TECHNICAL WHITE PAPER: March 2025

Integrating VMware Cloud Foundation with Azure Machine Learning

Technical Overview

vmware by Broadcom

Table of contents

Note: This solution provides general solution design and deployment guidelines for running the Microsoft Azure Machine Learning workloads on VMware Cloud Foundation. The hardware used in this paper is not strictly tied to the solution architecture. The solution architecture applies to any compatible hardware platforms running VMware Cloud Foundation.

Executive Summary

Business Case

The integration of VMware Cloud Foundation (VCF) with Azure Machine Learning (AML), facilitated by Azure Arc and VMware vSphere Kubernetes Service (VKS), formerly known as Tanzu Grid Service (TKGs), provides organizations with a powerful solution for running hybrid AI/ML workloads spanning both the Azure Cloud and on-premises data center or edge sites. This approach enables businesses to leverage AML's advanced capabilities even when their data and compute resources must stay on-premises. By utilizing the AML Arc extension on VKS, organizations can implement the AI-driven analytics and machine learning models they develop in the cloud, on-prem, effectively transforming their local data into actionable insights.

VMware Cloud Foundation offers enterprises a flexible and scalable cloud-like infrastructure that supports various workloads, including machine learning applications. This integration allows organizations to deploy GPU-accelerated and non-GPU-accelerated environments optimized for diverse ML tasks, ensuring they can harness the full potential of their data. The combination of VMware's robust management tools and Azure's extensive machine learning services fosters an environment where businesses can innovate rapidly and accelerate their digital transformation, while maintaining consistency across their entire estate, including on-premises.

This solution enhances operational efficiency by allowing organizations to utilize familiar VMware and Azure tools for managing both on-premises and cloud-based ML workloads. IT staff can leverage existing skills, minimizing the need for extensive retraining and accelerating the adoption of new ML technologies. As organizations increasingly seek to modernize their IT infrastructure and implement AI/ML capabilities, the ability to run Azure Machine Learning workloads on-premises through VMware Cloud Foundation becomes a critical differentiator in driving business innovation and competitiveness.

The collaboration between VMware and Microsoft empowers businesses to maximize the value of their data through advanced machine learning capabilities. By integrating Azure Machine Learning within a VMware environment using Kubernetes, organizations can drive innovation, enhance productivity, and gain valuable insights, ultimately leading to improved business outcomes and data-driven decision-making.

Audience

This VMware Validated Solution paper is intended for the following audiences:

- Enterprises and users who are using Azure cloud who also have a business or technical need to run AI services on-premises.
- Corporate CTOs and CIOs who are interested in learning more about how Broadcom and Microsoft can work together to bring Azure AI services on-premises.
- vSphere VI administrators who are familiar with VMware virtualized infrastructure and are looking to deploy and manage AI workloads in a private virtualized environment.
- Any other engineer/operator/end-user who are interested in AI workloads, Kubernetes, VMware Cloud Foundation and have a good understanding about VMware Cloud Foundation, vSAN, NSX, Cloud Native Storage (CNS), Container Storage Interface (CSI), Kubernetes, and AI.

Technology Overview

VMware Cloud Foundation

VMware Cloud Foundation is an integrated software stack that combines compute virtualization (VMware vSphere), storage virtualization (VMware vSAN), network virtualization (VMware NSX), and cloud management and monitoring (VMware Cloud Foundation Automation and VMware Cloud Foundation Operations, formerly known as VMware Aria Automation and VMware Aria Operations) into a single platform that can be deployed on-premises as a private cloud or run as a service within a public cloud.

This documentation focuses on the private cloud use case. VMware Cloud Foundation bridges the traditional administrative silos in data centers, merging compute, storage, network provisioning, and cloud management to facilitate end-to-end support for application deployment. See Getting Started with VMware Cloud Foundation for details.

Microsoft Azure

Microsoft Azure is a comprehensive cloud computing platform offering over 200 services across categories such as compute, storage, networking, databases, AI, and IoT. It provides Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software-as-a-Service (SaaS) solutions, enabling organizations to build, deploy, and manage applications and services through Microsoft-managed data centers. Azure's global infrastructure supports scalable deployment options with a pay-as-you-go pricing model, making it suitable for businesses of all sizes.

Microsoft Azure Arc

Microsoft Azure Arc extends Azure's capabilities beyond the cloud, enabling organizations to run Azure services on-premises, in other clouds, or at the edge—from a single interface. With Azure Arc, users can deploy Azure services on any infrastructure, including Kubernetes clusters on VMware Cloud Foundation, while leveraging GitOps principles for configuration management.

Microsoft Azure Machine Learning

Azure Machine Learning (Azure ML) is a comprehensive cloud service that accelerates the machine learning lifecycle, offering tools for model training, deployment, and management. Using Azure Arc to extend Azure services, Azure ML can be leveraged in hybrid environments using Kubernetes clusters, providing flexibility for organizations with specific infrastructure requirements or data governance needs. Through Azure Arc-enabled Kubernetes, Azure ML supports end-to-end ML lifecycle management on-premises, allowing for model training and deployment within your own data center. This approach is particularly valuable for enterprises with strict data residency requirements or those seeking to optimize existing on-premises compute resources for ML workloads.

Microsoft Azure Machine Learning Arc Extension

The Azure Machine Learning Arc extension enables enterprises to run Azure ML capabilities on-premises with Kubernetes clusters. It supports hybrid machine learning workflows, such as training in the cloud and deploying on-prem, while optimizing infrastructure, reducing latency, and maintaining control over resources. This extension facilitates efficient machine learning lifecycles by bringing computation closer to data.

Solution Configuration

A note on terminology: For the remainder of this paper, in order to be consistent with the official terminology used for VMware Cloud Foundation version 5.2.x (VCF 5.2.x), vSphere Kubernetes Service (VKS) and its associated components will be referred-to by their original names including Tanzu Kubernetes Grid Service (TKGs), Tanzu Kubernetes Cluster (TKC), and Tanzu Kubernetes release (TKr). There are no technical differences between TKGs and VKS in VCF 5.2.x.

This section introduces the following resources and configurations:

- Solution Architecture
  - VMware Cloud Foundation Standard Architecture
- Hardware resources
- Solution Deployment
  - vSAN File Service Deployment (Optional)
  - Tanzu Kubernetes Grid Service (TKGs) Deployment
  - Tanzu Kubernetes Cluster (TKC) Preparation
- Azure
  - Connect on-prem Kubernetes Cluster via Azure Arc
  - Deploy Azure Machine Learning Arc Extension
  - Attach Tanzu Kubernetes Cluster to Azure ML Workspace
  - Create and Manage Instances Type
- Run On-Premises Workload Test.

Solution Architecture

The VMware Cloud Foundation validation environment is composed of a standard architecture, including a management domain and one workload domain.

The Azure Machine Learning Arc extension will be deployed in the workload domain on top of Tanzu Kubernetes Grid Service Cluster. The TKGs cluster will connect to Azure Arc, and once it connects, the Azure Machine Learning Extension will be used to leverage on-prem compute resources.

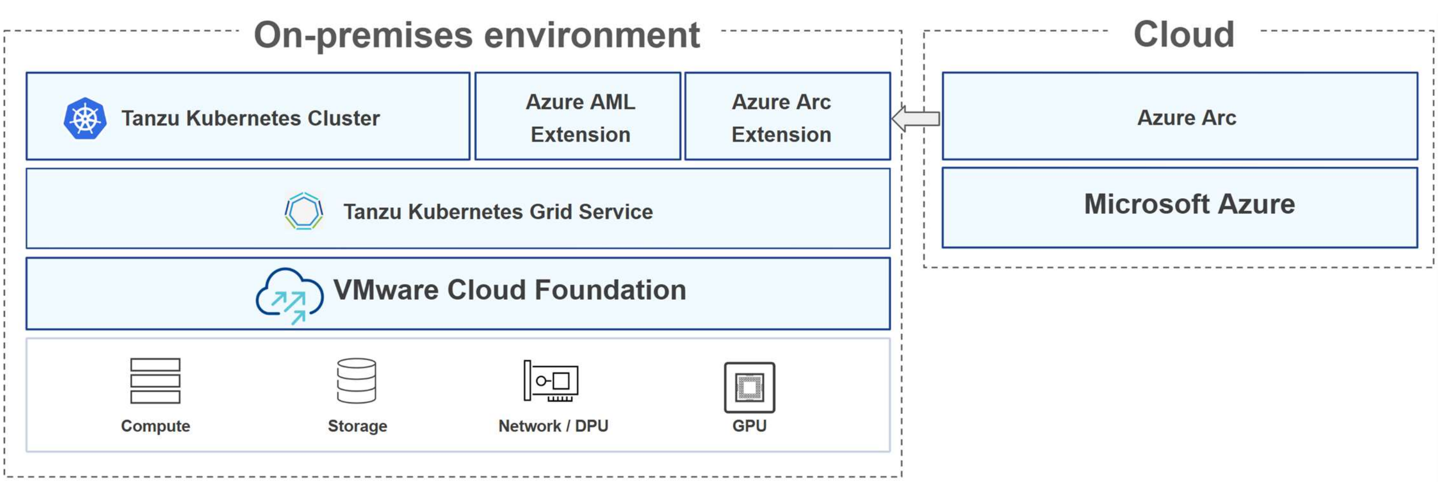

Figure 1: Architecture Diagram.

VMware Cloud Foundation Standard Architecture

For the standard architecture, a 4-node ESXi cluster was used for the VMware Cloud Foundation management domain, running management virtual machines and appliances.

Table 1. Management Domain VMs

VM Role	vCPU	Memory (GB)	VM Count
Management Domain vCenter Server	4	16	1
SDDC Manager	4	16	1
Management Domain NSX Manager	6	24	3
Workload Domain NSX Manager	12	48	3
Workload Domain vCenter Server	8	28	1

For the workload domain, we created another 4-node ESXi cluster with a separate NSX Fabric, deployed an NSX Edge Cluster, deployed vSAN File Service, and TKGs. To learn more about VMware Cloud Foundation domain architecture, click here.

Table 2. Workload Domain VMs

VM Role	vCPU	Memory (GB)	Storage (GB)	GPU	VM Count
Control Plane TKC	1	16	25	-	1
Worker Node TKC	64	256	500	1	1

Hardware Resources

In this solution, for the VMware Cloud Foundation workload domain cluster, a total of four Dell PowerEdge R750xa were deployed. Please consult our Compute Design for VMware Cloud Foundation for more information.

Each ESXi node in the cluster had the following configuration, as shown in Table 3.

Table 3. Hardware Configuration used for the Workload Cluster

VM Role	vCPU
Server model	Dell PowerEdge R750xa
CPU	2 x Intel(R) Xeon(R) Gold 6330 CPU @ 2.00GHz
Memory	512GB
Network Adapter	2 x Mellanox ConnectX-5 25GbE SFP

Storage Adapter	Dell H755
Disks	2 x Micron MTFDDAV240TDU 240GB M.2 2 x Dell Ent NVMe P5600 MU U.2 1.6TB
GPU	2 x NVIDIA H100 NVL PCIe 94GB

Recommended Hardware Requirements for AI

Hardware infrastructure requirements for AI workloads depend on the specific task, dataset size, model complexity, or performance expectations. Please consult VMware's documentation for more information.

Software Resources

This section describes software resources used in this solution.

Table 4. Software Information

Software	Version	Purpose
VMware Cloud Foundation	5.2.1	A unified Software Defined Data Center (SDDC) platform that brings together VMware vSphere, vSAN, NSX, Tanzu, and optionally, VMware Cloud Foundation Automation, into a natively integrated stack to deliver enterprise-ready cloud infrastructure for the private and public cloud. See VMware Cloud Foundation 5.2.1 Release Notes for details.
Tanzu Kubernetes releases (TKr)	v1.32.0---vmware.6-fips-vkr.2	Ubuntu 22.04 image used for the Kubernetes Cluster deployment.
AzureCLI	2.48.0	AzureCLI minimum required version to interact with Azure services.
Nvidia GPU Operator	24.6.2	Required for GPU functionality in Kubernetes Cluster. Version must match with Nvidia driver (vib) installed in ESXi.

Table 5. vSphere Kubernetes Cluster VM Configuration

VM Class	Configuration	Purpose
guaranteed-large (Default Class)	4 x vCPUs, 16GB RAM	• Control Plane node (1 node). Can add 3 nodes for redundancy if required.

h100-large (Custom Class)	64 x vCPUs, 256GB RAM, 500GB Storage, 1 x H100 94GB PCIe (in vGPU mode - C-Series)	• Ideal for larger deployments. • Custom VM classes must be provisioned prior to TKC deployment. Consult documentation for more information

Solution Deployment

VMware Cloud Foundation Installation

Follow the Official Document for detailed information about VMware Cloud Foundation installation steps. The key steps for VMware Cloud Foundation installation are as follows:

1. Deploy SDDC Manager and management workload domain through Cloud Builder.
2. Commission ESXi hosts to SDDC.
3. Create a workload domain with the idle ESXi hosts.
4. Deploy Edge Cluster (Active/Passive).
5. Deploy TKGs on the workload domain cluster.

Deploying VMware vSAN File Service (Optional)

Machine learning often requires NFS for shared data access across distributed systems, enabling scalable and efficient data management. VMware vSAN File Services enhances this by offering integrated, and scalable shared storage solutions, simplifying management and supporting AI/ML workloads with seamless scalability. Enabling vSAN File Service allows TKC to use ReadWriteMany (RWX) storage class. Follow the steps below to enable vSAN File Service and enable file volume support for the vSphere Kubernetes Cluster:

1. Enable File Service:
    a. Navigate to the vSAN cluster in the vSphere Client.
    b. Click on Configure > vSAN > Services.
    c. On the File Service row, click Enable.
2. Download the OVF File: Choose to automatically download the OVF file needed for deployment.
3. Configure Domain Settings:
    a. Specify a unique File Service domain name.
    b. Enter DNS server details (A and PTR records), subnet mask, and gateway information.
    c. Assign IP Addresses: Provide a pool of static IP addresses for the file servers, ensuring that the number of IPs matches the number of hosts in your cluster.
4. Finalize Configuration: Review all settings and click Finish to complete the setup.

It is also required to activate file volume support on the Supervisor Cluster. The steps for that are:

1. In the vSphere Client, navigate to Workload Management.
2. Click the Supervisors tab and select the Supervisor to edit from the list.
3. Click the Configure tab and click Storage.
4. Select File Volume.
5. Activate file volume support.

Deploying Tanzu Kubernetes Cluster

The first step in our integration is to deploy a Tanzu Kubernetes Cluster (TKC) on VMware Cloud Foundation. This cluster will serve as the foundation for our on-premises workloads and enable seamless connectivity with Azure services. The steps to deploy a TKC are:

1. Create Namespace within vSphere.
    a. Log into vSphere Client and navigate to the Workload Management section.
    b. Select the Supervisor cluster where you want to create the namespace.
    c. Launch the Create Namespace wizard:
        i. Choose a name for the namespace (must be DNS-compliant).
        ii. Select the workload network.
2. Configure namespace settings:
    a. Permission
        i. Add users/groups and assign roles (View, Edit, Owner).
    b. Storage
        i. Assign one or more vSphere storage policies.
    c. Resource Limits
        i. Set CPU, memory, and storage limits.
    d. VM Classes
        i. Select VM classes to make available (e.g. Small, Medium, Large)
    e. Associate the namespace with a TKr (Tanzu Kubernetes Release) content library.
3. Review settings and create the namespace.
4. Deploy Cluster.

There are multiple ways to deploy a TKC. If running the VMware Private AI Foundation with Nvidia add-on on VCF, it is possible to simply create a namespace and cluster using the VMware Cloud Foundation Automation. This is a great option if GPU workloads are being deployed as the automation will take care of GPU configuration. It is also possible to create the Cluster via vSphere client, if the Local Consumption Interface Supervisor Service is installed as well as CLI via “kubectl”.

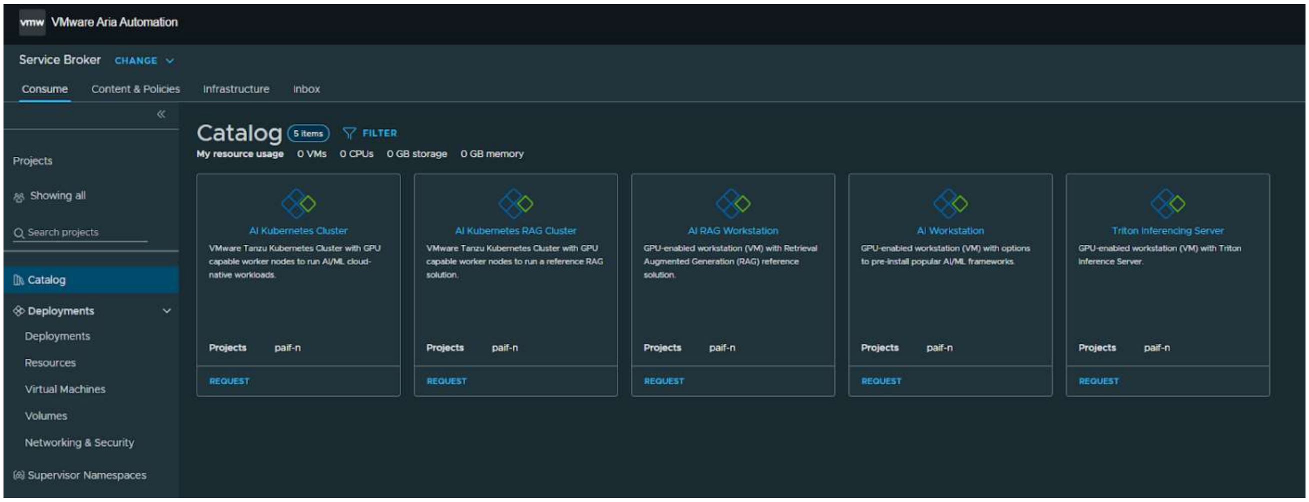

Figure 2: Deploying Cluster via VMware Cloud Foundation Automation Catalog

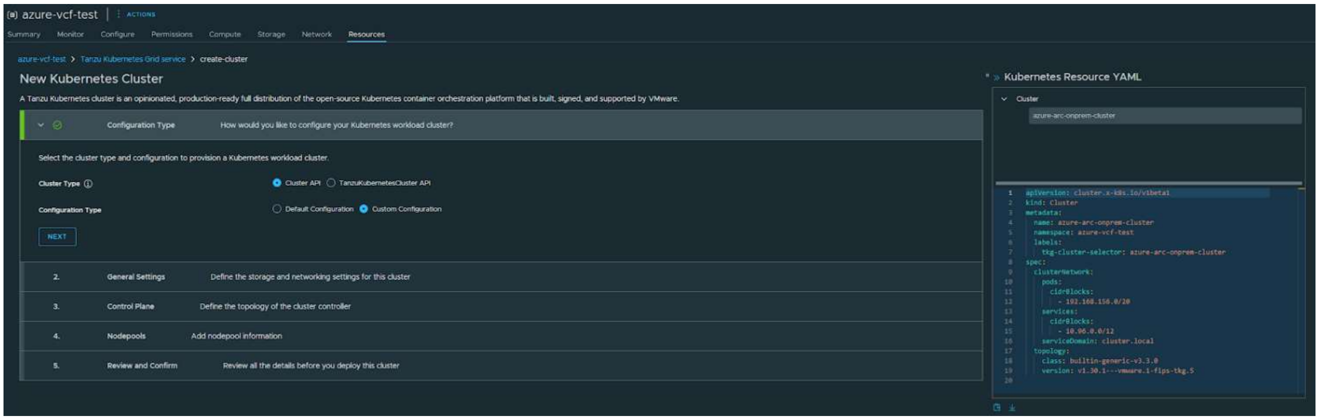

Figure 3: Deploying Cluster via vSphere Local Consumption Interface, Workload Management Section

Example - Deploying TKC via CLI (kubectl)

1. Create your cluster definition file (YAML). See example below:

```
apiVersion: cluster.x-k8s.io/v1beta1
kind: Cluster
metadata:
  name: azure-onprem-cluster
  namespace: azure-vcf-test
  labels:
    tkg-cluster-selector: azure-onprem-cluster
spec:
```

```yaml
clusterNetwork:
  pods:
    cidrBlocks:
      - 192.168.156.0/20
  services:
    cidrBlocks:
      - 10.96.0.0/12
  serviceDomain: cluster.local
topology:
  class: builtin-generic-v3.1.0
  version: v1.32.0---vmware.6-fips-vkr.2
  variables:
    - name: vmClass
      value: guaranteed-large
    - name: storageClass
      value: lamport-k8s-vsan
    - name: nodePoolVolumes
      value: []
  controlPlane:
    replicas: 1
    metadata:
      annotations:
        run.tanzu.vmware.com/resolve-os-image: os-name=ubuntu
  workers:
    machineDeployments:
      - class: node-pool
        name: gpu
        replicas: 1
        metadata:
          annotations:
            run.tanzu.vmware.com/resolve-os-image: os-name=ubuntu
        variables:
          overrides:
            - name: vmClass
              value: h100-large
            - name: nodePoolVolumes
              value:
                - name: containerd
                  mountPath: /var/lib/containerd
                  storageClass: lamport-k8s-vsan
```

```
            capacity:
              storage: 200Gi
- class: node-pool
  name: cpu
  replicas: 1
  metadata:
    annotations:
      run.tanzu.vmware.com/resolve-os-image: os-name=ubuntu
  variables:
    overrides:
      - name: vmClass
        value: guaranteed-4xlarge
      - name: nodePoolVolumes
        value:
          - name: containerd
            mountPath: /var/lib/containerd
            storageClass: lamport-k8s-vsan
            capacity:
              storage: 200Gi
```

2. Connect to the namespace created in the previous steps.

   ```
   kubectl vsphere login --server=x.x.x.x --tanzu-kubernetes-cluster-namespace namespace_name --vsphere-username user@domain --insecure-skip-tls-verify
   ```

3. Deploy the cluster. This process takes about 5-10 min.

   ```
   kubectl create -f deploy-azure-onprem-cluster.yaml
   ```

Prepare Kubernetes Cluster

Before connecting the Tanzu Kubernetes Cluster to Azure Machine Learning workspace, it is required to:

1. Connect to your newly created cluster. There are different ways to do that:
   a. Using vSphere Plugin

      ```
      kubectl vsphere login --server=x.x.x.x --tanzu-kubernetes-cluster-name cluster_name --tanzu-kubernetes-cluster-namespace namespace_name --vsphere-username user@domain --insecure-skip-tls-verify
      ```

   b. Downloading Cluster kubeconfig file from vSphere. This requires the Local Consumption Interface Supervisor service.

```
kubectl --kubeconfig kubeconfigfile -parameters
```

2. Provision the following namespaces within TKC with the proper labels.
    a. azure-arc
    b. azure-arc-release
    c. azureml
    d. arc-compute-on-prem

```
#Provision required namespaces
kubectl create namespace azure-arc
kubectl create namespace azure-arc-release
kubectl create namespace azureml
kubectl create namespace arc-compute-on-prem

#Apply necessary labels and annotations to the namespaces
kubectl label namespace azure-arc app.kubernetes.io/managed-by=Helm
kubernetes.io/metadata.name=azure-arc pod-security.kubernetes.io/enforce=privileged

kubectl annotate namespaces azure-arc meta.helm.sh/release-name=azure-arc meta.helm.sh/release-
namespace=azure-arc-release

kubectl label namespace azure-arc-release kubernetes.io/metadata.name=azure-arc-release
name=azure-arc-release pod-security.kubernetes.io/enforce=privileged

kubectl label namespace azureml kubernetes.io/metadata.name=azureml pod-
security.kubernetes.io/enforce=privileged
kubectl label namespace arc-compute-on-prem pod-security.kubernetes.io/enforce=privileged
```

3. Configure Nvidia GPU Operator (Optional, required only if leveraging NVIDIA GPUs)
    a. VMware Cloud Foundation Automation AI Kubernetes Cluster. GPU operator will be automatically deployed in this configuration. Requires VMware Private AI Foundation with Nvidia add-on.
    b. Deploying NVIDIA GPU Operator manually.
    c. Deploying NVIDIA GPU Operator via script.

Connecting On-Prem Kubernetes Cluster in Azure Arc

Follow Microsoft's Quickstart guide to connect your on-premises Tanzu Kubernetes Cluster (TKC) to Azure Arc.

Deploying Azure Machine Learning Extension

To enable machine learning capabilities on your Tanzu Kubernetes Cluster, it is necessary to deploy the Azure Machine Learning extension. This extension allows for the execution of training jobs and inference workloads. The deployment and lifecycle management of this extension can be efficiently handled using the Azure CLI k8s-extension command. The details can be found here.

Example:

az k8s-extension create --name arc-compute-on-prem --extension-type Microsoft.AzureML.Kubernetes --config enableTraining=True enableInference=True inferenceRouterServiceType=NodePort allowInsecureConnections=True --cluster-type connectedClusters --cluster-name azure-arc-onprem-cluster --resource-group my-rg --scope cluster

Please consult Azure Machine Learning Extension Documentation for more details regarding the implementation.

Attaching Tanzu Kubernetes Cluster to Azure ML Workspace

An Azure Machine Learning workspace is a fundamental resource that serves as the central hub for machine learning activities in Azure. It provides a collaborative environment for data scientists and machine learning engineers to create, manage, and organize their machine learning artifacts and related work.

To create an Azure Machine Learning workspace, users can utilize various methods, including the Azure Machine Learning studio, Azure portal, Visual Studio Code extension, Azure CLI, or programmatically using the Azure Machine Learning SDK for Python.

Learn more about how to attach your TKC here.

Create and Manage Instances Type

Instance types in Azure Machine Learning are a concept that enables targeting specific compute node types for training and inference workloads. In Kubernetes clusters, these are implemented through a custom resource definition (CRD) installed with the Azure Machine Learning extension.

Example of Instance Type Definition (my-instance.yaml):

```
apiVersion: amlarc.azureml.com/v1alpha1
kind: InstanceType
metadata:
  name: gpu-large
spec:
  nodeSelector:
    mylabel: mylabelvalue
  resources:
    limits:
      cpu: "64"
      nvidia.com/gpu: 1
      memory: "224Gi"
```

```
requests:
  cpu: "16"
  memory: "64Gi"
```

Applying the definition:

- Connect to the vSphere Kubernetes Cluster
- Apply the definition – i.e. kubectl apply -f my-instance.yaml

Consult Microsoft Documentation for more information on how to configure instance types.

Run On-Premises Workload Test

This section illustrates how to run a Machine Learning image classification training job via Azure AI Machine Learning Studio. It uses a logistic regression model with regularization rate of 0.5 based on an AML Kubernetes example, described here, but adapted to run on AML Studio instead of Azure CLI. Follow the steps below to deploy the example:

1. Pull the required files from Microsoft AML Kubernetes Github Repository.
2. Login to AML Studio.
3. Select Jobs on the left side menu.
4. Click on Create job.
5. Select the option “Run a custom training script”, then click on “Start configuring job”.
6. Provide a job name and description. Leave the “Existing experiment” option set to Default, then hit Next.
7. Upload the “src” folder (make sure you pulled the example from Git).
8. Add command “python train.py --data-folder ./mnist-data --regularization 0.5” to the Command box then hit next.

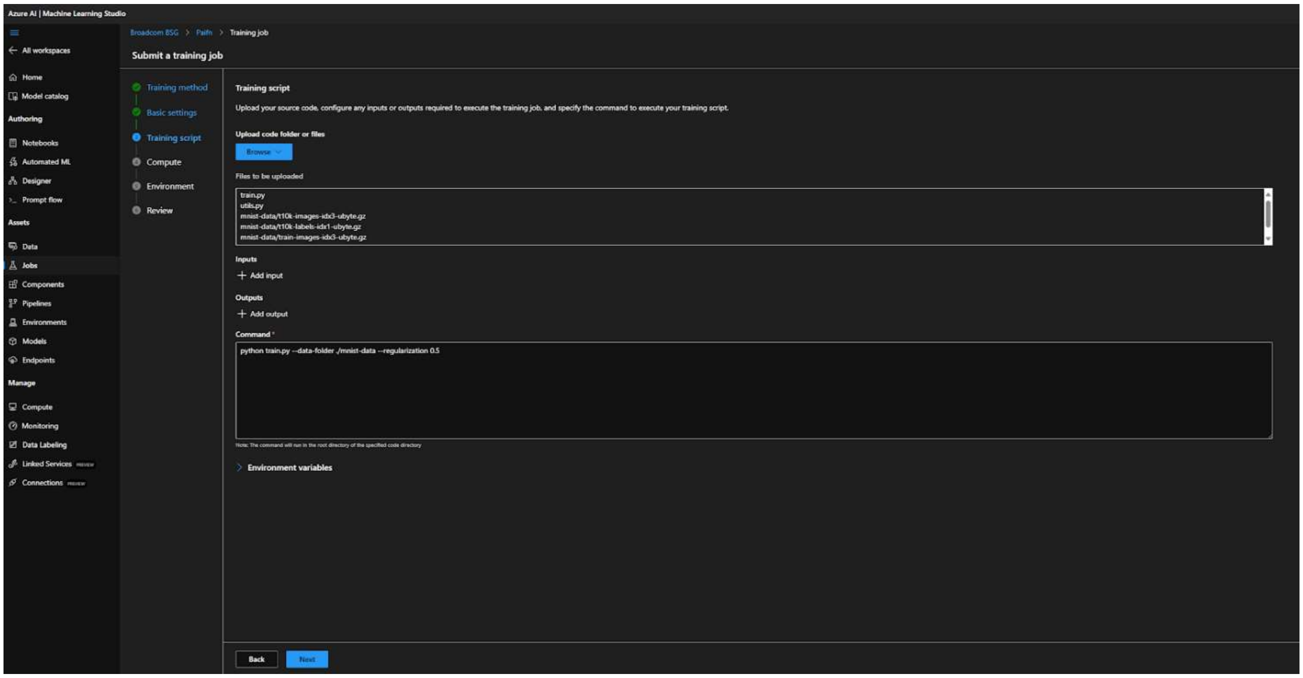

Figure 4: AML Studio - Submitting a training job, uploading required files and adding python command

9. Select Compute type “Kubernetes compute”.
10. Select your on-prem Tanzu Kubernetes Cluster under ML Kubernetes compute.
11. Select the instance type you created previously during the solution deployment, then hit Next.

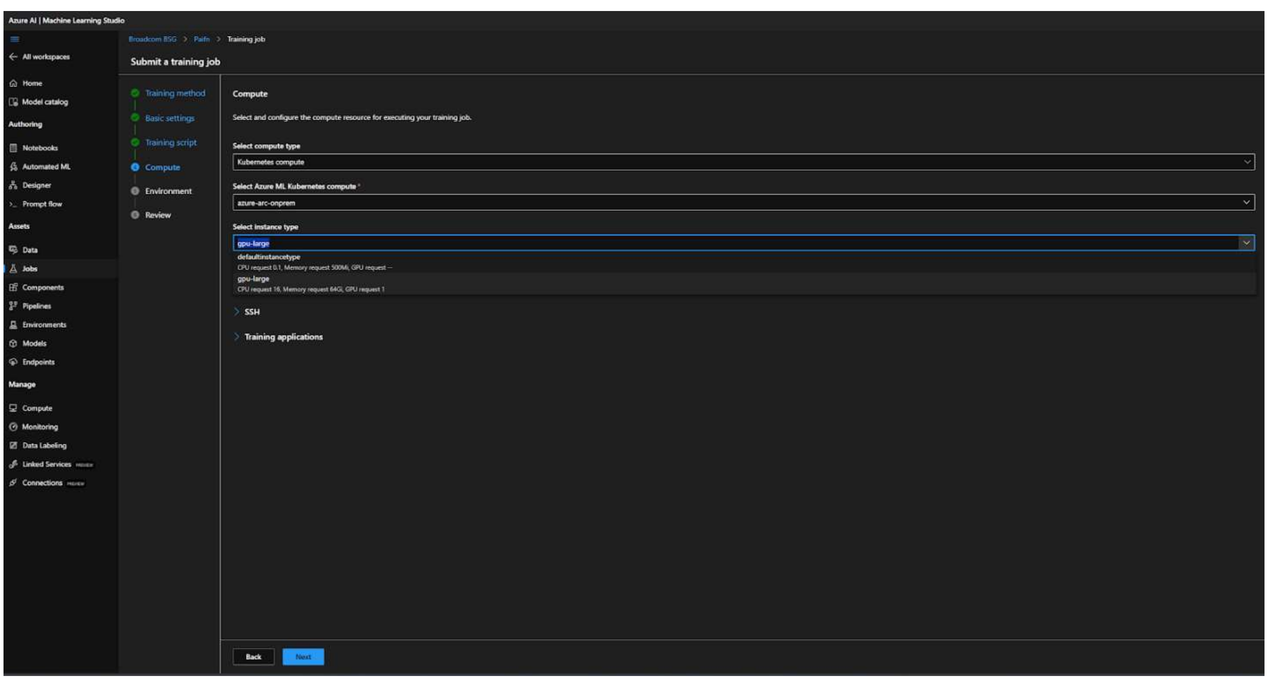

Figure 5: AML Studio – Submitting job, compute selection and instance type screen

12. Under Environment, select Curated environments, then search for “sklearn-1.5”, then hit Next.

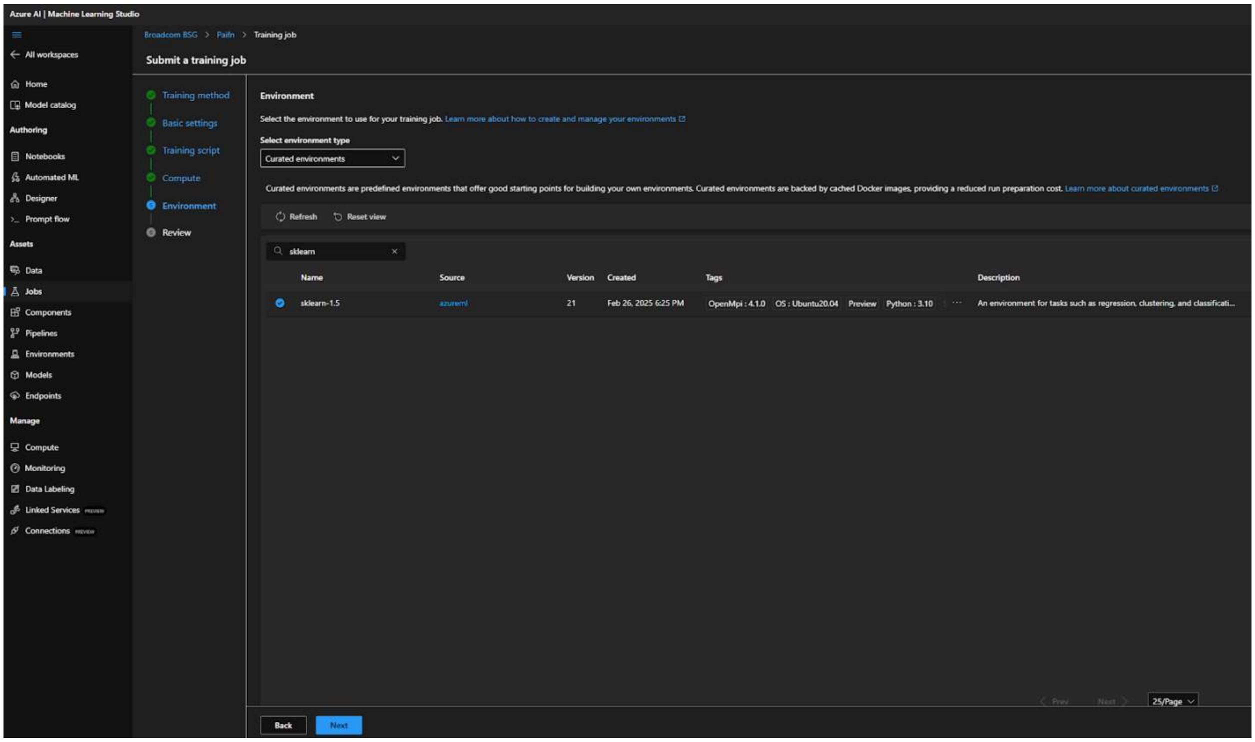

Figure 6: AML Studio – Submitting job, selecting curated environment type sklearn-1.5

13. Review your settings then click on Submit training job.

14. Select the submitted job to check for more details.

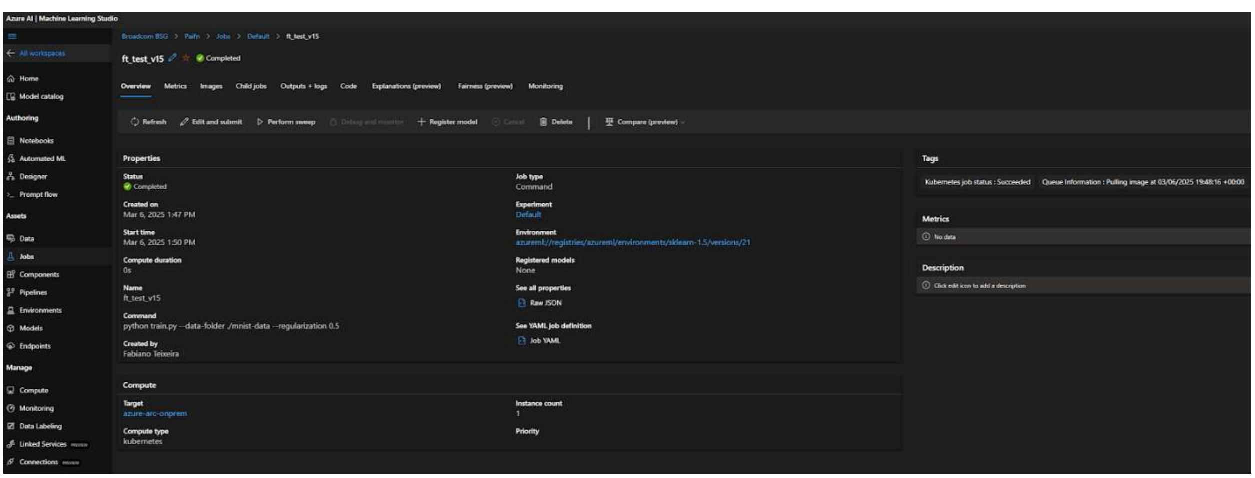

Figure 7: AML Studio – Job Overview

15. Select the Output + Logs tab to verify the results of the training job.

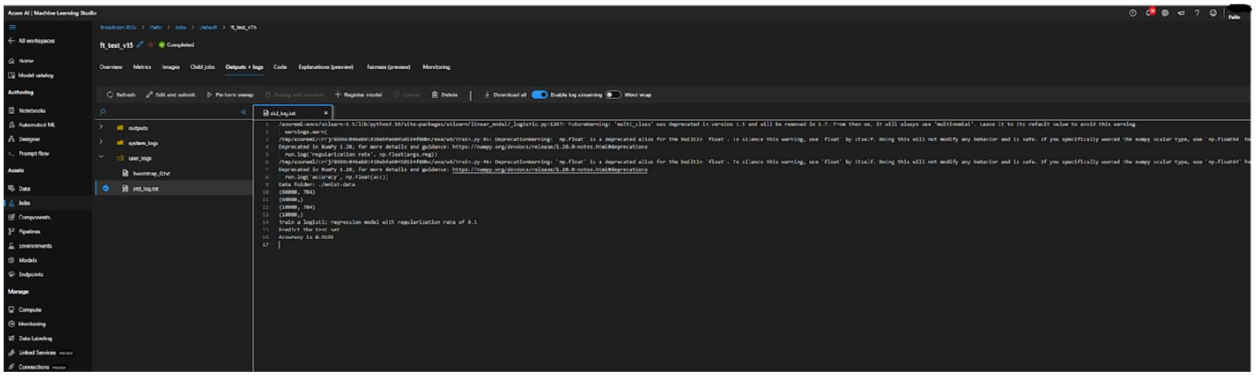

Figure 8: AML Studio – Job Overview, Output + Log

Another great example you can try out is to deploy an image classification model on-prem for inference. The implementation in this example is done through Azure CLI. More information detailed here.

This section demonstrated how to leverage your on-prem VCF environment to run Azure Machine Learning jobs.

Conclusion

The integration of VMware Cloud Foundation (VCF) and Azure Machine Learning (AML), enabled by Azure Arc and VMware vSphere Kubernetes Service (VKS), formerly known as Tanzu Kubernetes Grid Service (TKGs), represents a great advancement on-premises machine learning implementation. This solution addresses the critical needs of modern enterprises by providing a robust platform for running advanced Azure ML workloads on-premises within the flexible and scalable VMware infrastructure.

By leveraging this integrated solution, organizations can leverage the power of AI-driven analytics and machine learning while maintaining control over their data and compute resources. This approach not only enhances operational efficiency but also drives innovation and cost-effectiveness. As businesses continue to navigate the complexities of digital transformation and AI adoption, this VMware-Microsoft collaboration offers a clear path forward, enabling them to stay competitive in an increasingly data-driven world.

Ultimately, this integration empowers organizations to unlock the full potential of their data, drive deeper insights, and make more informed decisions—all while operating within a flexible and familiar IT environment. As we look to the future, solutions like this will contribute to shaping enterprise IT and ML strategies, offering organizations more options to seamlessly bridge on-premises and cloud environments in their pursuit of data-driven excellence.

References

- VMware Cloud Foundation | Private Cloud Platform
- VCF AI/ML Solutions | VMware
- Azure Arc – Hybrid and Multi-Cloud Management and Solution
- Microsoft Azure Machine Learning
- Azure Machine Learning Extension

About the Author

- Fabiano Teixeira, Solutions Architect, VMware by Broadcom

Item No: vmw-bc-wp-tech-temp-uslet-word-2024 1/24