

# Private AI Ready Infrastructure for VMware Cloud Foundation

Modified on 23 JUL 2024

VMware Cloud Foundation services

You can find the most up-to-date technical documentation on the VMware by Broadcom website at:

<https://docs.vmware.com/>

**VMware by Broadcom**  
3401 Hillview Ave.  
Palo Alto, CA 94304  
[www.vmware.com](http://www.vmware.com)

Copyright © 2024 Broadcom. All Rights Reserved. The term “Broadcom” refers to Broadcom Inc. and/or its subsidiaries. For more information, go to <https://www.broadcom.com>. All trademarks, trade names, service marks, and logos referenced herein belong to their respective companies.

# Contents

About Private AI Ready Infrastructure for VMware Cloud Foundation	6
<b>1 Design Objectives of Private AI Ready Infrastructure for VMware Cloud Foundation</b>	<b>11</b>
<b>2 Detailed Design of Private AI Ready Infrastructure for VMware Cloud Foundation</b>	<b>13</b>
Logical Design for Private AI Ready Infrastructure for VMware Cloud Foundation	13
Compute Design for Private AI Ready Infrastructure for VMware Cloud Foundation	15
Physical Infrastructure Resource Design for Private AI Ready Infrastructure for VMware Cloud Foundation	18
Network Design for Private AI Ready Infrastructure for VMware Cloud Foundation	22
Accelerators Design for Private AI Ready Infrastructure for VMware Cloud Foundation	25
Storage Design for Private AI Ready Infrastructure for VMware Cloud Foundation	27
Deployment Specification for Private AI Ready Infrastructure for VMware Cloud Foundation	29
Deployment Model for Private AI Ready Infrastructure for VMware Cloud Foundation	30
Sizing Compute and Storage Resources for Private AI Ready Infrastructure for VMware Cloud Foundation	33
Life Cycle Management for Private AI Ready Infrastructure for VMware Cloud Foundation	34
Information Security and Access Control Design for Private AI Ready Infrastructure for VMware Cloud Foundation	35
<b>3 Detailed Design for VMware Private AI Foundation with NVIDIA for Private AI Ready Infrastructure for VMware Cloud Foundation</b>	<b>40</b>
NVIDIA Licensing System Design for Private AI Ready Infrastructure for VMware Cloud Foundation	44
VMware Data Services Manager Design for Private AI Ready Infrastructure for VMware Cloud Foundation	46
<b>4 Planning and Preparation for Private AI Ready Infrastructure for VMware Cloud Foundation</b>	<b>50</b>
<b>5 Implementation of Private AI Ready Infrastructure for VMware Cloud Foundation</b>	<b>51</b>
Configure the vSphere Environment for Private AI Ready Infrastructure for VMware Cloud Foundation	54
Enable vSphere vMotion for vGPU-Enabled Virtual Machines for Private AI Ready Infrastructure for VMware Cloud Foundation	54
Install the Vendor GPU Driver on the ESXi Hosts for Private AI Ready Infrastructure for VMware Cloud Foundation	55

Deploy and Configure a Tanzu Kubernetes Grid Cluster for vSphere with Tanzu for Private AI Ready Infrastructure for VMware Cloud Foundation	56
Create a Namespace for the Tanzu Kubernetes Grid Cluster for Private AI Ready Infrastructure for VMware Cloud Foundation	57
Assign the New Tanzu Cluster Namespace Roles to Active Directory Groups for VMware Cloud Foundation	57
Add GPU-Enabled VM Classes for the Tanzu Kubernetes Grid Cluster for Private AI Ready Infrastructure for VMware Cloud Foundation	58
Provision a Tanzu Kubernetes Grid Cluster for Private AI Ready Infrastructure for VMware Cloud Foundation	61
Install the NVIDIA GPU Operator for Private AI Ready Infrastructure for VMware Cloud Foundation	63
Install the NVIDIA Network Operator for Private AI Ready Infrastructure for VMware Cloud Foundation	67
Adding VMware Private AI Foundation with NVIDIA to Private AI Ready Infrastructure for VMware Cloud Foundation	71
Create AI Self-Service Catalog Items in VMware Aria Automation for Private AI Ready Infrastructure for VMware Cloud Foundation	73
Deploy VMware Data Services Manager for Private AI Ready Infrastructure for VMware Cloud Foundation	74
Create a Content Library with Deep Learning VM Images for Private AI Ready Infrastructure for VMware Cloud Foundation	74
Additional Setup of VMware Private AI Foundation with NVIDIA for Private AI Ready Infrastructure in a Disconnected VMware Cloud Foundation Environment	76
Configure a Replicated Harbor Instance for Private AI Ready Infrastructure in a Disconnected VMware Cloud Foundation Environment	76
Upload Container Images to a Private Harbor Registry for Private AI Ready Infrastructure in a Disconnected VMware Cloud Foundation Environment	78
Configure a Content Library with Ubuntu TKr for Private AI Ready Infrastructure in a Disconnected VMware Cloud Foundation Environment	80
Deploy AI Workloads by Using VMware Aria Automation for Private AI Ready Infrastructure for VMware Cloud Foundation	81
Deploy a Deep Learning VM from the VMware Aria Automation Self-Service Catalog for Private AI Ready Infrastructure for VMware Cloud Foundation	81
Provision a Tanzu Kubernetes Grid Cluster from the VMware Aria Automation Self-Service Catalog for Private AI Ready Infrastructure for VMware Cloud Foundation	83
Create a Vector Database Catalog Item in VMware Aria Automation for RAG Workloads for Private AI Ready Infrastructure for VMware Cloud Foundation	85
Deploy a Vector Database by Using a Self-Service Catalog Item in VMware Aria Automation for RAG Workloads for Private AI Ready Infrastructure for VMware Cloud Foundation	86

## 6 Operational Guidance for Private Private AI Ready Infrastructure for VMware Cloud Foundation 88

Personas in Private AI Ready Infrastructure for VMware Cloud Foundation	88
Operational Verification of VMware Private AI Foundation with NVIDIA	89
Verify the Status of the ESXi Host Components for Private AI Ready Infrastructure for VMware Cloud Foundation	90

Verify the Status of the GPU Operator for Private AI Ready Infrastructure for VMware Cloud Foundation	90
Verify the Status of the Network Operator for Private AI Ready Infrastructure for VMware Cloud Foundation	91
Operational Verification of vSphere with Tanzu for Private AI Ready Infrastructure for VMware Cloud Foundation	92
Verify the Status of the vSphere with Tanzu Service for Private AI Ready Infrastructure for VMware Cloud Foundation	92
Verify the Status of the Harbor Supervisor Service for Private AI Ready Infrastructure for VMware Cloud Foundation	93
Verify the Status of the vSphere Namespace for Private Private AI Ready Infrastructure for VMware Cloud Foundation	94
Verify the Status of the vSphere with Tanzu Resources for Private Private AI Ready Infrastructure for VMware Cloud Foundation	94
Certificate Management for Private Private AI Ready Infrastructure for VMware Cloud Foundation	96
Operational Verification of VMware Data Services Manager for Private AI Ready Infrastructure for VMware Cloud Foundation	96
Operational Verification for Private AI Ready Infrastructure for VMware Cloud Foundation by Deploying a RAG Workload	98
Password Management for Private AI Ready Infrastructure for VMware Cloud Foundation	104
Scale Management for Private AI Ready Infrastructure for VMware Cloud Foundation	105
Shutdown and Startup of Private AI Ready Infrastructure for VMware Cloud Foundation	107
Shut Down the Virtual Machines of vSphere with Tanzu for Private AI Ready Infrastructure for VMware Cloud Foundation	107
Start the vSphere with Tanzu Virtual Machines for Private AI Ready Infrastructure for VMware Cloud Foundation	108
<b>7 Solution Interoperability for Private AI Ready Infrastructure for VMware Cloud Foundation</b>	<b>109</b>
Monitoring and Alerting for Private AI Ready Infrastructure for VMware Cloud Foundation	109
Logging for Private AI Ready Infrastructure for VMware Cloud Foundation	110
Data Protection for Private AI Ready Infrastructure for VMware Cloud Foundation	110
Disaster Recovery for Private AI Ready Infrastructure for VMware Cloud Foundation	111
Life Cycle Management for Private AI Ready Infrastructure for VMware Cloud Foundation	111
<b>8 Design Decisions for Private AI Ready Infrastructure for <i>VMware Cloud Foundation</i></b>	<b>113</b>

# About Private AI Ready Infrastructure for VMware Cloud Foundation

The *Private AI Ready Infrastructure for VMware Cloud Foundation* validated solution provides design, implementation, and operational guidance for AI GPU-enabled accelerated workload domains that run on vSphere with Tanzu in the Software-Defined Data Center (SDDC) as part of VMware Cloud Foundation.

A VMware by Broadcom validated solution is a well-architected and validated implementation, built and tested by VMware to help customers deliver common business use cases. VMware validated solutions are operational, cost-effective, reliable, and secure. Each solution contains detailed design, implementation, and operational guidance.

This validated solution for private AI ready infrastructure provides technical guidance on how to design and implement a robust private AI ready infrastructure based on VMware Cloud Foundation.

The validated solution also covers the VMware Private AI Foundation with NVIDIA add-on solution on top of VMware Cloud Foundation. You can use VMware Private AI Foundation with NVIDIA for the following use cases:

- Running deep learning VMs for AI development based on NVIDIA GPUs and NVIDIA DL workloads.
- Provisioning Tanzu Kubernetes Grid (TKG) clusters for running AI container workloads on top of NVIDIA GPUs.

## Automation for This Design in VMware Cloud Foundation

VMware Cloud Foundation™ SDDC Manager® automates the implementation tasks for some design decisions. For the rest of the design decisions, as noted in the design implications, you must perform the implementation steps manually.

To provide a fast and efficient path for automating the *AI Ready Infrastructure for VMware Cloud Foundation* implementation, this document provides Microsoft PowerShell cmdlets using an open-source module as code-based alternatives to completing certain procedures in each SDDC component's user interface. You can directly reuse the PowerShell commands by replacing the provided sample values with values from your *VMware Cloud Foundation Planning and Preparation Workbook*.

## Intended Audience

The *Private AI Ready Infrastructure for VMware Cloud Foundation* document is intended for cloud architects, cloud administrators, DevOps/MLOps practitioners who are familiar with and want to use VMware software to deploy and manage a workload domain that runs vSphere with Tanzu workloads in the SDDC to meet specific and advanced technical requirements of AI workloads and providing the best performance possible. The document provides guidance for capacity, scalability, backup and restore, and extensibility for disaster recovery support.

## Support Matrix

*Private AI Ready Infrastructure for VMware Cloud Foundation* is compatible with certain versions of the VMware products that are used for implementing the solution.

Table 1-1. Software Components in Private AI Ready Infrastructure for VMware Cloud Foundation

VMware Cloud Foundation Version	Product Group	Component Versions
5.2.0	Products part of VMware Cloud Foundation	See <a href="#">VMware Cloud Foundation 5.2.0 Release Notes</a> .
	Solution-added products	VMware Data Services Manager 2.0.2. See <a href="#">VMware Data Services Manager 2.0 Release Notes</a> . VMware Data Services Manager is added as part of VMware Private AI Foundation with NVIDIA.
5.1.1	Products part of VMware Cloud Foundation	See <a href="#">VMware Cloud Foundation 5.1.0 Release Notes</a> .
	Solution-added products	VMware Data Services Manager 2.0.2. See <a href="#">VMware Data Services Manager 2.0 Release Notes</a> . VMware Data Services Manager is added as part of VMware Private AI Foundation with NVIDIA.

## Before You Apply This Guidance

To design and implement the *Private AI Ready Infrastructure for VMware Cloud Foundation* validated solution, your environment must have a certain configuration.

Table 1-2. Supported VMware Cloud Foundation Deployment

Workload Domain	Deployment Details
Management Domain	<p>Automated deployment by using VMware Cloud Builder. See the following <a href="#">VMware Cloud Foundation documentation</a>:</p> <ul style="list-style-type: none"> <li>■ For information on designing the management domain, see <a href="#">VMware Cloud Foundation Design Guide</a>.</li> <li>■ For information on deploying the management domain, see <a href="#">Getting Started with VMware Cloud Foundation</a> and <a href="#">VMware Cloud Foundation Deployment Guide</a>.</li> <li>■ For information on operating the management domain, see <a href="#">VMware Cloud Foundation Administration Guide</a> and <a href="#">VMware Cloud Foundation Operations Guide</a>.</li> </ul>
One or more virtual infrastructure workload domains with GPU-enabled ESXi hosts and using the vSphere Lifecycle Manager images.	<p>Automated deployment by using SDDC Manager. See the following <a href="#">VMware Cloud Foundation documentation</a>:</p> <ul style="list-style-type: none"> <li>■ For information on designing a VI workload domain, see <a href="#">VMware Cloud Foundation Design Guide</a>.</li> <li>■ For information on deploying the VI workload domains, see <a href="#">Getting Started with VMware Cloud Foundation</a> and <a href="#">VMware Cloud Foundation Administration Guide</a>.</li> <li>■ For information on operating the VI Workload domain, see <a href="#">VMware Cloud Foundation Operations Guide</a>.</li> </ul> <p>To view compatible NVIDIA GPU devices, see the <a href="#">VMware Compatibility Guide</a>.</p>
NSX Edge cluster in the VI workload domain	<p>Automated deployment by using SDDC Manager. See the following <a href="#">VMware Cloud Foundation documentation</a>:</p> <ul style="list-style-type: none"> <li>■ For information on designing the NSX Edge cluster, see <a href="#">VMware Cloud Foundation Design Guide</a>.</li> <li>■ For information on deploying the NSX Edge cluster, see <a href="#">Getting Started with VMware Cloud Foundation</a> and <a href="#">VMware Cloud Foundation Operations Guide</a>.</li> </ul> <p><b>Note</b> You must deploy the NSX Edge cluster with large-sized nodes. A cluster with smaller nodes is not compatible with Supervisor deployment.</p>



Table 1-2. Supported VMware Cloud Foundation Deployment (continued)

Workload Domain	Deployment Details
VMware Cloud Foundation integrated with Active Directory	Manual or PowerShell automated configuration of Active Directory over LDAP. See the <a href="#">Identity and Access Management for VMware Cloud Foundation</a> validated solution.
Deploy and configure vSphere with Tanzu on VMware Cloud Foundation	Manual or PowerShell automated configuration of vSphere with Tanzu. See the <a href="#">Developer Ready Infrastructure for VMware Cloud Foundation</a> validated solution.

Table 1-3. Components Required for VMware Private AI Foundation with NVIDIA

Component	Deployment Details
NVIDIA GPU device	Before you start using VMware Private AI Foundation with NVIDIA, make sure that the GPUs on your ESXi hosts are supported by VMware: <ul style="list-style-type: none"> <li>■ NVIDIA A100</li> <li>■ NVIDIA L40S</li> <li>■ NVIDIA H100</li> </ul>
Supported GPU sharing mode	<ul style="list-style-type: none"> <li>■ Time slicing</li> <li>■ Multi-Instance GPU (MIG)</li> </ul>
VMware Aria Automation	Manual or PowerShell automated deployment of VMware Aria Automation 8.18.0. See the <a href="#">Private Cloud Automation for VMware Cloud Foundation</a> validated solution.

## Overview of *Private AI Ready Infrastructure for VMware Cloud Foundation*

By applying the *Private AI Ready Infrastructure for VMware Cloud Foundation* validated solution, you implement Kubernetes natively on VMware vSphere within your VMware Cloud Foundation instance.

**Table 1-4. Implementation Overview of *Private AI Ready Infrastructure for VMware Cloud Foundation***

Stage	Steps
Plan and prepare the Private AI Ready VMware Cloud Foundation environment	Work with the technology team in your organization on configuring the physical servers, network, and storage in the data center. Collect the environment details and write them down in the <a href="#">VMware Cloud Foundation Planning and Preparation Workbook</a> .
Deploy and configure vSphere with Tanzu on VMware Cloud Foundation	<ol style="list-style-type: none"> <li>1 Configure vSphere for AI GPU-Enabled workloads.</li> <li>2 Deploy a Tanzu Kubernetes Grid cluster on the Supervisor for AI ready workloads.</li> <li>3 Deploy and configure NVIDIA Kubernetes Operators.</li> </ol>
VMware Private AI Foundation with NVIDIA	<ol style="list-style-type: none"> <li>1 Create self-service catalog items for Service Broker in VMware Aria Automation.</li> <li>2 Deploy VMware Data Services Manager and configure</li> <li>3 Configure a content library for deep learning VM images.</li> <li>4 If your environment is not connected to the Internet, perform additional configuration required for making deep learning VM images, TKr images, and container images from the NVIDIA NGC catalog available in the environment.</li> <li>5 Deploy AI workloads and deploy vector databases for RAG workloads by using the catalog items in Automation Service Broker.</li> </ol>

## Update History

The *Private AI Ready Infrastructure for VMware Cloud Foundation* validated solution is updated when necessary.

Revision	Description
23 JUL 2024	<ul style="list-style-type: none"> <li>■ This validated solution now supports VMware Cloud Foundation 5.2.0.</li> <li>■ The <code>PowerValidatedSolutions</code> PowerShell module is now version 2.11.0.</li> </ul>
28 MAY 2024	Initial release.

# Design Objectives of Private AI Ready Infrastructure for VMware Cloud Foundation

1

The *Private AI Ready Infrastructure for VMware Cloud Foundation* solution has objectives to deliver prescriptive content about the solution so that it is fast to deploy and is suitable for use in production environments.

Objective	Description
Main objective	Provide enterprise private AI GPU-enabled infrastructure to build an AI platform for data scientists and ML engineers using VMware Cloud Foundation as its foundational layer.
VMware Cloud Foundation architecture support	<ul style="list-style-type: none"><li>■ vSAN ReadyNodes<ul style="list-style-type: none"><li>■ Standard</li><li>■ Single VMware Cloud Foundation instance</li></ul></li></ul>
Workload domain type support	VI Workload domain
Scope of implementation	<ul style="list-style-type: none"><li>■ Configuration of VMware Cloud Foundation components:<ul style="list-style-type: none"><li>■ vCenter Server</li><li>■ VMware NSX</li><li>■ VMware Aria Automation</li><li>■ VMware Aria Operations</li></ul></li><li>■ Deployment and configuration of VMware solution components:<ul style="list-style-type: none"><li>■ vSphere with Tanzu</li></ul></li><li>■ Deployment and configuration of add-on solution components:<ul style="list-style-type: none"><li>■ VMware Private AI Foundation with NVIDIA</li></ul></li></ul>
Scope of guidance	<ul style="list-style-type: none"><li>■ Deployment and initial configuration of the Supervisor in vSphere with Tanzu.</li><li>■ Deployment and initial configuration of Tanzu Kubernetes Grid clusters using Tanzu Kubernetes Grid Service and NVIDIA Kubernetes Operators.</li><li>■ VMware Private AI Foundation with NVIDIA configuration including VMware Aria Automation setup, integration of VMware Data Services Manager, and deep learning VM configuration.</li><li>■ Operations for included components, such as monitoring and alerting, backup and restore, post-maintenance validation, and upgrade.</li></ul>
Cloud type	Private Cloud

Objective	Description
Availability	99%
AI Application Starter Packs	Retrieval Augmented Generation (RAG) Starter Pack

# Detailed Design of Private AI Ready Infrastructure for VMware Cloud Foundation

## 2

The *Private AI Ready Infrastructure for VMware Cloud Foundation* validated solution uses vSphere with Tanzu deployed on top of a VMware Cloud Foundation VI workload domain to transform vSphere into a platform for running GPU-accelerated AI workloads as VMs or Kubernetes workloads natively on the Supervisor. It also provides design for running add-on solutions.

Read the following topics next:

- [Logical Design for Private AI Ready Infrastructure for VMware Cloud Foundation](#)
- [Compute Design for Private AI Ready Infrastructure for VMware Cloud Foundation](#)
- [Physical Infrastructure Resource Design for Private AI Ready Infrastructure for VMware Cloud Foundation](#)
- [Network Design for Private AI Ready Infrastructure for VMware Cloud Foundation](#)
- [Accelerators Design for Private AI Ready Infrastructure for VMware Cloud Foundation](#)
- [Storage Design for Private AI Ready Infrastructure for VMware Cloud Foundation](#)
- [Deployment Specification for Private AI Ready Infrastructure for VMware Cloud Foundation](#)
- [Life Cycle Management for Private AI Ready Infrastructure for VMware Cloud Foundation](#)
- [Information Security and Access Control Design for Private AI Ready Infrastructure for VMware Cloud Foundation](#)

## Logical Design for Private AI Ready Infrastructure for VMware Cloud Foundation

The logical design consists of multiple elements which you can use to deploy and manage infrastructure used to run AI and GPU-enabled high performance workloads.

AI infrastructure is comprised of hardware and software elements essential for fine-tuning and serving (inferencing) models and other AI workloads. This includes specialized processors such as GPUs (hardware), as well as optimization and deployment tools (software), all managed by VMware Cloud Foundation. The software components within VMware Cloud Foundation provide the functionality to host compute, network, and storage for virtual machines and container-based workloads using vSphere with Tanzu, NSX, and vSAN.

You activate and configure vSphere with Tanzu on your shared edge and workload vSphere cluster in the VI workload domain. NSX Edge nodes provide load balancing, north-south connectivity, and all required networking for the Kubernetes services. The ESXi hosts in the shared edge and workload vSphere cluster are prepared as NSX transport nodes to provide distributed routing and firewall services to your customer workloads as well as GPU resources by using multiple technologies like NVIDIA vGPU or MIG.

Private AI ready infrastructure environment consists of multiple elements.

### **ESXi Host Vendor Components**

Vendor-provided ESXi components, such as VIB files, are used to enable GPUs and for communication channels for metrics and detailed information of their devices.

### **Supervisor**

The Supervisor is a unique kind of Kubernetes cluster that uses ESXi hosts as worker nodes through an additional process called Spherelet that is created on each host.

### **Harbor Supervisor Service**

Harbor is deployed as a service in a Supervisor.

### **VM Service**

The VM Service in vSphere with Tanzu enables DevOps engineers to deploy and run VMs, in addition to containers, in a shared Kubernetes environment. Both containers and VMs share the same vSphere namespace resources and can be managed from a single vSphere with Tanzu interface.

### **Tanzu Kubernetes Grid Service**

The Tanzu Kubernetes Grid Service deploys Tanzu Kubernetes Grid clusters as Photon OS appliances on top of the Supervisor.

### **Tanzu Kubernetes Grid Cluster**

Upstream-compliant Kubernetes cluster to run container workloads.

### **Tanzu vSphere Pods**

The Container Runtime for ESXi (CRX) uses the same hardware virtualization techniques as VMs and it has a VM boundary around it. A direct boot technique is used, which allows the Linux guest of CRX to initiate the main init process without passing through kernel initialization. This allows vSphere Pods to boot nearly as fast as containers, allowing standalone VMs to run on top of the Supervisor Cluster.

### **Kubernetes Operators**

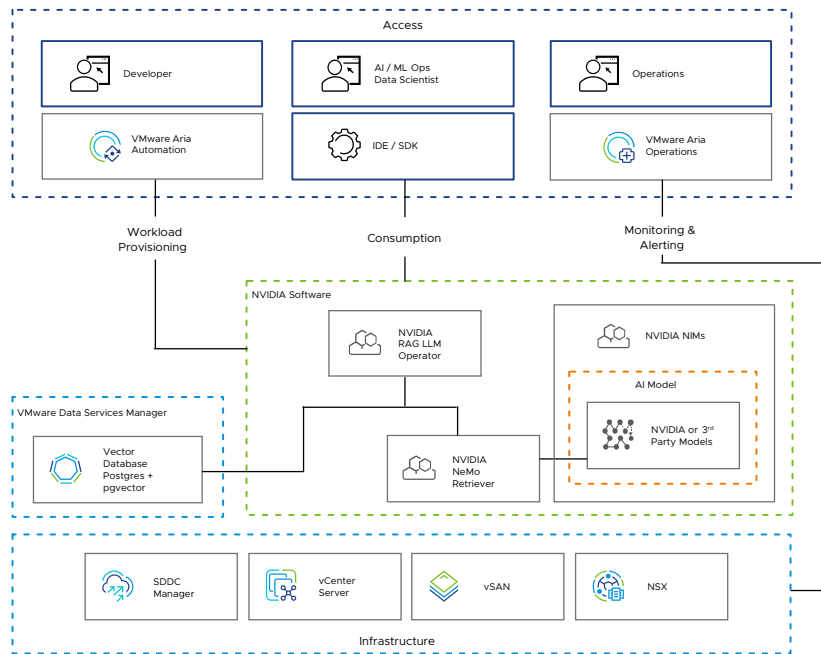
Kubernetes Operators on the Tanzu Kubernetes Clusters are used to automate the installation of drivers for GPU and high-speed interconnect devices.

### **VMware Data Services Manager**

VMware Data Services Manager offers database management in vSphere. It supports on-demand provisioning and automated management of PostgreSQL and MySQL databases in a vSphere environment.

Private AI ready infrastructure serves as the foundational building blocks based on VMware Cloud Foundation for running joint private AI solutions with partners such as NVIDIA.

**Figure 2-1. SDDC Logical Design for a Private AI Ready VI Workload Domain**



## Compute Design for Private AI Ready Infrastructure for VMware Cloud Foundation

Hardware infrastructure requirements for AI workloads depend on the specific task, dataset size, model complexity, or performance expectations.

The following example configuration provides optimal configuration for fine-tuning and serving large language models (LLMs) which matches with NVIDIA DGX solutions. Because the requirements of your organization might be different, contact your OEM to determine the proper solution.

Table 2-1. Example of Optimal Compute Hardware Choices

Category	Hardware	Description	Example Optimal Configuration (Based on NVIDIA DGX)
CPU	Intel <a href="#">VMware Compatibility Guide - Intel Xeon</a>	Latest Intel Xeon 4th Gen (Sapphire Rapids) recommended, 3rd Gen (Ice Lake) acceptable, with a balance between CPU Frequency and number of cores. The latest Intel Gen offers advanced features related to AI/ML such as Intel AMX (Advanced Matrix Extensions) and support for DDR5 and CXL (Compute Express Link). Use Peripheral Component Interconnect Express (PCIe) Gen5 (recommended) or PCIe Gen4 (acceptable) for faster interconnects.	2 x Intel Xeon (Sapphire Rapids or later)
	AMD EPYC <a href="#">VMware Compatibility Guide - AMD EPYC</a>	Latest AMD EPYC 4th Gen (Genoa) recommended, 3rd Gen (Milan) acceptable with a balance between CPU Frequency and number of cores. EPYC CPUs offer a high core count, exceptional memory bandwidth, and support for multi-socket configurations. They are suitable for both AI/ML and LLM workloads. Use PCIe Gen5 (recommended) or PCIe Gen4 (acceptable) for faster interconnects, .	2 x AMD EPYC (Genoa or later)



Table 2-1. Example of Optimal Compute Hardware Choices (continued)

Category	Hardware	Description	Example Optimal Configuration (Based on NVIDIA DGX)
Memory	DDR5	Faster memory with higher bandwidth can reduce data transfer bottlenecks and enable faster access to the large datasets involved in AI/ML tasks. Additionally, the increased memory density provided by DDR5 allows for larger models and more extensive training datasets to be stored in memory, which can improve the overall performance and efficiency of AI/ML algorithms.	2 TB RAM per node, according to the configuration
GPU	NVIDIA: H100, H100 NVL, A100, L40s <a href="#">VMware Compatibility Guide - GPUS</a>	NVIDIA GPUs with compute capacity greater or equal to 8.0 are essential for LLM training. The support for bfloat16 in these GPUs balances precision and range, aiding in training neural networks efficiently without losing accuracy. NVLink enables efficient GPU-to-GPU communication and memory sharing, while NVSwitch enables large-scale GPU collaboration across multiple servers, facilitating the training and deployment of advanced AI models on very big datasets.	<ul style="list-style-type: none"> <li>■ 8 x NVIDIA H100 GPUs (80 GB) for models above 40B parameters</li> <li>■ 4 x NVIDIA H100 GPUs (80 GB) for models less than 40B parameters</li> </ul>

**Table 2-2. Design Decisions for Compute Configuration for Private AI Infrastructure for VMware Cloud Foundation**

Decision ID	Design Decision	Design Justification	Design Implication
AIR-COMPUTE-001	Select servers with CPUs with a high number of cores.	To optimize computational efficiency and minimize the need for scaling out by adding more nodes, consider scaling up the CPU core count in each server. By choosing CPUs with a high number of cores, you can effectively handle multiple inference threads simultaneously. This approach maximizes hardware utilization and enhances the capacity to manage parallel tasks, leading to improved performance and resource utilization in inference workloads	High-end CPUs might increase the overall cost of the solution.
AIR-COMPUTE-002	Select a fast-access memory.	Minimal latency for data retrieval is crucial for real-time inference applications. Increased latency reduces inference performance and give a poor user experience.	Re-purposing available servers might not be a feasible option and overall cost of the solution might increase.
AIR-COMPUTE-003	Select CPUs with Advanced Vector Extensions (AVX, AVX2, or AVX-512).	CPUs with support for AVX or AVX2 can improve performance in deep learning tasks by accelerating vector operations.	Re-purposing available servers might not be a feasible option and overall cost of the solution might increase.

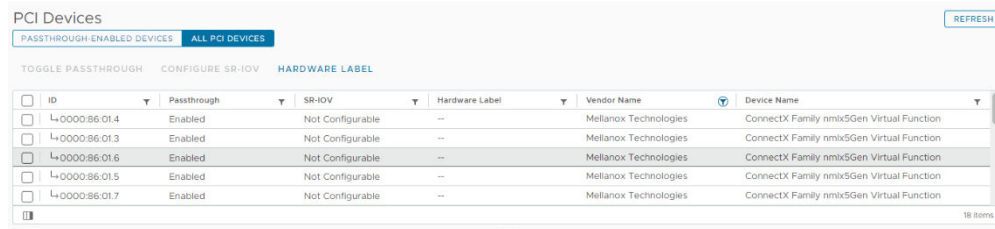
## Physical Infrastructure Resource Design for Private AI Ready Infrastructure for VMware Cloud Foundation

The design of physical infrastructure resources for private AI ready infrastructure for VMware Cloud Foundation include hardware components and features to support AI workloads on a virtualized platform.

### Global SR-IOV

Single Root I/O Virtualization (SR-IOV) is a specification that allows a single PCIe physical device under a single root port to appear as multiple separate Virtual Functions to the hypervisor or the guest operating system.

SR-IOV is required for this validated solution to enable virtual functions from a NIC and HBA and is also required for both MIG and time-slicing modes of NVIDIA vGPU. For MIG, a vGPU is associated with a virtual function at boot time. For example, for a Mellanox ConnectX-6 NIC, you have the following virtual functions available:

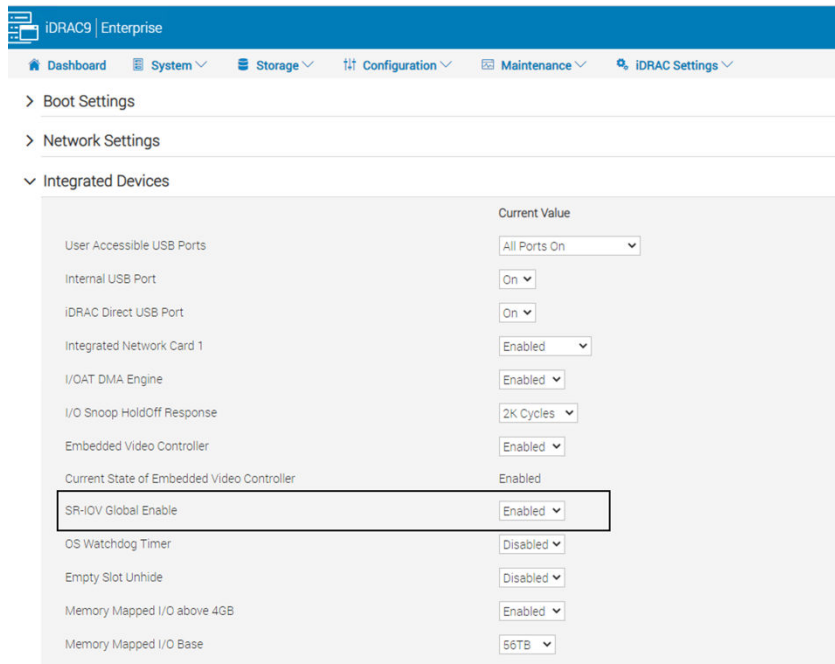


The screenshot shows the 'PCI Devices' page in the vSphere Web Client. It has tabs for 'PASSTHROUGH-ENABLED DEVICES' and 'ALL PCI DEVICES'. Below the tabs are links for 'TOGGLE PASSTHROUGH', 'CONFIGURE SR-IOV', and 'HARDWARE LABEL'. A table lists five PCI devices, all of which are Mellanox Technologies ConnectX Family nmlx5Gen Virtual Functions. The 'SR-IOV' column for all devices is 'Not Configurable'.

ID	Passthrough	SR-IOV	Hardware Label	Vendor Name	Device Name
0000:86:01.4	Enabled	Not Configurable	--	Mellanox Technologies	ConnectX Family nmlx5Gen Virtual Function
0000:86:01.3	Enabled	Not Configurable	--	Mellanox Technologies	ConnectX Family nmlx5Gen Virtual Function
0000:86:01.6	Enabled	Not Configurable	--	Mellanox Technologies	ConnectX Family nmlx5Gen Virtual Function
0000:86:01.5	Enabled	Not Configurable	--	Mellanox Technologies	ConnectX Family nmlx5Gen Virtual Function
0000:86:01.7	Enabled	Not Configurable	--	Mellanox Technologies	ConnectX Family nmlx5Gen Virtual Function

For information on how to enable SR-IOV at the BIOS/UEFI level, see the documentation from your server vendor. For example, you set Global SR-IOV by using Dell iDRAC in the following way:

**Figure 2-2. Example of Configuring SR-IOV**



## NVIDIA GPUDirect RDMA, NVLink, and NVSwitch

The combination of GPUDirect RDMA (Remote Direct Memory Access), NVLINK, and NVSwitch are suitable for generative AI.

- GPUDirect RDMA enables direct GPU-to-GPU memory access through network devices, efficiently reducing latency and amplifying data sharing capabilities.
- NVLINK serves as a high-speed, low-latency bridge between GPUs, suitable for managing extensive datasets.

- NVSwitch orchestrates communication in multi-GPU configurations, offering a foundation for scaling generative AI.

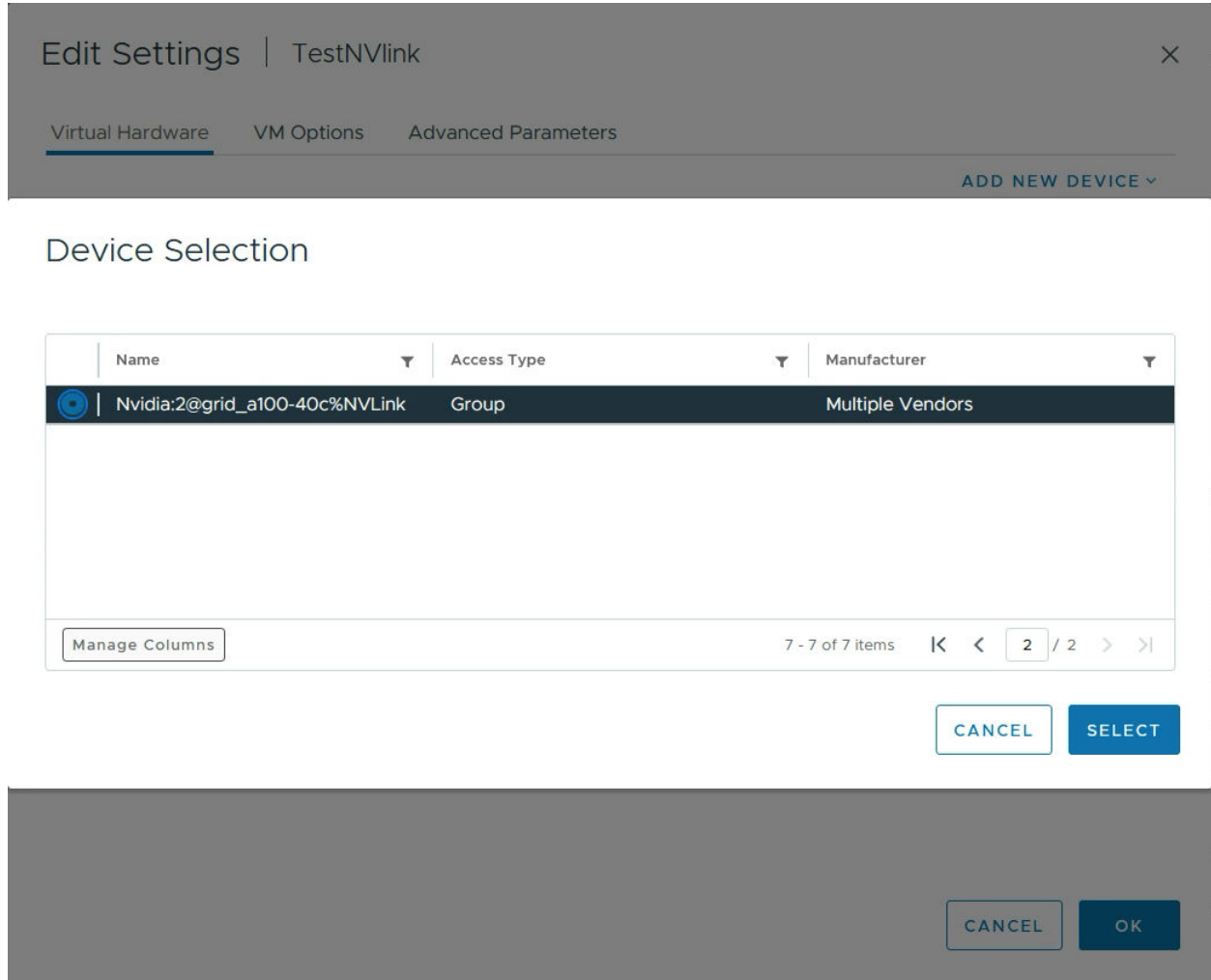
## GPUDirect RDMA on VMware vSphere

GPUDirect on vSphere works by using the NVIDIA vGPU technology and RDMA-capable network adapters like NVIDIA ConnectX-6. For virtual machines, to enable peer-to-peer communication between the PCIe devices on the same PCIe Root Complex, relax Access Control Services (ACS) in the virtual machine configuration settings and configure NUMA (Non-Uniform Memory Access) affinity or device groups. Device groups are supported with Tanzu Kubernetes Grid Service and you can configure the ACS relax setting by using advanced parameters in a VM class.

## Using Device Groups in vSphere with NVIDIA GPUs and NVLink

In vSphere 8, device groups simplify the use of virtual machines with complementary hardware devices. These groups can include hardware devices connected via a common PCIe switch or direct interconnect. They are identified at the hardware level and presented to vSphere as a unified entity, which can be added to virtual machines and VM classes by using the standard workflow for adding a PCI device. Both vSphere DRS and vSphere HA recognize device groups and ensure virtual machine placement aligns with device group requirements.

Figure 2-3. Device Group



A device group can comprise at least two GPUs connected by NVIDIA NVLink, enhancing inter-GPU communication vital for ML/AI performance. NVLinks enable GPUs to share high bandwidth memory (HBM), facilitating larger machine learning models. NVLink can be programmatically activated or deactivated to form new device groups.

Typically, machine learning models are trained with Floating-Point 32-bit precision due to its high numerical precision, which helps in accurately representing complex mathematical computations during training. However, when it comes to deploying these models for inference tasks, especially in production environments, there's often a need to optimize for computational efficiency and resource utilization, this is achieved by converting to lower precisions such as Floating-Point 16-bit, Brain Floating Point 16 (BF16), or 8-bit Integer Precision (Int8) for inference deployment. This conversion reduces model size and computational demands, enabling deployment on fewer GPUs for inference as well as faster inferencing times.

# Network Design for Private AI Ready Infrastructure for VMware Cloud Foundation

Private AI ready infrastructure requires multiple networks. The network design includes choosing the physical network devices and creating physical network setup for running AI workloads.

The requirement for network devices for AI workloads depend on the specific task, dataset size, model complexity, or performance expectations.

**Table 2-3. Recommended Physical Network Devices**

Category	Hardware	Description	Example of Optimal Configuration (Based on NVIDIA DGX)
Management network	<a href="#">VMware Compatibility Guide - NICs</a>	<ul style="list-style-type: none"> <li>■ 10 Gbps, 25 Gbps, or above.</li> <li>■ Host baseboard management controller (BMC) with RJ45.</li> </ul>	<ul style="list-style-type: none"> <li>■ NIC  Broadcom 57504, Mellanox ConnectX-4, or Intel similar products</li> <li>■ Switch  <a href="#">Broadcom StrataXGS Switch Solutions</a> BCM56080 Series or similar products</li> </ul>
Workloads and VMware services such as vSphere vMotion, vSAN, network overlays, and so on.	<a href="#">VMware Compatibility Guide – NICs with SR-IOV and RoCE v2</a>	<p>LLM inference and fine-tuning within a single host is sufficient with standard 2*25 Gbps Ethernet ports.</p> <p>For fine-tuning models larger than 40B parameters, efficient multi-node communication requires low latency, and optimal performance requires 100 Gbps or higher RDMA network (for example, RoCE or InfiniBand).</p>	<ul style="list-style-type: none"> <li>■ RDMA over Converged Ethernet (RoCE) NIC  Broadcom 5750X, NVIDIA Mellanox ConnectX-5/6/7, or similar products</li> <li>■ RoCE Switch  Broadcom StrataXGS Switch Solutions (Trident4-X11C/BCM56890 Series) or similar products</li> <li>■ InfiniBand Host Channel Adapter (HCA)  NVIDIA Mellanox ConnectX-5/6/7 VPI</li> <li>■ InfiniBand Switch  NVIDIA QM9700</li> </ul>

**Note** For optimal speed of data transfer for the devices, during the installation of NICs or HCAs on servers, consider PCIe generation and lane compatibility on the server motherboard.

## Management Domain VLANs

For the management network, each network type is associated with a specific VLAN. See [VMware Cloud Foundation Design Guide](#).

## Workload Domain Network VLANs

The workload network on the workload cluster is configured with dedicated switch and network adapters for optimal performance. You deploy all vSphere with Tanzu workloads to overlay-backed NSX segments. NSX Edge nodes in the shared edge and workload vSphere cluster are deployed to VLAN-backed port groups.

**Figure 2-4. Networks for vSphere with Tanzu in a Workload Domain**

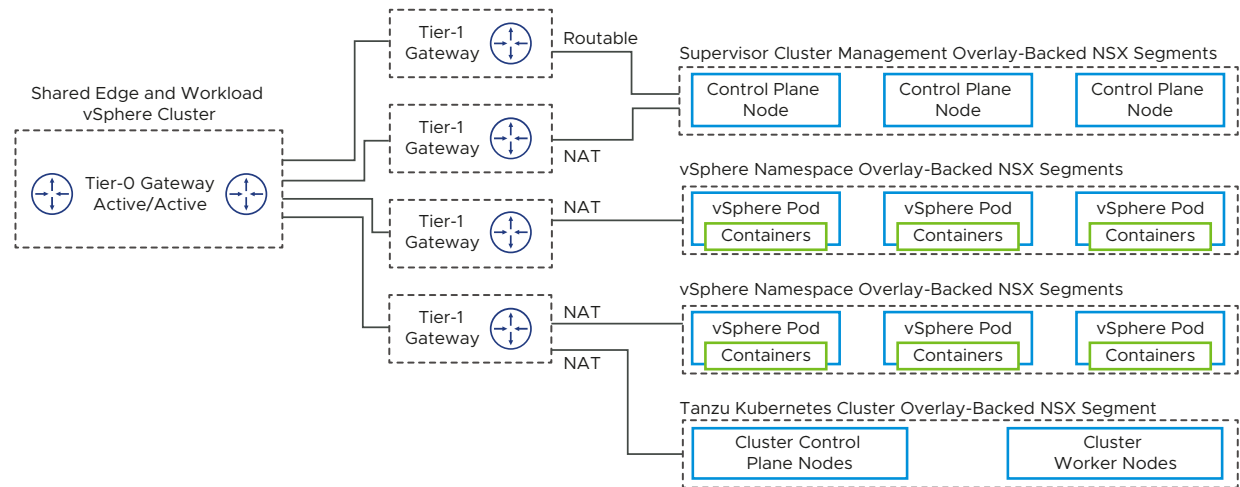


Table 2-4. Networks Used by vSphere with Tanzu

Network	Routable / NAT	Usage
Supervisor Control Plane network	Routable	Used by the Supervisor control plane nodes.
Pod Networks	NAT	<p>Used by Kubernetes pods that run in the cluster. Any Tanzu Kubernetes Grid Clusters instantiated in the Supervisor also use this pool.</p> <ul style="list-style-type: none"> <li>■ For LLM inferencing tasks and fine-tuning tasks within a single host, existing 25-Gb Ethernet network infrastructure is sufficient to accommodate the bandwidth requirements of textual data.</li> <li>■ For fine-tuning larger models with more than 40B parameters among GPUs across different nodes, the substantial demand for information exchange (including weights) requires the adoption of RDMA networking (RoCE/ InfiniBand) with 100 Gb or higher bandwidth for optimal performance.</li> </ul>
Service IP Pool Network	NAT	Used by Kubernetes applications that need a service IP address.
Ingress IP Pool Network	Routable	Used by NSX to create an IP pool for load balancing.
Egress IP Pool Network	Routable	Used by NSX to create an IP pool for NAT endpoint use.
Namespace Networks	NAT	When you create a namespace, a /28 overlay-backed NSX segment and corresponding IP pool is instantiated to service pods in that namespace. If that IP space runs out, an additional /28 overlay-backed NSX segment and IP pool are instantiated.
Tanzu Kubernetes Grid Networks	NAT	When you create a Tanzu Kubernetes Grid cluster, an NSX Tier-1 Gateway is instantiated in NSX. On that NSX Tier-1 Gateway, a /28 overlay-backed NSX segment and IP pool is also instantiated.



## Design Decisions on the Network Design for Private AI Ready Infrastructure

Table 2-5. Design Decisions on Networking for Private AI Ready Infrastructure for VMware Cloud Foundation

Decision ID	Design Decision	Design Justification	Design Implication
AIR-TZU-NET-001	Set up networking for 100 Gbps or higher if possible.	100 Gbps networking provides enough bandwidth and very low latency for inference and fine-tuning use cases backed by vSAN ESA.	The cost of the solution is increased.
AIR-TZU-NET-002	Add a /28 overlay-backed NSX segment for use by the Supervisor control plane nodes.	Supports the Supervisor control plane nodes.	You must create the overlay-backed NSX segment.
AIR-TZU-NET-003	Use a dedicated /20 subnet for pod networking.	A single /20 subnet is sufficient to meet the design requirement of 2000 pods.	You must set up a private IP space behind a NAT that you can use in multiple Supervisors.
AIR-TZU-NET-004	Use a dedicated /22 subnet for services.	A single /22 subnet is sufficient to meet the design requirement of 2000 pods.	Private IP space behind a NAT that you can use in multiple Supervisors.
AIR-TZU-NET-005	Use a dedicated /24 or larger subnet on your corporate network for ingress endpoints.	A /24 subnet is sufficient to meet the design requirement of 2000 pods in most cases.	This subnet must be routable to the rest of the corporate network. A /24 subnet will be sufficient for most use cases, but you should evaluate your ingress needs before deployment.
AIR-TZU-NET-006	Use a dedicated /24 or larger subnet on your corporate network for egress endpoints.	A /24 subnet is sufficient to meet the design requirement of 2000 pods in most cases.	This subnet must be routable to the rest of the corporate network. A /24 subnet will be sufficient for most use cases, but you should evaluate your egress needs before to deployment.

## Accelerators Design for Private AI Ready Infrastructure for VMware Cloud Foundation

Designing a system for inference with LLMs requires consideration of the hardware, particularly GPUs and potentially CPUs.

vSphere optimizes the parallel processing architecture of GPUs for AI computing in virtualized environments, such as deep learning, which involve intensive mathematical operations on large datasets. In vSphere, you can distributed tasks across multiple GPU cores, relying on parallelization to improve GPU performance and streamline inference processes. vSphere also optimizes GPU use by supporting direct access to GPU resources from virtual machines through technologies, such as NVIDIA vGPU (time-sliced or MIG) or VMDirect Path I/O.

Consider these guidelines:

- **Large Memory Capacity:** To accommodate the size of LLMs, choose GPUs with large memory capacities. For instance, models like Mistral-7b served by vLLM might consume almost 37 GB of VRAM. Larger models like Mistral 8×7B might require 7-8 times more VRAM.
- **GPUs that support Brain Floating Point 16 (BF16)** are recommended because they provide an optimal balance between performance and precision. You can use BF16 for faster computations for the extensive processing demands of large language models. You can achieve quicker training and inference times without a significant sacrifice in accuracy.

**Table 2-6. Examples GPUs for Inference**

GPU	Architecture	Memory	Usage
NVIDIA GPUs NVIDIA H100 Tensor Core GPU	Hopper	Up to 80 GB HBM3	<ul style="list-style-type: none"> <li>■ Upper range of LLM sizes exceeding the 30 billion parameters range</li> <li>■ High-performance inference with large batch sizes</li> </ul>
NVIDIA A100 Tensor Core GPU	Ampere	40 GB or 80 GB HBM2e	<ul style="list-style-type: none"> <li>■ High-performance inference</li> <li>■ Middle-range LLMs (between 7 and 13 billion parameters) and embedding models</li> <li>■ Versatile deployment options</li> </ul>
NVIDIA L40 GPU	Ada Lovelace	48 GB GDDR6X	<ul style="list-style-type: none"> <li>■ Middle-range LLMs (between 7 and 13 billion parameters) and embedding models</li> <li>■ Workstations and edge deployments</li> <li>■ Real-time inference with lower power consumption</li> </ul>

**Table 2-7. Design Decisions on Accelerators for Private AI Ready Infrastructure for VMware Cloud Foundation**

Decision ID	Design Decision	Design Justification	Design Implication
AIR-ACCELERATE-001	Select GPUs with high memory bandwidth	AI workloads require high memory bandwidth to efficiently handle large amounts of data. Look for GPUs with high memory bandwidth specifications.	<ul style="list-style-type: none"> <li>■ The cost of the solution is increased.</li> <li>■ GPU choice might be limited.</li> </ul>
AIR-ACCELERATE-002	Select GPUs with large memory capacity.	To handle efficiently LLMs, select GPUs equipped with substantial memory capacities. LLMs containing billions of parameters demand significant GPU memory resources for model fine-tuning and inference.	<ul style="list-style-type: none"> <li>■ The cost of the solution is increased.</li> <li>■ GPU choice might be limited.</li> </ul>
AIR-ACCELERATE-003	Evaluate and compare compute performance of the available options of GPUs.	Assess the GPU's compute performance based on metrics like CUDA cores (for NVIDIA GPUs) or stream processors (for AMD GPUs). Higher compute performance provide support for faster model training and inference, particularly beneficial for complex AI tasks.	<ul style="list-style-type: none"> <li>■ The cost of the solution is increased.</li> <li>■ GPU choice might be limited.</li> </ul>
AIR-ACCELERATE-004	Evaluate cooling and power efficiency of GPUs.	To manage the strain large language models place on GPUs, prioritize systems with efficient cooling and power management to mitigate high power consumption and heat generation.	You must select server platforms focused on GPU.

## Storage Design for Private AI Ready Infrastructure for VMware Cloud Foundation

This section discusses the storage design for Private AI Ready Infrastructure for VMware Cloud Foundation.

Storage design for Private AI Ready Infrastructure for VMware Cloud Foundation includes the vSAN design for principal and supplemental storage.

For design information on supported storage types in VMware Cloud Foundation, see [VMware Cloud Foundation Design Guide](#).

## vSAN ESA for AI Workloads

vSAN ESA (Express Storage Architecture) is optimized for performance through several mechanisms and architectural enhancements, making it suitable for high-performance workloads, such as AI, when deployed with 100 Gbps networking and faster. vSAN ESA also supports RDMA over Converged Ethernet (RoCE v2). When configured on compatible switches and NICs, RoCE v2 can significantly reduce host CPU utilization and enhance performance. vSAN over RoCE can efficiently handle high-throughput and low-latency data transfers, making it well-suited for demanding AI applications.

Storage platforms must provide high capacity, high performance, substantial bandwidth, and minimal latency to efficiently support the stages of generative AI workflows, including data ingestion, preparation, fine-tuning, and inference. Fine-tuning generative AI models, particularly LLMs with billions of parameters and numerous intermediate outputs, requires considerable performance and storage capacity. Additionally, the storage requirements during AI model inference fluctuate according to the specific needs of the AI model and the deployment environment's characteristics. vSAN ESA provides the essential operating system and container storage capabilities to manage both GPU-resident models and data stored outside GPU memory with the required performance.

As one of the popular use case for inference, vector databases are an important component of RAG systems. As RAG systems accumulate more data over time, the size of the vector database grows. This growth is influenced by factors such as the frequency of user queries, the diversity of content being indexed, and the rate of new data ingestion. You can store such vector databases on vSAN clusters in a VI workload domain, mitigating noisy neighbor events and resource contention with productive inference workloads.

Storing vector databases on vSAN ESA clusters also benefits from the advanced compression available with vSAN ESA, enabled by default. By compressing data at the top of the storage stack, all data written to other hosts in the cluster is transmitted across the network in a compressed state, optimizing storage efficiency and network bandwidth utilization.

## Providing Read/Write Shared Persistent Volumes for Containerized Workloads with vSAN ESA

Tanzu Kubernetes Grid clusters can use ReadWriteMany persistent volumes backed by vSAN File Service for model repositories, model versioning and management, model ensembles, and the storage and archiving of inference data. vSphere with Tanzu uses cloud native storage (CNS) file volumes backed by vSAN file shares for ReadWriteMany persistent volumes.

To use vSAN shares, you set up vSAN File Service on the vSAN datastore and activate file volume support on the Supervisor.

When using vSAN File Service for ReadWriteMany volumes as shared data repositories, consider the following limits:

- A maximum of 100 file shares per vSAN cluster.
- The maximum size of a file share is equal to the maximum available capacity of the vSAN cluster.

For more considerations regarding vSAN File Services, see [Limitations and Considerations of vSAN File Service](#) in the *Administering VMware vSAN* documentation.

**Note** External storage solutions can be leveraged as supplemental storage for the VI workload domain. You can use S3-compatible object storage and NFS exports, provided directly to containers and virtual machines together with vSAN for shared datasets, model repositories, and archival purposes. This approach is particularly beneficial for highly scalable architectures. For more detailed guidance and best practices, see the documentation from your storage vendor.

## Design Decisions on Storage Design for Private AI Infrastructure

Table 2-8. Design Decisions on Storage for Private AI Ready Infrastructure for VMware Cloud Foundation

Decision ID	Design Decision	Design Justification	Design Implication
AIR-STORAGE-001	Use vSAN ESA with 100 Gbps networking and, if possible, RDMA.	Provides high performance and efficiency. Although the minimum bandwidth for vSAN ESA is 25 Gbps, 100 Gbps and faster provides the best performance in terms of bandwidth and latency for all AI use cases.	<ul style="list-style-type: none"> <li>■ The cost of the solution is increased.</li> <li>■ RDMA increases the design complexity.</li> <li>■ The choice of vSAN ReadyNodes is limited to nodes that are approved for use with vSAN ESA.</li> </ul>
AIR-STORAGE-002	Use vSAN ESA RAID 5 or RAID 6 erasure coding.	Provides performance equal to RAID 1 mirroring.	None.
AIR-STORAGE-003	Leave data compression enabled for vSAN ESA.	Enables transmitting data in compressed state across hosts in the cluster. Data compression in vSAN ESA is controllable using storage policies.	None.

## Deployment Specification for Private AI Ready Infrastructure for VMware Cloud Foundation

When vSphere with Tanzu is activated on a vSphere cluster running in a VI workload domain, a Kubernetes control plane is instantiated by using Photon OS virtual machines. This layer contains multiple objects that activate the capability to run Kubernetes workloads natively in the ESXi hosts, instantiating the Supervisor.

## Deployment Model for Private AI Ready Infrastructure for VMware Cloud Foundation

You determine the use of the different services, the sizing of those resources, and how they are deployed and managed based on the design objectives for the *Private AI Ready Infrastructure for VMware Cloud Foundation* validated solution.

### vSphere Storage Based Policy Management Configuration

You must configure a datastore with the activation requirements before activating a Supervisor. The Supervisor configuration requires the use of vSphere Storage Policy Based Management (SPBM) policies for control plane nodes, ephemeral disks, and image cache. These policies correlate to Kubernetes storage policies that can be assigned to vSphere Namespaces. These policies are consumed in a Supervisor or a Tanzu Kubernetes Grid cluster.

**Table 2-9. Design Decisions on vSphere Storage Policy Based Management for Private AI Ready Infrastructure for VMware Cloud Foundation**

Decision ID	Design Decision	Design Justification	Design Implication
AIR-SPBM-CFG-001	Create a vSphere tag and tag category, and apply the vSphere tag to the vSAN datastore in the shared edge and workload vSphere cluster in the VI workload domain.	Supervisor activation requires the use of vSphere Storage Based Policy Management (SPBM).  To assign the vSAN datastore to the Supervisor, you need to create a vSphere tag and tag category to create an SPBM rule.	You must perform this operation manually or by using PowerCLI.
AIR-SPBM-CFG-002	Create a vSphere Storage Policy Based Management (SPBM) policy that specifies the vSphere tag you created for the Supervisor.	When you create the SPBM policy and define the vSphere tag for the Supervisor, you can then assign that SPBM policy during Supervisor activation.	You must perform this operation manually or by using PowerCLI.

### Supervisor

A vSphere cluster that is activated for vSphere with Tanzu is called a Supervisor. After a Supervisor is instantiated, a vSphere administrator can create vSphere Namespaces. Developers can run modern applications that consist of containers running inside vSphere Pods and create Tanzu Kubernetes clusters when upstream Kubernetes compliant clusters are required.

The Supervisor uses ESXi hosts as worker nodes. This is achieved by using an additional process, Spherelet, that is created on each host. Spherelet is a kubelet that is ported natively to the ESXi host and allows the host to become part of the Kubernetes cluster.

You can use vSphere zones and multi-zone Supervisor architecture to implement high availability at the vSphere cluster level for your workloads running within a Tanzu Kubernetes Grid cluster.

**Table 2-10. Design Decisions on the Supervisor for Private AI Ready Infrastructure for VMware Cloud Foundation**

Decision ID	Design Decision	Design Justification	Design Implication
AIR-TZU-CFG-001	Activate vSphere with Tanzu on the shared edge and workload vSphere cluster in the VI workload domain.	The Supervisor is required to run Kubernetes workloads natively and to deploy Tanzu Kubernetes Grid clusters natively using Tanzu Kubernetes Grid Service.	Ensure the shared edge and workload vSphere cluster is sized to support the Supervisor control plane, any additional integrated management workloads, and any customer workloads.
AIR-TZU-CFG-002	Deploy the Supervisor with small-size control plane nodes.	Deploying the control plane nodes as small-size appliances gives you the ability to run up to 2,000 pods within your Supervisor.  If your pod count is higher than 2,000 for the Supervisor, you must deploy control plane nodes that can handle that level of scale.	You must consider the size of the control plane nodes.
AIR-TZU-CFG-003	Use NSX as provider of the software-defined networking for the Supervisor.	You can deploy a Supervisor either by using NSX or vSphere networking .  VMware Cloud Foundation uses NSX for software-defined networking across the SDDC. Deviating for vSphere with Tanzu would increase the operational overhead.	None.
AIR-TZU-CFG-004	Deploy the NSX Edge cluster with large-size nodes.	Large-size NSX Edge nodes are the smallest size supported to activate a Supervisor.	You must account for the size of the NSX Edge nodes.
AIR-TZU-CFG-005	Deploy a single-zone Supervisor.	A three-zone Supervisor requires three separate vSphere clusters.	No change to existing design or procedures with single-zone Supervisor.

## Harbor Supervisor Service

To use Harbor with vSphere with Tanzu, you deploy it as a Supervisor Service. Before you install Harbor as a service, you must install Contour.

All Tanzu Kubernetes Grid clusters running on the host Supervisor trust the Harbor Registry running as a Supervisor Service by default. Tanzu Kubernetes Grid clusters running on Supervisors different from the Supervisor where Harbor is installed must have network connectivity to Harbor. These Tanzu Kubernetes Grid clusters must be able to resolve the Harbor FQDN and establish trust with the Harbor Registry.

**Table 2-11. Design Decisions on the Harbor Supervisor Service for AI Ready Infrastructure for VMware Cloud Foundation**

Decision ID	Design Decision	Design Justification	Design Implication
AIR-HRB-CFG-001	Deploy Contour as an Ingress Supervisor Service.	Harbor requires Contour on the target Supervisor to provide Ingress Service. The Ingress IP address provided by Contour must be resolved to the Harbor FQDN.	None.
AIR-HRB-CFG-002	Deploy the Harbor Registry as a Supervisor Service.	Harbor as a Supervisor Service has replaced the integrated registry in previous vSphere versions.	You must provide the following configuration: <ul style="list-style-type: none"> <li>■ Harbor FQDN</li> <li>■ Record and Pointer Record (PTR) for the Harbor Registry IP (this IP is provided by the Contour Ingress Service)</li> <li>■ Manage Supervisor Services privilege in vCenter Server.</li> </ul>

## Tanzu Kubernetes Grid Cluster

A Tanzu Kubernetes Grid cluster is a full distribution of the open-source Kubernetes container orchestration software that is packaged, signed, and supported by VMware. Tanzu Kubernetes clusters are provisioned by the VMware Tanzu™ Kubernetes Grid™ Service in the Supervisor. The cluster consists of at least one control plane node and at least one worker node. The Tanzu Kubernetes Grid Service deploys the clusters as Photon OS appliances on top of the Supervisor. You determine the deployment parameters (the size and the number of control plane and worker nodes, Kubernetes distribution version, etc.) to be deployed by using a YAML definition through `kubectl`.

You can provide high-availability to Tanzu Kubernetes Grid clusters when they are deployed on a three vSphere Zone Supervisor. A vSphere zone maps to a vSphere cluster, which means that when you deploy a Supervisor on three vSphere zones, it uses the resources of all three underlying vSphere clusters. This architecture protects your Kubernetes workloads running on Tanzu Kubernetes Grid clusters from failure at a vSphere cluster level. In a single-zone deployment, high availability for Tanzu Kubernetes Grid clusters is provided on an ESXi host level by vSphere HA.



**Table 2-12. Design Decisions on the Tanzu Kubernetes Grid Cluster for Private AI Ready Infrastructure for VMware Cloud Foundation**

Decision ID	Design Decision	Design Justification	Design Implication
AIR-TZU-CFG-006	Deploy a Tanzu Kubernetes Grid Cluster in the Supervisor.	For applications that require upstream Kubernetes compliance, a Tanzu Kubernetes Grid Cluster is required.	None.
AIR-TZU-CFG-007	For a disconnected environment, configure a local content library for Tanzu Kubernetes releases (TKRs) for use in the shared edge and workload vSphere cluster.	In a disconnected environment, the Supervisor is unable to pull TKR images from the central public content library maintained by VMware. To deploy a Tanzu Kubernetes Grid on a Supervisor, you must configure a content library in the shared edge and workload vSphere cluster with the required images, downloaded from the public library.	You must manually configure the content library.
AIR-TZU-CFG-008	Use Antrea as the container network interface (CNI) for your Tanzu Kubernetes Grid clusters.	Antrea is the default CNI for Tanzu Kubernetes Grid clusters.	New Tanzu Kubernetes Grid clusters are deployed with Antrea as the CNI, unless you specify Calico.

## Sizing Compute and Storage Resources for Private AI Ready Infrastructure for VMware Cloud Foundation

Consider compute and storage requirements when sizing the necessary resources for the validated solution.

You size the compute and storage requirements for the vSphere with Tanzu management workloads, Tanzu Kubernetes Grid cluster management workloads, NSX Edge nodes, and GPU-enabled workloads deployed on vSphere, VM Service in the Supervisor, or a Tanzu Kubernetes Grid cluster.

**Table 2-13. Compute and Storage Resource Requirements for vSphere with Tanzu**

Virtual Machine	Nodes	Total vCPUs	Total Memory	Total Storage
Supervisor with small-size control plane for up to 10 workloads	3	12	48 GB	200 GB
Harbor Supervisor Service	N/A	7	7 GB	200 GB
NSX Edge nodes (large nodes)	Minimum of 2	16	64 GB	400 GB

**Table 2-14. Design Decisions on Sizing the Tanzu Kubernetes Grid Cluster for Private AI Ready Infrastructure for VMware Cloud Foundation**

Decision ID	Design Decision	Design Justification	Design Implication
AIR-TZU-CFG-009	Deploy Tanzu Kubernetes Grid clusters with a minimum of three control plane nodes.	Deploying three control plane nodes ensures the control plane state of your Tanzu Kubernetes Grid cluster stays if a node failure occurs.  Horizontal and vertical scaling of the control plane is supported. See <a href="#">Scale a TKG Cluster on Supervisor Using Kubectrl</a> .	None.
AIR-TZU-CFG-010	For production environments, deploy Tanzu Kubernetes Grid clusters with a minimum of three worker nodes.	Deploying three worker nodes provides a higher level of availability of your workloads deployed to the cluster.	You must configure your customer workloads to use effectively the additional worker nodes in the cluster for high availability at an application-level.
AIR-TZU-CFG-011	Deploy Tanzu Kubernetes Grid clusters with small-size control plane nodes if your cluster will have less than 10 worker nodes.	You must size the control plane of a Tanzu Kubernetes Grid cluster according to the amount of worker nodes and pod density.	The size of the cluster nodes impacts the scale of a given cluster. If you must add a node to a cluster, consider the use of larger nodes. For AI GPU-enabled workloads, the GPU is the constraining factor for the amount of worker nodes that could be deployed.

## Life Cycle Management for Private AI Ready Infrastructure for VMware Cloud Foundation

Life cycle management design details the decisions for life cycle management of the GPU-enabled ESXi hosts in the VI workload domain and of the vSphere with Tanzu instance.

**Table 2-15. Life Cycle Management for Private AI Ready Infrastructure**

VMware Cloud Foundation Component	Description
VI workload domain	The GPU-enabled hosts that are part of a VI workload domain in VMware Cloud Foundation must be managed with a vSphere Lifecycle Manager image that includes the right components for the vendor-specific GPU, for example, the NVIDIA host driver and management daemon for ESXi.
vSphere with Tanzu life cycle management	You perform life cycle management of vSphere with Tanzu by using the vSphere Client and integrated life cycle management functions available in the <code>kubectl</code> command line tool.

**Table 2-16. Design Decisions on Life Cycle Management for Private AI Ready Infrastructure for VMware Cloud Foundation**

Decision ID	Design Decision	Design Justification	Design Implication
AIR-TZU-LCM-001	For life cycle management of a GPU-enabled VI workload domain, use a vSphere Lifecycle Manager image with a custom ESXi image that includes the GPU driver and any other core components from the GPU vendor.	<ul style="list-style-type: none"> <li>■ Eases maintaining the right host driver versions and daemons.</li> <li>■ Introduces consistency across the GPU-enabled hosts.</li> </ul>	You must create the customer vSphere Lifecycle Manager image before you deploy the VI workload domain.
AIR-TZU-LCM-002	Use the vSphere Client for life cycle management of a Supervisor.	Life cycle management of a Supervisor is not integrated in SDDC Manager.	You perform deployment, patching, updates, and upgrades of a Supervisor and its components manually.
AIR-TZU-LCM-003	Use <code>kubectl</code> for life cycle management of a Tanzu Kubernetes Grid cluster.	Life cycle management of a Tanzu Kubernetes Grid cluster is not integrated in SDDC Manager.	You perform deployment, patching, updates, and upgrades of a Tanzu Kubernetes Grid cluster and its components manually.

## Information Security and Access Control Design for Private AI Ready Infrastructure for VMware Cloud Foundation

You design authentication access, controls, certificate management, and firewall for vSphere with Tanzu according to industry standards and the requirements of your organization.

### vSphere with Tanzu Authentication and Access Control

You integrate vSphere with Tanzu with vCenter Single Sign-On for authentication. You can use the configured identity sources for vCenter Single Sign-On, such as Active Directory, in the Supervisor. You can assign permissions to users on a given Supervisor object, such as a namespace.

You must configure vCenter Server to be able to use Active Directory as an identity source. The in-depth design and configuration guidance for Active Directory over LDAP as an identity provider for vCenter Server is part of the [Identity and Access Management for VMware Cloud Foundation](#) validated solution. It is not mandatory to implement the entire solution, but you must complete the minimum applicable design and implementation sections for vCenter Server.

**Table 2-17. Design Decisions on Authentication and Access Control for Private AI Ready Infrastructure for VMware Cloud Foundation**

Decision ID	Design Decision	Design Justification	Design Implication
AIR-TZU-SEC-001	Create a security group in Active Directory for DevOps administrators. Add users who need <b>edit</b> permissions within a namespace to the group and grant <b>Can Edit</b> permissions to the namespace for that group.  If you require different permissions per namespace, create additional groups.	Necessary for auditable role-based access control within the Supervisor and Tanzu Kubernetes Grid clusters.	You must define and manage security groups, group membership, and security controls in Active Directory.
AIR-TZU-SEC-002	Create a security group in Active Directory for DevOps administrators. Add users who need <b>read-only</b> permissions in a namespace to the group, and grant <b>Can View</b> permissions to the namespace for that group.  If you require different permissions per namespace, create additional groups.	Necessary for auditable role-based access control within the Supervisor and Tanzu Kubernetes Grid clusters.	You must define and manage security groups, group membership, and security controls in Active Directory.

## Certificate Management

By default, vSphere with Tanzu uses a self-signed Secure Sockets Layer (SSL) certificate. This certificate is not trusted by end-user devices or Web browsers.

As a best practice, replace self-signed certificates with certificates that are signed by a third-party or enterprise Certificate Authority (CA).

**Table 2-18. Design Decisions on Certificate Management for Private AI Ready Infrastructure for VMware Cloud Foundation**

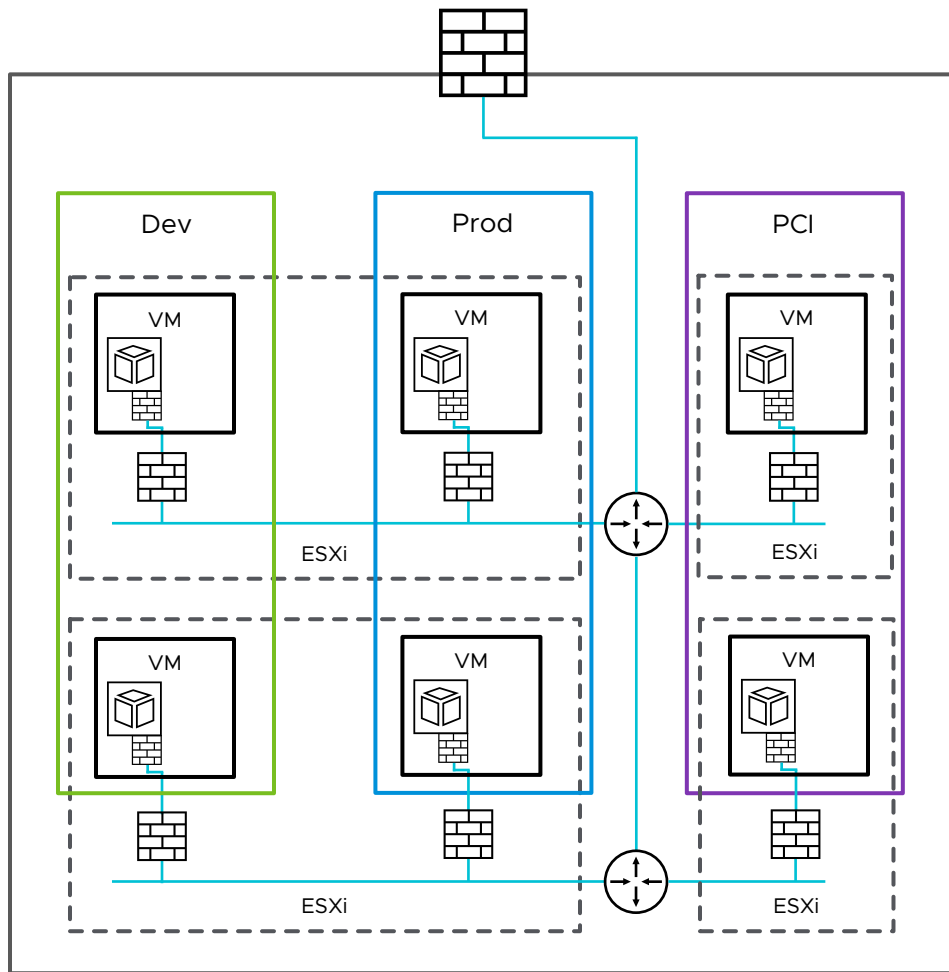
Decision ID	Design Decision	Design Justification	Design Implication
AIR-TZU-SEC-003	Replace the default self-signed certificate for the Supervisor management interface with a PEM-encoded, CA-signed certificate.	Ensures that the communication between administrators and the Supervisor management interface is encrypted by using a trusted certificate.	You must replace and manager certificates manually, outside certificate management automation of SDDC Manager.
AIR-TZU-SEC-004	Use a SHA-2 or higher algorithm when signing certificates.	The SHA-1 algorithm is considered less secure and has been deprecated.	Not all certificate authorities support SHA-2.

## Network Segmentation with NSX

NSX has two main firewall services.

- A gateway firewall serves as a centralized stateful firewall service for north-south traffic. It is implemented per Tier-0 gateway or Tier-1 gateway. Gateway firewalls are independent from NSX distributed firewalls from policy configuration and enforcement perspective, providing a means for defining perimeter security control in addition to distributed security control.
- A distributed firewall, embedded in the hypervisor kernel, provides visibility and control for virtualized workloads and networks for east-west traffic. Because the data plane of a distributed firewall runs at the vNIC level of each virtual machine, it can enforce access controls and inspect every flow for threats without traffic hair-pinning to a gateway firewall.

Figure 2-5. Microsegmentation for Private AI Infrastructure



## Distributed Firewall for Antrea-Based Kubernetes Clusters

NSX distributed firewall works with Antrea to secure traffic within an Antrea-based Kubernetes cluster. Antrea is a Container Network Interface (CNI) plug-in for Kubernetes that supports networking for container workloads. Tanzu Kubernetes Grid clusters use Antrea as the default CNI.

You must register Antrea-based Kubernetes clusters in NSX Manager to connect the NSX Manager control plane to the Antrea Central Control Plane Adapter. You can then create distributed firewall policies (security policies) in NSX and apply them to one or more Antrea-based Kubernetes clusters. The Antrea network plug-in creates an Antrea cluster network policy (ACNP) for each security policy that is applied to the Antrea-based Kubernetes clusters. If the rules contain sources, corresponding ingress rules are created in the ACNP. If the rules contain destinations, corresponding egress rules are created in the ACNP.

While you can update and delete the Antrea cluster network policies from the `kubectl` command line tool, keep in mind that such changes are not displayed in NSX Manager.

For more information on how to integrate Antrea with NSX distributed firewall, see [Integration of Kubernetes Clusters with Antrea CNI](#) in the *VMware NSX Administration Guide*.

Figure 2-6. Example of Distributed Firewall Rule Applied to an Antrea-Based Tanzu Kubernetes Grid Cluster

The screenshot displays the 'Distributed Firewall' configuration page. At the top, there are tabs for 'All Rules', 'Category Specific Rules' (which is active), 'Saved Drafts', and 'Settings'. Below the tabs, there are buttons for 'ACTIONS', 'REVERT', and 'PUBLISH'. A breadcrumb trail shows categories: 'ETHERNET (1)' > 'EMERGENCY (0)' > 'INFRASTRUCTURE (2)' > 'ENVIRONMENT (1)' > 'APPLICATION (18)'. Below the breadcrumb, there are buttons for '+ ADD POLICY', '+ ADD RULE', 'CLONE', 'UNDO', 'DELETE', and a menu icon. A filter bar says 'Filter by Name, Path and more'. The main table lists firewall rules with columns: Name, ID, Sources, Destinations, Services, Context Profiles, Applied To, and Action. Two rules are visible: 1. 'ai-ready-antrea' with ID '(1)', Sources 'Applied To: 1 Container Clusters', and Action 'Success'. 2. 'dcgm-exporter-out' with ID '1112', Sources 'dcgm-exporter', Destinations 'N/A', Services 'Any', Context Profiles 'N/A', Applied To 'Any', and Action 'Allow'.

Name	ID	Sources	Destinations	Services	Context Profiles	Applied To	Action
ai-ready-antrea	(1)	Applied To: 1 Container Clusters					Success
dcgm-exporter-out	1112	dcgm-exporter	N/A	Any	N/A	Any	Allow

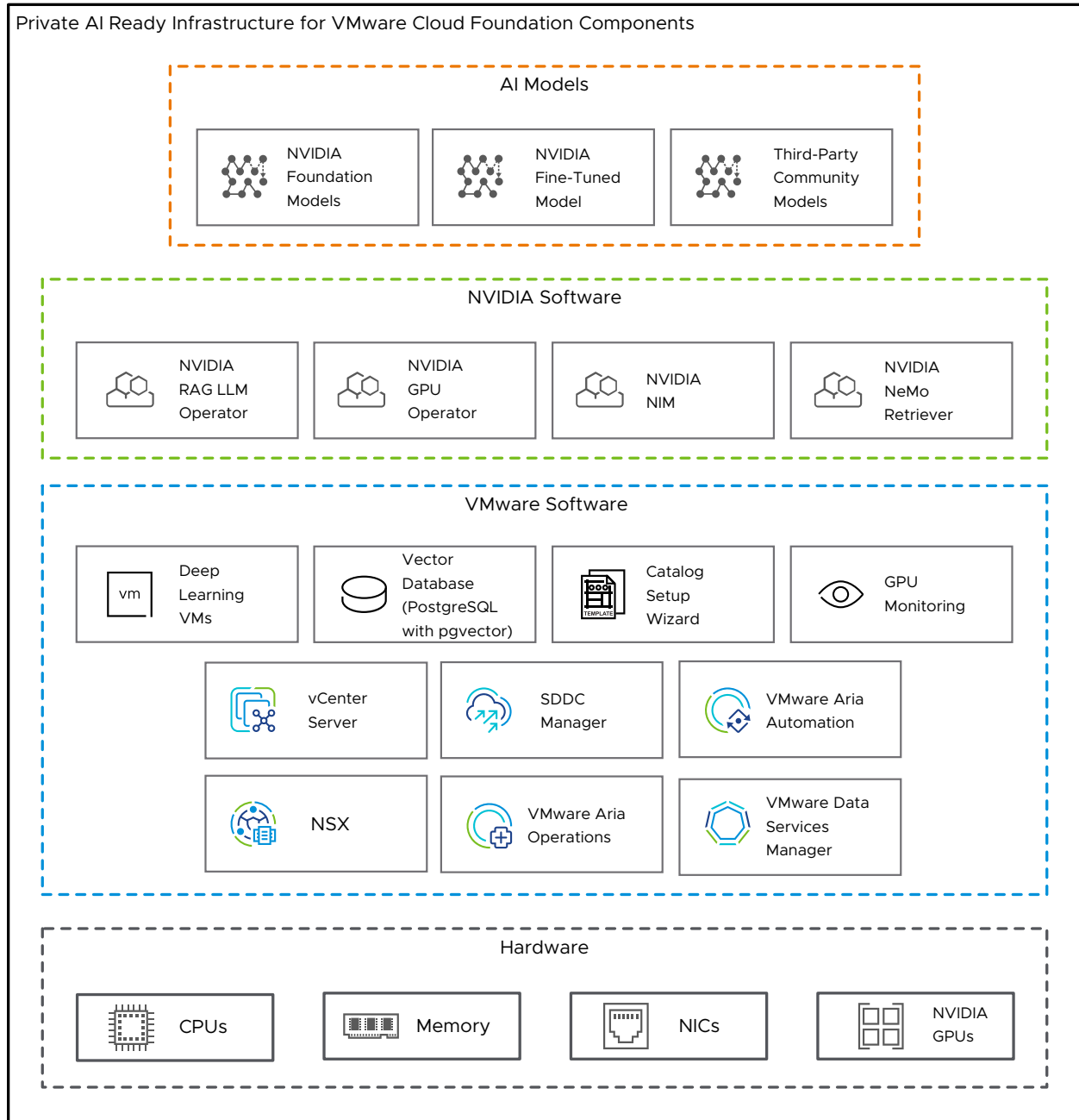
# Detailed Design for VMware Private AI Foundation with NVIDIA for Private AI Ready Infrastructure for for VMware Cloud Foundation

3

VMware Private AI Foundation with NVIDIA is an add-on solution on top of VMware Cloud Foundation that consists of multiple elements which you can use to deploy and manage your AI workloads delivered by VMware and NVIDIA.

Building on top of VMware Cloud Foundation, VMware Private AI Foundation with NVIDIA is an integrated solution designed to accelerate the enterprise GenAI journey. It includes support for open LLM Models, AI frameworks and software technology for developers to build, customize, and deploy generative AI models with billions of parameters.



**Figure 3-1. Components of Private AI Foundation with NVIDIA**

VMware Private AI Foundation with NVIDIA supports two use cases:

### Development use case

Data scientists dealing with complex machine learning tasks and GPUs face significant challenges in managing software versions and dependencies. VMware Private AI Foundation with NVIDIA includes custom Deep Learning VM images that are pre-configured with popular frameworks and optimized for deep learning workloads. These deep learning VMs align with the underlying infrastructure to facilitate inferencing and simplify AI developer workflows.

Cloud administrators and DevOps engineers can provision AI workloads, including Retrieval-Augmented Generation (RAG), in the form of deep learning virtual machines, pre-configured with popular frameworks and optimized for deep learning workloads. Data scientists can use these deep learning virtual machines for AI development.

### Production use case

Cloud administrators can provide DevOps engineers with a VMware Private AI Foundation with NVIDIA environment for provisioning production-ready AI workloads on Tanzu Kubernetes Grid (TKG) clusters on vSphere with Tanzu.

VMware Cloud Foundation automation enables the initialization and deployment of GPU-enabled VMs for specific use cases, including options for NGC packages and TKG clusters. Cloud administrators can use the Private AI Automation Services Wizard in VMware Aria Automation to set up self-service catalog items for various GenAI applications.

## Components Added by VMware Private AI Foundation with NVIDIA

VMware Private AI Foundation with NVIDIA adds certain components on top of the private AI infrastructure on top of vSphere and vSphere with Tanzu discussed in [Chapter 2 Detailed Design of Private AI Ready Infrastructure for VMware Cloud Foundation](#).

- Deep Learning VM images.
- Self-service catalog items in Service Broker in VMware Aria Automation for provisioning deep learning VMs and AI-accelerated Tanzu Kubernetes Grid cluster.
- GPU-related metrics in VMware Aria Operations for monitoring GPU use on ESXi hosts with NVIDIA GPUs and across vSphere clusters with such hosts.
- Vector databases, managed by VMware Data Services Manager, for use in Retrieval Augmented Generation (RAG) workloads.

## Deep Learning VM Images

The deep learning VM images included in VMware Private AI Foundation with NVIDIA come preconfigured with leading machine learning libraries, DL workloads, and tools. They are specifically optimized and tested by NVIDIA and VMware to leverage GPU acceleration as part of a VMware Cloud Foundation environment.

Deep learning VM images are delivered as vSphere VM templates, hosted and published by VMware in a content library. You can use these images to deploy a deep learning VM by using the vSphere Client or VMware Aria Automation.

The content library with deep learning VM images for VMware Private AI Foundation with NVIDIA is available at the <https://packages.vmware.com/dl-vm/lib.json> URL. In a connected environment, you create a subscribed content library connected to this URL, and in a disconnected environment - a local content library where you upload images from the central content library.

There are several ways to deploy a deep learning VM:

- Directly in to vSphere from the content library.
- As a virtual machine in the VM Service of the Supervisor by using the `kubect1` command line tool.
- By using the Service Broker catalog in VMware Aria Automation

You can deploy a deep learning VM with one of the following DL workloads from NVIDIA.

Software Bundle	Description
PyTorch	The PyTorch NGC Container is optimized for GPU acceleration, and contains a validated set of libraries that enable and optimize GPU performance. This container also contains software for accelerating ETL (DALI, RAPIDS), training (cuDNN, NCCL), and inference (TensorRT) workloads.
TensorFlow	The TensorFlow NGC Container is optimized for GPU acceleration, and contains a validated set of libraries that enable and optimize GPU performance. This container might also contain modifications to the TensorFlow source code in order to maximize performance and compatibility. The container also contains software for accelerating ETL (DALI, RAPIDS), training (cuDNN, NCCL), and inference (TensorRT) workloads.
CUDA Samples	This is a collection of containers to run CUDA workloads on the GPUs. The collection includes containerized CUDA samples for example, vectorAdd (to demonstrate vector addition), nbody (or gravitational n-body simulation) and other examples. These containers can be used for validating the software configuration of GPUs in the system or simply to run some example workloads.
DCGM Exporter	NVIDIA Data Center GPU Manager (DCGM) is a suite of tools for managing and monitoring NVIDIA datacenter GPUs in cluster environments. The monitoring stacks usually consist of a collector, a time-series database to store metrics and a visualization layer. DCGM-Exporter is an exporter for Prometheus to monitor the health and get metrics from GPUs.
Triton Inference Server	Triton Inference Server provides a cloud and edge inferencing solution optimized for both CPUs and GPUs. Triton supports an HTTP/REST and GRPC protocol that allows remote clients to request inferencing for any model being managed by the server. For edge deployments, Triton is available as a shared library with a C API that allows the full functionality of Triton to be included directly in an application.
Generative AI Workflow - RAG	This reference solution demonstrates how to find business value in generative AI by augmenting an existing foundational LLM to fit your business use case. This is done using RAG which retrieves facts from an enterprise knowledge base containing a company's business data. Pay special attention to the ways in which you can augment an LLM with your domain-specific business data to create AI applications that are agile and responsive to new developments.

For information on deep learning VM images in VMware Private AI Foundation with NVIDIA, see [About Deep Learning VM Images in VMware Private AI Foundation with NVIDIA](#) and [Deep Learning Workloads in VMware Private AI Foundation with NVIDIA](#) in the *VMware Private AI Foundation with NVIDIA Guide*.

Read the following topics next:

- [NVIDIA Licensing System Design for Private AI Ready Infrastructure for VMware Cloud Foundation](#)
- [VMware Data Services Manager Design for Private AI Ready Infrastructure for for VMware Cloud Foundation](#)

# NVIDIA Licensing System Design for Private AI Ready Infrastructure for VMware Cloud Foundation

This section details two NVIDIA AI Enterprise (NVAIE) License Systems, i.e., Cloud License Service (CLS) and Delegated License Service (DLS).

NVIDIA AI Enterprise (NVAIE) is licensed on a per-GPU basis with flexible licensing options to accommodate the versatile needs of enterprise AI deployments. For more information, see [NVIDIA AI Enterprise Packaging, Pricing, and Licensing Guide](#).

The NVIDIA License System (NLS) provides monitoring and reporting on license usage for capacity planning. NLS supports two types of service instances.

## Cloud License Service (CLS)

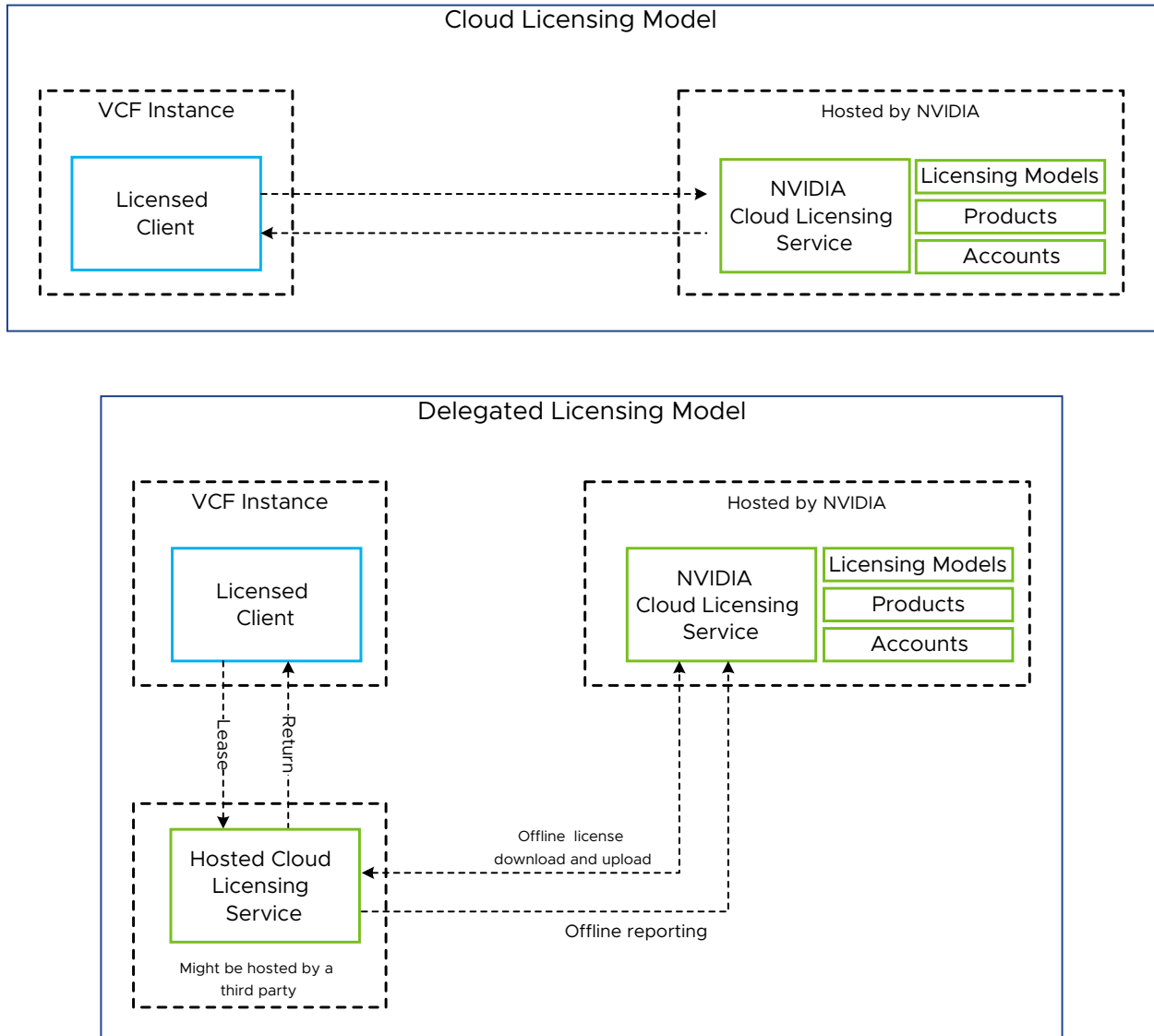
A Cloud License Service (CLS) instance is hosted on the NVIDIA Licensing Portal. You are not required to manually download licenses from the portal and upload them to the instance. CLS instances can scale dynamically as needed. Maintenance and feature updates for a CLS instance are typically handled by NVIDIA and the cloud service provider.

## Delegated License Service (DLS)

A Delegated License Service (DLS) instance is deployed on-premises within a location that is reachable from your private network, such as within your data center. Because the DLS instance is isolated from the NVIDIA Licensing Portal, you must download licenses manually from the portal and then upload them to the instance for activation.

For more information, see the [NVIDIA License System User Guide](#).

Figure 3-2. Design for a Cloud License Service and a Delegated License Service



## NVIDIA Licensing System Design

**Table 3-1. Design Decisions on NVIDIA Licensing System Design for Private AI Ready Infrastructure for VMware Cloud Foundation**

Decision ID	Design Decision	Design Justification	Design Implication
AIR-NVD-LIC-001	For Delegated License Service (DLS) instances, account for extra compute, storage, and network resources as part of your management domain.	DLS is deployed as a virtual appliance with specific hardware requirements. The appliance can also be configured in a high availability setup, independent from vSphere HA.	<ul style="list-style-type: none"> <li>■ Increased resources for the management stack.</li> <li>■ You must perform life cycle management of the DLS instance.</li> </ul>
AIR-NVD-LIC-002	For Cloud License Service (CLS) instances, Internet access is required.	Internet access is required between a licensed client and a CLS instance. Ports 80 and 443 (Egress) must be allowed.	<ul style="list-style-type: none"> <li>■ Introduces potential security risks.</li> <li>■ You must enforce firewall rules, intrusion detection systems, and monitoring.</li> </ul>

## VMware Data Services Manager Design for Private AI Ready Infrastructure for for VMware Cloud Foundation

VMware Data Services Manager is a unified control plane for managing data services in VMware Cloud Foundation. It simplifies deployment, management, and scaling of databases like Postgres, MySQL for AI applications.

VMware Data Services Manager offers advanced data services integrated with vSphere and VMware Cloud Foundation. It supports easy deployment, simplified operations, and life cycle management of databases, such as MySQL and PostgreSQL with self-service features. The platform provides customized Kubernetes access and an API for tailored application needs.

The pgvector extension available in PostgreSQL databases deployed by VMware Data Services Manager facilitates vector database integration for generative AI applications. Integration with VMware Private AI Foundation with NVIDIA enables adoption of RAG pipelines for various use cases like code generation, contact center resolutions, IT automation, and advanced data retrieval.

SPBM-based policies are consumed by infrastructure policies in VMware Data Services Manager. An infrastructure policy includes compute, storage, network, and virtual machine resource details. vSphere administrators use infrastructure policies to allocate vSphere infrastructure resources for provisioning data services with VMware Data Services Manager.

SPBM storage policies must meet data availability and performance requirements. Run the DSM-managed databases on a separate vSAN cluster that could be part of the same VI WLD where the GPU-enabled clusters are to reduce noisy neighbor scenarios and to tailor cluster services and limit operational and maintenance domains.

## VMware Data Services Manager Design

**Table 3-2. Design Decisions on VMware Data Services Manager Design for Private AI Ready Infrastructure for VMware Cloud Foundation**

Decision ID	Design Decision	Design Justification	Design Implication
AIR-DSM-001	Deploy VMware Data Services Manager in the management domain.	A 1:1 relationship between a VMware Data Services Manager appliance and a vCenter Server instance is required. The vCenter Server instances for the VI workload domains in a VMware Cloud Foundation instance run in the management domain.	You must deploy one VMware Data Services Manager appliance per vCenter Server which impacts the required resources for the management domain and its clusters.
AIR-DSM-002	For production-grade deployments, deploy PostgreSQL databases in HA mode (3 or 5 nodes).	High Availability of Vector Databases, increasing the overall availability of the whole system that depends on the DBs.	Increased resource consumption of the target VI WLD, and increased number used IP Addresses.
AIR-DSM-003	Allocate enough IP addresses for the IP pools of infrastructure policies.	You determine the number of IP addresses reserved for the IP pools according to the requirements and to the high availability topology of the database deployed by using VMware Data Services Manager. For example, a 5-node PostgreSQL cluster requires 7 IP addresses - one for each node, one for kube_VIP, and one for database load balancing).	You must consider planning and subnet sizing.
AIR-DSM-004	Define VM classes in VMware Data Services Manager that align to your resource requirements.	Consider the use case, types of workloads using the databases, amount of data, Transactions per Second (TPS), and other factors, such as target infrastructure overcommitment if applicable.  See <a href="#">Data Services Manager Documentation</a> and <a href="#">Data Modernization with VMware Data Services Manager</a> .	You must consider VMware Data Services Manager planning and design.

**Table 3-2. Design Decisions on VMware Data Services Manager Design for Private AI Ready Infrastructure for VMware Cloud Foundation (continued)**

Decision ID	Design Decision	Design Justification	Design Implication
AIR-DSM-005	Configure LDAP as Directory Service for VMware Data Services Manager.	LDAP (TLS available if needed) can be configured as the identity provider to import users and assign roles on VMware Data Services Manager.	Increased security operation costs. You must allow port access from VMware Data Services Manager to the LDAP identity source: <ul style="list-style-type: none"> <li>■ LDAP - 389 TCP</li> <li>■ LDAPS - 636 TCP/UDP</li> </ul>
AIR-DSM-006	Configure the S3-compatible object store, for example, MinIO, with TLS.	The provider repositories for core VMware Data Services Manager storage, backup, logs and database backups must be enabled with TLS.	<ul style="list-style-type: none"> <li>■ Security and complexity is increased.</li> <li>■ You must manage TLS certificates.</li> </ul>
AIR-DSM-007	Create a <a href="#">VMware Tanzu Network account</a> and use it to configure a refresh token in VMware Data Services Manager.	Database templates and software updates are uploaded to VMware Tanzu Network.  In a connected environment, you must configure a Tanzu Network Refresh Token as part of the VMware Data Services Manager setup. In a disconnected environment, you must download the air-gapped environment repository and uploaded it manually to the Provider Repository.	You must perform this operation manually.



**Table 3-2. Design Decisions on VMware Data Services Manager Design for Private AI Ready Infrastructure for VMware Cloud Foundation (continued)**

Decision ID	Design Decision	Design Justification	Design Implication
AIR-DSM-008	If you plan to run databases managed by VMware Data Services Manager on SAN ESA clusters, create a vSphere SPBM policy that is based on erasure coding.	<p>Provides performance that is equivalent to RAID 1 but with no compromises and with better space efficiency.</p> <p>The available erasure coding, RAID 5 or RAID 6, depends on the size of the all-flash vSAN ESA cluster. Erasure Coding 5 RAID 5 erasure coding requires a minimum of 4 ESXi hosts while RAID 6 erasure coding requires a minimum of 6 ESXi hosts.</p>	<ul style="list-style-type: none"> <li>■ Design complexity, cost, and management overhead of the solution are increased.</li> <li>■ You must perform this operation manually or by using PowerCLI.</li> </ul>
AIR-DSM-009	Use RAID 5 or RAID 6 erasure coding as the default vSAN storage policy for databases.	Eliminates the trade-off of performance and deterministic space efficiency. Set FTT=1 for RAID 5 and FTT=2 for RAID 6 according to the number of hosts in the vSAN ESA cluster and your data availability requirements.	Design complexity is increased.

# Planning and Preparation for Private AI Ready Infrastructure for VMware Cloud Foundation

## 4

Before you start implementing the components of the *Private AI Ready Infrastructure for VMware Cloud Foundation* validated solution, you must ensure the environment has a specific compute, storage, and network configuration, and provides external services to the components of the solution.

To capture environment specific input values that are required during the implementation, use the *VMware Cloud Foundation Planning and Preparation Workbook*.

Carefully review the *VMware Cloud Foundation Planning and Preparation Workbook* ahead of implementation to avoid costly rework and delays. Capture input values that are specific to your environment and verify that the components, required by this solution, are available.

Ensure that any GPUs or specialized high speed networking adapters that will be used are supported by VMware Cloud Foundation 5.1.1 according to [VMware Compatibility Guide for Shared Pass-Through Graphics](#). GPU support depends on the technology used, e.g. NVIDIA vGPU (Shared Direct Mode) or passthrough. To verify that your device is supported for AI and Machine Learning use cases, on the *VMware Compatibility Guide* page, apply the following filter:

<i>VMware Compatibility Guide</i> Filter	Value
GPU Technology	Compute
Comput	AI/ML

To confirm vGPU support for your GPU for VMware Cloud Foundation, see [NVIDIA® Virtual GPU \(vGPU\) Software Documentation](#).

# Implementation of Private AI Ready Infrastructure for VMware Cloud Foundation

## 5

Implementing the *Private AI Ready Infrastructure for VMware Cloud Foundation* validated solution includes configuring vCenter Server and NSX and activating vSphere with Tanzu.

To implement and configure private AI ready infrastructure, two alternative methods exist. You can use the user interface of each component in the solution or you can use the open-source PowerShell cmdlets. You can directly reuse the PowerShell commands by replacing the provided sample values with values from your *VMware Cloud Foundation Planning and Preparation Workbook*.

This guidance provides a prescriptive path for deploying vSphere with Tanzu, Tanzu Kubernetes clusters using the Tanzu Kubernetes Grid Service, and sample Kubernetes applications in a VI workload domain. For more information on other deployment options and configurations, see the *vSphere with Tanzu* product documentation on the [vSphere documentation page](#).

## Prerequisites

To complete the implementation of *Private AI Ready Infrastructure for VMware Cloud Foundation* validated solution, verify that your system fulfills the following prerequisites.

**Table 5-1. Prerequisites for Implementation of Private AI Ready Infrastructure for VMware Cloud Foundation**

Category	Prerequisite
Environment	<ul style="list-style-type: none"><li>■ Verify that your VMware Cloud Foundation version is listed in the <a href="#">Support Matrix</a> for this solution.</li><li>■ Verify that you configure your environment according to <a href="#">Before You Apply This Guidance</a>.</li><li>■ Verify that you capture all parameters for the <i>Private AI Ready Infrastructure</i> tab of the <a href="#">VMware Cloud Foundation Planning and Preparation Workbook</a>.</li><li>■ Verify that your VMware Cloud Foundation instance is healthy and fully operational. See <a href="#">VMware Cloud Foundation Operations Guide</a>.</li></ul>
Domain Name Service	<ul style="list-style-type: none"><li>■ Verify that the required DNS entries are created in the DNS server for the associated forward and reverse zones.</li></ul>
Active Directory	<ul style="list-style-type: none"><li>■ Verify that Active Directory Domain Controllers are available in the environment.</li><li>■ Verify that the required service accounts are created in Active Directory.</li><li>■ Verify that the required security groups are created in Active Directory.</li></ul>

**Table 5-1. Prerequisites for Implementation of Private AI Ready Infrastructure for VMware Cloud Foundation (continued)**

Category	Prerequisite
Certificate Authority	<ul style="list-style-type: none"> <li>■ Verify that a Microsoft Certificate Authority is available for the environment.</li> <li>■ Verify that you download the CertGenVVS utility and it is available for generating certificates. See <a href="#">Certificate Generation Utility for VMware Validated Solutions</a></li> </ul>
Identity and Access Management	<ul style="list-style-type: none"> <li>■ Verify that your Active Directory domain is added as an identity provider for vCenter Single Sign-On as defined in <a href="#">Identity and Access Management for VMware Cloud Foundation</a>.</li> </ul>

If you want to use the `kubectl` command line tool, for the implementation and configuration of the validated solution, verify that the following tools are installed on a machine in your environment.

**Table 5-2. Prerequisites for Using the Command Prompt for Implementation of Private AI Ready Infrastructure for VMware Cloud Foundation**

Category	Prerequisite
<code>kubectl</code> Command Line Tool	<ul style="list-style-type: none"> <li>■ Download the Kubernetes CLI Tools for vSphere from your Supervisor.</li> <li>■ Install the standard open-source <code>kubectl</code> utility and the vSphere <code>kubectl</code> plugin.</li> </ul> <p>For the PowerShell procedure, both the <code>kubectl.exe</code> and <code>kubectl-vsphere.exe</code> binaries must be in the same folder.</p> <p>See <a href="#">Download and Install the Kubernetes CLI Tools for vSphere</a>.</p>

If you want to use the open-source infrastructure-as-code method for the implementation and configuration of the *Private AI Ready Infrastructure for VMware Cloud Foundation* validated solution, verify that your system fulfills the following prerequisites.

**Table 5-3. Prerequisites for CLI Implementation of Private AI Ready Infrastructure for VMware Cloud Foundation**

CLI Method	Prerequisite
PowerShell	<ul style="list-style-type: none"> <li>■ Verify that your system has Microsoft PowerShell installed. See <a href="#">Microsoft PowerShell</a>.</li> <li>■ Install the <a href="#">PowerValidated Solutions</a> PowerShell module together with the supporting modules from the PowerShell Gallery. <div data-bbox="544 470 1342 732"> <pre>Install-Module -Name VMware.PowerCLI -MinimumVersion 13.2.1 -Scope AllUsers Install-Module -Name VMware.vSphere.SsoAdmin -MinimumVersion 1.3.9 -Scope AllUsers Install-Module -Name ImportExcel -MinimumVersion 7.8.5 -Scope AllUsers Install-Module -Name PowerVCF -MinimumVersion 2.4.0 -Scope AllUsers Install-Module -Name PowerValidatedSolutions -MinimumVersion 2.11.0 -Scope AllUsers</pre> </div> </li> <li>■ Import the <a href="#">PowerValidatedSolutions</a> and the <a href="#">PowerCLI</a> PowerShell modules. <div data-bbox="544 821 1106 844"> <pre>Import-Module -Name PowerValidatedSolutions</pre> </div> </li> </ul> <p><b>Note</b> To report issues, obtain support, or suggest enhancements to the open-source PowerShell Module, use <a href="#">GitHub Issues</a> in the GitHub repository.</p>

## Procedure

### 1 [Configure the vSphere Environment for Private AI Ready Infrastructure for VMware Cloud Foundation](#)

Before you activate vSphere with Tanzu, create storage policies for the Supervisor and vSphere namespaces. The policies represent available datastores in the vSphere environment. They control the storage placement of such objects as control plane VMs, pod ephemeral disks, container images, and persistent storage volumes. When you use VMware Tanzu™ Kubernetes Grid™ Service, the storage policies also dictate how the Tanzu Kubernetes Cluster nodes are deployed.

### 2 [Deploy and Configure a Tanzu Kubernetes Grid Cluster for vSphere with Tanzu for Private AI Ready Infrastructure for VMware Cloud Foundation](#)

To complete the configuration of the vSphere with Tanzu environment, after the Supervisor is configured, deploy a Tanzu Kubernetes Grid (TKG) cluster on the Supervisor using the `kubectl` command line tool.

### 3 [Adding VMware Private AI Foundation with NVIDIA to Private AI Ready Infrastructure for VMware Cloud Foundation](#)

Implementing VMware Private AI Foundation with NVIDIA on top of the private AI infrastructure components includes creating vector databases by using VMware Data Services Manager, and deploying VMs for AI development based on NVIDIA DL workloads images and GPU-enabled Tanzu Kubernetes Grid (TKG) clusters for running NVIDIA NGC container images.

## Configure the vSphere Environment for Private AI Ready Infrastructure for VMware Cloud Foundation

Before you activate vSphere with Tanzu, create storage policies for the Supervisor and vSphere namespaces. The policies represent available datastores in the vSphere environment. They control the storage placement of such objects as control plane VMs, pod ephemeral disks, container images, and persistent storage volumes. When you use VMware Tanzu™ Kubernetes Grid™ Service, the storage policies also dictate how the Tanzu Kubernetes Cluster nodes are deployed.

### Enable vSphere vMotion for vGPU-Enabled Virtual Machines for Private AI Ready Infrastructure for VMware Cloud Foundation

You must explicitly turn on vSphere vMotion for virtual machines that use NVIDIA vGPUs without causing data loss.

During the stun time, you are unable to access the VM. Once the migration is completed, access to the VM resumes and all applications continue from their previous state.

The expected VM stun time (the time when the VM is inaccessible to users during vMotion) can vary depending on the amount of GPU memory that is currently being consumed by the VM. For information on frame buffer size in vGPU profiles, refer to the [NVIDIA Virtual GPU documentation](#). GPUs with more memory or vSphere infrastructure leveraging high speed networking (25 GbE, 100 GbE, etc.) could potentially have a direct impact on the stun time.

Starting with vSphere 8.0 U2, DRS can estimate the Stun Time for a given vGPU VM configuration. When the DRS Cluster Advanced Options are set and the Estimated VM Devices Stun Time for a VM is lower than the VM Devices vMotion Stun Time limit, DRS will automate VM migrations potentially overriding the Default 100 seconds if required. For more information on how to setup this DRS advanced configuration refer to the following [vGPU Virtual Machine automated migration for Host Maintenance Mode in a DRS Cluster \(88271\)](#).

#### Procedure

- 1 Log in to the VI workload domain vCenter Server at `https://<vcenter_server_fqdn>/ui` as `administrator@vsphere.local`.
- 2 In the **Hosts and clusters** inventory, select the vCenter Server for the VI workload domain.
- 3 On the **Configure** tab, select **Settings > Advanced Settings** and click **Edit settings**.
- 4 In the **Edit vCenter Server Advanced Settings** dialog box, find the `vgpu.hotmigrate.enabled` property and set it to **Enabled**.

If the `vgpu.hotmigrate.enabled` property is not available in the advanced settings table, add it, setting its value to `true`.

- 5 Click **Save**.

## Install the Vendor GPU Driver on the ESXi Hosts for Private AI Ready Infrastructure for VMware Cloud Foundation

You upload the driver to the vSphere Lifecycle image for the default cluster of the GPU-enabled workload domain and remediate the hosts.

### Prerequisites

### Procedure

- 1 Download the driver version for your GPU according to the *VMware Compatibility Guide*.
  - a Go to the [Shared Passthrough Graphics and AI/ML](#) section of the *VMware Compatibility Guide*, click your GPU model, and take a note of the supported driver versions and other add-ons, such as GPU monitoring daemons.
  - b Download the drivers directly from the vendor Web site.

Vendor	GPU Driver Download Location
NVIDIA	<a href="#">NVIDIA Application Hub</a>

- 2 Log in to the management domain vCenter Server at **`https://<vcenter_server_fqdn>/ui`** as **`administrator@vsphere.local`**.
- 3 Import the GPU driver for the ESXi version compatible with this validated solution into the vSphere Lifecycle Manager depot.
  - a From the vSphere Client Menu, select **Lifecycle Manager**.
  - b On the **Lifecycle Manager** page, click **Actions > Import Updates**
  - c In the **Import Updates** dialog box, click **Browse**, locate the GPU driver ZIP file, and click **Open**.
  - d Click **Import**.

The driver appears in the **Components** table on the **Lifecycle Manager** page.

- 4 Add the GPU driver to the image for the default cluster of the workload domain.
  - a In the **Hosts and clusters** inventory, select the cluster.
  - b On the **Updates** tab, select **Hosts > Image**.
  - c In the **Image** pane, click **Edit**.
  - d In the **Edit Image** pane, next to **Components**, click the **Show details** link.
  - e Click **Add components** above the component table that appears.
  - f Select the driver component and version that you plan to use on the GPU-enabled ESXi hosts in the workload domain and click **Select**.

- g Click **Validate**.
- h Click **Save**.

A warning message that the ESXi hosts in the cluster are non-compliant appears.

- 5 Remediate the default cluster with the cluster image containing the GPU driver.
  - a On the **Updates** tab for the cluster, select **Hosts > Image**.
  - b In the **Image Compliance** pane, click **Remediate All**.
  - c In the **Review Remediation Impact** dialog box, review the impact summary, the applicable remediation settings, and the EULA.
  - d Accept the EULA.
  - e Click **Start remediation**.
- 6 After the remediation process is complete, place each host in maintenance mode and restart it.
- 7 Log in to SDDC Manager at [https://<sddc\\_manager\\_fqdn>](https://<sddc_manager_fqdn>) with a user the **Admin** role.
- 8 Navigate to **Lifecycle Management > Image Management**.
- 9 Extract the vSphere Lifecycle Manager image with the GPU driver component.
  - a On the **Import Image** tab, under the **Option 1** section, select the management domain and the empty cluster.
  - b Click **Extract cluster image**.

The extracted cluster image appears on the **Available Images** tab. It can be used for a new VI workload domain or a new cluster in a VI workload domain enabled for vSphere Lifecycle Manager images.

#### What to do next

For information on MIG and how to enable it, see [NVIDIA Multi-Instance GPU User Guide](#).

## Deploy and Configure a Tanzu Kubernetes Grid Cluster for vSphere with Tanzu for Private AI Ready Infrastructure for VMware Cloud Foundation

To complete the configuration of the vSphere with Tanzu environment, after the Supervisor is configured, deploy a Tanzu Kubernetes Grid (TKG) cluster on the Supervisor using the `kubectl` command line tool.

To perform these operations, you can also use the fully-automated self-service approach that is part of the VMware Private AI Foundation with NVIDIA solution add-on.



## Create a Namespace for the Tanzu Kubernetes Grid Cluster for Private AI Ready Infrastructure for VMware Cloud Foundation

To run applications that require upstream Kubernetes compliance, you can provision a Tanzu Kubernetes Grid cluster.

Tanzu Kubernetes clusters are fully upstream-compliant Kubernetes clusters that run on top of your Supervisor.

To help you to organize and manage your development projects, you can optionally divide the clusters into vSphere namespaces.

### Procedure

- 1 Log in to the VI workload domain vCenter Server at **https://<vi\_workload\_vcenter\_server\_fqdn>/ui** as **administrator@vsphere.local**.
- 2 From the vSphere Client Menu, select **Workload Management**.
- 3 On the **Workload Management** page, click the **Namespaces** tab and click **New Namespace**.
- 4 In the **Create Namespace** dialog box, select the Supervisor, enter name for the namespace, and click **Create**.
- 5 Click the **Storage** tab for the newly-created vSphere namespace.
- 6 Under **Storage Policies**, click **Edit**.
- 7 In the **Select Storage Policies** dialog box, select the storage policy that you created earlier and click **OK**.

## Assign the New Tanzu Cluster Namespace Roles to Active Directory Groups for VMware Cloud Foundation

You assign roles for the Namespace to Active Directory groups, . You can later assign access to users by adding them to these groups. You assign access to separate Active Directory groups for the edit and view roles in the Namespace. External Identity Providers such as Okta are also supported.

### Procedure

- 1 Log in to the VI workload domain vCenter Server at **https://<vi\_workload\_vcenter\_server\_fqdn>/ui** as **administrator@vsphere.local**.
- 2 From the vSphere Client Menu, select **Workload Management**.
- 3 On the **Workload management** page, on the **Namespaces** tab, click the new Namespace.
- 4 Click the **Permissions** tab.
- 5 Provide **edit** permissions to your Active Directory group intended for admins for the Namespace.
  - a Click **Add**.

- b In the **Add Permissions** dialog box, enter the **Identity source** and **User/Group** for edit access according to your values in the *VMware Cloud Foundation Planning and Preparation Workbook*, set the **Role** to **Can edit**, and click **OK**.
- 6 Provide **read-only** permissions to your Active Directory group intended for viewers for the Namespace.
  - a Click **Add**.
  - b In the **Add Permissions** dialog box, enter the **Identity source** and **User/Group** for read-only access according to your values in the *VMware Cloud Foundation Planning and Preparation Workbook*, set the **Role** to **Can view**, and click **OK**.

## Add GPU-Enabled VM Classes for the Tanzu Kubernetes Grid Cluster for Private AI Ready Infrastructure for VMware Cloud Foundation

Before you deploy a GPU-enabled Tanzu Kubernetes Grid cluster that can run AI workloads, you must add one or more VM classes defining access to the GPUs. You then assign these VM classes to the worker nodes of the cluster.

This example uses a `guaranteed-large` configuration for the control plane nodes and `vgpu-a100-16vcpu-128gb` for the worker nodes.

### Procedure

- 1 Log in to the VI workload domain vCenter Server at **`https://<vi_workload_vcenter_server_fqdn>/ui`** as **`administrator@vsphere.local`**.
- 2 From the vSphere Client Menu, select **Workload Management**.
- 3 On the **Workload Management** page, on the **Services** tab, click the **VM Service** card.
- 4 On the **VM Service** page, click the **VM Classes** tab.
- 5 Click the **Create VM Class**.
- 6 On the **Name** page of the **Create VM Class** wizard, enter a name for the VM class and click **Next**.

For example: `vgpu-a100-16vcpu-128gb`.

- 7 On the **Compatibility** page, select ESXi 8.0 U2 and later and click **Next**.
- 8 Click the **Configuration > Virtual Hardware** tab.
- 9 Add the GPU device to the VM class.
  - a Select **Add New Device > PCI Device**.
  - b Select the desired NVIDIA Grid vGPU device from the list according to the GPU model and GPU sharing mode.

There are two types of NVIDIA Grid vGPU profiles: Time sharing and Multi-Instance GPU (MIG) sharing. The profile is detected by the system when you select the device.

**Note** You can add only one NVIDIA GRID vGPU device of type MIG profile to a VM class.

- c Click **Select**.

A **New PCI device** device appears on the **Virtual Hardware** tab.

- 10 Configure the desired settings for **CPU**, **Memory**, **New PCI Device**, **Video Card**, and **Security Devices**.

**Table 5-4. CPU Configuration**

Setting	Configuration
CPU	Assign at least 16 virtual CPU cores.
CPU Topology	Assigned at power on
Reservation	Reservation must be between 0 and 10 MHz
Limit	Limit must be greater than or equal to 10 MHz
Shares	Options are Low, Normal, High, Custom
Hardware virtualization	Select this option to expose hardware assisted virtualization to the guest OS
Performance Counters	Enable virtualized CPU performance counters
Scheduling Affinity	Select a physical processor affinity for this virtual machine. Use '-' for ranges and ',' to separate values. For example, "0, 2, 4-7" would indicate processors 0, 2, 4, 5, 6 and 7. Clear the string to remove affinity settings.
I/O MMU	Select to enable memory management unit (page to disk)

**Table 5-5. Memory Configuration**

Setting	Configuration
Memory	Set it to at least 64 GB memory.
Reservation	Specify the guaranteed minimum allocation for a virtual machine, or reserve all guest memory. If the reservation cannot be met, the VM cannot run.
Limit	Select the amount of memory to limit to place a limit on the consumption of memory for a VM.

Table 5-5. Memory Configuration (continued)

Setting	Configuration
Shares	Select the amount of memory to share. Shares represent a relative metric for allocating memory capacity. For more information, see <a href="#">Memory Sharing</a> .
Memory Hot Plug	Enable (check) to allow the addition of memory resources to a VM that is powered on. See <a href="#">Memory Hot Add Settings</a> for details.

Table 5-6. Configure Video Card

Setting	Configuration
Video Card	Choose to auto-detect settings from the hardware or enter custom settings. If you select auto-detect, other settings are not configurable.
Number of displays	Select the number of displays.
Total video memory	Enter the total video memory, in MB.
3D Graphics	Select to enable 3D support.

Table 5-7. Configure Security Devices

Settings	Configuration
Security Device	If the SGX security device is installed, you can configure the VM settings here, otherwise this field is not configurable. See the <a href="#">SGX documentation</a> for details.

- 11 For the GPUDirect feature, click the **Advanced Parameters** tab and add the following attribute-value pairs.
  - `pciPassthru.allowP2P=True`
  - `pciPassthru.RelaxACSforP2P=True`
- 12 Click **Next** and click **Finish**.
- 13 Repeat the steps to create VM classes for the other vGPU profiles you plan to use for cluster worker nodes.
- 14 Add the VM class to the namespace for the GPU-enabled Tanzu Kubernetes Grid clusters.
  - a On the **Workload Management** page, click the **Namespaces** tab and click the **Summary** tab.
  - b In the **VM Service** card, click the **Manage VM Classes** link.
  - c Select the **vgpu-a100-16vcpu-128gb** GPU-enabled class and the `guaranteed-large` VM classes.
  - d Select other VM classes required for your cluster control and worker nodes.

- e Click **OK**.

## Provision a Tanzu Kubernetes Grid Cluster for Private AI Ready Infrastructure for VMware Cloud Foundation

Provision a Tanzu Kubernetes Grid cluster by using `kubectl` and a YAML file for input. The command prompt procedure uses example values from the *VMware Cloud Foundation Planning and Preparation Workbook*.

For the PowerShell procedure, you must know the path where `kubectl` and `kubectl-vsphere` binaries are located. The path is required in the `$kubectlBinLocation` variable.

### Command Prompt Procedure

- 1 In a command prompt, log in to the Supervisor by using `kubectl`.

```
kubectl vsphere login --server 192.168.21.2 --vsphere-username Supervisor_Cluster_User
```

- 2 Switch the `kubectl` context to the `sfo-w01-tkc01` namespace.

```
kubectl config use-context Tanzu_Kubernetes_Namespace
```

- 3 Create a `sfo-w01-tkc01.yaml` text file with the following specifications.

```
apiVersion: cluster.x-k8s.io/v1beta1
kind: Cluster
metadata:
  name: sfo-w01-tkc01
  namespace: Tanzu_Kubernetes_Namespace
spec:
  clusterNetwork:
    topology:
    services:
      cidrBlocks: ["198.51.100.0/12"]
  pods:
    cidrBlocks: ["192.0.2.0/16"]
  serviceDomain: "cluster.local"
  topology:
    class: tanzukubernetescluster
    version: v1.26.5---vmware.2-fips.1-tkg.1
    controlPlane:
      replicas: 3
      metadata:
        annotations:
          run.tanzu.vmware.com/resolve-os-image: os-name=ubuntu
    workers:
      machineDeployments:
        - class: node-pool
          name: node-pool-gpu
          replicas: 2
          metadata:
            annotations:
              run.tanzu.vmware.com/resolve-os-image: os-name=ubuntu
```

```

    variables:
      overrides:
        - name: vmClass
          value: vgpu-a100-16vcpu-128gb
variables:
  - name: vmClass
    value: guaranteed-large
  - name: storageClass
    value: vsphere-with-tanzu-storage-policy
  - name: defaultStorageClass
    value: vsphere-with-tanzu-storage-policy
  - name: nodePoolVolumes
    value:
      - name: containerd
        capacity:
          storage: 50Gi
        mountPath: /var/lib/containerd
        storageClass: vsphere-with-tanzu-storage-policy
      - name: kubelet
        capacity:
          storage: 25Gi
        mountPath: /var/lib/kubelet
        storageClass: vsphere-with-tanzu-storage-policy

```

- 4 Use `kubectl` to deploy the Tanzu Kubernetes Grid cluster from your YAML file input.

```
kubectl apply -f ./sfo-w01-tkc01.yaml
```

- 5 After the deployment of the Tanzu Kubernetes Grid cluster completes, run `kubectl` to verify the Tanzu Kubernetes Grid cluster status.

```

kubectl get cluster
NAME          PHASE          AGE    VERSION
sfo-w01-tkc01 Provisioned     6m     v1.26.5+vmware.2-fips.1

```

- 6 Log in to the new Tanzu Kubernetes Grid cluster and run `kubectl` to verify the status of the control plane and worker nodes.

```

kubectl vsphere login --server 192.168.21.2 --vsphere-username Supervisor_Admin --tanzu-
kubernetes-cluster-namespace Tanzu_Kubernetes_Namespace --tanzu-kubernetes-cluster-name
Tanzu_Kubernetes_Cluster_Name --insecure-skip-tls-verify

```

```
kubectl get nodes
```

NAME	STATUS	ROLES	AGE	VERSION
sfo-w01-tkc01-node-pool-gpu-9w5jr-768f85ccd-5d6zn	Ready	<none>	13m	v1.26.5+vmware.2-fips.1
sfo-w01-tkc01-node-pool-gpu-9w5jr-768f85ccd-nxscv	Ready	<none>	13m	v1.26.5+vmware.2-fips.1
sfo-w01-tkc01-vvjgd-2ptdr	Ready	control-plane	15m	v1.26.5+vmware.2-fips.1

sfo-w01-tkc01-vvjgd-2vnx6	Ready	control-plane	11m
v1.26.5+vmware.2-fips.1			
sfo-w01-tkc01-vvjgd-66hxn	Ready	control-plane	13m
v1.26.5+vmware.2-fips.1			

## Install the NVIDIA GPU Operator for Private AI Ready Infrastructure for VMware Cloud Foundation

Install the NVIDIA GPU Operator to automate the management of all NVIDIA software components needed to provision vGPU.

The command prompt steps use example values from the *VMware Cloud Foundation Planning and Preparation Workbook*.

For more information on the deployment and verification procedures, see the Appendix of [Deploying Enterprise-Ready Generative AI on VMware Private AI](#).

### Prerequisites

- Determine the required NVIDIA GPU Operator version according to the GPU model, required features, operating system version, and driver version.  
See [NVIDIA GPU Operator Component Matrix](#) and the [NVIDIA GPU Operator Release Notes](#).
- Verify that you have the NVIDIA vGPU license file, downloaded from the NVIDIA Licensing Portal.
- Verify that you have the API key to pull NVAIE containers from NVIDIA NGC enterprise catalog.
- On the machine with the Kubernetes CLI Tools, install Helm.

### Procedure

- 1 In a command prompt on the machine with the Kubernetes CLI tools, log in to the Tanzu Kubernetes Grid cluster by running `kubectl`.

```
kubectl vsphere login --server 192.168.21.2 --vsphere-username Supervisor_Cluster_User
--tanzu-kubernetes-cluster-namespace Tanzu_Kubernetes_Namespace --tanzu-kubernetes-cluster-
name Tanzu_Kubernetes_Cluster_Name
```

- 2 Create a gpu-operator namespace.

```
kubectl create namespace gpu-operator
```

- 3 Verify that the namespace has been created.

```
kubect get namespaces
```

NAME	STATUS	AGE
default	Active	64m
gpu-operator	Active	6s
kube-node-lease	Active	64m

kube-public	Active	64m
kube-system	Active	64m
secretgen-controller	Active	62m
tkg-system	Active	63m
vmware-system-antrea	Active	62m
vmware-system-auth	Active	62m
vmware-system-cloud-provider	Active	63m
vmware-system-csi	Active	63m
vmware-system-tkg	Active	63m

#### 4 Create a gridd.conf configuration file.

```
sudo touch gridd.conf
```

#### 5 Create a ConfigMap in the gpu-operator namespace.

You can use ConfigMap to store non-confidential data in key-value pairs. You add both the vGPU configuration file and the NVIDIA license token to this ConfigMap.

```
kubectl create configmap licensing-config -n gpu-operator --from-file=<path>/gridd.conf --from-file=<path>/client_configuration_token.tok
```

#### 6 Verify that the contents of the ConfigMap has been successfully populated by describing the ConfigMap.

```
Name:          licensing-config
Namespace:     gpu-operator
Labels:        <none>
Annotations:   <none>

Data
====
gridd.conf:
----

client_configuration_token.tok:
----
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

BinaryData
=====

Events:  <none>
```



## 7 Create a pull secret object in the gpu-operator namespace.

A secret is an object that contains a small amount of sensitive data such as a password, a token, or a key. Such information might otherwise be put in a pod specification or in a container image. Using a secret object means that you do not need to include confidential data in your application code. We will use this secret object to pull the required images from NVIDIA NGC registry.

```
export REGISTRY_SECRET_NAME=<your-ngc-secret>
export PRIVATE_REGISTRY=nvcr.io/nvaie
kubectl create secret docker-registry ${REGISTRY_SECRET_NAME} \
--docker-server=${PRIVATE_REGISTRY} \
--docker-username='${soauthtoken}' \
--docker-password=${NGC_API_KEY} \
--docker-email='YOUREMAIL' \
-n gpu-operator
```

## 8 Add the NVAIE Helm repository where the password is the NGC API key for accessing the NVIDIA NGC catalog.

```
helm repo add nvaie https://helm.ngc.nvidia.com/nvaie \ --username='${soauthtoken}' --
password=${NGC_API_KEY} \ && helm repo update
```

## 9 Set the required Pod Security admission policy on the gpu-operator namespace.

```
kubectl label --overwrite ns gpu-operator pod-security.kubernetes.io/enforce=privileged
```

## 10 Install NVIDIA GPU Operator by using Helm.

```
helm install --wait gpu-operator nvaie/gpu-operator-4-2 -n gpu-operator --
set driver.repository=nvcr.io/nvaie --set operator.repository=nvcr.io/nvaie --set
driver.imagePullPolicy=Always --set migStrategy=mixed --set driver.rdma.enabled=True
```

## 11 Verify that the GPU Operator pods are running.

```
kubectl get pods -n gpu-operator
```

NAME	READY	STATUS
RESTARTS    AGE		
gpu-feature-discovery-9zv52	1/1	Running
0            7d6h		
gpu-feature-discovery-pv4p4	1/1	Running
0            7d6h		
gpu-feature-discovery-zms5s	1/1	Running
0            55d		
gpu-operator-dc844b566-w9mj1	1/1	Running
0            55d		
gpu-operator-node-feature-discovery-master-79bc547944-rzp4v	1/1	Running
0            55d		
gpu-operator-node-feature-discovery-worker-7m5ht	1/1	Running
0            7d6h		
gpu-operator-node-feature-discovery-worker-1lz7k	1/1	Running

0	7d6h		
gpu-operator-node-feature-discovery-worker-zk7mt		1/1	Running
0	55d		
nvidia-container-toolkit-daemonset-pswbb		1/1	Running
0	7d6h		
nvidia-container-toolkit-daemonset-tlqfn		1/1	Running
0	7d6h		
nvidia-container-toolkit-daemonset-zm48q		1/1	Running
0	55d		
nvidia-cuda-validator-fmwsh		0/1	Completed
0	55d		
nvidia-cuda-validator-qdz6r		0/1	Completed
0	7d6h		
nvidia-cuda-validator-x7mkj		0/1	Completed
0	7d6h		
nvidia-dcgm-exporter-c7dwd		1/1	Running
0	7d6h		
nvidia-dcgm-exporter-mc4x8		1/1	Running
0	55d		
nvidia-dcgm-exporter-xnpvp		1/1	Running
0	7d6h		
nvidia-device-plugin-daemonset-92pf4		1/1	Running
0	7d6h		
nvidia-device-plugin-daemonset-m276d		1/1	Running
0	55d		
nvidia-device-plugin-daemonset-v62nj		1/1	Running
0	7d6h		
nvidia-device-plugin-validator-8d2jr		0/1	Completed
0	7d6h		
nvidia-device-plugin-validator-cfkr1		0/1	Completed
0	7d6h		
nvidia-device-plugin-validator-wltdz		0/1	Completed
0	55d		
nvidia-driver-daemonset-7g6nj		1/1	Running
0	7d6h		
nvidia-driver-daemonset-8bwsx		1/1	Running
0	55d		
nvidia-driver-daemonset-fhz56		1/1	Running
0	7d6h		
nvidia-operator-validator-5zs4b		1/1	Running
0	55d		
nvidia-operator-validator-hp5dt		1/1	Running
0	7d6h		
nvidia-operator-validator-grfj8		1/1	Running
0	7d6h		

## 12 Verify that the license is valid.

```
ctnname=`kubectl get pods -n gpu-operator | grep driver-daemonset | head -1 | cut -d " " -f1`
```

```
kubectl -n gpu-operator exec -it $ctnname -- /bin/bash -c "/usr/bin/nvidia-smi -q | grep -i lic"
```

```
Defaulted container "nvidia-driver-ctr" out of: nvidia-driver-ctr, k8s-driver-manager (init)
```

```
vGPU Software Licensed Product
```

```
License Status : Licensed (Expiry: 2024-2-28 21:22:44 GMT)
```

```
Applications Clocks
```

```
Default Applications Clocks
```

```
Clock Policy
```

## Install the NVIDIA Network Operator for Private AI Ready Infrastructure for VMware Cloud Foundation

The NVIDIA Network Operator leverages Kubernetes custom resources and the Kubernetes Operator framework to optimize the networking for vGPU.

The command prompt steps use example values from the *VMware Cloud Foundation Planning and Preparation Workbook*.

For more information on the deployment and verification procedures, see the Appendix of [Deploying Enterprise-Ready Generative AI on VMware Private AI](#).

### Prerequisites

- Determine the required NVIDIA Network Operator version.  
See the [NVIDIA Network Operator documentation](#).
- Provide an RDMA NIC on each host in the GPU-enabled VI workload domain.

### Procedure

- 1 In a command prompt on the machine with the Kubernetes CLI tools, log in to the Tanzu Kubernetes Grid cluster by running `kubectl`.

```
kubectl vsphere login --server 192.168.21.2 --vsphere-username Supervisor_Cluster_Admin --tanzu-kubernetes-cluster-namespace Tanzu_Kubernetes_Namespace --tanzu-kubernetes-cluster-name Tanzu_Kubernetes_Cluster_Name --insecure-skip-tls-verify
```

- 2 List all the nodes in the cluster.

```
kubectl get nodes
```

NAME	STATUS	ROLES	AGE	VERSION
sfo-w01-tkc01-node-pool-gpu-9w5jr-768f85ccd-5d6zn	Ready	<none>	18h	
v1.26.5+vmware.2-fips.1				

```
sfo-w01-tkc01-node-pool-gpu-9w5jr-768f85ccd-nxscv Ready <none> 18h
v1.26.5+vmware.2-fips.1
sfo-w01-tkc01-vvjgd-2ptdr Ready control-plane 18h
v1.26.5+vmware.2-fips.1
sfo-w01-tkc01-vvjgd-2vnx6 Ready control-plane 18h
v1.26.5+vmware.2-fips.1
sfo-w01-tkc01-vvjgd-66hxn Ready control-plane 18h
v1.26.5+vmware.2-fips.1
```

### 3 Label the worker node roles.

Do not change the Control Plane label.

```
kubectl label node sfo-w01-tkc01-node-pool-gpu-9w5jr-768f85ccd-5d6zn sfo-w01-tkc01-node-
pool-gpu-9w5jr-768f85ccd-nxscv node-role.kubernetes.io/worker=worker
```

### 4 Verify that the worker nodes are properly labelled.

```
kubectl get nodes
```

NAME	STATUS	ROLES	AGE	VERSION
sfo-w01-tkc01-node-pool-gpu-9w5jr-768f85ccd-5d6zn	Ready	worker	18h	
v1.26.5+vmware.2-fips.1				
sfo-w01-tkc01-node-pool-gpu-9w5jr-768f85ccd-nxscv	Ready	worker	18h	
v1.26.5+vmware.2-fips.1				
sfo-w01-tkc01-vvjgd-2ptdr	Ready	control-plane	18h	
v1.26.5+vmware.2-fips.1				
sfo-w01-tkc01-vvjgd-2vnx6	Ready	control-plane	18h	
v1.26.5+vmware.2-fips.1				
sfo-w01-tkc01-vvjgd-66hxn	Ready	control-plane	18h	
v1.26.5+vmware.2-fips.1				

### 5 Create a nvidia-network-operator namespace.

```
kubectl create namespace nvidia-network-operator
```

### 6 Verify if namespace has been created properly.

```
kubectl get namespaces
```

NAME	STATUS	AGE
default	Active	18h
gpu-operator	Active	17h
kube-node-lease	Active	18h
kube-public	Active	18h
kube-system	Active	18h
nvidia-network-operator	Active	5s
secretgen-controller	Active	18h
tkg-system	Active	18h
vmware-system-antrea	Active	18h

vmware-system-auth	Active	18h
vmware-system-cloud-provider	Active	18h
vmware-system-csi	Active	18h
vmware-system-tkg	Active	18h

- 7 To be able to pull the network operator images during Helm installation, create a secret on the `nvidia-network-operator` namespace.

```
kubectl create secret docker-registry ngc-image-secret -n nvidia-network-operator --docker-server=nvcr.io --docker-username='$oauthtoken' --docker-password='YOUR NVIDIA API KEY' --docker-email='YOUR NVIDIA NGC EMAIL'
```

- 8 Create the `values.yaml` file that for the Network Operator deployment.

```
nfd:
  enabled: true
sriovNetworkOperator:
  enabled: false
# NicClusterPolicy CR values:
deployCR: true
ofedDriver:
  deploy: true

rdmaSharedDevicePlugin:
  deploy: true
  imagePullSecrets: <ngc-image-secret>

sriovDevicePlugin:
  deploy: true
  imagePullSecrets: <ngc-image-secret>
  resources:
    - name: hostdev
      vendors: [15b3]
secondaryNetwork:
  deploy: true
  multus:
    deploy: true
  cniPlugins:
    deploy: true
  ipamPlugin:
    deploy: true
```

- 9 Add the NVIDIA NGC Helm repository.

```
helm repo add nvidia https://helm.ngc.nvidia.com/nvidia \ --username='$oauthtoken' --password=${NGC_API_KEY} \ && helm repo update
```

- 10 Install the NVIDIA Network Operator, passing `values.yaml` file as a parameter.

```
helm install network-operator nvidia/network-operator -n nvidia-network-operator --create-namespace --version v23.5.0 -f values.yaml --debug
```

Wait until the operation completes.

**11** Verify that the Network Operator pods are running.

```
kubectl -n nvidia-network-operator get pods
```

NAME	READY	STATUS
RESTARTS    AGE		
cni-plugins-ds-jqd8f	1/1	Running
0            4d23h		
cni-plugins-ds-pqq4x	1/1	Running
0            4d23h		
kube-multus-ds-2h7sm	1/1	Running
0            4d23h		
kube-multus-ds-hwfdg	1/1	Running
0            4d23h		
mofed-ubuntu20.04-ds-27815	1/1	Running
0            4d23h		
mofed-ubuntu20.04-ds-4zth4	1/1	Running
0            4d23h		
network-operator-57cf95446-722tl	1/1	Running
0            4d23h		
network-operator-node-feature-discovery-master-848d8b8cdf-667wh	1/1	Running
0            4d23h		
network-operator-node-feature-discovery-master-worker-h5x74	1/1	Running
0            4d23h		
network-operator-node-feature-discovery-master-worker-j5stf	1/1	Running
0            4d23h		
rdma-shared-dp-ds-7g6s5	1/1	Running
0            4d23h		
rdma-shared-dp-ds-b6pgc	1/1	Running
0            4d23h		
rdma-shared-dp-ds-j2m84	0/1	Running
0            4d23h		
sriov-device-plugin-22cv9	0/1	Running
0            4d23h		
sriov-device-plugin-6ktpf	0/1	Running
0            4d23h		
whereabouts-c1951	0/1	Running
0            4d23h		
whereabouts-tkw8t	0/1	Running
0            4d23h		

**12** Apply a [host-device-net.yaml](#) file.

```
kubectl apply -f host-device-net.yaml
```

`host-device-net.yaml` is the configuration file for Kubernetes networking deployment. The YAML file defines the creation of a `hostdev` custom resource that can be requested while creating a pod. Keep in mind that the Whereabouts IPAM configuration can be customized according to your needs.

**13** Verify that the custom resource was created successfully.

```
kubectl get HostDeviceNetwork
```

NAME	STATUS	AGE
hostdev-net	ready	2024-02-28T17:22:38Z

**14** Verify that the nvidia-peermem-ctr container was successfully loaded in the nvidia-peermem Kernel module.

```
kubectl logs -n gpu-operator ds/nvidia-driver-daemonset -c nvidia-peermem-ctr
```

```
Found 4 pods, using pod/nvidia-driver-daemonset-66rnx
DRIVER_ARCH is x86_64
waiting for mellanox ofed and nvidia drivers to be installed
waiting for mellanox ofed and nvidia drivers to be installed
waiting for mellanox ofed and nvidia drivers to be installed
waiting for mellanox ofed and nvidia drivers to be installed
waiting for mellanox ofed and nvidia drivers to be installed
waiting for mellanox ofed and nvidia drivers to be installed
waiting for mellanox ofed and nvidia drivers to be installed
waiting for mellanox ofed and nvidia drivers to be installed
successfully loaded nvidia-peermem module, now waiting for signal
```

## Adding VMware Private AI Foundation with NVIDIA to Private AI Ready Infrastructure for VMware Cloud Foundation

Implementing VMware Private AI Foundation with NVIDIA on top of the private AI infrastructure components includes creating vector databases by using VMware Data Services Manager, and deploying VMs for AI development based on NVIDIA DL workloads images and GPU-enabled Tanzu Kubernetes Grid (TKG) clusters for running NVIDIA NGC container images.

You can deploy and configure VMware Private AI Foundation with NVIDIA according to two implementation models - a cloud-connected or a disconnected environment.

For information on the VMware Private AI Foundation with NVIDIA design, see [Chapter 3 Detailed Design for VMware Private AI Foundation with NVIDIA for Private AI Ready Infrastructure for VMware Cloud Foundation](#).

### Prerequisites

To complete the implementation of *VMware Private AI Foundation with NVIDIA for Private AI Ready Infrastructure for VMware Cloud Foundation* validated solution, verify that your system fulfills the following prerequisites.

**Table 5-8. Prerequisites for Adding VMware Private AI Foundation with NVIDIA to Private AI Ready Infrastructure for VMware Cloud Foundation**

Category	Prerequisite
Environment	<ul style="list-style-type: none"> <li>■ Verify that your VMware Cloud Foundation version is listed in the <a href="#">Support Matrix</a> for this solution.</li> <li>■ Verify that you configure your environment according to <a href="#">Before You Apply This Guidance</a>.</li> <li>■ Verify that you have implemented the private AI ready infrastructure in your VMware Cloud Foundation instance according to <a href="#">Chapter 5 Implementation of Private AI Ready Infrastructure for VMware Cloud Foundation</a>.</li> <li>■ Verify that you capture all parameters for the <i>Private AI Ready Infrastructure</i> tab of the <a href="#">VMware Cloud Foundation Planning and Preparation Workbook</a>.</li> <li>■ Verify that your VMware Cloud Foundation instance is healthy and fully operational. See <a href="#">VMware Cloud Foundation Operations Guide</a>.</li> </ul>
NVIDIA NGC API key	Verify that you have an API key for access to the <code>nvcf.io</code> private registry.
Docker client	Verify that your system has a machine with Docker installed.
VMware Tanzu Network account	Verify that you have an account for <a href="#">VMware Tanzu Network</a> so that you can download VMware Data Services Manager software.

## What to read next

### Procedure

#### 1 [Create AI Self-Service Catalog Items in VMware Aria Automation for Private AI Ready Infrastructure for VMware Cloud Foundation](#)

Use the catalog setup wizard in VMware Aria Automation to create catalog items for GPU-enabled deep learning VMs and for GPU-enabled Tanzu Kubernetes Grid clusters with NVIDIA GPU Operator deployed and configured.

#### 2 [Deploy VMware Data Services Manager for Private AI Ready Infrastructure for VMware Cloud Foundation](#)

You deploy VMware Data Services Manager as a client plug-in in the vCenter Server instance for the VI workload domain.

#### 3 [Create a Content Library with Deep Learning VM Images for Private AI Ready Infrastructure for VMware Cloud Foundation](#)

Deep learning VM images in VMware Private AI Foundation with NVIDIA are distributed in a shared content library published by VMware. You use a content library to pull specific VM images in your VI workload domain during VM deployment.

#### 4 [Additional Setup of VMware Private AI Foundation with NVIDIA for Private AI Ready Infrastructure in a Disconnected VMware Cloud Foundation Environment](#)

In a disconnected environment with VMware Private AI Foundation with NVIDIA, you must perform additional steps to make the container and deep learning VM images, TKr images, and NVIDIA NGC containers available.



## 5 Deploy AI Workloads by Using VMware Aria Automation for Private AI Ready Infrastructure for VMware Cloud Foundation

After you set up VMware Private AI Foundation with NVIDIA, you can start deploying deep learning VMs and Tanzu Kubernetes Grid clusters, and PostgreSQL databases from Automation Service Broker.

## Create AI Self-Service Catalog Items in VMware Aria Automation for Private AI Ready Infrastructure for VMware Cloud Foundation

Use the catalog setup wizard in VMware Aria Automation to create catalog items for GPU-enabled deep learning VMs and for GPU-enabled Tanzu Kubernetes Grid clusters with NVIDIA GPU Operator deployed and configured.

The Private AI Automation Services wizard in VMware Aria Automation can be run multiple times for the following scenarios:

- Add a new vCenter Cloud Account and an associated Supervisor.
- Accommodate a change in your NVIDIA AI Enterprise license, including license server and client configuration token.
- Add or remove VM Classes and content libraries.
- Create the catalog items for a new project.

### Prerequisites

- VMware Aria Automation has been deployed with version 8.16+ refer to (LINK TO ILIANA'S ARIA DEPLOYMENT) and [Implementation of Private Cloud Automation for VMware Cloud Foundation](#).

### Procedure

- 1 Log in to VMware Aria Automation as an administrator.
- 2 Under **My Services**, click **Quickstart**.
- 3 In the **Private AI Automation Services** card, click **Start**.
- 4 Add a cloud account.
  - a Select the vCenter Server for the VI workload domain.
  - b Select the Supervisor.
  - c Select **NVIDIA Cloud License Service (CLS)** for connected environments or **Self hosted - NVIDIA Delegated License Service (DLS)** for disconnected environments.

Refer to [NVIDIA's documentation](#) on how to setup the Delegated License Service (DLS) for disconnected environments.

- d For connected environments, enter the contents of your NVIDIA Client Configuration token file (.tok) and your NVIDIA Licensing Portal API Key, and click **Next Step**.
  - e For disconnected environments, enter the contents of your NVIDIA Client Configuration token file (.tok) and a vGPU guest driver local URL, and click **Next Step**.
- 5 Configure the catalog items and click **Next Step**.
- a Select the VM image name to use for provisioning deep learning VMs.
  - b Select the GPU-enabled and non-GPU enabled VM classes for your deep learning VMs and Tanzu Kubernetes Grid cluster nodes.
  - c Select the storage policy that you created earlier .
  - d For connected environments, select **Cloud - NVIDIA NGC Catalog**, for disconnected environment, **Local - self-hosted registry** and click **Next Step**.
- 6 Select the users and user groups to have access to the project that will be created by the Quickstart wizard, and click **Next Step**.
- See [Personas in Private AI Ready Infrastructure for VMware Cloud Foundation](#).
- 7 Review the configuration and click **Run Quickstart**.
- 8 Verify that the catalog items are created successfully.
- a Select **Service Broker > Catalog**.
  - b Verify that the catalog contains two items - **AI Kubernetes Cluster** and **AI Workstation**.

## Deploy VMware Data Services Manager for Private AI Ready Infrastructure for VMware Cloud Foundation

You deploy VMware Data Services Manager as a client plug-in in the vCenter Server instance for the VI workload domain.

In a VMware Private AI Foundation with NVIDIA environment, you use VMware Data Services Manager to create PostgreSQL databases with the pgvector extension.

### Procedure

- ◆ Deploy the plug-in for the VMware Data Services Manager by using the vSphere Client.

See [Deploying the VMware Data Services Manager Plugin in vSphere Client](#).

## Create a Content Library with Deep Learning VM Images for Private AI Ready Infrastructure for VMware Cloud Foundation

Deep learning VM images in VMware Private AI Foundation with NVIDIA are distributed in a shared content library published by VMware. You use a content library to pull specific VM images in your VI workload domain during VM deployment.

In a disconnected environment, as a cloud administrator, you must manually upload the required VM images to the content library.

**Procedure**

- 1 In a disconnected environment, download the deep learning VM images from <https://packages.vmware.com/dl-vm/>.  
For each image, download the relevant .ovf, .vmdk, .mf, and .cert files.
- 2 Log in to the VI workload domain vCenter Server at [https://<vcenter\\_server\\_fqdn>/ui](https://<vcenter_server_fqdn>/ui) as **administrator@vsphere.local**.
- 3 From the vSphere Client Menu, select **Content Libraries**.
- 4 On the **Content Libraries** page, click **Create**.
- 5 On the **Name and location** page of the **New Content Libraries** wizard, enter a library name and select the vCenter Server instance for the VI workload domain, and click **Next**.
- 6 On the **Configure content library** page, configure the content library downloads and click **Next**.

**Table 5-9. Content Library Configuration for a Connected Environment**

Setting	Value
Subscribed content library	Selected
Subscription URL	<a href="https://packages.vmware.com/dl-vm/lib.json">https://packages.vmware.com/dl-vm/lib.json</a>
Enable authentication	Deselected
Download content	<i>According to the need of your organization.</i>

**Table 5-10. Content Library Configuration for a Disconnected Environment**

Setting	Value
Local content library	Selected
Enable publishing	<i>According to the need of your organization.</i>
Enable authentication	<i>According to the need of your organization.</i>

- 7 In a connected environment, confirm the SSL certificate thumbprint.  
The SSL certificate thumbprint is stored on your system until you delete the subscribed content library from the inventory.
- 8 On the **Apply Security Policy** page, select **Apply security policy** and click **Next**.
- 9 On the **Add storage** page, select the datastore in the VI workload domain for the VM templates.
- 10 Review the details on content library creation and click **Finish**.

- 11 In a disconnected environment, import the downloaded deep learning VM images in the content library.
  - a On the **Content Libraries** page in the vSphere Client, select **Actions > Import item**.
  - b Locate the VM template files from your local file system and click **Import**.

## Additional Setup of VMware Private AI Foundation with NVIDIA for Private AI Ready Infrastructure in a Disconnected VMware Cloud Foundation Environment

In a disconnected environment with VMware Private AI Foundation with NVIDIA, you must perform additional steps to make the container and deep learning VM images, TKr images, and NVIDIA NGC containers available.

### Procedure

- 1 [Configure a Replicated Harbor Instance for Private AI Ready Infrastructure in a Disconnected VMware Cloud Foundation Environment](#)  
 As a cloud administrator, by setting up replication between an Internet-connected Harbor instance to a Harbor instance that has no Internet connectivity, you can to pull images from different container registries, such as NVIDIA NGC, without making your private AI infrastructure vulnerable.
- 2 [Upload Container Images to a Private Harbor Registry for Private AI Ready Infrastructure in a Disconnected VMware Cloud Foundation Environment](#)  
 VMware Private AI Foundation with NVIDIA requires access to the container images on the NVIDIA NGC catalog. For disconnected environments, as a DevOps engineer, you must manually upload these images to the Harbor Supervisor Service.
- 3 [Configure a Content Library with Ubuntu TKr for Private AI Ready Infrastructure in a Disconnected VMware Cloud Foundation Environment](#)  
 As a cloud administrator, configure a local vSphere content library to store TKr (Tanzu Kubernetes releases) to be used as the base images for Tanzu Kubernetes Grid clusters and associate the library with the Supervisor.

### Configure a Replicated Harbor Instance for Private AI Ready Infrastructure in a Disconnected VMware Cloud Foundation Environment

As a cloud administrator, by setting up replication between an Internet-connected Harbor instance to a Harbor instance that has no Internet connectivity, you can to pull images from different container registries, such as NVIDIA NGC, without making your private AI infrastructure vulnerable.

## Prerequisites

Verify that the following prerequisites are in place:

- A working Harbor instance that has access to the Internet. This harbor instance can be running as a Supervisor Service or as a standalone Harbor instance.
- A working Harbor instance running as a Supervisor Service to serve as the disconnected instance.
- Both Harbor instances must be able to reach each other over the network over ports 443 and 80. Notary enablement is not supported.

## Procedure

- 1 Log in to the Internet-connected Harbor instance as a system administrator.
- 2 On the **Projects** page, click **New Project**.
- 3 In the **New Project** dialog box, enter a project name and activate **Proxy Cache**, and click **OK**.
- 4 Navigate to the **Administration > Registries** page.
- 5 Add the NVIDIA NGC container registry.
  - a Click **New Endpoint** and fill in the following information.

Setting	Value
Provider	Docker Registry
Name	NGC Registry
Endpoint URL	<a href="https://nvcr.io">https://nvcr.io</a>
Access ID	\$oauthtoken
Access Secret	<i>Your NVIDIA NGC API key</i>
Verify Remove Cert	Selected

- b Test the connection and click **OK**.

- 6 Create a second endpoint on the same Internet-facing Harbor instance to add the target disconnected Harbor instance.

- a Click **New Endpoint** and provide the following information.

Setting	Value
Provider	Harbor
Name	Disconnected Harbor
Endpoint URL	<i>FDQN of the remote Harbor Instance</i>
Access ID	<i>Local user account or AD credentials</i>
Access Secret	<i>Password for the access ID</i>
Verify Remove Cert	Deselected for self-signed certificates. Otherwise, selected.

- b Test the connection and click **OK**.

- 7 Navigate to **Administration > Replications** page.

- 8 Click **New Replication Rule**, provide the following information and click **Save**.

Setting	Value
Name	<i>A name and description for the replication rule</i>
Replication mode	Push-based
Source resource filter	<i>Any desired filters such as tags</i>
Destination registry	Registry that you created in <a href="#">Step 6</a> .
Destination	<i>Name of the namespace in which to replicate resources.</i> If you do not enter a namespace, resources are placed in the same namespace as in the source registry.
Trigger mode	<i>How and when to run the rule</i>
Bandwidth	<i>Network bandwidth for each execution of the replication rule if required</i>

- 9 On the machine running the Docker client, to verify the images are available on both the Internet-connected and the disconnected Harbor instances, log in to each registry and run the `docker image list` command.

## Upload Container Images to a Private Harbor Registry for Private AI Ready Infrastructure in a Disconnected VMware Cloud Foundation Environment

VMware Private AI Foundation with NVIDIA requires access to the container images on the NVIDIA NGC catalog. For disconnected environments, as a DevOps engineer, you must manually upload these images to the Harbor Supervisor Service.

## Procedure

- 1 Install the Harbor Supervisor Service certificate on the client machine.
  - a Log into Harbor as an administrator.
  - b Navigate to **Administration > Configuration** and click the **System Settings** tab.
  - c To download the registry root certificate, click **Download**.
  - d Copy the `ca.crt` file that you downloaded to the client machine with Docker installed.
  - e On the client machine, the `/etc/docker/certs.d/private-registry-FQDN/` folder, create a directory path for the private registry.
  - f Move the `ca.crt` file to the folder you created.
  - g Restart the docker daemon.

```
sudo systemctl restart docker.service
```

- h To verify that the TLS certificate is trusted, log in to Harbor by using Docker.

```
docker login harborfqdn.example.com
```

- 2 Log in to the NVIDIA NGC registry by using Docker with a user name `$oauthtoken` and the NGC API key.

```
docker login nvcr.io
```

- 3 Pull the desired image from the NVIDIA NGC registry.

For example, to pull a Triton Inference Server image, run this command.

```
docker pull nvcr.io/nvidia/tritonserver:24.04-py3-igpu
```

- 4 Export the container image to a `.tar` file.

```
docker save > tritonserver.tar nvcr.io/nvidia/tritonserver:24.04-py3-igpu
```

- 5 Copy the `.tar` file to the machine that has access to the disconnected Harbor Supervisor Service

- 6 Load the container image on the machine with network access to the Harbor Supervisor Service.

```
docker load < tritonserver.tar
```

- 7 Tag the image.

```
docker tag image:tag <harbor_address>/paif-n/tritonserver:24.04-py3-igpu
```

- 8 Push the image to a project in Harbor.

```
docker push <harbor_address>/paif-n/tritonserver:24.04-py3-igpu
```

## Configure a Content Library with Ubuntu TKr for Private AI Ready Infrastructure in a Disconnected VMware Cloud Foundation Environment

As a cloud administrator, configure a local vSphere content library to store TKr (Tanzu Kubernetes releases) to be used as the base images for Tanzu Kubernetes Grid clusters and associate the library with the Supervisor.

### Procedure

- 1 Download an Ubuntu-based TKr from <https://wp-content.vmware.com/v2/latest/> according to the required Kubernetes version, for example, `v1.26.5---vmware.2-fips.1-tkg.1`.
- 2 Log in to the VI workload domain vCenter Server at `https://<vcenter_server_fqdn>/ui` as `administrator@vsphere.local`.
- 3 From the vSphere Client Menu, select **Content Libraries**.
- 4 On the **Content Libraries** page, click **Create**.
- 5 On the **Name and location** page of the **New Content Library** wizard, enter a library name and select the vCenter Server instance for the VI workload domain, and click **Next**.
- 6 On the **Configure content library** page, configure the content library downloads and click **Next**.

Setting	Value
Local content library	Selected
Enable publishing	<i>According to the need of your organization.</i>
Enable authentication	<i>According to the need of your organization.</i>

- 7 On the **Apply Security Policy** page, select **Apply security policy** and click **Next**.
- 8 On the **Add storage** page, select the datastore in the VI workload domain for the VM templates.
- 9 Review the details on content library creation and click **Finish**.
- 10 Add the content library to the Supervisor so that it is available to all the namespaces there.
  - a From the vSphere Client Menu, select **Workload Management**.
  - b Navigate to the Supervisor for AI workloads.
  - c On the **Configure** tab, select **Supervisor > General**.



- d On the **General** page that appears, expand **Tanzu Kubernetes Grid Service**, and next to **Content Library**, click **Edit**.
- e Select the content library with the TKr images and click **OK**.

## Deploy AI Workloads by Using VMware Aria Automation for Private AI Ready Infrastructure for VMware Cloud Foundation

After you set up VMware Private AI Foundation with NVIDIA, you can start deploying deep learning VMs and Tanzu Kubernetes Grid clusters, and PostgreSQL databases from Automation Service Broker.

### Procedure

#### 1 [Deploy a Deep Learning VM from the VMware Aria Automation Self-Service Catalog for Private AI Ready Infrastructure for VMware Cloud Foundation](#)

As a data scientist, you can use the Automation Service Broker catalog to deploy a GPU-enabled deep learning VM with a pre-configured DL workload. An AI Workstation is based on the Deep Learning VM images and Software bundles.

#### 2 [Provision a Tanzu Kubernetes Grid Cluster from the VMware Aria Automation Self-Service Catalog for Private AI Ready Infrastructure for VMware Cloud Foundation](#)

As a DevOps engineer, use the Automation Service Broker catalog to provision a GPU-enabled Tanzu Kubernetes Grid cluster with the NVIDIA GPU Operator configured and licensed, and ready to run GPU-enabled container workloads.

#### 3 [Create a Vector Database Catalog Item in VMware Aria Automation for RAG Workloads for Private AI Ready Infrastructure for VMware Cloud Foundation](#)

In VMware Private AI Foundation with NVIDIA, as a cloud administrator, add a catalog item for provisioning databases in VMware Data Services Manager to Automation Service Broker in VMware Aria Automation.

#### 4 [Deploy a Vector Database by Using a Self-Service Catalog Item in VMware Aria Automation for RAG Workloads for Private AI Ready Infrastructure for VMware Cloud Foundation](#)

In VMware Private AI Foundation with NVIDIA, as a data scientist or a DevOps engineer, you can deploy a PostgreSQL database from VMware Aria Automation on VMware Data Services Manager by using a catalog item in Automation Service Broker.

### Deploy a Deep Learning VM from the VMware Aria Automation Self-Service Catalog for Private AI Ready Infrastructure for VMware Cloud Foundation

As a data scientist, you can use the Automation Service Broker catalog to deploy a GPU-enabled deep learning VM with a pre-configured DL workload. An AI Workstation is based on the Deep Learning VM images and Software bundles.

### Procedure

- 1 Log in to VMware Aria Automation at **`https://<aria_automation_cluster_fqdn>/csp/gateway/portal`** as a user that has access to the project for Private AI.

- 2 On the main navigation bar, click **Services**.
- 3 On the My Services page, click **Service Broker**.
- 4 On the **Consume** tab, on the navigation bar, click Catalog.
- 5 In the **AI Workstation** card, click **Request**.
- 6 Configure the following settings and click **Submit**.

Setting	Value
Version	<i>Version of the catalog item</i>
Project	<i>Project where you want it to be deployed and the name for the deployment</i>
Deployment name	<i>Name for the resulting deployment</i>
VM Class	GPU-Enabled VM class in the VM Service for the deep learning VM. You can use the VM class for the Tanzu Kubernetes Grid cluster you created earlier in the implementation flow.
User Password	<i>Initial password for the vmware user on the deep learning VM</i>
SSH public keys	Import SSH Keys if needed
Software bundle	DL workload from NVIDIA
Custom cloud-init	Selected if you plan to add customization to cloud-init. See <a href="#">Cloud config examples</a> .
NVIDIA NGC API key	The API key for access to the NVIDIA NGC registry.

- 7 Monitor the deployment process.
  - a On the **Consume** tab, click **Deployments > Deployments**.
  - b Click the name of deployment and then click the **History** tab.
- 8 After the deployment is completed, review the details to access applications installed on the deep learning VM or the deep learning VM over SSH.
  - a On the **Consume** tab, click **Deployments > Deployments**.
  - b Click the name of deployment and then click the **Overview** tab.

**Figure 5-1. Example of Access Details for a PyTorch Deep Learning VM**

AI workstation
PyTorch
The PyTorch NGC Container is optimized for GPU acceleration, and contains a validated set of libraries that enable and optimize GPU performance. This container also contains software for accelerating ETL (DALL, RAPIDS), Training (cuDNN, NCCL), and inference (TensorRT) workloads. <a href="https://catalog.ngc.nvidia.com/orgs/nvidia/containers/pytorch">https://catalog.ngc.nvidia.com/orgs/nvidia/containers/pytorch</a>
Applications
• <a href="#">JupyterLab</a>
Services
• SSH - 10.203.84.5:22
Workstation VM
• ssh vmware@10.203.84.5
• Data storage mounted at /opt/data
• Manage VM and Supervisor Namespace via <a href="#">Cloud Consumption interface</a>

## 9 Verify the deployment.

- a Log in to the deep learning VM over SSH.
- b Verify the vGPU guest driver installation by running the following command.

```
tail -f /var/log/nvidia-installer.log
```

- c Verify the execution of cloud-init by running the following command.

```
tailf -f /var/log/dl.log
```

## Provision a Tanzu Kubernetes Grid Cluster from the VMware Aria Automation Self-Service Catalog for Private AI Ready Infrastructure for VMware Cloud Foundation

As a DevOps engineer, use the Automation Service Broker catalog to provision a GPU-enabled Tanzu Kubernetes Grid cluster with the NVIDIA GPU Operator configured and licensed, and ready to run GPU-enabled container workloads.

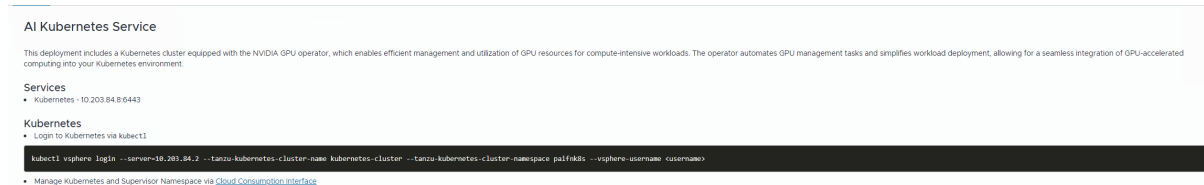
### Procedure

- 1 Log in to VMware Aria Automation at [https://<aria\\_automation\\_cluster\\_fqdn>/csp/gateway/portal](https://<aria_automation_cluster_fqdn>/csp/gateway/portal).
- 2 On the main navigation bar, click **Services**.
- 3 On the My Services page, click **Service Broker**.
- 4 On the **Consume** tab, on the navigation bar, click Catalog.
- 5 In the **AI Kubernetes Cluster** card, click **Request**.
- 6 Configure the following settings and click **Submit**.

Setting	Value
Version	<i>Version of the catalog item</i>
Project	<i>Project where you want it to be deployed and the name for the deployment</i>
Deployment Name	<i>Name for the resulting deployment</i>
Control plane > Node count	<i>Number of control plane nodes</i>
Control plane > VM Class	VM class, based on CPU and memory requirements, for the control plane nodes
Workers > Node count	<i>Number of worker nodes</i>
Workers > VM Class	GPU-enabled VM class for the worker nodes
NVIDIA AI enterprise API key	<i>The API key for access to the NVIDIA NGC registry.</i> The API key is required to download the Helm charts of NVIDIA GPU Operator.

- 7 Monitor the deployment process.
  - a On the **Consume** tab, click **Deployments > Deployments**.
  - b Click the name of deployment and then click the **History** tab.
- 8 After the deployment is completed, review the details to access the AI-ready TKG cluster by using `kubectl`.

Figure 5-2. Example of Access Details for an AI -Ready TKG Cluster



- 9 Verify the NVIDIA GPU Operator deployment by running the following `kubectl` command on the Supervisor.

```
kubectl get pods -n gpu-operator
```

Figure 5-3. Example of Successful NVIDIA GPU Operator deployment

```
root@photon-machine [ ~ ]# kubectl get pods -n gpu-operator
```

NAME	READY	STATUS	RESTARTS	AGE
gpu-feature-discovery-t6t7v	1/1	Running	0	9m1s
gpu-operator-5c9fb6954b-9xrdk	1/1	Running	0	9m31s
gpu-operator-node-feature-discovery-gc-859bbc78cc-rx7n8	1/1	Running	0	9m31s
gpu-operator-app-node-feature-discovery-master-d7869b4b8-z9kxm	1/1	Running	0	9m31s
gpu-operator-app-node-feature-discovery-worker-hfkpp	1/1	Running	0	9m31s
gpu-operator-app-node-feature-discovery-worker-rzhcv	1/1	Running	0	9m31s
nvidia-container-toolkit-daemonset-tlb79	1/1	Running	0	9m1s
nvidia-cuda-validator-2ggqr	0/1	Completed	0	6m41s
nvidia-dcgm-exporter-zdzgt	1/1	Running	0	9m1s
nvidia-device-plugin-daemonset-7xj27	1/1	Running	0	9m1s
nvidia-driver-daemonset-x6jbv	1/1	Running	0	9m16s
nvidia-operator-validator-kn5dh	1/1	Running	0	9m1s

- 10 Verify the NVIDIA license by running the following command.

You run a Bash shell into the `nvidia-driver-daemonset` pod where you can run the `nvidia-smi` command with the `-q` argument to check the license status.

```
kubectl exec nvidia-driver-daemonset-x6jbv -n gpu-operator -it -- /bin/bash
```

Figure 5-4. Example of a Licensed AI Kubernetes Cluster

```
root@photon-machine [ ~ ]# kubectl exec nvidia-driver-daemonset-x6jbv -n gpu-operator -it -- /bin/bash
root@nvidia-driver-daemonset-x6jbv:/drivers# nvidia-smi -q |grep Licensed
vGPU Software Licensed Product
License Status : Licensed (Expiry: 2024-4-18 18:26:28 GMT)
root@nvidia-driver-daemonset-x6jbv:/drivers#
```

## Create a Vector Database Catalog Item in VMware Aria Automation for RAG Workloads for Private AI Ready Infrastructure for VMware Cloud Foundation

In VMware Private AI Foundation with NVIDIA, as a cloud administrator, add a catalog item for provisioning databases in VMware Data Services Manager to Automation Service Broker in VMware Aria Automation.

To create the catalog item, you run a Python script. The script creates the custom resources in VMware Aria Automation that are required for using VMware Data Services Manager for database provisioning.

### Prerequisites

- Verify that you have the VMware Aria Automation organization ID.  
You can see the ID from **Identity and Access Management** UI in the top right corner of the VMware Aria Automation console or by using the VMware Aria Automation API.
- Provide a machine that has the following software installed and has network access to the VMware Data Services Manager and VMware Aria Automation instances.
  - Python 3.10
  - requests module for Python
  - PyYAML
  - urllib3

### Procedure

- 1 On the machine running Python, download the `AriaAutomation_DataServicesManager` bundle for VMware Data Services Manager 2.0.2 from [VMware Tanzu Network](#) and extract its content.
- 2 Update the `config.json` file in the folder where you extracted the bundle with the host name of VMware Data Services Manager, VMware Aria Automation base URL, VMware Aria Automation Org ID, and user credentials for VMware Data Services Manager and VMware Aria Automation.

You can also set the name of the catalog item, Automaton Assembler project, and other parameters.

- 3 To create the catalog items in VMware Aria Automation, run the `aria.py` Python script in the following way.

```
python3 aria.py enable-blueprint-version-2
```

- 4 Verify that all the custom resources are created in VMware Aria Automation.
  - a Log in to the VMware Aria Automation console as an administrator.
  - b On the main navigation bar, click **Services**.
  - c On the My Services page, click **Assembler**.

- d In Automation Assembler, click the **Infrastructure** tab and verify that a new project is available according to the settings in `config.json`.

The default project name is `DSM_Project`.

- e Click the **Design** tab and verify that a new cloud template is available according to the settings in `config.json`.

The default template name is `DSM_DBaaS`.

- f Click **Administration > Secrets** and verify that the environmental variables `dsm_hostname`, `dsm_user_id` and `dsm_password` are created according to the settings in the `config.json` file.

- g On the **Extensibility** tab, click **Library > Actions** and verify that two new CRUD (Create, Read, Update, Destroy) actions `DSM-DB-CRUD` and `DSM-Day2-Operations` are created.

## Results

For more information on the Python script operation, see the `readme.md` file in the `AriaAutomation_DataServicesManager` bundle.

## Deploy a Vector Database by Using a Self-Service Catalog Item in VMware Aria Automation for RAG Workloads for Private AI Ready Infrastructure for VMware Cloud Foundation

In VMware Private AI Foundation with NVIDIA, as a data scientist or a DevOps engineer, you can deploy a PostgreSQL database from VMware Aria Automation on VMware Data Services Manager by using a catalog item in Automation Service Broker.

You activate the `pgvector` extension in the database . This extension introduces specialized data types, operators, and functions to support the effective storage, manipulation, and analysis of vector data directly within PostgreSQL databases.

## Prerequisites

Verify with your cloud administrator that the prerequisites for creating a PostgreSQL database are in place. See [Creating Databases](#).

## Procedure

- 1 Log in to VMware Aria Automation at `https://<aria_automation_cluster_fqdn>/csp/gateway/portal`.
- 2 On the main navigation bar, click **Services**.
- 3 On the My Services page, click **Service Broker**.
- 4 On the **Consume** tab, on the navigation bar, click **Catalog**.
- 5 Locate the catalog item for database deployment according to the information from your cloud administrator.

By default, the catalog item is called **DSM DBaaS**.

- 6 In the catalog item card, click **Request** and enter the details for the new PostgreSQL database.

For more information on the settings for the database, see [Creating Databases](#) in the *VMware Data Services Manager* documentation.

- 7 After the deployment is completed, get the connection string of the deployed database.
  - a Click **Deployments > Deployments** .
  - b Select the deployment entry for the database.
  - c On the **Topology** tab, select the cloud template for the database deployment and from the **Actions** menu for the template, select **Get Connection String**.

- 8 Activate the pgvector extension.

- a Connect to the database.

```
psql -h pgvector_db_ip_address -p 5432 -d pgvector_db_name -U pgvector_db_admin -W
```

- b Activate the pgvector extension.

```
pgvector_db_name=# CREATE EXTENSION vector;
```

## Results

Configure a RAG workload with the newly-created database.

For more information on provisioning and performing operations on databases in VMware Data Services Manager from VMware Aria Automation, see the `readme.md` file in the `AriaAutomation_DataServicesManager` bundle.

# Operational Guidance for Private Private AI Ready Infrastructure for VMware Cloud Foundation

## 6

After you complete the implementation of the *Private AI Ready Infrastructure for VMware Cloud Foundation* validated solution, you perform common operations on the environment, such as examining the operational state of the components added to the environment during the implementation and updating the certificates and account passwords for these components.

For operational guidance on the components that are deployed automatically in VMware Cloud Foundation or complement the basic VMware Cloud Foundation configuration, see the *VMware Cloud Foundation Operations and Administration Guide* in the [VMware Cloud Foundation documentation](#).

Read the following topics next:

- [Personas in Private AI Ready Infrastructure for VMware Cloud Foundation](#)
- [Operational Verification of VMware Private AI Foundation with NVIDIA](#)
- [Operational Verification of vSphere with Tanzu for Private AI Ready Infrastructure for VMware Cloud Foundation](#)
- [Certificate Management for Private Private AI Ready Infrastructure for VMware Cloud Foundation](#)
- [Operational Verification of VMware Data Services Manager for Private AI Ready Infrastructure for VMware Cloud Foundation](#)
- [Operational Verification for Private AI Ready Infrastructure for VMware Cloud Foundation by Deploying a RAG Workload](#)
- [Password Management for Private AI Ready Infrastructure for VMware Cloud Foundation](#)
- [Scale Management for Private AI Ready Infrastructure for VMware Cloud Foundation](#)
- [Shutdown and Startup of Private AI Ready Infrastructure for VMware Cloud Foundation](#)

## Personas in Private AI Ready Infrastructure for VMware Cloud Foundation

Personas describe types of system users, aligned with real people and their functions within the organization. You build a persona set based on your organization requirements for role-based access control.



The following is an example of personas defined by the *Private AI Ready Infrastructure for VMware Cloud Foundation* validated solution and their equivalent access. To delegate roles and define access based on roles and responsibilities within your organizational structure, you use these personas as a baseline for defining and building a set of personas.

**Table 6-1. Example Personas for Private AI Ready Infrastructure**

Persona	Responsibility	Solution Component	Component Role or Group
Cloud Administrator	Full administrative access to vSphere with Tanzu infrastructure - configuring and activating Supervisors and vSphere namespaces	vCenter Server, SDDC Manager.	Administrator
DevOps Engineer	Deploying vSphere Pods, VMs, and Tanzu Kubernetes Grid clusters on vSphere namespaces within a Supervisor	vSphere namespace	Can edit
Data Scientist	Deploying VMs in vSphere with Tanzu by leveraging the VM Service and Kubernetes API.	VMware Aria Automation Service Broker, Tanzu Kubernetes Cluster, vSphere Namespace	Can edit
Auditor	Read-only access for security and compliance review	vSphere namespace	Can view

## Operational Verification of VMware Private AI Foundation with NVIDIA

After you complete the implementation of the *Private AI Ready Infrastructure for VMware Cloud Foundation* validated solution and VMware Private AI Foundation with NVIDIA, you perform common operations on the environment, such as examining the operational state of the NVAIE components added to the environment during the implementation.

Verify the operational state of the NVIDIA AI Enterprise (NVAIE) Kubernetes Operators and host components by checking their state and health status.

Validate that the NVAIE components are properly functioning and ready for GPU-enabled workloads running on top of VMware Cloud Foundation.

### Prerequisites

Install the vSphere kubectl plug-in to connect to the Supervisor as a vCenter Single Sign-On user. See [Download and Install the Kubernetes CLI Tools for vSphere](#).

## Verify the Status of the ESXi Host Components for Private AI Ready Infrastructure for VMware Cloud Foundation

Verify the operational state of the ESXi host by checking its state and health status.

### Expected Outcomes

- The ESXi host has access to the NVIDIA System Management Interface (nvidia-smi).

### Procedure

- 1 Enable SSH on the ESXi host.

For instructions, see [Enable Access to the ESXi Shell](#).

- 2 Log to the ESXi server as **root** over SSH.

- 3 Run the `nvidia-smi` command.

## Verify the Status of the GPU Operator for Private AI Ready Infrastructure for VMware Cloud Foundation

Verify the operational state of the NVIDIA GPU Operator by checking its state and health status.

### Expected Outcomes

- All GPU Operator pods have a **Running** status.
- All GPU Operator pods have a **Ready** status.
- The GPU Operator License returns a **Licensed** status.

### Procedure

- 1 Log in to the Supervisor as a **vCenter Server Single Sign-On** user by running the command.

```
kubectl vsphere login --server Supervisor_cluster_IP_address --vsphere-username Supervisor_cluster_administrator --insecure-skip-tls-verify
```

- 2 Verify that the GPU Operator pods have a **Running** status by running the command.

```
kubectl get pods -n gpu-operator
```

- 3 Verify that the GPU Operator license has a **Licensed** status by running the command.

```
ctnname=`kubectl get pods -n gpu-operator | grep driver-daemonset | head -1 | cut -d " " -f1`
```

```
kubectl -n gpu-operator exec -it $ctnname -- /bin/bash -c "/usr/bin/nvidia-smi -q | grep -i lic"
```

```
Defaulted container "nvidia-driver-ctr" out of: nvidia-driver-ctr, k8s-driver-manager (init)
vGPU Software Licensed Product
```

License Status	: Licensed (Expiry: 2024-2-28 21:22:44 GMT)
Applications Clocks	
Default Applications Clocks	
Clock Policy	

## What to do next

### Troubleshooting Tips

- Ensure that the GPU Operator is properly installed. See [Install the NVIDIA GPU Operator for Private AI Ready Infrastructure for VMware Cloud Foundation](#).
- Ensure that the NVIDIA GPU driver is properly installed on the ESXi host. See [Install the Vendor GPU Driver on the ESXi Hosts for Private AI Ready Infrastructure for VMware Cloud Foundation](#).

## Verify the Status of the Network Operator for Private AI Ready Infrastructure for VMware Cloud Foundation

Verify the operational state of the NVIDIA Network Operator by checking its state and health status.

### Expected Outcomes

- All Network Operator pods have a `Running` status.
- The `hostdev-net` Network Operator custom resource has a `Ready` status.
- The `nvidia-peermem-ctr` container is loaded the `nvidia-peermem` kernel module.

## Procedure

- 1 Log in to the Supervisor as a **vCenter Server Single Sign-On** user by running the command.

```
kubectl vsphere login --server Supervisor_cluster_IP_address --vsphere-username Supervisor_cluster_administrator --insecure-skip-tls-verify
```

- 2 Verify that the Network Operator pods have a `Running` status by running the command.

```
kubectl -n nvidia-network-operator get pods
```

- 3 Verify that the Network Operator custom resource has a `Ready` status by running the command.

```
kubectl get HostDeviceNetwork
```

- 4 Verify that the `nvidia-peermem-ctr` container is loaded the `nvidia-peermem` kernel module by running the command.

```
kubectl logs -n gpu-operator ds/nvidia-driver-daemonset -c nvidia-peermem-ctr
```

## What to do next

### Troubleshooting Tips

- Ensure that the Network Operator is properly installed. See [Install the NVIDIA Network Operator for Private AI Ready Infrastructure for VMware Cloud Foundation](#)
- Ensure that the NVIDIA GPU driver is properly installed on the ESXi host. See [Install the Vendor GPU Driver on the ESXi Hosts for Private AI Ready Infrastructure for VMware Cloud Foundation](#).

## Operational Verification of vSphere with Tanzu for Private AI Ready Infrastructure for VMware Cloud Foundation

You must verify that the newly-implemented infrastructure components are operational and functioning within expected parameters.

### Verify the Status of the vSphere with Tanzu Service for Private AI Ready Infrastructure for VMware Cloud Foundation

Verify the operational state of the vSphere with Tanzu service on the VI workload domain vCenter Server by checking the state and health of the service.

Validate that the vSphere with Tanzu service is healthy by running the `vmon-cli -s wcp` command on the VI workload domain vCenter Server.

### Expected Outcome

The vSphere with Tanzu service has a `Started` run state and a `Healthy` health state.

## Procedure

- 1 Log in to the VI workload domain vCenter Server by using a Secure Shell (SSH) client with the **root** credentials.
- 2 Enter a bash prompt by running the command.

```
shell
```

- 3 Verify that the vSphere with Tanzu service has a `Started` run state and a `Healthy` health state by running the command.

```
vmon-cli -s wcp
```

## What to do next

If you encounter issues while performing this procedure, use the following troubleshooting tips:

**Troubleshooting Tips**

- Ensure that the VI workload domain vCenter Server is up and running.
- Ensure that there is network connectivity between the Active Directory domain and the VI workload domain vCenter Server.
- Ensure that there are no configuration issues with the vSphere with Tanzu service on the VI workload domain vCenter Server.

## Verify the Status of the Harbor Supervisor Service for Private AI Ready Infrastructure for VMware Cloud Foundation

Verify that the Harbor Supervisor Service configured on the Supervisor of the VI workload domain vCenter Server is operational.

Validate that the Harbor Supervisor Service has an active status on the VI workload domain vCenter Server.

**Expected Outcome**

The Harbor Supervisor Service has an *active* status.

### Procedure

- 1 Log in to the VI workload domain vCenter Server at **https://<vi\_workload\_domain\_vcenter\_server\_fqdn>/ui** by using an account with **Administrator** privileges.
- 2 From the vSphere Client Menu, select **Workload Management**.
- 3 Click **Services**.
- 4 On the Services tab, verify that the Harbor Service tile has an *Active* status.
- 5 Within the Harbor Service tile, click **Supervisors** and verify that the associated Supervisor has a *Configured* status.

### What to do next

If you encounter issues while performing this procedure, use the following troubleshooting tips:

**Troubleshooting Tips**

- Ensure that the VI workload domain vCenter Server is up and running.
- Ensure that the vSphere with Tanzu service is running on the VI workload domain vCenter Server.
- Ensure that there are no configuration issues with the vSphere with Tanzu service on the VI workload domain vCenter Server.
- Ensure that the Contour Supervisor Service is also running properly by performing the same steps above for Contour.

## Verify the Status of the vSphere Namespace for Private Private AI Ready Infrastructure for VMware Cloud Foundation

Verify the operational state of the vSphere namespace for the vSphere with Tanzu service by checking the state and health of the namespace.

Validate that the Tanzu Kubernetes cluster is operational and capable of deploying workloads by checking its status on the vSphere namespace.

### Expected Outcome

The Tanzu Kubernetes cluster has a *Running* configuration state and an *Active* Kubernetes status.

## Procedure

- 1 Log in to the VI workload domain vCenter Server at **https://<vi\_workload\_domain\_vcenter\_server\_fqdn>/ui** by using an account with **Administrator** privileges.
- 2 Select **Menu > Workload management**.
- 3 Click the **Namespaces** tab and click the Supervisor namespace.
- 4 On the *namespace* page, click the **Summary** tab and verify that the namespace has a *Running* configuration status and an *Active* Kubernetes status.
- 5 In the **Link to CLI tools** section, click **Open** and verify that you can open the **Kubernetes CLI tools** page.

## What to do next

If you encounter issues while performing this procedure, use the following troubleshooting tips:

### Troubleshooting Tips

- Ensure that the VI workload domain vCenter Server is up and running.
- Ensure that the vSphere with Tanzu service is running on the VI workload domain vCenter Server.
- Ensure that there are no configuration issues with the vSphere with Tanzu service on the VI workload domain vCenter Server.
- Ensure that the network settings of the vSphere with Tanzu workload domain is properly configured.

## Verify the Status of the vSphere with Tanzu Resources for Private Private AI Ready Infrastructure for VMware Cloud Foundation

Verify the operational state of the vSphere with Tanzu resources by checking the state and health of the vSphere namespaces, Kubernetes nodes, and configuration parameters using `kubect`.

Validate that the vSphere namespaces, Tanzu Kubernetes nodes, and cluster are healthy and that the necessary resources are allocated to the Supervisor.

**Expected Outcome**

- All vSphere namespaces have a *Running* status.
- All Kubernetes nodes have a *Ready* status.
- The Kubernetes master and the KubeDNS have a *Running* status.
- The Tanzu Kubernetes cluster has a *Running* phase status.

**Prerequisites**

Install the vSphere kubectl plug-in to be able to connect to the Supervisor as a vCenter Single Sign-On user. See [Download and Install the Kubernetes CLI Tools for vSphere](#).

**Procedure**

- 1 Log in to the Supervisor cluster as a vCenter Server Single Sign-On user by running the command in PowerShell or command prompt.

```
kubectl vsphere login --server Supervisor_cluster_IP_address --vsphere-username Supervisor_cluster_administrator --insecure-skip-tls-verify
```

- 2 Verify that the configured vSphere Namespaces have the *Active* status by running the command.

```
kubectl get namespaces
```

- 3 Verify that the configured Kubernetes nodes have a *Ready* status by running the command.

```
kubectl get nodes
```

- 4 Verify that the Kubernetes master and the KubeDNS have a *Running* status by running the command.

```
kubectl cluster-info
```

- 5 Verify that you get the expected configuration properties for the Tanzu Kubernetes cluster and that the phase has a *Running* status by running the command.

```
kubectl get tanzukubernetesclusters
```

**What to do next**

If you encounter issues while performing this procedure, use the following troubleshooting tips:

**Troubleshooting Tips**

- Ensure that the VI workload domain vCenter Server is up and running.
- Ensure that the vSphere with Tanzu service is running on the VI workload domain vCenter Server.
- Ensure that there are no configuration issues with the vSphere with Tanzu service on the VI workload domain vCenter Server.
- Ensure that the network configuration of the vSphere with Tanzu workload domain is properly configured.

## Certificate Management for Private Private AI Ready Infrastructure for VMware Cloud Foundation

Consider replacing the default self-signed certificates for components, where possible, to improve the security of SSL/TLS connections to those components.

The security of the environment depends on the validity and trust of the management component certificates. As a best practice, you replace certificates in the following cases:

- Before certificates expire.
- When a certificate is compromised.
- When the attributes related to a certificate change. For example, when the host name or the organization name change.

The certificate replacement process consists of the following phases:

- 1 Complete the process for creating a certificate signing request (CSR) for the Supervisor Kubernetes API endpoint in the vSphere Client.
- 2 Use the CSR to generate a CA-signed certificate.
- 3 Import the certificate in the vSphere Client.

For step-by-step instructions for this process, see [#unique\\_55](#).

After the certificate replacement process has been completed, you can successfully connect to the Supervisor Kubernetes API endpoint using `kubectl` without specifying the `--insecure-skip-tls-verify` option as the signed certificate is used to encrypt communications over TCP/443 and TCP/6443.

## Operational Verification of VMware Data Services Manager for Private AI Ready Infrastructure for VMware Cloud Foundation

You must verify the operational state of VMware Data Services Manager by deploying a test database and connect to it.

### Expected Outcomes

- The database managed by VMware Data Services Manager has a `Ready` status.
- You can connect to the database using `psql`.

### Procedure

- 1 Log in to the VMware Data Services Manager console at `https://<your-host-fqdn-or-ip>/login` as a **Local** user or as an **LDAP** user with the **DSM Admin** role.

- 2 Select **Databases > Create Database**.

The Create Database form appears.



- 3 In the **Basic Information** pane, configure the following information and then click **Next**.

Setting	Value
Database Engine	Postgres
Database Version	15.5+vmware.v2.0.0
Instance Name	Enter a name for the database. In this example, the database name is <code>Test</code> .
Replica Mode	Single vSphere Cluster
Topology	3 (1 Primary, 1 Replica, 1 Monitor)

- 4 In the **Infrastructure** pane, configure the following information and then click **Next**.

Setting	Value
Infrastructure Policy	Select the desired policy
Select Placement	Deselected.
Storage Policy	Select the desired policy.
VM Class	Select the desired VM class.  <b>Note</b> The VM class defines the compute and memory resources that are assigned to the database nodes. This is different from the VM class used with Tanzu Kubernetes Grid.
Disk Size	60

- 5 In the **Backup and Maintenance** pane, configure backup details and then click **Next**.

Setting	Value
Enable Backups	Selected.
Backup Location	Select the location for the backups to be stored.
Backup Retention Period	Leave the default value of 30 days.

- 6 In the **Advanced Settings** pane, do not enable the `Database Options`.
- 7 Review your configuration in the **Summary** pane.
- 8 Click **Create Database**.
- 9 To monitor the creation process, click **Databases** and then click the information icon in the Status column.

Wait for the status to change from `InProgress` to `Ready`.

- 10 Click the database instance name and then click **Copy Connection String** to copy the string to the clipboard.

The connection string can be used with psql or other tools to test connectivity. For example, in psql, the connection string has the following format:

```
postgres://<username>:<password>@<host-name>:<port>/<database-name>
```

- 11 Test the connection to the database by running the following command in psql.

```
psql -h <host-name> -p <port> -d <database-name> -p <password> -U <username> -W
```

## Operational Verification for Private AI Ready Infrastructure for VMware Cloud Foundation by Deploying a RAG Workload

This document details the setup of a Jupyter Notebook that implements the ingestion process of text documents so you build a specialized knowledge base, used by an LLM to answer questions about that knowledge domain. This process is called Retrieval Augmented Generation (RAG).

The knowledge domain used for the LLM in this example use case is based on NASA history books.

### Prerequisites

Verify that you have the following:

- Access to Docker Hub registry to pull PostgreSQL pgvector database containers.
- Access to NVIDIA NGC registry to pull PyTorch containers.
- A vSphere cluster that has the following available resources:
  - 16 vCPUs
  - 128 GB RAM
  - GPU with 40GB+ of memory (vGPU Time Slice or MIG)
  - 100 GB of disk space
- A deep learning VM running a PyTorch container.

For information on deploying deep learning VMs, see [Adding VMware Private AI Foundation with NVIDIA to Private AI Ready Infrastructure for VMware Cloud Foundation](#).

- (Optionally) A PostgreSQL database with the pgvector extension deployed by VMware Data Services Manager.

See [Deploy AI Workloads by Using VMware Aria Automation for Private AI Ready Infrastructure for VMware Cloud Foundation](#).

## Procedure

### 1 Using a pgvector instance already provisioned by DSM.

This step only applies if you already deployed a DSM Postgres Database by following the section Deploy a Vector Database by Using a Self-Service Catalog Item in VMware Aria Automation for RAG Workloads for Private AI Ready Infrastructure for VMware Cloud Foundation. If you are not using DSM to provision pgvector, then proceed to STEP 2.

- a Take note of the database connection string provided by DSM, which should look like the following example. Note that the values for your environment might be different, so make sure you get the right values.

```
# DB connection string example
postgres://pgadmin:cXx27Eb2gy3gGI1pHS54AMwuS9d7R@10.203.80.135:5432/paiftest
```

- b Split the connection string into the following values which will be needed to modify default values from the RAG Jupyter notebook.

```
# Example values extracted from the previous
# connection string.
DB_USER = "pgadmin"
DB_PASSWD = "cXx27Eb2gy3gGI1pHS54AMwuS9d7R" DEFAULT_DB = "paiftest"
DB_NAME = "paiftest"
DB_HOST = "10.203.80.135"
```

### 2 Deploy pgvector inside the deep learning VM.

STEP 2 only applies when NOT using a pgvector instance deployed by DSM.

- a Log in to the deep learning VM as the VMware user over SSH.
- b Create the pgvector home directory.
- c In the pgvector home directory, create an empty Docker file for the pgvector.

```
# Create and get into the pgvector directory.
mkdir pgvector; cd pgvector

# Create an empty Docker compose file.
touch docker-compose.yaml
```

- d In the `docker-compose.yaml` file, paste the following manifest to specify the configuration of the PostgreSQL database with the pgvector RAG workload requirements.

```
services:
  db:
    image: pgvector/pgvector:pg12
    ports:
      - '0.0.0.0:5432:5432'
    restart: always
    environment:
      - POSTGRES_DB=postgres
      - POSTGRES_USER=demouser
      - POSTGRES_PASSWORD=demopasswd
    volumes:
      - ./data:/var/lib/postgresql/data
```

- e To establish communication between the PostgreSQL and PyTorch containers, launch the PostgreSQL container and create a user-defined Docker network.

```
# Launch the PostgreSQL container
sudo docker compose up -d

# Jot down the IDs of the PostgreSQL & PyTorch containers
sudo docker ps

# Create the user-defined network "my-network"
sudo docker network create my-network

# Add each container to the network by providing its ID
docker network connect my-network <container ID>

# Confirm both are members of the user-defined network:
sudo docker network inspect my-network | grep Name

# Here is an example of the output you could get:
#
#   "Name": "my-network".      # Network name
#   "Name": "pgvector-db-1", # PostgreSQL
#   "Name": "eager_johnson", # PyTorch's random name
```

### 3 Start a JupyterLab session.

- a On your local machine, open an SSH tunnel to a PyTorch-type deep learning VM.

For example, if your VM has an IP address 10.10.10.10 and user name `vmware`, you run a command like the following example:

```
# Create a SSH tunnel to access a remote Jupyter Lab session
# using the "vmware" name.
ssh -L 8888:localhost:8888 vmware@10.10.10.10
```

- b When prompted, enter the user password.

```
The authenticity of host ' (10.10.10.10)' can't be established.
ED25519 key fingerprint is SHA256:XXXXXXXXXX
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.10.10.10' (ED25519) to the list of known hosts.
vmware@10.10.10.10's password:
=====
Welcome to the VMware Deep Learning VM
=====

Resources:
* VMware support: https://xxxxx

To reinstall Nvidia driver (if needed) run:
sudo /opt/dlvm/get-vgpu-driver.sh

System information as of Sun Apr 14 03:50:22 PM UTC 2024...
```

- c In a web browser, enter `http://localhost:8888` to start a JupyterLab session.

### 4 Download the NASA history books.

- a In JupyterLab, navigate to **File > New > Terminal** and open a terminal tab.

- b Clone the following GitHub repository.

```
# Clone the VMware GenAI Reference architecture repository
git clone \
https://github.com/vmware-private-ai/VMware-generative-ai-reference-architecture.git
```

- c Navigate to the folder where the NASA history books knowledge base is stored and run the document download script.

```
# Move to the NASA history documents folder
cd /workspace/VMware-generative-ai-reference-architecture/\
Starter-Packs/Improved_RAG/02-KB-Documents/

# Download the NASA documents
./get_NASA_books.sh
```

## 5 Install the required Python packages.

- a In the JupyterLab terminal, create a `requirements.txt` file and paste the following list of Python packages.

```
vllm=="0.4.0.post1"
transformers==4.39.3
llama-index==0.10.27
llama-index-agent-openai==0.2.2
llama-index-cli==0.1.11
llama-index-core==0.10.27
llama-index-embeddings-huggingface==0.2.0
llama-index-indices-managed-llama-cloud==0.1.5
llama-index-legacy==0.9.48
llama-index-llms-openai==0.1.14
llama-index-llms-openai-like==0.1.3
llama-index-multi-modal-llms-openai==0.1.4
llama-index-program-openai==0.1.5
llama-index-question-gen-openai==0.1.3
llama-index-readers-file==0.1.15
llama-index-readers-llama-parse==0.1.4
llama-index-storage-docstore-postgres==0.1.3
llama-index-storage-index-store-postgres==0.1.3
llama-index-storage-kvstore-postgres==0.1.2
llama-index-vector-stores-postgres==0.1.5
llama-parse==0.4.0
llamaindex-py-client==0.1.16
psycpg2-binary==2.9.9
```

- b Install the Python packages.

```
# Install the packages listed in the requirements file
pip install -r requirements.txt
```

- 6 To serve the TheBloke/zephyr-7B-alpha-AWQ vLLM used in the RAG workload, run the following command.

```
# Serve the TheBloke/zephyr-7B-alpha-AWQ LLM with vLLM
python -m vllm.entrypoints.openai.api_server --model TheBloke/zephyr-7B-alpha-AWQ --port 8010 --enforce-eager
```

---

**Attention** You must keep the terminal tab open for vLLM to keep running.

---

- 7 Create a RAG Question/Answer system from the NASA history books.

- a In JupyterLab, click the folder icon in the top left of the screen.
- b Navigate to the `VMware-generative-ai-reference-architecture/Starter-Packs/Improved_RAG/03-Document_ingestion` folder and then double-click the `Document_ingestion_pipeline.ipynb` script.
- c Modify the original document ingestion script to run on a single A100 40 GB GPU.  
The original script requires over 60 GB of VRAM.

- d Reduce the VRAM footprint and set the PostgreSQL database name.

Inside the script, scroll down to the cell below `Global Config Setup` and replace the following values.

**Table 6-2.**

Parameter	Previous Value	New Value	Description
LLM_MODEL	"HuggingFaceH4/zephyr-7b-alpha"	"TheBloke/zephyr-7B-alpha-AWQ".	The AWQ model type is quantized, which requires a smaller VRAM footprint than the original model.
EMB_MODEL	"BAAI/bge-base-en-v1.5"	"BAAI/bge-small-en-v1.5".	Requires less memory and processing cycles to produce embeddings. However, it provides less accurate contexts for the LLM.
DEVICE	"cuda:0"	"cpu"	Uses CPU RAM when calculating embeddings.
NUM_WORKERS	4	2	Reduces the number of concurrent processes allocating VRAM.
DB_HOST	"localhost"	"pgvector-db-1"	ONLY applies when using a local pgvector instance created in STEP 2. Enables Python to open connections to the pgvector store.

- e ONLY If you are using pgvector deployed by DSM, you need to replace the following default values from the Jupyter notebook and apply the values extracted from the connection string from STEP 1.

Variable	Value
DB_PASSWD	<Check your string>
DEFAULT_DB	"paiftest"
DB_NAME	"paiftest"
DB_HOST	<Check you IP address or hostname>
DB_USER	"pgadmin"

- f In the Jupyter Notebook toolbar, click the double-arrow to run all cells from the script.

- g When prompted, confirm the kernel restart.

The notebook starts to execute cell by cell. This process might take a few minutes.

- h Scroll down to the final cell, which executes a query to the LLM.

This query is augmented by the context retrieved from the pgvector store. In this example, the LLM is asked to respond to the question, "What are the main Hubble telescope discoveries about exoplanets?". Verify that the response is similar to the following text:

```
The Hubble Space Telescope has revealed exceedingly valuable information about hundreds of other worlds. Using Hubble, astronomers have probed an exoplanet's atmosphere for the first time more than 20 years ago, and have even identified atmospheres that contain sodium, oxygen, carbon, hydrogen, carbon dioxide, methane, and water vapor. While most of the planets Hubble has studied to date are too hot to host life as we know it, the telescope's observations demonstrate that the basic organic components for life can be detected and measured on planets orbiting other stars, setting the stage for more detailed studies with future observatories...
```

## Results

You successfully deployed and tested the core components of a RAG workload for Private AI Ready Infrastructure for VMware Cloud Foundation.

## What to do next

- To learn more about the core elements of a RAG workload, explore the contents and output of each cell in the script.
- To learn more about different RAG approaches and their evaluation, you can deploy a deep learning VM with at least two A100 40 GB GPUs and follow the instructions from the README files inside the rest of the folders within the Improved RAG Starter Pack repository.

# Password Management for Private AI Ready Infrastructure for VMware Cloud Foundation

Manage the account passwords of the components in your VMware Cloud Foundation environment according to the design objectives and design guidance for the *Private AI Ready Infrastructure for VMware Cloud Foundation* solution.

For step-by-step instructions on password rotation for ESXi, vCenter Server, NSX, and SDDC Manager, see [Manage Passwords](#) in the VMware Cloud Foundation product documentation.

For information on authentication and password management for vSphere with Tanzu and Tanzu Kubernetes Grid Service, see [vSphere with Tanzu Authentication](#) and [Connecting to vSphere with Tanzu Clusters](#).



# Scale Management for Private AI Ready Infrastructure for VMware Cloud Foundation

To operate your *Private AI Ready Infrastructure for VMware Cloud Foundation* validated solution in an enterprise environment, you must be able to scale up and scale out efficiently.

## Scaling up Within a VI Workload Domain

Scaling up can be an effective method for providing additional resources for your workloads. The most straightforward of mechanisms available to scale up within a Tanzu Enabled VI workload domain is resizing the Supervisor Cluster control plane nodes. When you resize the control plane nodes, you expand your ability to manage additional vSphere Pods and Tanzu Kubernetes clusters within a single vSphere Cluster. You must comply with increased CPU and memory requirements for the three Supervisor Cluster control plane nodes.

**Table 6-3. Supervisor Cluster control plane node sizing**

Control plane node size	vCPUs per node	Memory per node	Maximum vSphere Pods
Tiny	2	8 GB	1000
Small	4	16 GB	2000
Medium	8	24 GB	4000
Large	16	32 GB	8000

The compute resource requirements for three Supervisor Cluster control plane nodes at the recommended small sizing starting point is 12 vCPUs and 48GB of memory. Within that compute footprint, you can run up to 2000 vSphere Pods. If you want to scale that up to 8000 vSphere pods within that Supervisor Cluster, the aggregate compute resource requirements for the control plane increases to 48 vCPUs and 96GB of memory. At the resource levels required to effectively run 8000 vSphere Pods and the user workloads contained therein, this is a good option to keep the Supervisor Cluster operating effectively.

You can also scale up individual worker nodes in a Supervisor Cluster. Those worker nodes are the ESXi hosts that make up the vSphere Cluster. You can scale a cluster up without incurring additional licensing cost by increasing the number of physical cores, up to 32 cores per socket before additional licensing is required, and the amount of physical memory in each ESXi host. Since this design recommends for at least N+1 sizing within the Supervisor Cluster, you can perform rolling hardware upgrades until each ESXi host is upgraded.

Scaling up NSX resources for the software-defined network, load balancers, NAT, and so on, is less effective than doing so with compute resources. To support activation of a Supervisor Cluster, the associated NSX Edge cluster nodes must already be sized large at a minimum. You do not have much room to expand into those deployed NSX Edge cluster nodes. Scaling out is much more effective here.

Scaling up GPU resources on an existing vSphere Cluster depends on the amount of GPUs that the physical server can have, power, and cool. A homogenous GPU configuration within the same cluster streamlines resource allocation, workload distribution, and troubleshooting processes.

## Scaling Out Within a VI Workload Domain

There are multiple ways in which resources can be effectively scaled out within a VI workload domain. The primary approach is to scale out a Supervisor Cluster by adding ESXi hosts. This approach is effective only to a certain point, at which time adding vSphere Clusters to the workload domain and activating them as Supervisor Clusters is the preferred method. The point at which this method becomes necessary depends on many factors, including but not exclusively:

- Availability – placing many eggs into a single basket, some applications might require scale outside of a single Kubernetes cluster or a fault domain.
- Manageability – time to remediate when applying updates or upgrades, complexity of many namespaces or tenants within a domain.
- Performance – noisy neighbors, overloading individual components (network, CPU, memory), scalability limits within the domain.
- Recoverability – loss of the entire fault domain in larger clusters drives up recovery time, recovery point objectives are harder to meet with more backup activity within a domain.
- Security – separation of duties, RBAC.

When you decide to add another vSphere Cluster for use as a Supervisor Cluster, you have to decide how to scale your NSX Edge cluster resources, as well. Additional Supervisor Clusters within the VI workload domain must have their own set of software-defined network (SDN) components, so you must instantiate a new NSX Edge cluster. At this point, you have two choices:

- Deploy an additional NSX Edge cluster within the workload domain with an additional Tier-0 Gateway, or
- Deploy an additional NSX Edge cluster without the Tier-0 Gateway and attach a Tier-1 Gateway running on the NSX Edge cluster to the Tier-0 Gateway running on the original NSX Edge cluster.

This also applies in the case where software-define networking resources are scaled out by adding another NSX Edge cluster without adding more compute resources with an additional Supervisor Cluster. The Tier-0 Gateway in NSX provides dynamic routing from the SDN to the top-of-rack switches. In this case, it might be additional management overhead without much benefit. The opposite side of that manageability argument is consistency across NSX Edge cluster deployed in the environment. Aberrant configurations can make troubleshooting harder, especially if the SDN environment is not well-documented.

# Shutdown and Startup of Private AI Ready Infrastructure for VMware Cloud Foundation

In certain cases, for example, during hardware or power maintenance of the data center, you must shut down the virtual machines deployed by vSphere with Tanzu in a VMware Cloud Foundation environment in a way that prevents data loss and appliance malfunction, and start it up restoring component integration after the maintenance operation is over.

## Shut Down the Virtual Machines of vSphere with Tanzu for Private AI Ready Infrastructure for VMware Cloud Foundation

Shut down the Supervisor control plane virtual machines, Tanzu Kubernetes Cluster control plane and worker virtual machines, and the Harbor virtual machine in the vSphere with Tanzu workload domain.

For the full-stack shutdown order of VMware Cloud Foundation and of a VI workload domain with vSphere with Tanzu, see [Shutting Down VMware Cloud Foundation](#).

### Procedure

- 1 Shut down the Supervisor control plane virtual machines.
  - a Log in to the ESXi host that runs the first Supervisor control plane virtual machine at `https://<esxi_host_fqdn>` as **root**.
  - b In the navigation pane, click **Virtual Machines**.
  - c Right-click the supervisor control plane virtual machine, and select **Guest OS > Shut down**.
  - d In the confirmation dialog box, click **Yes**.
  - e Repeat the steps to shut down the remaining Supervisor control plane virtual machines on the workload domain ESXi hosts which run them.
- 2 To shut down the Tanzu Kubernetes Cluster control plane and worker virtual machines, repeat the previous step on the ESXi hosts that are running these machines.  
Start with the first Tanzu Kubernetes Cluster control plane virtual machine.
- 3 Shut down the Harbor virtual machines.
  - a Log in to the ESXi host that runs the first Tanzu Kubernetes Cluster control plane virtual machine at `https://<esxi_host_fqdn>` as **root**.
  - b In the navigation pane, click **Virtual machines**.
  - c Right-click the Harbor virtual machine and select **Power > Power off**.
  - d In the **Warning** dialog box, click **Yes**.
  - e Repeat the procedure to power off the remaining Harbor virtual machines on the ESXi hosts that run them.

## Start the vSphere with Tanzu Virtual Machines for Private AI Ready Infrastructure for VMware Cloud Foundation

The virtual machines that are deployed by vSphere with Tanzu for running containerized workloads over Kubernetes are automatically started by vCenter Server after vCenter Server and NSX for the VI workload domain are back online.

For the full-stack startup order of VMware Cloud Foundation and of a VI workload domain, see [Starting Up VMware Cloud Foundation](#).

# Solution Interoperability for Private AI Ready Infrastructure for VMware Cloud Foundation

# 7

Integrate the *Private AI Ready Infrastructure for VMware Cloud Foundation* validated solution with components added to your VMware Cloud Foundation environment by other validated solutions for operations management and business continuity. You can use such validated solutions for monitoring and alerting, logging, backup and restore, disaster recovery, and life cycle management with certain considerations.

Performing the deployments and configurations that are part of validated solutions for operations management and business continuity, are out of scope of the *Private AI Ready Infrastructure for VMware Cloud Foundation* validated solution. However, the solution provides either design or implementation guidance to facilitate the integration with such solutions.

Read the following topics next:

- [Monitoring and Alerting for Private AI Ready Infrastructure for VMware Cloud Foundation](#)
- [Logging for Private AI Ready Infrastructure for VMware Cloud Foundation](#)
- [Data Protection for Private AI Ready Infrastructure for VMware Cloud Foundation](#)
- [Disaster Recovery for Private AI Ready Infrastructure for VMware Cloud Foundation](#)
- [Life Cycle Management for Private AI Ready Infrastructure for VMware Cloud Foundation](#)

## Monitoring and Alerting for Private AI Ready Infrastructure for VMware Cloud Foundation

After you implement the *Private AI Ready Infrastructure for VMware Cloud Foundation* validated solution, monitor the parameters of new or reconfigured components in the VMware Cloud Foundation system.

For validated monitoring solutions, see the [VMware Cloud Foundation Validated Solutions](#).

If your environment is running VMware Aria Operations 8.16.1 or later, its native integration with VMware Cloud Foundation allows you to monitor your Supervisor Clusters, any Tanzu Kubernetes clusters, and Deep Learning VMs you have deployed. See [Properties for vCenter Server Components in VMware Aria Operations](#).

In the vSphere Client, you can monitor GPU metrics in the following way:

- At the host level. See [Hosts Performance Charts in vSphere](#).

- At the cluster level in custom charts. See [Working with Advanced and Custom Charts in vSphere](#).

## Logging for Private AI Ready Infrastructure for VMware Cloud Foundation

After you implement the *Private AI Ready Infrastructure for VMware Cloud Foundation* validated solution, by using VMware or third-party components, collect log data in a central place from the components that are newly-added to or re-configured in your VMware Cloud Foundation system.

For validated logging solutions, see the [VMware Cloud Foundation Validated Solutions](#) main page.

If your environment is running VMware Aria Operations for Logs, its native integration with *VMware Cloud Foundation* allows for log collection and observability for your Supervisors, Tanzu Kubernetes Grid clusters, and deep learning VMs.

## Data Protection for Private AI Ready Infrastructure for VMware Cloud Foundation

After you implement the *Private AI Ready Infrastructure for VMware Cloud Foundation* validated solution, back up of the components ensures that you can keep your environment operational if a data loss or failure occurs.

### Velero as a Backup Solution

You can implement a backup solution that supports backup and recovery of Kubernetes native objects, such as Velero, a free and open source Kubernetes backup and recovery solution.

You must install the Velero plug-in for vSphere and the Velero Data Manager. The plug-in supports backup and recovery of Kubernetes objects within the Supervisor and of Tanzu Kubernetes Grid clusters that run in a Supervisor. You must also ensure that the backup target has sufficient disk space to store the backups.

For more information on Velero, the Velero plug-in for vSphere, and the Velero Data Manager, see:

- [Velero](#)
- [Velero plug-in for vSphere](#)
- [Velero Data Manager](#)

You implement backups to prepare for:

- A critical failure of a component
- An upgrade of a component
- A certificate update of a component

You take the following backup types:

- Scheduled backups, which ensure that at any given point in time, you can restore from a recent backup.
- Manual backups for a one-off or a point-in-time recovery.
- Manual backups after a recovery of a failed part of the system.

To back up Kubernetes components or Tanzu Kubernetes Grid clusters deployed into a Supervisor by using Velero, you create backups using the Velero CLI. You can use the Velero CLI to create both one-off backups and to create scheduled backup jobs. Scheduled jobs must be configured to meet your recovery time objectives and recovery point objectives.

## Disaster Recovery for Private AI Ready Infrastructure for VMware Cloud Foundation

After you implement the *Private AI Ready Infrastructure for VMware Cloud Foundation* validated solution, set up planned migration and disaster recovery for the new components in your VMware Cloud Foundation system.

Because the entire Supervisor cannot be backed up using available tools, you must have a disaster recovery site with a ready Supervisor that has access to your backup repository or rebuild the existing Supervisor in the event of a disaster or outage. After the new Supervisor is in place, you can take existing Velero backups of individual Kubernetes components inside the Supervisor or backups of your Tanzu Kubernetes Grid clusters from the backup repository and recover them by using the Velero CLI.

For more information on disaster recovery by using Velero, see [Disaster recovery with Velero 1.6](#).

## Life Cycle Management for Private AI Ready Infrastructure for VMware Cloud Foundation

After you implement the *Private AI Ready Infrastructure for VMware Cloud Foundation* validated solution, activate upgrade and patching of the new components for your VMware Cloud Foundation system.

For information on the impact of performing life cycle management of the products in this validated solution on VMware Cloud Foundation and other validated solutions that might be deployed in your environment, see VMware Knowledge Base Article [KB89745](#).

To perform life cycle management of the products included within this validated solution, see the vSphere with Tanzu product documentation.

For step-by-step upgrade procedures for the Supervisor, see:

- [About vSphere with Tanzu Upgrades](#)
- [Update the Supervisor Cluster by Performing a vSphere Namespaces Update](#)

For step-by-step procedures for Tanzu Kubernetes clusters updates, see [Update Tanzu Kubernetes Clusters](#).

For the step-by-step upgrade procedure for VMware Cloud Foundation, see the [VMware Cloud Foundation Lifecycle Management](#).



# Design Decisions for Private AI Ready Infrastructure for *VMware Cloud Foundation*



The appendix aggregates all design decisions of the *Private AI Ready Infrastructure for VMware Cloud Foundation* validated solution. You can use this design decision list for reference related to the end state of the environment and potentially to track your level of adherence to the design and any justification for deviations.

For full design details, see [Chapter 2 Detailed Design of Private AI Ready Infrastructure for VMware Cloud Foundation](#) and [Chapter 3 Detailed Design for VMware Private AI Foundation with NVIDIA for Private AI Ready Infrastructure for for VMware Cloud Foundation](#).

## Compute Design

**Table 8-1. Design Decisions for Compute Configuration for Private AI Infrastructure for VMware Cloud Foundation**

Decision ID	Design Decision	Design Justification	Design Implication
AIR-COMPUTE-001	Select servers with CPUs with a high number of cores.	To optimize computational efficiency and minimize the need for scaling out by adding more nodes, consider scaling up the CPU core count in each server. By choosing CPUs with a high number of cores, you can effectively handle multiple inference threads simultaneously. This approach maximizes hardware utilization and enhances the capacity to manage parallel tasks, leading to improved performance and resource utilization in inference workloads	High-end CPUs might increase the overall cost of the solution.
AIR-COMPUTE-002	Select a fast-access memory.	Minimal latency for data retrieval is crucial for real-time inference applications. Increased latency reduces inference performance and give a poor user experience.	Re-purposing available servers might not be a feasible option and overall cost of the solution might increase.
AIR-COMPUTE-003	Select CPUs with Advanced Vector Extensions (AVX, AVX2, or AVX-512).	CPUs with support for AVX or AVX2 can improve performance in deep learning tasks by accelerating vector operations.	Re-purposing available servers might not be a feasible option and overall cost of the solution might increase.

## Network Design

**Table 8-2. Design Decisions on Networking for Private AI Ready Infrastructure for VMware Cloud Foundation**

Decision ID	Design Decision	Design Justification	Design Implication
AIR-TZU-NET-001	Set up networking for 100 Gbps or higher if possible.	100 Gbps networking provides enough bandwidth and very low latency for inference and fine-tuning use cases backed by vSAN ESA.	The cost of the solution is increased.
AIR-TZU-NET-002	Add a /28 overlay-backed NSX segment for use by the Supervisor control plane nodes.	Supports the Supervisor control plane nodes.	You must create the overlay-backed NSX segment.
AIR-TZU-NET-003	Use a dedicated /20 subnet for pod networking.	A single /20 subnet is sufficient to meet the design requirement of 2000 pods.	You must set up a private IP space behind a NAT that you can use in multiple Supervisors.
AIR-TZU-NET-004	Use a dedicated /22 subnet for services.	A single /22 subnet is sufficient to meet the design requirement of 2000 pods.	Private IP space behind a NAT that you can use in multiple Supervisors.
AIR-TZU-NET-005	Use a dedicated /24 or larger subnet on your corporate network for ingress endpoints.	A /24 subnet is sufficient to meet the design requirement of 2000 pods in most cases.	This subnet must be routable to the rest of the corporate network. A /24 subnet will be sufficient for most use cases, but you should evaluate your ingress needs before deployment.
AIR-TZU-NET-006	Use a dedicated /24 or larger subnet on your corporate network for egress endpoints.	A /24 subnet is sufficient to meet the design requirement of 2000 pods in most cases.	This subnet must be routable to the rest of the corporate network. A /24 subnet will be sufficient for most use cases, but you should evaluate your egress needs before to deployment.

## Accelerators Design

**Table 8-3. Design Decisions on Accelerators for Private AI Ready Infrastructure for VMware Cloud Foundation**

Decision ID	Design Decision	Design Justification	Design Implication
AIR-ACCELERATE-001	Select GPUs with high memory bandwidth	AI workloads require high memory bandwidth to efficiently handle large amounts of data. Look for GPUs with high memory bandwidth specifications.	<ul style="list-style-type: none"> <li>■ The cost of the solution is increased.</li> <li>■ GPU choice might be limited.</li> </ul>
AIR-ACCELERATE-002	Select GPUs with large memory capacity.	To handle efficiently LLMs, select GPUs equipped with substantial memory capacities. LLMs containing billions of parameters demand significant GPU memory resources for model fine-tuning and inference.	<ul style="list-style-type: none"> <li>■ The cost of the solution is increased.</li> <li>■ GPU choice might be limited.</li> </ul>
AIR-ACCELERATE-003	Evaluate and compare compute performance of the available options of GPUs.	Assess the GPU's compute performance based on metrics like CUDA cores (for NVIDIA GPUs) or stream processors (for AMD GPUs). Higher compute performance provide support for faster model training and inference, particularly beneficial for complex AI tasks.	<ul style="list-style-type: none"> <li>■ The cost of the solution is increased.</li> <li>■ GPU choice might be limited.</li> </ul>
AIR-ACCELERATE-004	Evaluate cooling and power efficiency of GPUs.	To manage the strain large language models place on GPUs, prioritize systems with efficient cooling and power management to mitigate high power consumption and heat generation.	You must select server platforms focused on GPU.

## Storage Design

**Table 8-4. Design Decisions on Storage for Private AI Ready Infrastructure for VMware Cloud Foundation**

Decision ID	Design Decision	Design Justification	Design Implication
AIR-STORAGE-001	Use vSAN ESA with 100 Gbps networking and, if possible, RDMA.	Provides high performance and efficiency. Although the minimum bandwidth for vSAN ESA is 25 Gbps, 100 Gbps and faster provides the best performance in terms of bandwidth and latency for all AI use cases.	<ul style="list-style-type: none"> <li>■ The cost of the solution is increased.</li> <li>■ RDMA increases the design complexity.</li> <li>■ The choice of vSAN ReadyNodes is limited to nodes that are approved for use with vSAN ESA.</li> </ul>
AIR-STORAGE-002	Use vSAN ESA RAID 5 or RAID 6 erasure coding.	Provides performance equal to RAID 1 mirroring.	None.
AIR-STORAGE-003	Leave data compression enabled for vSAN ESA.	Enables transmitting data in compressed state across hosts in the cluster. Data compression in vSAN ESA is controllable using storage policies.	None.

## Deployment Specification Design

**Table 8-5. Design Decisions on vSphere Storage Policy Based Management for Private AI Ready Infrastructure for VMware Cloud Foundation**

Decision ID	Design Decision	Design Justification	Design Implication
AIR-SPBM-CFG-001	Create a vSphere tag and tag category, and apply the vSphere tag to the vSAN datastore in the shared edge and workload vSphere cluster in the VI workload domain.	<p>Supervisor activation requires the use of vSphere Storage Based Policy Management (SPBM).</p> <p>To assign the vSAN datastore to the Supervisor, you need to create a vSphere tag and tag category to create an SPBM rule.</p>	You must perform this operation manually or by using PowerCLI.
AIR-SPBM-CFG-002	Create a vSphere Storage Policy Based Management (SPBM) policy that specifies the vSphere tag you created for the Supervisor.	When you create the SPBM policy and define the vSphere tag for the Supervisor, you can then assign that SPBM policy during Supervisor activation.	You must perform this operation manually or by using PowerCLI.

**Table 8-6. Design Decisions on the Supervisor for Private AI Ready Infrastructure for VMware Cloud Foundation**

Decision ID	Design Decision	Design Justification	Design Implication
AIR-TZU-CFG-001	Activate vSphere with Tanzu on the shared edge and workload vSphere cluster in the VI workload domain.	The Supervisor is required to run Kubernetes workloads natively and to deploy Tanzu Kubernetes Grid clusters natively using Tanzu Kubernetes Grid Service.	Ensure the shared edge and workload vSphere cluster is sized to support the Supervisor control plane, any additional integrated management workloads, and any customer workloads.
AIR-TZU-CFG-002	Deploy the Supervisor with small-size control plane nodes.	Deploying the control plane nodes as small-size appliances gives you the ability to run up to 2,000 pods within your Supervisor.  If your pod count is higher than 2,000 for the Supervisor, you must deploy control plane nodes that can handle that level of scale.	You must consider the size of the control plane nodes.
AIR-TZU-CFG-003	Use NSX as provider of the software-defined networking for the Supervisor.	You can deploy a Supervisor either by using NSX or vSphere networking .  VMware Cloud Foundation uses NSX for software-defined networking across the SDDC. Deviating for vSphere with Tanzu would increase the operational overhead.	None.
AIR-TZU-CFG-004	Deploy the NSX Edge cluster with large-size nodes.	Large-size NSX Edge nodes are the smallest size supported to activate a Supervisor.	You must account for the size of the NSX Edge nodes.
AIR-TZU-CFG-005	Deploy a single-zone Supervisor.	A three-zone Supervisor requires three separate vSphere clusters.	No change to existing design or procedures with single-zone Supervisor.

**Table 8-7. Design Decisions on the Harbor Supervisor Service for AI Ready Infrastructure for VMware Cloud Foundation**

Decision ID	Design Decision	Design Justification	Design Implication
AIR-HRB-CFG-001	Deploy Contour as an Ingress Supervisor Service.	Harbor requires Contour on the target Supervisor to provide Ingress Service. The Ingress IP address provided by Contour must be resolved to the Harbor FQDN.	None.
AIR-HRB-CFG-002	Deploy the Harbor Registry as a Supervisor Service.	Harbor as a Supervisor Service has replaced the integrated registry in previous vSphere versions.	<p>You must provide the following configuration:</p> <ul style="list-style-type: none"> <li>■ Harbor FQDN</li> <li>■ Record and Pointer Record (PTR) for the Harbor Registry IP (this IP is provided by the Contour Ingress Service)</li> <li>■ Manage Supervisor Services privilege in vCenter Server.</li> </ul>

**Table 8-8. Design Decisions on the Tanzu Kubernetes Grid Cluster for Private AI Ready Infrastructure for VMware Cloud Foundation**

Decision ID	Design Decision	Design Justification	Design Implication
AIR-TZU-CFG-006	Deploy a Tanzu Kubernetes Grid Cluster in the Supervisor.	For applications that require upstream Kubernetes compliance, a Tanzu Kubernetes Grid Cluster is required.	None.
AIR-TZU-CFG-007	For a disconnected environment, configure a local content library for Tanzu Kubernetes releases (TKRs) for use in the shared edge and workload vSphere cluster.	In a disconnected environment, the Supervisor is unable to pull TKR images from the central public content library maintained by VMware. To deploy a Tanzu Kubernetes Grid on a Supervisor, you must configure a content library in the shared edge and workload vSphere cluster with the required images, downloaded from the public library.	You must manually configure the content library.
AIR-TZU-CFG-008	Use Antrea as the container network interface (CNI) for your Tanzu Kubernetes Grid clusters.	Antrea is the default CNI for Tanzu Kubernetes Grid clusters.	New Tanzu Kubernetes Grid clusters are deployed with Antrea as the CNI, unless you specify Calico.



**Table 8-9. Design Decisions on Sizing the Tanzu Kubernetes Grid Cluster for Private AI Ready Infrastructure for VMware Cloud Foundation**

Decision ID	Design Decision	Design Justification	Design Implication
AIR-TZU-CFG-009	Deploy Tanzu Kubernetes Grid clusters with a minimum of three control plane nodes.	Deploying three control plane nodes ensures the control plane state of your Tanzu Kubernetes Grid cluster stays if a node failure occurs.  Horizontal and vertical scaling of the control plane is supported. See <a href="#">Scale a TKG Cluster on Supervisor Using Kubectrl</a> .	None.
AIR-TZU-CFG-010	For production environments, deploy Tanzu Kubernetes Grid clusters with a minimum of three worker nodes.	Deploying three worker nodes provides a higher level of availability of your workloads deployed to the cluster.	You must configure your customer workloads to use effectively the additional worker nodes in the cluster for high availability at an application-level.
AIR-TZU-CFG-011	Deploy Tanzu Kubernetes Grid clusters with small-size control plane nodes if your cluster will have less than 10 worker nodes.	You must size the control plane of a Tanzu Kubernetes Grid cluster according to the amount of worker nodes and pod density.	The size of the cluster nodes impacts the scale of a given cluster. If you must add a node to a cluster, consider the use of larger nodes. For AI GPU-enabled workloads, the GPU is the constraining factor for the amount of worker nodes that could be deployed.

## Life Cycle Management Design

**Table 8-10. Design Decisions on Life Cycle Management for Private AI Ready Infrastructure for VMware Cloud Foundation**

Decision ID	Design Decision	Design Justification	Design Implication
AIR-TZU-LCM-001	For life cycle management of a GPU-enabled VI workload domain, use a vSphere Lifecycle Manager image with a custom ESXi image that includes the GPU driver and any other core components from the GPU vendor.	<ul style="list-style-type: none"> <li>■ Eases maintaining the right host driver versions and daemons.</li> <li>■ Introduces consistency across the GPU-enabled hosts.</li> </ul>	You must create the customer vSphere Lifecycle Manager image before you deploy the VI workload domain.
AIR-TZU-LCM-002	Use the vSphere Client for life cycle management of a Supervisor.	Life cycle management of a Supervisor is not integrated in SDDC Manager.	You perform deployment, patching, updates, and upgrades of a Supervisor and its components manually.
AIR-TZU-LCM-003	Use <code>kubect1</code> for life cycle management of a Tanzu Kubernetes Grid cluster.	Life cycle management of a Tanzu Kubernetes Grid cluster is not integrated in SDDC Manager.	You perform deployment, patching, updates, and upgrades of a Tanzu Kubernetes Grid cluster and its components manually.

## Information Security and Access Design

**Table 8-11. Design Decisions on Authentication and Access Control for Private AI Ready Infrastructure for VMware Cloud Foundation**

Decision ID	Design Decision	Design Justification	Design Implication
AIR-TZU-SEC-001	Create a security group in Active Directory for DevOps administrators. Add users who need <b>edit</b> permissions within a namespace to the group and grant <b>Can Edit</b> permissions to the namespace for that group.  If you require different permissions per namespace, create additional groups.	Necessary for auditable role-based access control within the Supervisor and Tanzu Kubernetes Grid clusters.	You must define and manage security groups, group membership, and security controls in Active Directory.
AIR-TZU-SEC-002	Create a security group in Active Directory for DevOps administrators. Add users who need <b>read-only</b> permissions in a namespace to the group, and grant <b>Can View</b> permissions to the namespace for that group.  If you require different permissions per namespace, create additional groups.	Necessary for auditable role-based access control within the Supervisor and Tanzu Kubernetes Grid clusters.	You must define and manage security groups, group membership, and security controls in Active Directory.

**Table 8-12. Design Decisions on Certificate Management for Private AI Ready Infrastructure for VMware Cloud Foundation**

Decision ID	Design Decision	Design Justification	Design Implication
AIR-TZU-SEC-003	Replace the default self-signed certificate for the Supervisor management interface with a PEM-encoded, CA-signed certificate.	Ensures that the communication between administrators and the Supervisor management interface is encrypted by using a trusted certificate.	You must replace and manager certificates manually, outside certificate management automation of SDDC Manager.
AIR-TZU-SEC-004	Use a SHA-2 or higher algorithm when signing certificates.	The SHA-1 algorithm is considered less secure and has been deprecated.	Not all certificate authorities support SHA-2.

## NVIDIA Licensing System Design

**Table 8-13. Design Decisions on NVIDIA Licensing System Design for Private AI Ready Infrastructure for VMware Cloud Foundation**

Decision ID	Design Decision	Design Justification	Design Implication
AIR-NVD-LIC-001	For Delegated License Service (DLS) instances, account for extra compute, storage, and network resources as part of your management domain.	DLS is deployed as a virtual appliance with specific hardware requirements. The appliance can also be configured in a high availability setup, independent from vSphere HA.	<ul style="list-style-type: none"> <li>■ Increased resources for the management stack.</li> <li>■ You must perform life cycle management of the DLS instance.</li> </ul>
AIR-NVD-LIC-002	For Cloud License Service (CLS) instances, Internet access is required.	Internet access is required between a licensed client and a CLS instance. Ports 80 and 443 (Egress) must be allowed.	<ul style="list-style-type: none"> <li>■ Introduces potential security risks.</li> <li>■ You must enforce firewall rules, intrusion detection systems, and monitoring.</li> </ul>

## VMware Data Services Manager Design

**Table 8-14. Design Decisions on VMware Data Services Manager Design for Private AI Ready Infrastructure for VMware Cloud Foundation**

Decision ID	Design Decision	Design Justification	Design Implication
AIR-DSM-001	Deploy VMware Data Services Manager in the management domain.	A 1:1 relationship between a VMware Data Services Manager appliance and a vCenter Server instance is required. The vCenter Server instances for the VI workload domains in a VMware Cloud Foundation instance run in the management domain.	You must deploy one VMware Data Services Manager appliance per vCenter Server which impacts the required resources for the management domain and its clusters.
AIR-DSM-002	For production-grade deployments, deploy PostgreSQL databases in HA mode (3 or 5 nodes).	High Availability of Vector Databases, increasing the overall availability of the whole system that depends on the DBs.	Increased resource consumption of the target VI WLD, and increased number used IP Addresses.

**Table 8-14. Design Decisions on VMware Data Services Manager Design for Private AI Ready Infrastructure for VMware Cloud Foundation (continued)**

Decision ID	Design Decision	Design Justification	Design Implication
AIR-DSM-003	Allocate enough IP addresses for the IP pools of infrastructure policies.	You determine the number of IP addresses reserved for the IP pools according to the requirements and to the high availability topology of the database deployed by using VMware Data Services Manager. For example, a 5-node PostgreSQL cluster requires 7 IP addresses - one for each node, one for kube_VIP, and one for database load balancing).	You must consider planning and subnet sizing.
AIR-DSM-004	Define VM classes in VMware Data Services Manager that align to your resource requirements.	Consider the use case, types of workloads using the databases, amount of data, Transactions per Second (TPS), and other factors, such as target infrastructure overcommitment if applicable.  See <a href="#">Data Services Manager Documentation</a> and <a href="#">Data Modernization with VMware Data Services Manager</a> .	You must consider VMware Data Services Manager planning and design.
AIR-DSM-005	Configure LDAP as Directory Service for VMware Data Services Manager.	LDAP (TLS available if needed) can be configured as the identity provider to import users and assign roles on VMware Data Services Manager.	Increased security operation costs. You must allow port access from VMware Data Services Manager to the LDAP identity source: <ul style="list-style-type: none"> <li>■ LDAP - 389 TCP</li> <li>■ LDAPS - 636 TCP/UDP</li> </ul>
AIR-DSM-006	Configure the S3-compatible object store, for example, MinIO, with TLS.	The provider repositories for core VMware Data Services Manager storage, backup, logs and database backups must be enabled with TLS.	<ul style="list-style-type: none"> <li>■ Security and complexity is increased.</li> <li>■ You must manage TLS certificates.</li> </ul>

**Table 8-14. Design Decisions on VMware Data Services Manager Design for Private AI Ready Infrastructure for VMware Cloud Foundation (continued)**

Decision ID	Design Decision	Design Justification	Design Implication
AIR-DSM-007	Create a <a href="#">VMware Tanzu Network account</a> account and use it to configure a refresh token in VMware Data Services Manager.	Database templates and software updates are uploaded to VMware Tanzu Network.  In a connected environment, you must configure a Tanzu Network Refresh Token as part of the VMware Data Services Manager setup. In a disconnected environment, you must download the air-gapped environment repository and upload it manually to the Provider Repository.	You must perform this operation manually.
AIR-DSM-008	If you plan to run databases managed by VMware Data Services Manager on SAN ESA clusters, create a vSphere SPBM policy that is based on erasure coding.	Provides performance that is equivalent to RAID 1 but with no compromises and with better space efficiency.  The available erasure coding, RAID 5 or RAID 6, depends on the size of the all-flash vSAN ESA cluster. Erasure Coding 5 RAID 5 erasure coding requires a minimum of 4 ESXi hosts while RAID 6 erasure coding requires a minimum of 6 ESXi hosts.	<ul style="list-style-type: none"> <li>■ Design complexity, cost, and management overhead of the solution are increased.</li> <li>■ You must perform this operation manually or by using PowerCLI.</li> </ul>
AIR-DSM-009	Use RAID 5 or RAID 6 erasure coding as the default vSAN storage policy for databases.	Eliminates the trade-off of performance and deterministic space efficiency. Set FTT=1 for RAID 5 and FTT=2 for RAID 6 according to the number of hosts in the vSAN ESA cluster and your data availability requirements.	Design complexity is increased.