

Accelerate FlashStack with the Cisco UCS X440p PCIe Node

Contents

Executive summary	3
Overview.....	3
Data center solution design with FlashStack.....	4
Cisco Intersight managed mode configuration.....	7
Cisco UCS X440p GPU node hardware and software requirements	8
NVIDIA driver installation and licensing requirements	18
vGPU installation requirements and verification on VMware.....	20
vGPU installation requirements and verification on Microsoft Windows Server 2022 as a virtual machine on VMware.....	30
vGPU installation requirements and verification on an RHEL 8.6 server as a VMware ESXi host virtual machine	41
GPU installation requirements and verification on FlashStack bare-metal Microsoft Windows Server 2016 and 2022.....	48
GPU installation requirements and verification on FlashStack bare-metal RHEL 8.6.....	52
GPU installation requirements and verification on FlashStack bare-metal Ubuntu 20.04.4	55
For more information.....	56

Executive summary

The Cisco UCS® X-Series Modular System simplifies your data center, adapting to the unpredictable needs of modern applications while also providing for traditional scale-out and enterprise workloads. The X-Series supports both blade and rack server functions by offering computing density, storage capacity, and expandability in a single system. This modular system is managed from the cloud using the Cisco Intersight™ platform. It has an adaptable, future-ready, modular design that meets the needs of modern applications and improves operational efficiency, agility, and scalability. Powered by the Cisco Intersight cloud-operations platform, it shifts your focus from administrative details to business outcomes with hybrid-cloud infrastructure that is assembled from the cloud, shaped to your workloads, and continuously optimized.

The Cisco UCS X-Series was designed to be modular to offer adaptability and workload flexibility. With the introduction of the PCIe node and Cisco UCS X-Fabric Technology, the X-Series now supports workloads that require graphics processing units (GPUs). The unique architecture simplifies the addition and removal of GPUs to computing nodes. The Cisco UCS X440p PCIe Node allows you to add up to four GPUs to a Cisco UCS X210c M6 Compute Node with X-Fabric Technology.

This document helps Cisco customers and business partners deploy the Cisco UCS X440p PCIe Node on the Cisco UCS X210c M6 Compute Node in Cisco UCS X-Series systems in a FlashStack environment. The document describes the X440p PCIe Node and its components and the hardware and software requirements for installing this node on the X210c M6 Compute Node. The document also shows how to verify the installed X440p PCI Node on servers in the Cisco Intersight interface after installation.

This document demonstrates the installation and verification of multiple operating systems on the X210c M6 Compute Node with the X440p PCIe Node installed. The document also explains how to verify multiple operating systems on the X210c M6 Compute Node with the X440 PCIe Node installed and describes the NVIDIA virtual GPU (vGPU) capability on VMware vSphere, which allows customers to share a single GPU across multiple virtual machines. For each operating system, this document presents the main steps required to boot from Pure Storage and install drivers and licensing from NVIDIA.

Overview

The Cisco UCS X-Series provides a distinctive architecture through which users can customize the X-Series chassis based on their application needs. The X-Series chassis is designed to add resources to the Cisco UCS X210c Compute Node by using specific PCIe nodes. These PCIe nodes on the X-Series chassis can be connected to the compute nodes using the Cisco UCS X9416 X-Fabric Module. The X9416 serves as a high-speed PCIe connector to the X210c node. This unique architecture seamlessly adds resources to the X-Series compute nodes based on application needs. The PCIe node module can be a dedicated GPU module, a memory module (future), or a disk space module (future).

The Cisco UCS X440p PCIe node with GPUs can be integrated into the Cisco UCS X-Series Modular System for applications that require the additional processing power of the GPUs. The Cisco UCS X440p PCIe Node discussed in this document contains NVIDIA Tesla T4 GPUs. This document focuses on integration of the Cisco UCS X440p PCIe Node into a FlashStack environment. Customers interested in understanding the FlashStack design and deployment details presented in this document should refer to the following Cisco® Validated Design:

https://www.cisco.com/c/en/us/td/docs/unified_computing/ucs/UCS_CVDs/FlashStack@_vsi_vmware70_ucsx.html.

Data center solution design with FlashStack

The FlashStack solution is a validated, converged infrastructure developed jointly by Cisco and Pure Storage. The solution offers a predesigned data center architecture that incorporates computing, storage, and network design best practices to reduce IT risk by validating the architecture and helping ensure compatibility among the components. The FlashStack solution is successful because of its ability to evolve and incorporate technology and product innovations in the areas of management, computing, storage, and networking. This document discusses how to incorporate the Cisco UCS X-Series Modular System into the FlashStack virtual server infrastructure (VSI) and how you can manage and orchestrate FlashStack components from the cloud using the Cisco Intersight™ platform.

Important advantages of integrating the Cisco UCS X-Series into the FlashStack infrastructure include the following:

- Simpler and programmable infrastructure: Infrastructure as code delivered through an open API
- Power and cooling innovations: More power headroom and less energy loss through 54-volt (V) DC power delivery to the chassis
- Better airflow: Midplane-free design with fewer barriers, and thus lower impedance
- Fabric innovations: PCIe and Compute Express Link (CXL) topology for heterogeneous computing and memory composability
- Innovative cloud operations: Continuous feature delivery and no need to manage virtual machines
- Investment protection: Ready for future technologies such as liquid-cooling and high-wattage CPUs; CXL ready
- Exceptional performance density: Greater consolidation of business applications
- Simplified management
- Always-on data protection: Protected with safe mode
- Reduced environmental, social, and governance (ESG) environment footprint: Less environmental impact through smart product design
- Evergreen stateless architecture: Nondisruptive upgrade (NDU) for everything

In addition to the computing-specific hardware and software innovations, integration of the Cisco Intersight cloud platform with VMware vCenter and Pure Storage FlashArray//X delivers monitoring, orchestration, and workload optimization capabilities for different layers (virtualization and storage) of the FlashStack solution. The modular nature of the Cisco Intersight platform also provides an easy upgrade path to additional services such as workload optimization and Kubernetes.

Customers interested in understanding FlashStack design and deployment details, including the configuration of various elements and associated best practices, should refer to the Cisco Validated Design for FlashStack at <https://www.cisco.com/c/en/us/solutions/design-zone/data-center-design-guides/data-center-design-guides-all.html> - FlashStack.

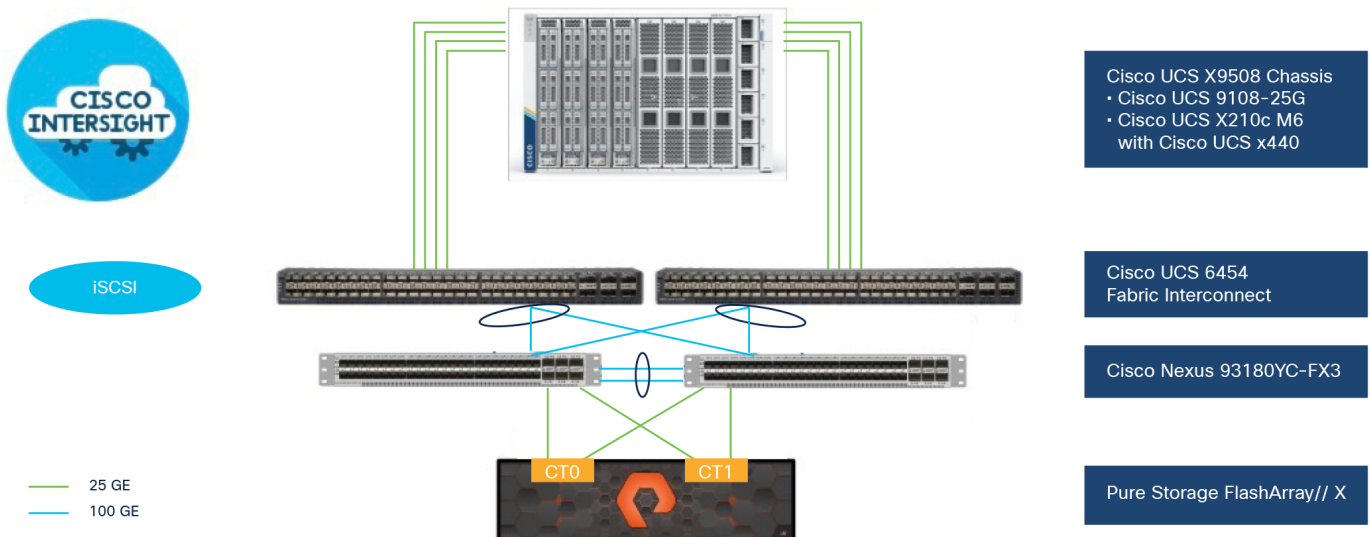
Design hardware and software

The FlashStack solution provides the flexibility to connect a Pure FlashArray//X controller to a Cisco UCS chassis using either Fibre Channel-based storage access or IP-based storage access. Figure 1 shows the physical connectivity for both cases. For this document, all the servers use boot from SAN and Small Computer System Interface over IP (iSCSI) storage. Several operating systems are installed on the Cisco UCS X210c Compute Nodes, and GPUs from the Cisco UCS X440p PCIe Node are available for use by the operating system.

The following components are used in this setup:

- Cisco UCS X440p PCIe Node contains two Tesla T4 GPUs presented to the adjacent compute node for 1:1 mapping.
- Cisco UCS 6454 Fabric Interconnects provide the chassis and network connectivity.
- Cisco UCS X9508 Chassis connects to the fabric interconnects using Cisco UCS 9108 25G Intelligent Fabric Modules (IFMs), where four 25 Gigabit Ethernet ports are used on each IFM to connect to the appropriate fabric interconnect. If additional bandwidth is required, all eight 25 Gigabit Ethernet ports can be used.
- Cisco UCS X210c M6 Compute Nodes contain the fourth-generation Cisco UCS Virtual Interface Card (VIC) 14425.
- Cisco Nexus® 93180YC-FX3 Switches in Cisco NX-OS mode provide the switching fabric.
- Cisco UCS 6454 Fabric Interconnect 100 Gigabit Ethernet uplink ports connect to Cisco Nexus 93180YC-FX3 Switches in a virtual port-channel (vPC) configuration.
- Pure FlashArray//X controller connects to the Cisco Nexus 93180YC-FX3 Switches using two 25 Gigabit Ethernet iSCSI ports from each controller, providing four paths for iSCSI target reachability.

Physical topology for IP-based FlashStack



Physical topology for Fibre Channel-based FlashStack

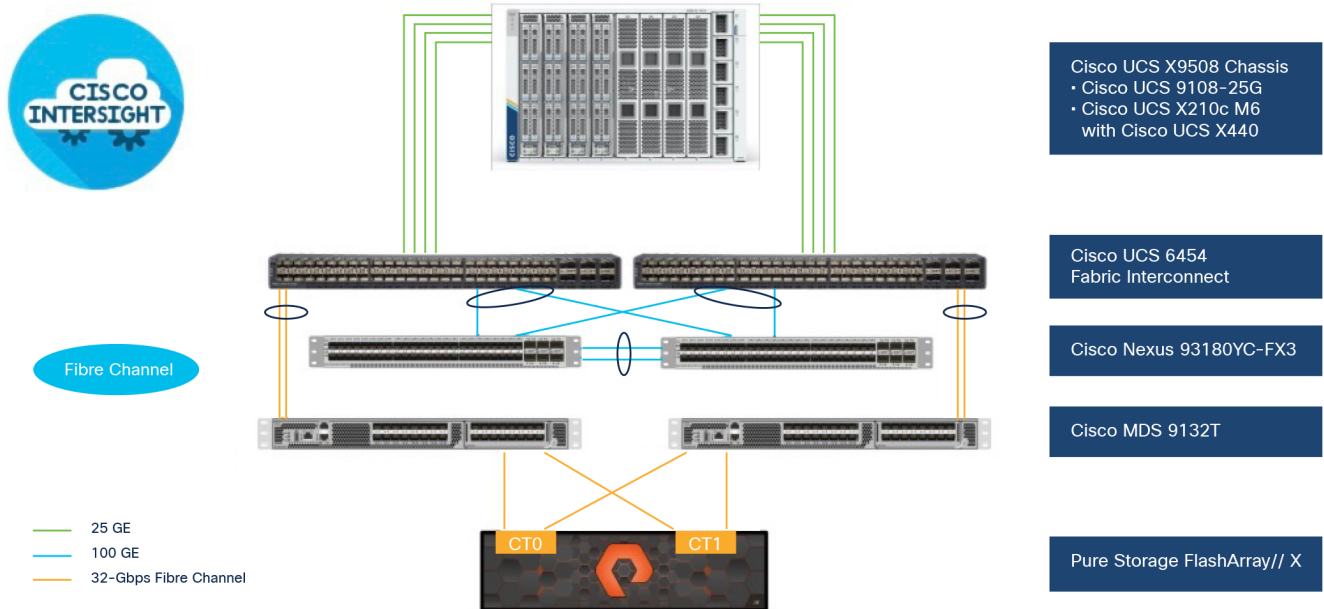


Figure 1.
IP- and Fibre Channel-based storage access

Physical connectivity of FlashStack setup for Cisco UCS X-Series chassis with Cisco UCS X440p PCIe Node

Table 1 lists the hardware and software versions used to validate the solution described here. Note that the validated FlashStack solution explained in this document adheres to the Cisco, Pure Storage, NVIDIA, and VMware interoperability matrix to determine support for various software and driver versions. You can use the same interoperability matrix to determine support for components that are different from the current validated design.

Table 1. Hardware and software versions used in this solution

	Component	Software version	Comments
Network	Cisco Nexus 93180YC-FX3	Release 9.3(7)	Download from Cisco.com
Computing	Cisco UCS Fabric Interconnect 6454 and Cisco UCS 9108 25G IFM	Release 4.2(2b)	Cisco UCS X-Series GA release for infrastructure including fabric interconnects and I/O module (IOM) and IFM
	Cisco UCS X210c with Cisco UCS VIC 14425	Release 5.0(2b)	
	VMware ESXi	Release 7.0 U2	Custom image for ESXi 7.0 U2 installation ISO
	VMware vCenter	Enterprise Plus Edition (required)	
	Cisco VIC driver for ESXi	Release 1.0.35.0	Download from Cisco.com under Cisco UCS X-Series software driver downloads

	Microsoft Windows	Release 2016, 2019, and 2022	Used as bare metal and virtual machine
	Red Hat Enterprise Linux (RHEL)	Release 8.6	Used as bare metal and virtual machine
	Ubuntu Linux	Release 20.04.4	Used as bare metal and virtual machine
Storage	Pure Storage FlashArray//X	Purity//FA 6.3.1	
NVIDIA vGPU drivers	VMware ESXi	vGPU Manager Release 510.85.02vG Product Version 14.2 Platform Version 7.0	Requires NVIDIA account; download from Nvidia.com and refer release notes at https://docs.nvidia.com/grid/index.html
NVIDIA AI Enterprise (NVAIE) drivers	VMware ESXi	NVAIE Manager Release 510.85.03	
	Microsoft Windows 2016, 2019, and 2022	516.31-NVIDIA-data-center-tesla-desktop-winsrv-2016-2019-2022-dch-international	https://ui.licensing.nvidia.com/software
	RHEL	Release 8.6 NVIDIA-Linux-x86_64-510.85.02-grid.run	https://ui.licensing.nvidia.com/software
	Ubuntu	20.04.4-live-server-amd64 NVIDIA-GRID-Ubuntu-KVM-510.73.06-510.73.08-512.78	https://ui.licensing.nvidia.com/software

Cisco Intersight managed mode configuration

The Cisco Intersight platform is a management solution delivered as a service with embedded analytics for Cisco and third-party IT infrastructure. The Cisco Intersight managed mode (also referred to as Cisco IMM or Intersight managed mode) is a new architecture that manages Cisco Unified Computing System™ (Cisco UCS) fabric interconnect-attached systems through a Redfish-based standard model. To deploy and test server profiles for different operating systems, Intersight managed mode was used.

The following guide was referenced to create all the policies to boot the server from storage:

https://www.cisco.com/c/en/us/td/docs/unified_computing/ucs/UCS_CVDs/FlashStack@_vsi_vmware70_ucsx.html.

When creating the server profile, refer to the following link for system BIOS and CPU optimization policies; the settings depend on whether you are using a bare-metal server or ESX virtualization:

<https://www.cisco.com/c/en/us/products/collateral/servers-unified-computing/ucs-b-series-blade-servers/performance-tuning-guide-ucs-m6-servers.html>.

Note: Refer to Table 6 in the performance tuning guide for various settings to tweak the BIOS for better CPU optimization.

Cisco UCS X440p GPU node hardware and software requirements

The Cisco UCS X9508 Chassis has eight node slots. Up to four of these slots can be X440p PCIe Nodes when paired with a Cisco UCS X210c M6 Compute Node.

The Cisco UCS X440p PCIe Node (Figures 2 and 3) supports two x16 full-height, full-length dual-slot PCIe cards or four x8 full-height, full-length single-slot PCIe cards and requires both Cisco UCS X9416 X-Fabric modules for PCIe connectivity. This configuration provides up to 16 GPUs per chassis to accelerate your applications. If your application needs even more GPU acceleration, you can add up to two more GPUs on each compute node using the optional GPU front mezzanine card on the X210c Compute Node. The vertically oriented X210c Compute Nodes and the X440p PCIe Nodes connect directly to horizontally oriented X-Fabric modules. The Cisco UCS X440p PCIe Node is a full-height module that supports GPU computing offload for Cisco UCS X-Series Compute Nodes. The module is hot swappable and supports PCIe connectivity for a variety of GPU form factors. This feature enables seamless upgrades to future computing nodes, resource nodes, and X-Fabric modules without requiring a major system upgrade.

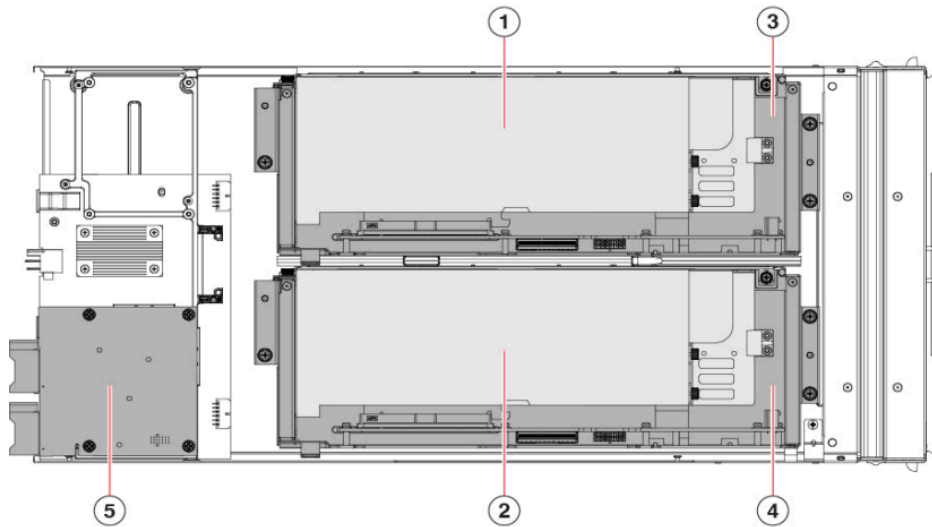


Front view



Rear view

Figure 2.
Cisco UCS X440p PCIe Node



1	Riser slot 1 Supports both Type A and Type B risers.	2	Riser slot 2 Supports both Type A and Type B risers.
3	GPU slot 1 (FHFL GPU shown) Supports either FHFL or HHHL GPU depending on the riser type.	4	GPU slot 2 (FHFL GPU shown) Supports either FHFL or HHHL GPU depending on the riser type.
5	mezzanine connector (included)	-	-

Figure 3.

Top view of the Cisco UCS X440p PCIe Node system board

Cisco UCS X440p supports the following GPU options:

- NVIDIA A100 Tensor Core GPU (dual slot)
- NVIDIA A16 GPU (dual slot)
- NVIDIA A40 GPU (dual slot)
- NVIDIA T4 Tensor Core GPU (single slot)

System requirements

Note the following guidelines and limitations when installing the Cisco UCS X440p PCIe Node:

- The X440p PCIe Node is supported in the Cisco UCS X9508 Chassis only. Do not attempt to install the PCIe node in any other Cisco UCS server chassis.
- Each X440p PCIe Node must be paired with an X-Series Compute Node and therefore has specific configurations. Either of the following are supported. Be aware of these configurations, especially when removing and installing computing nodes and PCIe nodes in your deployment.
 - The PCIe node must be installed in a pair with a compute node. The module and the node work together to provide computing and GPU processing acceleration.
 - When you install, replace, or upgrade the PCIe node, make sure that you install it next to a supported companion module, such as a Cisco UCS X210c Compute Node.
 - Cisco X-Fabric Technology does not limit or restrict slot use. PCIe nodes are not required. Computing nodes can be installed in any slot without restriction where PCIe nodes are not used.

The X-Fabric Technology solution has software, hardware, and chassis slot installation requirements that must be met for the X440p to be detected. GPU nodes can only be installed adjacent to X210c nodes. A chassis can either have an X440p on all even slots or on all odd slots. Either of the two node placement configurations shown in Figure 4 are required for a GPU node to be detected by the chassis.

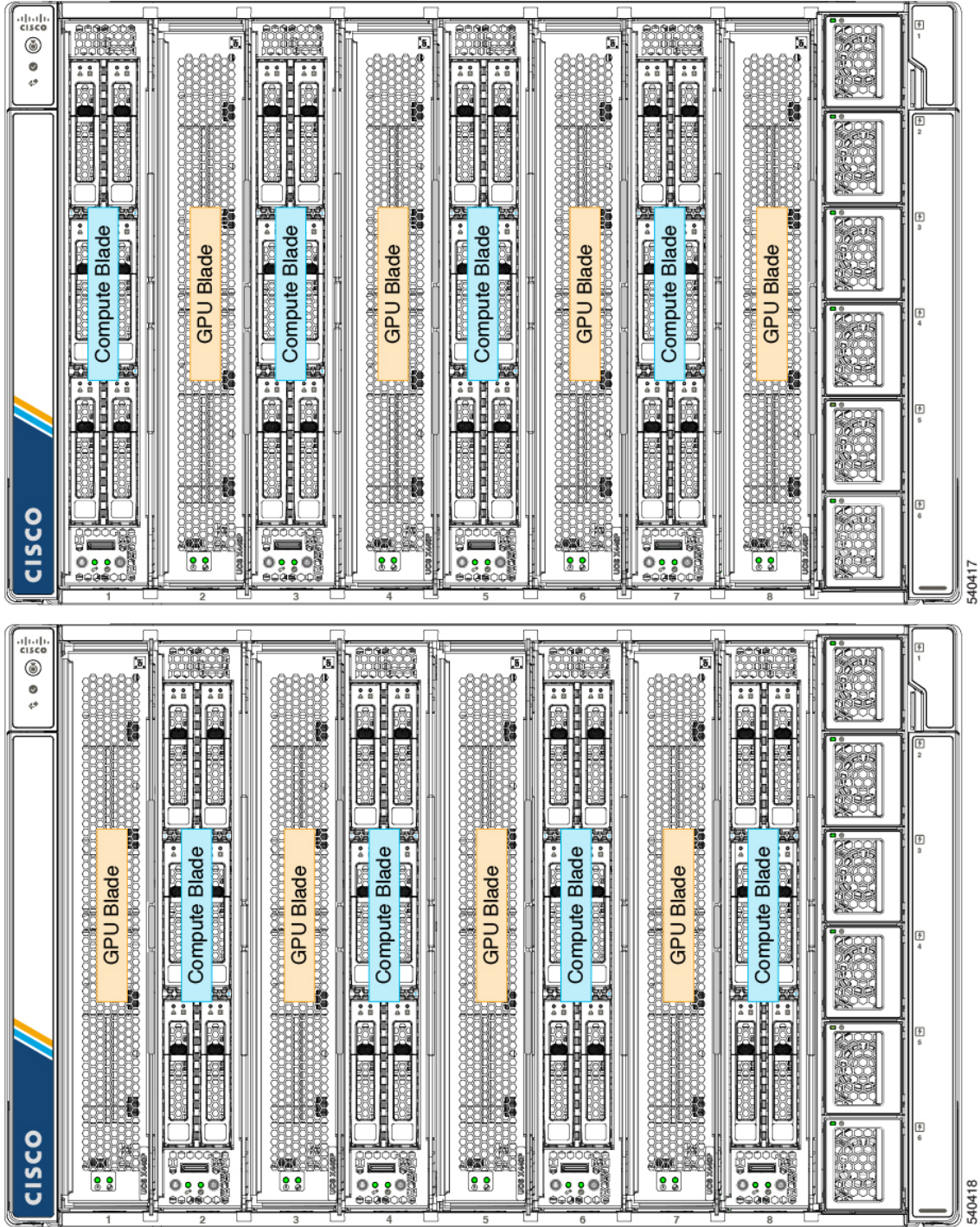


Figure 4.
GPU node installation configurations

Hardware requirements

For the X440p to be detected, four components are needed in addition to the X210c node already installed in the X-Fabric chassis:

- Cisco UCS X440p module with a GPU selected
- GPU riser cards
- Cisco UCS X9416 X-Fabric modules
- Rear mezzanine VIC and bridge adapters

The X440p node consists of a GPU card and the riser module. The X210c node needs a Cisco UCS PCI mezzanine card for Cisco UCS X-Fabric or a rear-mezzanine VIC, and the connectivity between the GPU node and the compute node requires both X9416 X-Fabric modules.

The following sections provide details about each component.

Cisco UCS X440p PCIe Node hosting the GPU

The base Cisco UCS X440p PCIe Node, part number UCSX-440P, does not include any components or options. Appropriate components or options must be selected during product ordering from the list of available PCIe GPU cards, shown in Table 2.

Table 2. Available PCIe GPU cards

GPU part number	Part description	Riser slot compatibility
UCSX-GPU-T4-16 ¹	NVIDIA T4 PCIE 75W 16 GB	Riser 1B (Gen 4), Riser 2B (Gen 4)
UCSX-GPU-A16 ²	NVIDIA A16 PCIE 250W 4 X 16 GB	Riser 1A (Gen 4), Riser 2A (Gen 4)
UCSX-GPU-A40 ²	TESLA A40 RTX, PASSIVE, 300W, 48 GB	Riser 1A (Gen 4), Riser 2A (Gen 4)
UCSX-GPU-A100-80 ²	TESLA A100, PASSIVE, 300W, 80 GB	Riser 1A (Gen 4), Riser 2A (Gen 4)

Notes:

1. The maximum number of GPUs per node is 4.
2. The maximum number of GPUs per node is 2.
3. Required power cables are included with the riser cards.

Note: Each PCIe node must be configured with the same type of GPU. Mixed use of a x16 and x8 PCIe cards in the same PCIe node is not supported.

The riser card is selected based on the GPU node selected, as shown in Table 3. The PCIe node requires both the risers to be configured.

Table 3. Available riser cards

Part number	Part description
UCSX-RIS-A-440P	Riser A for 1 dual-slot GPU per riser, 440P PCIe Node <ul style="list-style-type: none"> Riser 1A (controlled with CPU1 on X210c) Riser 2A (controlled with CPU2 on X210c)
UCSX-RIS-B-440P	Riser B for 2 single-slot GPUs per riser, 440P PCIe Node <ul style="list-style-type: none"> Riser 1B (controlled with CPU1 on X210c) Riser 2B (controlled with CPU2 on X210c)

Note: Each PCIe node must have the same type of riser: either two Type A or two Type B risers. You cannot mix riser types in the same PCIe node.

Cisco UCS X9416 X-Fabric Module

The Cisco UCS X9416 X-Fabric Module is required for connectivity between the Cisco UCS X210c Compute Node and X440p PCIe Node (Table 4). When a computing node is inserted into the chassis, the computing node's mezzanine card plugs directly into the two fabric module slots (with no midplane) for PCIe connectivity to the X440p PCIe Node.

Table 4. Available X-Fabric Module

Part number	Part description
UCSX-F-9416	X9416 X-Fabric module for 9508 chassis

Rear mezzanine VIC and bridge adapters

The Cisco UCS X210c Compute Node can connect to the X440p PCIe Node through the computing node's mezzanine card. The mezzanine card can be either the Cisco UCS PCIe mezzanine card for X-Fabric connectivity or the Cisco UCS VIC 14825. The VIC 14825 requires a bridge connector to connect the X210c Compute Node to the Cisco UCS 9108 25G IFM for network connectivity.

The X210c Compute Node has one rear mezzanine adapter connector (Table 5).

Table 5. Available VICs and bridge cards

Part number	Part description	Number of CPUs required	Connector type
Cisco VIC			
UCSX-V4-Q25GME	Cisco UCS VIC 14825 4 x 25-Gbps mezzanine card for the X210c Compute Node	2 CPUs required	Rear mezzanine connector on motherboard
UCSX-V4-PCIME1	Cisco UCS PCI mezzanine card for X-Fabric connectivity	2 CPUs required	Rear mezzanine connector on motherboard

Cisco VIC²

UCSX-V4-BRIDGE	Cisco UCS VIC 14000 bridge to connect the Cisco VIC 14425 mLOM and Cisco VIC mezzanine card for the X210c Compute Node	2 CPUs required	One connector on mezzanine card and one connector on mLOM card
----------------	--	-----------------	--

Notes:

1. The rear mezzanine options provide one x16 PCIe connection each from CPU1 to XFM1 and from CPU2 to XFM2.
2. The bridge is included with the Cisco UCS VIC 14825.

Figure 5 provides a simplified block diagram of the Cisco UCS X440p PCIe Node system board.

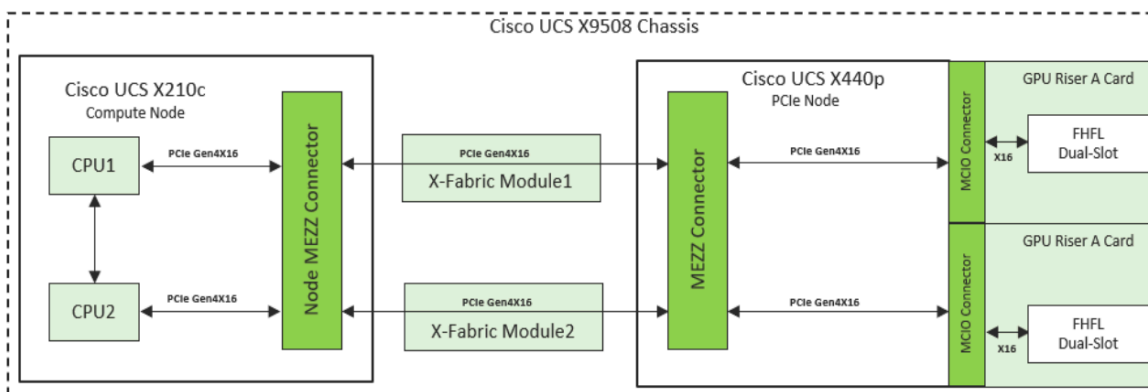


Figure 5.

Cisco UCS X440p PCIe Node simplified block diagram with Riser A

Figure 6 shows a simplified block diagram of the Cisco UCS X 440p PCIe Node system board.

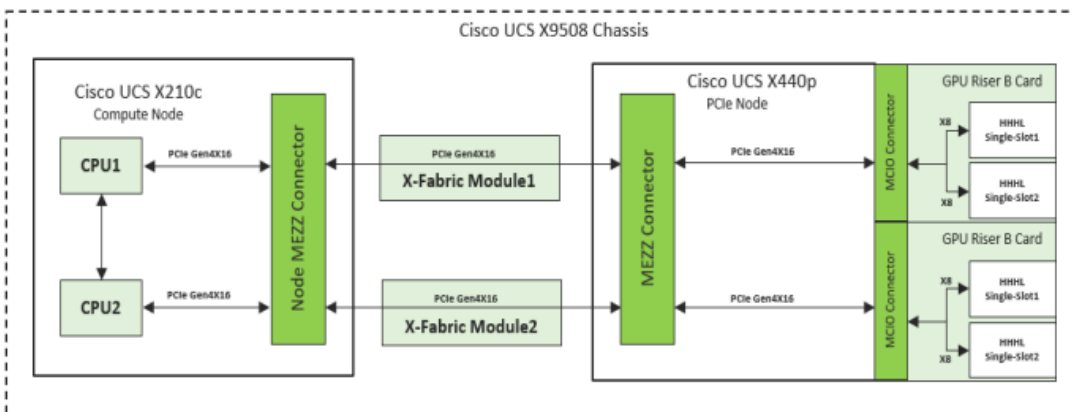


Figure 6.

Cisco UCS X440p PCIe Node simplified block diagram with Riser B

For more details and step-by-step hardware selection and configuration guidance, refer to the X440p specifications sheet at <https://www.cisco.com/c/dam/en/us/products/collateral/servers-unified-computing/ucs-x-series-modular-system/x440p-specsheet.pdf>.

Software requirements

These are the minimum software versions required on the fabric interconnect and the computing node:

- Fabric Interconnects: Release 4.2(2b)
- Cisco UCS X210c Compute Node firmware: Release 5.0(2b)

Firmware versions can be verified using the Cisco Intersight platform and upgraded before installing the Cisco UCS X440p PCIe Node.

Installation verification

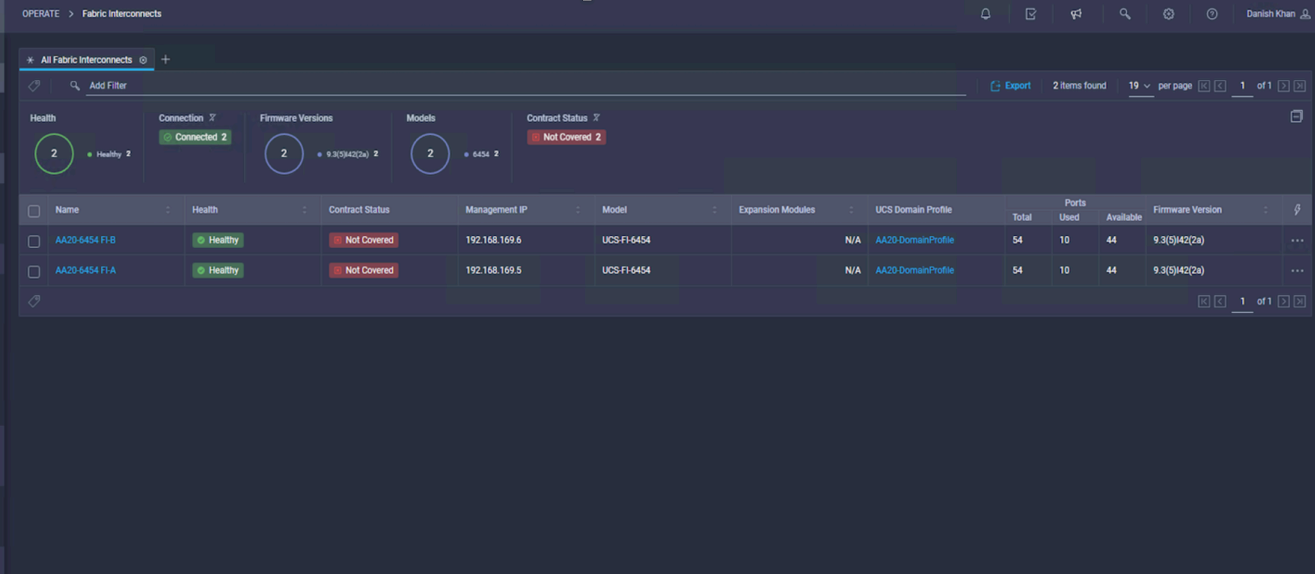
Follow the steps in this section to verify your software and hardware for the X440c node. The module is hot swappable and supports PCIe connectivity for a variety of GPU form factors. When you insert the module, you should be able to detect it as described here.

Software verification

You need to verify your software version.

To verify the software versions running on the fabric interconnect, follow these steps:

1. Log in to Cisco intersight.com.
2. Click the fabric interconnect in the left panel.
3. Select the required fabric interconnect pair and on the firmware version tab verify that the version is 9.3(5)I42(2a.316) or later (Figure 7).



The screenshot shows the Cisco Intersight interface for Fabric Interconnects. It features a summary dashboard with four cards: Health (2 Healthy), Connection (2 Connected), Firmware Versions (2, 9.3(5)I42(2a) 2), and Models (2, 6454 2). Below the dashboard is a table with the following data:

Name	Health	Contract Status	Management IP	Model	Expansion Modules	UCS Domain Profile	Total	Ports Used	Available	Firmware Version
AA20-6454 FI-B	Healthy	Not Covered	192.168.169.6	UCS-FI-6454		N/A AA20 DomainProfile	54	10	44	9.3(5)I42(2a)
AA20-6454 FI-A	Healthy	Not Covered	192.168.169.5	UCS-FI-6454		N/A AA20 DomainProfile	54	10	44	9.3(5)I42(2a)

Figure 7.

Checking the firmware version

4. To verify the Cisco UCS X210c Compute Node firmware, click the server in the left panel and check the firmware version (Figure 8).

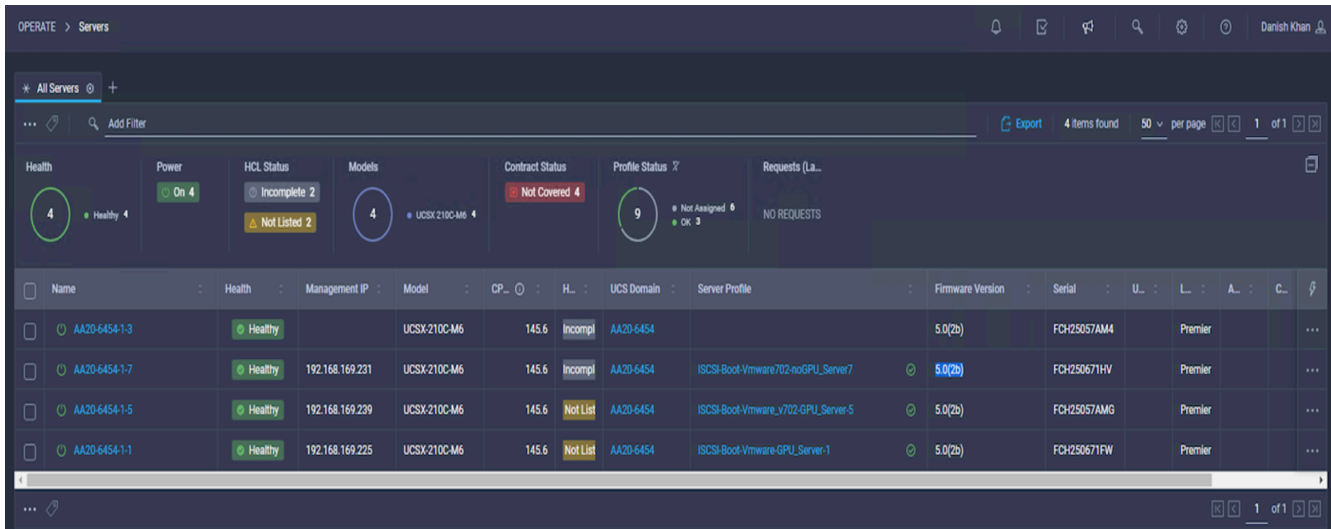


Figure 8.
Checking the Cisco UCS 210c firmware

Hardware verification

After you have confirmed that the software firmware is correct, verify that the hardware is correctly installed and detected on the computing node. Several ways are available to confirm that the node is detected and is ready for server profile and OS installation.

1. Select the correct chassis in the Chassis panel to display the hardware installed. You should see the PCIe node installed in the expected slot (Figure 9).



Figure 9.
PCIe nodes in slot 2 and slot 6 next to X210c Compute Nodes in slot 1 and slot 5

2. The X-Fabric is in the bottom two slots of the chassis. Click the rear view to confirm that the X-Fabric is installed (Figure 10).

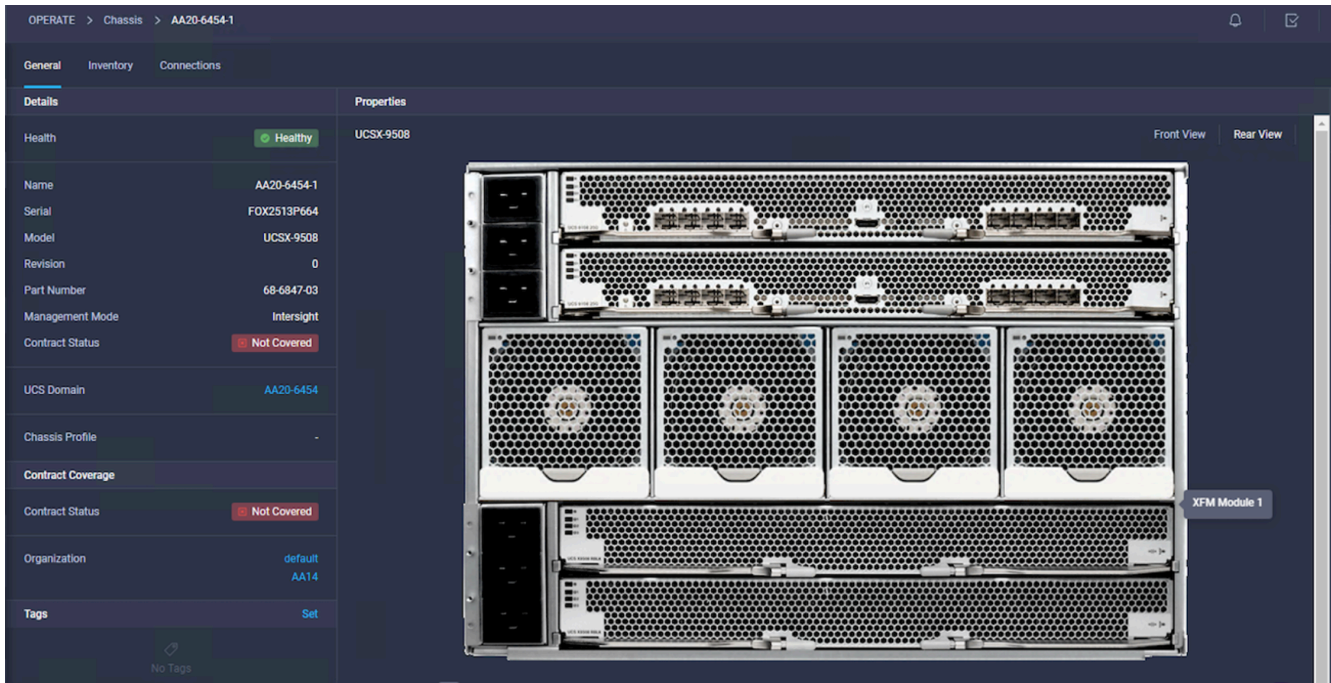


Figure 10.
Rear view to see X-Fabric installation

3. Check the Inventory tab on the chassis X-Fabric Module (XFM) to confirm that the correct version is installed (Figure 11).

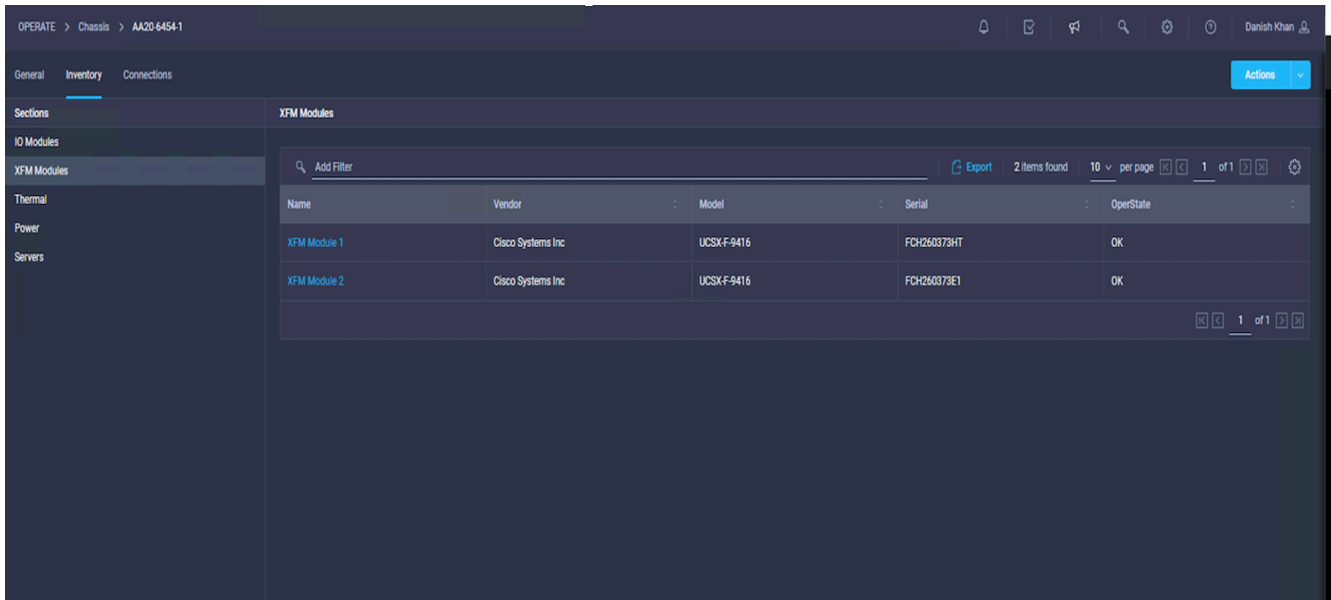


Figure 11.
XFM version confirmation in bottom two slots of the chassis

4. Choose Operate > Servers and the server number in the left panel and click the Inventory tab. Expand the PCIe and GPUs nodes to verify the node installation. Verify the mezzanine card and GPU node installation (Figures 12 and 13).

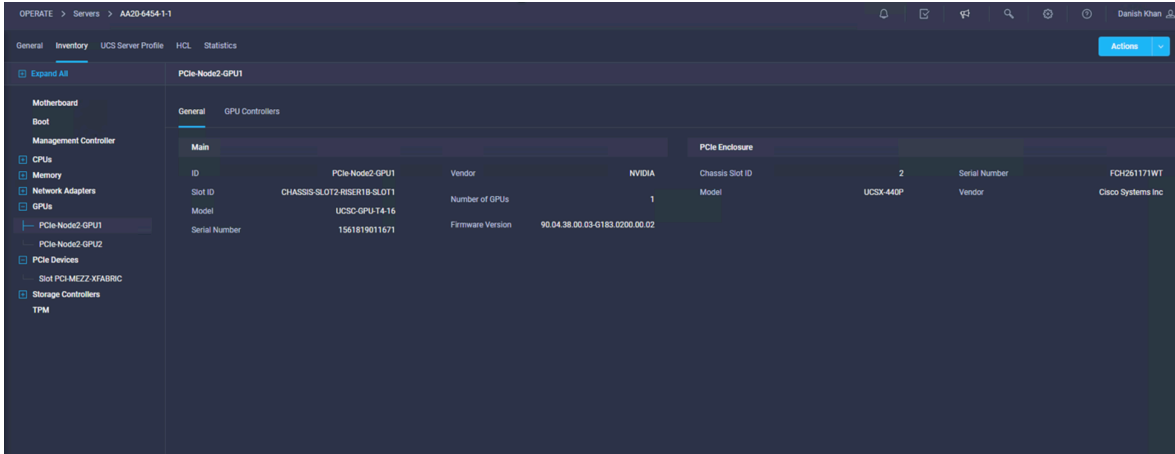


Figure 12.
GPUs and mezzanine cards installed

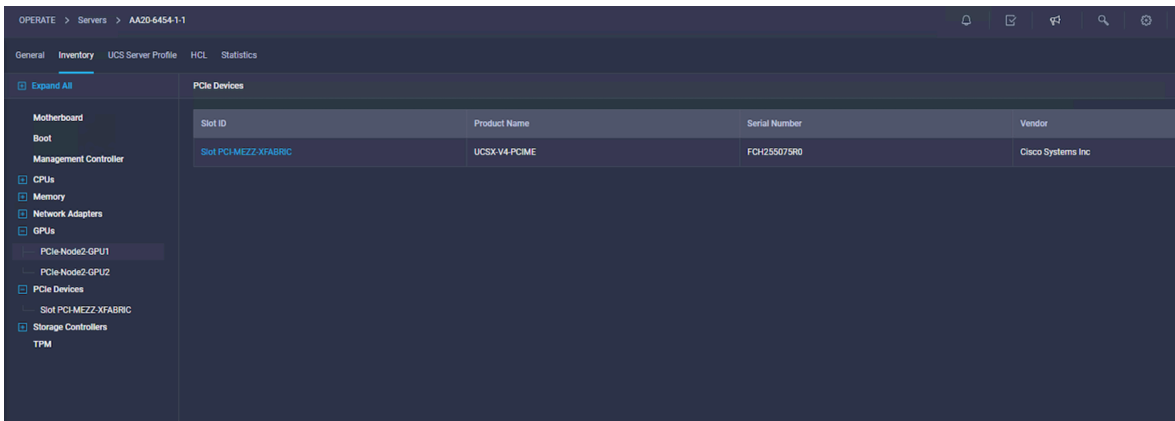


Figure 13.
GPUs and mezzanine cards installed

If you cannot detect the GPU node, you need to rediscover the server node as shown in Figure 14.

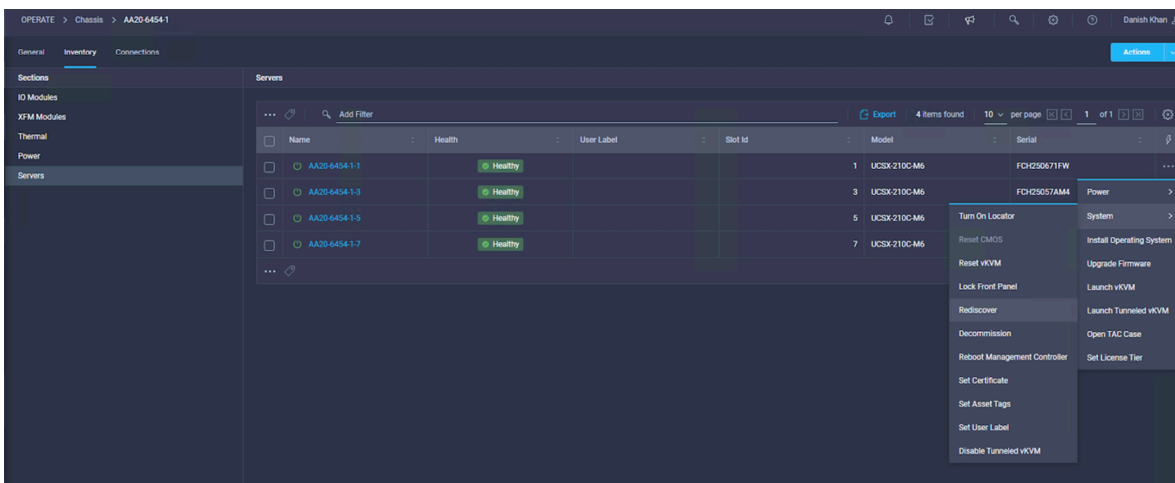


Figure 14.
Where to click to rediscover the server in the chassis

NVIDIA driver installation and licensing requirements

After you have verified the installation of the Cisco UCS X440 PCIe Node hardware and the operating system, you need to install drivers and licenses from NVIDIA for proper functioning. Both are explained in this section using several different operating systems.

NVIDIA driver installation

NVIDIA drivers and licenses are required for the GPUs and virtual GPUs (vGPUs) to work properly. NVIDIA presents two types of drivers: hypervisor drivers, known as host drivers, and client OS (virtual machine) drivers, known as guest drivers. A bare-metal installation of the OS requires just the guest driver, whereas hypervisor platforms such as VMware require host drivers for the hypervisor and different OS drivers for the guest machines matching the version of the hypervisor. Requirements for installing each type of hypervisor driver depend on the type of licensing profile you have purchased from NVIDIA. Profiles and the driver types required are discussed in detail in the next section.

The NVIDIA portal lists drivers either individually or bundled in a compressed file. When you extract the bundle, you obtain both the host and guest drivers (Figure 15). Host and guest driver versions must match the compatibility requirements listed on the NVIDIA website. After the hypervisor drivers are installed, you allocate the GPUs and vGPUs to virtual machines and then install the guest drivers depending on the NVIDIA vGPU profile assigned. You then can operate NVIDIA vGPU Manager software on the host and OS (guest) running various commands to verify the installation and functions.

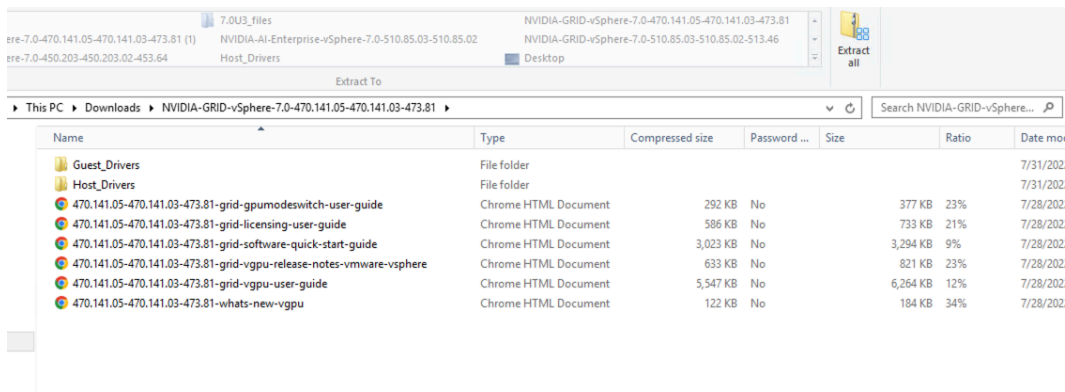


Figure 15. Extracted bundle of NVIDIA drivers showing both guest and host drivers for VMware

NVIDIA profiles and licenses

NVIDIA requires vGPU licenses to virtualize the GPUs presented to different virtual machines. Licenses for vGPUs are based on the number of active running virtual machines. Every time a virtual machine boots, the virtual machine registers itself to the NVIDIA licensing server. Therefore, when acquiring licenses, you need an estimate of the number of active virtual machines you will need.

Starting with NVIDIA Release 13.0, NVIDIA provides two methods for licensing servers:

- Cloud License Service (CLS), the preferred method: A CLS instance is hosted at the NVIDIA Licensing portal.
- Delegated License Service (DLS): A DLS instance is hosted on-premises at a location that is accessible from your private network, such as inside your data center.

Note: A legacy licensing method that reaches end-of-life in 2023 is not covered here.

Refer to the following NVIDIA document for more information about licensing: <https://docs.nvidia.com/license-system/latest/nvidia-license-system-user-guide/index.html#configuring-nls-licensed-client>.

You can obtain a license for four profiles defined according to your application requirements. You generate a client configuration token file and upload it to the required virtual machine for use (this process is demonstrated later in this document).

You can also generate a 90-day evaluation license for testing purposes. See <https://www.nvidia.com/en-us/data-center/resources/vgpu-evaluation/>.

NVIDIA defines vGPU licenses according to four software product types based on application needs and requirements (Table 6):

- NVIDIA RTX Virtual Workstation (vWS)
- NVIDIA Virtual PC (vPC)
- NVIDIA Virtual Applications (vApps)
- NVIDIA Virtual Compute Server (vCS)

Table 6. NVIDIA license types

NVIDIA vGPU software licensed product	Target users	Supported NVIDIA vGPU software deployments
vApps	Users of PC-level applications and server-based desktops that use Citrix Virtual Apps and Desktops, VMware Horizon, Remote Desktop Session Host (RDSH), or other application-streaming or session-based solutions	<ul style="list-style-type: none"> • A-series NVIDIA vGPUs • GPU passthrough • Microsoft Discrete Device Assignment (DDA) • VMware Virtual Dedicated Graphics Acceleration (vDGA) • Bare metal
vPC	Users of business virtual desktops who require a great user experience with PC applications for Microsoft Windows, web browsers, and high-definition video	<ul style="list-style-type: none"> • B-series NVIDIA vGPUs • Microsoft RemoteFX vGPU • VMware Virtual Shared Graphics Acceleration (vSGA)
vCS	Users of computing-intensive virtual servers for artificial intelligence (AI), deep learning, or high-performance computing (HPC) workloads	<ul style="list-style-type: none"> • C-series NVIDIA vGPUs • GPU passthrough • Microsoft DDA • VMware vDGA
vWS	Users of mid-range and high-end workstations who require access to remote professional graphics applications with full performance on any device anywhere	<ul style="list-style-type: none"> • Q-series NVIDIA vGPUs • C-series NVIDIA vGPUs • B-series NVIDIA vGPUs • GPU passthrough • Microsoft DDA • VMware vDGA • Bare metal

The vCS profile is designed for computing-intensive virtual servers for AI, deep learning, or HPC workloads. It is designed for Linux-based operating systems. Its profile is not supported on Windows servers, and the driver installation process does not detect hardware on a Windows server when you use this profile.

For NVIDIA vPC and vDWS profiles, verify that you have remote desktop access (using Remote Desktop Protocol [RDP]) for the virtual machine before you install the NVIDIA graphic card drivers. Otherwise, after the system installs the driver and reboots, you will see a black screen if you try to access the console from VMware vCenter or using VMware Remote Console (VMRC). Only RDP access will display the screen.

For more information about NVIDIA license types, see these resources:

- <https://docs.nvidia.com/grid/13.0/grid-licensing-user-guide/index.html>
- <https://www.nvidia.com/content/dam/en-zz/Solutions/design-visualization/solutions/resources/documents1/Virtual-GPU-Packaging-and-Licensing-Guide.pdf>

Starting with NVIDIA Release 13.0, vCS is not supported on VMware vSphere. C-series vGPU types are not available. Instead, vCS is supported with NVIDIA AI Enterprise Edition. See <https://docs.nvidia.com/grid/14.0/grid-vgpu-release-notes-vmware-vsphere/index.html#vcs-not-supported>.

The preceding discussion implies that you have separate vGPU Manager drivers (*.vib files) if you want to use vPC and vDWS profiles or the vCS profile on the ESX host, and you cannot use multiple profiles on the same host.

For more information, see the NVIDIA support matrix at <https://docs.nvidia.com/ai-enterprise/latest/product-support-matrix/index.html#support-matrix>.

This document presents the steps for installing vGPU Manager for NVIDIA vPC and vDWS profiles. It also presents the steps for assigning a profile and installing drivers for a Windows Server 2022 virtual machine and RHEL 8.6 virtual machine for these profiles. The document also shows how to install licenses and provides snapshots of verification commands.

This document also shows how to install NVIDIA AI Enterprise vGPU Manager on a separate host, assign the vCS profile, and install drivers and licenses on an RHEL 8.6 virtual machine.

In addition, the document shows how to install a bare-metal operating system and drivers and licenses.

vGPU installation requirements and verification on VMware

After the hardware has been detected and verified and the server profile is installed on Cisco Intersight IMM, you install the VMware ESXi 7.0 U2 platform. The ESXi host resides on Pure storage using iSCSI as the boot option. This ESXi host is then registered to vCenter. To use VMware with a vGPU, you must have the vCenter Enterprise Plus Edition.

After the ESXi installation is complete on the X210c node that contains the X440p PCIe Node, verify that the X440p GPUs are presented to the ESXi host software by the hardware.

1. In vCenter, navigate to Host > Configuration > Hardware > Graphics. Figure 16 shows two Tesla T4 cards presented to the ESXi software by the hardware.

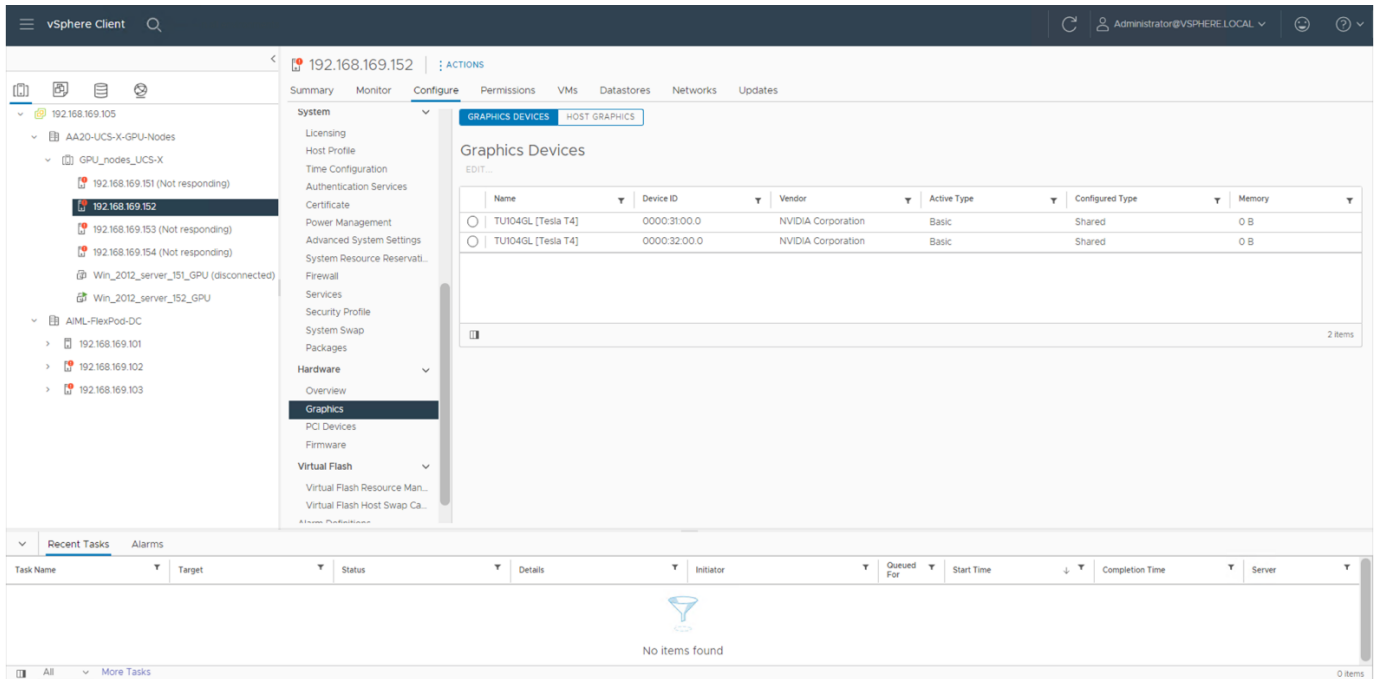


Figure 16.
Tesla T4 cards presented to the ESXi software by the hardware

2. At the host command-line interface (CLI), verify the GPUs (Figure 17).

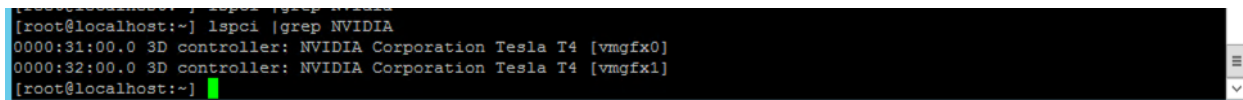


Figure 17.
Verifying the GPUs from the host CLI

NVIDIA drivers and licensing on the ESXi host

To present GPUs from the VMware ESXi host to virtual machines, you need the appropriate drivers. You also need licenses to operate the GPUs effectively. As stated earlier, hypervisor drivers are required on the ESXi host, and virtual machine (guest) drivers are required on the OS that is to be installed. Also, to virtualize the GPUs to be presented to virtual machines according to your application needs, you must have licenses from NVIDIA.

After you have obtained the licenses, follow the steps presented here to correctly operate vGPUs on virtual machines:

1. Although the NVIDIA driver for NVIDIA Manager is installed on the ESXi host itself, vCenter is used for all purposes for deploying and managing the vGPU. You must use vCenter Enterprise Plus Edition licenses for vGPU installation and usability.
2. For vGPU vMotion migration, enable `emotion.hot.migrate`.
3. In vCenter, change the ESXi host setting to Shared Direct. You will also need to reboot the host to allocate the vGPUs to the virtual machines from the ESXi host.
4. Install the ESXi host driver according to the profile to be used (this action also allows you to use NVIDIA vGPU Manager).

5. Create a virtual machine on the ESXi host on which vGPU Manager is installed.
6. Edit the virtual machine setting and add a PCIe vGPU device by selecting the appropriate vGPU profile.
7. Install the operating system and confirm that the hardware is detected on the virtual machine.
8. Install NVIDIA drivers according to the profile you selected. Verify that the installed OS driver matches the NVIDIA compatibility matrix. You can do this by using installation drivers from the same bundle from which you obtained the NVIDIA Manager driver.
9. Generate a token file from the NVIDIA licensing server and copy it to the required virtual machine directory.
10. In the virtual machine settings, allocate the correct vGPU resources profile according to virtual machine requirements.
11. In the virtual machine settings, allocate the correct CPU and memory according to your requirements.

See these documents for information about vGPU and AI Enterprise requirements:

- vGPU requirements: <https://docs.nvidia.com/grid/13.0/grid-vgpu-user-guide/index.html#prereqs-vgpu>
- AI Enterprise requirements: <https://docs.nvidia.com/ai-enterprise/latest/user-guide/index.html#prereqs-nvidia-ai-enterprise>

Deploying NVIDIA vGPU Manager on the ESXi host

NVIDIA has published the latest driver to date for VMware 7.0U2. To install the NVIDIA driver for the required VMware hypervisor, you need to download the latest driver from the NVIDIA website. The “platform version” is the version of the operating system installed, and the “product version” is the NVIDIA driver version for which you need to find the matching platform version; you then need to find the latest product version to get the latest driver.

Based on your license profile, select the product family and required drivers:

- For vPC and vDWS profiles, deploy NVIDIA vGPU Manager downloading from the NVIDIA website: <https://ui.licensing.nvidia.com/software>. In the product family list, filter the vGPU option and download the zip file that contains host drivers and guest drivers. Host drivers have a VIB file that you need to install on the ESXi host to make NVIDIA Manager commands available.
- For vCS, you need to deploy NVIDIA NVAIE Manager downloading from the NVIDIA website: <https://ui.licensing.nvidia.com/software>. In the product family list, filter the NVAIE option and download the zip file that contains host drivers and guest drivers. Host drivers have a VIB file that you need to install on the ESXi host to make NVIDIA Manager commands available.

See Figures 18 through 23.

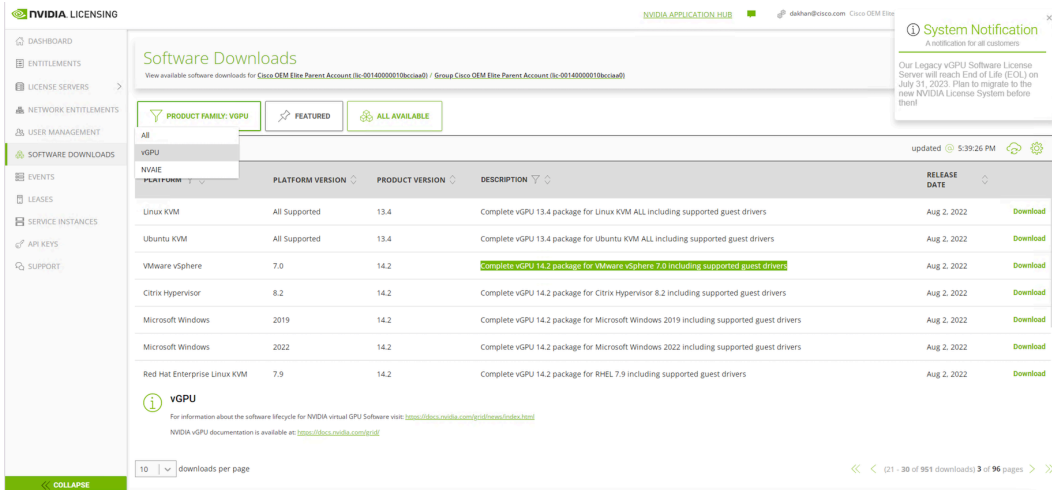


Figure 18.
vGPU product family vPC and vDWS profile highlighted in green

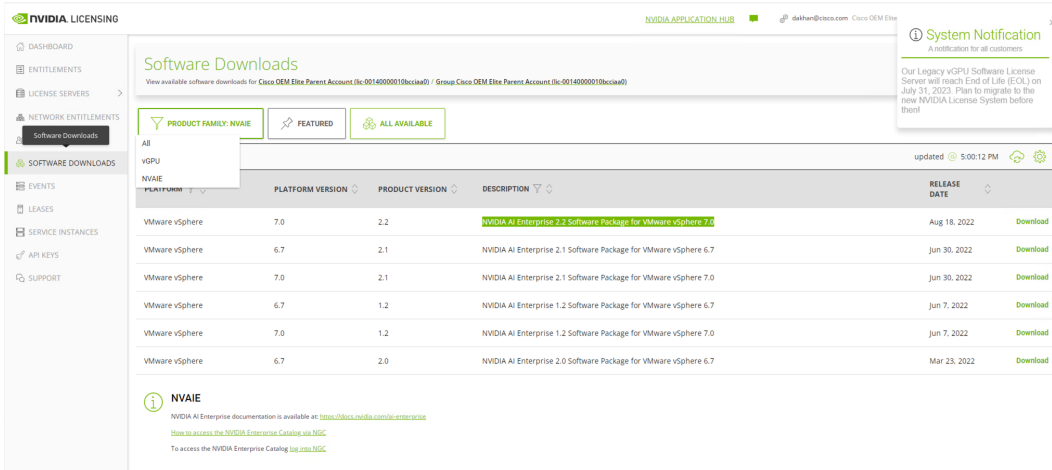


Figure 19.
NVAIE product family vCS profile highlighted in green

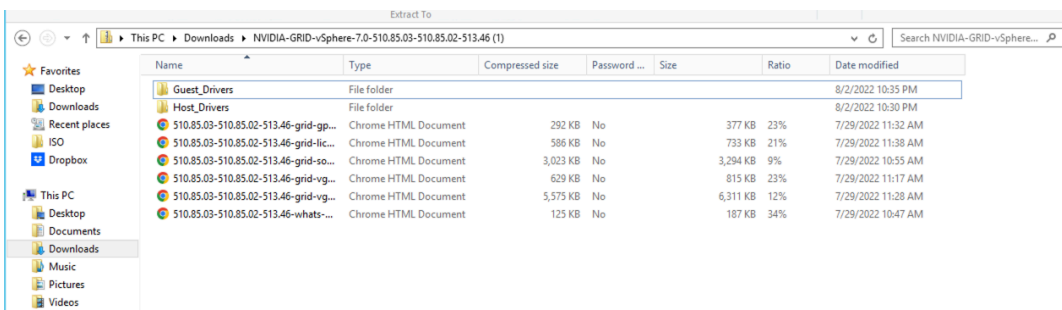


Figure 20.
vGPU product family: Downloaded zip file extraction presents host and guest drivers

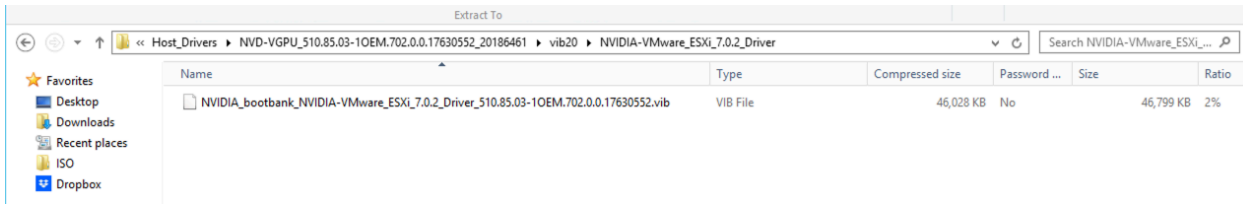


Figure 21.

Host (ESXi) vGPU product family driver provides a VIB file with this path:
 C:\Users\Administrator\AppData\Local\Temp\2\Temp1_NVIDIA-GRID-vSphere-7.0-510.85.03-510.85.02-513.46.zip\Host_Drivers\NVD-VGPU_510.85.03-10EM.702.0.0.17630552_20186461.zip\vib20\NVIDIA-VMware_ESXi_7.0.2_Driver

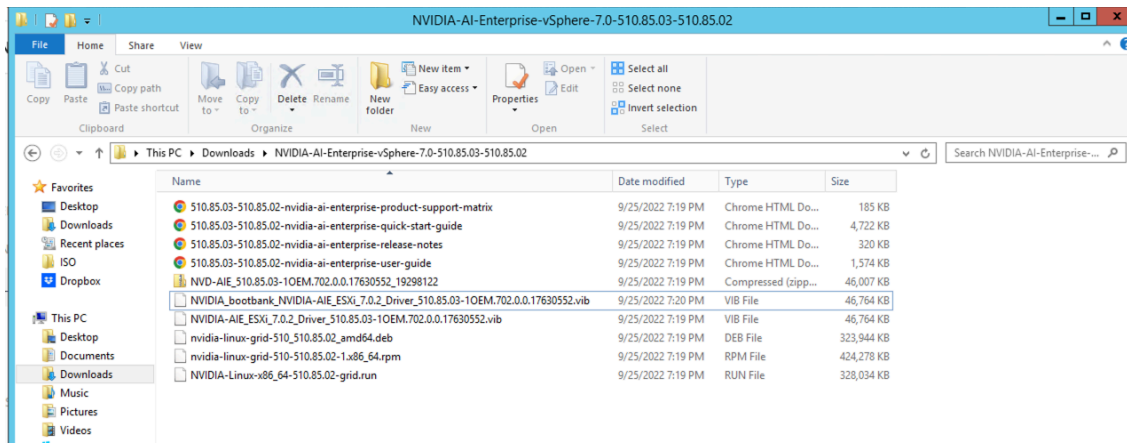


Figure 22.

NVAIE product family: Downloaded zip file extraction presents host and guest drivers

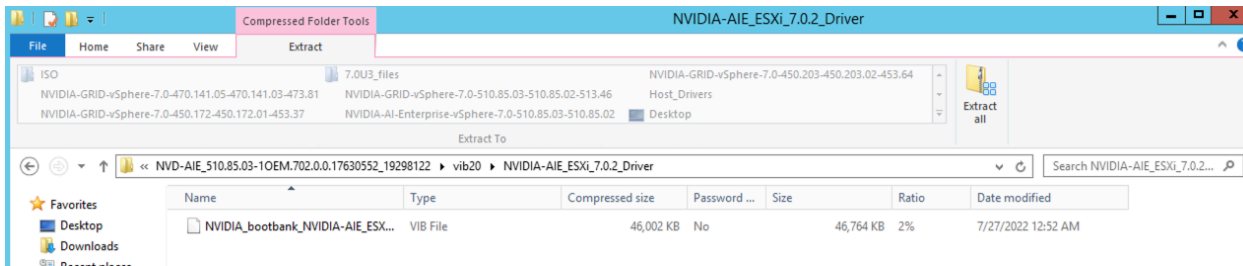


Figure 23.

Host (ESXi) vGPU product family driver provides a VIB file with this path: C:\Users\Administrator\Downloads\NVIDIA-AI-Enterprise-vSphere-7.0-510.85.03-510.85.02\NVD-AIE_510.85.03-10EM.702.0.0.17630552_19298122.zip\vib20\NVIDIA-AIE_ESXi_7.0.2_Driver

For this document, there are two ESXi hosts: 192.168.169.151 running the vGPU NVIDIA Manager for vPC and vDWS profiles, and 192.168.169.155 running the NVAIE NVIDIA Manager for the vCS profile. The snapshots that follow show the installation process for the host and virtual machines for each profile.

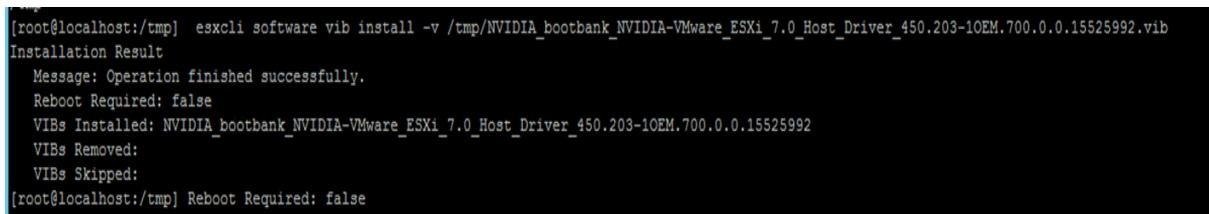
The installation link for the vGPU and AIE VIB is <https://docs.nvidia.com/grid/latest/grid-vgpu-user-guide/index.html#install-vgpu-vib-vmware-vsphere>.

As part of the process, the ESXi host needs to be in maintenance mode, and the VIB file needs to be uploaded to the ESXi host and installed on the host.

Here are the commands that need to be run:

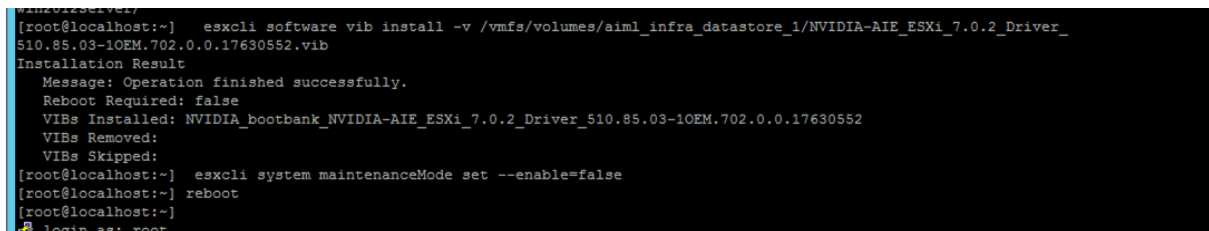
```
[root@localhost:~] esxcli system maintenanceMode set --enable=true
[root@localhost:~] esxcli software vib install -v
/vmfs/volumes/aiml_infra_datastore_1/NVIDIA_bootbank_NVIDIA-
VMware_ESXi_7.0.2_Driver_510.85.03-10EM.702.0.0.17630552.vib
[root@localhost:~] esxcli system maintenanceMode set --enable=false
[root@localhost:~] reboot
Verify installed vib via blow commands
[root@localhost:~] esxcli software vib list | grep -i nvidia
[root@localhost:~] nvidia-smi
```

On completion, you should see the message shown in Figure 24 or Figure 25. It says that a reboot is not required. However, as the NVIDIA documentation says, a reboot is required.



```
[root@localhost:/tmp] esxcli software vib install -v /tmp/NVIDIA_bootbank_NVIDIA-VMware_ESXi_7.0_Host_Driver_450.203-10EM.700.0.0.15525992.vib
Installation Result
  Message: Operation finished successfully.
  Reboot Required: false
  VIBs Installed: NVIDIA_bootbank_NVIDIA-VMware_ESXi_7.0_Host_Driver_450.203-10EM.700.0.0.15525992
  VIBs Removed:
  VIBs Skipped:
[root@localhost:/tmp] Reboot Required: false
```

Figure 24.
vGPU VIB installation host with successful installation



```
[root@localhost:~] esxcli software vib install -v /vmfs/volumes/aiml_infra_datastore_1/NVIDIA-AIE_ESXi_7.0.2_Driver_
510.85.03-10EM.702.0.0.17630552.vib
Installation Result
  Message: Operation finished successfully.
  Reboot Required: false
  VIBs Installed: NVIDIA_bootbank_NVIDIA-AIE_ESXi_7.0.2_Driver_510.85.03-10EM.702.0.0.17630552
  VIBs Removed:
  VIBs Skipped:
[root@localhost:~] esxcli system maintenanceMode set --enable=false
[root@localhost:~] reboot
[root@localhost:~]
login as: root
```

Figure 25.
AIE VIB installation host with successful installation

After the reboot is complete, you should be able to run NVIDIA commands from the CLI after you take the host out of maintenance mode. Use the command shown in Figure 26 or Figure 27 to verify the driver for ESXi.

```

192.168.169.151 - PuTTY
[root@localhost:~]
[root@localhost:~] nvidia-smi
Mon Sep 26 22:37:46 2022
+-----+
| NVIDIA-SMI 510.85.03      Driver Version: 510.85.03      CUDA Version: N/A      |
+-----+
| GPU   Name           Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf    Pwr:Usage/Cap|      Memory-Usage | GPU-Util  Compute M. |
|                                           MIG M.         |
+-----+-----+
|   0   Tesla T4              On         | 00000000:31:00.0 Off  |      0          Default |
| N/A   36C    P8     16W /  70W | 1970MiB / 15360MiB |      0%          N/A   |
+-----+-----+
|   1   Tesla T4              On         | 00000000:32:00.0 Off  |      0          Default |
| N/A   37C    P8     16W /  70W |  82MiB / 15360MiB |      0%          N/A   |
+-----+-----+

Processes:
+-----+
| GPU   GI    CI          PID   Type   Process name                      GPU Memory |
|      ID    ID                                   |            Usage                    |
+-----+-----+
|   0   N/A  N/A     2140597   C+G   ..._8.6_GPU_8.3_T4-4q (temp)     1888MiB |
+-----+
[root@localhost:~] █

```

Figure 26.
Host with vGPU VIB running the `nvidia-smi` command

```

192.168.169.155 - PuTTY
[root@localhost:~]
[root@localhost:~]
[root@localhost:~] nvidia-smi
Mon Sep 26 22:28:52 2022
+-----+
| NVIDIA-SMI 510.85.03      Driver Version: 510.85.03      CUDA Version: N/A      |
+-----+
| GPU   Name           Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf    Pwr:Usage/Cap|      Memory-Usage | GPU-Util  Compute M. |
|                                           MIG M.         |
+-----+-----+
|   0   Tesla T4              On         | 00000000:31:00.0 Off  |      0          Default |
| N/A   36C    P8     15W /  70W | 3858MiB / 15360MiB |      0%          N/A   |
+-----+-----+
|   1   Tesla T4              On         | 00000000:32:00.0 Off  |      0          Default |
| N/A   37C    P8     16W /  70W |  82MiB / 15360MiB |      0%          N/A   |
+-----+-----+

Processes:
+-----+
| GPU   GI    CI          PID   Type   Process name                      GPU Memory |
|      ID    ID                                   |            Usage                    |
+-----+-----+
|   0   N/A  N/A     2108975   C+G   RH_8.6_server_155_.T4_8c_84     3776MiB |
+-----+
[root@localhost:~] █

```

Figure 27.
Host with NVAIE VIB running the `nvidia-smi` command

After the host reboots and is out of maintenance mode, verify from vCenter the three steps listed in Figure 28.

1 Ensure you have **Enterprise Plus** licenses; you **NEED** vCenter

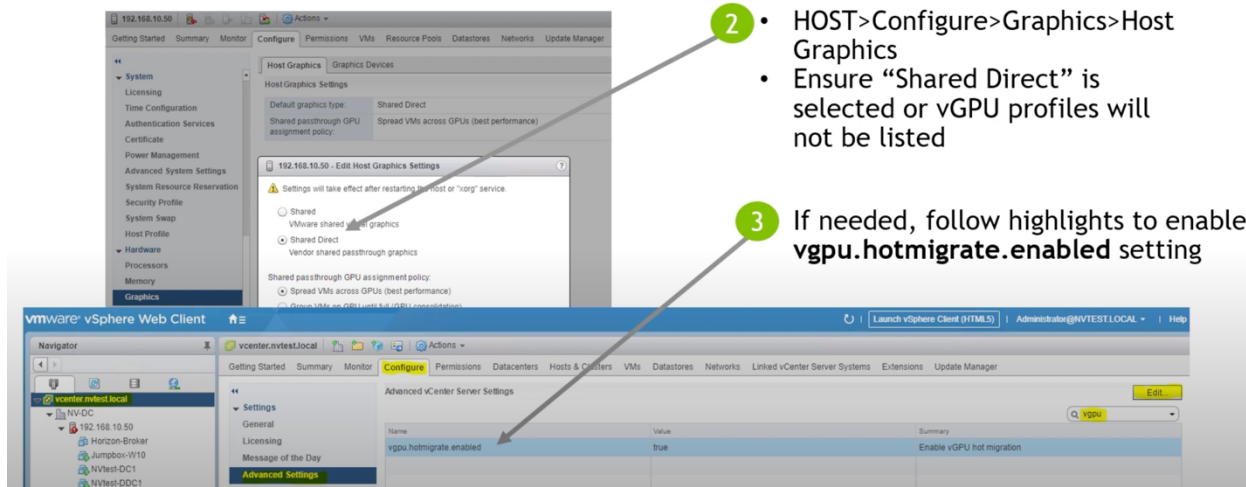


Figure 28.
Verifying the driver installation from vCenter

As stated in Figure 29, use only the drivers from the licensing portal.

GRID Drivers are downloaded from the Licensing Portal, not the driver download pages



Figure 29.
Downloading licenses

After the vGPU or NVAIE Manager is installed, you can edit the virtual machine setting for the PCIe slot. Adding the correct profile will complete the host requirements. Figures 30 and 31 show the settings for the vGPU and NVAIE Managers. Note that the t4-c profiles do not exist on the NVAIE vGPU.

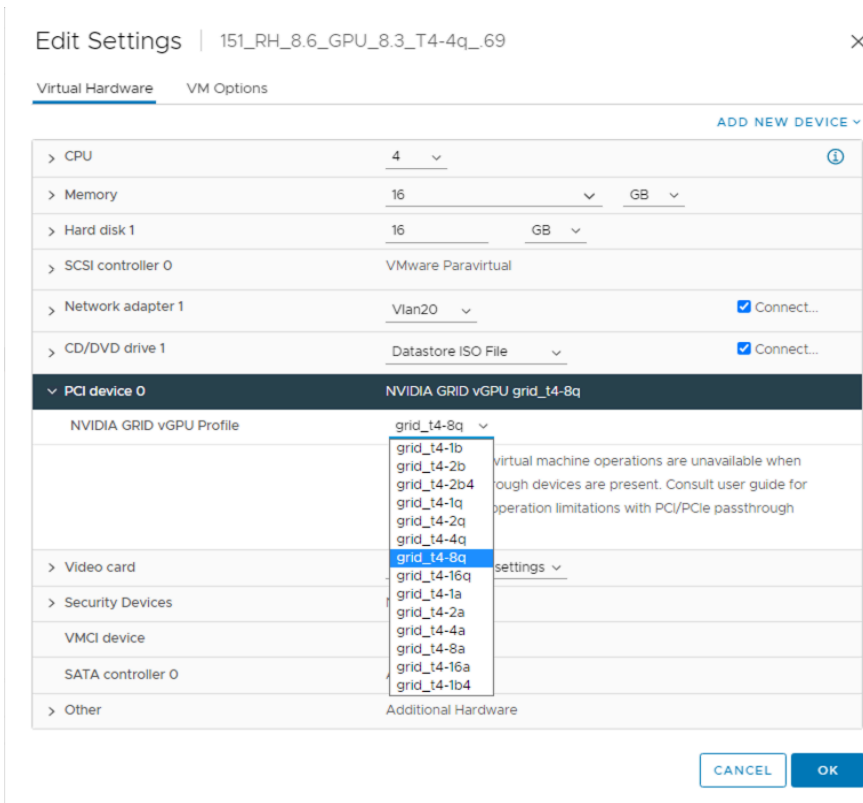


Figure 30. Host with NVAIE vGPU virtual machine setting; note that there are no t4-c profiles

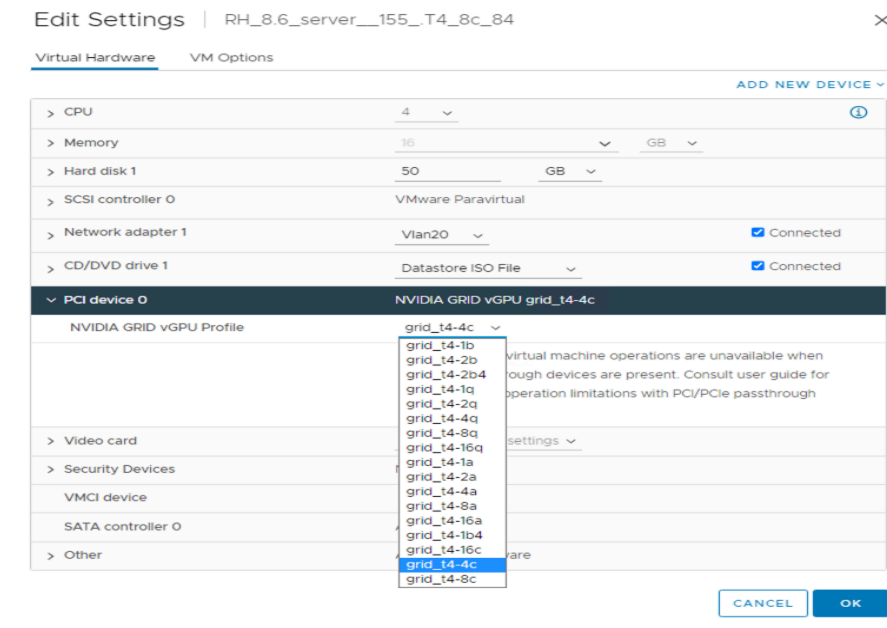


Figure 31. Host with NVAIE VIB virtual machine setting with t4-c profiles

Figure 32 summarizes profile naming conventions.

UNDERSTANDING PROFILES

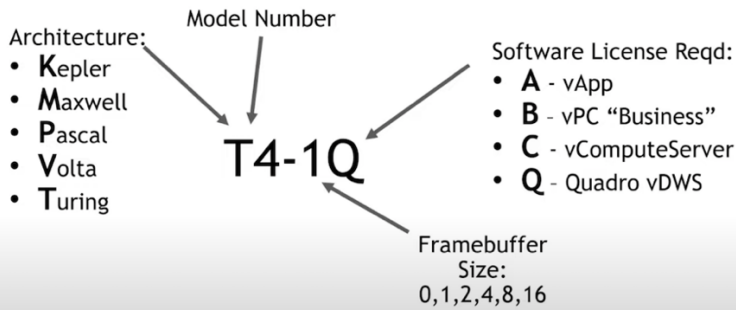


Figure 32.
Profile naming conventions

You may need to turn off error-correcting code (ECC) memory if you see the error message shown in Figure 33.

[If virtual machines won't boot, you may need to turn off ECC for the GPU](#)

Status: ❌ Could not initialize plugin 'usr/lib64/vmware/plugin/libnvidia-vgx.so' for vGPU 'grid_v100-16a'. ← If you see this ...

Initiator: NVTEST.LOCAL\Administrator
Target: NVTest-XenApp1
Server: vcenter.nvtest.local

Error stack:

- ↳ Failed to start the virtual machine.
- ↳ Module 'DevicePowerOn' power on failed.
- ↳ Could not initialize plugin 'usr/lib64/vmware/plugin/libnvidia-vgx.so' for vGPU 'grid_v100-16a'.

1 # nvidia-smi -q

```
=====NVSMI LOG=====
Timestamp                : Tue Dec 19 18:36:45 2017
Driver Version           : 384.99
Attached GPUs            : 1
GPU 0000:02:00.0
[...]
Ecc Mode
  Current                : Enabled
  Pending                : Enabled
[...]
```

2 # nvidia-smi -e 0 **Disable ECC for all cards**

```
[root@dc2xtmvmesx01-vsant1:~] nvidia-smi -i 00000000:02:00.0 -e 0
Disabled ECC support for GPU 00000000:02:00.0.
All done.
Reboot required.
```

Disable ECC for card id 00000000:02:00.0

3 Reboot

<https://docs.nvidia.com/#disabling-enabling-ecc-memory>

Figure 33.
Error message requiring ECC to be disabled

The host requirements for deployment are now complete. Next, you need to install the OS, drivers, and licenses on the virtual machines.

vGPU installation requirements and verification on Microsoft Windows Server 2022 as a virtual machine on VMware

With NVIDIA vGPU Manager installed on the hypervisor (in the solution here, VMware) and the vGPU profile assigned in the virtual machine PCIe settings, the next step is to install the OS and its required guest NVIDIA drivers. Windows Server is supported only with vPC and vDWS profiles. For additional details, see the following document: <https://docs.nvidia.com/grid/13.0/grid-vgpu-user-guide/index.html#guest-os-support-grid-vgpu>.

After the driver is installed, you need to apply the token file generated in the NVIDIA licensing portal to the OS so that you can use the vGPUs assigned.

Here is the sequence of steps you need to follow:

1. Install the virtual machine operating system (Windows Server 2022).
2. Install the NVIDIA driver from the guest drivers folder for Windows 2022.
3. Install VMware tools.
4. Assign IP addresses and configure a remote desktop to get essential remote access, because after you switch the display to vGPU, a black screen will appear on the VMware virtual machine console.
5. Verify that the Network Time Protocol (NTP) is synchronized between the virtual machine and the NVIDIA license server.
6. Verify that TCP ports 443, 8080, 8081, and 8082 are open to reach out to the NVIDIA license server.
7. Install the vGPU license.
8. Check the logs for verification.

Installing NVIDIA driver in the OS: Windows Server 2022 virtual machine on ESXi host

Figures 34 and 35 show two different profiles: one for each vPC and vDWS profile before and after driver installation.

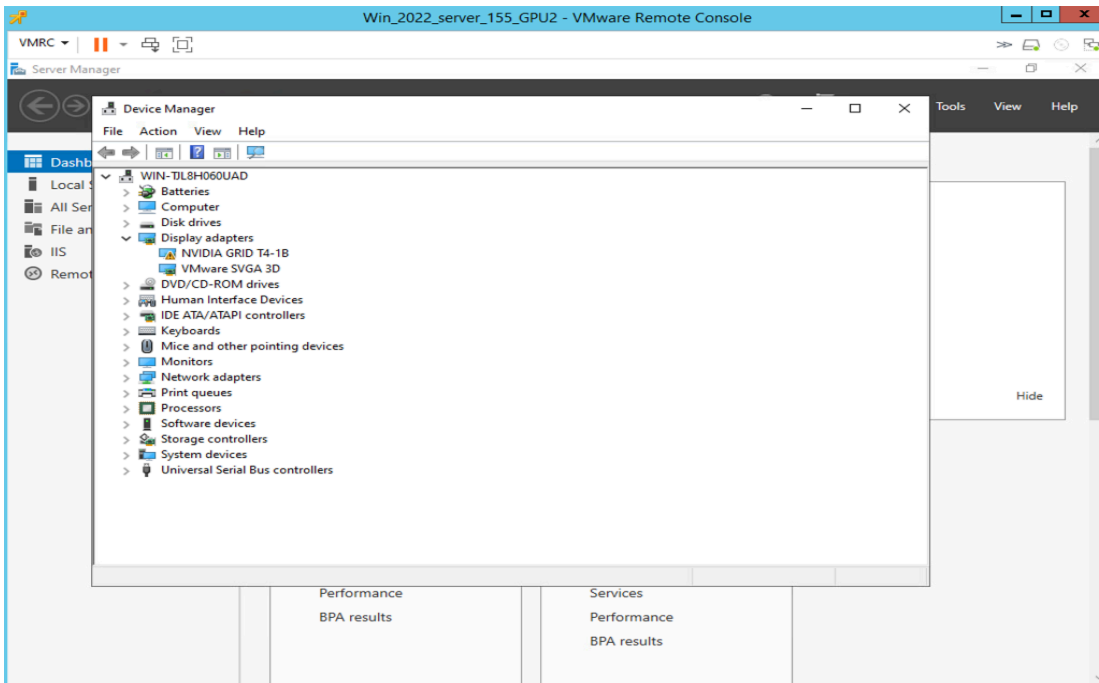


Figure 34. Device Manager settings in Windows Server 2022 before driver installation for T4-1B profile (vPC)

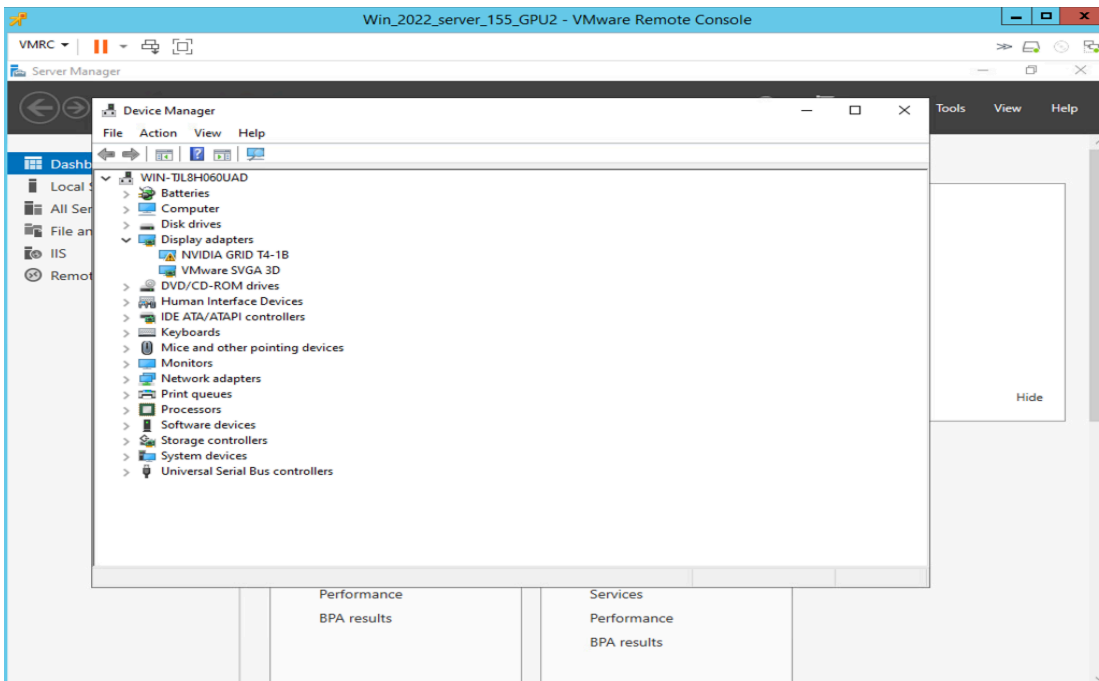
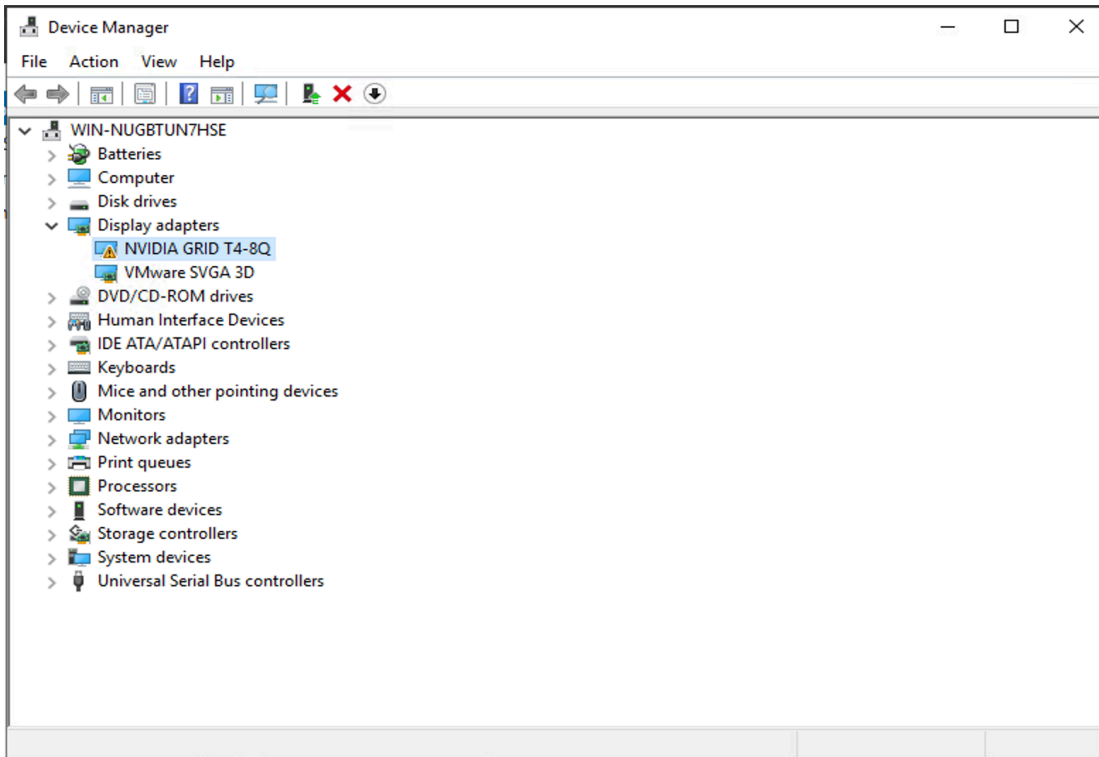


Figure 35. Device Manager settings in Windows Server 2022 before driver installation for T4-8Q profile (vDWS)



1. Install the required driver from the guest drivers folder (Figures 36 and 37).

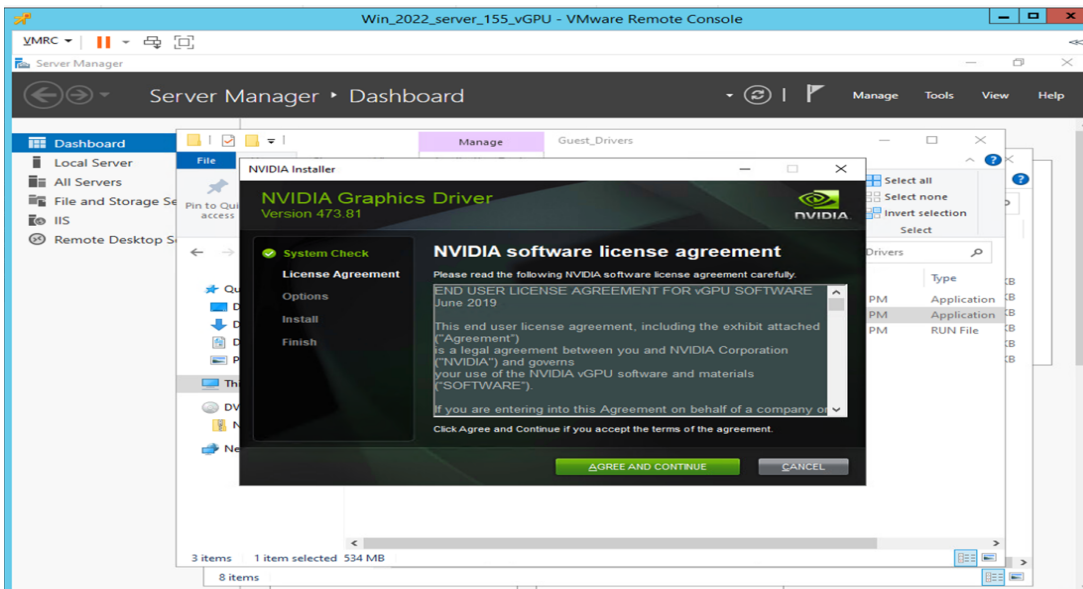


Figure 36.
NVIDIA license agreement

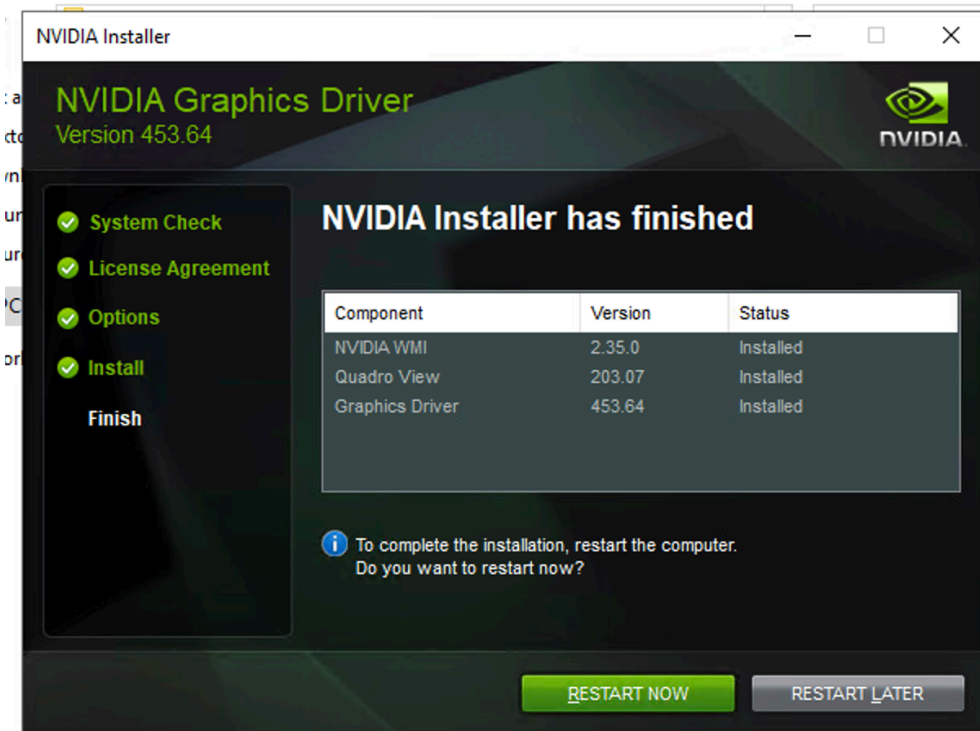


Figure 37.
Installing the NVIDIA driver

2. Before you restart the virtual machine, check the Device Manager to see the displayed graphic card options (Figure 38).

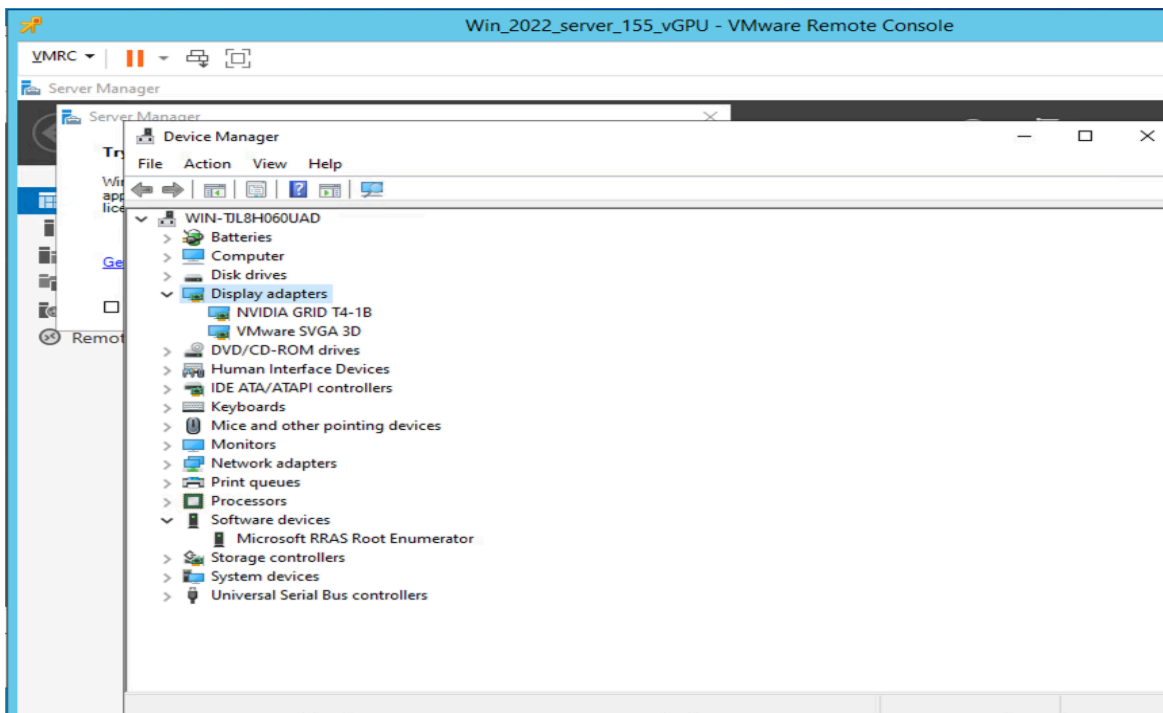


Figure 38.
After the driver is installed, you can see NVIDIA in the display adapter options

3. Similarly, for the vDWS profile, you can check to see the driver installed (Figure 39).

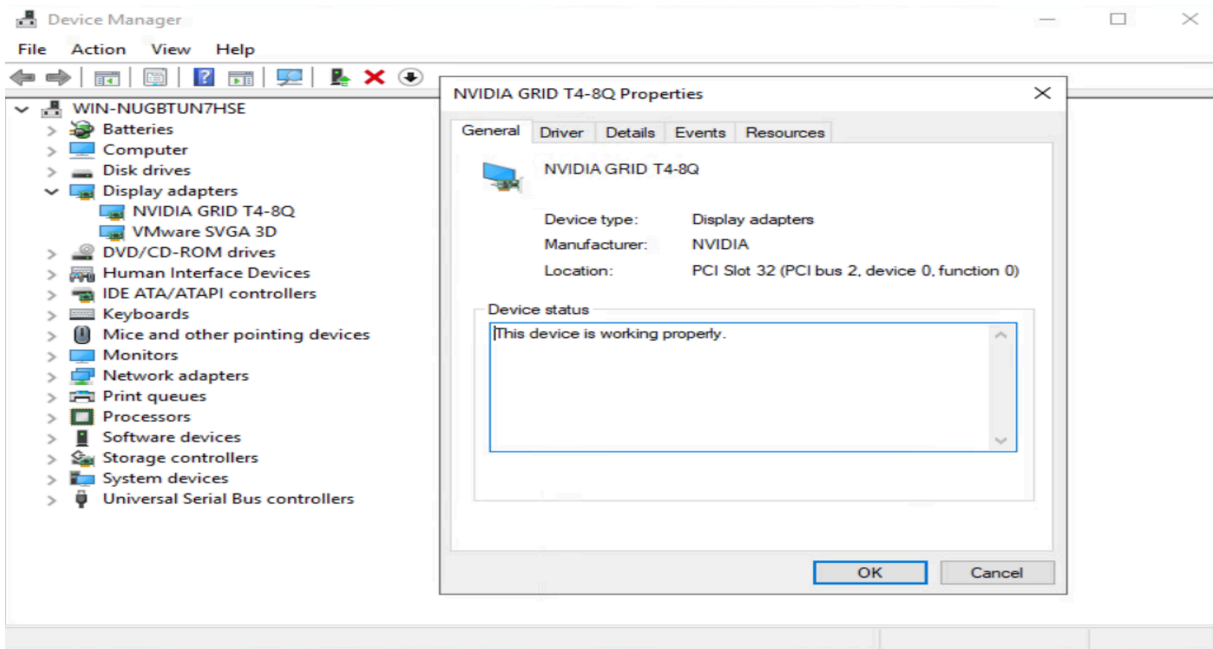


Figure 39.
NVIDIA driver installed for vDWS profile

At this point, restarting the virtual machine will apply the vGPU as the display driver. However, you need to use a remote desktop client from another system because the VMware console for the virtual machine displays a black screen.

Installing NVIDIA license in Windows Server OS

To use an NVIDIA vGPU software licensed product, each Microsoft Windows Server must be able to obtain a license from the NVIDIA license system directly. The Windows Server can be a virtual machine that is configured with an NVIDIA vGPU, a virtual machine that is configured for GPU passthrough, or a physical host to which a physical GPU is assigned in a bare-metal deployment.

Follow these steps to install a license on Windows Server:

1. Download the token file from the NVIDIA license server to the required directory in the operating system.
2. Apply the Microsoft Windows registry to create FeatureType DWord.
3. Restart the `NvDisplayContainer` service.
4. Verify the license installation using the events log and the `Nvidia-smi -q` command.

Note: To install licenses, synchronize the virtual machine time with the DLS server. Connect the hypervisor console of the DLS server, log in as `dls_system`, and run the `timedatectl status` command to confirm the time on the DLS server.

For the process for licensing the client OS, see <https://docs.nvidia.com/grid/14.0/grid-licensing-user-guide/index.html#configuring-nls-licensed-client-on-windows>.

Before you install the license, you will see notifications reporting that a license is not present (Figure 40).

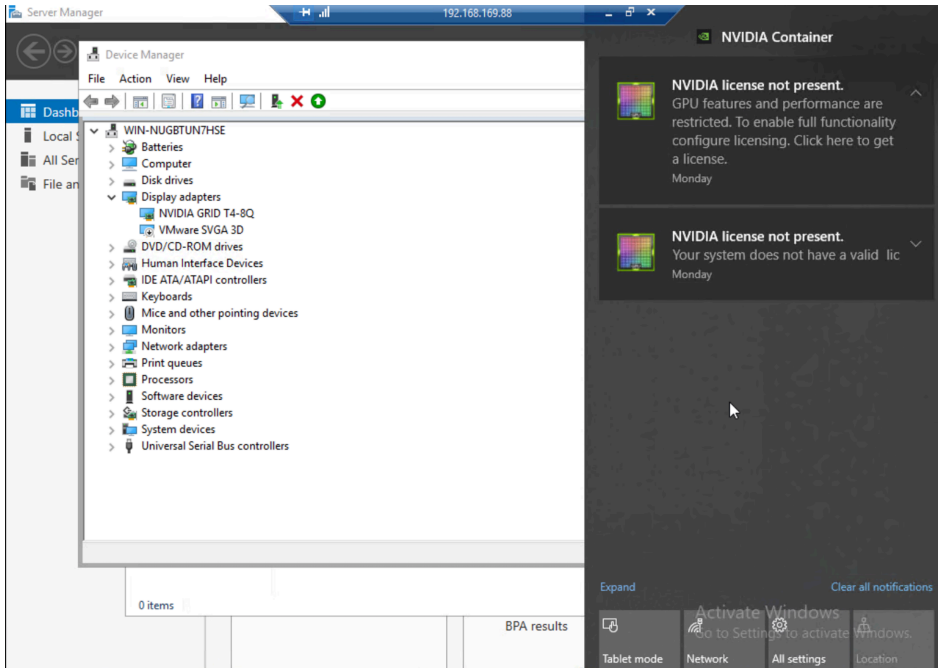


Figure 40.
Message reporting that no license is present

The steps for installing licensing on the Windows Server host are presented here.

1. To install the licensing token file, log in to the NVIDIA licensing portal at <https://ui.licensing.nvidia.com/login>.
2. Download the token file from the DLS server created following the licensing process (Figure 41).

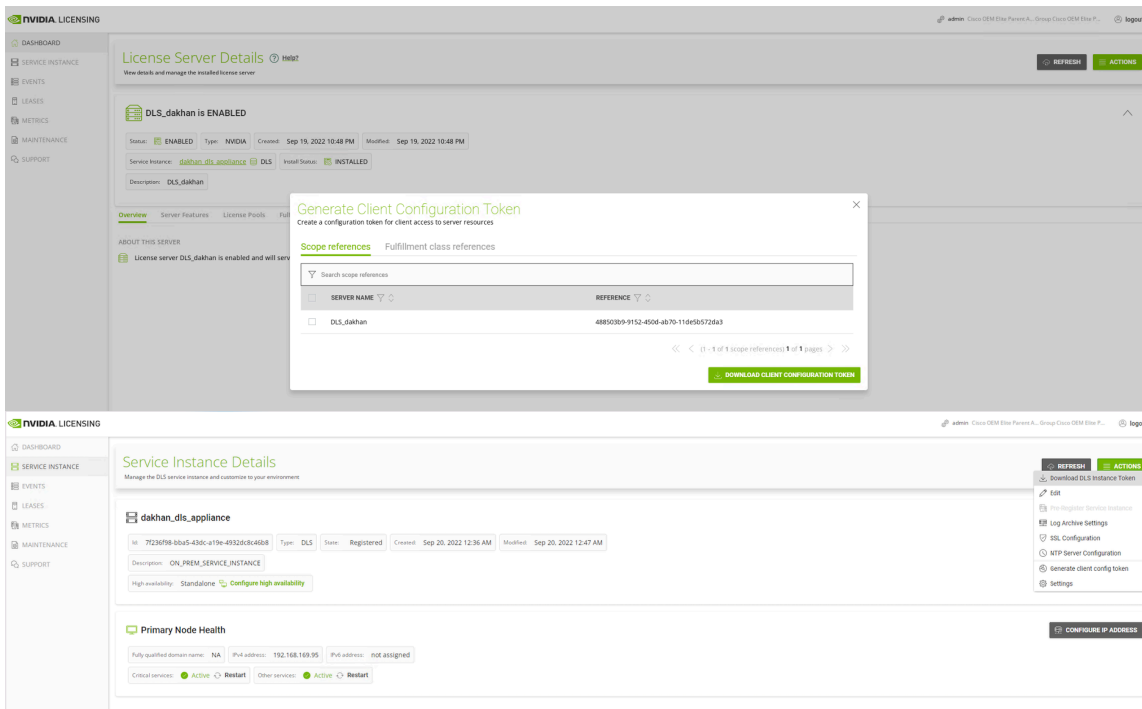


Figure 41.
Download the token file

- Copy the client configuration token to the %SystemDrive%\Program Files\NVIDIA Corporation\vGPU Licensing\ClientConfigToken folder (Figure 42).

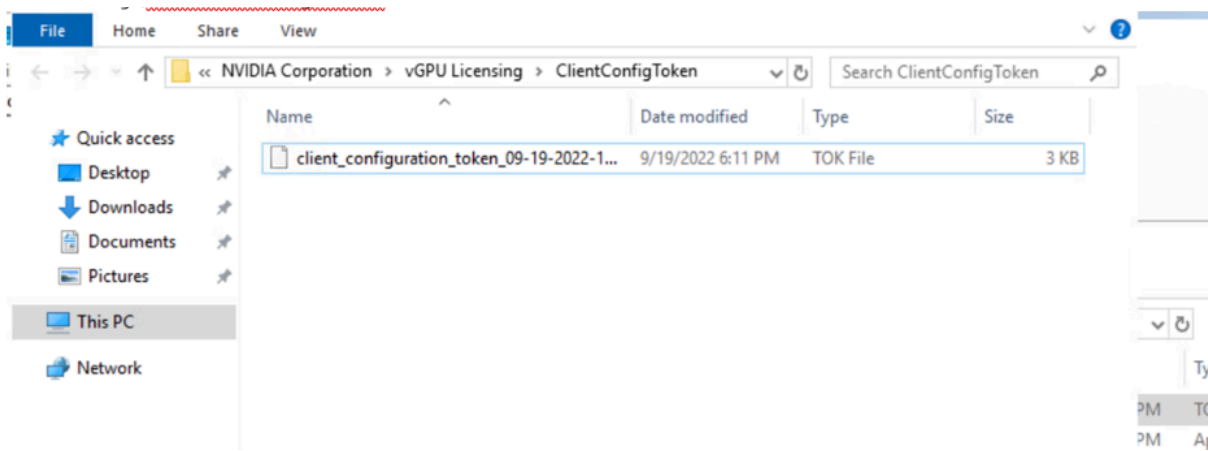


Figure 42.
Copy the client configuration token

- Add the FeatureType DWord (**REG_DWORD**) registry value to the Windows registry key HKEY_LOCAL_MACHINE\SOFTWARE\NVIDIA Corporation\Global\GridLicensing (Figure 43).

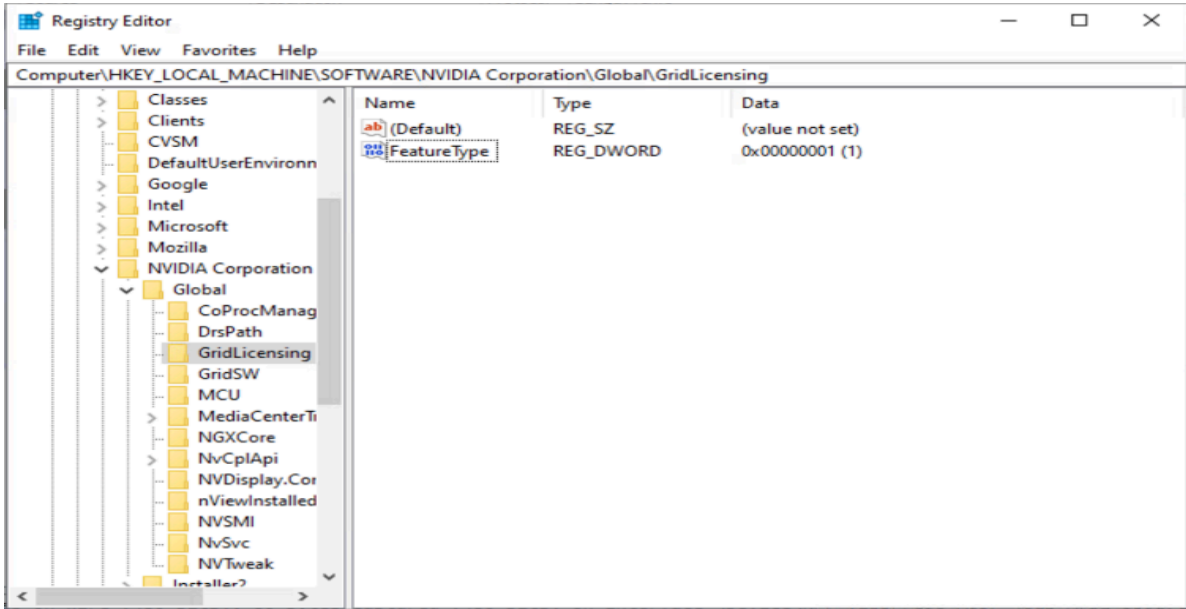


Figure 43.
Adding the REG_DWORD FeatureType

Table 7 shows the value of the FeatureType DWord.

Table 7. FeatureType DWord value

GPU type	Setting
NVIDIA vGPU	Do not change the value of this registry key. NVIDIA vGPU software automatically selects the correct type of license based on the vGPU type.
Physical GPU	The feature type of a GPU in passthrough mode or a bare-metal deployment is as follows: 0: NVIDIA vApps 2: NVIDIA RTX vWS

5. Restart the NvDisplayContainer service (Figure 44).

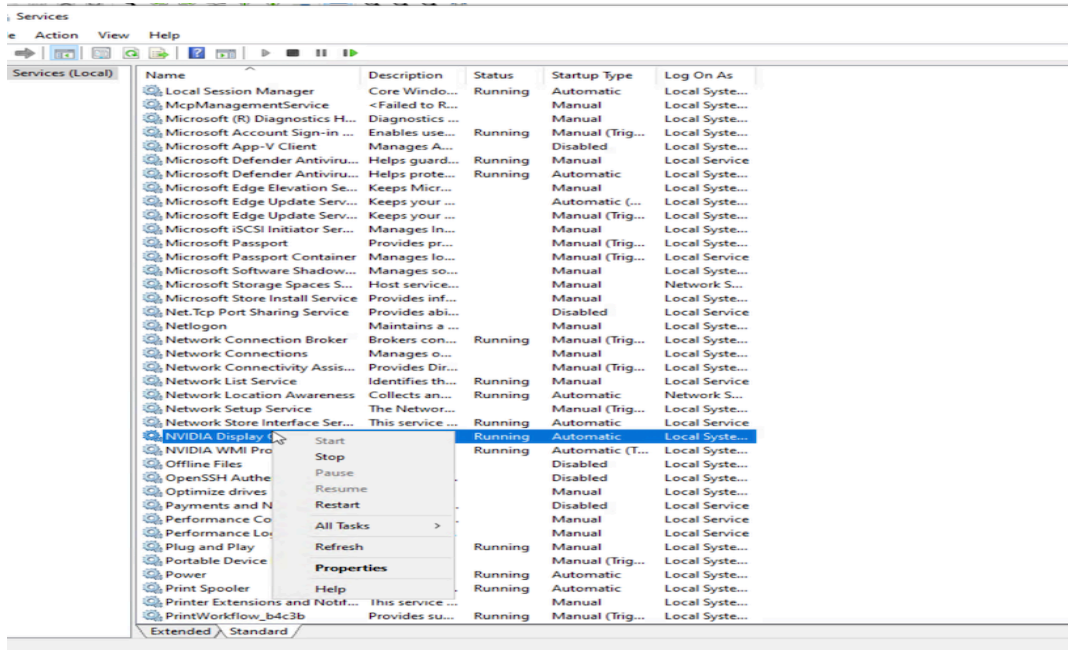


Figure 44. Restarting the NvDisplayContainer service

6. Verify the setup.

In Windows Server, licensing events are logged in the plain-text file %SystemDrive%\Users\Public\Documents\NvidiaLogging\Log.NVDisplay.Container.exe.log. View this file (Figures 45, 46, and 47).

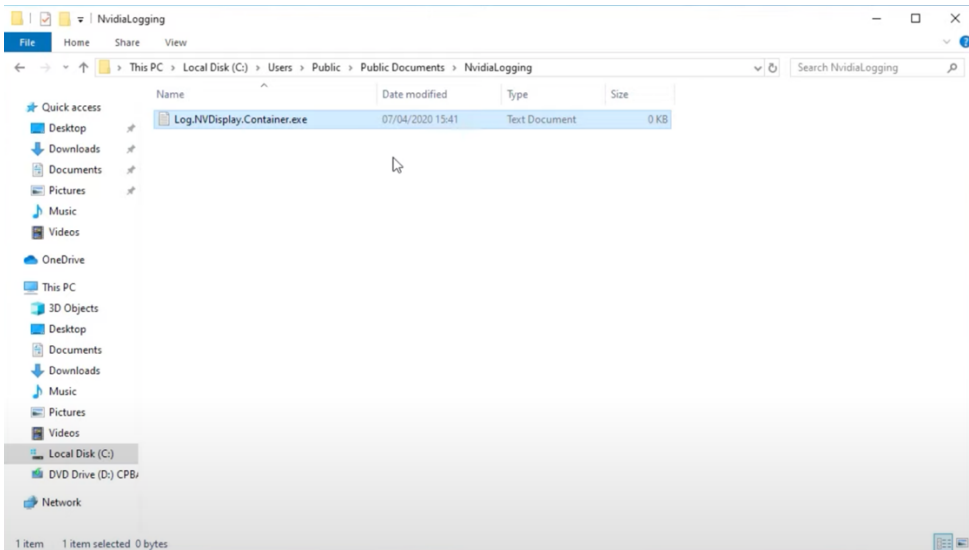


Figure 45.
Checking the plain-text log file in Windows Server

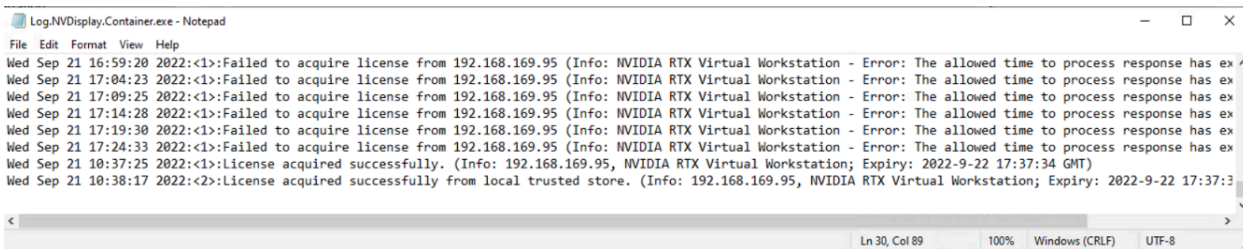


Figure 46.
Logged events in Windows Server

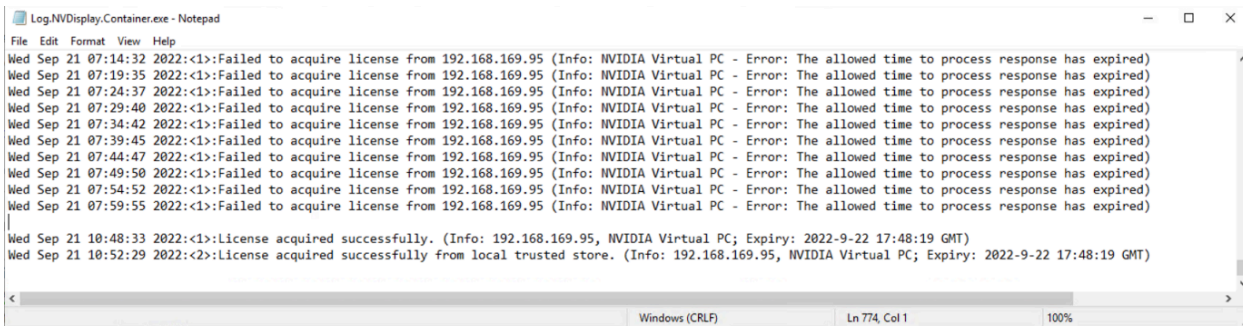


Figure 47.
Logged events in Windows Server

To verify the license status of a licensed client, run **nvidia-smi** with the **-q** or **--query** option. If the product is licensed, the expiration date is shown in the license status (Figures 48 and 49).

```

C:\Program Files\NVIDIA Corporation\NVSMI>nvidia-smi.exe -q
=====NVSMI LOG=====
Timestamp                : Wed Sep 21 14:46:38 2022
Driver Version           : 513.46
CUDA Version             : 11.6

Attached GPUs            : 1
GPU 00000000:02:00.0
  Product Name           : GRID T4-8Q
  Product Brand          : NVIDIA RTX Virtual Workstation
  Product Architecture    : Turing
  Display Mode           : Enabled
  Display Active         : Enabled
  Persistence Mode       : N/A
  MIG Mode
    Current              : N/A
    Pending              : N/A
  Accounting Mode        : Disabled
  Accounting Mode Buffer Size : 4000
  Driver Model
    Current              : WDDM
    Pending              : WDDM
  Serial Number          : N/A
  GPU UUID               : GPU-074f81c8-1dd6-11b2-bdc7-0e151a1d5c4b
  Minor Number           : N/A
  VBIOS Version          : 00.00.00.00.00
  MultiGPU Board        : No
  Board ID               : 0x200
  GPU Part Number        : N/A
  Module ID              : N/A
  Inforom Version
    Image Version        : N/A
    OEM Object           : N/A
    ECC Object           : N/A
    Power Management Object : N/A
  GPU Operation Mode
    Current              : N/A
    Pending              : N/A
  GSP Firmware Version  : N/A
  GPU Virtualization Mode : N/A
  Virtualization Mode    : VGPU
  Host VGPU Mode         : N/A
  vGPU Software Licensed Product
    Product Name         : NVIDIA RTX Virtual Workstation
    License Status       : Licensed (Expiry: 2022-9-22 20:37:34 GMT)
  IBMNPU
    Relaxed Ordering Mode : N/A
  PCI
    Bus                  : 0x02
    Device                : 0x00
    Domain                : 0x0000

```

Figure 48.
NVIDIA vPC displaying the installed driver

```

C:\Program Files\NVIDIA Corporation\NVSMI>nvidia-smi.exe -q
=====NVSMI LOG=====
Timestamp                : Wed Sep 21 14:43:24 2022
Driver Version           : 513.46
CUDA Version             : 11.6

Attached GPUs            : 1
GPU 00000000:02:00.0
  Product Name           : GRID T4-2B
  Product Brand          : NVIDIA Virtual PC
  Product Architecture    : Turing
  Display Mode           : Enabled
  Display Active         : Enabled
  Persistence Mode       : N/A
  MIG Mode
    Current              : N/A
    Pending              : N/A
  Accounting Mode        : Disabled
  Accounting Mode Buffer Size : 4000
  Driver Model
    Current              : WDDM
    Pending              : WDDM
  Serial Number          : N/A
  GPU UUID               : GPU-08ef128f-1dd6-11b2-8470-8af8942075b8
  Minor Number           : N/A
  VBIOS Version          : 00.00.00.00.00
  MultiGPU Board        : No
  Board ID               : 0x200
  GPU Part Number        : N/A
  Module ID              : N/A
  Inforom Version
    Image Version        : N/A
    OEM Object           : N/A
    ECC Object           : N/A
    Power Management Object : N/A
  GPU Operation Mode
    Current              : N/A
    Pending              : N/A
  GSP Firmware Version  : N/A
  GPU Virtualization Mode : N/A
  Virtualization Mode    : VGPU
  Host VGPU Mode         : N/A
  vGPU Software Licensed Product
    Product Name         : NVIDIA Virtual PC
    License Status       : Licensed (Expiry: 2022-9-22 20:48:19 GMT)
  IBMNPU
    Relaxed Ordering Mode : N/A
  PCI
    Bus                  : 0x02
    Device                : 0x00
    Domain                : 0x0000

```

Figure 49.
NVIDIA vCS displaying the installed driver

Check the NVIDIA Control Panel to verify the driver and other details (Figure 50).

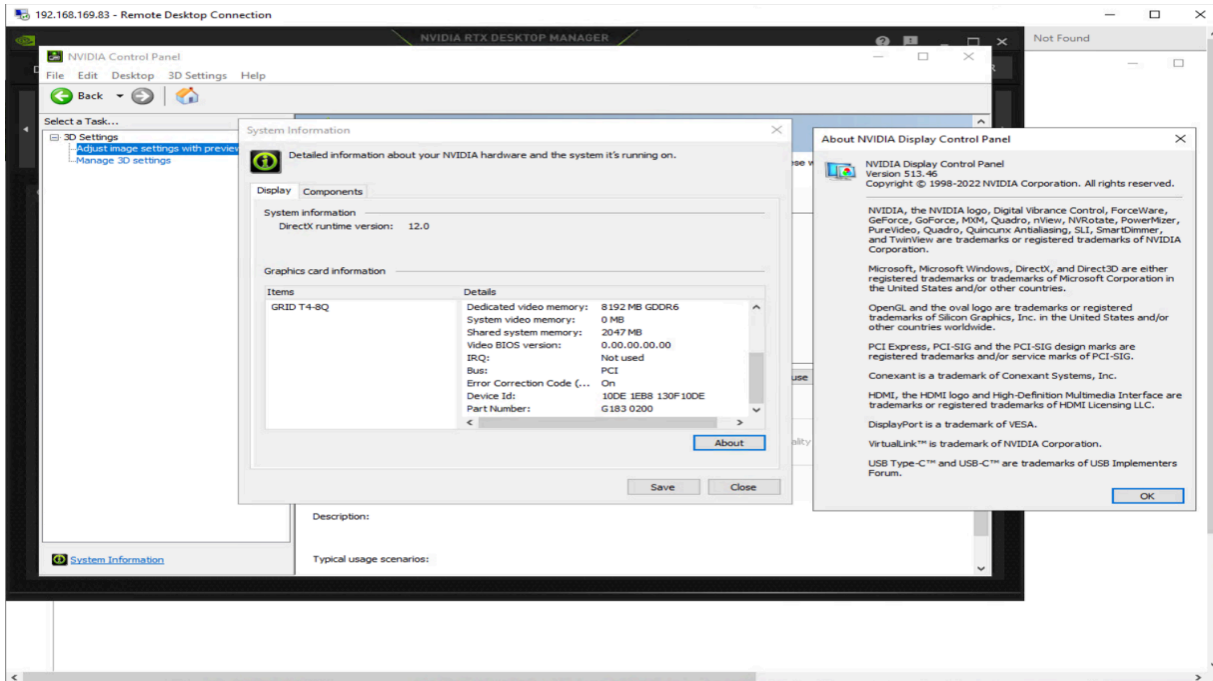


Figure 50.
Verifying the driver in the NVIDIA Control Panel

Verify leased licenses for virtual machines by checking the NVIDIA server (Figure 51).

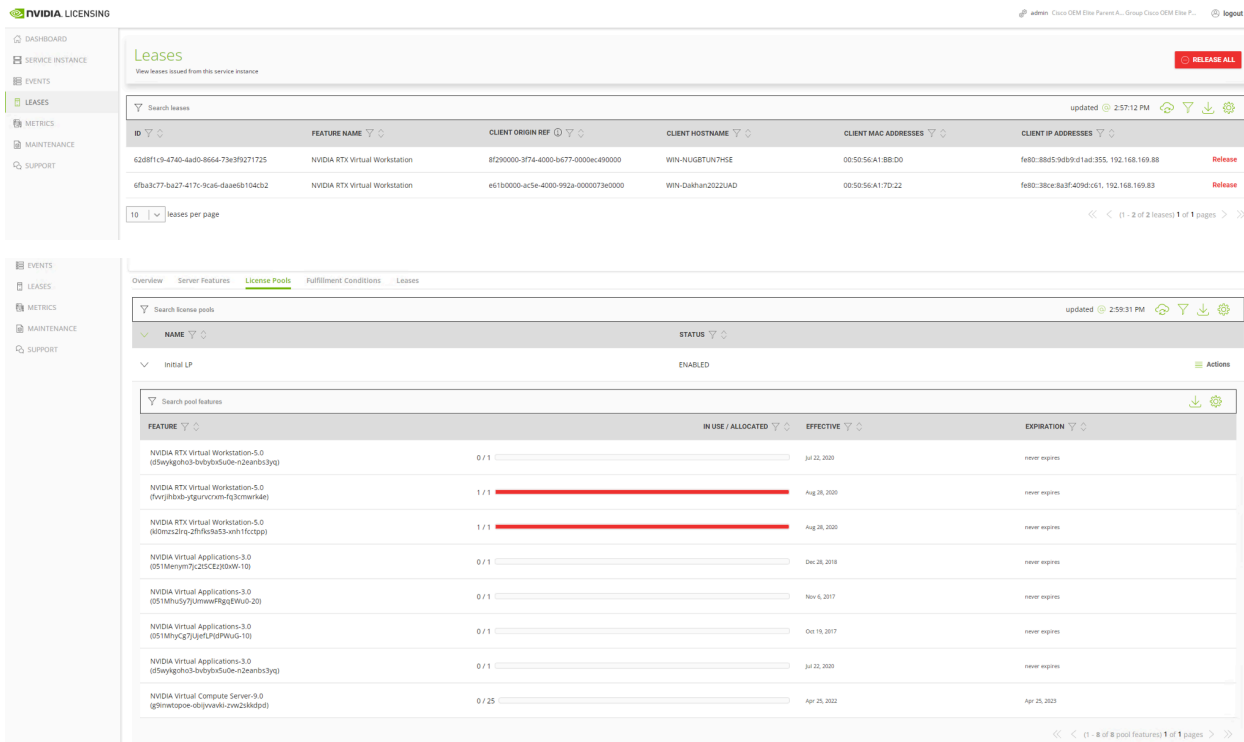


Figure 51.
Verifying leased licenses on the NVIDIA server

vGPU installation requirements and verification on an RHEL 8.6 server as a VMware ESXi host virtual machine

As with the Windows virtual machine installation process, after you have an NVIDIA vGPU profile assigned in the virtual machine settings of the VMware hypervisor, the next step is to install the OS. In this case, you install the RHEL OS and guest NVIDIA drivers for RHEL 8.6. You use the same token file generated for NVIDIA licensing to apply licenses; however, the process is slightly different. Linux-based operating systems are supported on all profiles and so consequently are supported on vGPU as well as NVAIE NVIDIA Managers. For computing-intensive virtual servers for AI, deep learning, or HPC workloads, the NVAIE profile and the vCS Manager and license is deployed, and for the vPC and vDWS profiles, the vGPU Manager and license is deployed.

The example presented here shows two virtual machines: one on each host running a vGPU and the NVAIE profile.

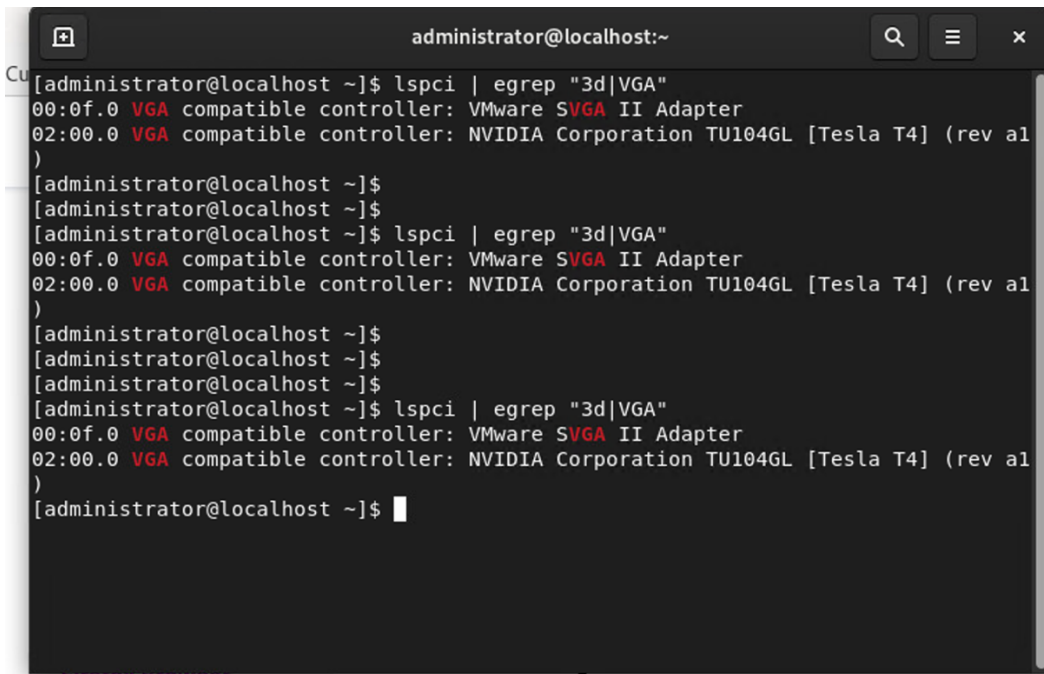
The following steps are required for RHEL 8.6 driver and license installation:

1. Install the virtual machine operating system (RHEL 8.6).
2. Install VMware tools.
3. Install the RHEL virtual machine base configuration.
4. Install RDP access.
5. Install the NVIDIA driver guest driver for RHEL 8.6.
6. Assign IP addressing to configure the remote desktop to get essential remote access, because after you switch the display to vGPU, a black screen will appear.
7. Install the vGPU licensing.
8. Verify that TCP ports 443, 8080, 8081, and 8082 are open to reach out to the NVIDIA license server.
9. Check the logs for verification.

Verify the installed hardware

After the virtual machine has been assigned vGPUs and the RHEL virtual machine is installed and present, verify that the hardware is installed.

1. To verify the vGPU assignment and RHEL virtual machine installation, use the Terminal application on the RHEL OS (Figure 52).

A terminal window titled 'administrator@localhost:~' with search, menu, and close icons. The terminal shows a series of commands and their outputs. The command 'lspci | egrep "3d|VGA"' is run three times, each time displaying two lines of hardware information: '00:0f.0 VGA compatible controller: VMware SVGA II Adapter' and '02:00.0 VGA compatible controller: NVIDIA Corporation TU104GL [Tesla T4] (rev a1)'.

```
administrator@localhost:~
[administrator@localhost ~]$ lspci | egrep "3d|VGA"
00:0f.0 VGA compatible controller: VMware SVGA II Adapter
02:00.0 VGA compatible controller: NVIDIA Corporation TU104GL [Tesla T4] (rev a1)
)
[administrator@localhost ~]$
[administrator@localhost ~]$
[administrator@localhost ~]$ lspci | egrep "3d|VGA"
00:0f.0 VGA compatible controller: VMware SVGA II Adapter
02:00.0 VGA compatible controller: NVIDIA Corporation TU104GL [Tesla T4] (rev a1)
)
[administrator@localhost ~]$
[administrator@localhost ~]$
[administrator@localhost ~]$
[administrator@localhost ~]$ lspci | egrep "3d|VGA"
00:0f.0 VGA compatible controller: VMware SVGA II Adapter
02:00.0 VGA compatible controller: NVIDIA Corporation TU104GL [Tesla T4] (rev a1)
)
[administrator@localhost ~]$
```

Figure 52.

Use the Terminal application to verify hardware installation

2. After verifying that the hardware is present in the OS, you are ready to install the driver and related base configurations. First, update the packages. Log in as root and run the following commands:

```
[root@localhost /]#yum -y update
[root@localhost /]# yum groupinstall "Development Tools"
[root@localhost /]# yum -y install kernel-devel
[root@localhost /]# yum install dkms
```

3. After the built-in driver is disabled for RHEL, you will see a black screen when you log in to the VMware Remote Console (VMRC). To gain access to the GUI when NVIDIA GPU is enabled, enable RDP on the Red Hat system by entering the following commands:

```
[root@localhost /]# yum install -y https://dl.fedoraproject.org/pub/epel/epel-release-latest-8.noarch.rpm
[root@localhost /]# yum install -y tigervnc-server xrdp
[root@localhost /]# systemctl start xrdp
[root@localhost /]# netstat -antup | grep xrdp
[root@localhost /]# systemctl enable xrdp
[root@localhost /]# firewall-cmd --permanent --add-port=3389/tcp
[root@localhost /]# firewall-cmd --reload
```

4. Blacklist the NVIDIA nouveau driver by entering the following commands:

```
[root@localhost /]# sudo bash -c "echo blacklist nouveau > /etc/modprobe.d/blacklist-nvidia-nouveau.conf"
[root@localhost /]# sudo bash -c "echo options nouveau modeset=0 >> /etc/modprobe.d/blacklist-nvidia-nouveau.conf"
[root@localhost /]# mv /boot/initramfs-`uname -r`.img /boot/initramfs-`uname -r`.img.nouveau
```

```
[root@localhost /]# dracut
[root@localhost /]# reboot
```

5. Confirm that nouveau is disabled and stop the X server by entering the following commands:

```
# lsmod | grep -i nouveau
```

If nouveau is not disabled, run the blacklist procedure again:

```
# init 3 or sudo service gdm3 stop to stop the X server
```

6. Install the NVIDIA GRID driver for Linux. NVIDIA drivers are available in three formats for use with Linux distributions:

- Runfile installers
- Package managers
- Containerized drivers

All methods are mentioned at <https://docs.nvidia.com/grid/latest/grid-vgpu-user-guide/index.html - installing-vgpu-drivers-linux>.

The example in this document uses the .run file method. The .run file is different for a vCS profile than for the vPC and vDWS profiles.

```
[root@localhost Desktop]# pwd
/root/Desktop
[root@localhost Desktop]# ls
NVIDIA-Linux-x86_64-510.85.02-grid.run.
[root@localhost Desktop]# sh NVIDIA-Linux-x86_64-510.85.02-grid.run
```

Figures 53, 54, and 55 show the process of installing drives using the .run package.

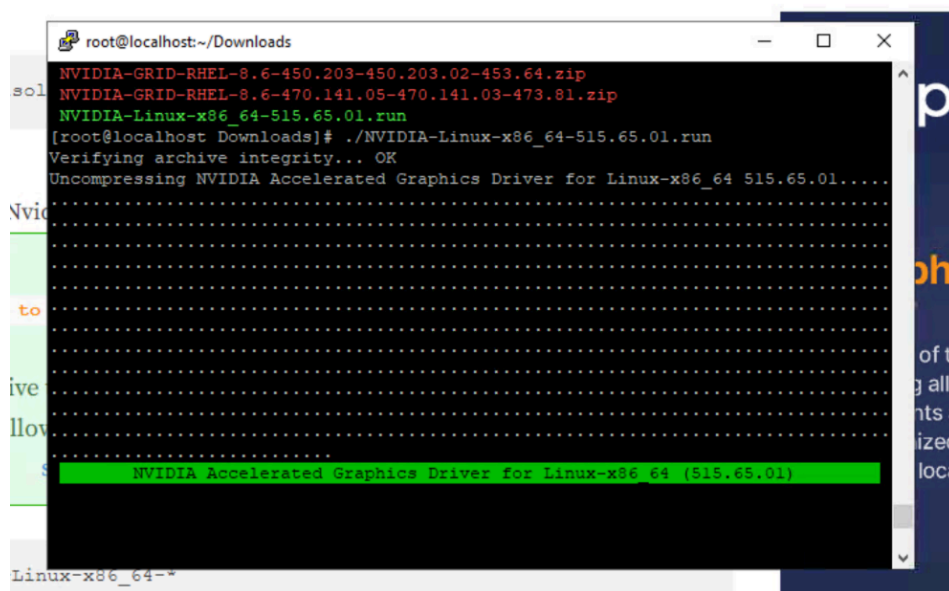


Figure 53.
Installing drives using the .run package

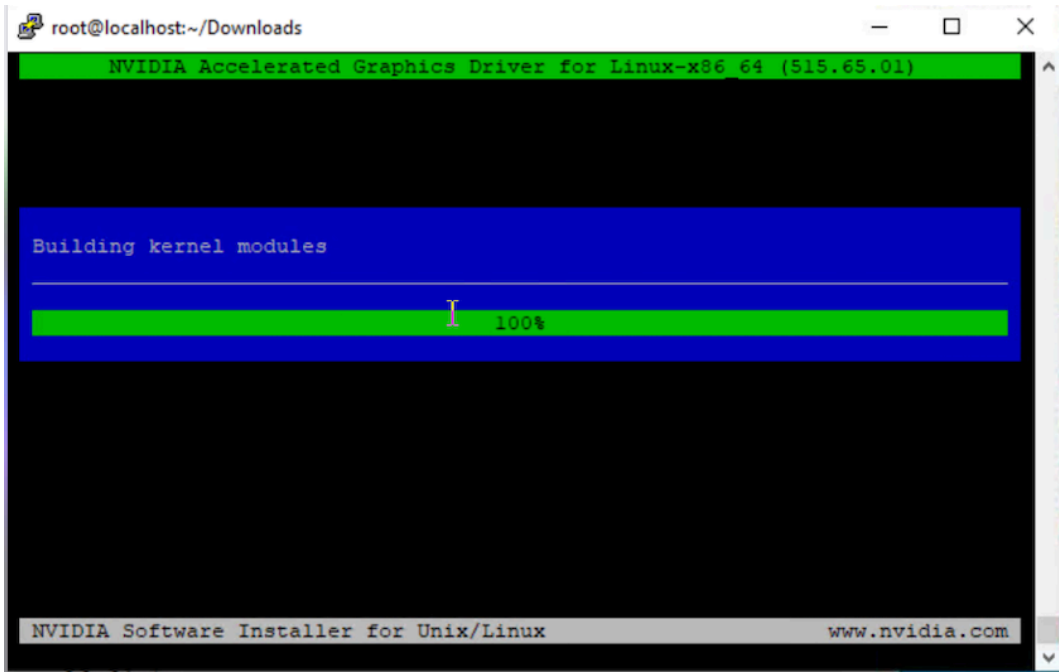


Figure 54.
Building kernel modules

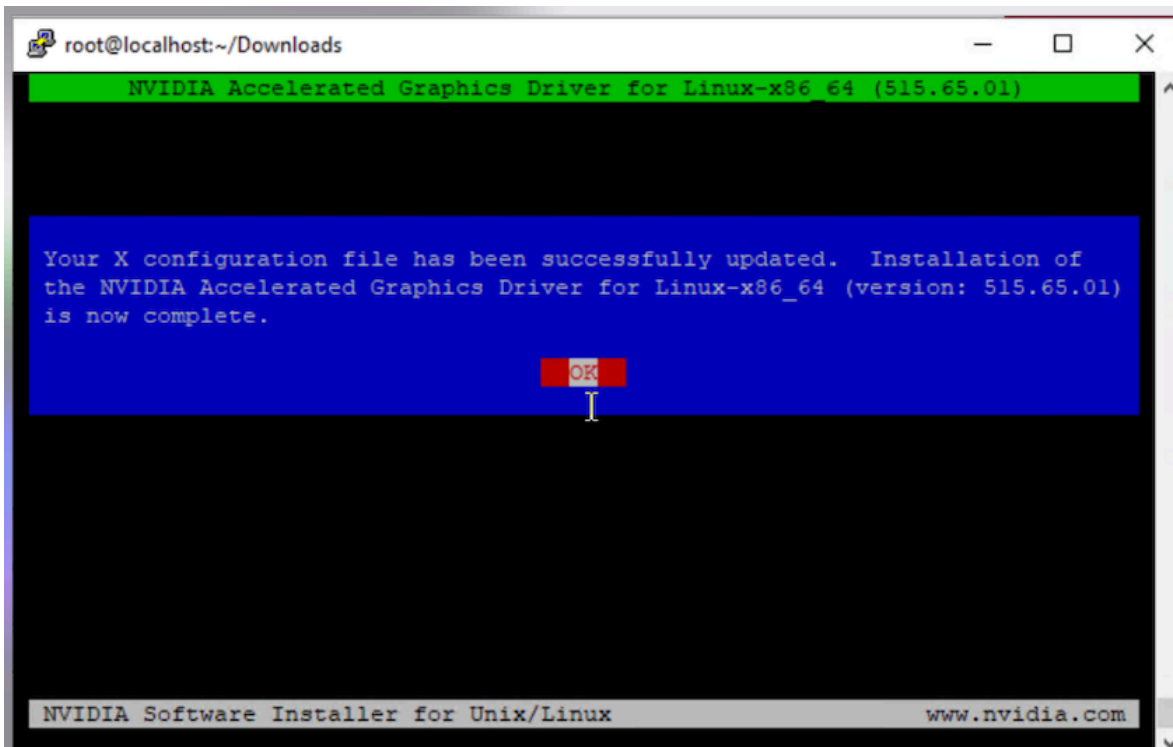
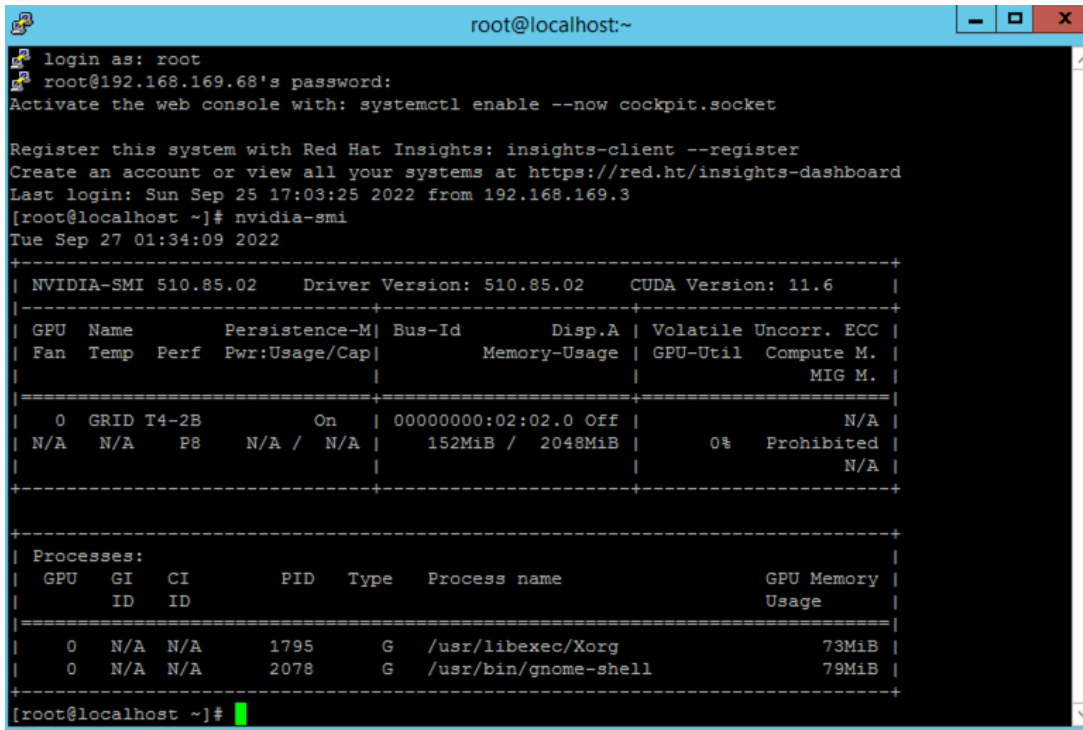


Figure 55.
X configuration file successfully updated

7. After the installation process succeeds, run `nvidia-smi` commands to verify that they are operable (Figures 56 and 57).

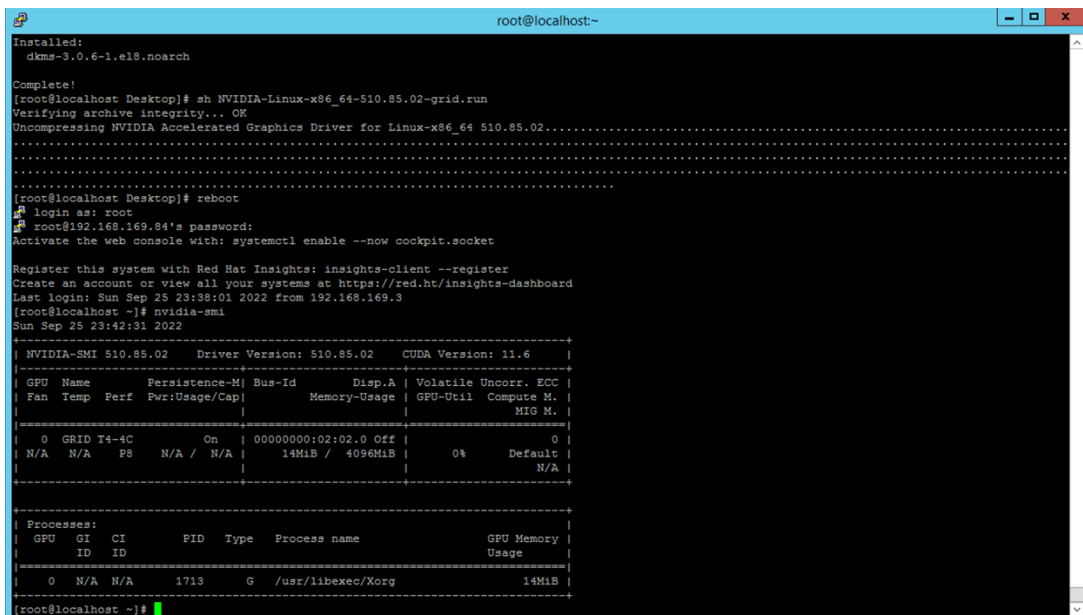


```
root@localhost:~  
login as: root  
root@192.168.169.68's password:  
Activate the web console with: systemctl enable --now cockpit.socket  
  
Register this system with Red Hat Insights: insights-client --register  
Create an account or view all your systems at https://red.ht/insights-dashboard  
Last login: Sun Sep 25 17:03:25 2022 from 192.168.169.3  
[root@localhost ~]# nvidia-smi  
Tue Sep 27 01:34:09 2022
```

NVIDIA-SMI 510.85.02 Driver Version: 510.85.02 CUDA Version: 11.6										
GPU	Name	Persistence-M	Bus-Id	Disp.A	Volatile Uncorr. ECC	Memory-Usage		GPU-Util	Compute M.	MIG M.
Fan	Temp	Perf	Pwr:Usage/Cap							
0	GRID T4-2B	On	00000000:02:02.0	Off	N/A	152MiB / 2048MiB	0%	Prohibited	N/A	N/A

Processes:							
GPU	GI	CI	PID	Type	Process name	GPU Memory Usage	
ID	ID	ID					
0	N/A	N/A	1795	G	/usr/libexec/Xorg	73MiB	
0	N/A	N/A	2078	G	/usr/bin/gnome-shell	79MiB	

Figure 56. Verification on vPC and vDWS profiles with the host running vGPU Manager



```
root@localhost:~  
Installed:  
dkms-3.0.6-1.el8.noarch  
  
Complete!  
[root@localhost Desktop]# sh NVIDIA-Linux-x86_64-510.85.02-grid.run  
Verifying archive integrity... OK  
Uncompressing NVIDIA Accelerated Graphics Driver for Linux-x86_64 510.85.02.....  
.....  
[root@localhost Desktop]# reboot  
login as: root  
root@192.168.169.84's password:  
Activate the web console with: systemctl enable --now cockpit.socket  
  
Register this system with Red Hat Insights: insights-client --register  
Create an account or view all your systems at https://red.ht/insights-dashboard  
Last login: Sun Sep 25 23:38:01 2022 from 192.168.169.3  
[root@localhost ~]# nvidia-smi  
Sun Sep 25 23:42:31 2022
```

NVIDIA-SMI 510.85.02 Driver Version: 510.85.02 CUDA Version: 11.6										
GPU	Name	Persistence-M	Bus-Id	Disp.A	Volatile Uncorr. ECC	Memory-Usage		GPU-Util	Compute M.	MIG M.
Fan	Temp	Perf	Pwr:Usage/Cap							
0	GRID T4-4C	On	00000000:02:02.0	Off	0	14MiB / 4096MiB	0%	Default	N/A	N/A

Processes:							
GPU	GI	CI	PID	Type	Process name	GPU Memory Usage	
ID	ID	ID					
0	N/A	N/A	1713	G	/usr/libexec/Xorg	14MiB	

Figure 57. Verification on a vCS profile with the host running NVAIE Manager

8. As another means of verification, check the video driver running after the driver is installed and the system rebooted (Figure 58).

```
root@localhost:~  
-----+  
[root@localhost ~]# sudo lshw -c video  
*-display  
  description: VGA compatible controller  
  product: SVGA II Adapter  
  vendor: VMware  
  physical id: f  
  bus info: pci@0000:00:0f.0  
  logical name: /dev/fb0  
  version: 00  
  width: 32 bits  
  clock: 33MHz  
  capabilities: vga_controller bus_master cap_list rom fb  
  configuration: depth=32 driver=vmwgfx latency=64 resolution=1672,1254  
  resources: irq:16 ioport:2040(size=16) memory:e8000000-efffffff memory:fb800000-fbffffff  
f memory:c0000-dffff  
*-display  
  description: VGA compatible controller  
  product: TU104GL [Tesla T4]  
  vendor: NVIDIA Corporation  
  physical id: 2  
  bus info: pci@0000:02:02.0  
  version: a1  
  width: 64 bits  
  clock: 66MHz  
  capabilities: msi vga_controller bus_master cap_list  
  configuration: driver=nvidia latency=0  
  resources: irq:65 memory:fa000000-faffffff memory:d0000000-dffffff memory:f8000000-f9ffff  
ffff  
[root@localhost ~]#
```

Figure 58.
Checking the video driver

NVIDIA licensing on RHEL 8.6 virtual machine

After you have downloaded the token file, follow the licensing installation steps described at the following link: <https://docs.nvidia.com/license-system/latest/nvidia-license-system-user-guide/index.html#configuring-nls-licensed-client-on-linux>.

1. Download the token file and copy it to the RHEL directory:

```
/etc/nvidia/ClientConfigToken
```

2. Change permissions for the client token file:

```
[root@localhost:~] cd /etc/nvidia/ClientConfigToken
```

```
[root@localhost:~] chmod 744 client_configuration_token_09-25-2022-16-45-15.tok
```

```
[root@localhost:~] service nvidia-gridd restart
```

3. Use **nvidia-smi -q** to confirm that the license is installed correctly (Figures 59 and 60).

```

root@localhost/etc/nvidia/ClientConfigToken
[root@localhost ClientConfigToken]#
[root@localhost ClientConfigToken]#
[root@localhost ClientConfigToken]# nvidia-smi -q
=====NVSMI LOG=====
Timestamp                : Sun Sep 25 15:52:21 2022
Driver Version           : 510.85.02
CUDA Version             : 11.6
Attached GPUs            : 1
GPU 00000000:02:02.0
  Product Name           : GRID T4-8Q
  Product Brand          : NVIDIA RTX Virtual Workstation
  Product Architecture   : Turing
  Display Mode           : Enabled
  Display Active         : Disabled
  Persistence Mode       : Enabled
  MIG Mode
    Current              : N/A
    Pending              : N/A
  Accounting Mode        : Disabled
  Accounting Mode Buffer Size : 4000
  Driver Model
    Current              : N/A
    Pending              : N/A
  Serial Number          : N/A
  GPU UUID               : GPU-c39a95d5-1f53-11b2-8ae7-7186d41f767b
  Minor Number           : 0
  VBIOS Version          : 00.00.00.00.00
  MultiGPU Board         : No
  Board ID               : 0x202
  GPU Part Number        : N/A
  Module ID              : N/A
  Inforom Version
    Image Version        : N/A
    OEM Object           : N/A
    ECC Object           : N/A
    Power Management Object : N/A
  GPU Operation Mode
    Current              : N/A
    Pending              : N/A
  GSP Firmware Version   : N/A
  GPU Virtualization Mode
    Virtualization Mode : VGPU
    Host VGPU Mode      : N/A
  vGPU Software Licensed Product
    Product Name         : NVIDIA RTX Virtual Workstation
    License Status       : Licensed (Expiry: 2022-9-26 20:51:4 GMT)
  IBMNPU

```

Figure 59. License verification on vPC and vDWS profile with host running vGPU Manager

```

root@localhost/etc/nvidia/ClientConfigToken
[root@localhost ClientConfigToken]# nvidia-smi -q
=====NVSMI LOG=====
Timestamp                : Sun Sep 25 23:47:24 2022
Driver Version           : 510.85.02
CUDA Version             : 11.6
Attached GPUs            : 1
GPU 00000000:02:02.0
  Product Name           : GRID T4-4C
  Product Brand          : NVIDIA Virtual Compute Server
  Product Architecture   : Turing
  Display Mode           : Enabled
  Display Active         : Disabled
  Persistence Mode       : Enabled
  MIG Mode
    Current              : N/A
    Pending              : N/A
  Accounting Mode        : Disabled
  Accounting Mode Buffer Size : 4000
  Driver Model
    Current              : N/A
    Pending              : N/A
  Serial Number          : N/A
  GPU UUID               : GPU-a719e14f-1dea-11b2-a255-e6767e650231
  Minor Number           : 0
  VBIOS Version          : 00.00.00.00.00
  MultiGPU Board         : No
  Board ID               : 0x202
  GPU Part Number        : N/A
  Module ID              : N/A
  Inforom Version
    Image Version        : N/A
    OEM Object           : N/A
    ECC Object           : N/A
    Power Management Object : N/A
  GPU Operation Mode
    Current              : N/A
    Pending              : N/A
  GSP Firmware Version   : N/A
  GPU Virtualization Mode
    Virtualization Mode : VGPU
    Host VGPU Mode      : N/A
  vGPU Software Licensed Product
    Product Name         : NVIDIA Virtual Compute Server
    License Status       : Licensed (Expiry: 2022-9-27 4:53:17 GMT)
  IBMNPU
  Relaxed Ordering Mode  : N/A
  PCI
    Bus                 : 0x02
    Device              : 0x02
    Domain              : 0x0000
    Device Id           : 0x1EB810DE
    Bus Id              : 00000000:02:02.0

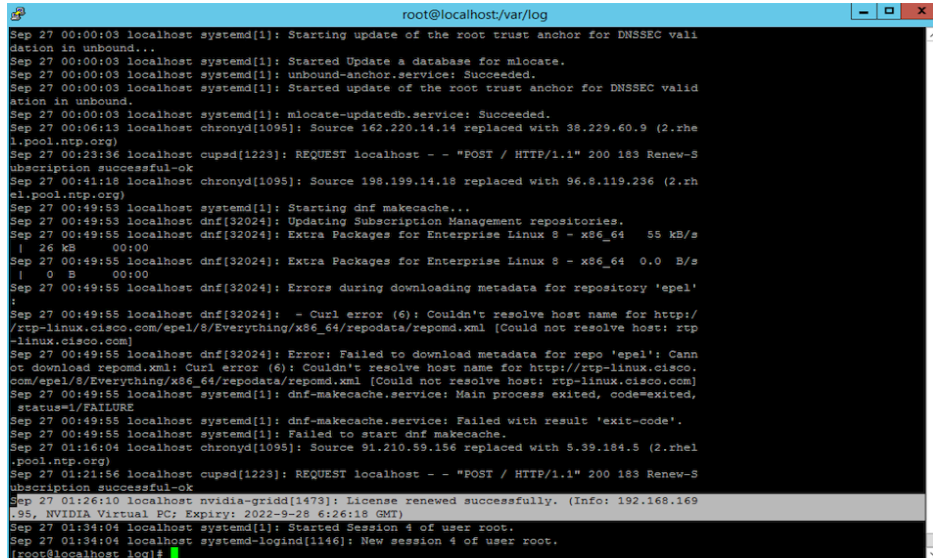
```

Figure 60. License verification on vCS profile with host running NVAIE Manager

Linux licensing event logs

In Linux, logging is enabled by default. The licensing events are logged in the standard activity log in the `/var/log` directory (Figure 61). The name of the log file in this directory depends on the Linux distribution. Here are some examples:

- For RHEL and CentOS distributions, the file name is **messages**.
- For Ubuntu distributions, the file name is **syslog**.



```
root@localhost:/var/log
Sep 27 00:00:03 localhost systemd[1]: Starting update of the root trust anchor for DNSSEC validation in unbound...
Sep 27 00:00:03 localhost systemd[1]: Started Update a database for mlocate.
Sep 27 00:00:03 localhost systemd[1]: unbound-anchor.service: Succeeded.
Sep 27 00:00:03 localhost systemd[1]: Started update of the root trust anchor for DNSSEC validation in unbound.
Sep 27 00:00:03 localhost systemd[1]: mlocate-updatedb.service: Succeeded.
Sep 27 00:06:13 localhost chronyd[1095]: Source 162.220.14.14 replaced with 38.229.60.9 (2.rhel.pool.ntp.org)
Sep 27 00:23:36 localhost cupsd[1223]: REQUEST localhost -- "POST / HTTP/1.1" 200 183 Renew-Subscription successful-ok
Sep 27 00:41:18 localhost chronyd[1095]: Source 198.199.14.18 replaced with 96.8.119.236 (2.rhel.pool.ntp.org)
Sep 27 00:49:53 localhost systemd[1]: Starting dnf makecache...
Sep 27 00:49:53 localhost dnf[32024]: Updating Subscription Management repositories.
Sep 27 00:49:55 localhost dnf[32024]: Extra Packages for Enterprise Linux 8 - x86_64 55 kB/s
| 26 kB 00:00
Sep 27 00:49:55 localhost dnf[32024]: Extra Packages for Enterprise Linux 8 - x86_64 0.0 B/s
| 0 B 00:00
Sep 27 00:49:55 localhost dnf[32024]: Errors during downloading metadata for repository 'epel':
+
Sep 27 00:49:55 localhost dnf[32024]: - Curl error (6): Couldn't resolve host name for http://rtp-linux.cisco.com/epel/8/Everything/x86_64/repodata/repomd.xml [Could not resolve host: rtp-linux.cisco.com]
Sep 27 00:49:55 localhost dnf[32024]: Error: Failed to download metadata for repo 'epel': Cannot download repomd.xml: Curl error (6): Couldn't resolve host name for http://rtp-linux.cisco.com/epel/8/Everything/x86_64/repodata/repomd.xml [Could not resolve host: rtp-linux.cisco.com]
Sep 27 00:49:55 localhost systemd[1]: dnf-makecache.service: Main process exited, code=exited, status=1/FAILURE
Sep 27 00:49:55 localhost systemd[1]: dnf-makecache.service: Failed with result 'exit-code'.
Sep 27 00:49:55 localhost systemd[1]: Failed to start dnf makecache.
Sep 27 01:16:04 localhost chronyd[1095]: Source 91.210.59.156 replaced with 5.39.184.5 (2.rhel.pool.ntp.org)
Sep 27 01:21:56 localhost cupsd[1223]: REQUEST localhost -- "POST / HTTP/1.1" 200 183 Renew-Subscription successful-ok
Sep 27 01:26:10 localhost nvidia-gridd[1473]: License renewed successfully. (Info: 192.168.169.195, NVIDIA Virtual PC; Expiry: 2022-9-28 6:26:18 GMT)
Sep 27 01:34:04 localhost systemd[1]: Started Session 4 of user root.
Sep 27 01:34:04 localhost systemd-logind[1146]: New session 4 of user root.
[root@localhost log]#
```

Figure 61.

Log file

GPU installation requirements and verification on FlashStack bare-metal Microsoft Windows Server 2016 and 2022

This section and the next two discuss how to install the OS on bare-metal hardware deployed in the FlashStack environment and how to boot these operating systems from Pure storage. The operating systems and versions used here are the same as those described previously for VMware ESXi virtual machines.

Although this document provides details for Windows Server 2022, Windows Server 2016 was also tested, and the same findings were confirmed.

When installing the Windows Server setup, when you are required to select a storage disk, you will not be able to see the Pure storage logical unit number (LUN) until you install the Cisco VIC drivers. You can download the Cisco drivers ISO file from [Cisco.com](https://software.cisco.com/download/home/286329080/type/283853158/release/5.0(2d)) by searching for X210c drivers at [https://software.cisco.com/download/home/286329080/type/283853158/release/5.0\(2d\)](https://software.cisco.com/download/home/286329080/type/283853158/release/5.0(2d)).

1. Download the “Image of UCS X-Series related Windows drivers only” ISO file and mount the file on the Linux Kernel-based Virtual Machine (KVM). Click to load the drivers and refresh the screen (Figure 62).
2. Select the appropriate OS and browse to the network drive (Figure 63).
3. Install the VIC driver and mount the Windows OS ISO file.
4. Refresh the drivers again. The Pure Storage boot LUN will appear to continue the installation.

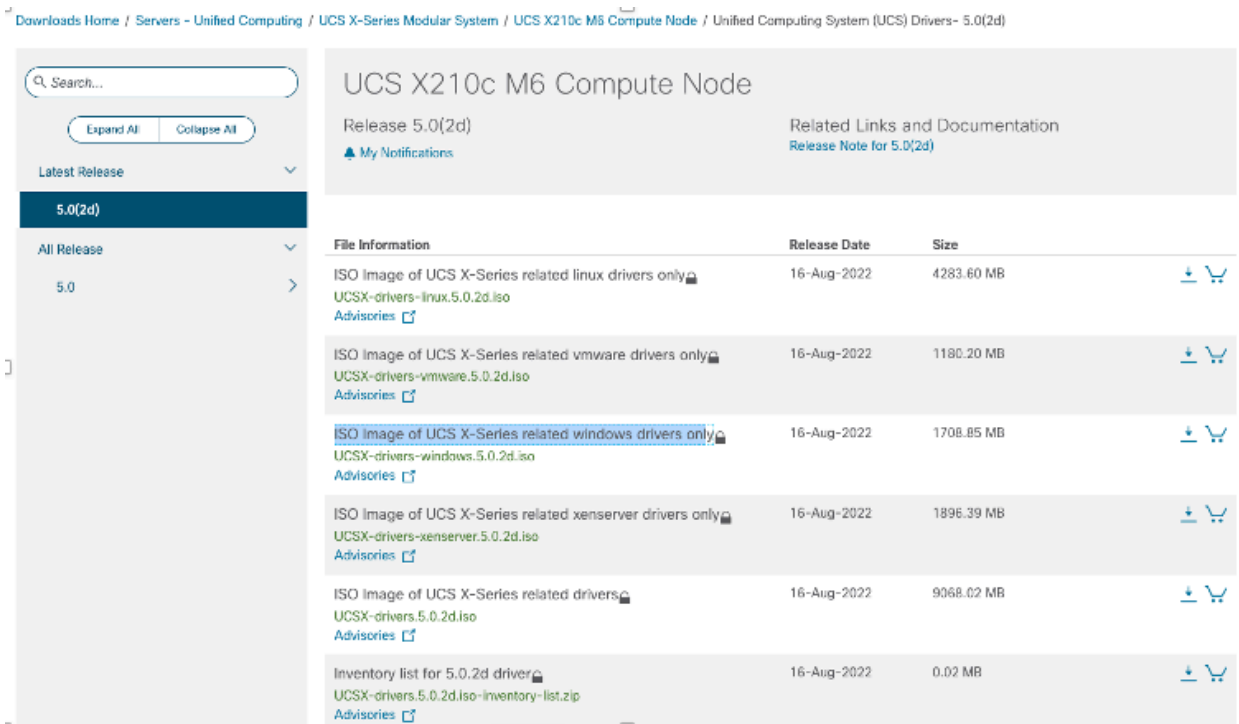


Figure 62.
Download the ISO file

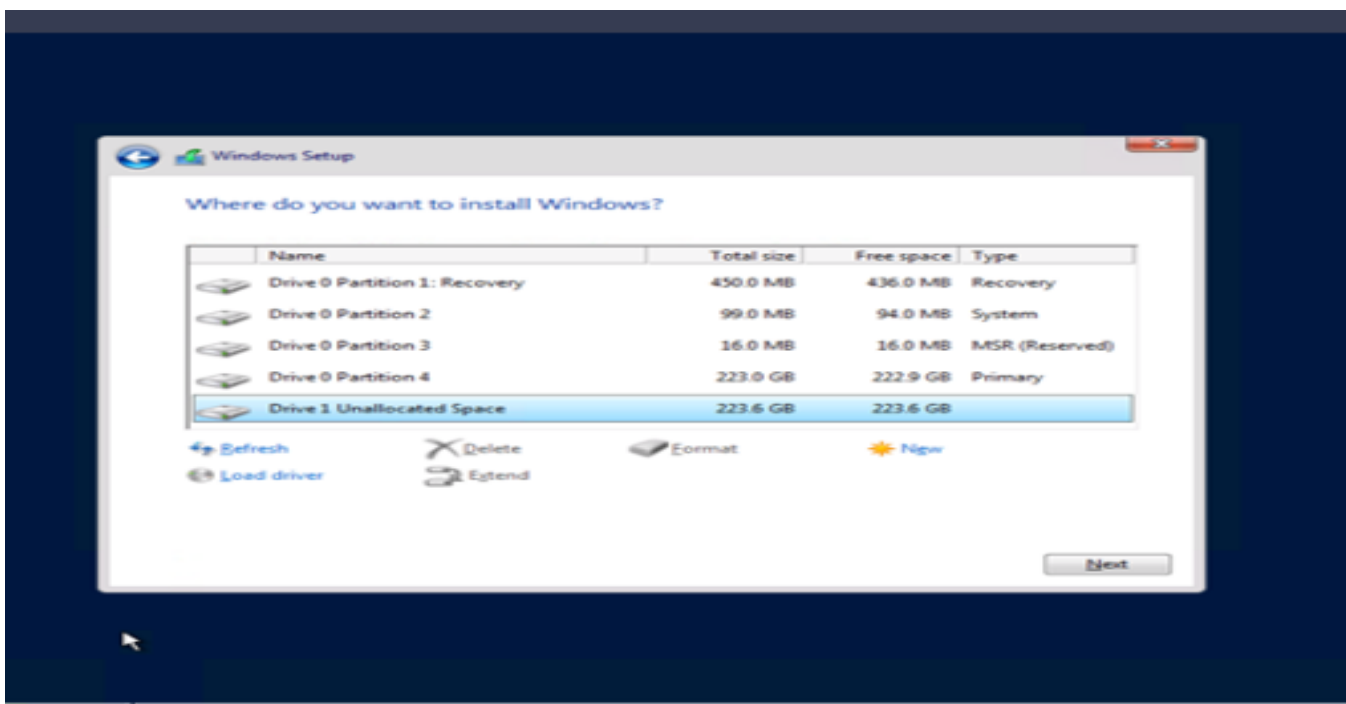


Figure 63.
Choose the network drive

- After the Windows Server OS is installed, each NVIDIA GPU will be listed as a 3D Video Controller instance under Other Devices and require driver installation (Figure 64).

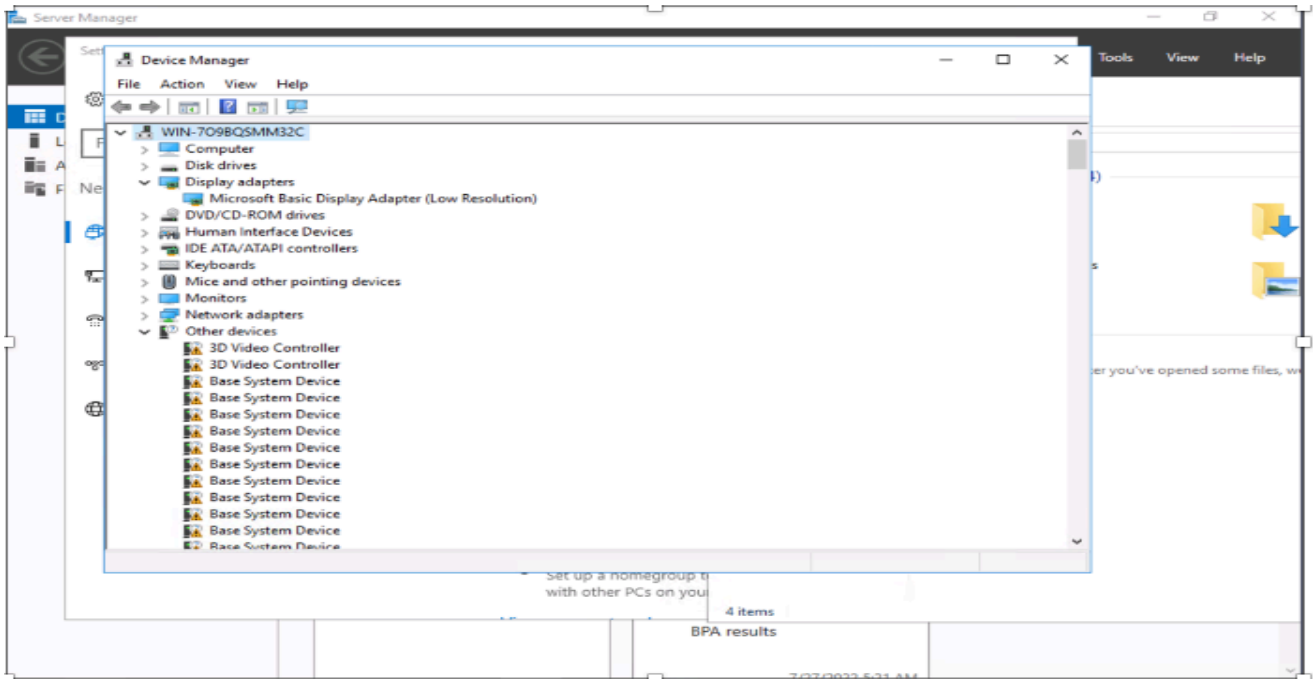


Figure 64.
NVIDIA GPUs listed as 3D Video Controller instances

6. Use the same ISO file mounted earlier and check the Video section to install NVIDIA drivers (Figures 65 and 66).

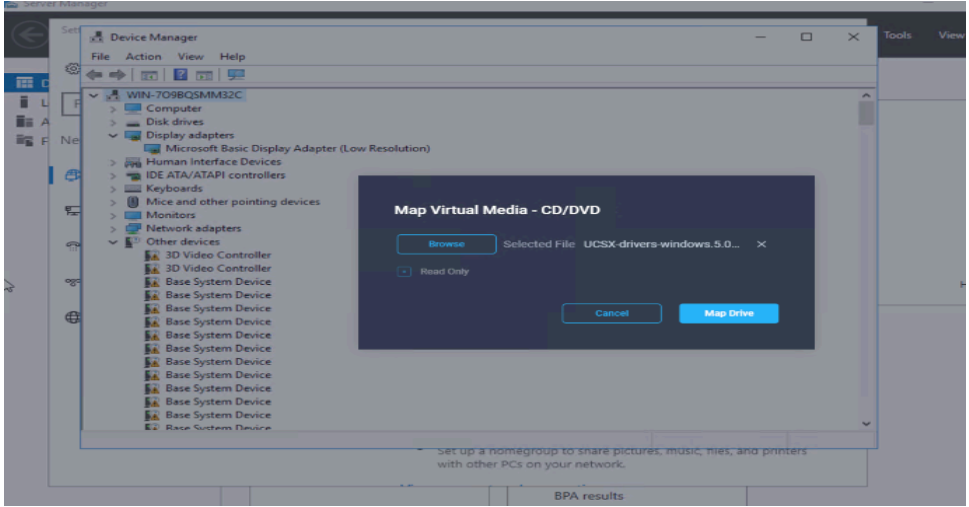


Figure 65.
Browsing to the drive

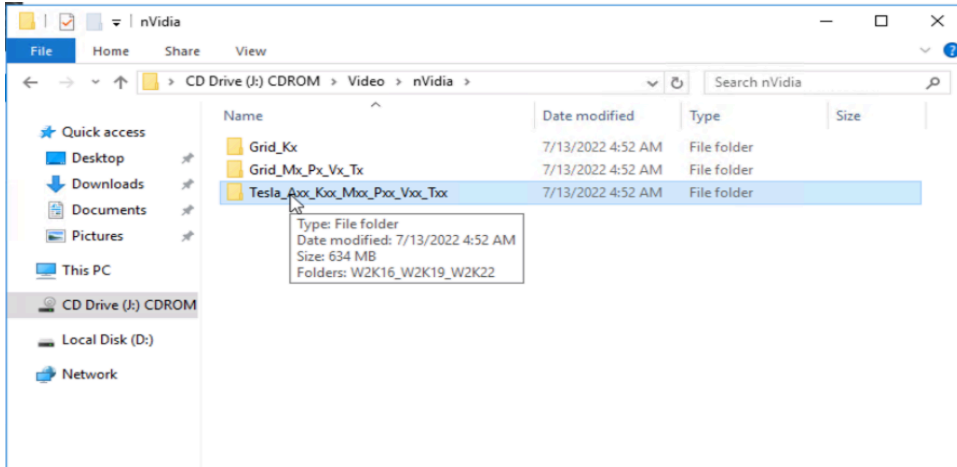


Figure 66.
Installing the NVIDIA drivers

Figure 67 shows the drivers successfully installed.

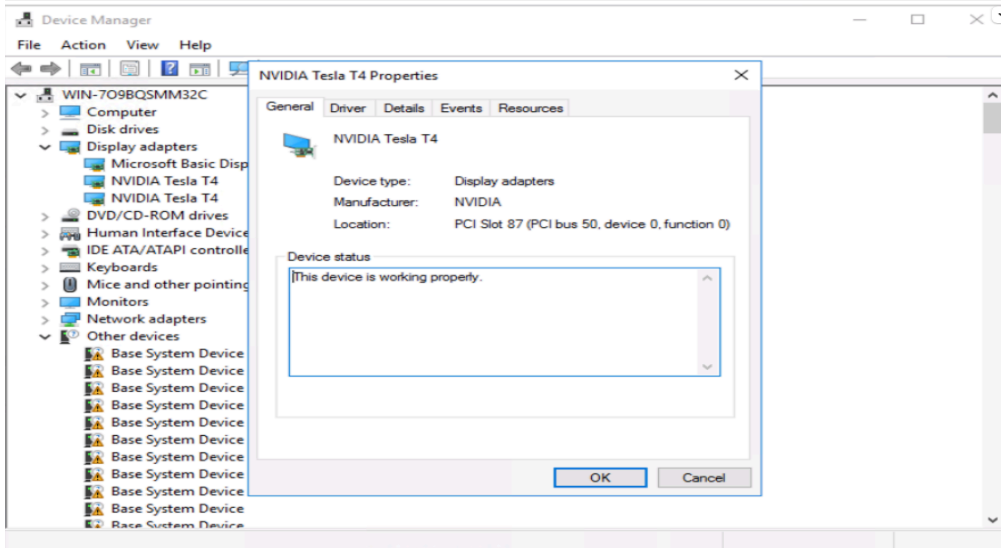


Figure 67.
Drivers successfully installed

The installation process for licenses on bare-metal servers is the same as the process discussed earlier for a Windows Server 2022 virtual machine.

GPU installation requirements and verification on FlashStack bare-metal RHEL 8.6

As was the case with Windows Server, multiple RHEL OS versions were tested: in this case, RHEL 8.4, 8.6, and 9.0. Release 8.6 was the one with the latest support drivers in the NVIDIA portal.

1. When attempting to boot the RHEL storage LUN as a disk, you need to modify the settings. Open the installation screen (Figure 68).

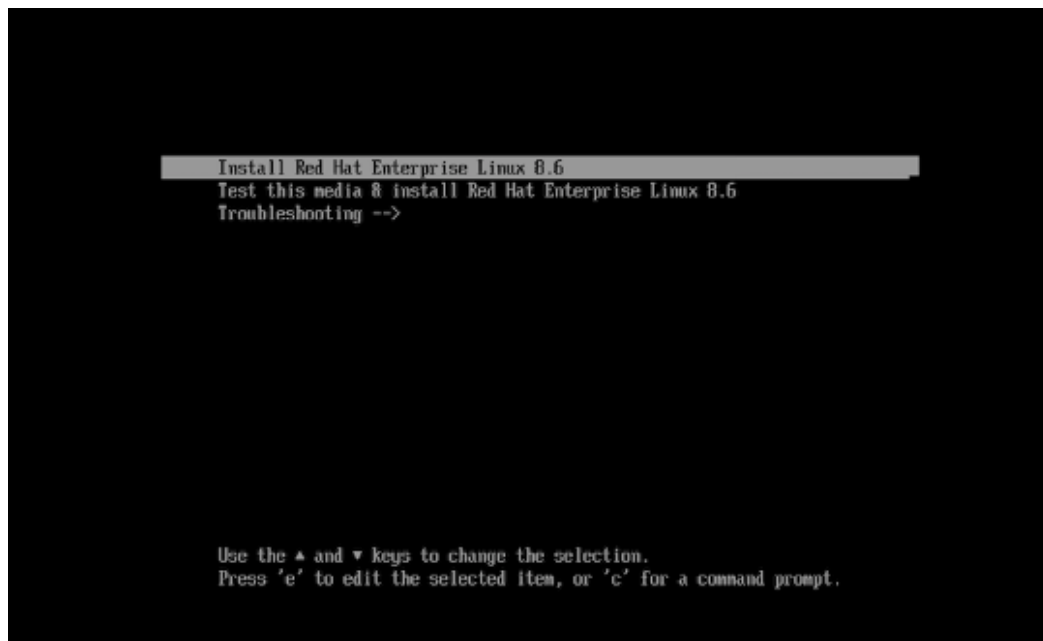


Figure 68.

Getting ready to modify the settings to boot the RHEL storage LUN as a disk

2. Press **e**. Then type **rd.iscsi.ibtf=1** and press Ctrl-X (Figure 69).

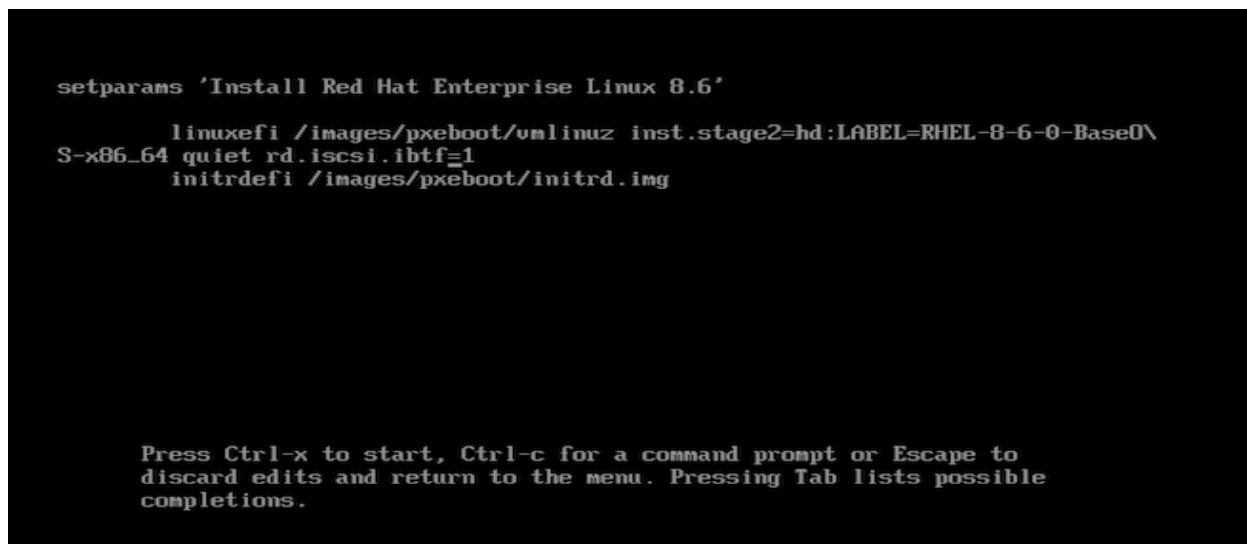


Figure 69.

Editing the file

- When the installation summary screen appears, select the installation destination. Then click Add a Disk and click Refresh List to see the storage LUN for the software installation (Figures 70 and 71).

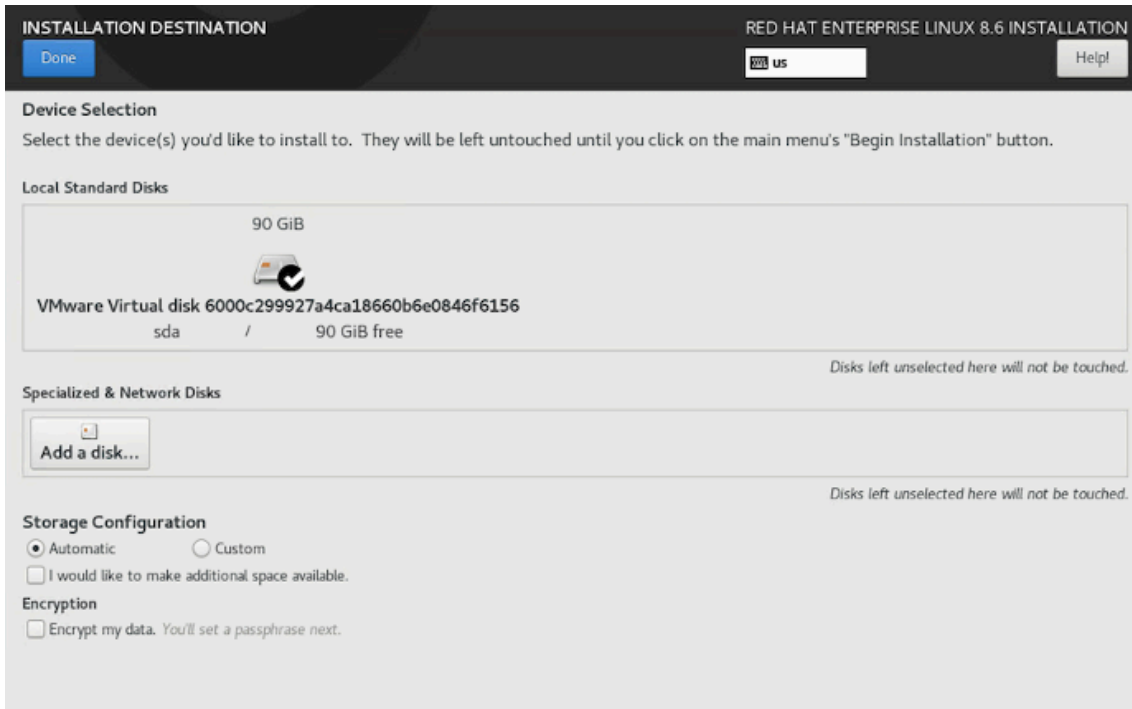


Figure 70.
Select the installation destination

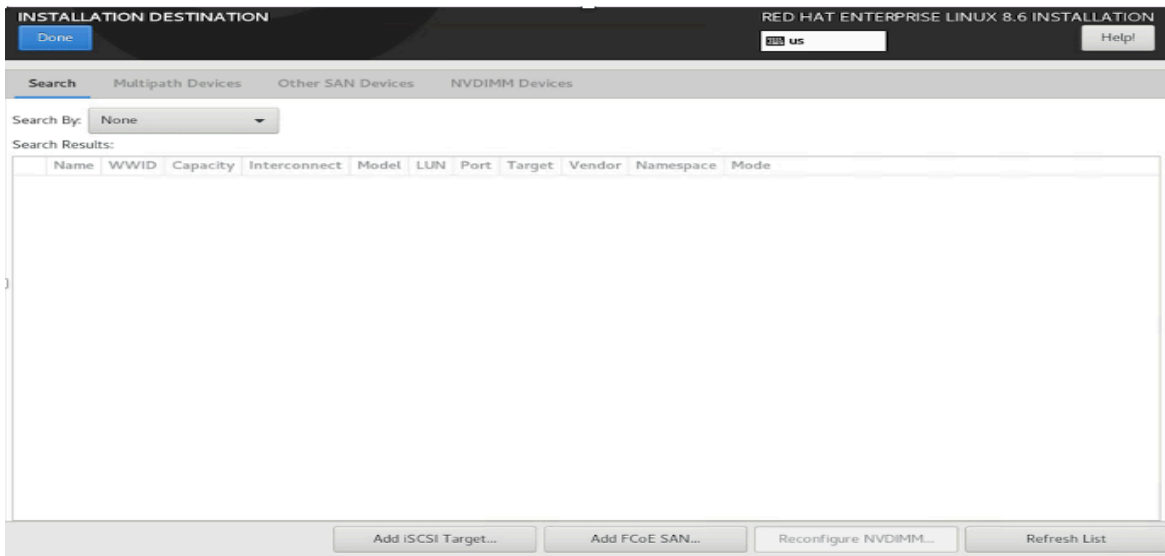
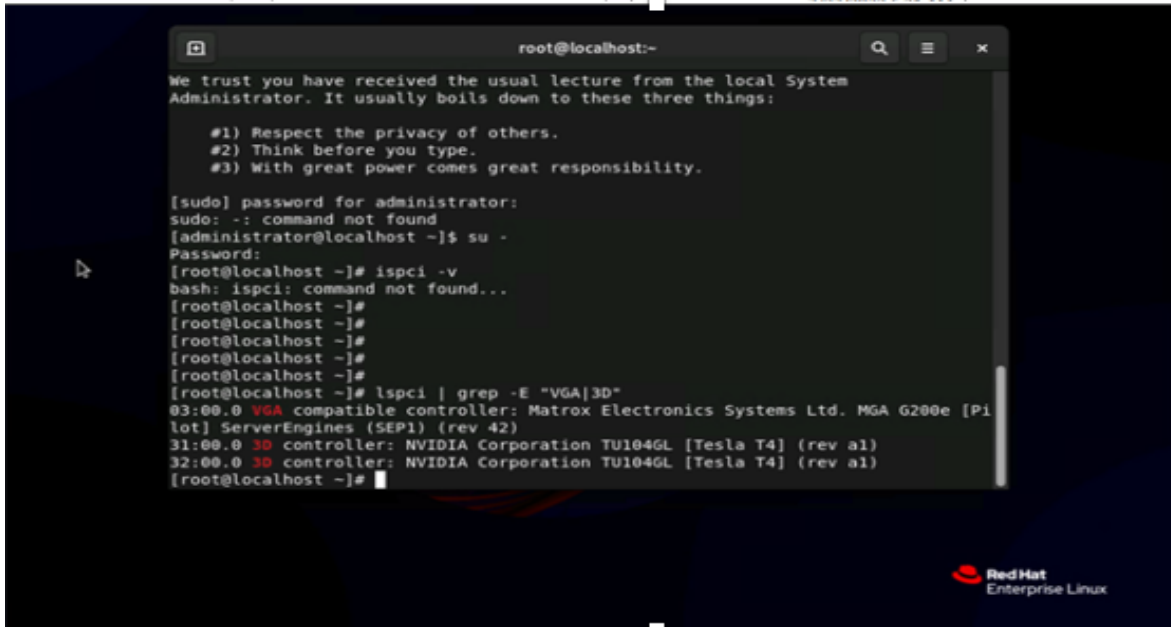


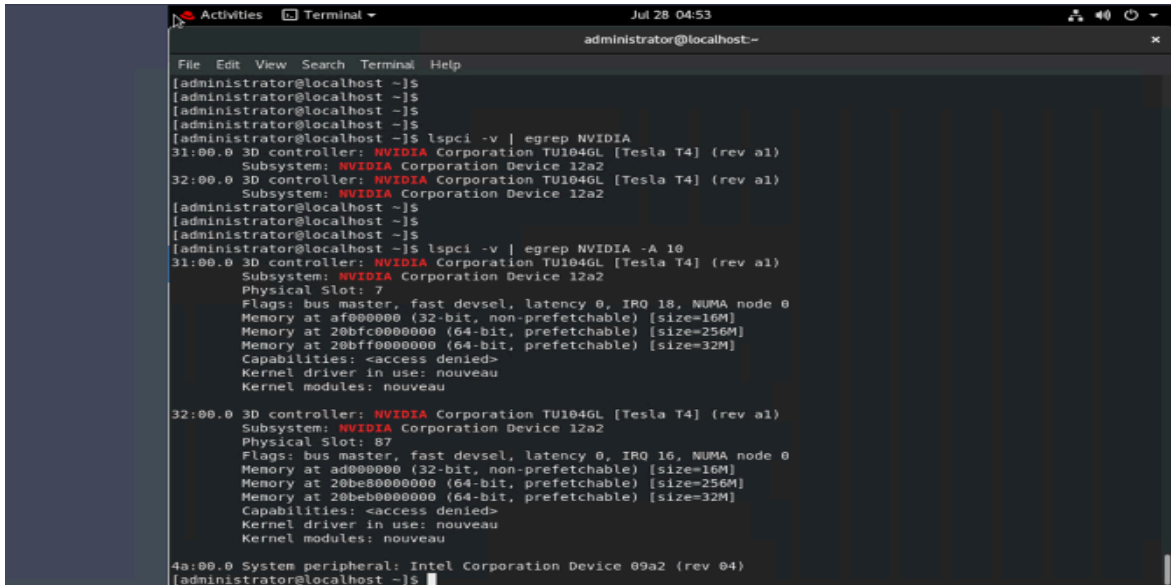
Figure 71.
Click Refresh List

- After the OS is installed, use the **grep** command `lspci -v | egrep NVIDIA -A10` to confirm that hardware GPUs are present for the OS (Figures 72 and 73).



```
root@localhost:~  
We trust you have received the usual lecture from the local System  
Administrator. It usually boils down to these three things:  
  
#1) Respect the privacy of others.  
#2) Think before you type.  
#3) With great power comes great responsibility.  
  
[sudo] password for administrator:  
sudo: -: command not found  
[administrator@localhost ~]$ su -  
Password:  
[root@localhost ~]# lspci -v  
bash: lspci: command not found...  
[root@localhost ~]#  
[root@localhost ~]#  
[root@localhost ~]#  
[root@localhost ~]#  
[root@localhost ~]#  
[root@localhost ~]# lspci | grep -E "VGA|3D"  
03:00.0 VGA compatible controller: Matrox Electronics Systems Ltd. MGA G200e [Pi  
lot] ServerEngines (SEP1) (rev 42)  
31:00.0 3D controller: NVIDIA Corporation TU104GL [Tesla T4] (rev a1)  
32:00.0 3D controller: NVIDIA Corporation TU104GL [Tesla T4] (rev a1)  
[root@localhost ~]#
```

Figure 72. Using the `grep` command to check for hardware GPUs



```
administrator@localhost:~  
[administrator@localhost ~]$  
[administrator@localhost ~]$  
[administrator@localhost ~]$  
[administrator@localhost ~]$  
[administrator@localhost ~]$ lspci -v | egrep NVIDIA  
31:00.0 3D controller: NVIDIA corporation TU104GL [Tesla T4] (rev a1)  
Subsystem: NVIDIA Corporation Device 12a2  
32:00.0 3D controller: NVIDIA Corporation TU104GL [Tesla T4] (rev a1)  
Subsystem: NVIDIA Corporation Device 12a2  
[administrator@localhost ~]$  
[administrator@localhost ~]$  
[administrator@localhost ~]$  
[administrator@localhost ~]$ lspci -v | egrep NVIDIA -A 10  
31:00.0 3D controller: NVIDIA Corporation TU104GL [Tesla T4] (rev a1)  
Subsystem: NVIDIA Corporation Device 12a2  
Physical Slot: 7  
Flags: bus master, fast devsel, latency 0, IRQ 18, NUMA node 0  
Memory at af000000 (32-bit, non-prefetchable) [size=16M]  
Memory at 20bfc0000000 (64-bit, prefetchable) [size=256M]  
Memory at 20bf00000000 (64-bit, prefetchable) [size=32M]  
Capabilities: <access denied>  
Kernel driver in use: nouveau  
Kernel modules: nouveau  
32:00.0 3D controller: NVIDIA Corporation TU104GL [Tesla T4] (rev a1)  
Subsystem: NVIDIA Corporation Device 12a2  
Physical Slot: 87  
Flags: bus master, fast devsel, latency 0, IRQ 16, NUMA node 0  
Memory at ad000000 (32-bit, non-prefetchable) [size=16M]  
Memory at 20be80000000 (64-bit, prefetchable) [size=256M]  
Memory at 20beb0000000 (64-bit, prefetchable) [size=32M]  
Capabilities: <access denied>  
Kernel driver in use: nouveau  
Kernel modules: nouveau  
4a:00.0 System peripheral: Intel Corporation Device 09a2 (rev 04)  
[administrator@localhost ~]$
```

Figure 73. Checking for hardware GPUs

RHEL can be used as a hypervisor or a regular OS client. If you use it as a hypervisor, you need to install the Red Hat hypervisor driver (host driver) from the NVIDIA portal to install NVIDIA GPU Manager. If you are using it as a regular client, you need to install guest drivers for Red Hat only.

You can use the same links and process as were described to install the drivers for the virtual machine installation of RHEL drivers. However, the licensing process differs a little. See the following link to learn more about the licensing process: <https://docs.nvidia.com/license-system/latest/nvidia-license-system-user-guide/index.html#configuring-nls-licensed-client-on-linux>.

GPU installation requirements and verification on FlashStack bare-metal Ubuntu 20.04.4

Ubuntu follows the same process as the RHEL OS (Figures 74 and 75).

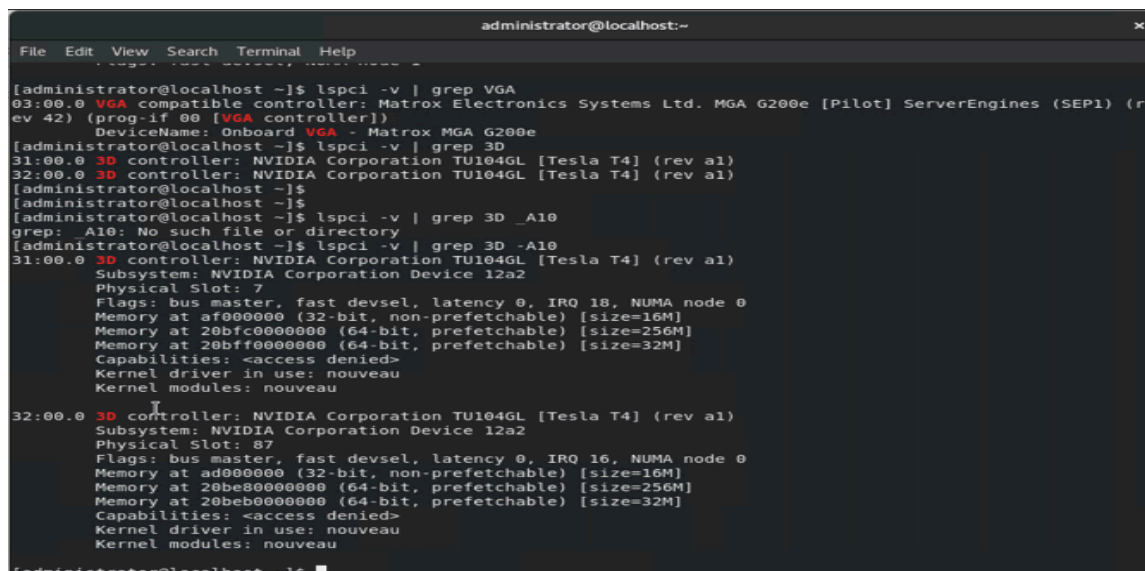


```
GNU GRUB version 2.04

getparams 'Install Ubuntu Server'

set gfxpayload=keep
linux /casper/vmlinuz quiet ---
initrd /casper/initrd
```

Figure 74. Editing and adding a path for the storage LUN boot by pressing **e**



```
administrator@localhost:~
File Edit View Search Terminal Help

[administrator@localhost ~]$ lspci -v | grep VGA
03:00.0 VGA compatible controller: Matrox Electronics Systems Ltd. MGA G200e [Pilot] ServerEngines (SEP1) (rev 42) (prog-if 00 [VGA controller])
DeviceName: Onboard VGA - Matrox MGA G200e
[administrator@localhost ~]$ lspci -v | grep 3D
31:00.0 3D controller: NVIDIA Corporation TU104GL [Tesla T4] (rev a1)
32:00.0 3D controller: NVIDIA Corporation TU104GL [Tesla T4] (rev a1)
[administrator@localhost ~]$
[administrator@localhost ~]$ lspci -v | grep 3D _A10
grep: _A10: No such file or directory
[administrator@localhost ~]$ lspci -v | grep 3D -A10
31:00.0 3D controller: NVIDIA Corporation TU104GL [Tesla T4] (rev a1)
Subsystem: NVIDIA Corporation Device 12a2
Physical Slot: 7
Flags: bus master, fast devsel, latency 0, IRQ 18, NUMA node 0
Memory at af000000 (32-bit, non-prefetchable) [size=16M]
Memory at 20bfc0000000 (64-bit, prefetchable) [size=256M]
Memory at 20bfff000000 (64-bit, prefetchable) [size=32M]
Capabilities: <access denied>
Kernel driver in use: nouveau
Kernel modules: nouveau

32:00.0 3D controller: NVIDIA Corporation TU104GL [Tesla T4] (rev a1)
Subsystem: NVIDIA Corporation Device 12a2
Physical Slot: 87
Flags: bus master, fast devsel, latency 0, IRQ 16, NUMA node 0
Memory at ad000000 (32-bit, non-prefetchable) [size=16M]
Memory at 20be80000000 (64-bit, prefetchable) [size=256M]
Memory at 20beb0000000 (64-bit, prefetchable) [size=32M]
Capabilities: <access denied>
Kernel driver in use: nouveau
Kernel modules: nouveau

[administrator@localhost ~]$
```

Figure 75. Edited file

The following links confirm that this setup was tested and that the NVIDIA GPU T4 appeared on the OS as a PCIe device:

- Driver installation follows same steps as earlier stated for RHEL OS: <https://docs.nvidia.com/grid/latest/grid-vgpu-user-guide/index.html#ubuntu-install-configure-vgpu>
- Licensing: <https://docs.nvidia.com/license-system/latest/nvidia-license-system-user-guide/index.html#configuring-nls-licensed-client-on-linux>

For more information

This section includes links to various product pages related to the topics in this document.

Computing

- Cisco Intersight: <https://www.intersight.com>
- Cisco Intersight managed mode: https://www.cisco.com/c/en/us/td/docs/unified_computing/Intersight/b_Intersight_Managed_Mode_Configuration_Guide.html
- Cisco Unified Computing System: <http://www.cisco.com/en/US/products/ps10265/index.html>
- Cisco UCS 6400 Series Fabric Interconnects: <https://www.cisco.com/c/en/us/products/collateral/servers-unified-computing/datasheet-c78-741116.html>
- NVIDIA licensing:
 - <https://docs.nvidia.com/license-system/latest/>
 - <https://docs.nvidia.com/license-system/latest/nvidia-license-system-user-guide/index.html#about-cls-instances>
- NVIDIA vGPU profile: <https://docs.nvidia.com/grid/13.0/grid-vgpu-user-guide/index.html>
- NVIDIA deploying virtual machine for virtual desktop: https://docs.omniverse.nvidia.com/prod_deployment/prod_deployment/vgpu-virtual-desktop.html
- NVIDIA software download: <https://ui.licensing.nvidia.com/software>
- NVIDIA vGPU Manager deployment: https://docs.omniverse.nvidia.com/prod_deployment/prod_deployment/installing-vgpu-Manager.html#preparing-the-vib-file-for-install
- NVIDIA license generation: <https://ui.licensing.nvidia.com/>

Network

- Cisco Nexus 9000 Series Switches: <http://www.cisco.com/c/en/us/products/switches/nexus-9000-series-switches/index.html>
- Cisco MDS 9132T Switches: <https://www.cisco.com/c/en/us/products/collateral/storage-networking/mds-9100-series-multilayer-fabric-switches/datasheet-c78-739613.html>

Storage

- [Pure Storage Interoperability](https://support.purestorage.com/FlashArray/Getting_Started/Compatibility_Matrix) (note that this interoperability list will require a support login form Pure): https://support.purestorage.com/FlashArray/Getting_Started/Compatibility_Matrix

Virtualization

- VMware vCenter Server: <http://www.vmware.com/products/vcenter-server/overview.html>
- VMware vSphere: <https://www.vmware.com/products/vsphere>

Interoperability matrix

- Cisco UCS Hardware Compatibility Matrix: <https://ucshcltool.cloudapps.cisco.com/public/>
- VMware and Cisco Unified Computing System: <http://www.vmware.com/resources/compatibility>
- Pure Storage Interoperability Matrix Tool: https://support.purestorage.com/FlashStack@/Product_Information/FlashStack@_Compatibility_Matrix

Americas Headquarters
Cisco Systems, Inc.
San Jose, CA

Asia Pacific Headquarters
Cisco Systems (USA) Pte. Ltd.
Singapore

Europe Headquarters
Cisco Systems International BV Amsterdam,
The Netherlands

Cisco has more than 200 offices worldwide. Addresses, phone numbers, and fax numbers are listed on the Cisco Website at <https://www.cisco.com/go/offices>.

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: <https://www.cisco.com/go/trademarks>. Third-party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1110R)