TECHNICAL WHITE PAPER
March 2025

VMware Private AI Foundation with NVIDIA on HGX Servers

Reference Design for Inference

Table of contents

Document history

Version	Date	Authors	Change Summary
1.0	2025-03-17	Yuankun Fu, Agustin Malanco, Ramesh Radhakrishnan	Initial release based on VCF 5.2.1

Executive summary

As AI continues to revolutionize industries, organizations—driven by concerns over cost efficiency, security, and agility—are increasingly adopting private cloud solutions to power inference workloads. VMware® Private AI Foundation with NVIDIA is a generative AI (GenAI) platform that enables AI professionals to run RAG workflows, fine-tune and customize LLM models, and run inference workloads in their on-prem data centers. The platform addresses critical issues related to privacy, choice, cost, performance, and compliance. VMware Private AI Foundation with NVIDIA comprises VMware Cloud Foundation® and NVIDIA AI Enterprise (featuring NVIDIA vGPU (C-Series), NVIDIA NIM™ and NVIDIA NeMo™ microservices, and NVIDIA AI Blueprints).

This reference design details our recommended architecture. With the products and guidelines listed here, IT teams can easily deploy a robust, future-proof infrastructure from which data scientists can easily deploy AI inference applications. The foundation of this solution is made up of NVIDIA-certified HGX servers with 8x H100 GPUs, NVSwitches and NVLinks for high-speed inter-GPU communication, and NVIDIA Spectrum™-X (Ethernet-based) networking.

These components are reliable, easy to manage, performant (high throughput and low latency), and provide exceptional AI inference capabilities.

This paper provides AI professionals with

- A list of core components and infrastructure choices
- Deployment considerations
- Performance validation of VMware Private AI Foundation with NVIDIA, offering organizations a comprehensive guide to optimizing AI inference workloads in a private cloud environment.

Introduction

The rapid evolution of artificial intelligence (AI) has driven enterprises to prioritize scalable, cost-efficient infrastructure for managing AI workloads. While cloud platforms initially provided agility for experimentation, the costs of large-scale AI inference in the cloud—along with risks of data exposure and governance—are compelling organizations to shift towards on-prem solutions. However, this transition introduces several key challenges:

1. **GPU underutilization**: On-prem GPUs are often underutilized in enterprise data centers, with scenarios such as assigning GPUs to a less frequently used model and over-provisioning for peak loads, leading to wasted resources and reduced return on investment (ROI). Optimizing GPU usage is crucial to prevent resource hoarding and enhance efficiency, similar to the optimization challenges faced in the early stages of CPU utilization.
2. **Giving data scientists a cloud-like interface**: The rapid pace of AI models and toolkit updates creates challenges for data scientists who need a flexible, cloud-like interface. At the same time, infrastructure provisioning remains largely an IT responsibility, while data scientists require the freedom to focus on model development.
3. **Model governance**: As AI models increasingly use sensitive data, effective governance is critical. Enterprises must enforce security policies, prevent model drift, and ensure compliance. Private AI frameworks provide the necessary control to secure proprietary data and maintain model reliability.
4. **Familiar management interface**: VMware's user-friendly infrastructure management interface is widely adopted in enterprise IT environments, enabling IT teams to manage AI workloads efficiently without the steep learning curve associated with new or open-source platforms. This familiarity enhances operational efficiency and reduces the potential for errors.

VMware Private AI Foundation with NVIDIA—a joint GenAI platform by Broadcom and NVIDIA—tackles these challenges by providing on-demand GPU allocation, enabling GPU sharing, and automating infrastructure management. It gives data scientists the resource flexibility they need to focus on actual development while allowing IT teams to streamline deployment, enforce governance policies, secure proprietary data, and ensure model compliance over time.

This paper presents a reference design for deploying VMware Private AI Foundation with NVIDIA to support AI inference workloads on NVIDIA-Certified HGX Systems equipped with 8x H100 GPUs, NVSwitches, and NVLinks over Ethernet networking. The proposed design offers a prescriptive solution to ensure efficient, secure, and agile AI development, deployment, and operation.

Intended audience

This reference design is for the following groups:

- **New users:** Organizations that own or plan to procure NVIDIA-Certified HGX Systems (8x H100 GPUs with NVSwitches and NVLinks) and plan to deploy VMware Private AI Foundation with NVIDIA for AI inference workloads on a VCF-based private cloud infrastructure.
- **Infrastructure architects:** VCF admins, DevOps, and SRE teams responsible for deploying and managing both physical and virtual infrastructures. This paper provides guidance on hardware, networking, and system configurations to integrate VMware Private AI Foundation with NVIDIA.
- **Software architects:** Data scientists and AI developers who will develop inference workloads on the platform. This paper offers architecture and performance insights to help optimize their AI models and workloads.

Terminology

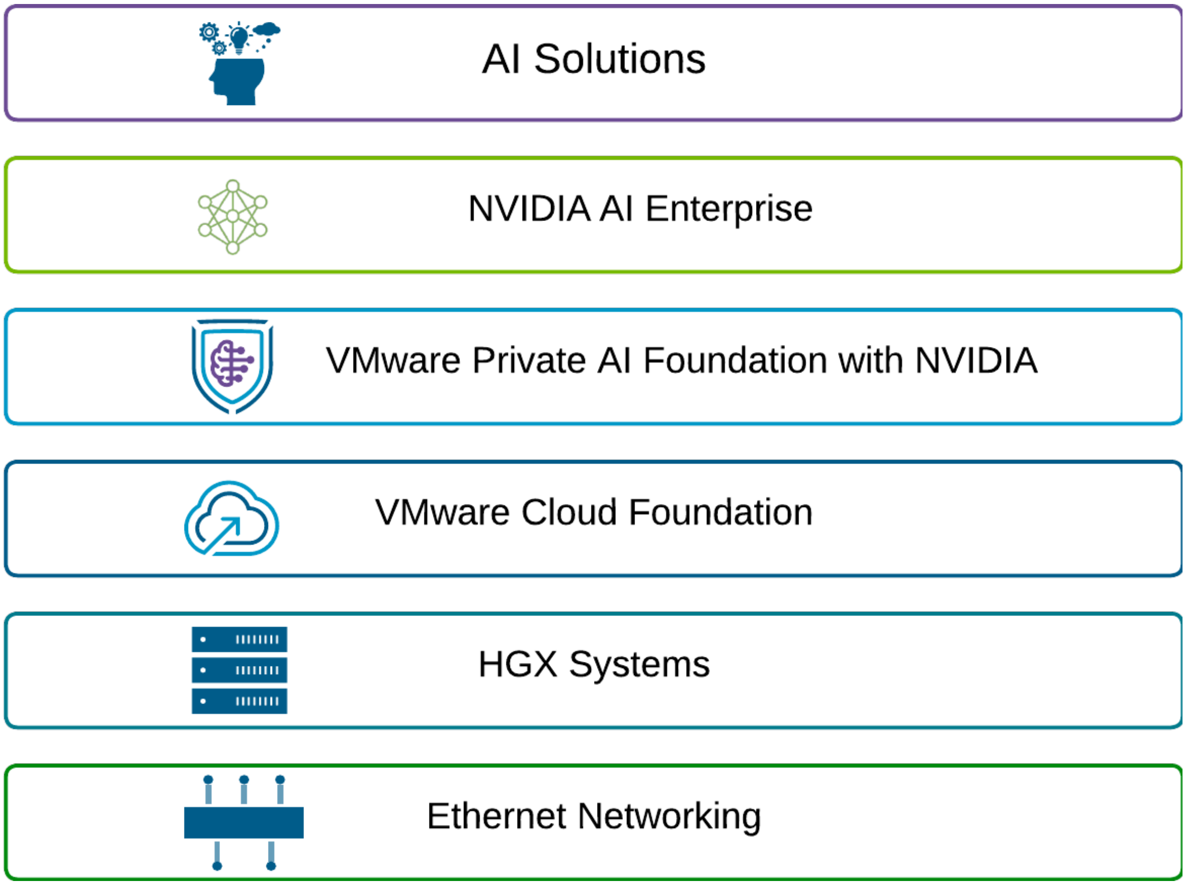
Table 1. Terms and acronyms used throughout this paper

Term	Definition
CNI	Container Network Interface
DLVM	Deep Learning Virtual Machines
DRS	Distributed Resource Scheduler
DSM	Data Service Manager
ESA	Express Storage Architecture for vSAN
HA	High Availability
LLM	Large Language Model
LCM	Life Cycle Management
NGC	NVIDIA GPU Cloud
OOB	Out-Of-Band
OSA	Original Storage Architecture for vSAN
RAG	Retrieval Augmented Generation
RDMA	Remote Direct Memory Access
SRE	Site Reliability Engineers
TEP	Tunnel End Point
VCF	VMware Cloud Foundation
VDS	Virtual Distributed Switch
vGPU	NVIDIA vGPU (C-Series)
VIB	vSphere Installation Bundles
VKS	vSphere Kubernetes Service
vLCM	vSphere Lifecycle Manager
VM	Virtual Machine
VI WLD	Virtual Infrastructure Workload Domain

Core components

Figure 1 depicts the layered architecture of the solution, where each layer represents a crucial integration point that often requires manual setup and configuration for the efficient execution of GenAI solutions. VMware Private AI Foundation with NVIDIA is an advanced service for VMware Cloud Foundation (VCF) that is available for purchase. NVIDIA AI Enterprise, which is purchased directly from NVIDIA, is also required to deliver essential software packages such as NVIDIA vGPU (C-Series), GPU Operator, NVIDIA Network Operator, and NVIDIA NIM. This reference design simplifies the deployment and optimization of each layer by providing validated, prescriptive guidance, ensuring a smoother and more efficient implementation.

Figure 1. VMware Private AI Foundation with NVIDIA on HGX systems layered architecture

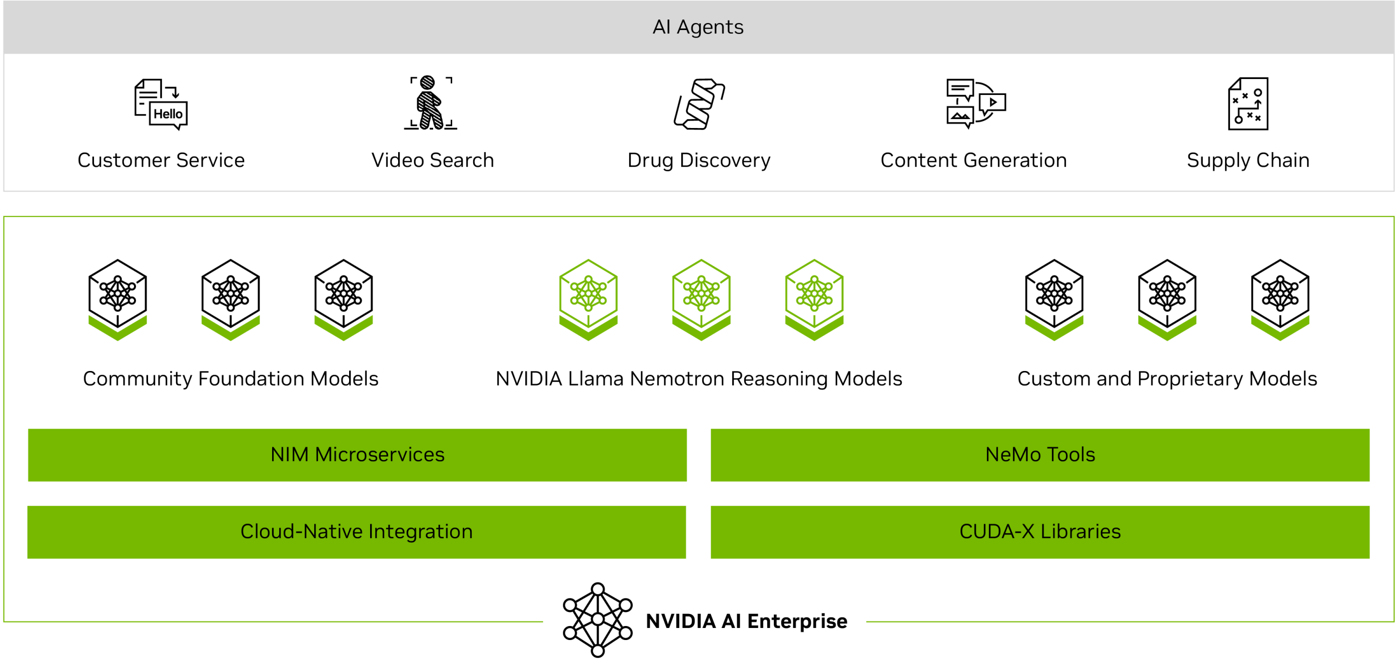


In the following sections, we provide a technical overview and key considerations for each core component as shown in Figure 1.

NVIDIA AI Enterprise

NVIDIA AI Enterprise is a cloud-native software platform that streamlines the development and deployment of production-grade, end-to-end generative AI pipelines and helps organizations build data flywheels for the next era of agentic AI. The product versions for each release are shown in the NVIDIA AI Enterprise release notes.

Figure 2. NVIDIA AI Enterprise overview



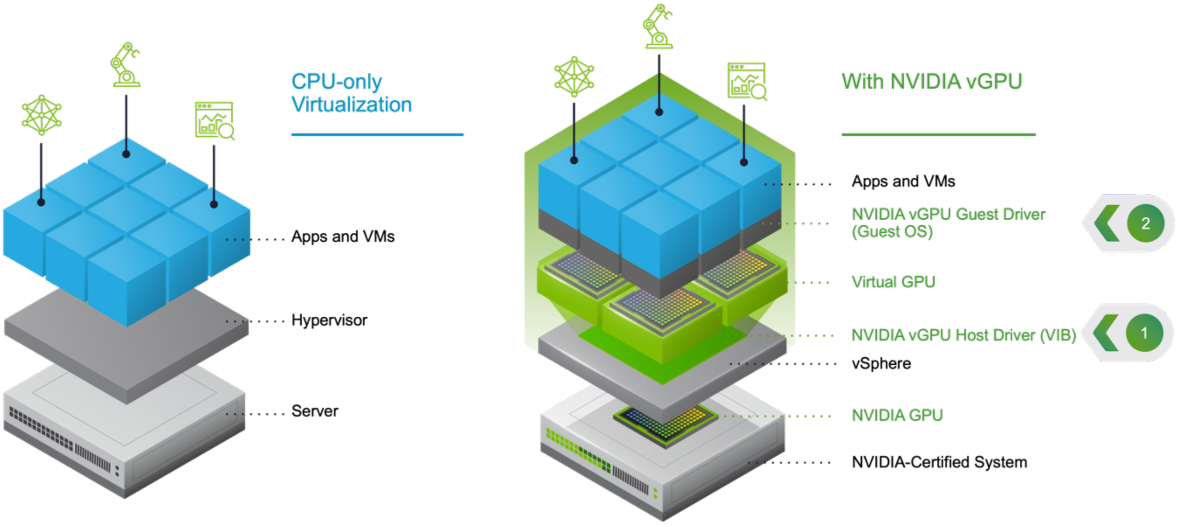
NVIDIA vGPU (C-Series)

NVIDIA virtual GPU (vGPU) technology enables efficient GPU resource pooling and sharing of one or multiple physical GPUs in a virtualized environment. It is built on a variation of passthrough, providing direct access to host physical GPUs for optimal performance while retaining the flexibility of virtualization with features like snapshots, vMotion, and more. This approach aligns with VMware software's ability to abstract physical infrastructure—compute, storage, and network—into a software layer (hypervisor) for resource pooling and consumption, allowing multiple VMs to leverage underlying hardware effectively. Additionally, all 8x GPUs (or a subset) connected via high-speed NVSwitch and NVIDIA NVLink™ in an HGX server can be allocated to a single VM with the vSphere device-group capability.

The NVIDIA vGPU (C-Series) is designed for compute-intensive workloads. As shown in Figure 3, unlike CPU-only virtualization, NVIDIA vGPU (C-Series) requires drivers to be installed at both the host and guest levels. At the host level, a vSphere installation bundle (VIB) is installed, which can be automated through vCenter Lifecycle Manager (vLCM). On the guest side, it is essential to install a vGPU guest driver version that matches the host VIB to ensure compatibility and optimal performance. Manually performing this installation can be error-prone and time-consuming. **Deploying a deep learning virtual machine (DLVM) simplifies this process by automatically installing the appropriate NVIDIA vGPU guest driver based on the host's VIB version.** This approach improves hardware utilization and simplifies the deployment and management of GPU resources in virtualized environments.

The NVIDIA vGPU host driver VIB can be downloaded from the NVIDIA NGC catalog. The `NVD-AIE-xx.zip` package also includes a mgmt-daemon VIB for monitoring GPU metrics in VMware Aria Operations. These two VIBs can be integrated into a vLCM image for use during the creation of a virtual infrastructure workload domain (VI WLD) or remediation of a cluster. The NVIDIA vGPU guest driver can also be downloaded from the NVIDIA NGC catalog. However, before obtaining these drivers, **an NVIDIA AI Enterprise license must first be acquired.** Additionally, an NVIDIA license server instance must be configured within the NVIDIA Application Hub, which generates a licensing portal API key and a client configuration token file. The key and token file are then used to enable the full capabilities of the guest vGPU driver in the AI workstation or the AI Kubernetes cluster.

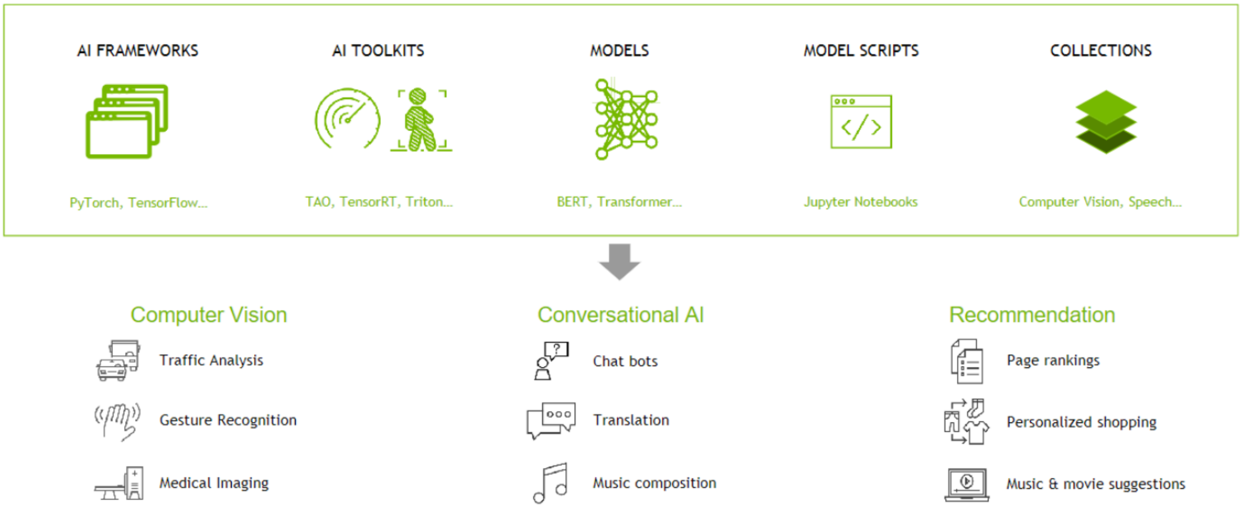
Figure 3. vGPU technology diagram



NGC

NVIDIA AI Enterprise includes NVIDIA NIM, other AI microservices, GPU Operator, NVIDIA NIM Operator, and NeMo Retriever. All of these components are available from NVIDIA NGC. Downloading these resources requires an NGC API Key.

Figure 4. NVIDIA NGC catalog overview

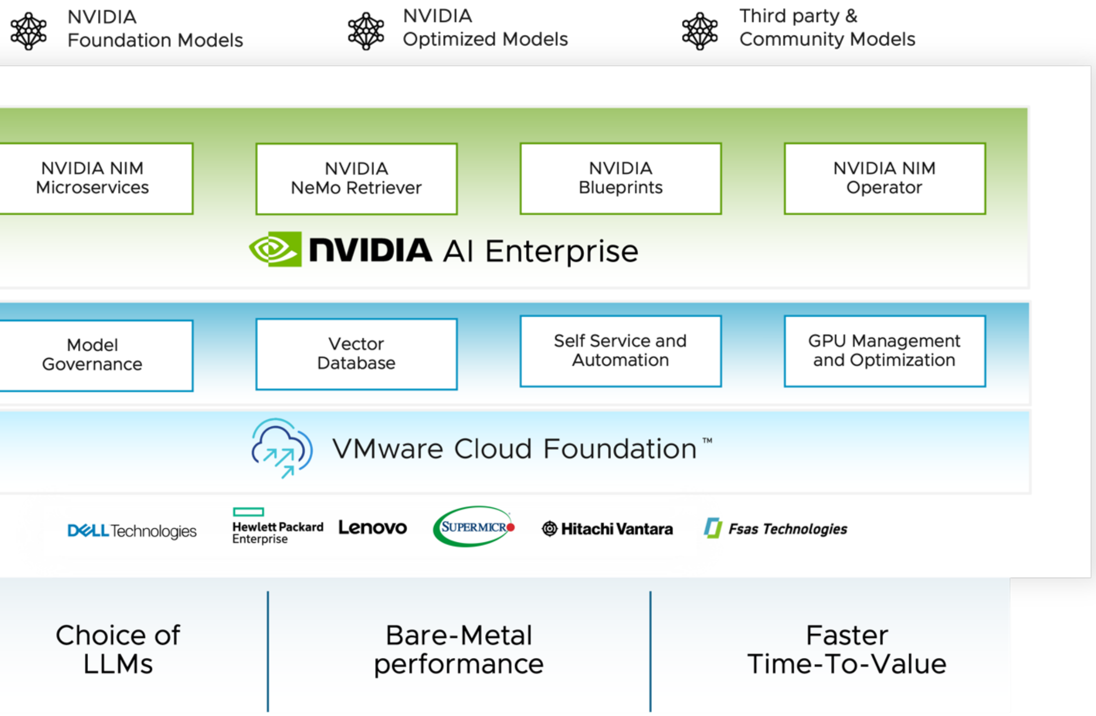


Software and containers hosted on NGC undergo scans against an aggregated set of common vulnerabilities and exposures (CVEs), crypto, and private keys. They are tested and designed to scale up to multiple GPUs and, in many cases, to multi-node. For more information, refer to the NVIDIA NGC User Guide.

VMware Private AI Foundation with NVIDIA

VMware Private AI Foundation with NVIDIA is an add-on advanced service to VCF for provisioning AI workloads based on NVIDIA AI Enterprise. It is a jointly engineered product between NVIDIA and Broadcom that enables enterprises to deploy generative AI workflows, such as retrieval augmented generation (RAG), using their proprietary data. The solution allows organizations to run inference workloads on models hosted in their private stores, addressing concerns related to privacy, choice, cost, performance, and compliance.

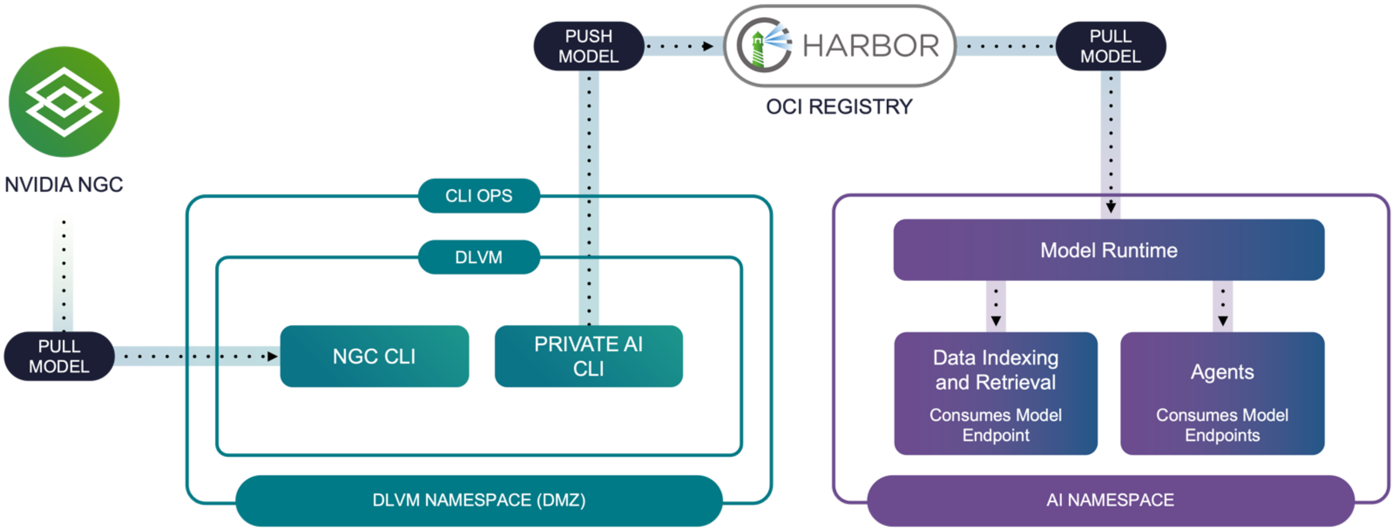
Figure 5. VMware Private AI Foundation with NVIDIA



VMware Private AI Foundation with NVIDIA offers a comprehensive solution for enterprises to develop and deploy AI applications securely and efficiently with the following key components (shown in Figure 5):

1. **Model governance** allows data scientists to test, evaluate, and store pre-trained LLMs or containers that are deemed safe and suitable for business use. The workflow, illustrated in Figure 6, begins with model testing and validation within a deep learning VM (DLVM) in an isolated environment to ensure safety and control. Once validated, these artifacts are stored in a model gallery hosted on the Harbor Registry, with evaluation processes customized to meet each enterprise's specific requirements. After passing evaluations, the models are promoted to a Kubernetes-based deployment, making them accessible to developers in a scalable environment.

Figure 6. Model Governance Journey in VMware Private AI Foundation with NVIDIA



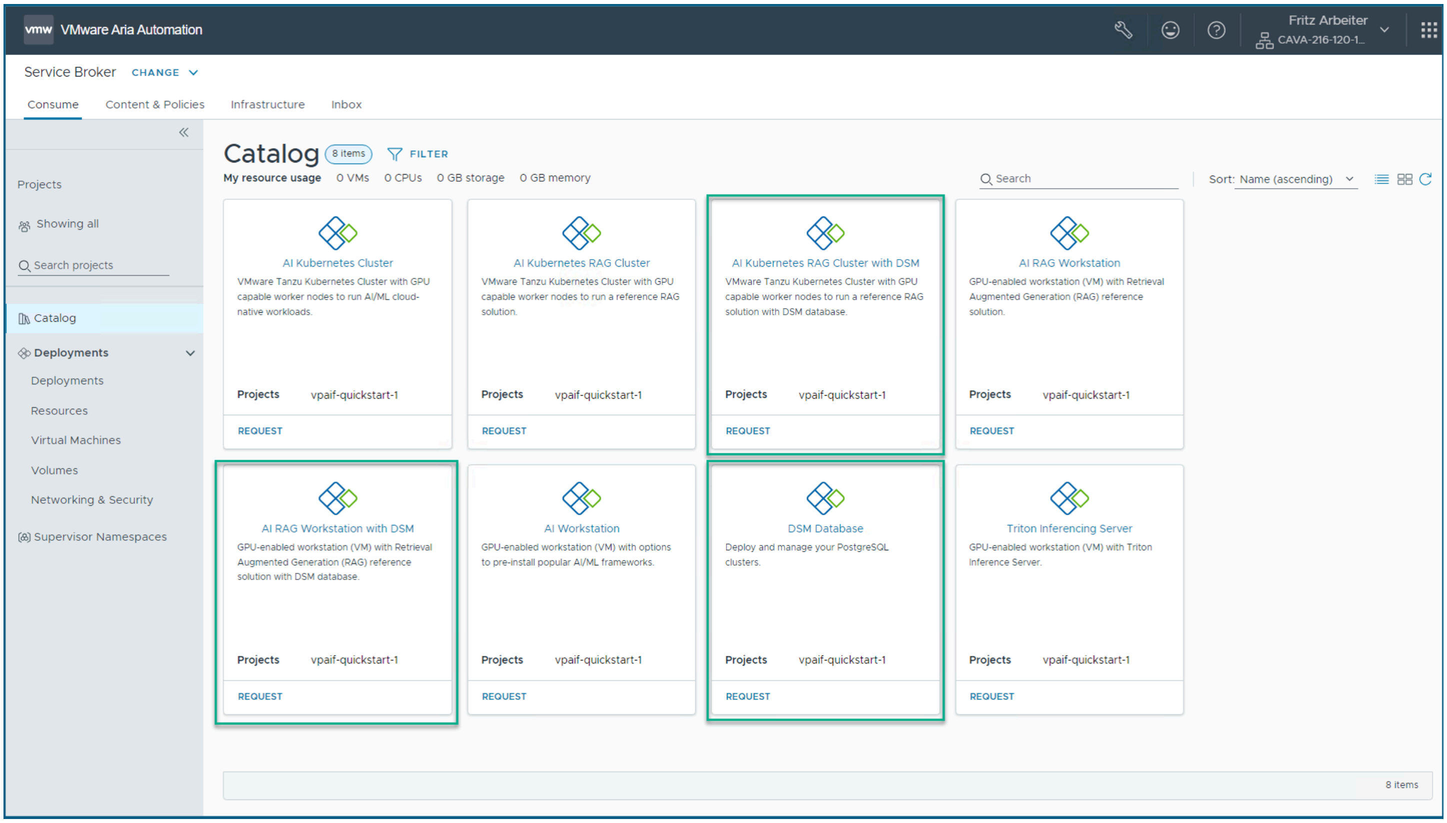
2. **Vector database** functionality is provided via PostgreSQL with the pgVector extension and deployed as a containerized service running in a Kubernetes cluster. Deployment can be automated through VMware Aria Automation Data Service Manager (DSM) integration. In addition to vector database services, DSM can automate provisioning databases of choice (for example, MySQL) for various use cases, such as storing conversation history in a real-world RAG application. This solution ensures the secure storage and retrieval of vector embeddings for private business data, which is essential for RAG applications, while enforcing RBAC and encryption. PostgreSQL's proven reliability and enterprise-grade features guarantee that private business data remains both protected and accessible for AI workflows.

3. **Self-service automation,** powered by VMware Aria Automation Service Broker, enables the provisioning of DLVMs for model development and Kubernetes clusters for production scaling. This solution benefits both the IT team and data scientists. Data scientists gain access to a cloud-like experience through a self-service catalog, equipped with the necessary software bundles. The Quickstart Wizard creates five basic catalog items that can be generally categorized into two use cases, and IT Ops can easily create or customize additional catalog items (shown in Figure 7) using Automation Assembler. With this approach, there is no need for tickets or lengthy interactions between the IT team and data scientists. Instead, there is a smooth, streamlined selection process that empowers both teams to quickly get what they need.

Table 2. Two fundamental use cases in self-service automation

Use Case	Target Users	Environment	Key Features
Development	Data scientists / Infrastructure Architects	Deep Learning VM	A GPU-enabled VM image with curated VM settings and software bundles, such as vGPU guest drivers, AI development tools, and LLM microservices (for example, NIM, NeMo Retriever, etc.)
Production	DevOps / Infrastructure Architects	Kubernetes cluster	Automate 50+ steps to simplify deployment of AI applications on vSphere Kubernetes Service once they are production-ready

Figure 7. Find catalog items related to VMware Private AI Foundation with NVIDIA in Aria Automation Service Broker



My resource usage 0 VMs 0 CPUs 0 GB storage 0 GB memory

AI Kubernetes Cluster
VMware Tanzu Kubernetes Cluster with GPU capable worker nodes to run AI/ML cloud-native workloads.
Projects vpaif-quickstart-1
REQUEST

AI Kubernetes RAG Cluster
VMware Tanzu Kubernetes Cluster with GPU capable worker nodes to run a reference RAG solution.
Projects vpaif-quickstart-1
REQUEST

AI Kubernetes RAG Cluster with DSM
VMware Tanzu Kubernetes Cluster with GPU capable worker nodes to run a reference RAG solution with DSM database.
Projects vpaif-quickstart-1
REQUEST

AI RAG Workstation
GPU-enabled workstation (VM) with Retrieval Augmented Generation (RAG) reference solution.
Projects vpaif-quickstart-1
REQUEST

AI RAG Workstation with DSM
GPU-enabled workstation (VM) with Retrieval Augmented Generation (RAG) reference solution with DSM database.
Projects vpaif-quickstart-1
REQUEST

AI Workstation
GPU-enabled workstation (VM) with options to pre-install popular AI/ML frameworks.
Projects vpaif-quickstart-1
REQUEST

DSM Database
Deploy and manage your PostgreSQL clusters.
Projects vpaif-quickstart-1
REQUEST

Triton Inferencing Server
GPU-enabled workstation (VM) with Triton Inference Server.
Projects vpaif-quickstart-1
REQUEST

4. **GPU Monitoring** in Aria Operations provides real-time visibility into hardware performance, tracking metrics like GPU utilization, memory consumption, and temperature across VMs on the system as a whole.

VMware Cloud Foundation

VMware Cloud Foundation (VCF) is a unified full-stack private cloud platform designed to streamline the deployment, automation, and management of software-defined infrastructure. The platform is optimized for modern workloads such as AI and container-based applications. It integrates core technologies—VMware vSphere® (compute), VMware® vSAN™ (storage), VMware NSX® (networking), and the VMware Aria suite (VMware Aria® Automation™ and VMware Aria Operations)—into a single, efficient, and consistent solution. By adopting a standardized, automated approach, VCF simplifies resource management, reduces technical debt, and enhances operational agility, enabling organizations to accelerate application delivery and embrace cloud-native technologies. The private cloud platform supports GPU virtualization for AI tasks and offers the flexibility of on-prem and edge deployments, providing a consistent cloud operating model for the scalable, efficient management of diverse workloads.

Table 3 lists the bill of materials (BOM) and corresponding versions in VCF 5.2.1 used in this design. Table 4 lists each component's function and key features.

Table 3. VCF 5.2.1 BOM used in this reference design

Software Component	Version
Cloud Builder VM	5.2.1
VMware SDDC Manager	5.2.1
VMware vCenter® Server Appliance	8.0 Update 3c
VMware ESXi™	8.0 Update 3b
VMware NSX	4.2.1
VMware Aria Suite Lifecycle	8.18
VMware Private AI Foundation with NVIDIA	5.2.1

Table 4. VCF components used in this reference design

Component	Function	Key Features
SDDC Manager	Centralized management for VCF	• Automates deployment/lifecycle of vSphere, vSAN, NSX • Manages workload domains • Enforces pre-validated architectures • Provisions with an API-driven infrastructure
vSphere	Enterprise virtualization platform	• Abstracts physical compute resources • Features foundation products ESXi hypervisor and vCenter appliance • Integrates vGPU host driver including Fabric Manager (to configure NVSwitch memory fabrics) into ESXi • Uses virtual distributed switch (VDS) to simplify networking • Uses VMware vSphere® Distributed Resource Scheduler™ (DRS) for workload balancing • Includes VMware vSphere® High Availability (HA) for automated failover
↳ vSAN	Software-defined storage solution	• Creates a shared storage pool from local or direct-attached storage • Supports original storage architecture (OSA) and express storage architecture (ESA) • Includes vSAN file services for enterprise-grade NFS shares

Component	Function	Key Features
↳ vSphere Kubernetes Service (VKS)	Kubernetes native infrastructure on vSphere	• Transforms vSphere into a Kubernetes platform by enabling a supervisor cluster that runs on ESX • Provides a VM service that manages VMs in a declarative manner through the Kubernetes API • Uses Harbor Registry for AI model/container governance • Uses vSphere CSI for container-native storage capabilities • Integrates NSX and Antrea CNI for container networking
NSX	Software-defined networking and security solution	• Supports multi-tenancy through virtual private clouds (VPCs) • Provides logical switching using GENEVE encapsulation • Uses a two-tier routing model (Tier-0 and Tier-1 gateways) with network services such as NAT, VPN, and more • Provides microsegmentation via distributed firewall (DFW) • Integrates load-balancing services
Workspace ONE Access	Identity and access management component within VCF	• Provides single sign-on capabilities with multi-factor authentication • Centralizes identity governance across all VCF services and workloads • Integrates with existing identity providers
Aria Lifecycle	Deploy, configure, manage, and upgrade Aria Suite products	• Deploys in the management domain • Integrates with SDDC Manager, allowing users to initiate deployments from the SDDC Manager UI and complete them in the Aria Suite Lifecycle interface
Aria Automation*	Automates infrastructure provisioning, configuration, and management	• Enables self-service provisioning and management of infrastructure as a service (IaaS) • Automates Kubernetes, network, data services, and private AI workloads
Aria Operations*	Monitor, optimize, plan, and scale applications and infrastructure	• Provides comprehensive monitoring, troubleshooting, and capacity management for VCF • Includes operations management: Optimizes the cost and performance of compute, storage, and network resources • Includes security management: Provides visibility into hardening, governance, and compliance

* Beginning with VCF 9.0, Aria Automation and Aria Operations will be rebranded as VCF Automation and VCF Operations, respectively.

↳ This symbol signifies the component is part of the larger component above it.

HGX systems

NVIDIA HGX series servers have become very popular. The HGX platform, inspired by the NVIDIA DGX series, provides a flexible and configurable GPU solution from your trusted OEM provider. Server vendors can directly buy the standard NVIDIA HGX baseboard. This baseboard features 8 SXM form-factor GPUs linked through 4 NVSwitches and NVLinks. OEM vendors can customize other components like CPU, RAM, storage, and NICs around the baseboard while keeping a consistent GPU setup. This allows them to submit their systems for HGX certification under the NVIDIA-Certified Systems program. This approach makes it easier to integrate multiple GPUs and advanced technologies, leading to more flexible, efficient, and powerful systems for AI and HPC.

Table 5 provides a filtered list of HGX systems retrieved from the Broadcom Compatibility Guide. These systems include 8x H100 SXM GPUs, and partners have certified them (in March 2025) for VMware Private AI Foundation with NVIDIA.

Table 5. 8x-H100-SXM-GPU HGX Systems in the Broadcom Compatibility Guide

Partner	System	NVIDIA GPU	CPU
Dell Technologies	PowerEdge XE9680	8x H100-80GB GPU	2x Intel Emerald Rapids
	PowerEdge XE9680	8x H100-80GB GPU	2x Intel Sapphire Rapids

Ethernet networking

Ethernet's simplicity, scalability, and cost-effectiveness make it an ideal choice for private AI inference deployments. Moreover, for future-proofing infrastructure, particularly for model customization or fine-tuning, Ethernet offers efficient scalability and flexibility.

Proper networking is critical to ensuring that VMware Private AI Foundation with NVIDIA on HGX servers does not have any bottlenecks or suffer performance degradation for AI workloads. Advancements in Ethernet technology, such as RoCE v2 and lossless fabrics, combined with its open ecosystem and scalability, make it a good fit for both AI inference and training.

For more information, refer to Broadcom Networking and NVIDIA Spectrum Ethernet.

Reference architecture

This reference architecture for VMware Private AI Foundation with NVIDIA on HGX servers acts as a blueprint that provides prescriptive architectures designed to meet the changing needs of AI workloads.

Physical architecture

The components of and required network for VMware Private AI Foundation with NVIDIA on HGX servers are described in tables 6–7. For assistance with larger-scale deployments, reach out to Broadcom's VCF Division Professional Services.

Table 6. Physical architecture components

Component	Technology
Inference servers in a Virtual Infrastructure Workload Domain (4–16)	• HGX System with 8x H100 SXM GPUs and 4x NVSwitches interconnected by Gen4 NVLink • Min 2x 25Gbps NICs for VCF infrastructure service network (for example, management, vSAN storage, etc.) • Min 2x 100GbE NICs for inference workload network • Although the UI of VMware Private AI Foundation with NVIDIA requires a minimum of 3 ESXi hosts, we recommend a 4-node (N+1) configuration for resilience, ensuring redundancy for replicas and RAID-5 erasure coding while mitigating failure risks during maintenance or unexpected issues.
Inference fabric	Min 100 GbE switch
Management and storage fabric	Min 25 GbE switch
Out-of-band management fabric	Min 1 GbE switch
Management servers in the management domain	• Min 4x vSAN ready nodes certified for vSAN OSA or ESA • Size memory and compute based on the VCF components planned to be deployed in the management domain • Use the vSAN Sizer Tool for sizing guidance

Figure 8 depicts the physical architecture for a minimum of 4-node vSAN-ready setup to create the VCF management domain, alongside up to 16 HGX servers to establish the VI WLD with the Ethernet networking switch's radix (ports per switch) equal to 32. Further scale-out is possible with a network redesign. Each HGX H100 system uses 8 connections to link the workload networks. The full architecture includes three distinct networks: an Ethernet-based workload network backed by two switches, each with at least 100 GbE; an Ethernet fabric for VCF management and storage backed by two switches, each with at least 2x 25 GbE switches; and an out-of-band Ethernet network.

> **Note:** Figure 8 represents an example of a recommended setup. For VI WLD, the minimum configuration requires 4 HGX servers and 1 workload network switch. If the workload Ethernet switch has a higher radix (more than 32) or if multiple switches are used in a multi-layer network design, additional HGX servers can be incorporated into the deployment. Any OEM HGX server that meets the minimum requirements outlined in Table 5 can be used.

Figure 8. Physical architecture with a minimum of 4 and up to 16 HGX servers based on Ethernet switch radix = 32

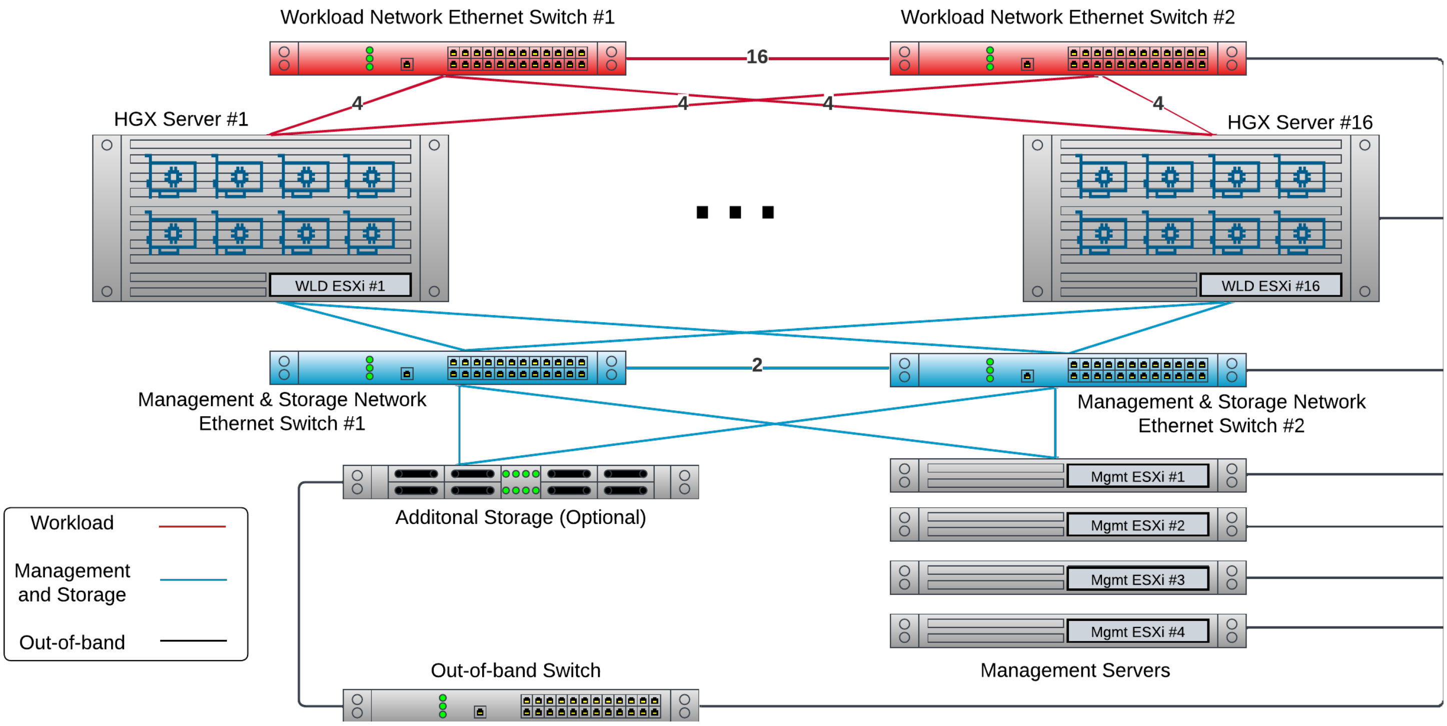

Workload Network Ethernet Switch #1
Workload Network Ethernet Switch #2
16
4
4
4
4
HGX Server #1
HGX Server #16
WLD ESXi #1
WLD ESXi #16
2
Management & Storage Network Ethernet Switch #1
Management & Storage Network Ethernet Switch #2
Additonal Storage (Optional)
Mgmt ESXi #1
Mgmt ESXi #2
Mgmt ESXi #3
Mgmt ESXi #4
Management Servers
Out-of-band Switch
Workload
Management and Storage
Out-of-band

Figure 9 presents the balanced topology diagram of a typical HGX server.

Figure 9. Topology diagram of a typical HGX server

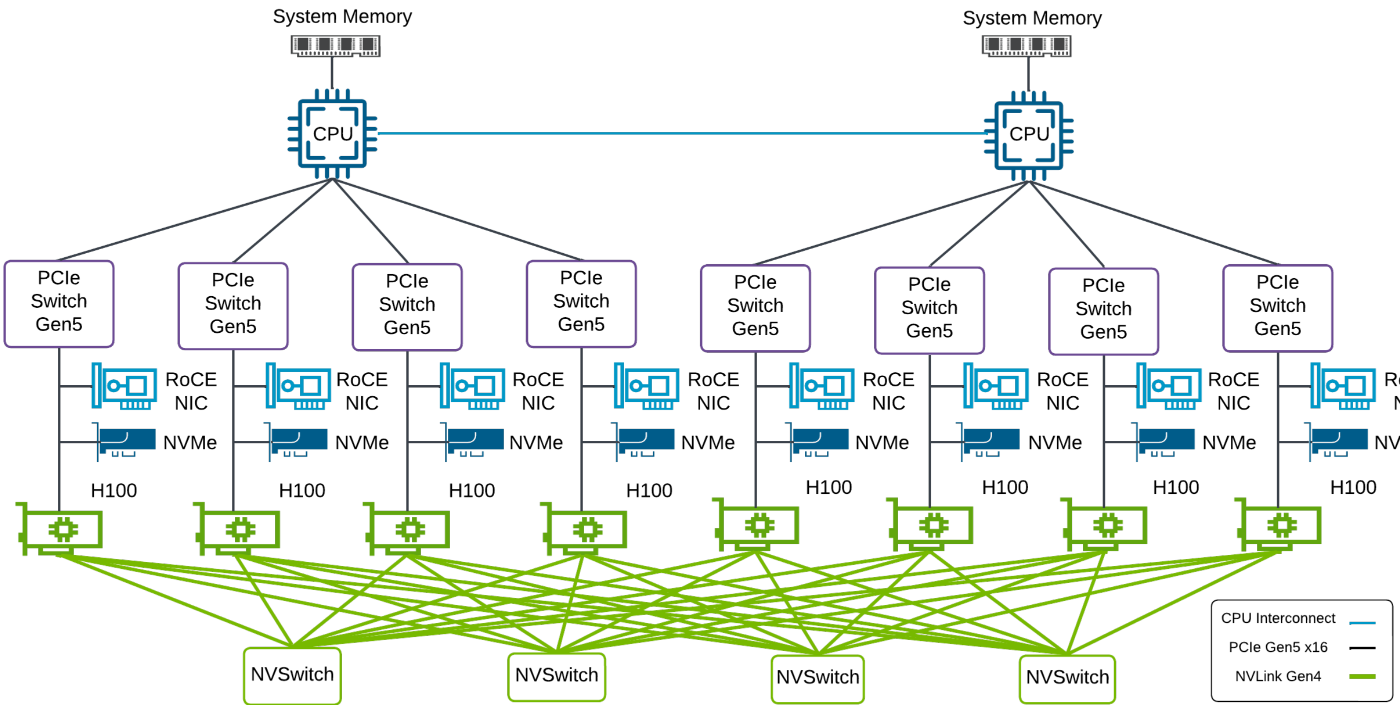


Table 7. Typical HGX server system configuration for GenAI inference

Parameter	Inference Server Configuration
GPU	H100 SXM
GPU configuration	8x GPUs connected by 4x NVSwitch chips and NVLinks within a server GPUs should be balanced across CPU sockets and root ports.
CPU	x86 PCIe Gen5-capable CPUs are recommended, such as Intel Xeon scalable processor (Sapphire Rapids) or AMD Genoa.
CPU sockets	Min 2-CPU sockets
CPU speed	Min 2.1 GHz base clock
CPU cores	Min 6x physical CPU cores per GPU
System memory	Min 1.5x (or 2x) of the total GPU memory size of DDR5 is recommended. Evenly spread across all CPU sockets and memory channels.
PCIe	Min 1x Gen5; 16 links per Gen5 GPU are recommended.
PCIe topology	For a balanced PCIe architecture, GPUs should be evenly distributed across CPU sockets and PCIe root ports. NICs and NVMe drives should be placed within the same PCIe switch or root complex as the GPUs.
PCIe switches	Direct CPU attached is preferred.
VCF network NICs	RDMA capability is not necessarily required for AI inference on an HGX server. NVIDIA ConnectX NICs or Broadcom Ethernet NICs
VCF network NIC speed	vSAN ESA requires a minimum of 25Gbps. The network for AI inference requires a minimum of 25 GbE per GPU to meet text data's bandwidth for AI inference demands. To future-proof the infrastructure, a minimum of 2x 100 GbE NICs per HGX server can be optionally used.
Out-of-band network NIC	1 GbE Ethernet NIC
Storage	Use at least 1x 3.84TB Gen4 NVMe per CPU socket. To future-proof the infrastructure, we recommend 1x 3.84TB U.2 NVMe drive for each GPU under the same PCIe switch of a GPU. For this inference reference architecture, all of the NVMe disks available will be pooled and controlled by vSAN ESA.

Table 8 provides the list of the network prerequisites for the enterprise networking team to prepare accordingly.

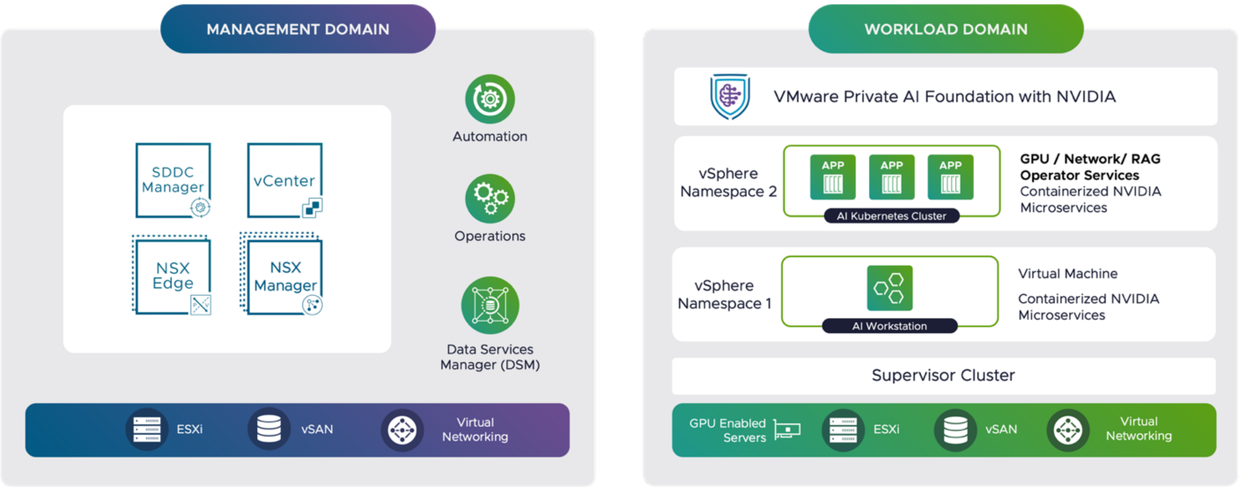
Table 8. Network to be prepared by the enterprise networking team

Network	Function	VLAN/Overlay	Routed or NAT	WLD
VM management	IP connectivity for VM management traffic	VLAN	Routed	Management WLD & VI WLD
ESXi management	IP connectivity for ESXi	VLAN	Routed	Management WLD & VI WLD
vSAN	vSAN traffic	VLAN	Routed or isolated based on L2 boundary	Management WLD & VI WLD
vSphere vMotion	vMotion traffic	VLAN	Routed or isolated based on L2 boundary	Management WLD & VI WLD
Host overlay	NSX TEP traffic	VLAN	Routed	Management WLD & VI WLD
Edge uplink01	Edge cluster northbound connectivity via eBGP	VLAN	Routed	Management WLD & VI WLD
Edge uplink02	Edge cluster northbound connectivity via eBGP	VLAN	Routed	Management WLD & VI WLD
Edge overlay	Edge node TEP VLAN	VLAN	Routed	Management WLD & VI WLD
Supervisor cluster management network	Used by the supervisor control plane nodes. This VLAN can be shared with the VM management VLAN.	VLAN	Routed	VI WLD
Supervisor namespace networks	Used to allocate IP addresses to the workload network in the supervisor namespace	Overlay	NAT	VI WLD
Service IP pool network	Used by Kubernetes applications that need a service IP address	Overlay	NAT	VI WLD
Egress IP pool network	Used by NSX to create an IP pool for load balancing	VLAN	Routed	VI WLD
Ingress IP pool network	Used by NSX to create an IP pool for NAT endpoint use	VLAN	Routed	VI WLD
Kubernetes cluster service pool network	Internal CIDR block from which IPs for Kubernetes cluster IP services will be allocated	Overlay	NAT	VI WLD

Virtual architecture

The virtual architecture in VCF is structured as two primary domains: the **management (mgmt) domain** and the **workload domain (VI WLD)**, as shown in Figure 10. This separation ensures isolation between infrastructure management and AI workloads; enhancing security, scalability, and operational efficiency; and separating lifecycle management.

Figure 10. Conceptual view of virtual architecture



The **management domain** is a dedicated set of infrastructure resources in the form of a vSphere cluster that hosts all the core management components and services required to operate the VCF environment. It requires **vSAN** and hosts critical components such as **SDDC Manager, vCenter, NSX, Automation, Operations,** and **Data Services Manager**. While these tools reside in the management domain, they extend their functionality to VI WLDs, enabling provisioning, orchestration, policy enforcement, and governance.

Workload domains exclusively run AI workloads and user applications. Best practices recommend creating separate VI WLDs for each business subsidiary in large organizations, ensuring resource isolation between different teams while operating within the same VCF infrastructure. Each VI WLD functions independently, with its own **vCenter** instance residing in the management domain to manage its virtualized resources.

Within a VI WLD, AI workloads are provisioned as **AI workstations (DLVMs)** within vSphere namespaces. As workload demands increase, **AI Kubernetes clusters** can be dynamically deployed or decommissioned based on priorities. Each cluster consists of nodes implemented as GPU-enabled VMs, allowing seamless scalability. **vSphere namespaces** not only serve as resource pool boundaries but also enforce permissions and apply policies (for example, storage policies), further enhancing isolation by segregating different users and projects. Additionally, other VMware Private AI Foundation with NVIDIA components and capabilities are deployed within vSphere namespaces.

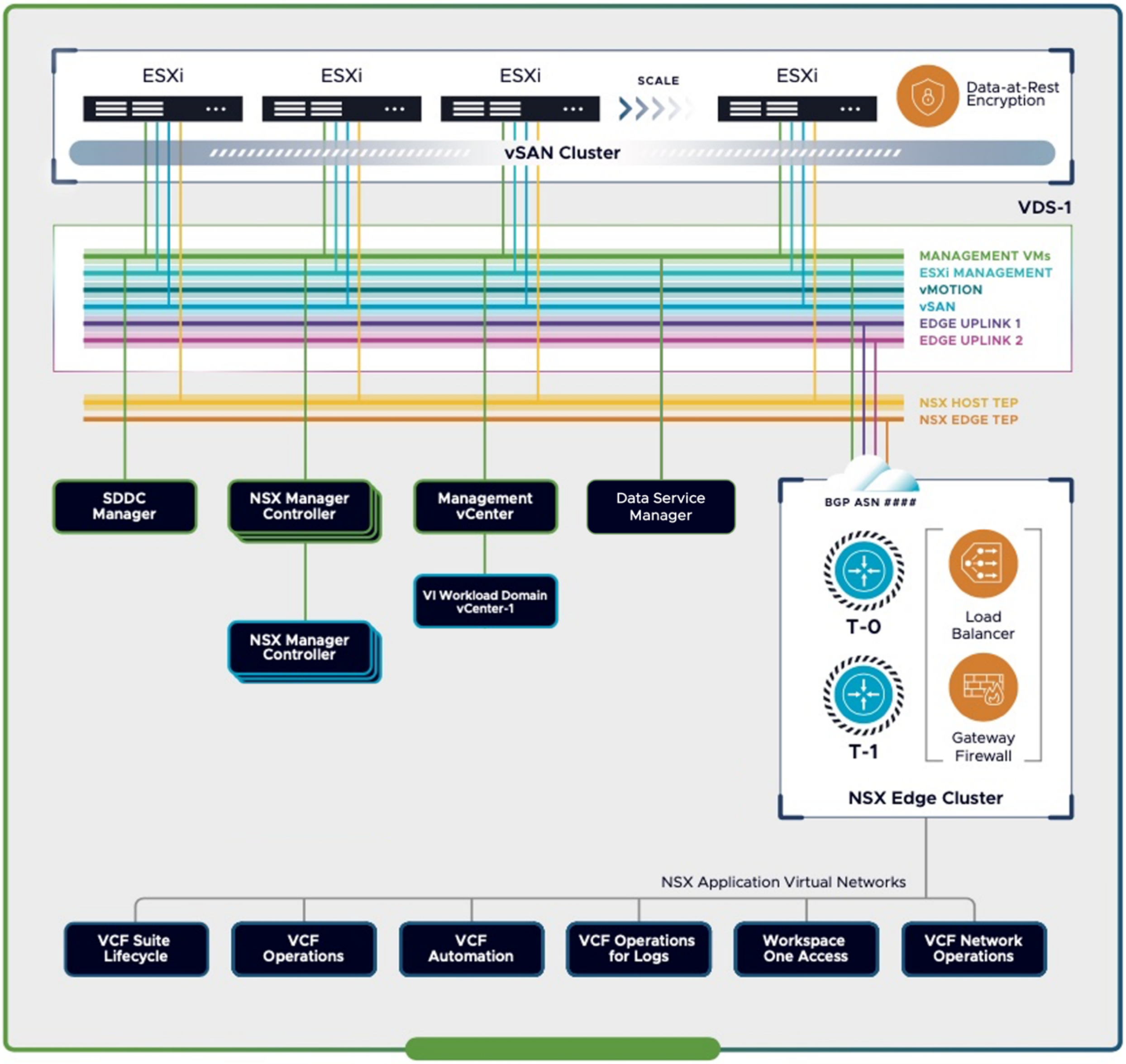
At the core of each VI WLD, layered on top of the vSphere infrastructure, is the **supervisor cluster**—a Kubernetes cluster responsible for provisioning vSphere namespaces and managing the resources within them. The supervisor serves as the critical control plane component, enabling Kubernetes orchestration within vSphere by running directly on ESXi hosts. Acting as a translation layer between Kubernetes and vSphere, the supervisor manages resource allocation and the full lifecycle of containerized workloads, including deployment, storage provisioning, and networking configuration, all while ensuring enterprise-grade security and control. New AI Kubernetes clusters can be deployed either through **Aria Automation** or a single `kubectl` command using the Kubernetes API, significantly streamlining production AI and data science operations.

Management domain

Management domain architecture

Figure 11 depicts the essential components in the VCF management domain, which requires a minimum of 4 ESXi hosts to enable the vSAN service. The network segments are color-coded to represent different port groups in the vSphere distributed switch (VDS-1); detailed settings for VDS-1 and port groups are provided in Table 10 and Table 11. The vCenter and NSX managers with green borders manage the mgmt domain, while those with blue borders oversee a workload domain (VI WLD). Additionally, the NSX Edge cluster manages ingress and egress traffic to the mgmt domain, providing security and load balancing. The Aria suite components are connected to NSX application virtual networks.

Figure 11. Management domain virtual architecture



Management domain settings

Table 9 shows the functions and settings of each component in the management domain.

Table 9. Management domain settings

Category	Component	Function	Number	Settings / Notes	Deployment Reference
SDDC	SDDC Manager	Central management UI for VCF	1	• Use Cloud Builder's default configuration of SDDC Manager to deploy • Protected by vSphere HA	SDDC Manager Design for VMware Cloud Foundation
vSphere	vCenter	Central management UI for configuring and monitoring vSphere infrastructure	• 1 for the mgmt domain • 1 for each VI WLD	• The mgmt domain's vCenter is deployed by a VMware Cloud Builder VM • VI WLD vCenter is deployed by SDDC Manager during VI WLD creation	vCenter Server Design for VMware Cloud Foundation
	vSAN	Create a single, shared storage pool across ESXi hosts in a domain	Requires at least 4 ESXi hosts	• Required by the mgmt domain • Support OSA or ESA	vSAN Design for VMware Cloud Foundation
VMware Private AI Foundation with NVIDIA	Data service manger (DSM)	Provisioning and lifecycle of PostgreSQL vector database	1	• Requires a 1:1 relationship between each vCenter and a DSM appliance to manage a VI WLD	VMware Data Services Manager Design for Private AI Ready Infrastructure for VMware Cloud Foundation
Aria Suite	Aria Automation appliance	Automate IT processes and service delivery	A cluster of 3 appliances	• Enable self-service catalog • Deployed and LCM controlled by Aria Lifecycle	Private Cloud Automation for VMware Cloud Foundation
	Aria Operations appliance	Monitor and optimize performance, capacity, and configuration	A cluster of 3 appliances (primary, data, replica)	• Provide the scale capacity required for monitoring up to 12,000 VMs or objects • Support scale-out with additional data nodes.	Intelligent Operations Management for VMware Cloud Foundation
	↳ Aria Operations cloud proxies	Enhance scalability by collecting data from each VCF instance and sending it to the Aria Operations cluster	2 appliances deployed on the local-instance NSX segment	• Deployed and LCM controlled by Aria Lifecycle	Configuring Cloud Proxies in VMware Aria Operations
	Aria Suite Lifecycle	Lifecycle management for Aria Suite and vIDM	1 on the cross-instance NSX segment	• Automate deployment, configuration, patching, upgrade, and content management across Aria Suite • Enable VCF mode • Protected by vSphere HA	VMware Aria Suite Lifecycle Design for VMware Cloud Foundation

Category	Component	Function	Number	Settings / Notes	Deployment Reference
Workspace ONE	VMware Identity Manager (vIDM)	Central identity and access management solution for SSO, authentication, and user directory integration	A cluster of medium-sized appliances	• Deployed and LCM controlled by Aria Lifecycle • Clustered vIDM to support clustered deployment of Aria Automation • Protected by vSphere HA	Workspace ONE Access Design for VMware Cloud Foundation
NSX	NSX Manager	Central management UI of NSX for configuring and monitoring network components in mgmt domain	A cluster of 3 appliances	• Deployed by Cloud Builder with a virtual IP (VIP) address and an anti-affinity rule to ensure NSX managers are running on different ESX hosts	Logical Design for NSX for VMware Cloud Foundation
	NSX Edge cluster	• Provide HA, scalability, and distributed firewall services • Support application virtual networks (AVNs)	1	• Profile Type: Set to Default. • Network Configuration: Layer 2 (L2) Uniform.	Installing NSX Edge
	↳ Tier-0 router	North-south connectivity between NSX logical network and physical infrastructure	1	• Active/Active (ECMP) routing for redundancy and load balancing • BGP configured with unique ASNs for local and remote peers • Large form factor to handle high throughput	NSX Design for VMware Cloud Foundation
	↳ Tier-1 router	East-west routing for internal network communication and stateful services	1 or more	• Active/Passive setup for stateful services like NAT and load balancing.	NSX Design for VMware Cloud Foundation

↳ This symbol signifies the component is part of the larger component above it.

Tables 10–11 present the VDS settings and its port groups (network) in the mgmt domain.

Table 10. Management domain VDS settings

VDS Configuration	Setting	Considerations	Deployment Reference
Single VDS	Single VDS with minimum 2 uplinks (NICs)	• A single VDS prepared for NSX enhances operational simplicity. • If using only 2 uplinks, enable traffic sharing on the same NICs. • Using 4+ uplinks is recommended to ensure proper traffic segmentation and bandwidth allocation.	vSphere Networking Design for VMware Cloud Foundation
Network I/O Control	Enabled	• Allow for proper bandwidth prioritization, increasing resiliency and performance of the network.	vSphere Networking Design for VMware Cloud Foundation
Data Path Mode	Enhanced Data Path	• Recommended mode for vSphere clusters running NSX Edge nodes.	vSphere Networking Design for VMware Cloud Foundation
NSX Transport Zones	VLAN and overlay transport zones	• Allows for flexible network segmentation and supports both traditional VLAN-based and overlay-(Geneve)-based networking.	vSphere Networking Design for VMware Cloud Foundation

Table 11. Management domain VDS port group settings

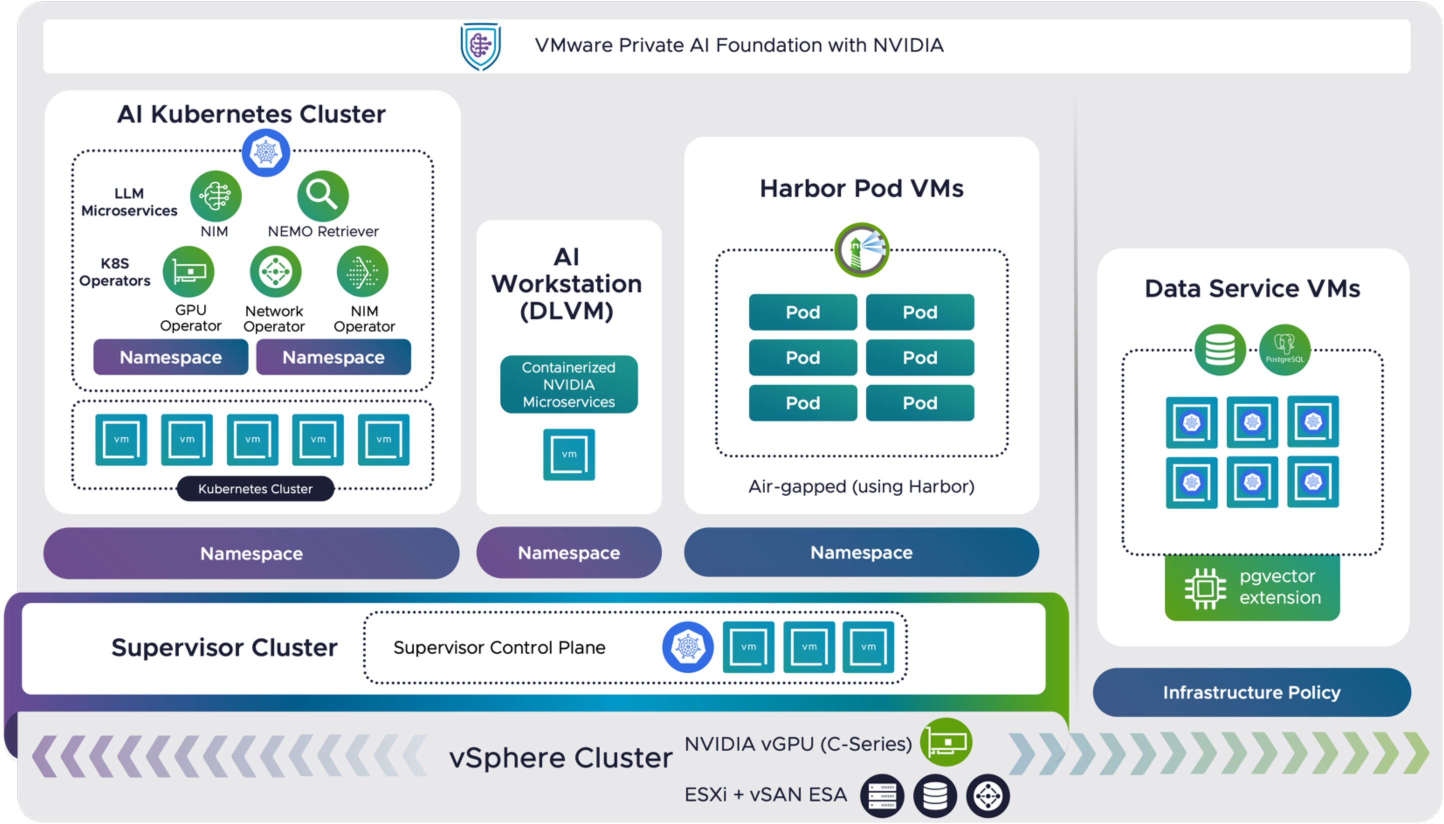
Port Group Name	Function	VMkernel Adapter Required?	MTU	Teaming Policy
VM management	IP connectivity for VM management traffic	No	1500	Route based on physical NIC
ESXi management	IP connectivity for ESXi	Yes	1500	Route based on physical NIC
vSAN	vSAN traffic	Yes	9000	Route based on physical NIC
NFS (optional)	NFS traffic for supplemental storage	Yes	9000	Route based on physical NIC
Host overlay	NSX Fabric TEP traffic	Yes	9000	Not applicable
Edge uplinks and overlay	NSX Ege Nodes TEP traffic and uplink fabric traffic	No	9000	Explicit failover

Workload domain

Workload domain architecture

Figure 12 provides an expanded view of the VI workload domain architecture compared to the conceptual overview in Figure 10. At its base, the vSphere cluster enables the Supervisor cluster, supporting vSphere namespaces that facilitate resource isolation, governance, and policy enforcement. Within these namespaces, users can deploy diverse workloads such as AI Kubernetes clusters, AI workstations, and RAG applications—provisioned and managed by the Supervisor. The Supervisor can also integrate Harbor, a unified repository for container images and AI models, supporting OCI-compatible formats for offline or air-gapped environments. VMware Data Services Manager (DSM), running in the management domain, uses the vSphere infrastructure policy to deploy database VMs to the registered VI WLD. These database VMs then initiate a Kubernetes cluster and run vector database containers within pods. All components are pre-configured or customizable as service catalogs within Aria Automation, ensuring seamless operation in VMware Private AI Foundation with NVIDIA deployments.

Figure 12. Workload domain virtual architecture



Workload domain settings

Table 12 details the functions and configurations of each component within the workload domain. Notably, the settings of the two vSphere distributed switches (VDS) and port groups in the workload domain mirror those used in the management domain.

Table 12. Workload domain settings

Category	Component	Function	Number	Settings/Notes	Deployment Reference
vSphere	vCenter in management domain	Central management UI for configuring and monitoring vSphere infrastructure	1 per VI WLD	• Deployed by SDDC Manager during VI WLD creation • Located within the management domain • Integrated into the pre-existing SSO domain in the management domain	VMware Cloud Foundation 5.2 Design Guide
	vSAN	Create a single, shared storage pool across ESXi hosts in a domain	Require at least 4 ESXi hosts	• Enable ESA with RAID5 policy for performance and storage space efficiency	vSAN Design for VMware Cloud Foundation
	VDS	Provide virtual networking across multiple ESXi hosts	2 per VI WLD	• Each VDS configured with 2 uplinks (NICs) to ensure traffic isolation between overlay-based networks and GPU/data networks on non-overlay networks. • The second VDS is prepared for NSX integration	NSX Reference Design Guide 4.2 vSphere Networking Design for VMware Cloud Foundation
	Host VIBs	Enable NVIDIA vGPU and fabric management for NVSwitch and NVlink-based systems	2 VIBs per ESXi host	• Recommend using vLCM images to install in the cluster of VI WLD • NVIDIA vGPU host VIB contains vGPU host driver and NVSwitch fabric management • mgmt-daemon VIB for monitoring GPU metrics in Aria Ops	Installing and configuring the NVIDIA vGPU Manager VIB Deploy a GPU-Accelerated VI Workload Domain for VMware Private AI Foundation with NVIDIA
VKS	Content Library	Centralized repository to store and manage DLVMs' templates, Kubernetes releases, and OVAs	1 or more	• Import DLVM template from link • Use the check-in/check-out feature for version control of VM templates • Configured as a subscribed library to synchronize DLVM templates published by VMware	Create a Content Library with Deep Learning VM Images for VMware Private AI Foundation with NVIDIA
	VM classes	Define resource allocations for Kubernetes cluster nodes or VMs provisioned by VM service	As needed	• Logical entities that specify the resources assigned to VMs or Kubernetes nodes, including CPU, memory, PCI devices, and device groups (vGPUs with NVSwitches,	Configure vGPU-Based VM Classes for AI Workloads for VMware Private AI Foundation with NVIDIA

Category	Component	Function	Number	Settings/Notes	Deployment Reference
				NVLinks, and virtual functions on NICs).	
	Supervisor Contour service	Ingress service for supervisor services	Per supervisor cluster	• Default values for Contour supervisor service. • Contour is required to enable the Harbor and model store supervisor services	Install Contour as a Supervisor Service
	Supervisor Harbor service	Serves as a private registry for storing and managing OCI-compliant artifacts, such as container images and AI models.	Per supervisor cluster	• Update `harbor-data-values.yml` file to specify settings like FQDN and Storage Class.	Setting Up a Private Harbor Registry in VMware Private AI Foundation with NVIDIA
NSX	NSX Manager in mgmt domain	Central management UI of NSX for configuring and monitoring network components in VI WLD	A cluster of 3 appliances per VI WLD	• Deployed by SDDC Manager during VI WLD creation • Separate NSX Manager deployment for isolating permissions, policies, services, etc.	NSX Design for VMware Cloud Foundation
	NSX Edge cluster	Grouping of one or more edge nodes, representing a pool of capacity for NSX services	1 or more per VI WLD	• Profile Type: Set to default. • Network configuration: Layer 2 (L2) uniform	
	↳ Tier-0 router	Provides north-south connectivity between the NSX logical network and the physical infrastructure	1	• Active/Active mode with ECMP routing • BGP configured • ASNs defined (local & remote) • Large form factor	
	↳ Tier-1 router	Offers east-west routing functionality, connecting to logical switches for internal network communication and providing stateful network services.	1 or more	• Configured in Active/Passive mode for VKS enablement	

VMware Private AI Foundation with NVIDIA prerequisites

After provisioning the above VI WLD, infrastructure architects must prepare it for the VMware Private AI Foundation with NVIDIA. This preparation involves installing the necessary ESXi VIBs to enable NVIDIA vGPU capabilities and GPU metrics at the host level, licensing the solution, and performing other essential infrastructure tasks as outlined in Table 13. For more information refer to Preparing VMware Cloud Foundation for a Private AI workload deployment.

Table 13. VMware Private AI Foundation with NVIDIA prerequisites

Category	Component	Function	Number	Settings / Notes	Deployment Reference
NVIDIA	License server	Manage and distribute licenses for NVIDIA software products including vGPU	1	• Can be deployed by Cloud License Service (CLS) or Delegated License Server (DLS)	NVIDIA License Server Documentation
vSphere Content Library	DLVM template	Used by VCF Automation when serving requests from self-service catalog	As needed	• A set of VM images is shipped and maintained by Broadcom • Stored as templates in a VCF content library	Create a Content Library with Deep Learning VM Images for VMware Private AI Foundation with NVIDIA
Aria Automation	NVIDIA licensing portal API key and client token file	Facilitates interactions with the NVIDIA Licensing Portal to obtain licenses from CLS or DLS	1	• API key with the following access types: – Licensing State – Software Downloads – Enterprise	Requirements for Deploying VMware Private AI Foundation with NVIDIA
	NGC API key	Authenticate access and download NGC catalog items	1	• Only NGC key with NVAIE entitlements can access the assets on NGC. • We recommend using an NGC API key that is not tied to an individual user's account and is managed by your IT department	

Validation

Table 14 presents product demonstrations of VMware Private AI Foundation with NVIDIA, highlighting its capabilities and providing deployment references for validation purposes. We encourage you to review these deployment procedures and capabilities to gain a comprehensive understanding of the platform. Additionally, we provide curated official documentation and blog posts to assist you in your deployment efforts. For more use cases for implementing VMware Private AI Foundation with NVIDIA, check these two blogs (part 1 and part 2) and this talk about a real customer RAG journey.

Table 14. VMware Private AI Foundation with NVIDIA use cases

Use Cases	Deployment Reference
Deploy a DLVM	• Deploy a DLVM by Using a Self-Service Catalog (official doc) • Section "Improved Experience for the Data Scientist" in VMware Private AI Foundation with NVIDIA a Technical Overview (blog post)
Deploying a RAG chatbot in a DLVM (product demo)	• Deploy a Deep Learning VM with a RAG Workload (official doc) • Section "A Chatbot Application that Uses Retrieval Augmented Generation (RAG)" in VMware Private AI Foundation with NVIDIA - a Technical Overview (blog post) • Building Production-grade AI-driven Apps on VMware Private AI Foundation with NVIDIA NIM (blog post)
Deploy an AI Kubernetes cluster	• Deploy a GPU-Accelerated TKG Cluster by Using a Self-Service Catalog Item in VMware Aria Automation (official doc)
Deploying a RAG chatbot app in a Kubernetes cluster (product demo)	• VMware Private AI Foundation with NVIDIA - a Technical Overview (blog post)
Model Governance (product demo)	• Storing ML Models in VMware Private AI Foundation with NVIDIA (official doc) • Onboarding Llama3 to the Private AI Model Gallery (blog post)
Integrate DSM into Service Catalog (product demo)	• VMware Data Services Manager Design for Private AI Ready Infrastructure (official doc) • Private AI Automation Services Enhancements in VMware Cloud Foundation 5.2.1 (blog post) • Step-by-step: Deploy DSM using Aria Automation (blog post)
GPU Monitoring Dashboard (product demo)	• Monitoring VMware Private AI Foundation with NVIDIA (official doc) • Section "Monitoring GPU Consumption and Availability" in VMware Private AI Foundation with NVIDIA - a Technical Overview (blog post)

Performance

This section explains how to run and validate the GenAI-Perf benchmark, which is a useful command-line tool for measuring the throughput and latency of generative AI models served through an inference server. This tool helps you determine whether the virtual infrastructure meets your performance requirements, or you can use it to establish a baseline. We provide a performance comparison between bare metal and virtualized configurations. The results show that the difference between virtual and bare metal falls into a statistically negligible variance. The following instructions are adapted from this step-by-step documentation on Using GenAI-Perf to Benchmark, We provide these instructions as reference examples and do not guarantee performance.

Benchmarking with GenAI-Perf

Here, we provide the steps we followed to collect performance data for our performance comparison between bare-metal and virtualized systems.

Step 1. Deploy a DLVM

We used the service catalog in Aria Automation to deploy a DLVM with 4x H100-80c vGPU connected via NVSwitch and NVLink. Table 15 specifies the DLVM's settings, NIM, and Triton-inference-server container used in our benchmarking.

Table 15. DLVM settings and components used in benchmarking

Component	Settings
DLVM	• 24 vCPU, 256 GB memory, 256 GB disk • Nvidia: 4@nvidia_h100xm-80c%NVLink • Enable UVM by setting `pciPassthru[0~3].cfg.enable_uvm = 1` • `pciPassthru.64bitMMIOSizeGB = 1024 GB`
`nvcr.io/nim/meta/llama-3.1-70b-instruct`	• 1.3.3 • Profile: tensorrt_llm-h100-fp8-tp4-pp1-throughput
tritonserver	24.10-py3-sdk

Note: (1) **VM RAM size**: The required RAM size for the VM is dependent on the specific workload. For the GenAI-Perf benchmark test, 16GB of RAM has been found to be sufficient. However, 256GB is allocated in this example to accommodate potential additional frameworks or applications that may demand more RAM.

(2) **vGPU and MMIO Size**: When utilizing vGPU, the `pciPassthru.64bitMMIOSizeGB` setting is not necessary. The default MMIO size for a VM is 32GB, which is sufficient for up to 8 H100 vGPU profiles (each requiring 4GB MMIO). This is more than adequate for the 4 H100 vGPU configuration in this example. If the VM is attached to 8 GPUs and also has InfiniBand or Ethernet NICs attached via passthrough, then `pciPassthru.64bitMMIOSizeGB` should be set to "64".

(3) **Direct/Dynamic Passthrough GPU and MMIO Size**: If direct or dynamic passthrough GPU is chosen instead of vGPU, each passthrough H100 GPU necessitates a 128GB MMIO size. Therefore, if 8 H100/H200 GPUs and 8 NICs are passed through to the VM, `pciPassthru.64bitMMIOSizeGB` must be set to "1056" (calculated as 128*8 + 32). Minimizing the MMIO size is crucial as a very large MMIO size can prolong the VM's boot process (a few seconds) due to the increased time required for the system to allocate and configure the MMIO address space for virtualized devices.

Step 2. Launch NIM in a DLVM

After deploying the DLVM, we followed the scripts shown in Code Example 1 to launch NIM. To initiate a TensorRT-LLM-backed NIM, we set the two additional environment variables specified in Table 16. For other NIM configuration parameters, refer to the NIM's Getting Started and Configuring a NIM pages.

Table 16. NIM parameters to launch TensorRT-LLM-backed NIM

Parameter	Value	Consideration
NCCL_CUMEM_ENABLE	0	Disables NCCL's cuMem allocator to help avoid certain NCCL-related issues; for example, increased memory overhead during CUDA graph captures.
shm-size	16GB or higher	Increasing the shared memory size is beneficial for applications that require interprocess communication or large shared memory segments. For example, Docker containers are allocated 64MB of shared memory by default.

Code Example 1 launches a Docker NIM container for the `llama-3.1-70b-instruct` model with `tensorrt_llm-h100-fp8-tp4-pp1-throughput` profile. To use a different NIM profile, replace the `NIM_MODEL_PROFILE` with a different value in the `list-model-profiles` command. To launch a container for a different NIM, replace the value of `Repository` with the value of the other NGC `image list` command using the NGC CLI tool and change the value of `CONTAINER_NAME` to something appropriate.

Code Example 1. TensorRT-LLM-backed NIM launch scripts

```
# Export the NGC API key, then Docker login to NGC
export NGC_API_KEY=YOUR_KEY
echo "$NGC_API_KEY" | docker login nvcr.io --username '$oauthtoken' --password-stdin

# (Optional) List available NIMs. This step requires to install NGC CLI.
ngc registry image list --format_type csv nvcr.io/nim/*

# Choose a container name for bookkeeping
export CONTAINER_NAME=Llama-3.1-70b-instruct

# The container name from the previous ngc registgry image list command
Repository=nim/meta/llama-3.1-70b-instruct
Tag=1.3.3

# Choose a LLM NIM Image from NGC
export IMG_NAME="nvcr.io/${Repository}:${Tag}"

# Choose a path on your system to cache the downloaded models
export LOCAL_NIM_CACHE=~/.cache/nim
mkdir -p "$LOCAL_NIM_CACHE"

# List the compatible profiles of the NIM
docker run --rm --runtime=nvidia -e NGC_API_KEY --gpus=all $IMG_NAME list-model-profiles

export PORT=8000
export NCCL_CUMEM_ENABLE=0
export NIM_MODEL_PROFILE=tensorrt_llm-h100-fp8-tp4-pp1-throughput

docker run -it --rm --name=$CONTAINER_NAME-$PORT \
    --runtime=nvidia \
    --gpus all \
    -e NGC_API_KEY \
    -v "$LOCAL_NIM_CACHE:/opt/nim/.cache" \
    -u $(id -u) \
    -p $PORT:8000 \
    -e NIM_MODEL_PROFILE \
    -e NCCL_CUMEM_ENABLE \
    --shm-size=16GB \
    $IMG_NAME
```

Step 3. Launch GenAI-Perf

While the GPU-enabled NIM container from the previous step remained active and running, we interactively launched a Triton inference server container without GPU support. Within this CPU-only container, we used GenAI-Perf with the parameters shown in Table 17 to conduct a warm-up load test on the NIM backend. (For additional model input parameters, refer to this link.) Consequently, the GenAI-Perf tool in the CPU-only Triton container sent prompt requests to the GPU-enabled NIM container, which validated its functionality.

Table 17. GenAI-perf launch parameters

Parameters	Value
Input Sequence Length	200
Output Sequence Length	50
Output Sequence Std	10
Concurrency	10

Note: With concurrency=N, GenAI-Perf maintains N active inference requests during profiling. For example, with a concurrency of 4, it sustains 4 simultaneous requests, issuing a new request as each one completes.

For understanding additional metrics and parameters to run GenAI-Perf, consult the Metrics and Parameters and Best Practices pages.

Code Example 2. Setting Up GenAI-Perf and warm-up load test

```
# Launch a triton inference server container with only CPU
export RELEASE="24.10"
docker run -it --rm --runtime=nvidia \
        --net=host \
        -v $(pwd):/workspace/host \
        nvcr.io/nvidia/tritonserver:${RELEASE}-py3-sdk \
        /bin/bash

# Log in with your Huggingface credential for accessing llama-3 tokenizer
pip install huggingface_hub
huggingface-cli login

# Run GenAI-Perf within triton inference server container for warm up
export INPUT_SEQUENCE_LENGTH=200
export INPUT_SEQUENCE_STD=0
export OUTPUT_SEQUENCE_LENGTH=50
export CONCURRENCY=10

genai-perf profile \
    -m meta/llama-3.1-70b-instruct \
    --endpoint-type chat \
    --service-kind openai \
    --streaming \
    -u localhost:8000 \
    --synthetic-input-tokens-mean $INPUT_SEQUENCE_LENGTH \
    --synthetic-input-tokens-stddev $INPUT_SEQUENCE_STD \
    --concurrency $CONCURRENCY \
    --output-tokens-mean $OUTPUT_SEQUENCE_LENGTH \
    --extra-inputs max_tokens:$OUTPUT_SEQUENCE_LENGTH \
    --extra-inputs min_tokens:$OUTPUT_SEQUENCE_LENGTH \
    --extra-inputs ignore_eos:true \
    --tokenizer meta-llama/Meta-Llama-3.1-70B-Instruct \
    -- \
    -v \
    --max-threads=256
```

Benchmark results of virtual vs. bare metal

After we completed Step 3 (above) in using GenAI-Perf to benchmark our virtual and bare-metal testbeds, a sweep across varying concurrency levels—1, 2, 5, …, up to 125 was performed—and results for larger workloads with an input sequence length (ISL) and output sequence length (OSL) were paired to represent a summarization chatbot (ISL=7000, OSL=1000).

Table 18 details the hardware configurations used for running the workload on both bare-metal and virtualized systems. The key distinction is that the virtualized setup uses virtualized NVIDIA H100 GPUs, labeled as H100-80c vGPU.

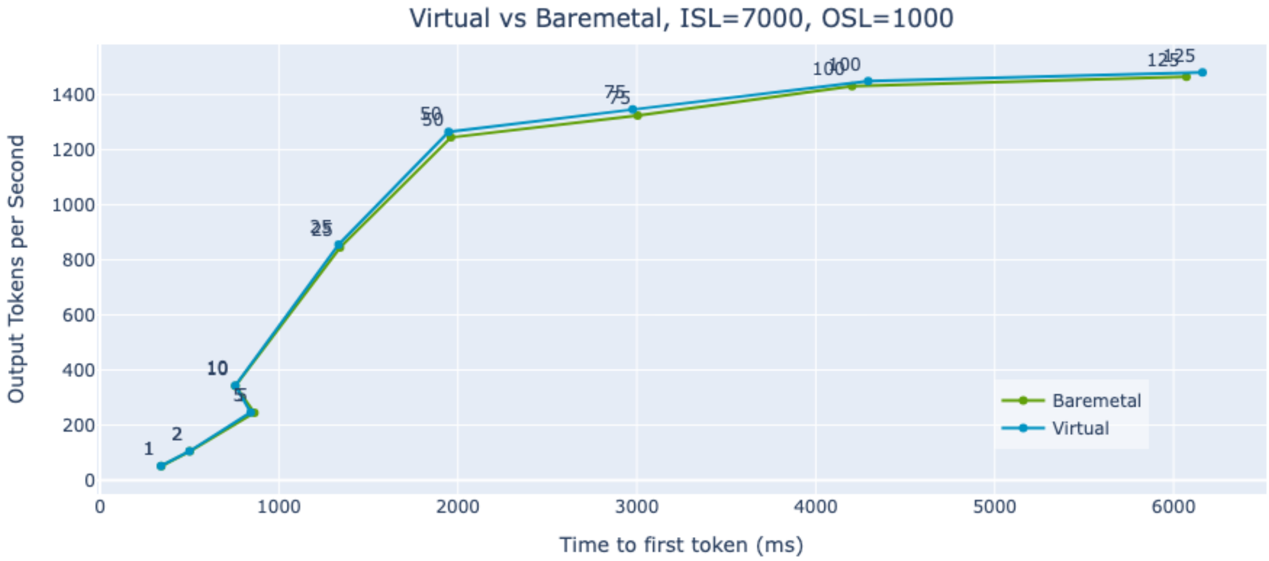
To ensure a fair comparison, we launched NIM with Tensor Parallelism using 4x H100 GPUs. In the bare-metal setup, only 4 out of 8 GPUs were utilized, allowing for a direct comparison to the virtual configuration, which used 4 NVIDIA vGPUs attached to the VM. We used online mode in GenAI-Perf by setting `--streaming` to the benchmark online application. The throughput (output tokens per second) for the summarization chatbot was then plotted against the time to first token (TTFT), which is shown in Figure 13.

> **Note:** The performance results presented here reflect observed results under specific conditions. We cannot guarantee identical performance because it may vary depending on your hardware, software configuration, and workload. Future releases or software updates might improve these results.

Table 18. Bare metal vs. virtual server configurations for virtualized H100

Component	Bare metal	Virtual
Logical processors	208	24 allocated to the VM (**184 available for other VMs/workloads**)
GPUs	8x NVIDIA H100-SXM-80GB 4x NVSwitch, NVLink Gen4	4x NVIDIA H100-80c vGPU via NVLink (4@nvidia_h100xm-80c%NVLink)
Memory	2TB	256 GB (**1.74 TB for other VMs/workloads**)
Storage	8x 3.84TB NVMe SSD	256 GB VM Hard Disk on vSAN
OS	Ubuntu 22.04.3	Ubuntu 22.04.03 DLVM in vSphere 8.0.3
GPU Drivers	Data Center Driver 550.144.03	• NVIDIA vGPU Host and Guest v17.5 • 550.144.02 (Host), 550.144.03 (Guest)
CUDA	12.4	12.4
NIM	• llama-3.1-70b-instruct • Tag: 1.3.3 • Profile: tensorrt_llm-h100-fp8-**tp4**-pp1-throughput	• llama-3.1-70b-instruct • Tag: 1.3.3 • Profile: tensorrt_llm-h100-fp8-**tp4**-pp1-throughput
tritonserver	24.10-py3-sdk	24.10-py3-sdk
GenAI-Perf	0.0.11	0.0.11

Figure 13. Throughput and TTFT: Virtual vs bare metal across concurrency levels (1~125)



Plot description:

- The x-axis represents latency (TTFT).
- The y-axis represents throughput (tokens per second).
- Each point with the same color on the plot corresponds to a measurement from the same underlying model (llama3-70b) and device configuration (type of GPUs in use).
- Points are connected by lines, with the concurrency level indicated (1, 2, …, 125).

Interpreting the graph:

- The optimal points lie closest to the top-left corner of the plot.
- Higher points indicate higher throughput.
- Points further to the left indicate lower latency.

Figure 14 helps illustrate the tradeoff between latency and throughput in inference. A higher concurrency leads to better throughput and, consequently, improved GPU utilization, but it increases latency.

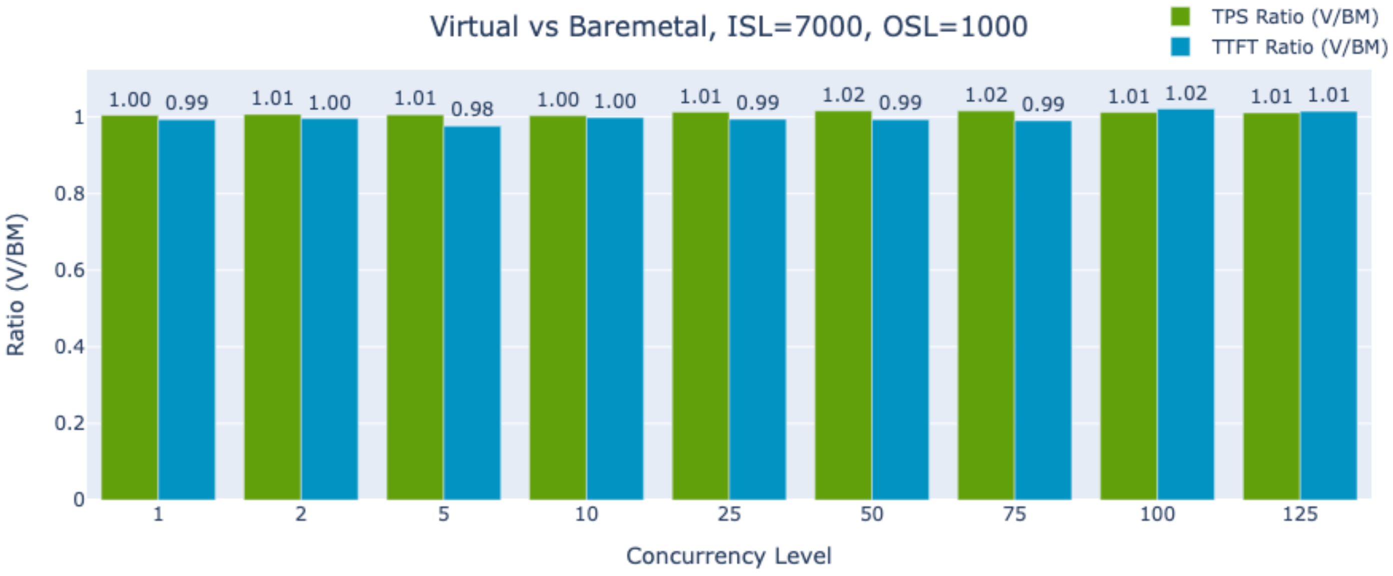
- **For online applications**, where minimizing latency is crucial, it's important to set a maximum acceptable TTFT. For example, if your chatbot requires a TTFT of no more than 1 second, you should select the highest concurrency level that doesn't exceed this threshold.
- **For offline applications**, where latency is less of a concern, you can use a higher concurrency level to maximize throughput.

Figure 14 compares throughput and TTFT ratios between virtual and bare-metal setups across varying concurrency levels, evaluating summarization use case with distinct ISL and OSL pairs. For throughput, higher values indicate better performance with virtual GPUs compared to bare metal, and vGPU delivers 1%~2% better throughput. For TTFT, lower values indicate lower latency, and we find that at certain concurrency levels, virtual GPUs exhibit 1%–2% lower latency, while at other concurrency levels, vGPU shows up to 2% higher latency.

Additionally, only 24 out of the total 208 logical CPU cores were utilized for inference in both configurations, leaving the remaining 184 logical CPU cores available for other tasks in the data center. Similarly, we used only 256GB of CPU memory for the inference workload, while reserving 1.74TB for other applications, highlighting the benefits of resource isolation.

From this, we conclude that NVIDIA vGPU (C-Series) offer near-bare-metal performance for this workload, positioning them in the "Goldilocks Zone." This means they strike an ideal balance between delivering near-native performance and offering the advantages of easier data center management and cost savings.

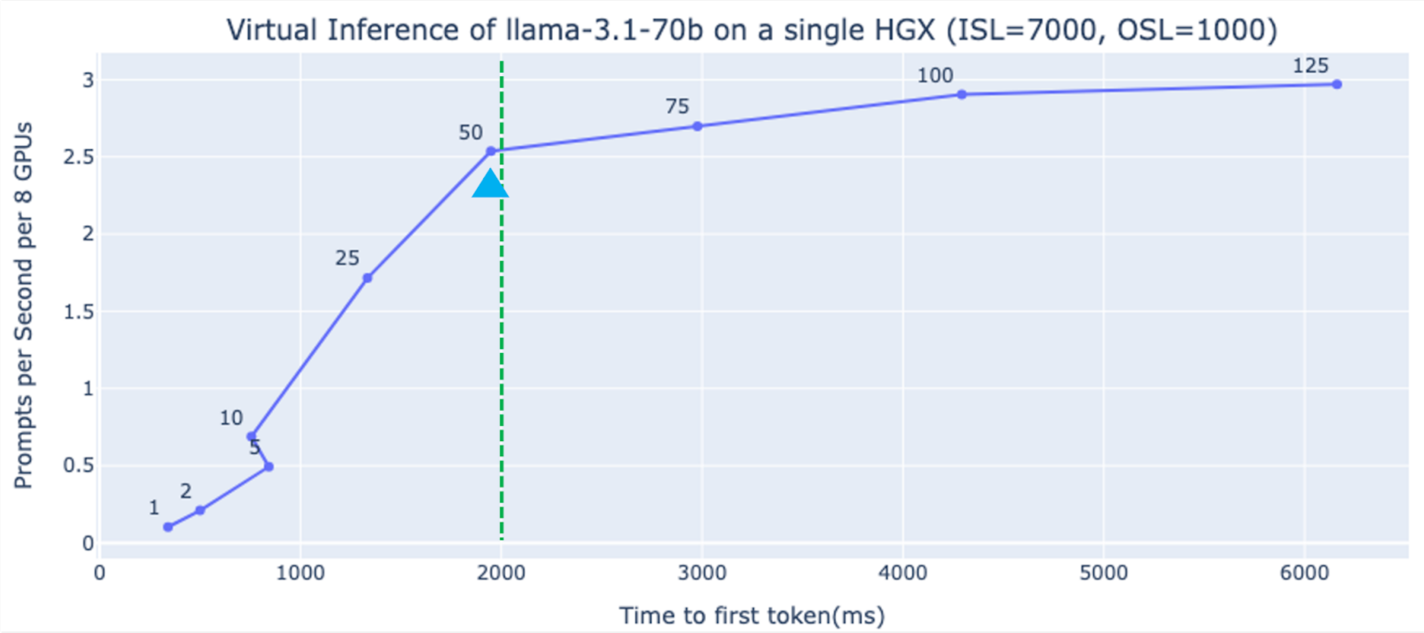
Figure 14. Throughput and TTFT Ratio: virtual vs. bare metal across concurrency levels (1~125)



Inference sizing guidance

Based on the GenAI-Perf benchmarks, you can estimate your system's capability for specific use cases. For example, a summarization chatbot with an input sequence length (ISL) of 7000 and an output sequence length (OSL) of 1000 requires a time to first token (TTFT) of under 2 seconds. Using a single HGX server with 8x H100 GPUs, you could deploy two DLVMs, each using a device group of 4 H100 vGPU profile with NVLink and NVSwitches support. According to the benchmark data, this configuration achieves a peak performance of approximately 2.54 prompts per second on an 8x H100 setup, as shown in Figure 15.

Figure 15. Throughput (prompts per second per 8 GPUs for ISL=7k, OSL=1k) on a single HGX server



Assuming an 8-hour workday, this results in 73.2K requests per day (2.54 × 28,800). If each user sends 3 requests daily, this system could support approximately 24.4K daily active users (73.2K ÷ 3). This workload would process around 512 million input tokens (7000 × 73.2K requests) and 73 million output tokens (1000 × 73.2K requests) per day. Similarly, a different use case with (ISL=200, OSL=200) is also listed in Table 19. In both cases, the number of daily active users is served with only one HGX H100 server. To support more users, additional replicas across multiple HGX servers working in parallel would be required.

These estimates demonstrate how the benchmark results can be extrapolated to assess capacity for real-world applications, such as chatbots, based on your infrastructure and performance requirements. Additionally, the data can be used to conduct a total cost of ownership (TCO) analysis—calculating costs per input/output token—by factoring in your total on-prem infrastructure expenses, including server costs, hosting costs, and software licensing.

Table 19. Estimation of Daily Active User using 1x HGX server

Metric	7000 in, 1000 out	200 in, 200 out
Input Tokens	7000	200
Output Tokens	1000	200
TTFT Requirement	< 2s	< 500 ms
Peak Prompts/s per 8x H100 NVIDIA vGPU	2.54	48.1
Concurrency level	50	125
Requests (k) per Day (8h)	73.2	1382.4
Requests per Person	3	3
Daily Active Users (k)	24.4	460.8
Total Input Million Tokens/Day	512.1	276.5
Total Output Million Tokens/Day	73.2	276.5

If you haven't purchased GPU servers yet, we recommend reviewing VMware's LLM Inference Sizing and Performance Guidance blog post. It provides hardware-based metrics for theoretical calculations, independent of benchmark results. Additionally, you can contact Broadcom's VCF Professional Services team, who can offer tailored recommendations and support to ensure your AI deployment is efficient and cost-effective.

Conclusion

The integration of VMware Private AI Foundation with NVIDIA on NVIDIA-certified HGX systems offers a robust and scalable infrastructure to address the growing demands of AI inference workloads in enterprise environments.

By optimizing GPU utilization and offering a cloud-like interface for data scientists, this architecture empowers teams to maximize their resources while maintaining governance and compliance. The seamless integration of VMware Cloud Foundation and VMware Private AI Foundation with NVIDIA ensures that IT teams can efficiently manage infrastructure and enforce security policies while offering data scientists the autonomy to innovate and focus on AI model development.

The detailed deployment considerations, product validation, and benchmarking results presented in this paper offer organizations the best of both performance and governance in the private cloud environment.

Additional information

- VMware Private AI Foundation with NVIDIA webpage
- Three Reasons Customers Choose VMware Private AI from Broadcom - Tech Field Day
- VMware Private AI Foundation Capabilities and Features Update from Broadcom - Tech Field Day
- AI Model Security and Governance - Broadcom VMware Private AI Model Gallery Demo - Tech Field Day
- Real-World Customer Journey with VMware Private AI from Broadcom - Tech Field Day
- DeepSeek-R1 Now Live With NVIDIA NIM

About the authors

Dr. Yuankun Fu is a performance engineer at Broadcom focusing on optimizing AI and HPC performance.

Agustin Malanco is a solution architect at Broadcom focusing on AI platforms and solutions.

Dr. Ramesh Radhakrishnan is a distinguished engineer leading the AI Platforms and Solutions team at Broadcom.

Acknowledgments

The authors thank Justin Murray, Vrushal Dongre, Shobhit Bhutani, and Julie Brodeur from Broadcom's VMware Cloud Foundation division and Joe Cullen from NVIDIA for reviewing and improving the paper.