA Enterprise Application Hub and click **NVIDIA LICENSING PORTAL** to go to the NVIDIA Licensing Portal.

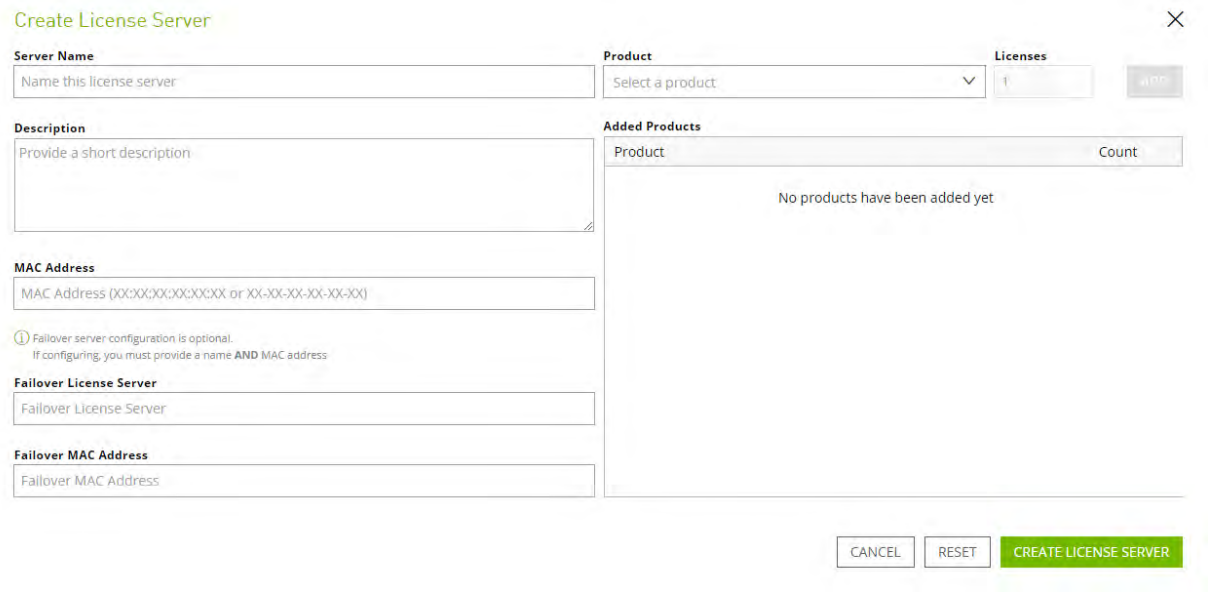
b) **Optional:** If your assigned roles give you access to multiple virtual groups, select the virtual group for which you are creating the license server from the list of virtual groups at the top right of the page.

If no license servers have been created for your organization or virtual group, the NVIDIA Licensing Portal dashboard displays a message asking if you want to create a license server.



2. On the NVIDIA Licensing Portal dashboard, click **CREATE LICENSE SERVER**.

   The Create License Server pop-up window opens.



3. Provide the details of your license server.
   a) In the **Server Name** field, enter the host name of the license server.
   b) In the **Description** field, enter a text description of the license server. This description is required and will be displayed on the details page for the license server that you are creating.
   c) In the **MAC Address** field, enter the MAC address of your license server.
4. Add the licenses for the products that you want to allocate to this license server. For each product, add the licenses as follows:
   a) From the **Product** drop-down list, select the product for which you want to add licenses.
   b) In the **Licenses** field, enter the number of licenses for the product that you want to add.
   c) Click **ADD**.
5. Leave the **Failover License Server** and **Failover MAC Address** fields unset.
6. Click **CREATE LICENSE SERVER**.

6.2.4.2 Downloading a License File

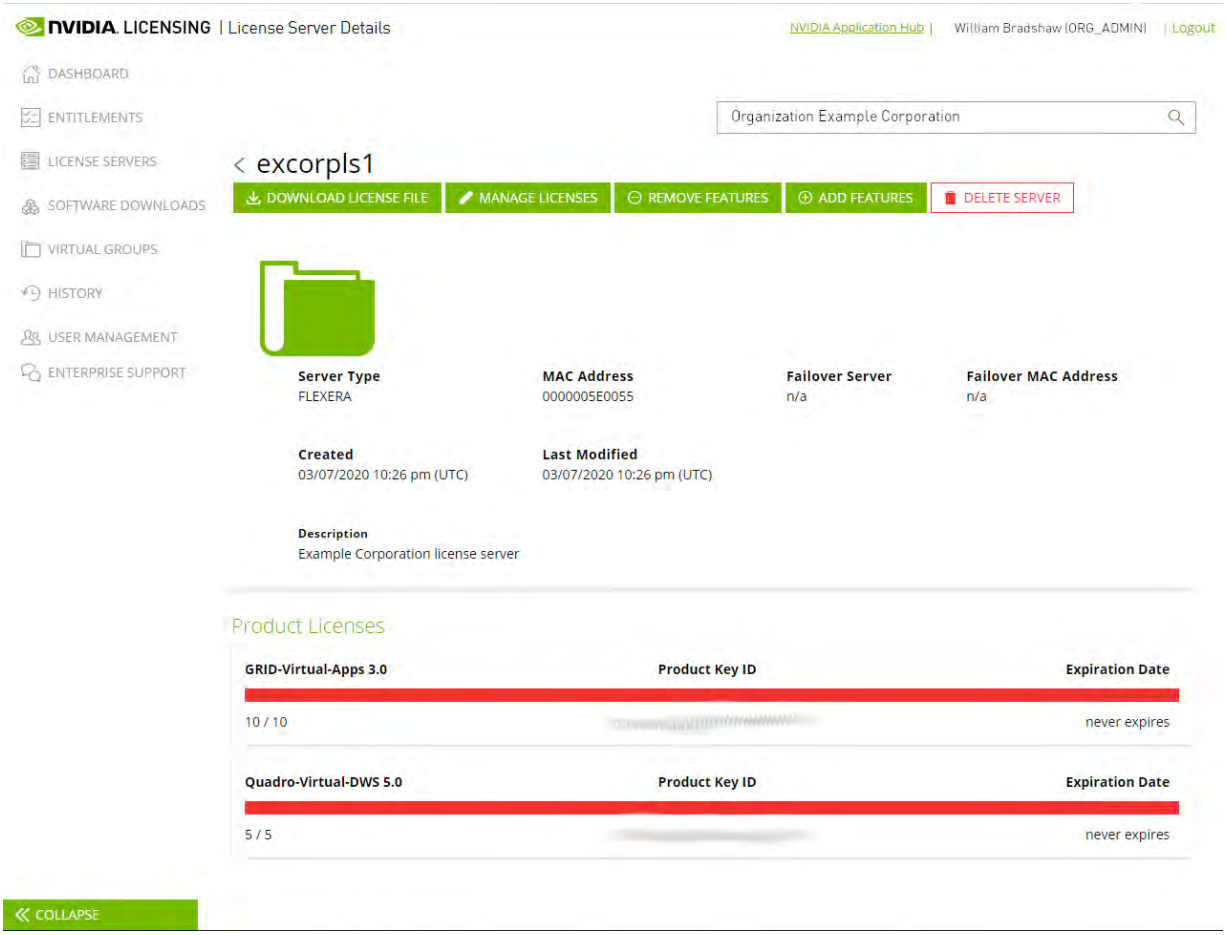
Each license server that you create has license file associated with it. The license file contains all the licenses that you allocated to the license server. After downloading the license file, you can install it on the license server host associated with the license server on the NVIDIA Licensing Portal.

1. In the NVIDIA Licensing Portal, navigate to the organization or virtual group for which you want to download the license file.
   a) If you are not already logged in, log in to the NVIDIA Enterprise Application Hub and click **NVIDIA LICENSING PORTAL** to go to the NVIDIA Licensing Portal.

b) **Optional:** If your assigned roles give you access to multiple virtual groups, select the virtual group for which you are downloading the license file from the list of virtual groups at the top right of the page.

2. In the list of license servers on the NVIDIA Licensing Portal dashboard, select the license server whose associated license file you want to download.

3. In the License Server Details page that opens, review the licenses allocated to the license server.



4. Click **DOWNLOAD LICENSE FILE** and save the .bin license file to your license server for installation.

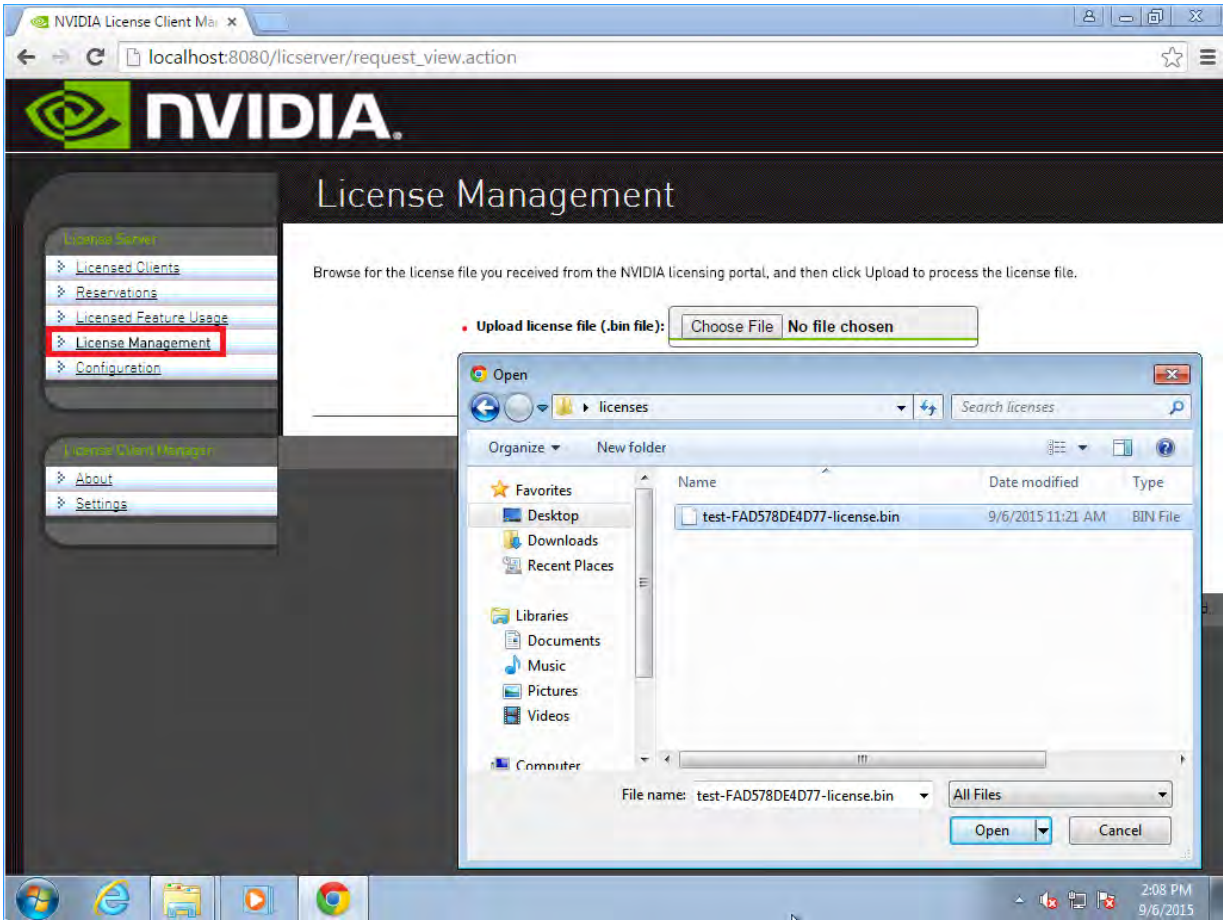
6.2.5 Installing a License

NVIDIA vGPU software licenses are distributed as .bin files for download from the NVIDIA Licensing Portal.

Before installing a license, ensure that you have downloaded the license file from the NVIDIA Licensing Portal.

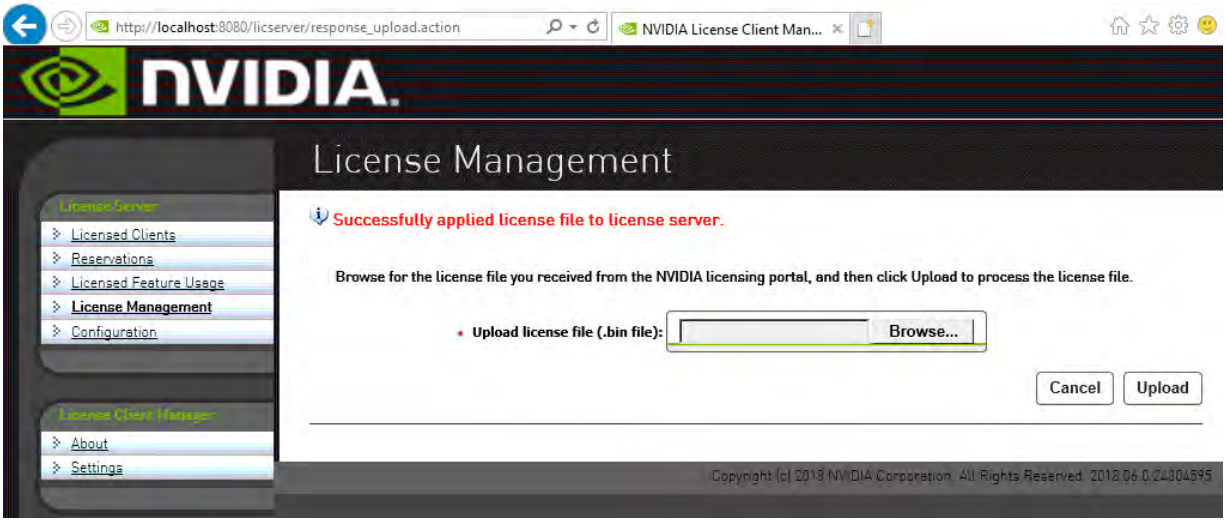
1. In the license server management interface, select **License Management**.

2. On the License Management page that opens, click **Choose File**.



3. In the file browser that opens, select the .bin file and click **Open**.

4. Back on the License Management page, click **Upload** to install the license file on the license server. The license server should confirm successful installation of the license file.



Note: For additional configuration options including Linux server deployment, securing your license server, and license provisioning, refer to the Virtual GPU Software License Server User Guide.

Chapter 7. Selecting the Correct vGPU Profiles

Choosing the right profile to maximize your stakeholders experience within the virtual instance is critical. Below, you will find guidance through the vGPU Manager and beyond to ensure your deployment is successful.

7.1 The Role of the vGPU Manager

NVIDIA vGPU profiles assign custom amounts of dedicated graphics memory to each virtual machine. NVIDIA vGPU Manager assigns the correct amount of memory to meet the specific needs within the workflow for the virtual machine user. Every virtual machine has dedicated graphics memory and must be assigned accordingly thus ensuring that it has the resources needed to handle the expected graphics load.

NVIDIA vGPU Manager allows multiple users to share each physical GPU by assigning the graphics resources of the available GPUs to virtual machines using a balanced approach. Depending on the number of GPUs within each NVIDIA card there can be multiple user types assigned.

7.2 The Full List of vGPU Profiles

vGPU profiles represent very flexible deployment options for virtual GPUs, varying the size of the allocated frame buffer memory depending on a number of factors, including the number and resolution of display heads. The division of frame buffer is what defines the number of users possible per GPU with that specific profile, while the number of heads defines the number of displays supported. Max resolution is consistent across all the profiles. The full list may also be found here: *https://docs.nvidia.com/grid/latest/grid-vgpu-user-guide/index.html*

Series	Optimal Workload
Q-series	Virtual workstations for creative and technical professionals who require the performance and features of RTX Enterprise Drivers.
C-series	Compute-intensive server workloads, such as artificial intelligence (AI), deep learning, or high-performance computing (HPC)
B-series	Virtual desktops for business professionals and knowledge workers
A-series	App streaming or session-based solutions for virtual applications users

Note: NVIDIA vGPU is a licensed product on all supported GPU boards. A software license is required to enable all vGPU features within the guest VM. The type of license required depends on the vGPU type.

- Q-series vGPU types require a RTX vWS license.
- C-series vGPU types require a NVIDIA Virtual Compute Server license but can also be used with a RTX vWS license.
- B-series vGPU types require a NVIDIA Virtual PC license but can also be used with a RTX vWS license.
- A-series vGPU types require a NVIDIA Virtual Applications license.

CAUTION: A NVIDIA Virtual Application license is required for Citrix Virtual Application deployments. A NVIDIA Virtual PC license is required for Citrix Virtual Desktop deployments.

Chapter 8. Creating Your First vGPU Virtual Desktop

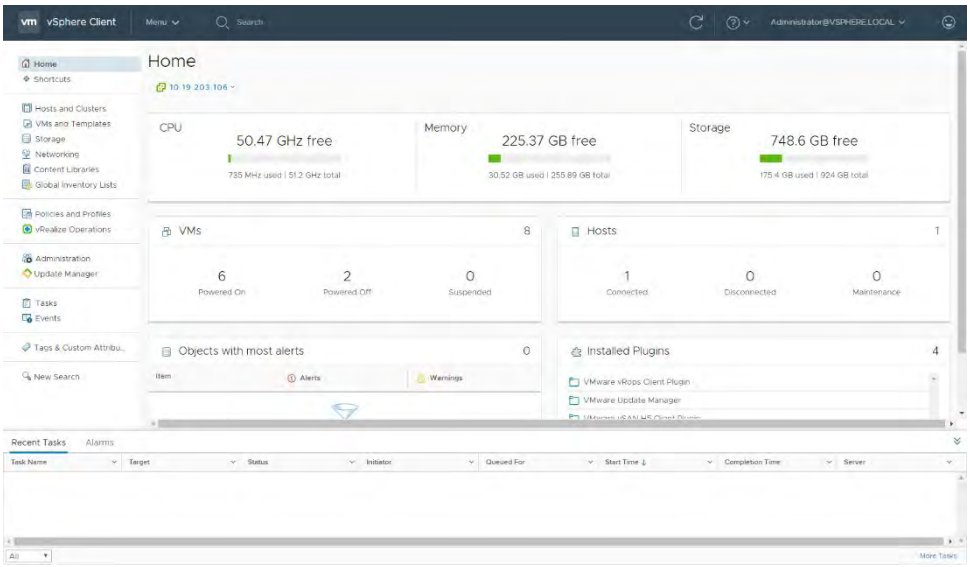
This chapter describes how to:

- Create and configure a virtual machine in vSphere
- Install Windows and VMware Tools on the VM
- Customize Windows settings
- Prepare the VM for use as the golden image for Citrix MCS or PVS
- Install Citrix Virtual Delivery Agent on the VM
- Adjust additional VM settings and enable VM console access
- Enable the NVIDIA vGPU and finalizing the installation

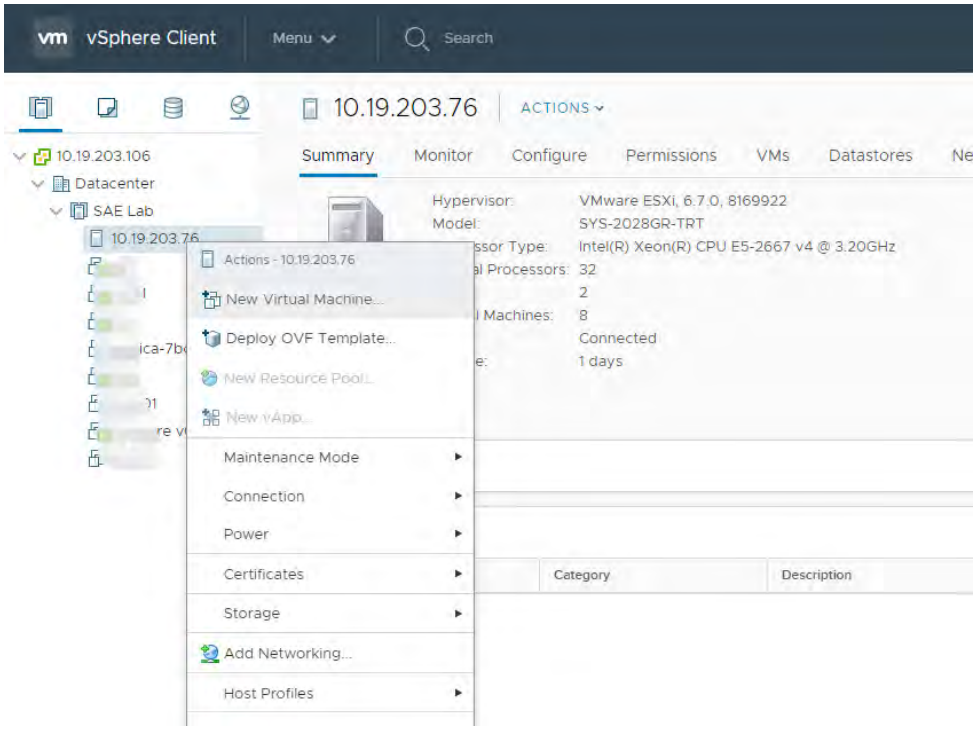
8.1 Creating a Virtual Machine

These instructions are to assist in making a VM from scratch that will support NVIDIA vGPU. Later the VM will be used as a gold master image. Use the following procedure to configure a vGPU for a single guest desktop:

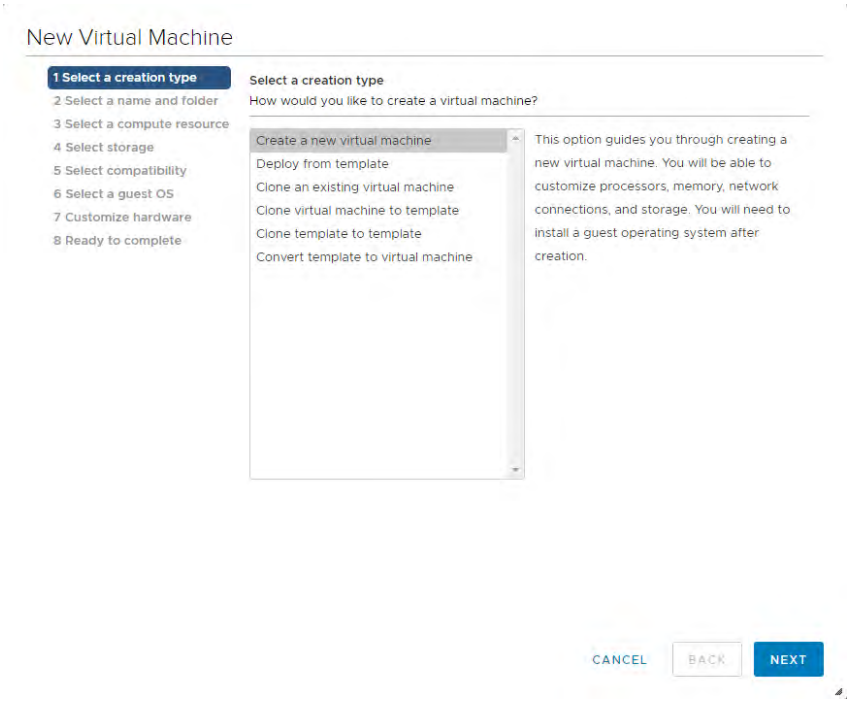
1. Browse to the host or cluster using the *vSphere Web Client*.



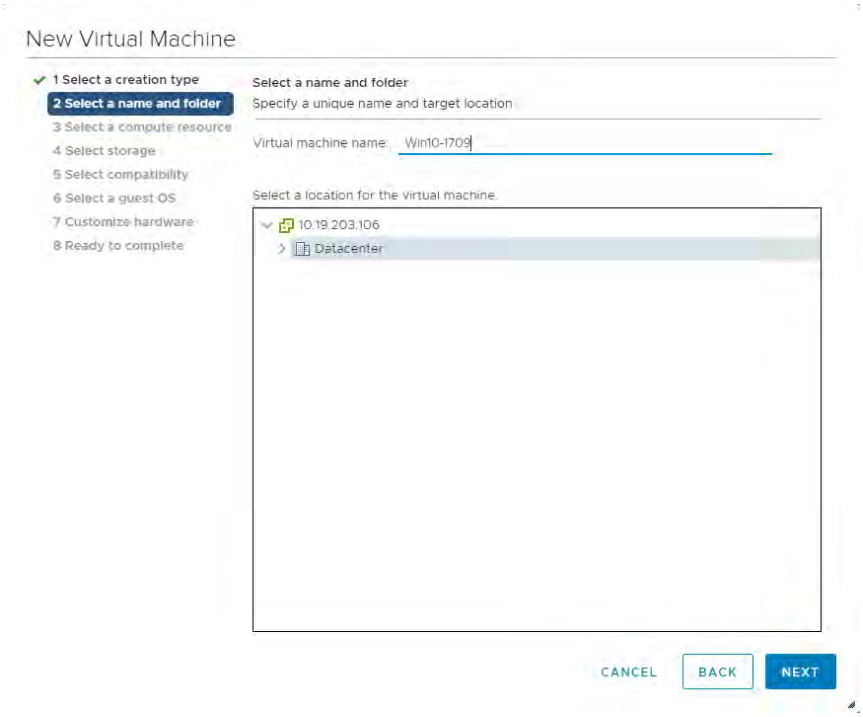
2. Right-click the desired host or cluster and select **New Virtual Machine**.
   The *New Virtual Machine* wizard begins.



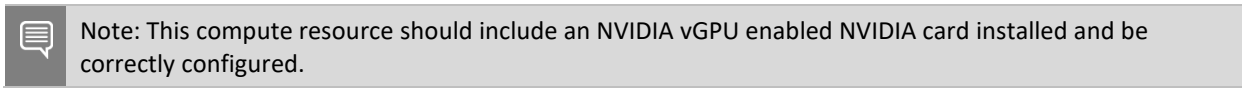
3. Select **Create a new virtual machine** and click **Next**.



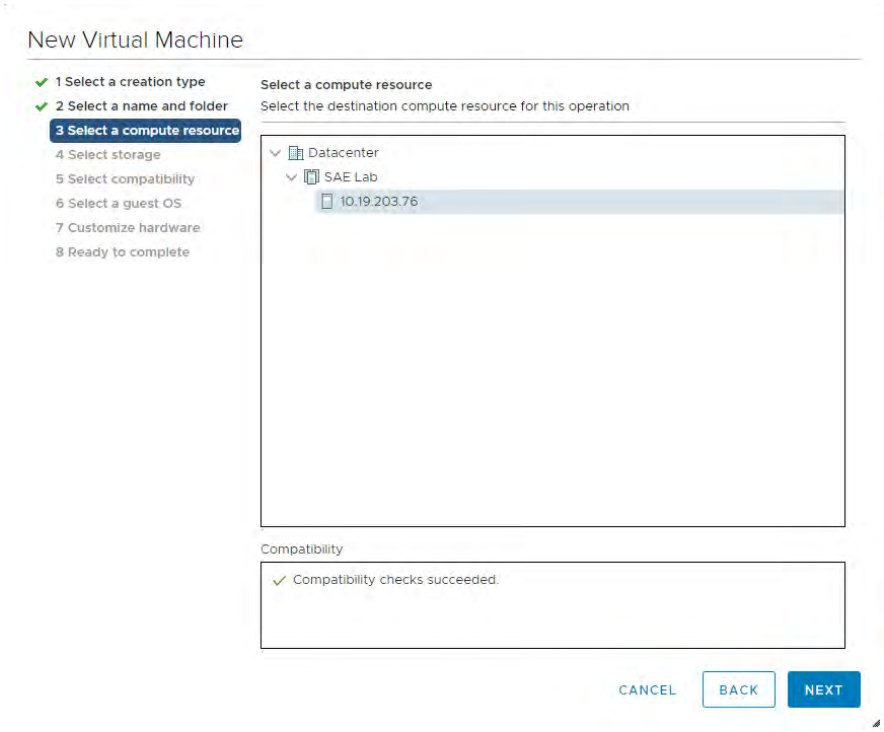
4. Enter a name for the virtual machine. Choose the location to host the virtual machine using the **Select a location for the virtual machine** section. Click Next to continue.



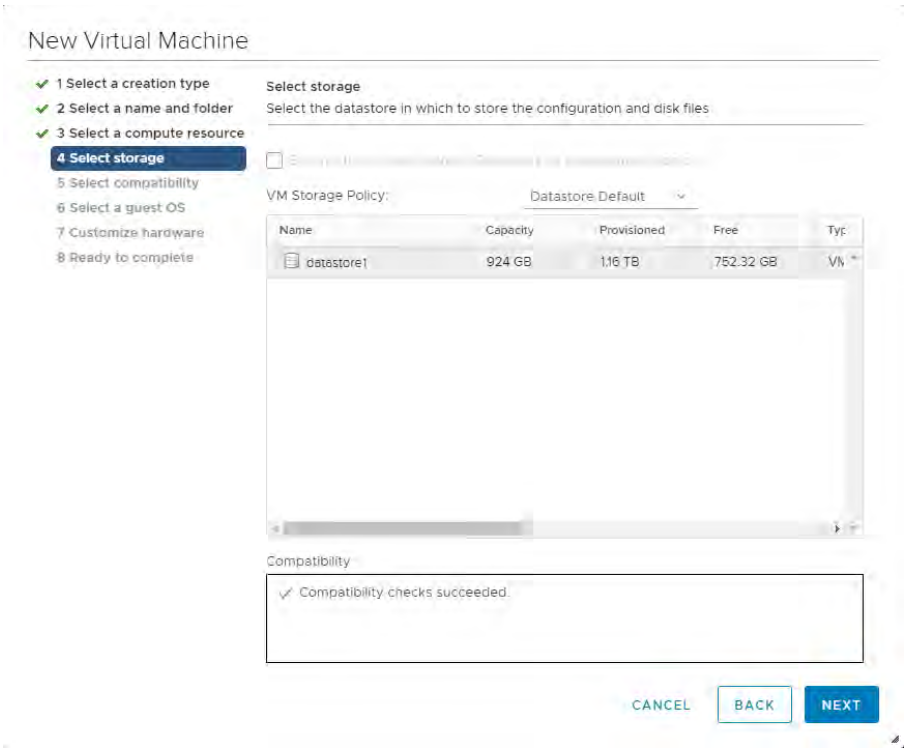
5. Select a compute resource to run the VM. Click Next to continue.



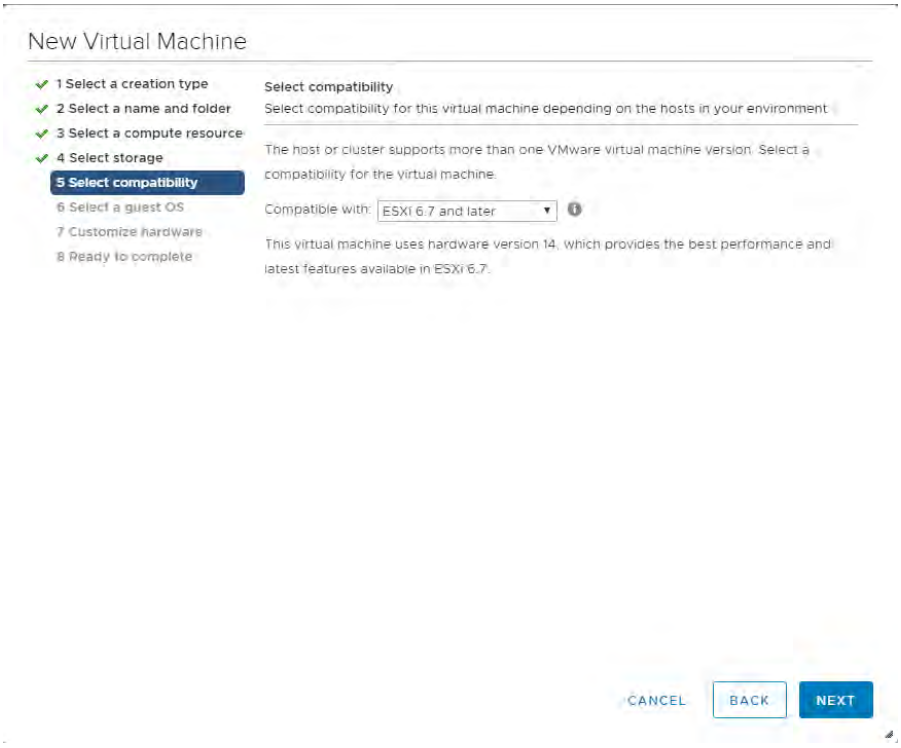
Note: This compute resource should include an NVIDIA vGPU enabled NVIDIA card installed and be correctly configured.



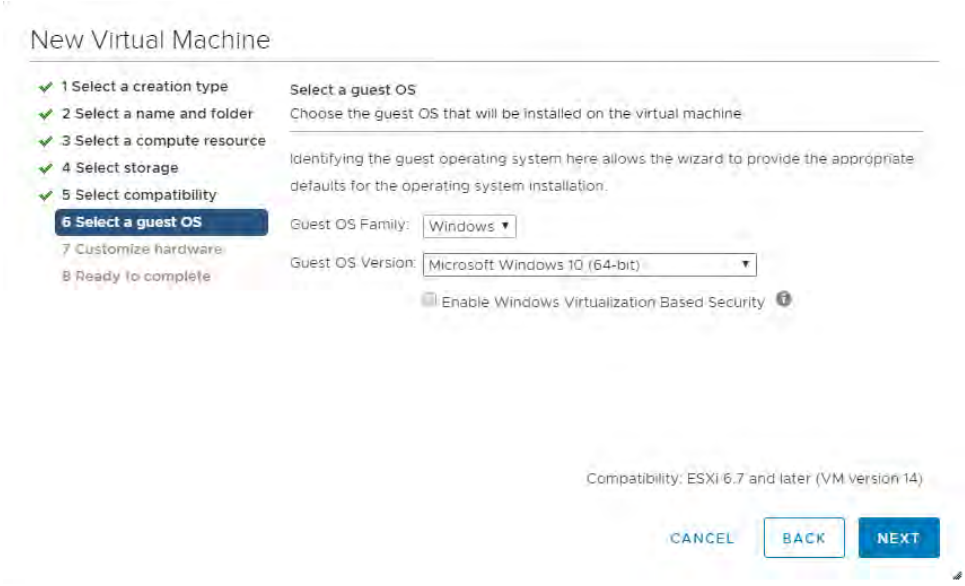
6. Select the datastore to host the virtual machine. Click **Next** to continue.



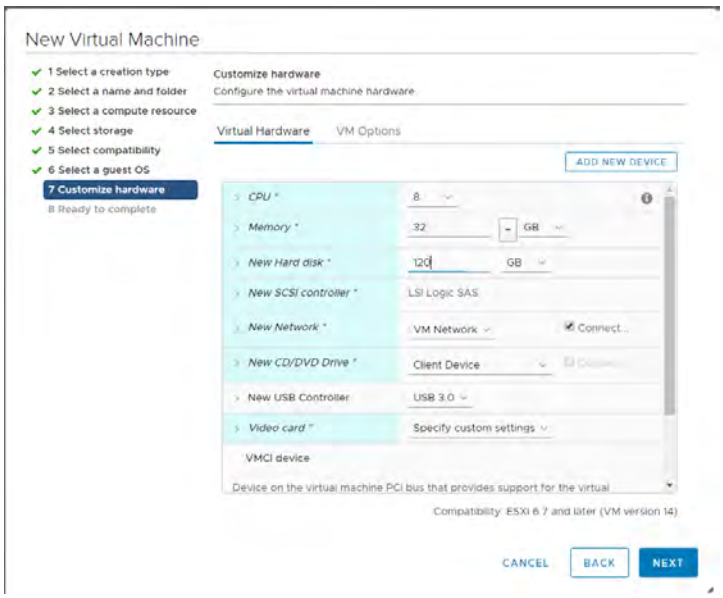
7. Select compatibility for the virtual machine. This allows VMs to run on different versions of vSphere. To run vGPU select ESXi 6.0 and later. Click **Next** to continue.



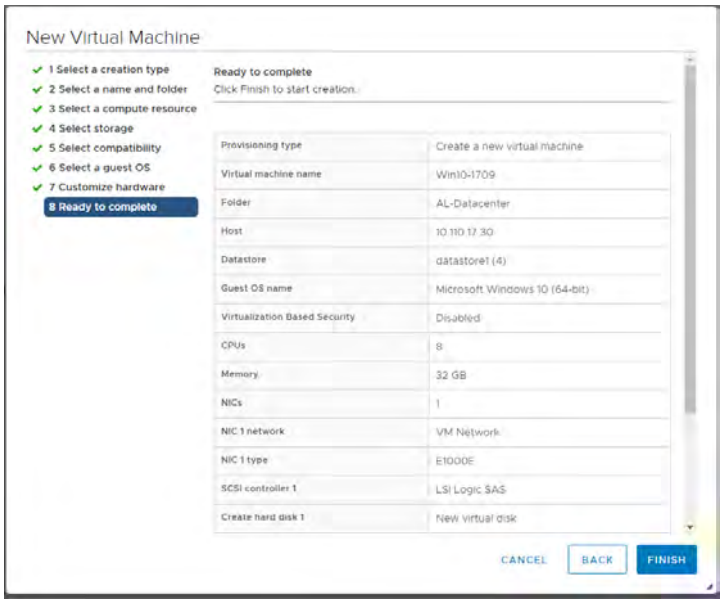
8. Select the appropriate Windows OS from the **Guest OS Family** and **Guest OS Version** pull-down menus. Click Next to continue.



9. Customize hardware is next. Set the virtual hardware based on your workload requirements. Click Next to continue.



10. Review the New Virtual Machine configuration prior to completion. Click **Finish** when ready.

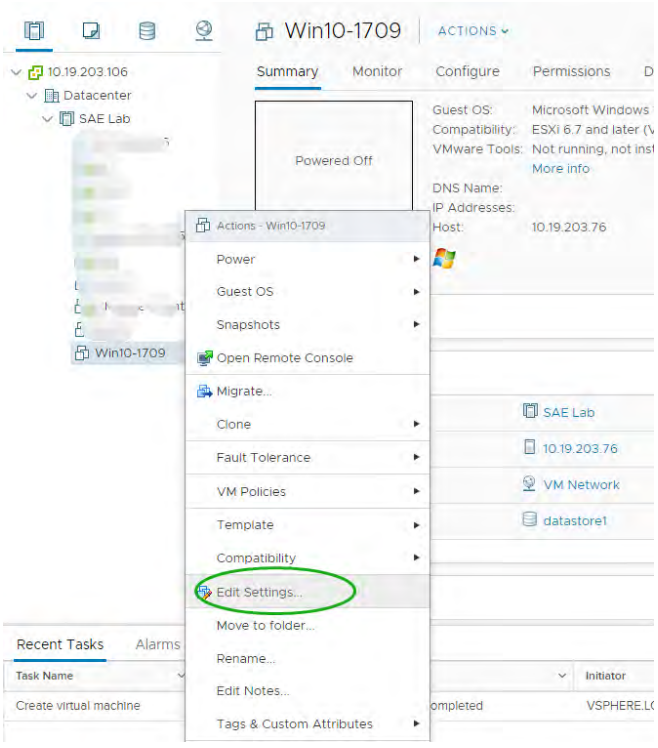


The new virtual machine has now been created.

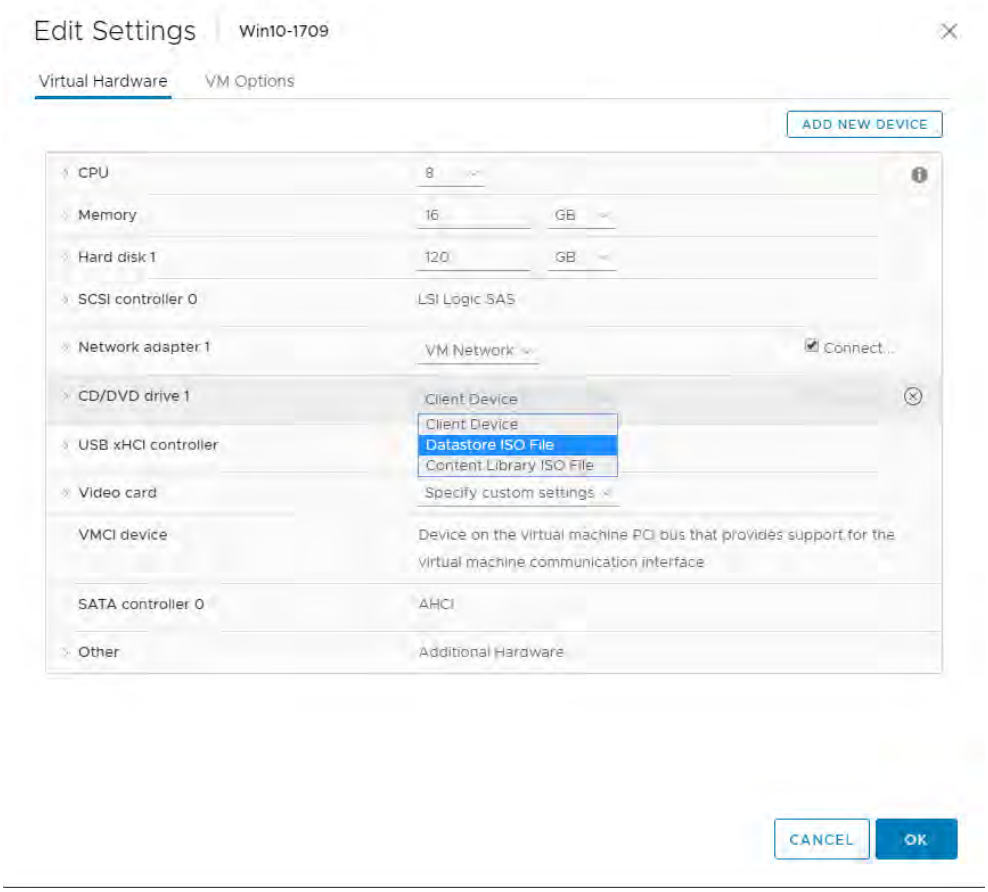
8.2 Installing Windows

Use the following procedure to install Windows on the virtual machine:

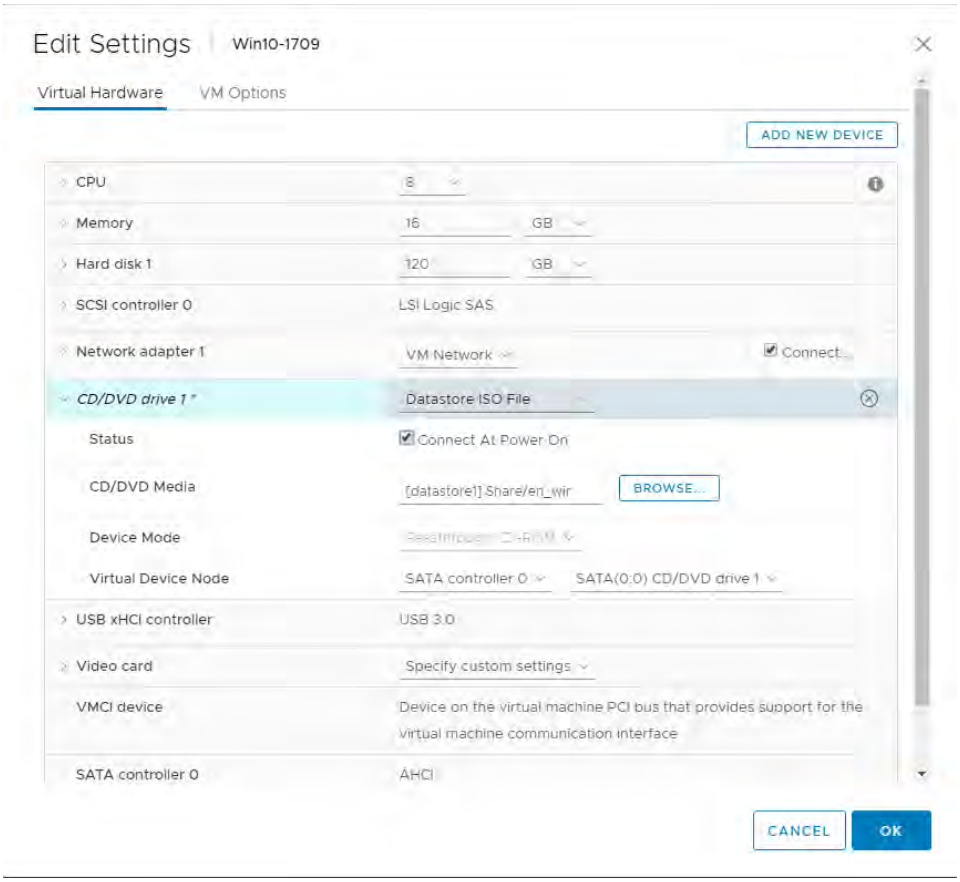
1. Select the virtual machine, right click on the virtual machine, and select Edit Settings.



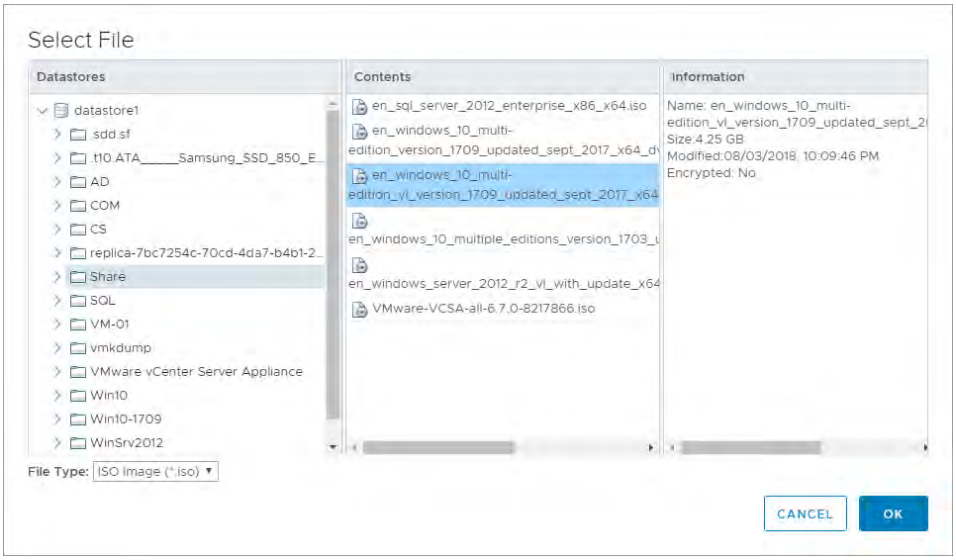
2. Locate the CD/DVD entry under the Virtual Hardware tab. Select the arrow drop down to reveal data sources for the CD/DVD media. (In this example a Datastore ISO file will be used.) Check the **Connect** checkbox for **CD/DVD drive 1**. This will connect the ISO file to the VMs virtual CD/DVD drive.



3. Toggle the carrot next to the CD/DVD drive 1 icon to reveal the details of the virtual device. For Status check the **Connect At Power On** checkbox. This will connect the ISO file to the VM's virtual CD/DVD drive during boot up. Next Click on the Browse button for the CD/DVD Media.



4. Navigate to and select the OS ISO file to be used for installation. Click OK to select the ISO.



5. Right-click the virtual machine, and then select **Power>Power On** to start the virtual machine and boot from the .ISO to install the operating system.

The virtual machine boots from the selected .ISO.

Note: If you are creating a new virtual machine and using the vSphere Web Client's VM console functionality, then the mouse may not be usable within the virtual machine until after the both the operating system and VMware tools have been installed.

6. Perform a Custom (fresh) installation of Windows 10 on the virtual machine.
   During installation, Windows reboots the VM several times.
7. Disconnect the .ISO from the VM when Windows is done installing.
8. Go through the initial Windows setup wizard to name the computer, create a local account, set the time-zone, choose update installation policy, etc.

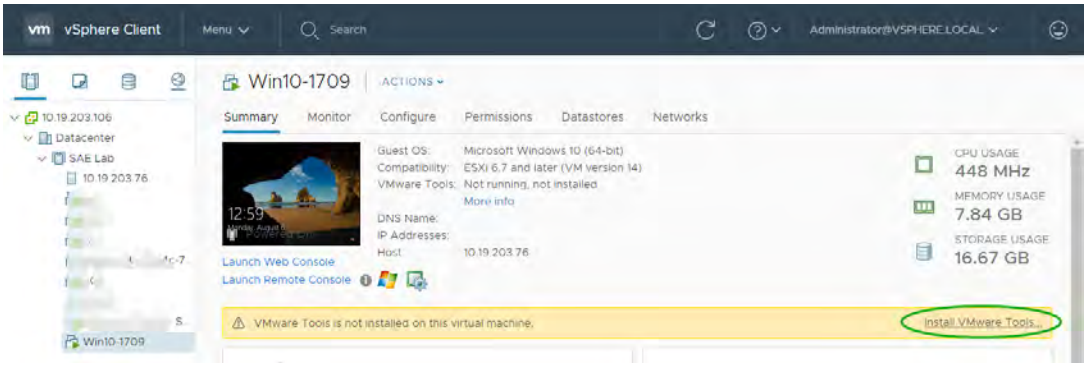
   Windows 10 is now installed on the virtual machine.

8.3 Installing VMware Tools on the VM

After Windows completes the initial installation and configuration process, the next step is to install VMware Tools on the virtual machine.

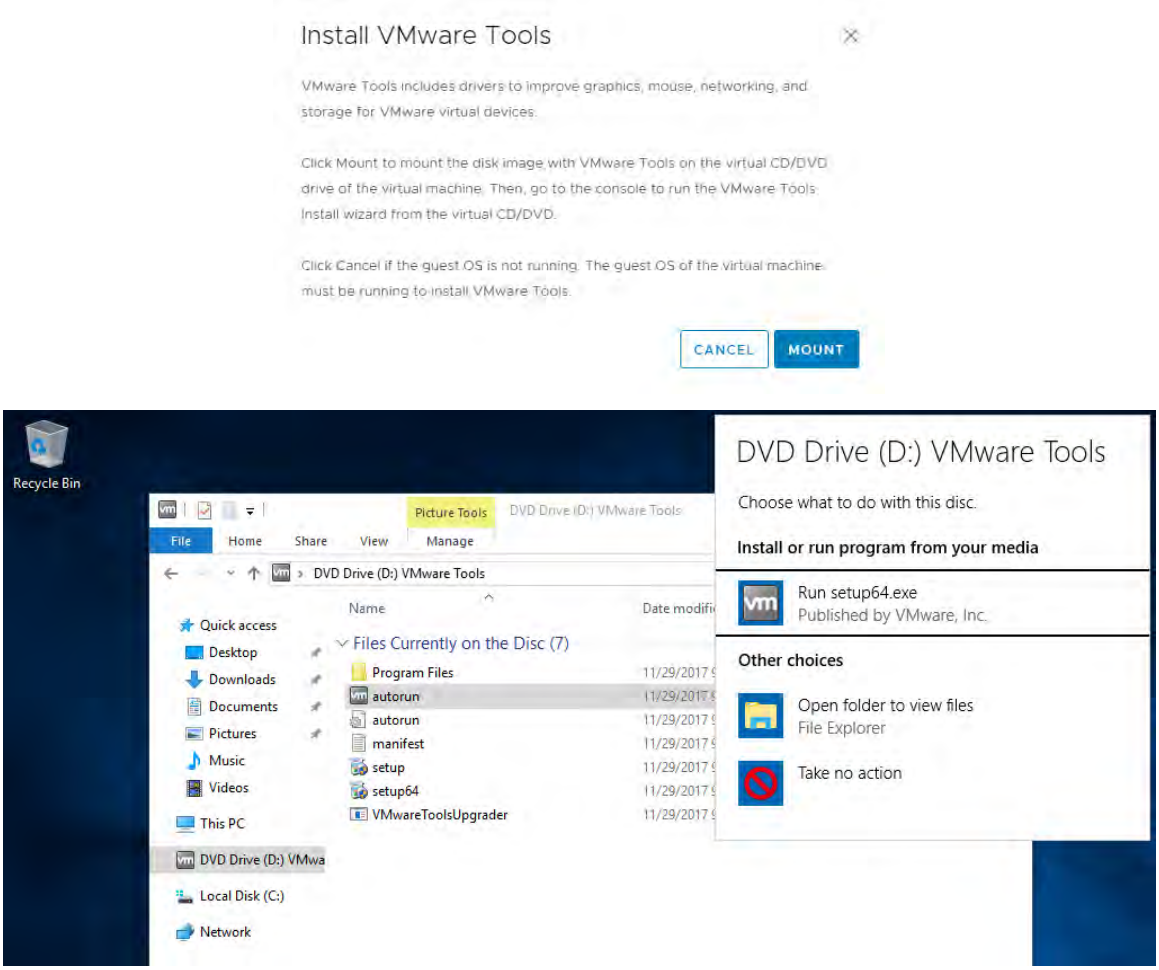
1. Select the **Summary** tab from the virtual machine console.
2. Click the **Install VMware Tools** link in the yellow bar. VM must be running to install VMware tools

The *Install VMware Tools* window displays.



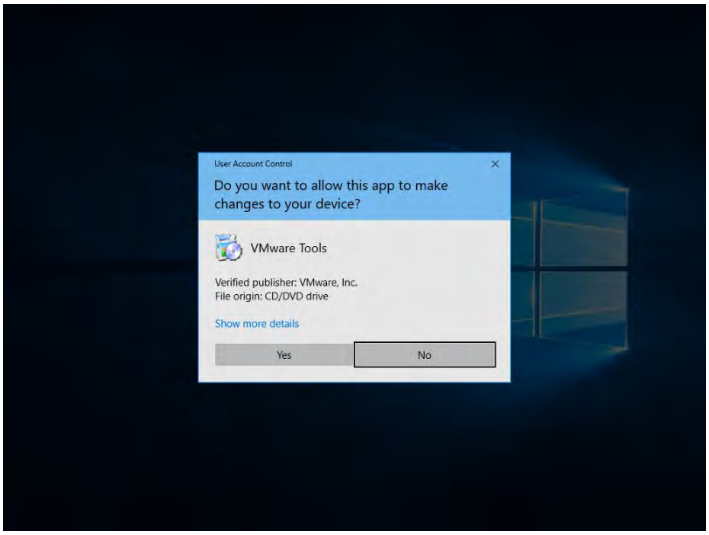
3. Click **Install VMware Tools**.

Back in the virtual machine console, Windows 10 detects the CD image and the *AutoPlay* window should open. If not, browse to the virtual CD-ROM in the virtual machine and access it manually.

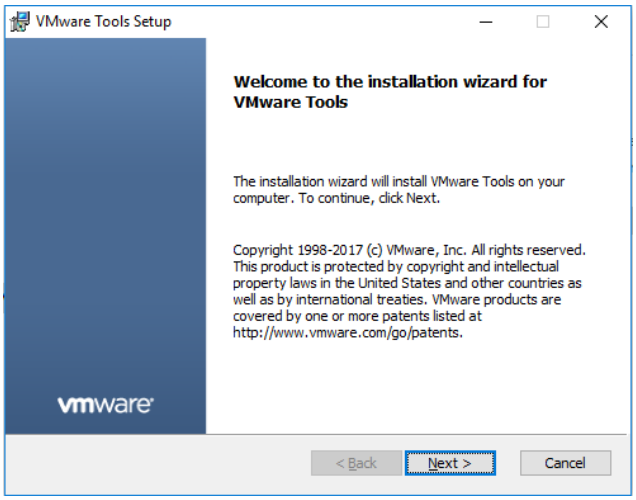


4. Click **Run setup.exe or setup64.exe**.

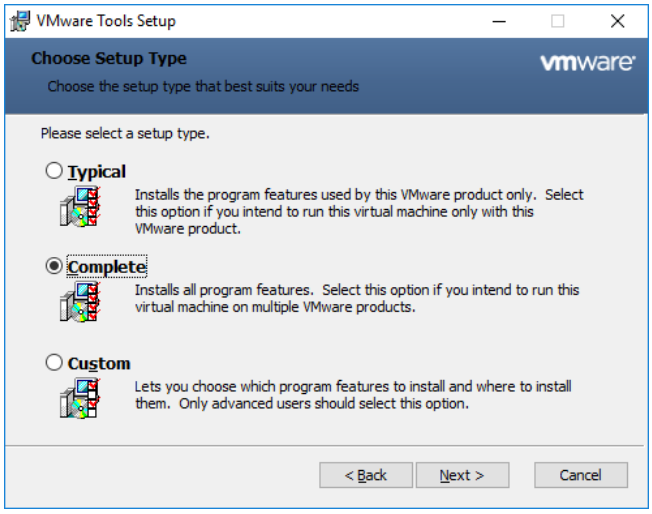
The *User Account Control* popup may display.



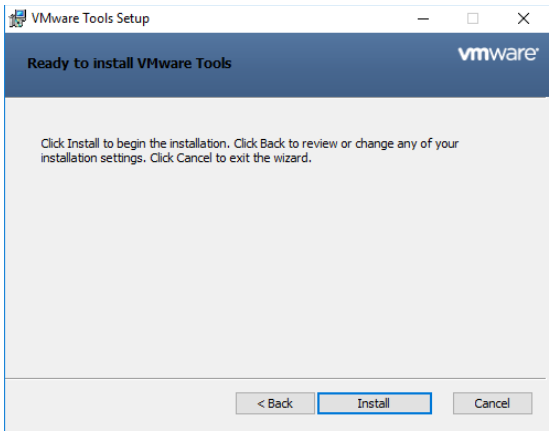
5. If UAC prompts, then click **Yes**.
6. The installer begins, click **Next.**



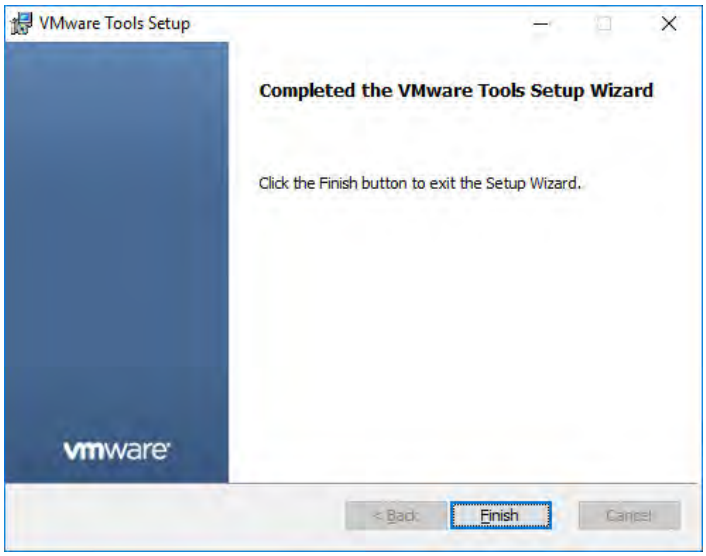
7. When prompted, select **Complete** installation, click **Next**, and accept all defaults.



8. When prompted, select **Install** to begin installation:



9. Click **Finish** in the VMware Tools installer and reboot the virtual machine when prompted. This reboot is critical to ensure the tools are now the loaded drivers.



10. VMware Tools is now installed on the virtual machine

8.4 Adding the VM to the Domain

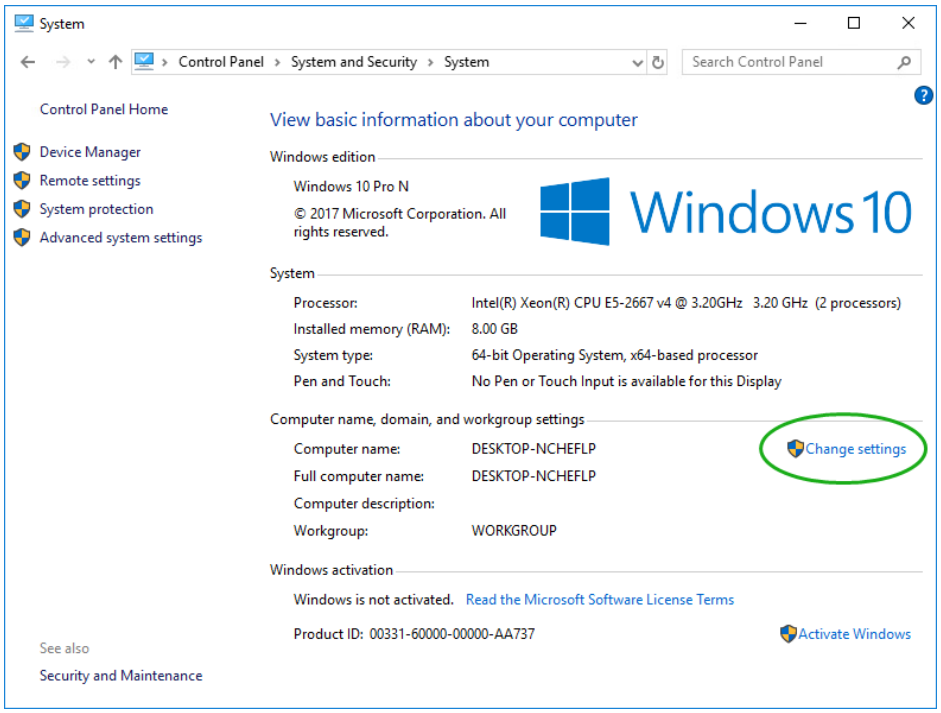
By joining the VM to the Windows Active Directory domain you are then able to manage it as you would any physical desktop in the domain.

Customize Windows on the virtual machine as follows:

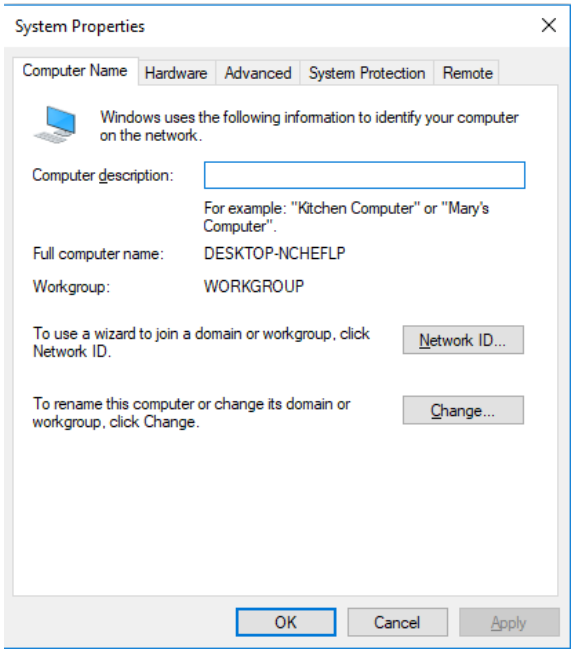
- Join the domain
- Add appropriate Domain groups to Local Administrators

Adding a VM to the domain:

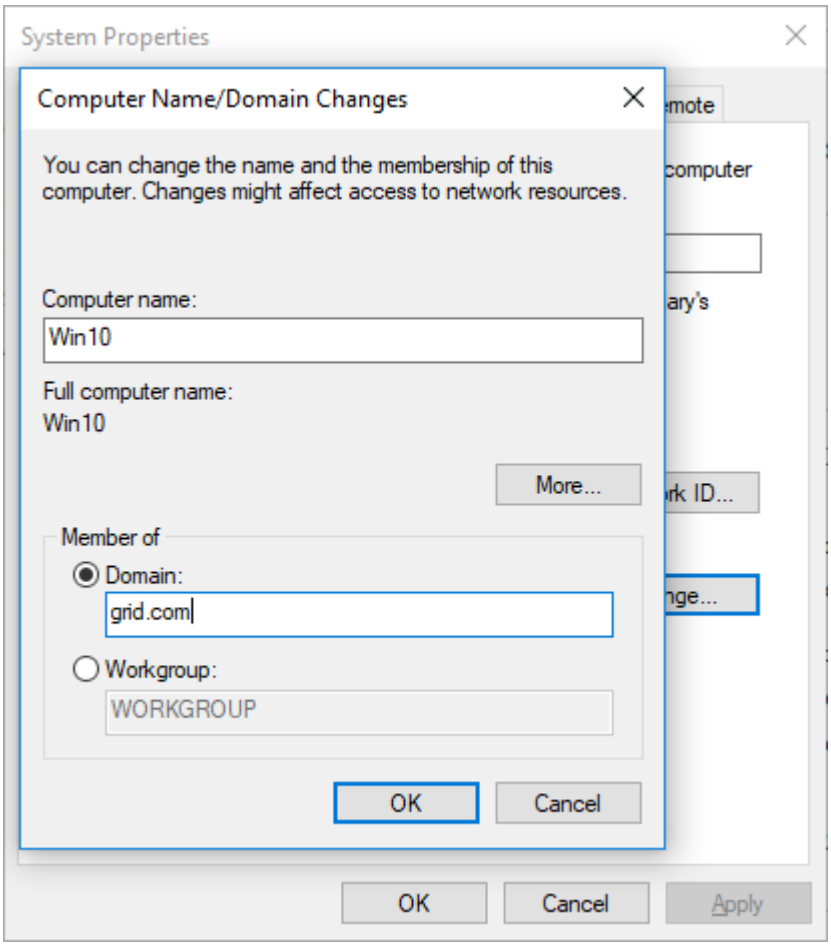
1. On the VM, go to **Control Panel**, **System and Security**, **System**



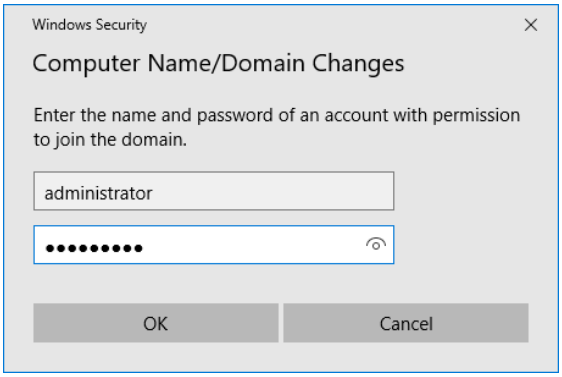
2. This brings up the System Properties window, on the Computer Name tab click Change:



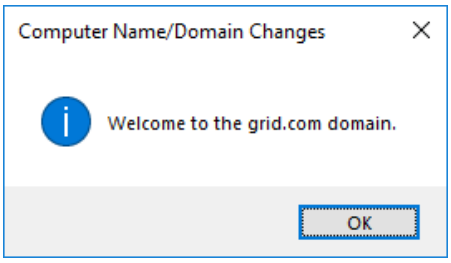
3. On the Computer Name/Domain Changes window, enter in an appropriate Computer name, then Domain name, and click OK. Our chosen naming is shown below, use what is appropriate for your POC/trial.



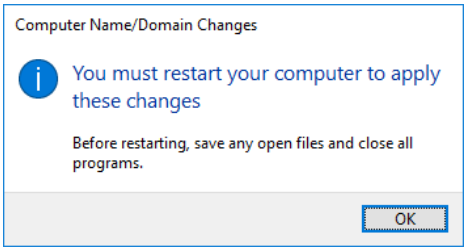
4. A security window pops up, fill in your specific domain administrator credentials and click OK:



5. On successful authentication you will see the following welcome pop-up showing your VM is now on the domain (the domain name should reflect your domain information):



6. Click OK and the VM needs to reboot to complete the process, click OK again and the VM reboots immediately.



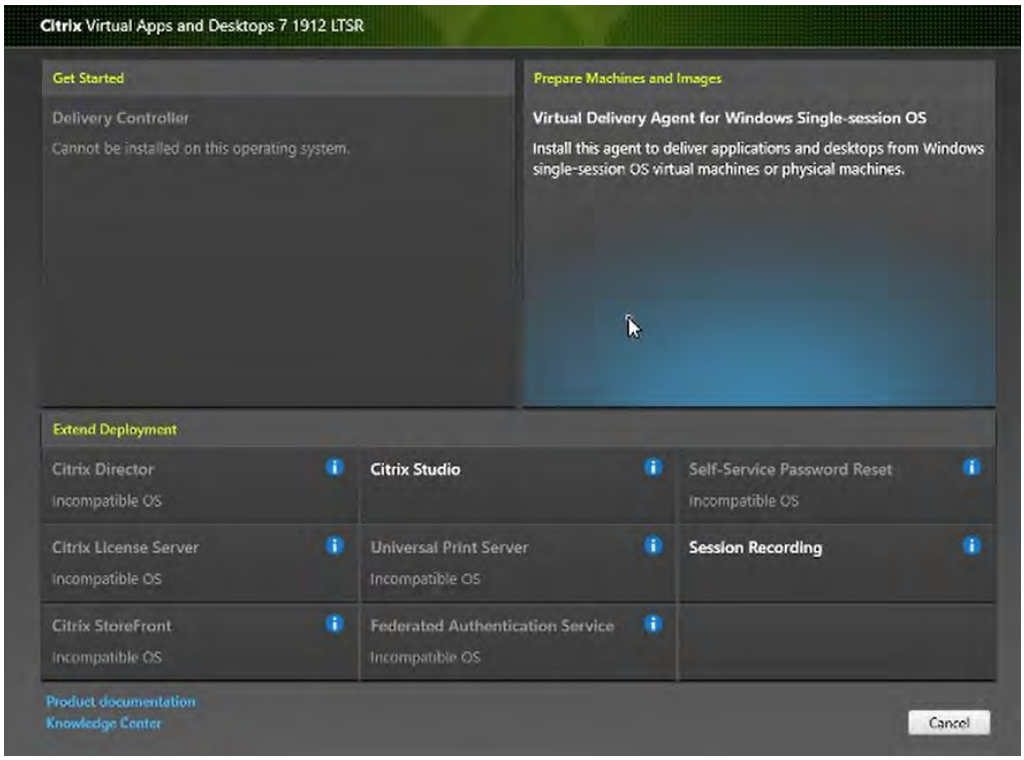
8.5 Installing the Citrix Virtual Delivery Agent

You need to install the correct version of the Citrix Virtual Delivery Agent (VDA) for your virtual machine. For the purpose of this guide, Citrix Virtual Apps and Desktop LTSR 7_1912 was used, therefore this guide uses VDA agent within the exact same LTSR service branch version.

Use the following procedure to install the Virtual Delivery Agent:

1. Attach the iso file to the server OS and open it via File Explorer.
2. Launch the ***AutoSelect*** Application and accept the Windows User Account Control Popup.
3. Click **Start** for ***Virtual Apps and Desktops*** section.

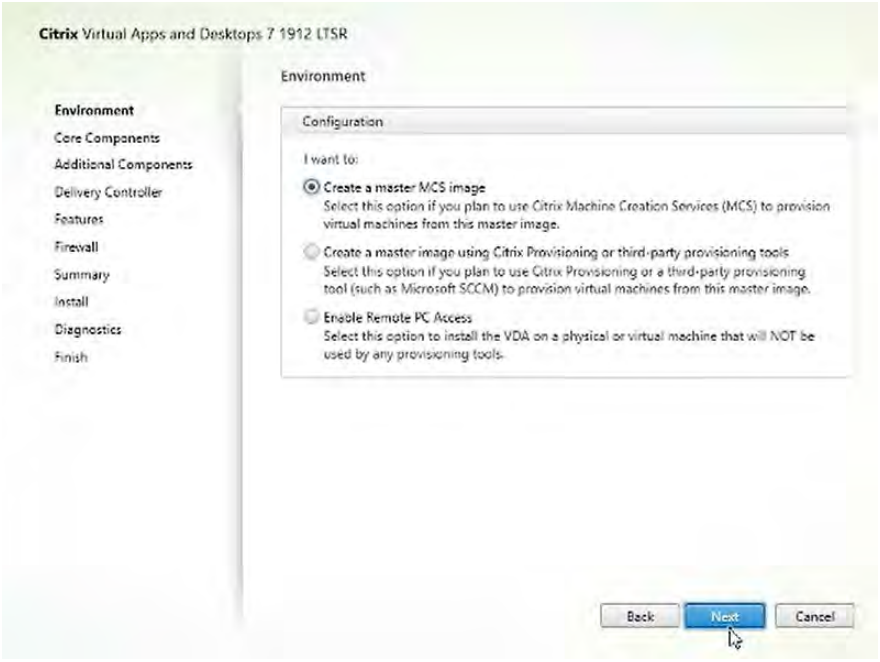
4. Select **Virtual Delivery Agent for Windows Single-session OS** to launch the Citrix Virtual Delivery Agent installer.



Note: If you are using an Operating System that supports multiple user sessions, you will select Virtual Delivery Agent for Windows Multi-session OS.

5. On the environment window, ensure the **Create a master MCS image** radio button is selected.

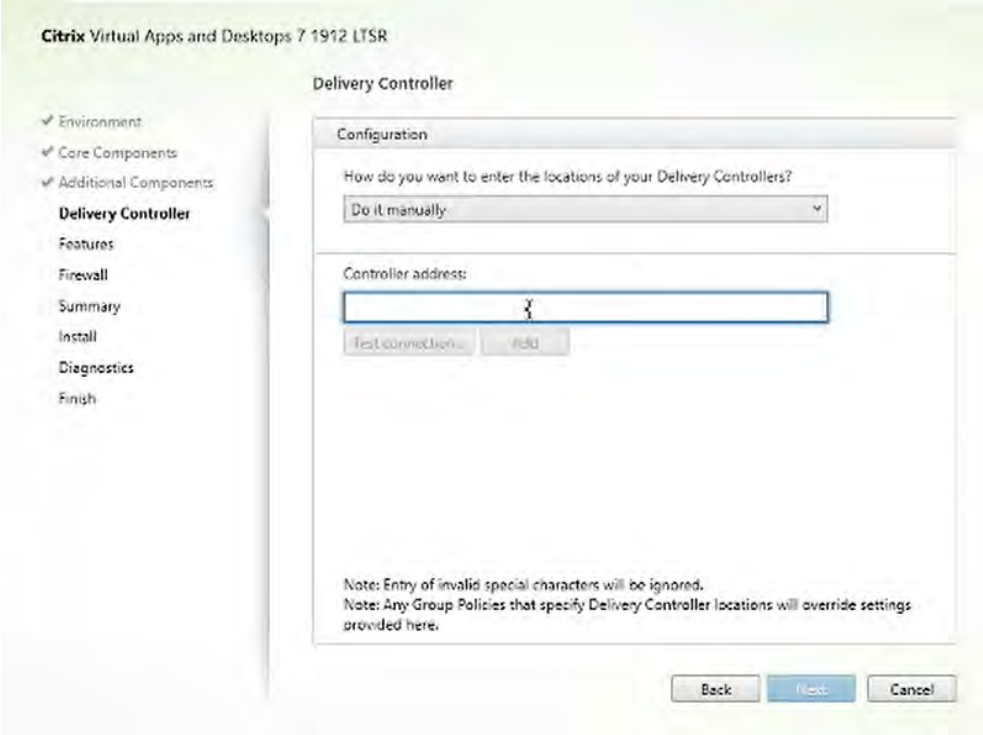
Note: In a production environment you may use a different VM provisioning method.



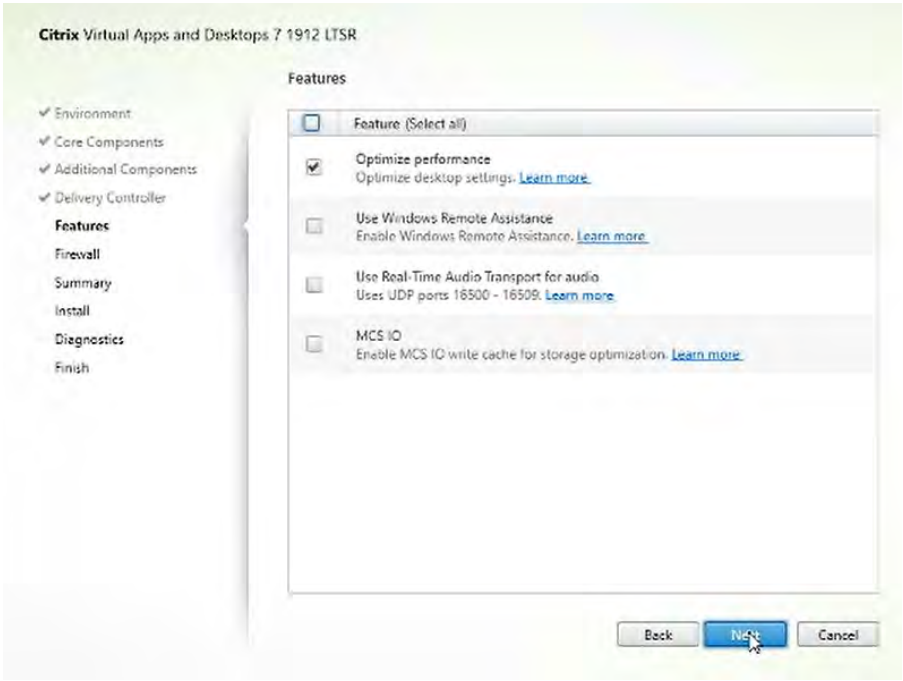
6. On the Core Components window, you can change the install location or leave the default location and click **Next.**

7. The Additional Components window allows you to choose which Components to install. Leave the defaults selected and click **Next.**

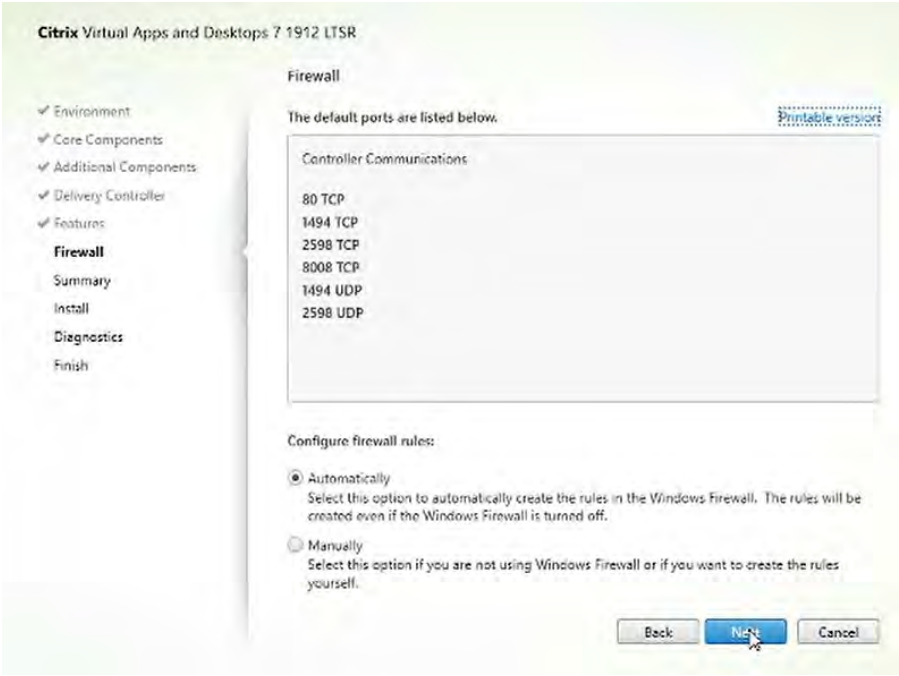
8. On the Delivery Controller window, type in the Fully Qualified Domain Name of your Delivery Controller in the **Controller address:** text field.



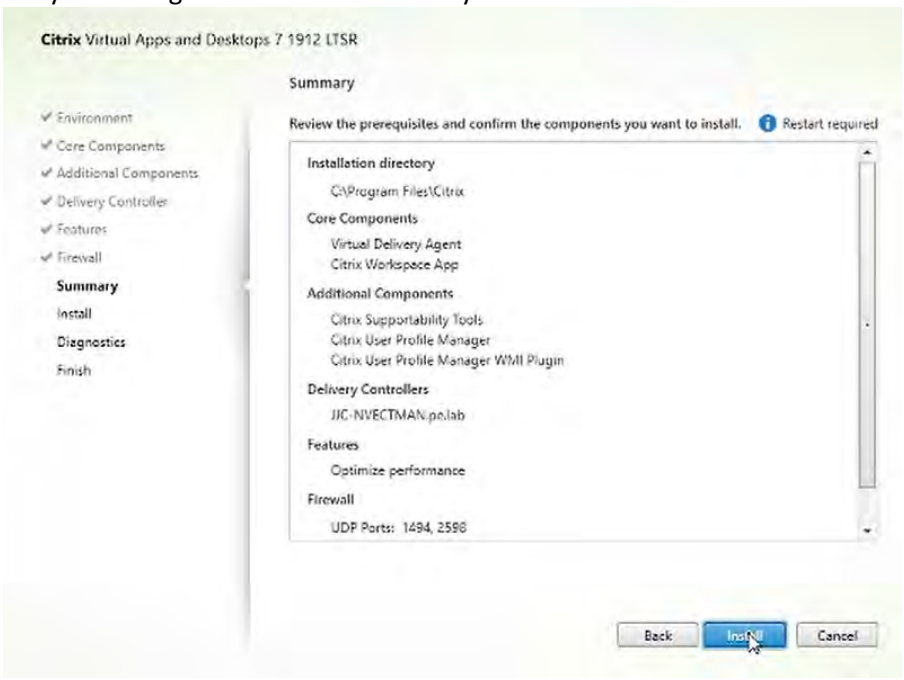
9. Click **Test connection..** then click **Add**.
10. Click **Next**.
11. The Features window allows you to choose which Features to install. Leave the defaults selected and click **Next**.



12. The Firewall window allows you to configure Windows Firewall. Select the **Automatically** radio button and click **Next**.

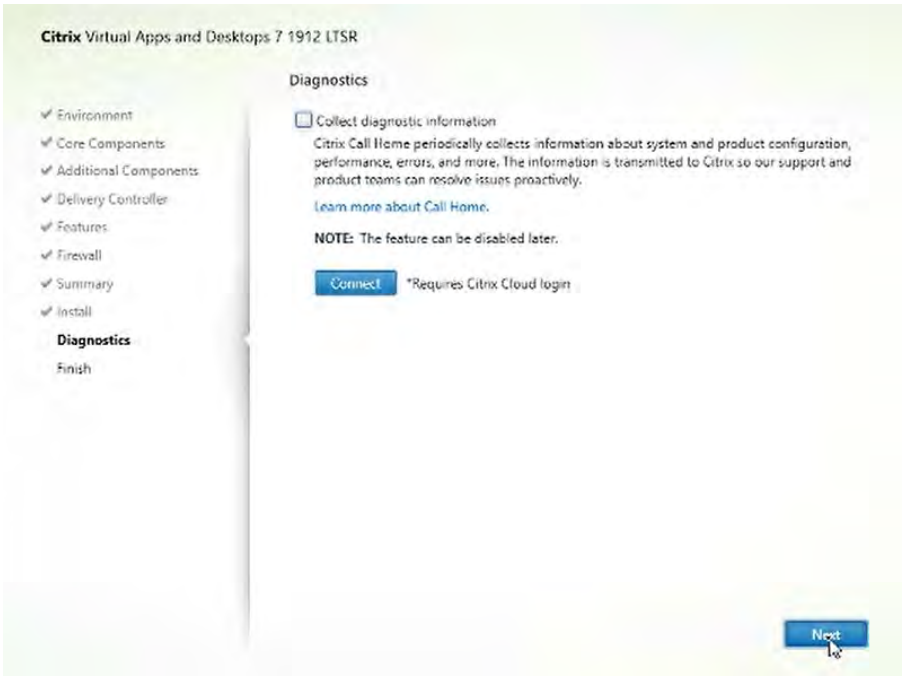


13. Review your configuration on the Summary window and click **Install**



14. Accept any ***Reboot Prompts*** and reconnect to the server.

15. In the Diagnostics window, uncheck ***Collect diagnostic information***.



16. Click **Next** to continue.
17. On the Finish window, click **Finish** to complete the install and allow the server to restart.

8.6 Additional Virtual Machine Settings

Perform the following additional tasks on the virtual machine as required:

Note: To evaluate browser based HTML5 applications, consider using newer browsers that utilize hardware acceleration within virtual desktop environments.

- Turn Off Windows Firewall for all network types.

CAUTION: THESE INSTRUCTIONS ASSUME THAT THE VM IS BEING USED AS A PROOF-OF-CONCEPT ONLY AND THAT DISABLING THE FIREWALL WILL THEREFORE POSE ONLY A MINIMAL SECURITY BREACH. ALWAYS FOLLOW YOUR ESTABLISHED SECURITY PROCEDURES AND BEST PRACTICES WHEN SETTING UP SECURITY FOR A PRODUCTION MACHINE OR ANY ENVIRONMENT THAT CAN BE ACCESSED FROM OUTSIDE YOUR NETWORK.

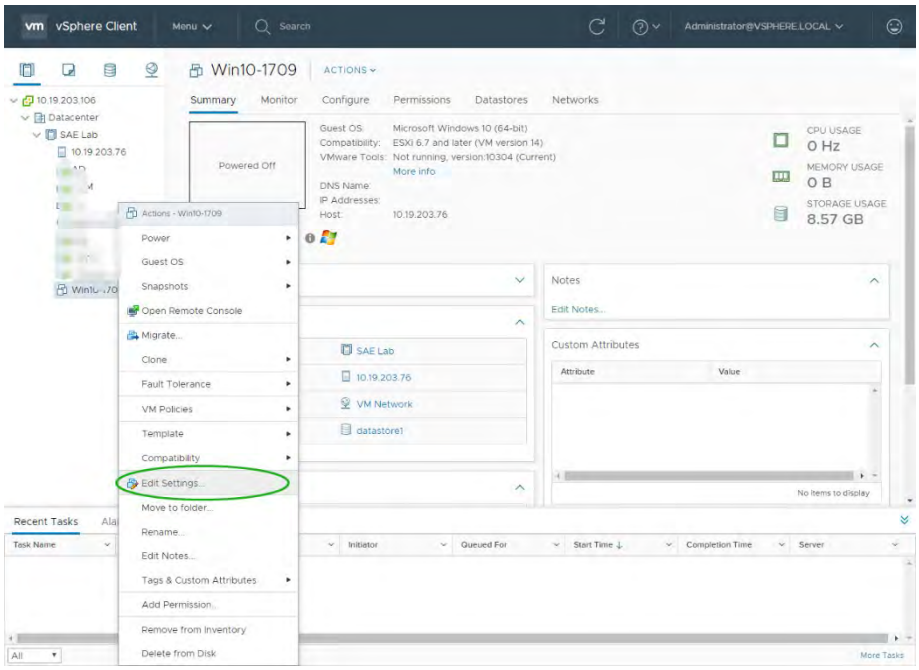
- Shut down the virtual machine once this is completed
- **Close the remote console; this will not be functional when vGPU is configured.**

Note: Take a snapshot of the virtual machine to preserve your work. Label this snapshot pre-vGPU and revert to it if you encounter any problems going forward, such as driver issues.

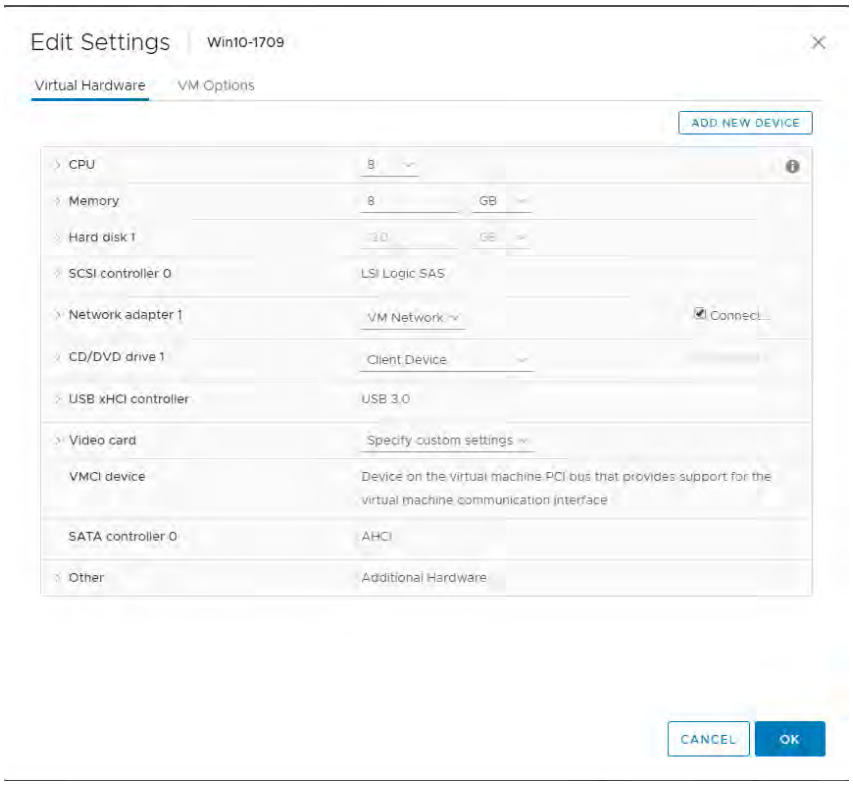
8.7 Enabling the NVIDIA vGPU

Use the following procedure to enable vGPU support for your virtual machine (you must edit the virtual machine settings):

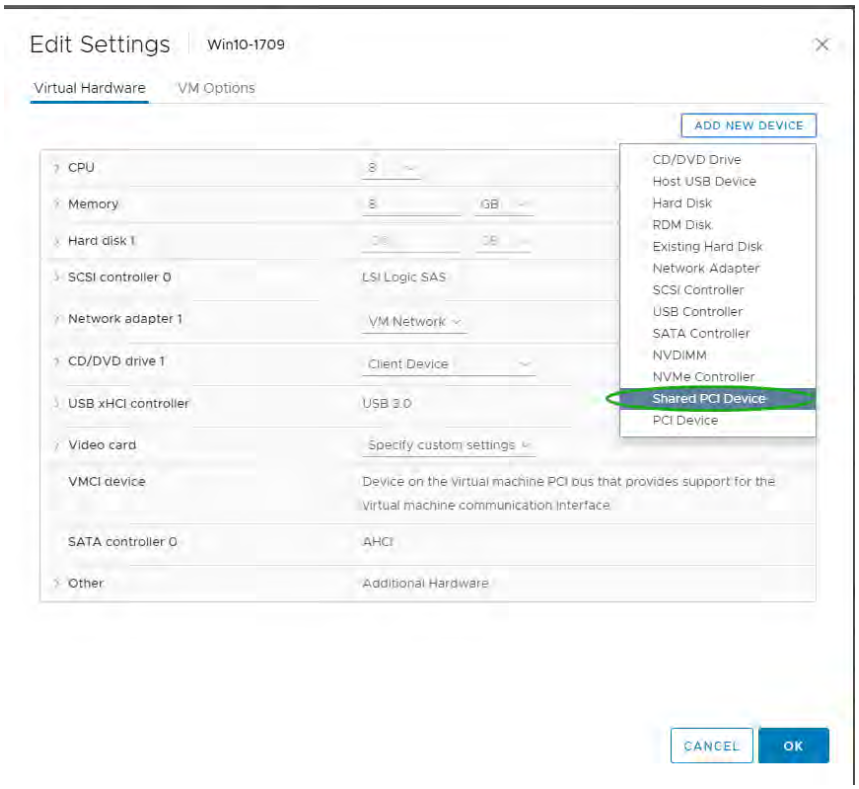
1. Power down the virtual machine.
2. Click on the VM in the Navigator window. Right click the VM and Edit Settings.



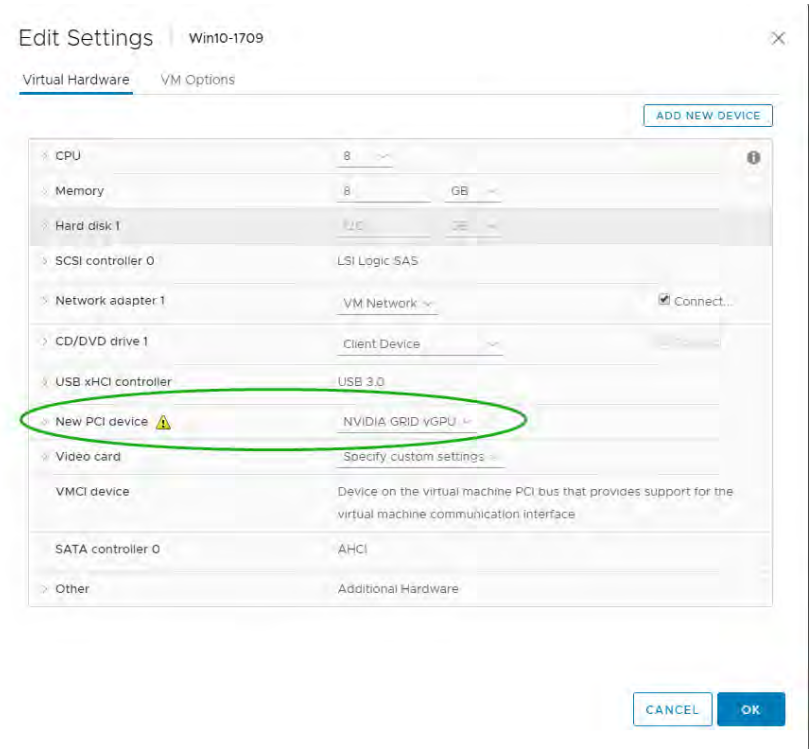
3. The Edit Settings dialog appears.



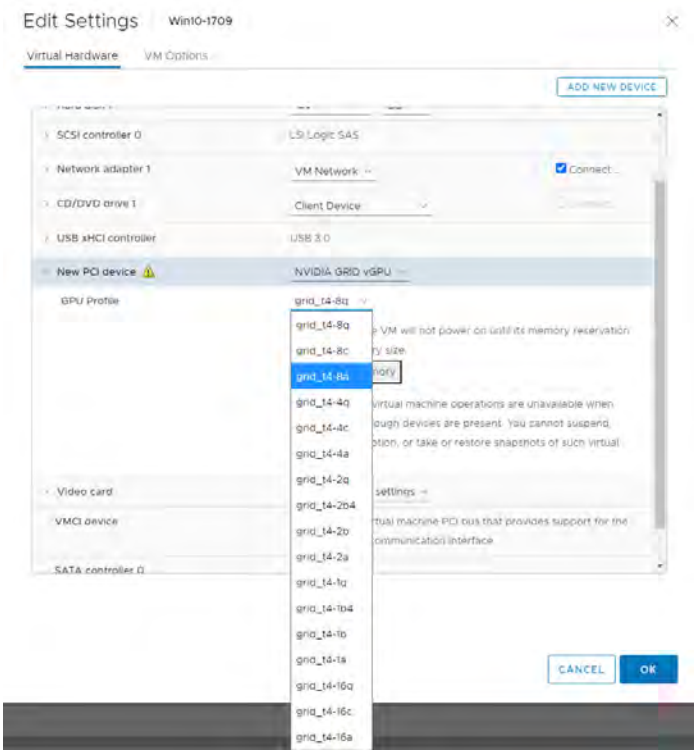
4. Click on the New Device bar and select Shared PCI device.



5. Click on Add to continue

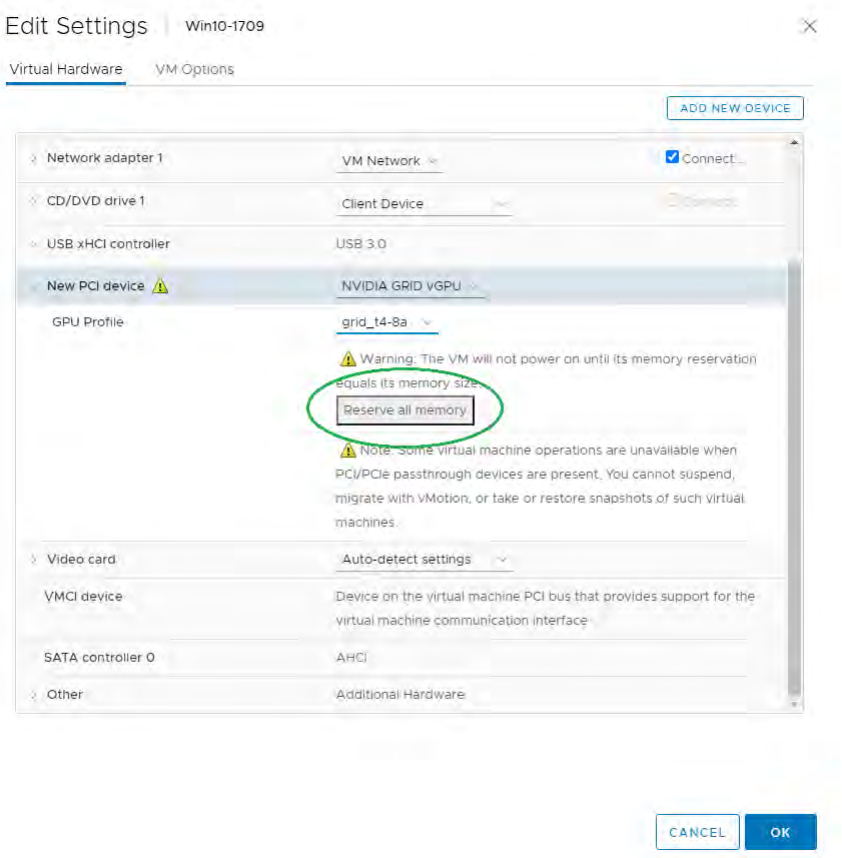


6. The new PCI device will show the NVIDIA vGPU device has been added.



7. Use the GPU Profile selection bar to configure the GPU. **Click** Reserve all memory!

Note if your VM does not start, it is possible that all guest memory was not reserved. You can reserve all guest memory by expanding the memory section of the VM settings and selecting the check box **Reserve all guest memory (All locked).**



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

8.8 Installing NVIDIA Driver in Windows Virtual Desktop

Start the virtual machine, and then connect to it using either VMware Remote Console or through the vSphere Web Client.

Note: When connected, a popup warning requesting that you restart the computer to apply changes will display the first time it is booted after enabling the NVIDIA vGPU.

Note: See Appendix B for installing & licensing NVIDIA Driver for a Linux Virtual Desktop.

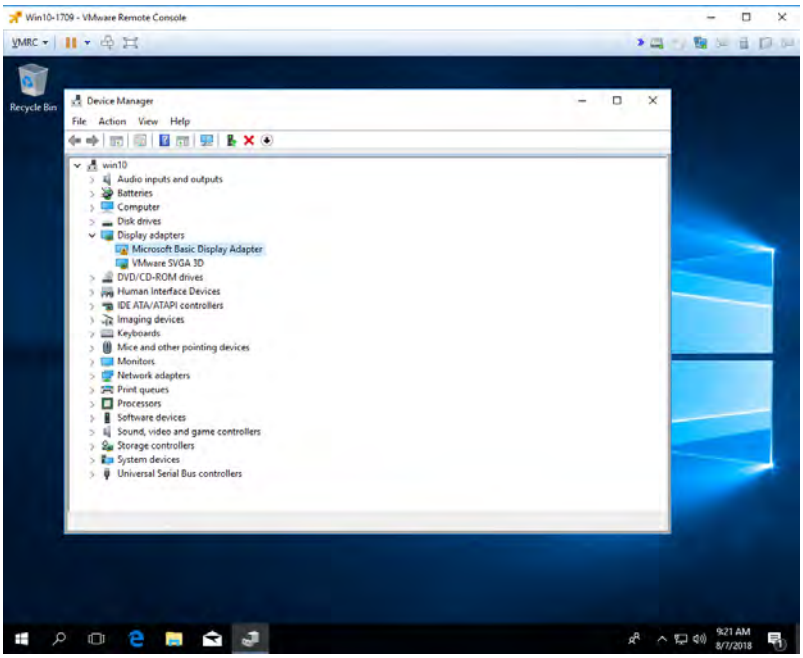
Note: Enabling Remote Desktop Connection within the Windows 10 VM before attaching a vGPU is suggested. Installing the vGPU drive over RDC is recommended, further admin via VMware Remote Console is not possible.

1. Click **Restart Later** to continue booting the virtual machine.

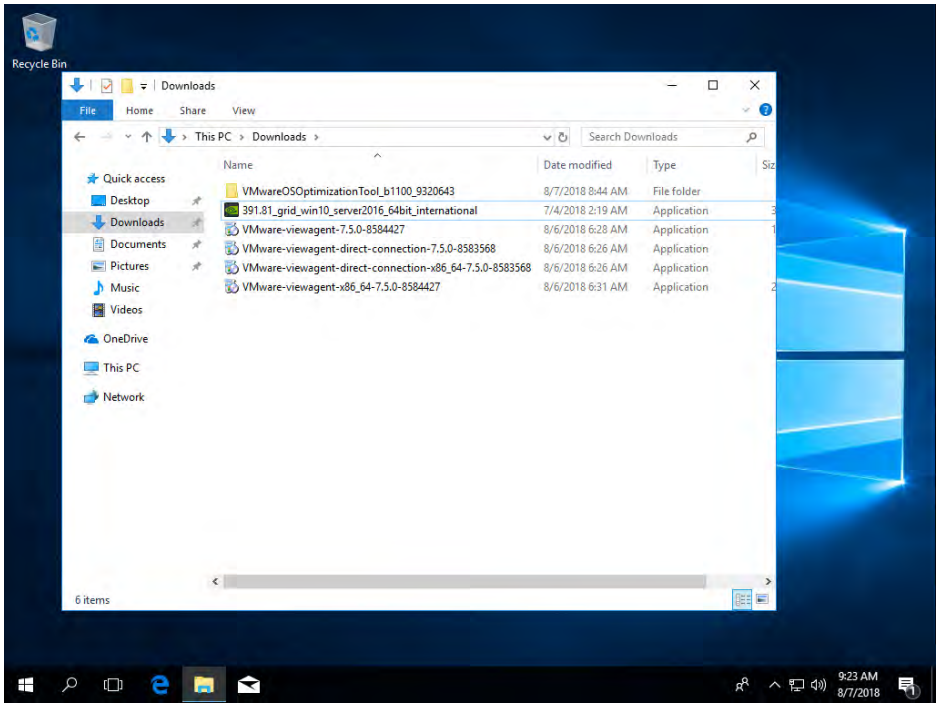
!	CAUTION: DO NOT REBOOT THE VIRTUAL MACHINE IF YOU HAVE OLDER NVIDIA DRIVERS INSTALLED. DOING THIS WILL CAUSE A BLUE SCREEN.

2. Log into Windows and open the Device Manager.

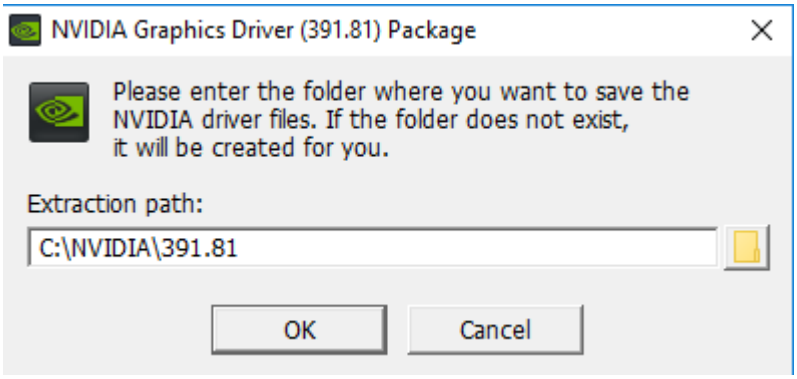
The **Standard VGA Graphics Adapter** displays in the **Display adapters** section of the Device Manager with an exclamation point by it to indicate a driver problem. This is normal.



3. Locate the NVIDIA driver and double-click the **Setup** icon to launch it. (Recommendation: Have the software on a share volume that can be mounted by the VM for quick access.).

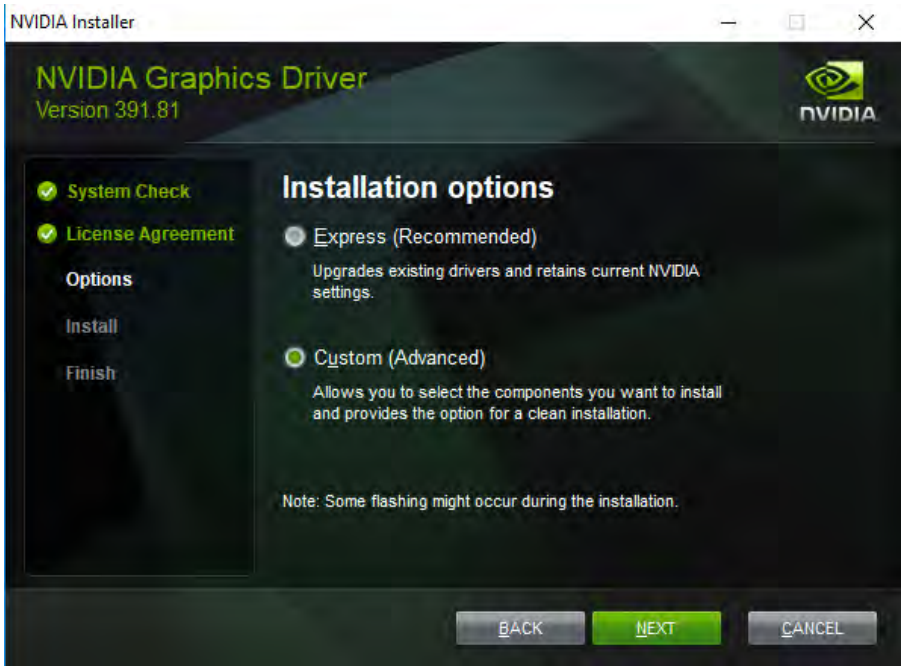


4. Click OK to continue install.



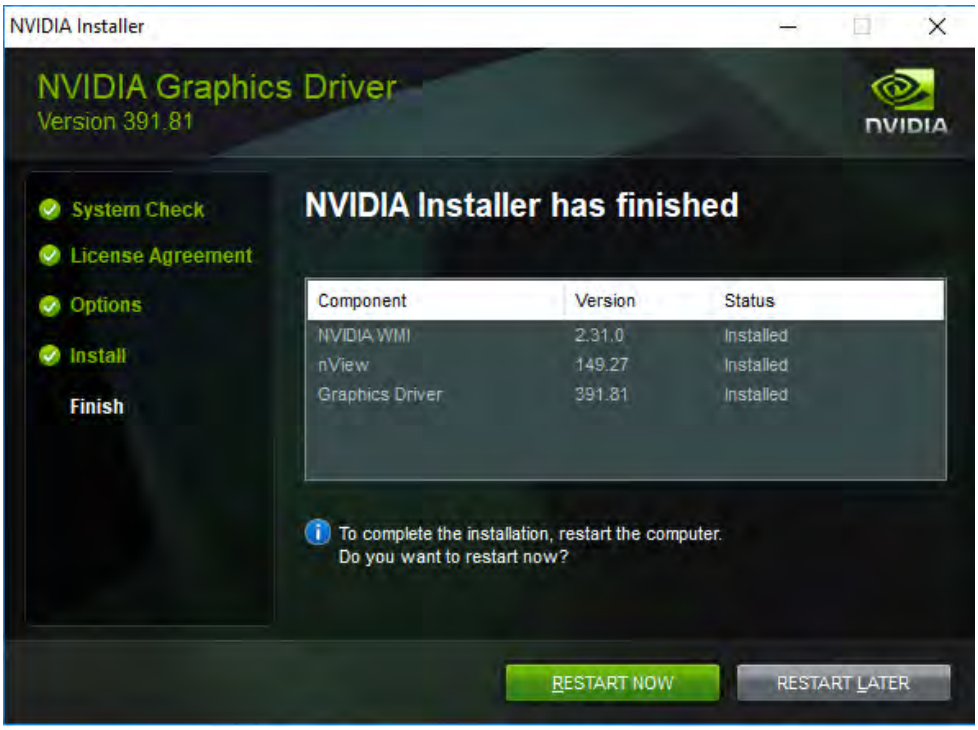
5. The NVIDIA software license agreement window displays. Click the **AGREE AND CONTINUE** button to proceed.

6. The ***Installation Options*** window displays. Check the **Custom (Advanced)** radio button, then click **Next**. The ***Custom installation*** options window appears.



7. Check the **Perform a clean installation** checkbox, and then click **Next**.

8. A window displays when the NVIDIA Graphics Driver installation is complete.



9. Reboot the VM to complete the install.

Note: After restarting, the mouse cursor may not track properly using VNC or vSphere console. If so, use View Client to Direct Connect.

8.9 Licensing NVIDIA vGPU (Update 11.0)

NVIDIA vGPU is a licensed product. When booted on a supported GPU, a vGPU runs at reduced capability until a license is acquired. The performance of an unlicensed vGPU is restricted as follows:

Elapsed Time	Performance Degradation
20 minutes	• Frame rate is capped at 15 frames per second. • The performance of applications and processes that use CUDA is degraded.
24 hours	• Frame rate is capped at 3 frames per second. • CUDA stops working and CUDA API function calls fail. • GPU resource allocations for a vGPU are limited, which will prevent some applications from running correctly.

These restrictions are removed when a license is acquired. After you license NVIDIA vGPU, the VM that is set up to use NVIDIA vGPU is capable of running all DirectX (up to and including DirectX12 and DX12-Raytracing on Turing architecture cards),OpenGL & Vulkan graphics applications.

If licensing is configured, the virtual machine (VM) obtains a license from the license server when a vGPU is booted on these GPUs. The VM retains the license until it is shut down. It then releases the license back to the license server. Licensing settings persist across reboots and need only be modified if the license server address changes, or the VM is switched to running GPU pass through.

Note: For complete information about configuring and using NVIDIA vGPU software licensed features, including vGPU, refer to Virtual GPU Client Licensing User Guide.

8.9.1.1 Licensing NVIDIA vGPU on Windows

1. Open NVIDIA Control Panel:

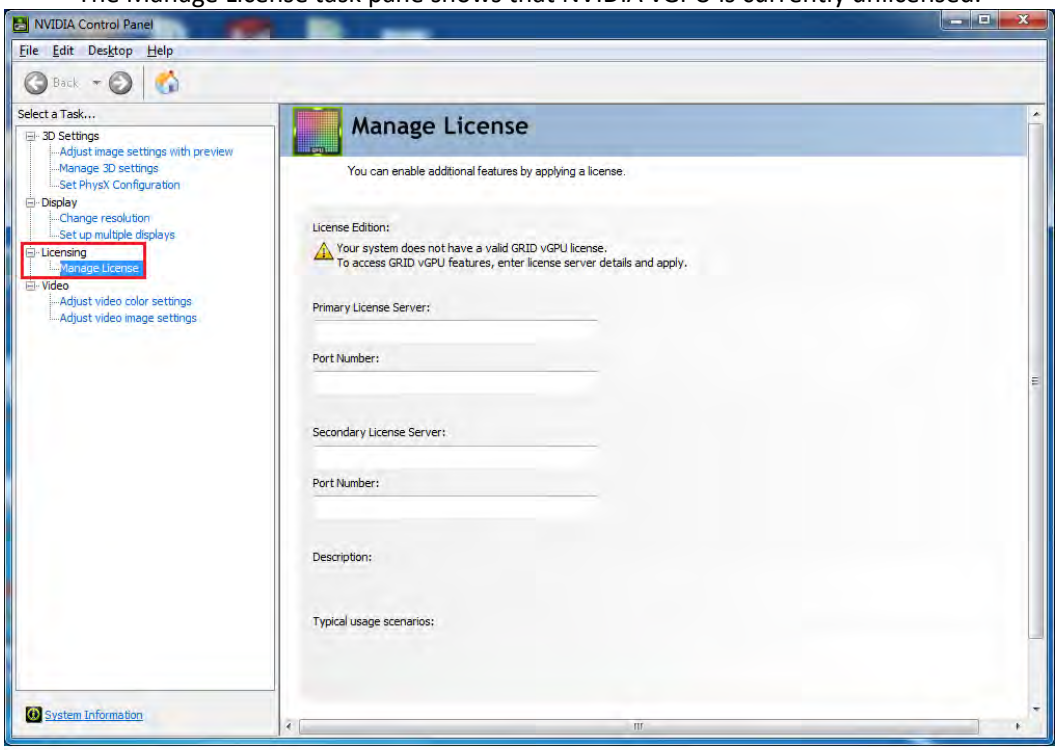
   a) Right-click on the Windows desktop and select **NVIDIA Control Panel** from the menu.

   b) Open Windows Control Panel and double-click the **NVIDIA Control Panel** icon.

2. In NVIDIA Control Panel, select the **Manage License** task in the **Licensing** section of the navigation pane.

Note: If the Licensing section and Manage License task are not displayed in NVIDIA Control Panel, the system has been configured to hide licensing controls in NVIDIA Control Panel. For information about registry settings, see Virtual GPU Client Licensing User Guide.

The Manage License task pane shows that NVIDIA vGPU is currently unlicensed.



3. In the **Primary License Server** field, enter the address of your primary NVIDIA vGPU software License Server. The address can be a fully qualified domain name such as gridlicense1.example.com, or an IP address such as 10.31.20.45. If you have only one license server configured, enter its address in this field.
4. Leave the **Port Number** field under the **Primary License Server** field unset. The port defaults to 7070, which is the default port number used by NVIDIA vGPU software License Server.
5. In the **Secondary License Server** field, enter the address of your secondary NVIDIA vGPU software License Server. If you have only one license server configured, leave this field unset. The address can be a fully qualified domain name such as gridlicense2.example.com, or an IP address such as 10.31.20.46.
6. Leave the **Port Number** field under the **Secondary License Server** field unset. The port defaults to 7070, which is the default port number used by NVIDIA vGPU software License Server.
7. Click **Apply** to assign the settings. The system requests the appropriate license for the current vGPU from the configured license server.

8. The vGPU within the VM should now exhibit full frame rate, resolution, and display output capabilities. The VM is now capable of running the full range of DirectX and OpenGL graphics applications.
9. If the system fails to obtain a license, see Virtual GPU Client Licensing User Guide for guidance on troubleshooting.

Chapter 9. Creating a Citrix Machine Catalog

This chapter describes the following:

- Creating a Citrix Machine Catalog using Citrix Machine Creation Services (MCS) to deploy a Virtual Desktop
- Creating a Citrix Machine Catalog using Citrix Machine Creation Services (MCS) to deploy a Virtual Application

To create a pool of virtual machines for users to remotely access, a Citrix Machine Catalog must be configured and deployed via Citrix Studio. We will use the Citrix Virtual Delivery Agent configured in the previous chapters to build out a pool of VMs. This chapter outlines how to create the pool of VMs by building a Citrix Machine Catalog with Citrix Machine Creation Services (MCS).

Note: Other machine deployment technologies are available like Citrix Provisioning (PVS) and is outside the scope of this document.

Please refer to Citrix product documentation for additional information regarding machine catalog and deliver group creation.

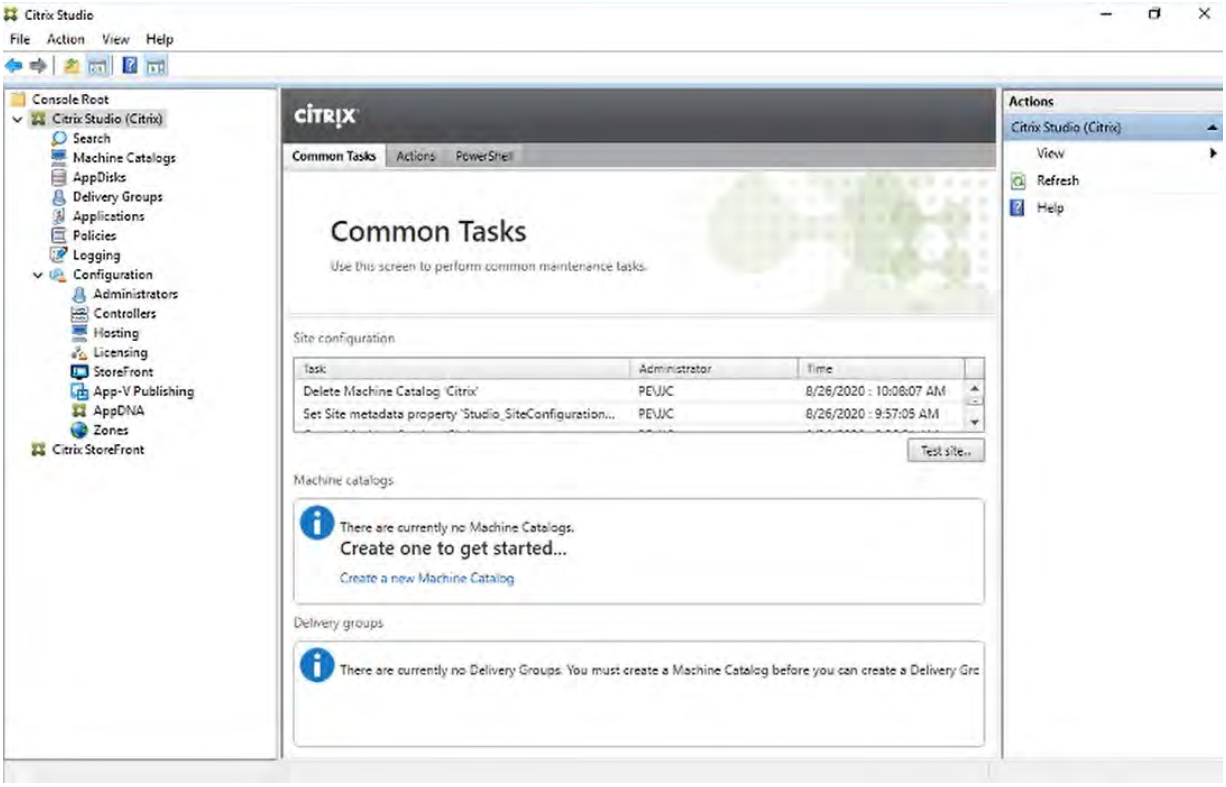
Part of creating a Citrix Machine Catalog involves choosing the correct type of operating system for your deployment. Citrix groups operating systems into three categories:

- Single-Session OS
  - Single-Session OS deployments allow for publishing only the Desktop or only the Application, but not within a single Citrix Machine Catalog.
- Multi-Session OS
  - Multi-Session OS deployments allow for the publishing of multiple desktops or multiple applications, as well as both desktops and applications within a single Citrix Machine Catalog.
- Remote PC Access.
  - Remote PC Access is outside the scope of a vGPU deployment.

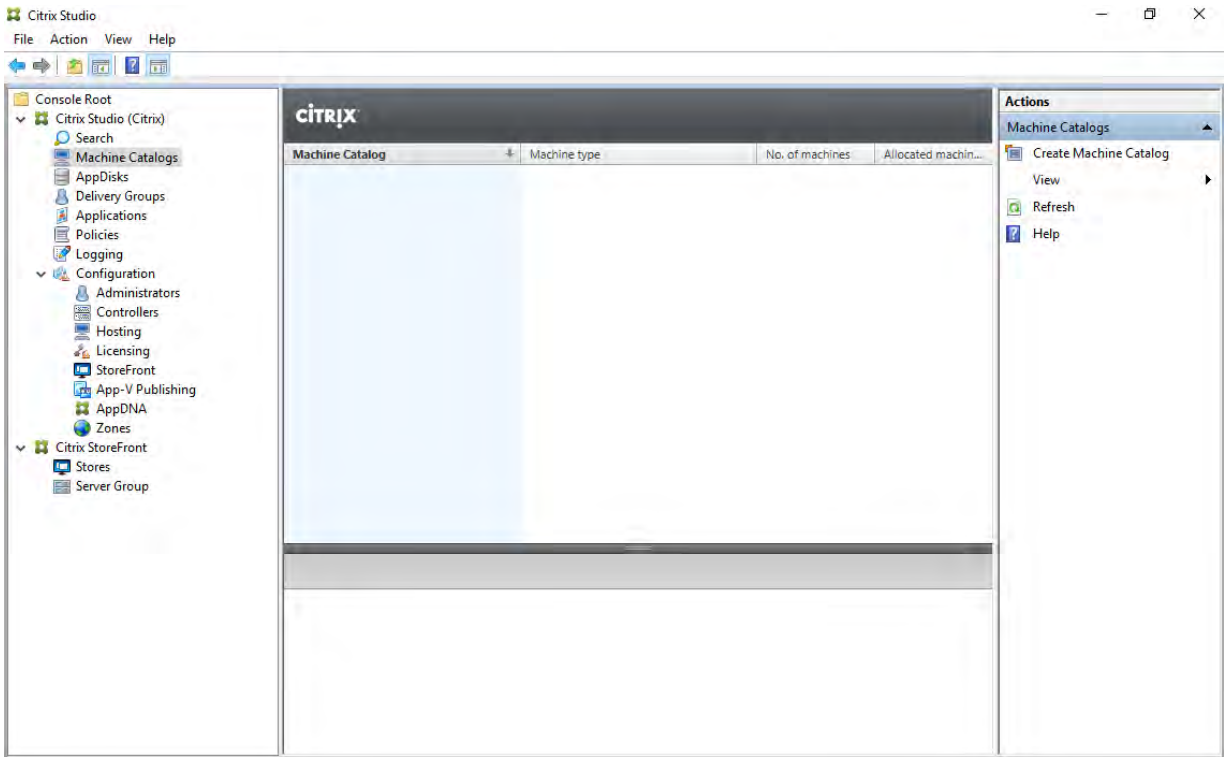
Additionally, consult Citrix and your ISV partners to determine the best deployment method for your environment.

9.1 Creating a Citrix Machine Catalog for Virtual Desktops and Apps

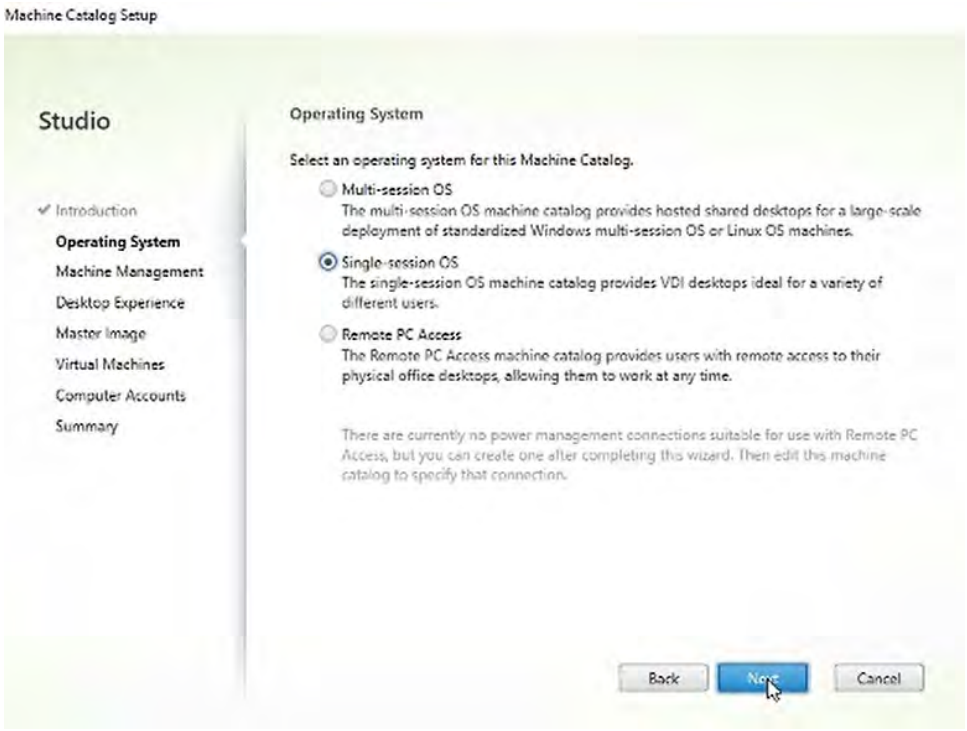
1. Log on to the Citrix Delivery Controller and Launch **Citrix Studio** from the Windows Start Menu



2. On the left menu panel, click **Machine Catalogs**.
3. Under the Actions Menu on the Right, click **Create Machine Catalog**.



4. On the Introduction page click **Next**.
5. On the Operating System select either the **Single-session OS** radio button or the **Multi-session OS** radio button, in accordance with your VDA's operating system and NVIDIA vGPU licensing level.

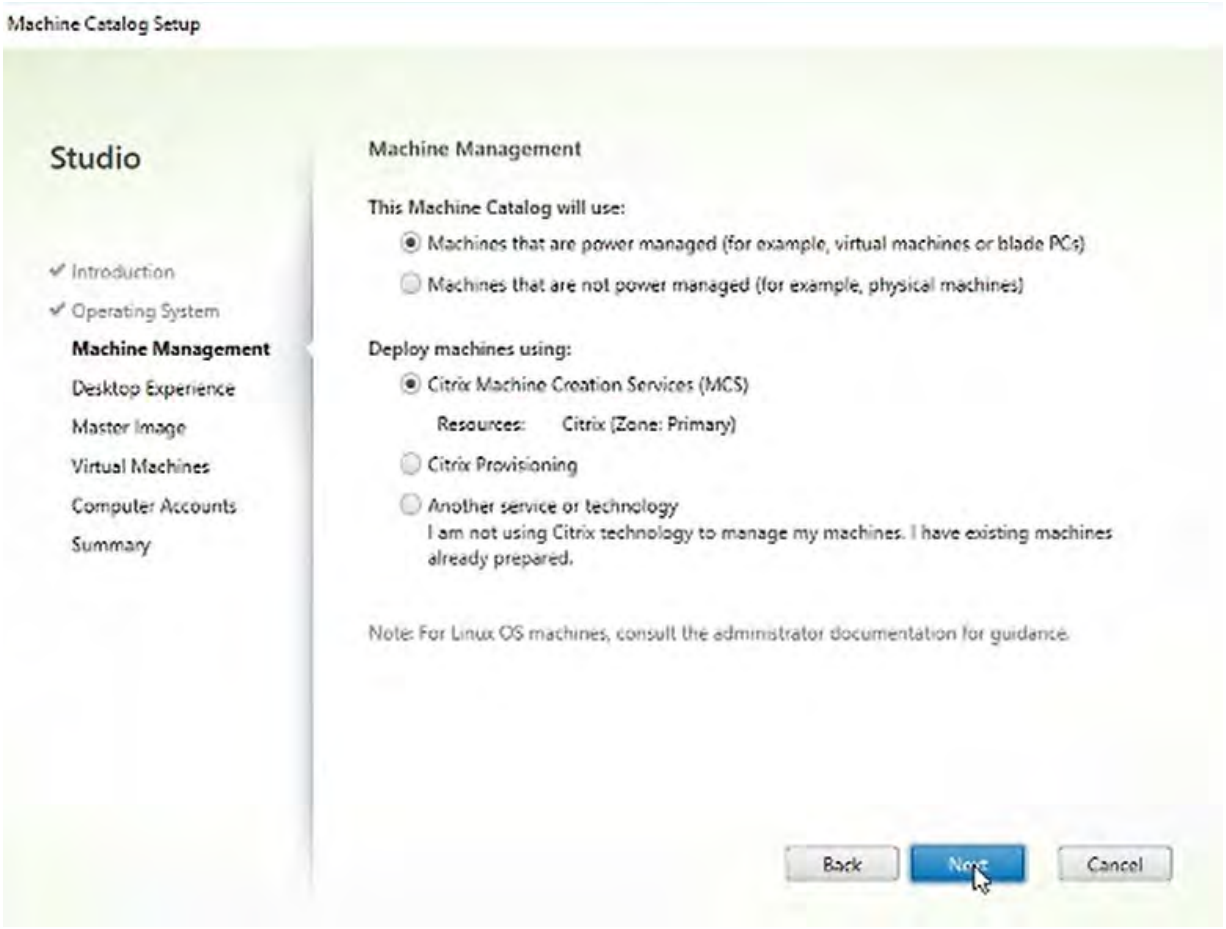


Citrix groups operating systems into three categories:

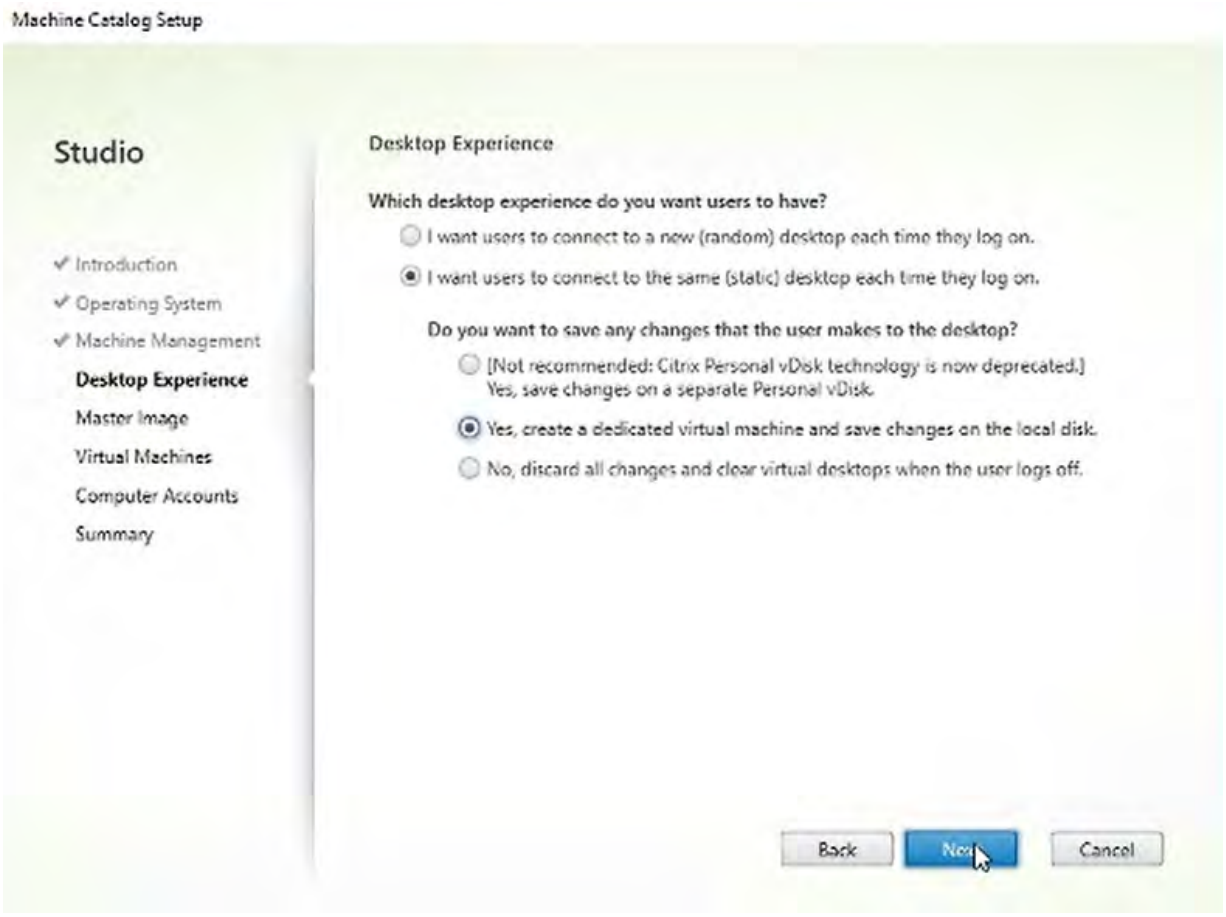
- Single-Session OS deployments allow for publishing only the Desktop or only the Application, but not within a single Citrix Machine Catalog.
- Multi-Session OS deployments allow for the publishing of multiple desktops or multiple applications, as well as both desktops and applications within a single Citrix Machine Catalog.
- Remote PC Access is outside the scope of a vGPU deployment. Additionally, consult Citrix and your ISV partners to determine the best deployment method for your environment.

Please refer to Citrix product documentation for additional information regarding machine catalog creation.

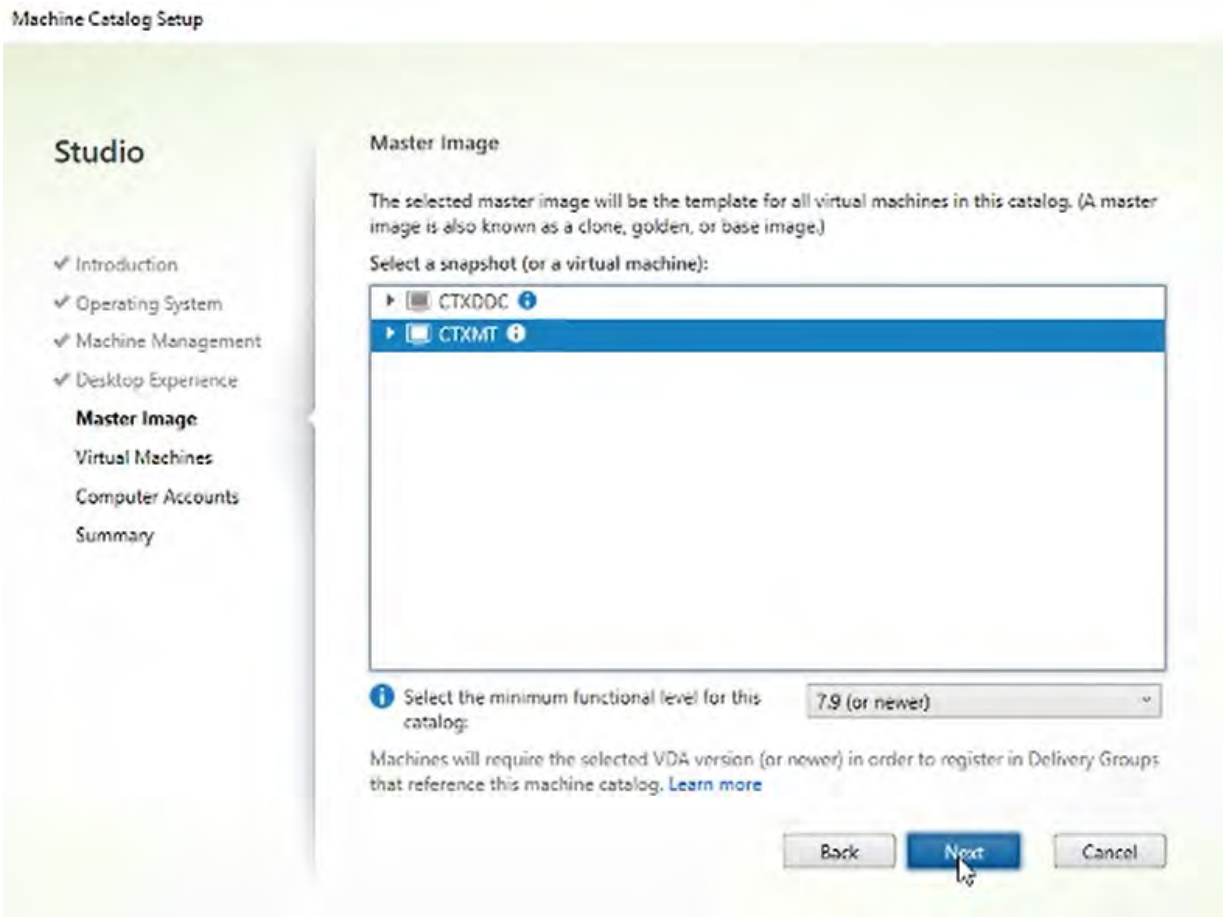
6. On the Machine Management Page, ensure the **Machines that are power managed (for example, virtual machines or blade PCs)** radio button is selected, as well as the **Citrix Machine Creation Service (MCS)** radio button and click **Next**.



7. On the Desktop Experience window, choose which type of desktop experience you would like. For purposes of POC/trail, we chose **I want users to connect to the same (static) desktop each time they log on.** Also choose **Yes, create a dedicated virtual machine and save changes on the local disk**.



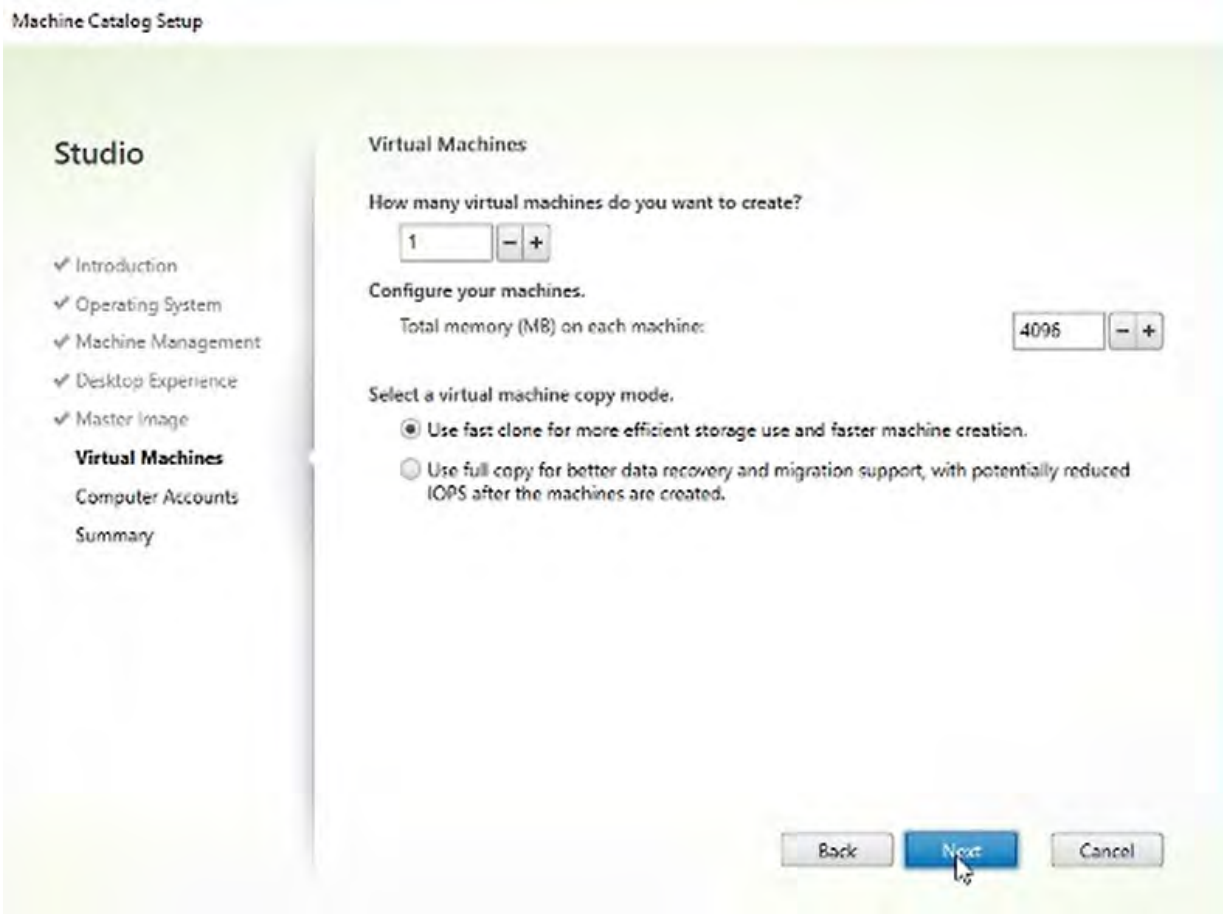
8. On the Master Image windows, select your previously created master image VM with the Citrix VDA installed.



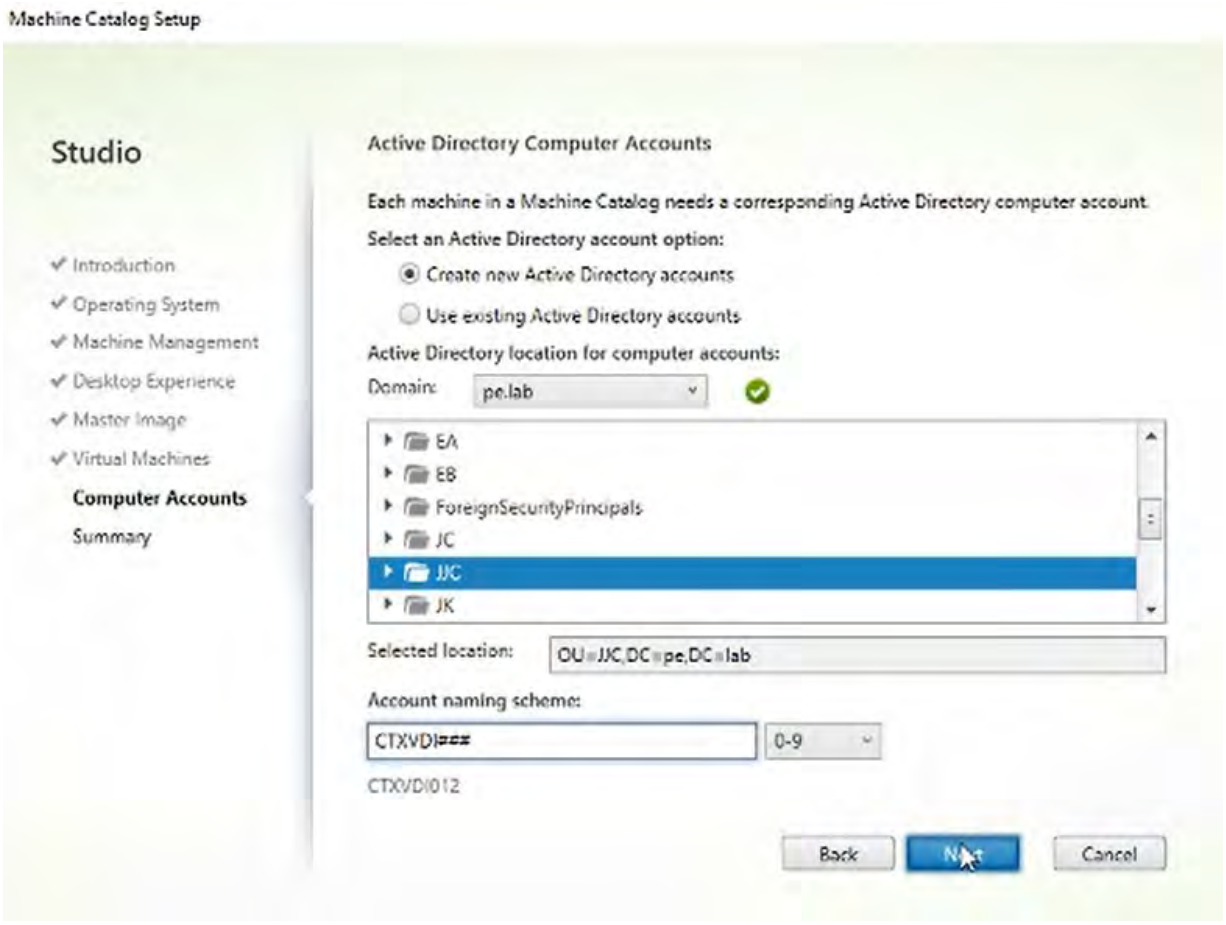
9. In the **Select the minimum functional level for this catalog:** drop-down menu ensures the latest functional level available is selected and click, **Next**.

Selecting a minimum function level lower than is available will result in a loss of new feature sets that may benefit or be required for you deployment needs.

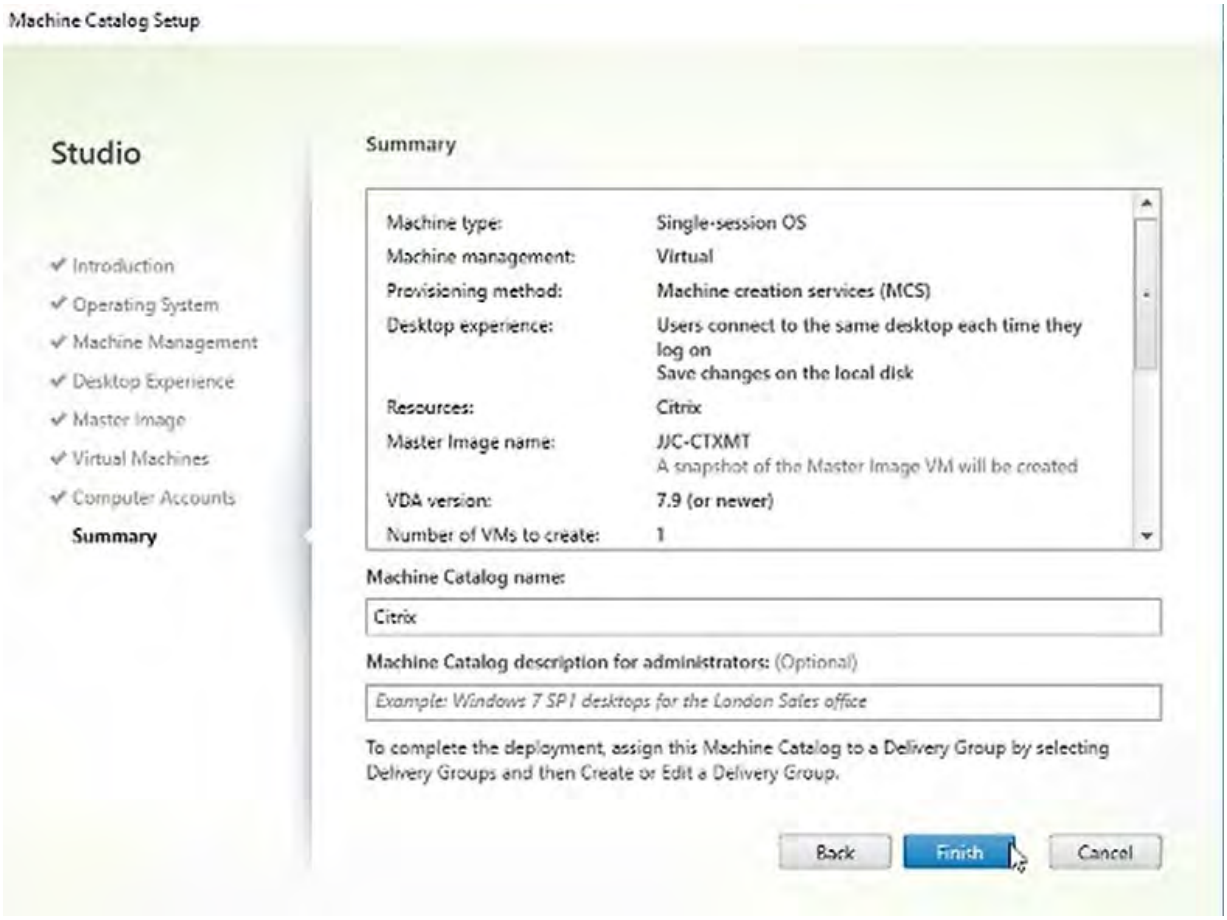
10. On the Virtual Machines windows, select how many virtual machines you want to create and the total memory on each machine. For purposes of POC/trial, we choose 1 virtual machine and leave the total memory at 4096 MB.
11. Leave the **Use fast clone for more efficient storage use and faster machine creation.** radio button selected and click **Next**.



12. On the Computer Accounts window, ensure the **Create new Active Directory accounts** radio button is select, and your domain is selected in **Domain:** drop down menu.
13. Highlight an OU to place your computer accounts and type in a naming scheme in the **Account naming scheme:** text field.
14. Click **Next**.



15. On the Summary window, type a name for your Machine Catalog in the **Machine Catalog name:** text field.



16. Click **Finish**.
17. Allow MCS to finish creating your machines. Once completed, you can see the new machines in vSphere and Active Directory.

Chapter 10. Creating a Citrix Delivery Group

This chapter describes the following:

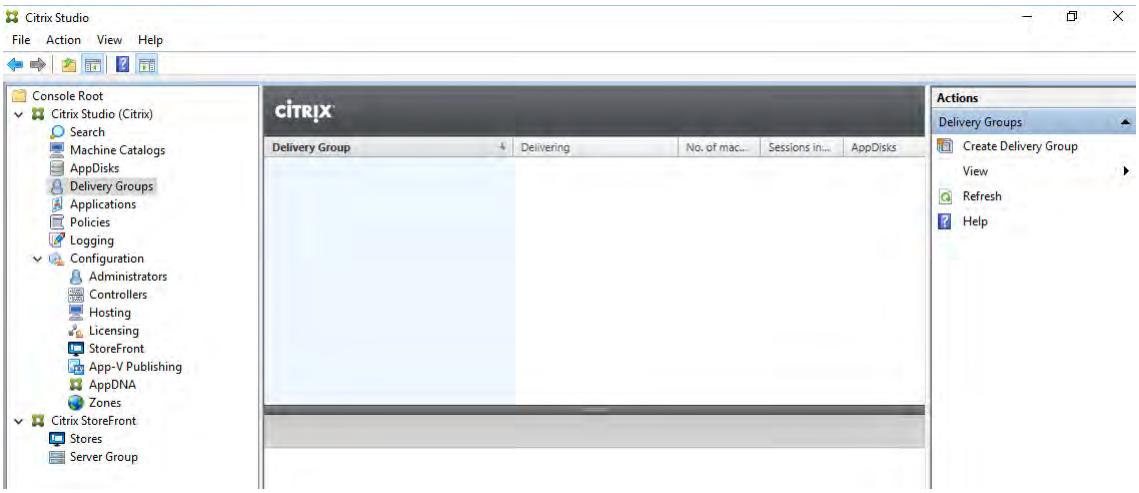
- Creating a Citrix Delivery Group to Deploy a Virtual Desktop
- Creating a Citrix Delivery Group to Deploy a Virtual Application

For users to remotely access virtual applications and desktops, a Citrix Delivery Group must be configured and deployed via Citrix Studio. We will use the Citrix Virtual Delivery Agent and the Citrix Machine Catalog created and configured in the previous chapters to build out the Delivery group.

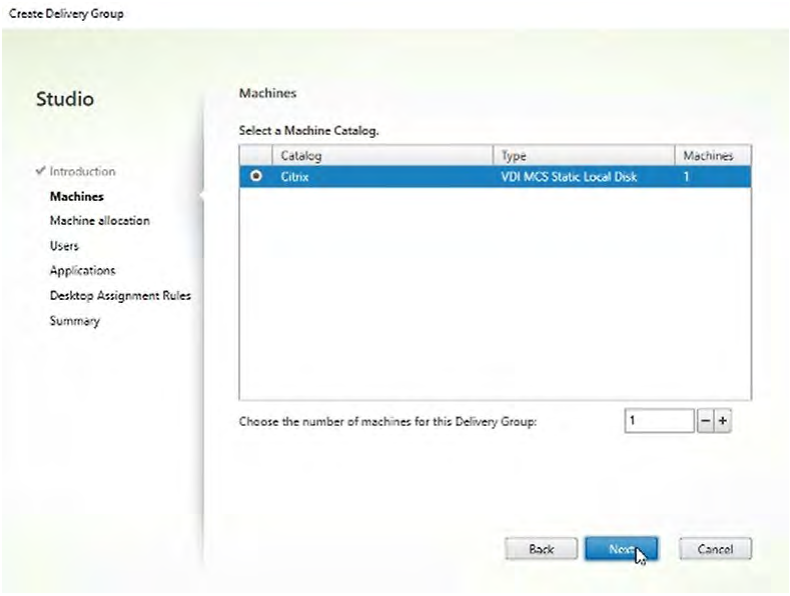
10.1 Creating a Citrix Delivery Group for Virtual Desktops

Now that you have a Citrix Machine Catalog created, the next step is to create a Delivery Group so that users can access the resource, in this case a Virtual Desktop

1. Log on to the Citrix Delivery Controller and Launch **Citrix Studio** from the Windows Start Menu
2. On the left menu pane, click **Delivery Groups**.
3. On the right Actions menu, click **Create Delivery Groups.**

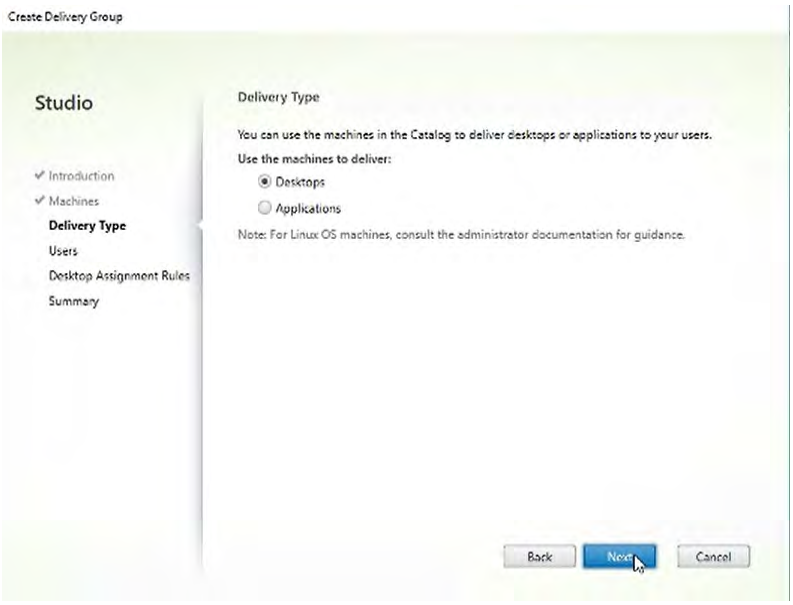


4. On the Introduction window, click **Next**.
5. On the Machines window, select the Machine Catalog that you created in the previous section and choose how many machines will be included in this delivery group. For POC/trial purposes, we choose 1 machine.



6. Click **Next.**

7. On the Delivery Type window, leave the **Desktops** radio button selected and click **Next.**

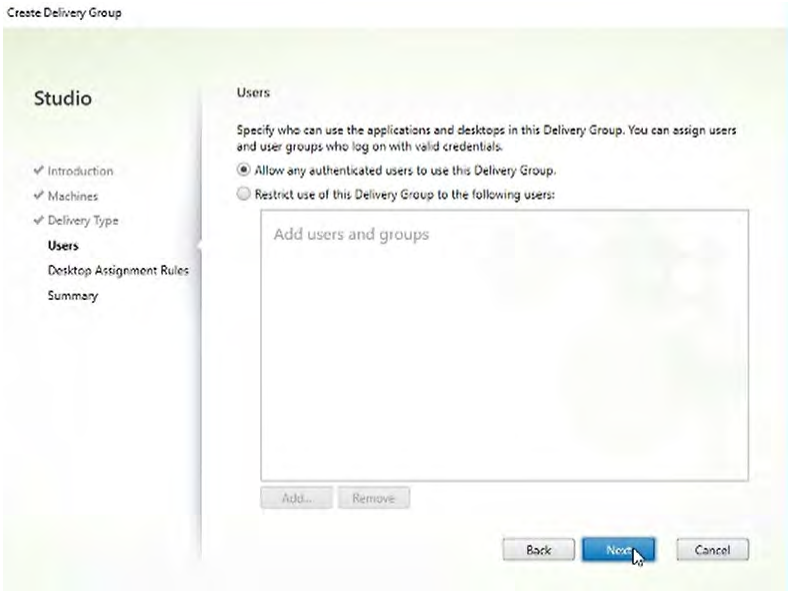


Because we previously chose a Machine Catalog type of "Single-Session OS" we must choose between deploying an application or a desktop. Had we created a Machine Catalog with a type of "Multi-Session OS," we would have had the option to deploy a desktop and an application.

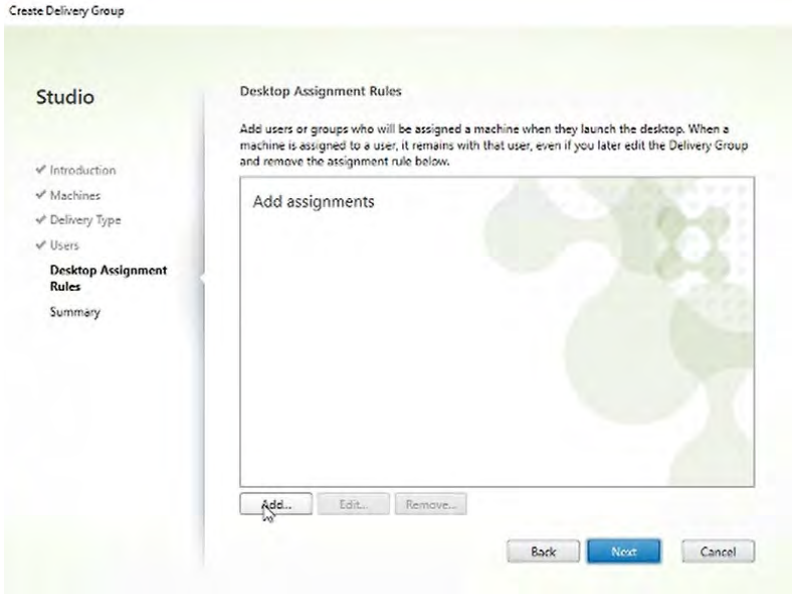
Please refer to Citrix product documentation for additional information regarding machine catalog and deliver group creation.

Note: For NVIDIA vApps you can use Citrix Virtual Apps with a multi-session OS, and for Nvidia vPC and RTX vWS you can use Citrix Virtual Desktop and is limited to single-session OS.

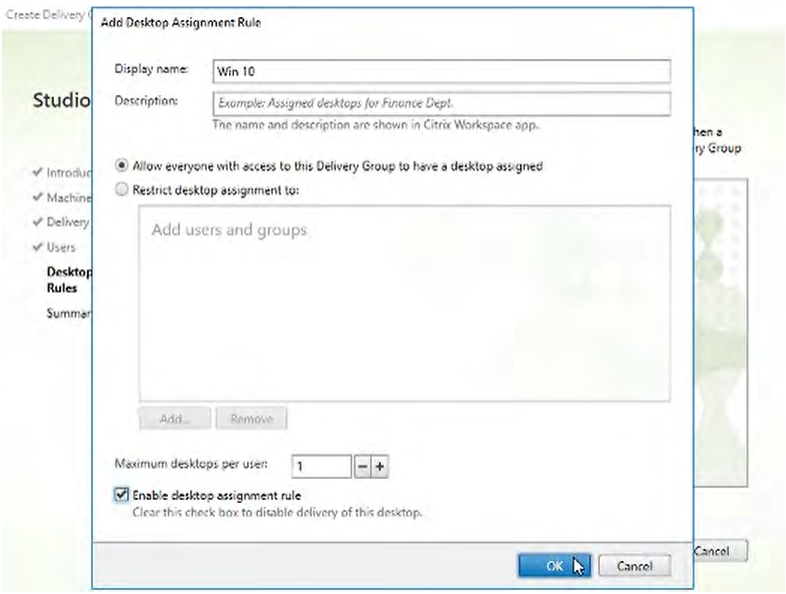
8. On the Users window, specify which users can access this delivery group. For POC/trial purposes we leave the **Allow any authenticated users to use this Delivery Group**. radio button selected.



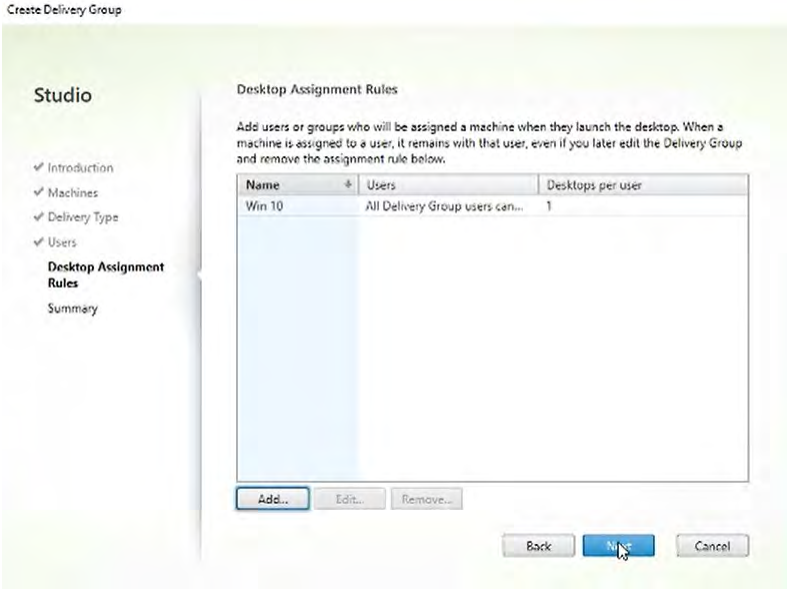
9. Click **Next**
10. On the Desktop Assignment Rules window, click **Add…**



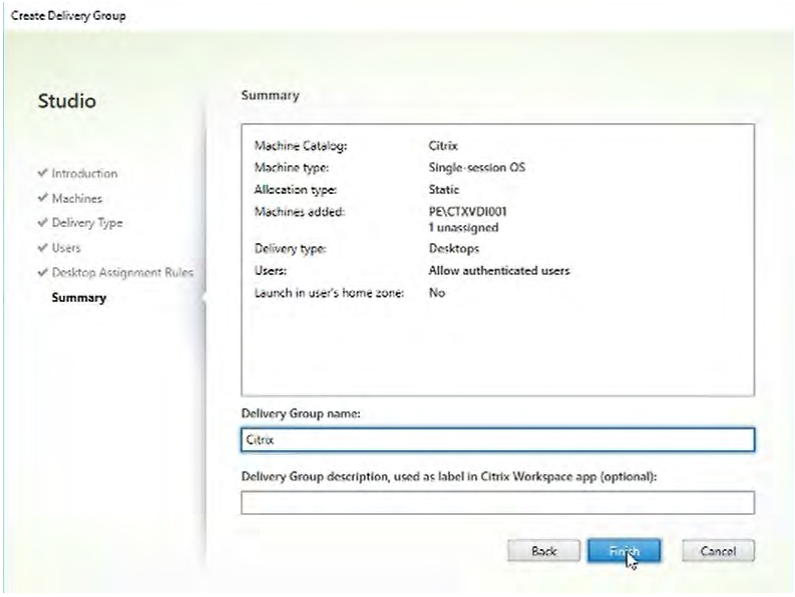
11. The Add Desktop Assignment Rule pops up. In the **Display name:** text field type a name for the Desktop that users will see.
12. For POC/trial purposes, leave the **Allow everyone with access to this Delivery Group to have desktop assigned** radio button selected.



13. Select the **Enable desktop assignment rule** check box and click **OK**
14. Click, **Next**.



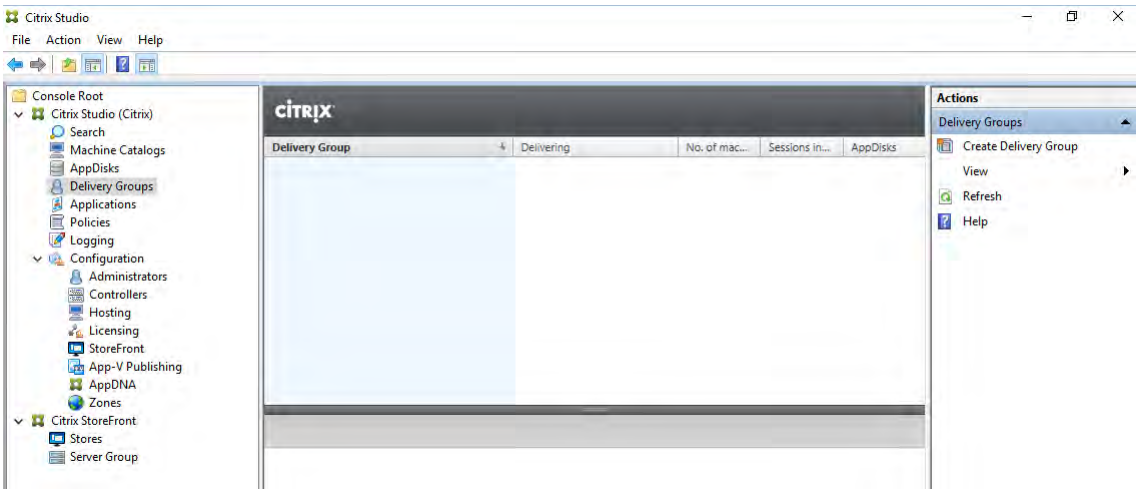
15. On the Summary window, type a name for the Delivery Group in the **Delivery Group name:** text field and click **Finish** to complete the creation of the Delivery Group.



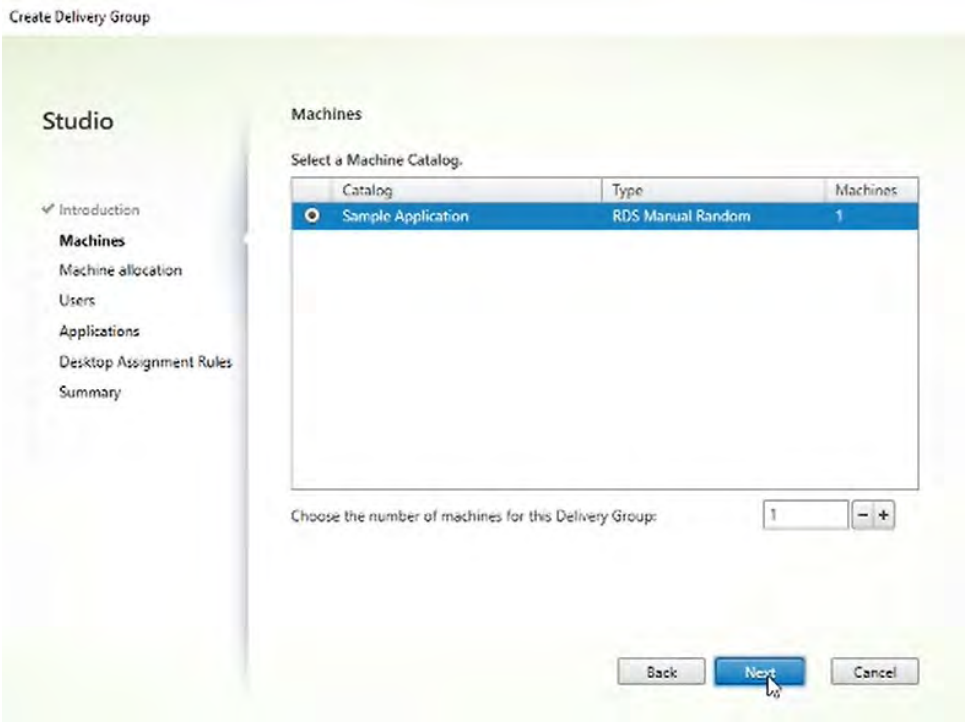
10.2 Creating a Citrix Delivery Group for Virtual Applications

Now that you have a Citrix Machine Catalog created, the next step is to create a Delivery Group so that users can access the resource, in this case a Virtual Application.

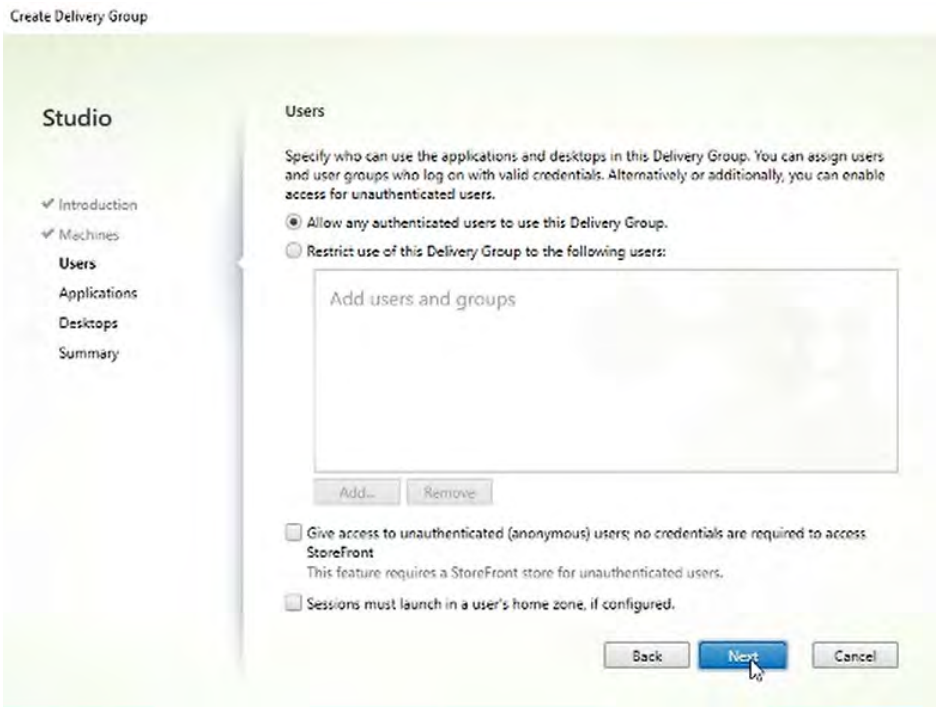
1. Log on to the Citrix Delivery Controller and Launch **Citrix Studio** from the Windows Start Menu
2. On the left menu pane, click **Delivery Groups**.
3. On the right Actions menu, click **Create Delivery Groups.**



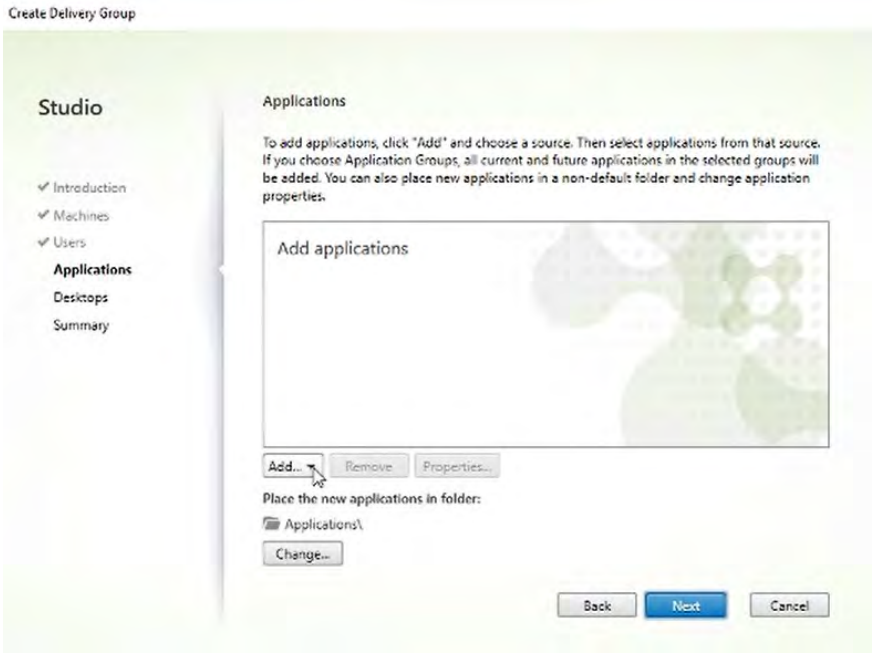
4. On the Introduction window, click **Next**.
5. On the Machines window, select the Machine Catalog that you created in the previous section and choose how many machines will be included in this delivery group. For POC/trial purposes, we choose 1 machine.



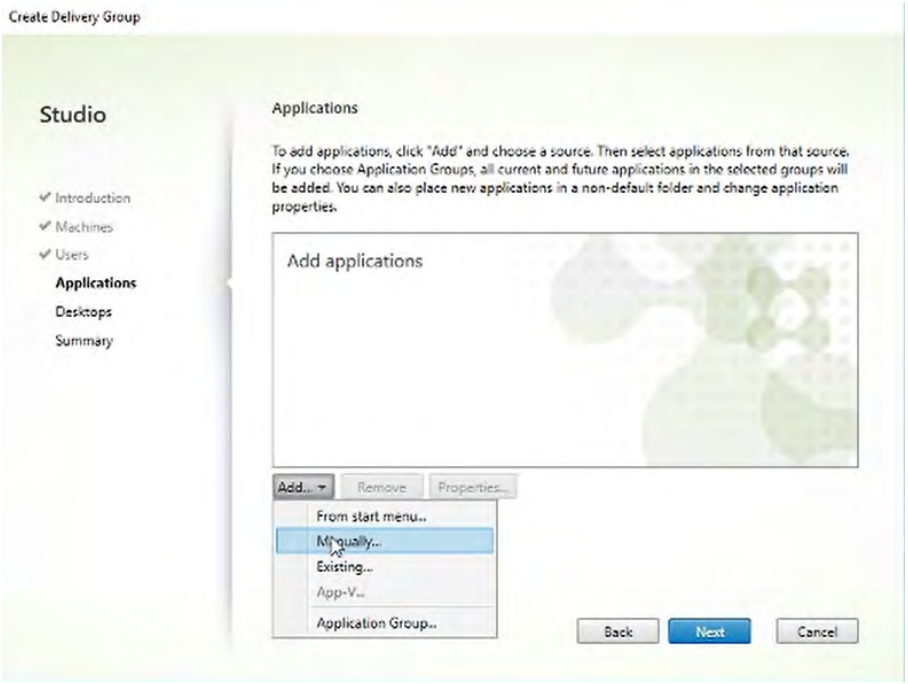
6. Click **Next.**
7. On the Users window, specify which users can access this delivery group. For POC/trial purposes we leave the **Allow any authenticated users to use this Delivery Group**. radio button selected.



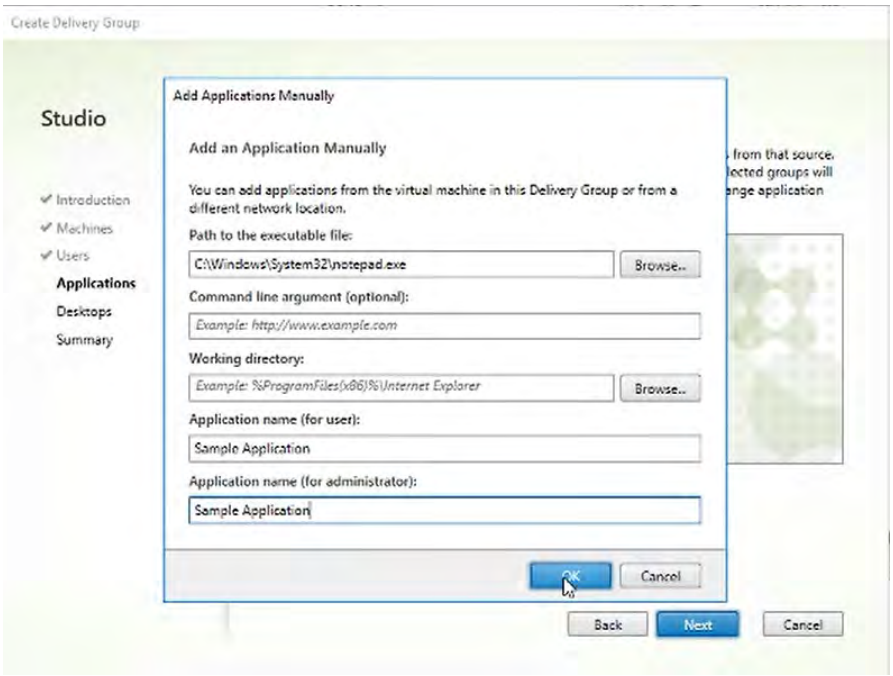
8. On the Application window, select the **Add…** dropdown menu.



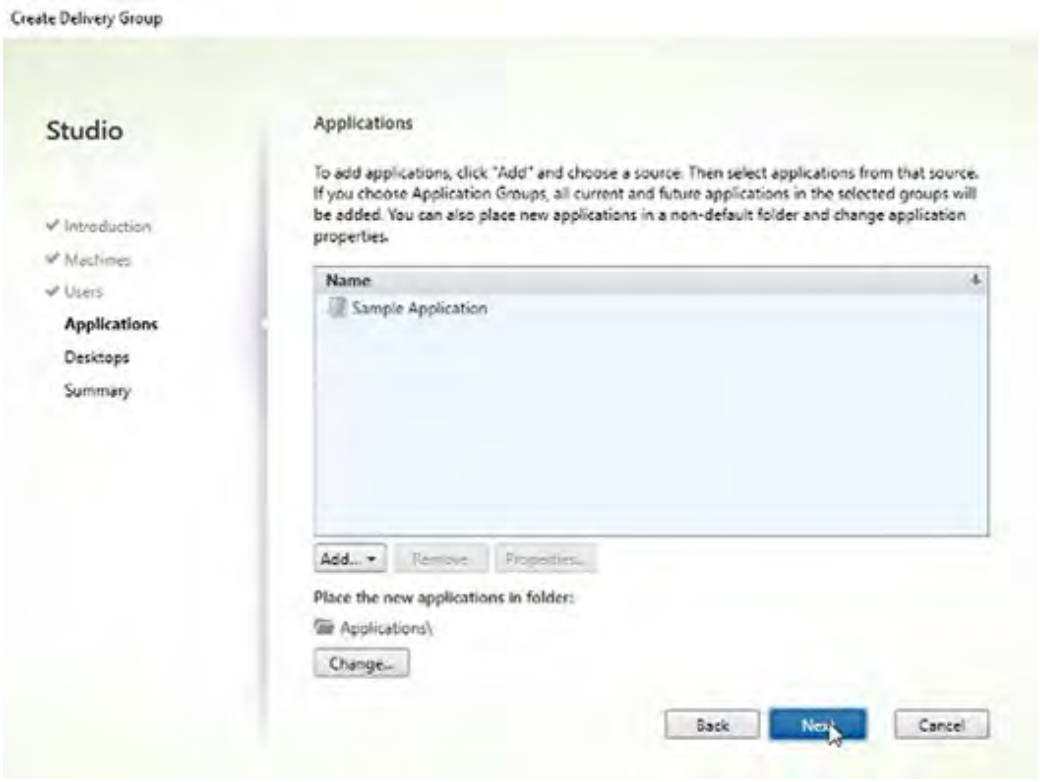
9. Select **Manually…**



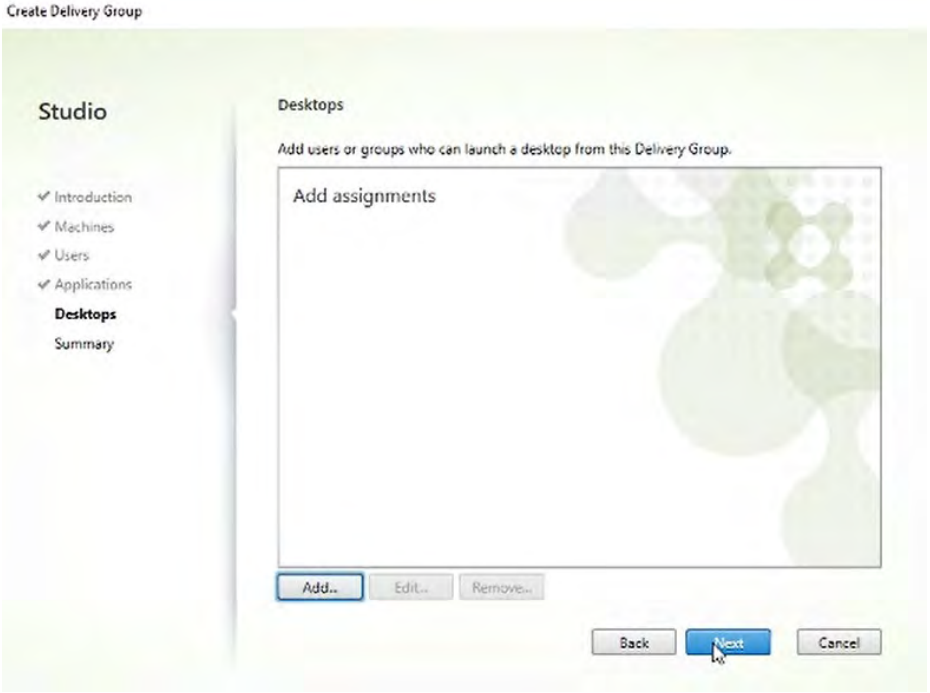
10. On the Add Applications Manually popup window, type a path to the executable, an application name for users and an application name for administrators, then Click **OK.**



11. Click **Next**



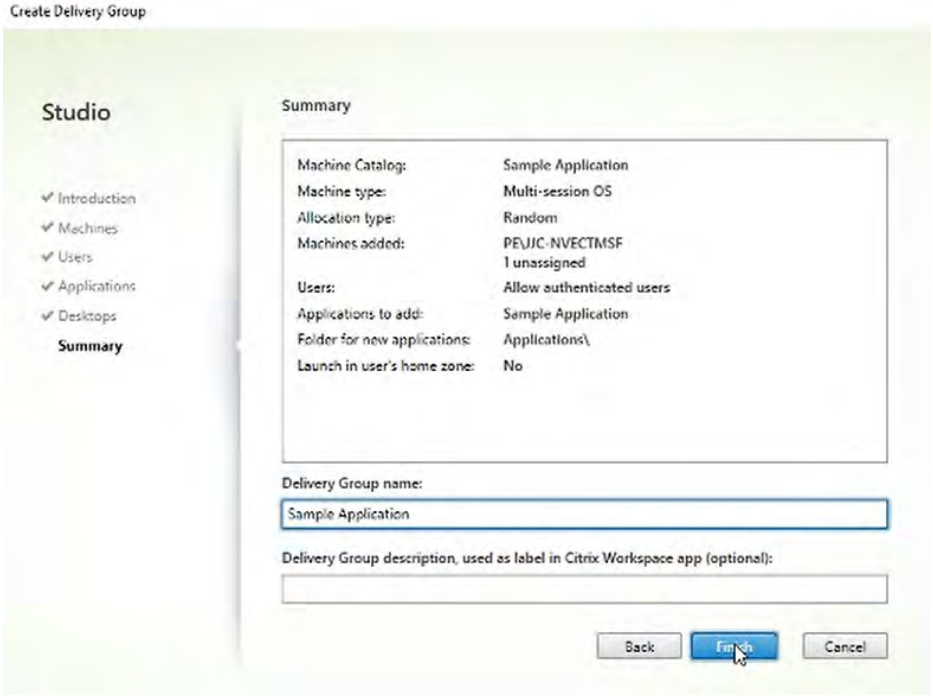
12. On the Desktops window, click **Next.**



Because we previously chose a Machine Catalog type of “Multi-Session OS” we have the option to deploy both Applications and Desktops. Had we created a Machine Catalog with type of “Single-Session OS,” we would have only been able to deploy an application or a desktop. This section focuses on Application deployment, so we choose to not deploy a desktop.

Please refer to Citrix product documentation for additional information regarding machine catalog and deliver group creation.

13. On the Summary window, type a name for the Delivery Group in the **Delivery Group name:** text field and click **Finish** to complete the creation of the Delivery Group.



Chapter 11. Creating Citrix Policies for NVIDIA vGPU

This chapter describes the following:

- Creating a Citrix Policy for NVIDIA vGPU
- Creating a Microsoft Policy

A vGPU is not automatically accessible to a Citrix session. Both Citrix and Microsoft policies must be configured for a Citrix session to utilize the vGPU.

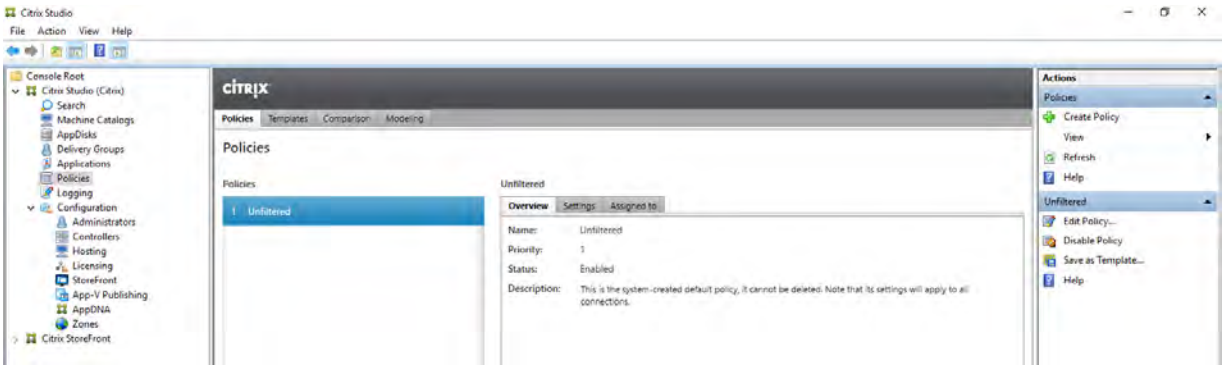
11.1 Creating a Citrix Policy for NVIDIA vGPU

The Citrix HDX 3D Pro protocol can utilize NVIDIA NVENC (Hardware-Accelerated Video Encoding). In order to utilize NVENC, the Optimize for 3D graphics workload Citrix Policy must be enabled.

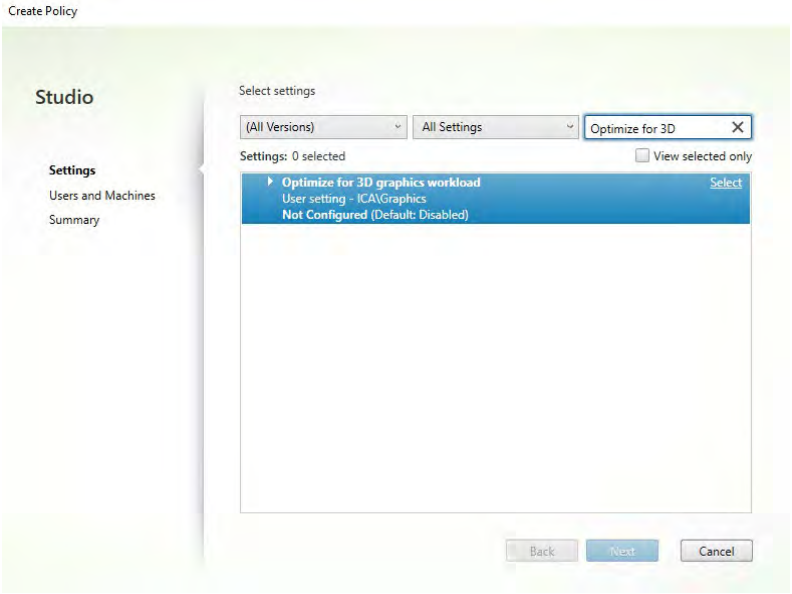
Your full Citrix policy set depends on multiple parameters like application requirements, bandwidth requirements, & image quality requirements, etc. Only enabling the Optimize for 3D graphics workloads policy is only sufficient for a POC/trail purpose. Consult Citrix and your application partners to ensure you full Citrix Policy set is optimized for you deployment needs.

Additionally, refer to the Graphics Section of the Citrix Virtual Apps and Desktop Product Documentation for additional details.

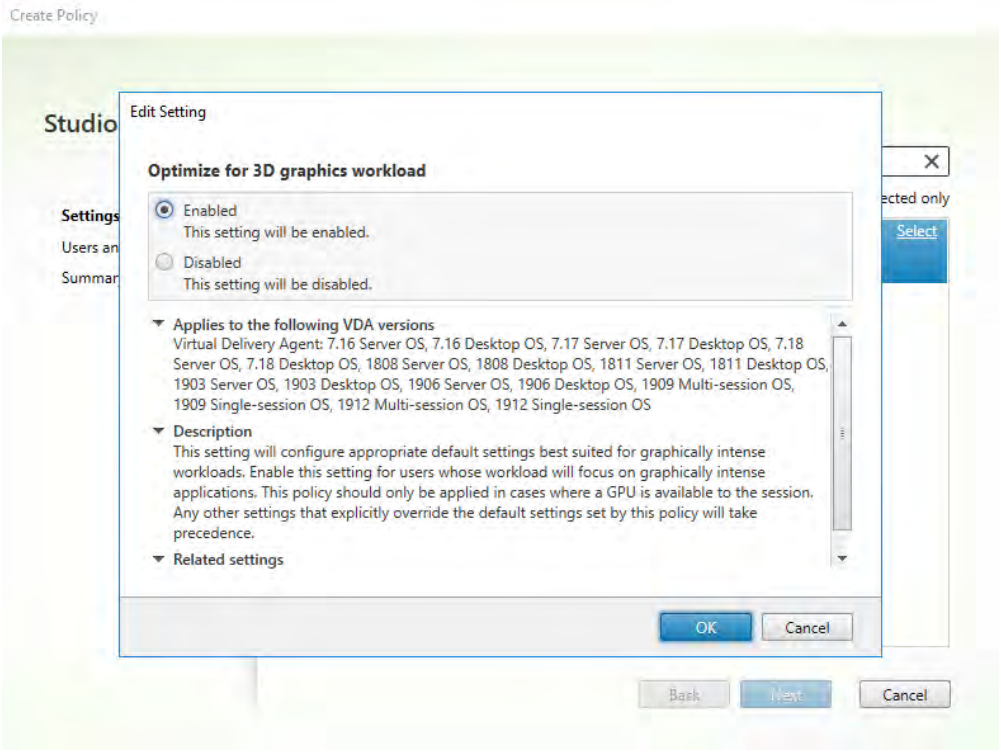
1. Log on to the **Citrix Delivery Controller**
2. Launch **Citrix Studio** from the Windows Start Menu
3. Select **Policies** on the left menu pane, then select **Create Policy** on the right-side Action menu.



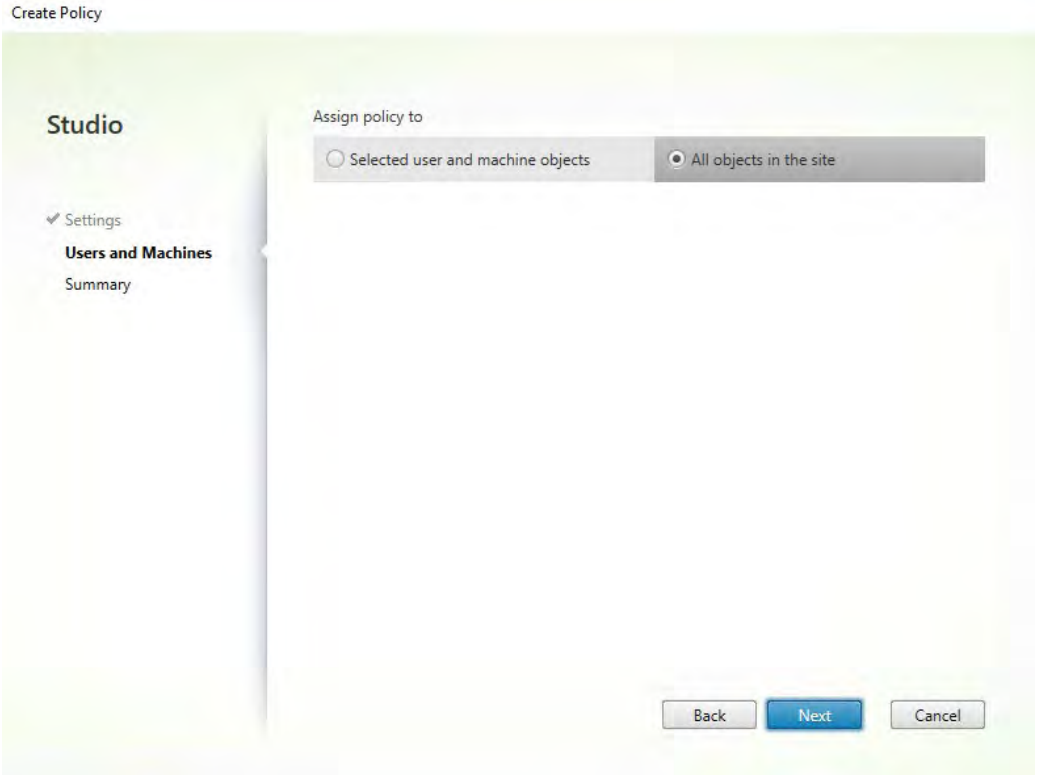
4. Search for **Optimize for 3D graphics workload** and click **Select**.



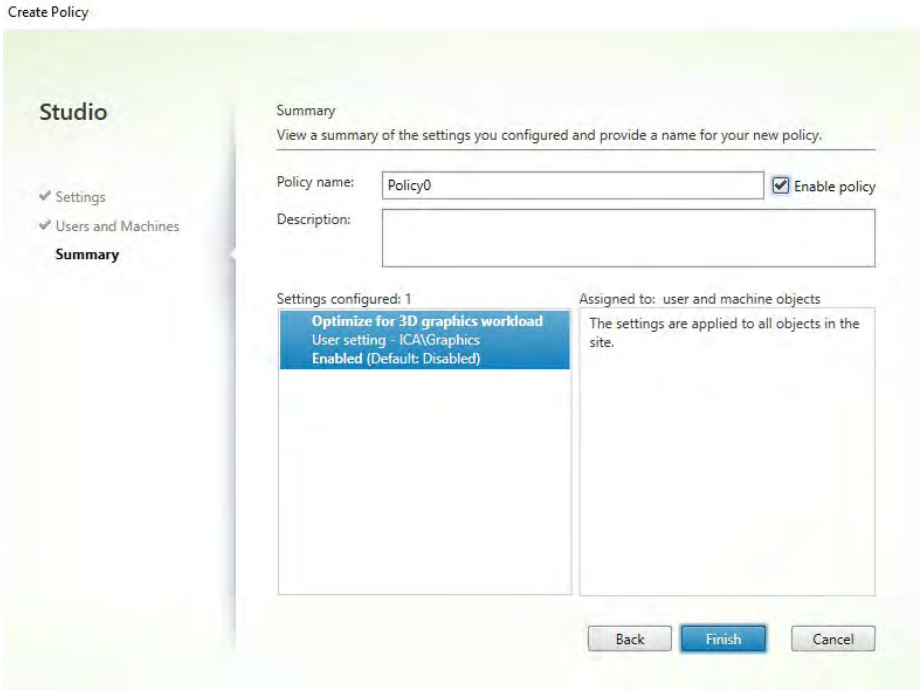
5. On the Edit Settings popup window, select the **Enabled** radio button and click **OK**.



6. Click **Next**
7. Select the appropriate machines to apply the policy to. For POC/trial purposes, select the **All objects in the site** radio button and click **Next.**



8. Check the **Enable policy** checkbox and provide a name for the policy.



9. Click **Finish**

Note: There are several policies within Citrix Studio that affect performance and the efficient use of Virtual GPUs. Policies such as **Use Video Codec for Compression, Use hardware encoding for video codec** and **Target Frame Rate**.

Please refer to the following Citrix document which refers to these settings in more detail:

https://docs.citrix.com/en-us/tech-zone/design/design-decisions/hdx-graphics.html

11.2 Creating Microsoft Group Policy for NVIDIA vGPU

On Windows Server 2016 and Windows Server 2012 R2, Remote Desktop Services (RDS) sessions on a RD Session Host server use the Microsoft Basic Render Driver as the default adapter. To use the virtual GPU in RDS sessions and Citrix HDX 3D Pro sessions enable the **Use the hardware default graphics adapter for all Remote Desktop Services sessions** setting in the group policy.

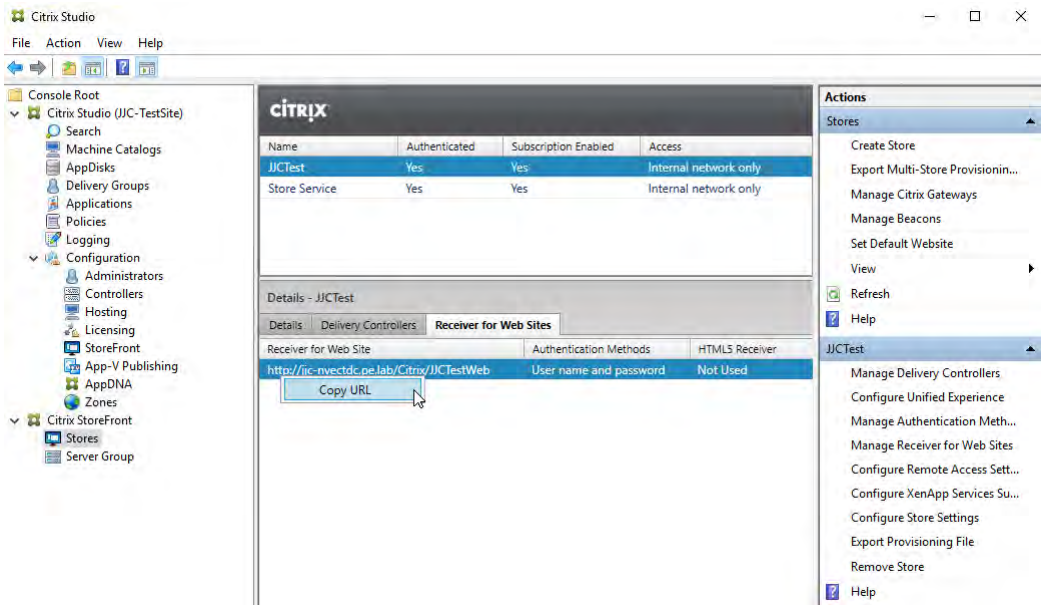
Local Computer Policy > Computer Configuration > Administrative Templates > Windows Components > Remote Desktop Services > Remote Desktop Session Host > Remote Session Environment.

Chapter 12. Citrix Workspace App

Before connecting to a virtual application or desktop over a Citrix HDX connection, the Citrix Workspace App will need to be installed and configured onto a desktop or device which the virtual desktop will be accessed from. For this guide, we will connect to the Citrix StoreFront and use the web site which can detect and download the receiver.

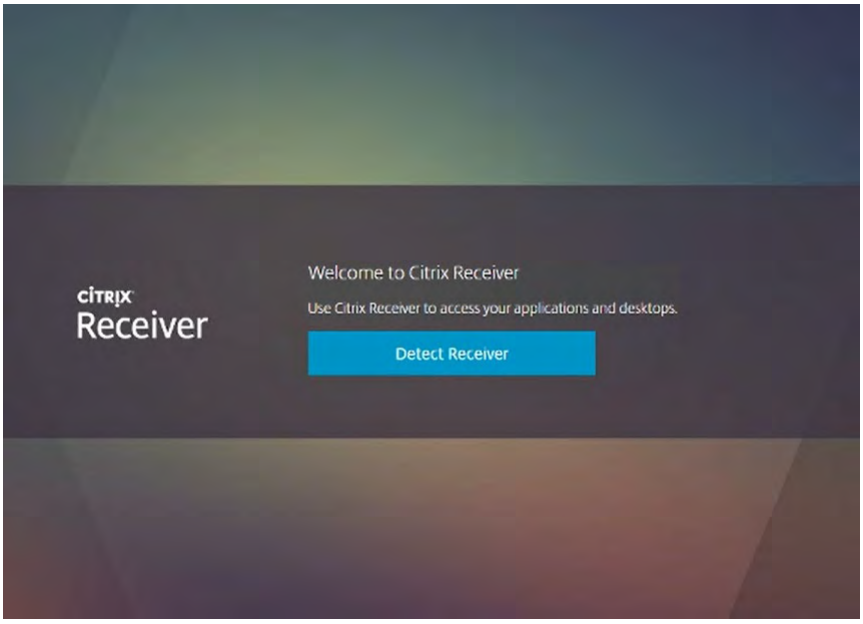
12.1 Locating Citrix StoreFront Web Site

1. Locate the Citrix Storefront URL by opening **Citrix Studio** from the Windows menu on your Citrix Delivery Controller Server.
2. Expand **Citrix StoreFront** on the left menu pane and Select **Stores**.
3. Highlight the Store for your deployment in the top menu pane and select the **Receiver for Web Sites** tab under the Details section.
4. Right Click the URL and select **Copy URL**

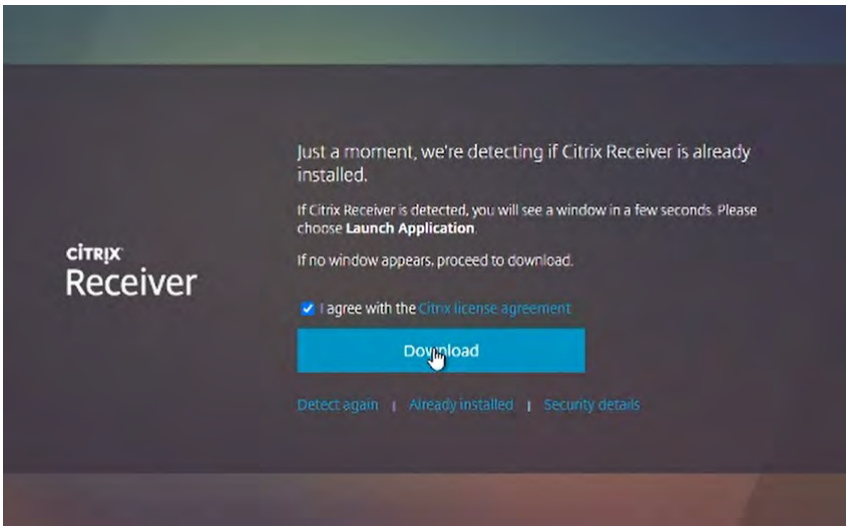


12.2 Installing Citrix Workspace App

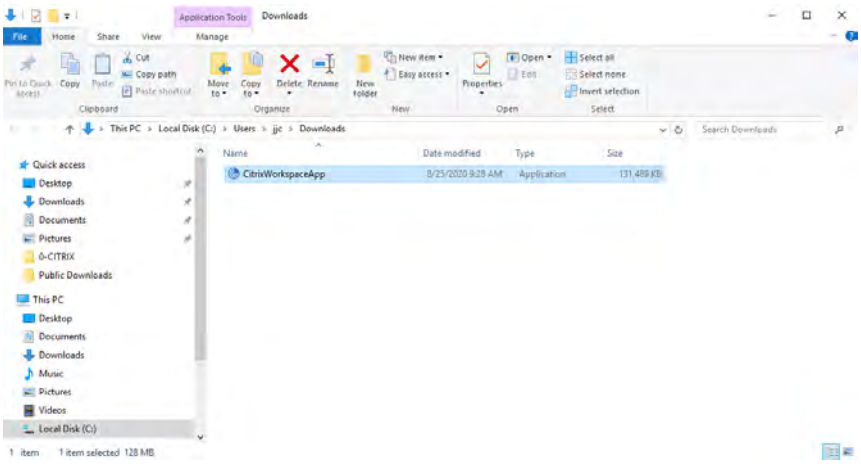
1. Log into the physical device where you will be launching the virtual desktop from and open an internet browser. Navigate to the Citrix Storefront URL you previously copied in step 11.1.4. Click on **Detect Receiver**.



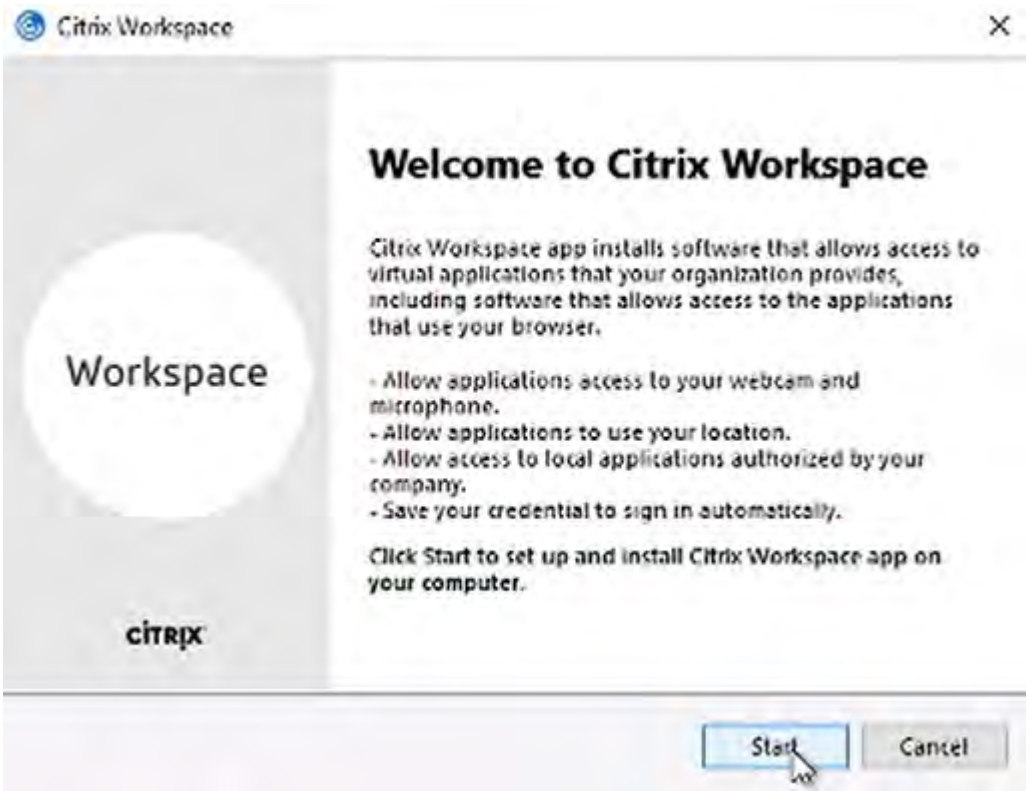
2. Review the Citrix license agreement and check the **I agree with the Citrix license agreement** checkbox and select **Download.**



3. Locate the downloaded installer program and double click to begin installation.



4. Click **Start** to begin the installation.



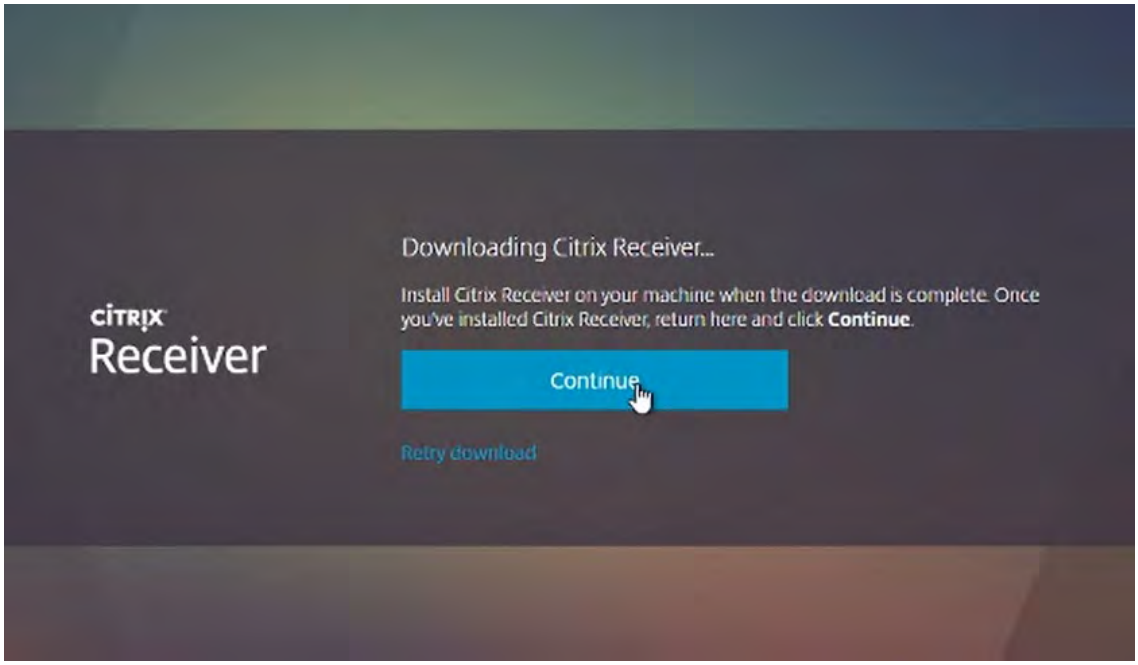
5. Select the **I accept the license agreement** check box and click **Next**.

6. Click **Install**

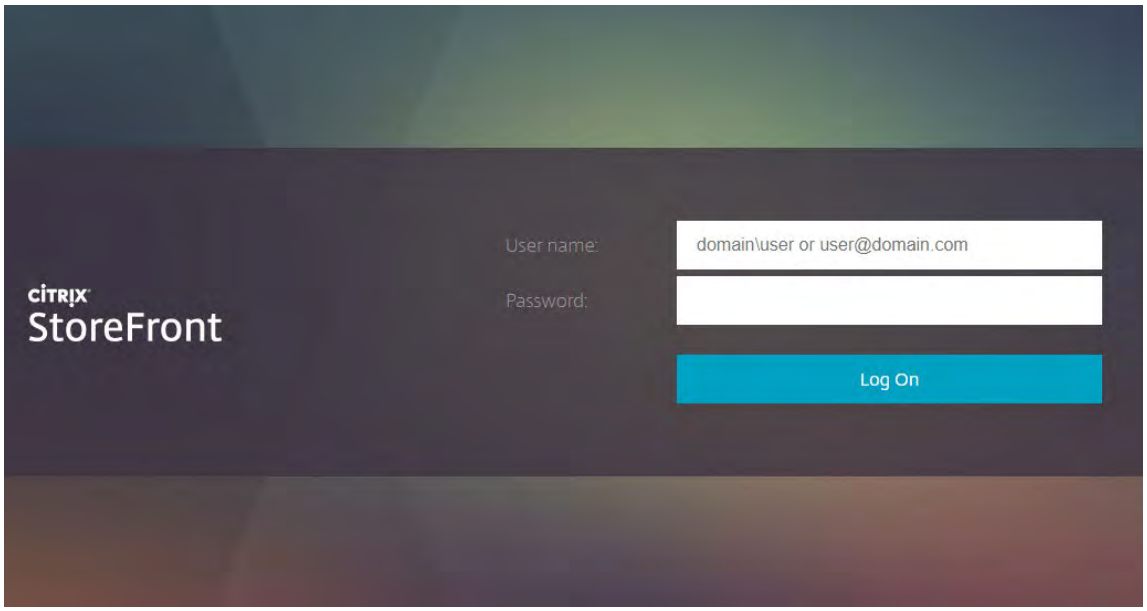
7. Click **Finish** to complete the install.

12.3 Launch a Citrix Virtual Desktop

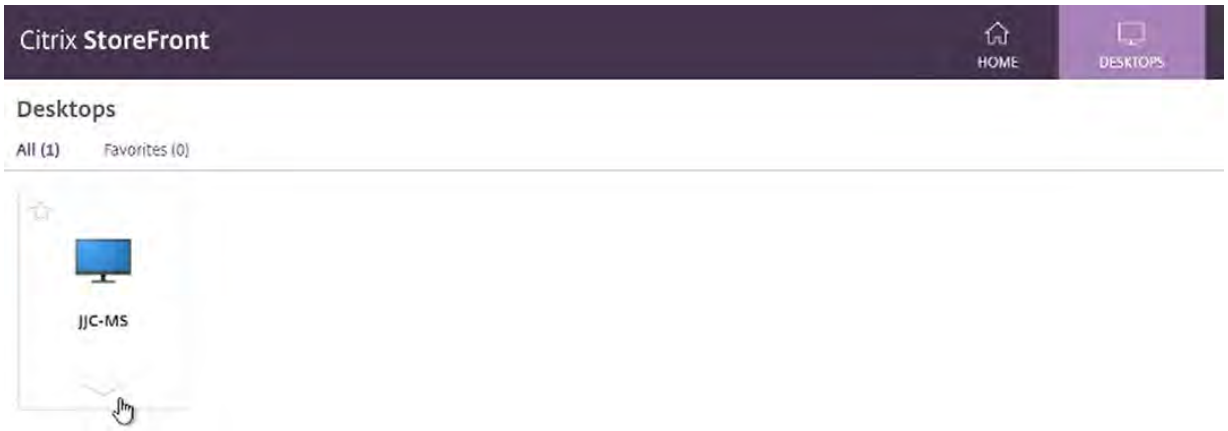
1. Navigate back to the Citrix StoreFront web browser window and click **Continue**.



2. Enter a username and password



3. Click **Desktops** in the top menu and select your desktop.



Chapter 13. Troubleshooting

This section includes links and examples of where to explore and post in order to find solutions or assistance.

Before troubleshooting or filing a bug report, review the release notes for information about known issues with the current release, and potential workarounds.

13.1 Forums

NVIDIA forums are a very inclusive source of solutions to many problems that may be faced when deploying a virtualized environment. Please search on the NVIDIA forums located here first:

https://gridforums.nvidia.com/

You may also wish to look through the NVIDIA Enterprise Services Knowledgebase to further find support articles and links here:

https://nvidia-esp.custhelp.com/app/answers/list/autologout/1

Keep in mind that not all issues within your deployment may be answered in the NVIDIA vGPU forums. You may also have to reference forums from the hardware supplier, the hypervisor and application themselves. Some examples of key forums to look through are here:

VMware Forums: https://communities.vmware.com/welcome

Citrix Forums: https://discussions.citrix.com/

HPE ProLiant Server Forums: https://community.hpe.com/t5/ProLiant/ct-p/proliant

Dell Server Forums: https://www.dell.com/community/Servers/ct-p/ESServers

Lenovo Server Forums: https://forums.lenovo.com/t5/Datacenter-Systems/ct-p/sv_eg

Autodesk Knowledge Network: https://knowledge.autodesk.com/

Adobe Forums: https://forums.adobe.com/welcome

Dassault Systèmes User Groups: https://www.3ds.com/support/users-communities/

13.2 Filing a Bug Report

When filing a bug or requesting support assistance, it is critical to include information about the environment, so that the technical staff that can help you resolve the issue. NVIDIA includes the nvidia-bug-report.sh script within the. vib installation package to collect and package this critical information. The script collects the following information:

- VMware version
- X.Org log and configuration
- PCI information
- CPU information
- GPU information
- **esxcfg** information for PLX devices
- **esxcfg** information for GPU devices
- VIB information
- NVRM messages from vmkernel.log
- System **dmesg** output
- Which virtual machines have vGPU or vSGA configured
- NSMI output

When running this script:

- You may specify the output location for the bug report using either the -o or –output switch followed by the output file name. If you do not specify an output directory, the script will write the bug report to the current directory.
- If you do not specify a file name, the script will use the default name nvidia-bug-report.log.gz.
- If the selected directory already contains a bug report file, then the script will change the name of that existing report file to nvidia-bug-report.log.old.gz before generating a new nvidia-bug-report.log.gz file.

To collect a bug report, issue the command:

```
$ nvidia-bug-report.sh
```

The system displays the following message during the collection process:

```
nvidia-bug-report.sh will now collect information about your system and
create the file 'nvidia-bug-report.log.gz' in the current directory. It may
take several seconds to run. In some cases, it may hang trying to capture
data generated dynamically by the vSphere kernel and/or the NVIDIA kernel
module. While the bug report log file will be incomplete if this happens, it
may still contain enough data to diagnose your problem.
```

Be sure to include the **nvidia-bug-report.log.gz log** file when reporting problems to NVIDIA.

Appendix A. About This Document

A.1 Related Documentation

Refer to the NVIDIA Virtual GPU (vGPU) resources page http://www.nvidia.com/gridresources for additional information about NVIDIA vGPU technology, including:

- NVIDIA Virtual GPU Technology

  https://www.nvidia.com/en-us/design-visualization/technologies/virtual-gpu/
- Purchasing Guide for NVIDIA vGPU Solutions

  https://www.nvidia.com/en-us/design-visualization/buy-grid/
- NVIDIA GPU Datasheets

  http://www.nvidia.com/object/grid-enterprise-resources.html#datasheet
- Application Deployment Guides and Solution Overviews

  http://www.nvidia.com/object/grid-enterprise-resources.html#guides
- Customer Success Stories

  http://www.nvidia.com/object/grid-enterprise-resources.html#case
- White Papers

  http://www.nvidia.com/object/grid-enterprise-resources.html#whitepapers
- Videos

  http://www.nvidia.com/object/grid-enterprise-resources.html#videos

A.2 Support Contact Information

For technical support there are several resources to assist you:

For community support, please post questions (and answers!) on our respective forums:

- VMware vGPU Community:
  - *https://communities.vmware.com/community/vmtn/vmware-nvidia-direct-access-program*

In addition, you should reach out to your local VMware Horizon and NVIDIA vGPU teams for guidance.

Should you find a bug please follow the steps in Section 12.2 to create a bug report and then submit to the VMware vGPU Community site: *https://communities.vmware.com/community/vmtn/vmware-nvidia-direct-access-program*

For support when architecting your solution, your VMware Horizon and NVIDIA vGPU teams are available to assist. Please be sure you are in touch with them and keep them up to date with your progress. If you do not know your correct account management teams, please reach out to the appropriate email below:

- NVIDIA vGPU team: gridteam@nvidia.com

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

Appendix B. Installing & Licensing NVIDIA Driver in Linux Virtual Desktop

B.1 Installing NVIDIA Driver in Linux Virtual Desktop

Installation in a VM: After you create a Linux VM on the hypervisor and boot the VM, install the NVIDIA vGPU software display driver in the VM to fully enable GPU operation.

Installation on bare metal: When the physical host is booted before the NVIDIA vGPU software display driver is installed, the vesa Xorg driver starts the X server. If a primary display device is connected to the host, use the device to access the desktop. Otherwise, use secure shell (SSH) to log in to the host from a remote host. If the Nouveau driver for NVIDIA graphics cards is present, disable it before installing the NVIDIA vGPU software display driver.

Installation of the NVIDIA vGPU software display driver for Linux requires:

Compiler toolchain

Kernel headers

1. Copy the NVIDIA vGPU software Linux driver package, for example NVIDIA-Linux_x86_64-390.75-grid.run, to the guest VM or physical host where you are installing the driver.

2. Before attempting to run the driver installer, exit the X server and terminate all OpenGL applications.

   a) On Red Hat Enterprise Linux and CentOS systems, exit the X server by transitioning to runlevel 3:

   ```
   [nvidia@localhost ~]$ sudo init 3
   ```

   b) On Ubuntu platforms, do the following:

      i. Use **CTRL-ALT-F1** to switch to a console login prompt.

      ii. Log in and shut down the display manager:

```
[nvidia@localhost ~]$ sudo service lightdm stop
```

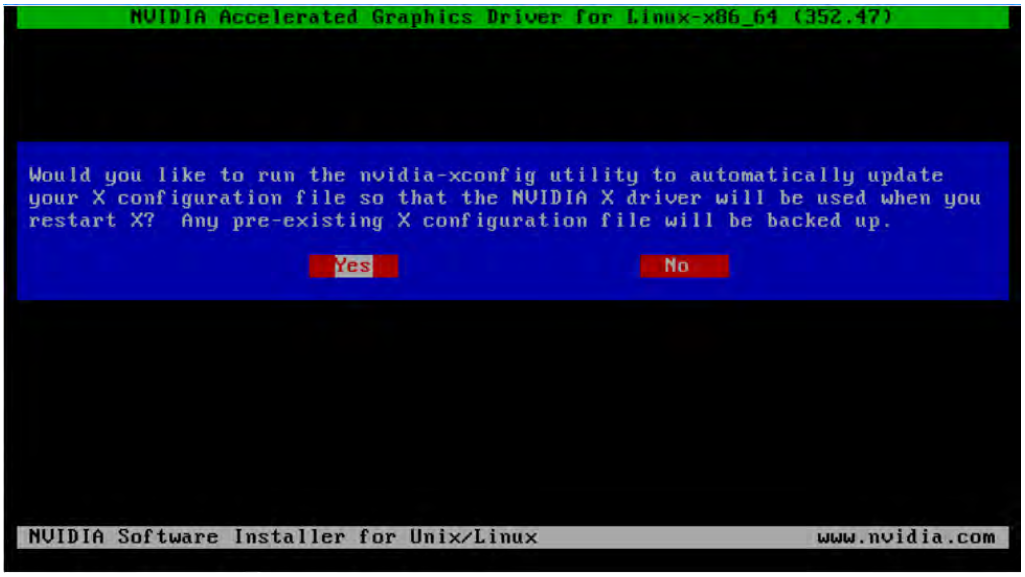
3. From a console shell, run the driver installer as the root user.

```
sudo sh ./ NVIDIA-Linux_x86_64-352.47-grid.run
```

In some instances, the installer may fail to detect the installed kernel headers and sources. In this situation, re-run the installer, specifying the kernel source path with the --kernel-source-path option:

```
sudo sh ./ NVIDIA-Linux_x86_64-352.47-grid.run \
–kernel-source-path=/usr/src/kernels/3.10.0-229.11.1.el7.x86_64
```

4. When prompted, accept the option to update the X configuration file (xorg.conf).



5. Once installation has completed, select **OK** to exit the installer.

6. Verify that the NVIDIA driver is operational.

   a) Reboot the system and log in.

   b) Run nvidia-settings.

```
[nvidia@localhost ~]$ nvidia-settings
```

The NVIDIA X Server Settings dialog box opens to show that the NVIDIA driver is operational.

Installation in a VM: After you install the NVIDIA vGPU software display driver, you can license any NVIDIA vGPU software licensed products that you are using. For instructions, refer to Virtual GPU Client Licensing User Guide.

B.2 Licensing NVIDIA vGPU on Linux

1. Start NVIDIA X Server Settings by using the method for launching applications provided by your Linux distribution. For example, on Ubuntu Desktop, open the Dash, search for NVIDIA X Server Settings, and click the **NVIDIA X Server Settings** icon.
2. In the NVIDIA X Server Settings window that opens, click **Manage License**. The License Edition section of the NVIDIA X Server Settings window shows that NVIDIA vGPU is currently unlicensed.
3. In the **Primary Server** field, enter the address of your primary NVIDIA vGPU software License Server. The address can be a fully qualified domain name such as gridlicense1.example.com, or an IP address such as 10.31.20.45. If you have only one license server configured, enter its address in this field.
4. Leave the **Port Number** field under the **Primary Server** field unset. The port defaults to 7070, which is the default port number used by NVIDIA vGPU software License Server.
5. In the **Secondary Server** field, enter the address of your secondary NVIDIA vGPU software License Server. If you have only one license server configured, leave this field unset. The address can be a fully qualified domain name such as gridlicense2.example.com, or an IP address such as 10.31.20.46.
6. Leave the **Port Number** field under the **Secondary Server** field unset. The port defaults to 7070, which is the default port number used by NVIDIA vGPU software License Server.

7. Click **Apply** to assign the settings. The system requests the appropriate license for the current vGPU from the configured license server.
8. The vGPU within the VM should now exhibit full frame rate, resolution, and display output capabilities. The VM is now capable of running the full range of DirectX and OpenGL graphics applications.
9. If the system fails to obtain a license, see Virtual GPU Client Licensing User Guide for guidance on troubleshooting.

NVIDIA Corporation | 2788 San Tomas Expressway, Santa Clara, CA 95051

http://www.nvidia.com