TECHNICAL WHITE PAPER: **October 2026**



H2O AI on vSphere Kubernetes Service on VMware Cloud Foundation

Reference Architecture

Table of Contents

Executive Summary 3
vSphere Kubernetes Service 3
H2O AI: The H2O Enterprise AI Platform 4
Solution Architecture 7
VKS Cluster Configuration 9
Kubernetes Packages 11
H2O.AI Charts 11
Solution Validation 11
Install GPU Operator 11
Install H2O AI Cloud on VKS 12
Validate H2O AI Cloud on VKS 15
Conclusion 26

Executive Summary

Enterprises adopting artificial intelligence face a common constraint: the data that makes AI valuable is frequently the data that cannot leave the organization's own infrastructure. Regulatory obligations, data residency requirements, intellectual property protection and cost predictability all push enterprises toward running AI workloads on infrastructure they own and control, rather than on a public cloud AI service.

This technical white paper documents the deployment and validation of the H2O Enterprise AI platform (Agentic AI, Generative AI, Predictive AI) on vSphere Kubernetes Service (VKS) — the CNCF-certified Kubernetes runtime built into VMware Cloud Foundation (VCF) from Broadcom. It is published as part of an Independent Software Vendor (ISV) partnership between H2O.ai and the VMware Cloud Foundation Division of Broadcom, under which H2O Enterprise AI platform is deployed, tested and supported as a validated technology stack on VKS.

The partnership addresses a requirement that both companies see in their shared customer base. Organizations already standardized on VCF for their private cloud platform want to run agentic, predictive and generative AI on that same platform — using the same operational model, the same skills, the same lifecycle management and the same security boundary as the rest of their estate — rather than building and operating a separate AI stack alongside it. Running H2O Enterprise AI platform on VKS allows an enterprise to keep its data, its models and its GPU investment inside its own data center while still offering data scientists and application developers a self-service, end-to-end AI platform.

This paper provides: an architectural overview of H2O Enterprise AI and how its services map onto a VKS cluster running on VCF, including NVIDIA GPU acceleration delivered through vGPU or passthrough; a reproducible Helm-based deployment procedure for the NVIDIA GPU Operator and the H2O Enterprise AI platform; a validation methodology expressed as a sequence of pass/fail gates that proves the platform is operational end to end, from cluster readiness through GPU-accelerated model training, model deployment and scoring, and generative AI inference; and the configuration of the validated stack, so that the environment can be reproduced.

The intended audience is platform engineers, cloud administrators, MLOps engineers and solution architects responsible for delivering AI capability on VMware Cloud Foundation. Familiarity with Kubernetes, vSphere and basic machine learning operations concepts is assumed.

vSphere Kubernetes Service

vSphere Kubernetes Service (VKS) is the Kubernetes runtime built directly into VMware Cloud Foundation (VCF). With CNCF-certified Kubernetes, VKS enables platform engineers to deploy and manage Kubernetes clusters while leveraging a comprehensive set of cloud services in VCF. Cloud admins benefit from the support for N-2 Kubernetes versions, enterprise-grade security, and simplified lifecycle management for modern apps adoption.

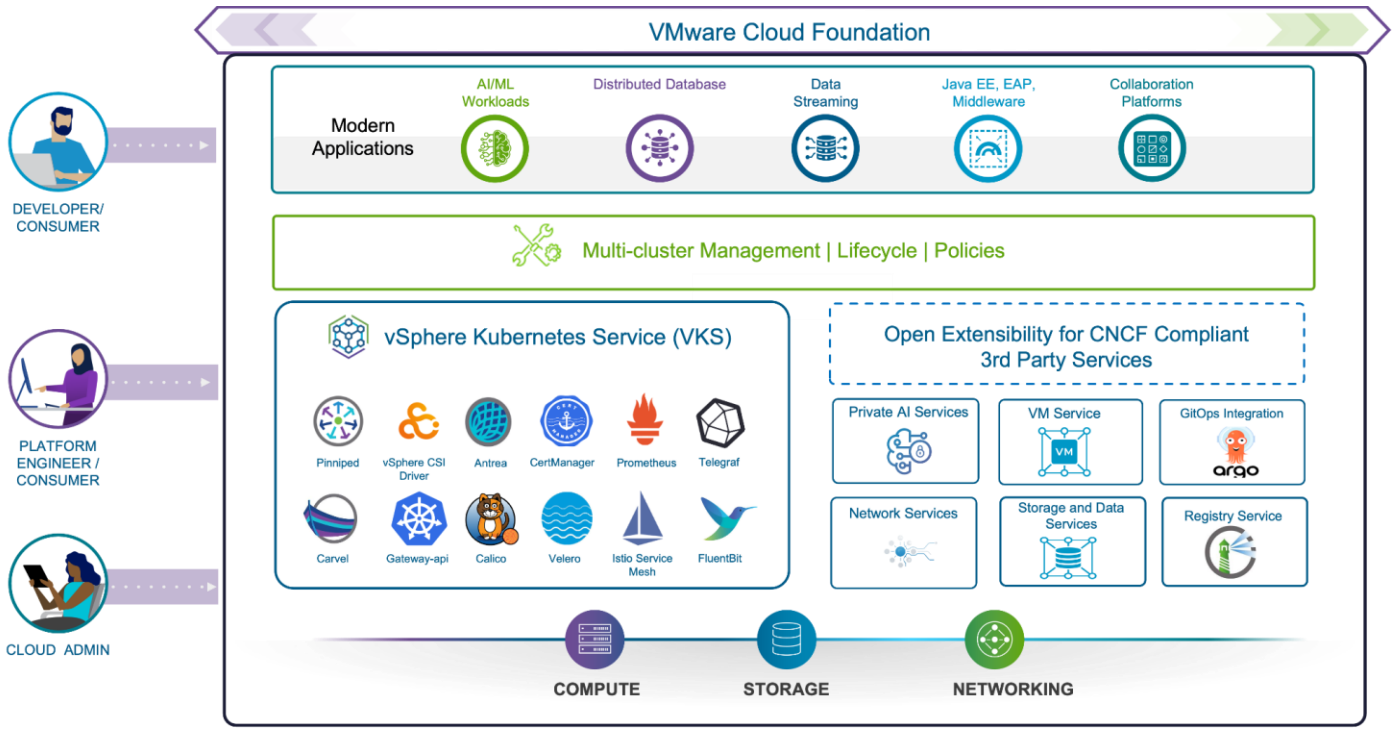


Figure 1: VKS on VCF Ecosystem

Benefits of using VKS with modern applications:

Lower TCO: With VKS organizations can reduce silos, leverage existing tools and skill sets without having to retrain staff and/or change existing processes. Utilizing unified lifecycle management across infrastructure components to stay up to date with the most recent patches and minimizing security risks.

Operational Simplicity: VKS is engineered for unparalleled operational simplicity, leveraging the familiarity of existing vSphere tools, skills, and workflows. This design philosophy significantly reduces the learning curve for IT teams and streamlines management processes. With VKS, organizations benefit from automated cluster provisioning, which accelerates deployment times and minimizes manual configuration errors. Furthermore, its robust capabilities extend to automated upgrades and comprehensive lifecycle management. This integrated approach ensures consistent operations, reduces overhead, and frees up valuable resources to focus on innovation rather than infrastructure maintenance.

Run and Manage Kubernetes at Scale: Effortlessly deploy and manage Kubernetes clusters at scale, leveraging a built-in, Cloud Native Computing Foundation (CNCF) certified Kubernetes distribution. VKS provides fully automated lifecycle management, streamlining operations from initial setup to ongoing maintenance and upgrades. This comprehensive approach ensures that organizations can harness the power of Kubernetes for their containerized applications with unparalleled efficiency and reliability, without the complexities typically associated with large-scale Kubernetes deployments.

H2O AI: The H2O Enterprise AI Platform

H2O Enterprise AI is H2O.ai's end-to-end enterprise AI platform. It provides a single, governed environment in which an organization can prepare data, build and train predictive and generative AI models, deploy and monitor those models in production, and publish the resulting capability to business users as AI applications. The platform is deployed on Kubernetes and is designed to run on public cloud, private cloud or on-premises infrastructure without change to its architecture — which is what makes it a natural fit for VKS on VCF.

H2O.ai also delivers the platform as a packaged on-premises configuration sized for a defined hardware footprint. The validated stack described in this paper is the reference configuration for delivering that experience on VMware Cloud Foundation, where VKS provides the Kubernetes substrate and vSphere provides the GPU, storage and network abstraction.

Platform components. A default H2O Enterprise AI installation includes the AI App Store, H2O AI Engine Manager, H2O Driverless AI, H2O-3, the H2O Wave SDK, H2O MLOps, H2O Workflows and the H2O Platform Services (AuthZ, Cloud Discovery, Token Server, Secure Store, Model Hub, Telemetry and the H2O Drive backend service). H2O Drive, H2O LLM Studio and H2O Admin Analytics are included as optional components. Further subscription components can be installed into the same environment, including Enterprise h2oGPTe, H2O Document AI, H2O Hydrogen Torch, H2O Feature Store, H2O eScorer, H2O Label Genie, H2O LLM Data Studio and H2O Eval Studio.

Workspaces and self-service. Users work inside Workspaces, the platform's mechanism for resource sharing and collaboration. From a Workspace, a data scientist can start an AI Engine — a Driverless AI, H2O-3 or Notebook engine — sized with a specific CPU, memory, GPU and disk allocation. AI Engine Manager owns the lifecycle of those engines, creating, pausing, resuming and deleting the underlying Kubernetes workloads on demand. This is the mechanism that makes GPU capacity in the VKS cluster consumable as a self-service resource rather than a statically assigned one, and it is the reason GPU utilization on a shared VKS cluster can be kept high.

Predictive AI. Driverless AI and H2O-3 provide automated machine learning: automated feature engineering, model selection, validation and tuning. Trained models are promoted into H2O MLOps, which handles model registration, deployment as a scoring endpoint, and ongoing monitoring of production performance and drift.

Generative AI. Enterprise h2oGPTe provides a secure, self-hosted generative AI and retrieval-augmented generation (RAG) capability, allowing users to query and extract insight from collections of private documents without that content leaving the environment. H2O LLM Studio supports fine-tuning of open large language models, and H2O Eval Studio provides evaluation and testing for RAG and LLM behavior. These components have the strongest dependency on GPU capacity, and therefore most directly justify the GPU-enabled VKS node pool described later in this paper.

Applications. The AI App Store is the delivery surface for the platform. Applications built with the open-source H2O Wave Python framework are published to an organization's internal App Store, where business users can run them against the models produced upstream. The App Store deploys each application instance as its own Kubernetes deployment; individual app instances are exposed on subdomains of the platform's base domain.

The end-to-end workflow, from data ingestion through to a governed AI application consumed by a business user, is shown in Figure 2.

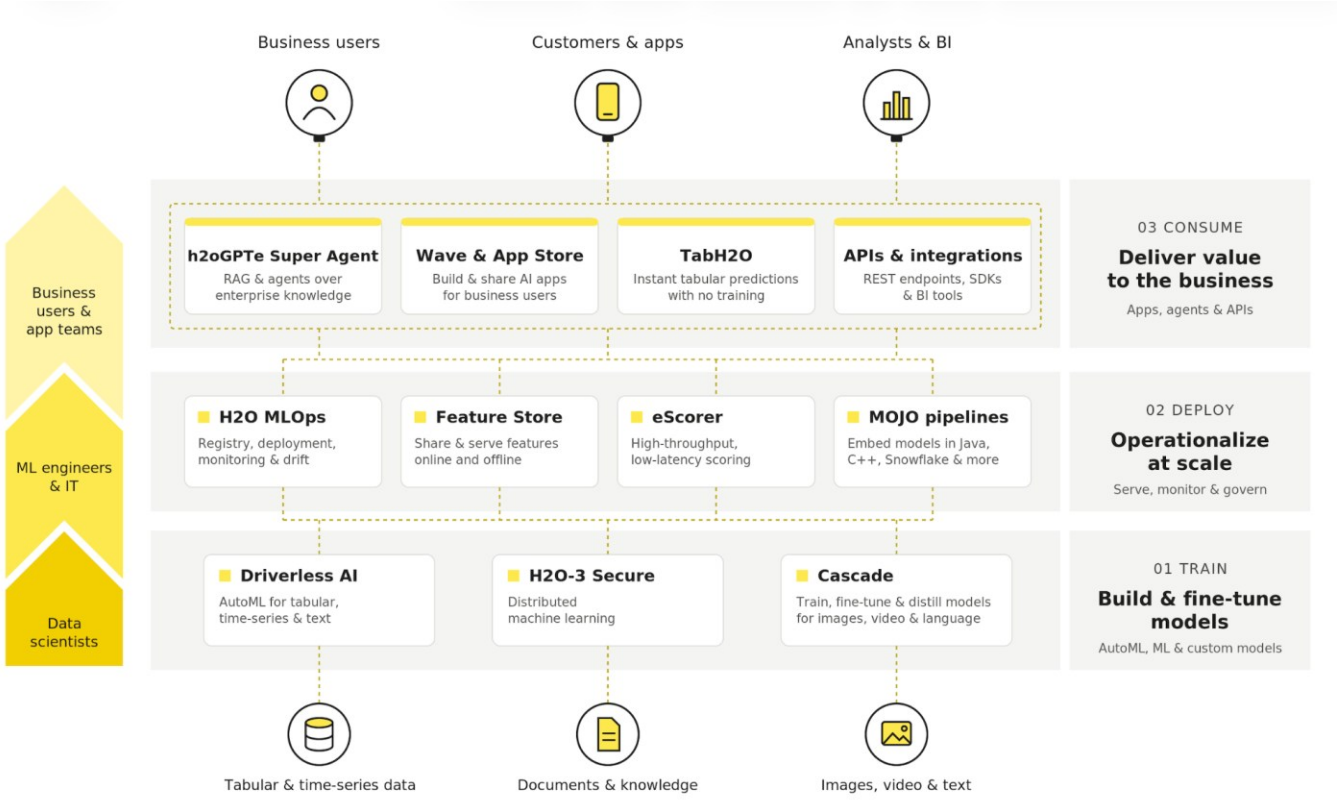


Figure 2: H2O AI Cloud Platform Workflow

Solution Architecture

Figure 3 shows the overall solution architecture, with H2O AI Cloud deployed on a VKS cluster running on VMware Cloud Foundation.

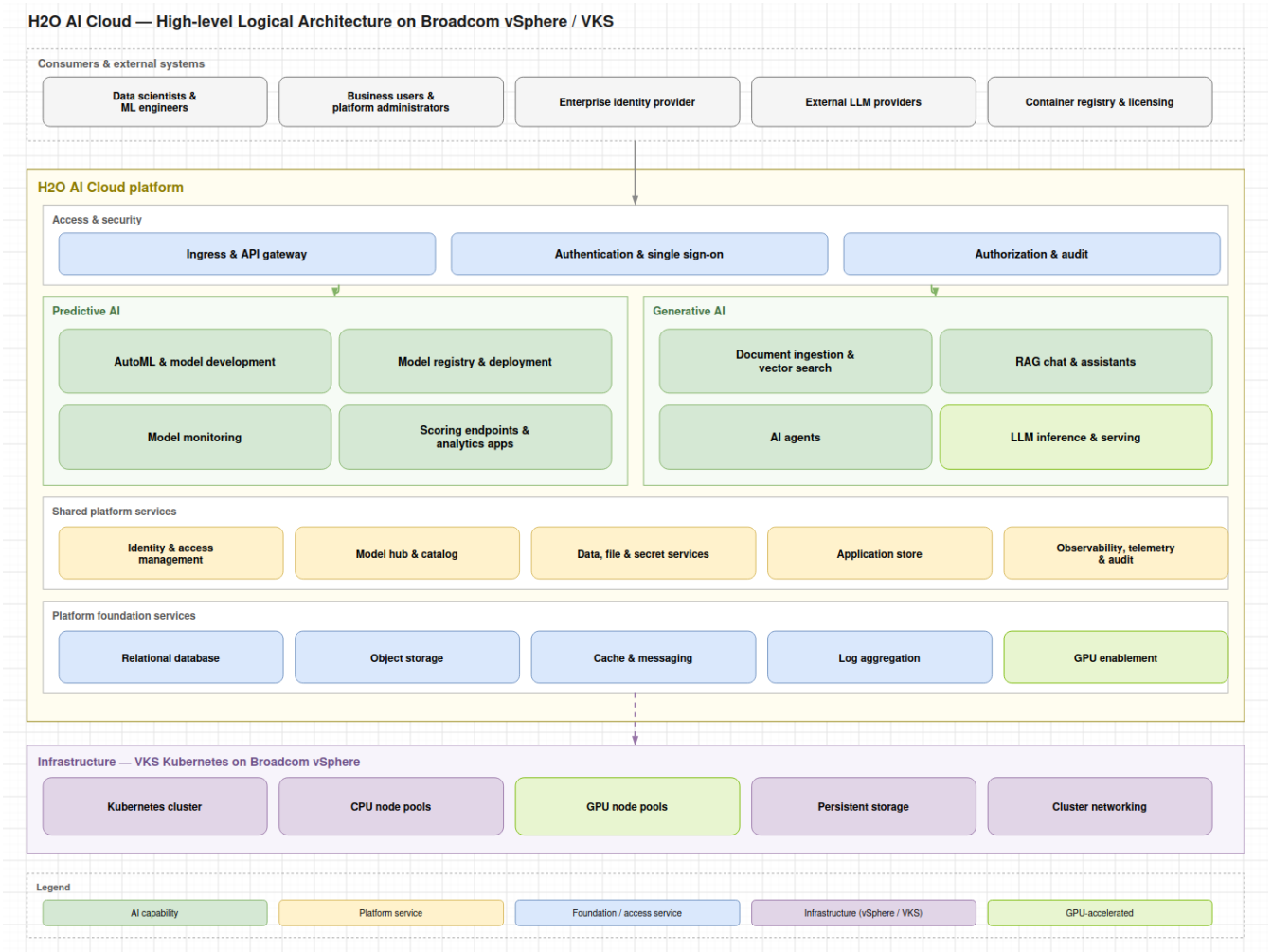


Figure 3: H2O AI Cloud on vSphere Kubernetes Service

At its core, the Supervisor control plane provides essential services, such as the Harbor container registry, the Contour ingress controller, and the Argo CD GitOps utility. For more information on vSphere Supervisor services and installation, visit: https://techdocs.broadcom.com/us/en/vmware-cis/vcf/vsphere-supervisor-services-and-standalone-components/latest.html

This control plane is responsible for the automated deployment and lifecycle management of CNCF-compliant Kubernetes clusters via the vSphere Kubernetes Service (VKS) through Cluster API. The specific version of VKS used in this white paper is 3.6.2.

Platform service placement. The H2O Enterprise AI control plane services — the AI App Store server, AI Engine Manager, MLOps services, AuthZ, Token Server, Secure Store, Model Hub, Cloud Discovery and Telemetry — are long-running services with modest and predictable resource requirements and are scheduled onto the CPU worker node pool. The GPU-dependent workloads — Driverless AI and H2O-3 engines configured with GPUs, and LLM inference and fine-tuning workloads for Enterprise h2oGPTe and H2O LLM Studio — are scheduled onto the GPU worker node pool through node affinity and tolerations. H2O MLOps supports node affinity and toleration shortcuts that are defined at install time, which allows a model to be deployed onto GPU-enabled nodes explicitly rather than by chance.

Ingress and DNS. External access is provided by the Contour ingress controller installed as a Supervisor service. H2O Enterprise AI requires a wildcard DNS record and a matching wildcard TLS certificate for the platform base domain, because each AI application instance launched

from the App Store is exposed on its own subdomain. The cert-manager package listed in Table 3 issues and renews the certificates used by the platform ingress.

Identity. H2O AI Cloud authenticates users against an OpenID Connect (OIDC) provider. In this configuration the platform is integrated with the Keycloak instance deployed by the H2O Helm chart, or the customer's existing enterprise identity provider. Kubernetes-level authentication for platform administrators is provided by Pinniped, which is included as a VKS core package.

Figure 4 shows the H2O AI Cloud realm in the Keycloak administration console, with the platform users provisioned for this validation.

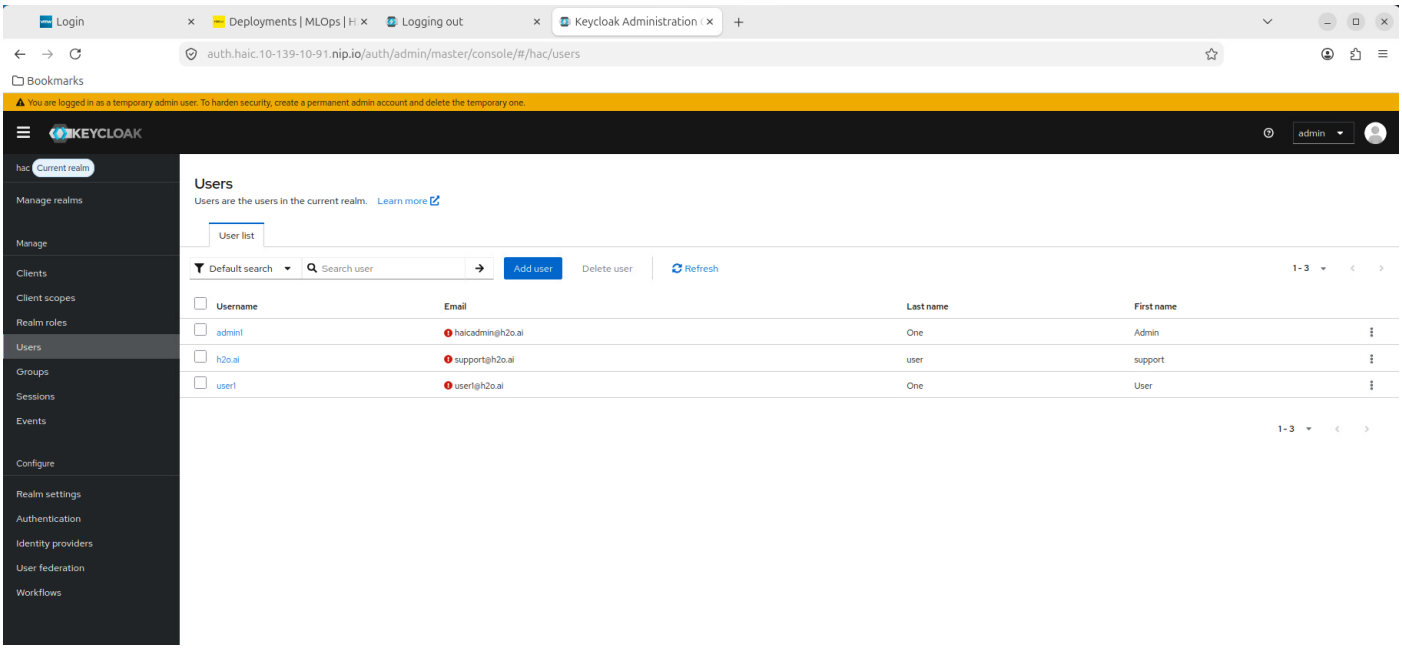


Figure 4: Keycloak Administration Console – H2O AI Cloud Realm Users

Container registry. H2O Enterprise AI container images are pulled from the H2O.ai registry. In connected environments the images are pulled directly. In air-gapped or restricted environments the images are mirrored into the local enterprise container registry, and the Helm values are overridden to reference the local mirror.

Storage. Persistent storage is provided by vSAN Express Storage Architecture (ESA) and surfaced to Kubernetes by the vSphere CSI driver through the vsan-esa-default-policy-raid5 storage class. Three distinct storage requirements exist. First, block storage through PersistentVolumeClaims for AI engine working directories, model artifacts and component state. Second, a PostgreSQL database, version 11.0 or later, for App Store and platform metadata. Third, S3-compatible object storage for large objects such as application bundles, datasets and model artifacts.

GPU acceleration. GPU capacity is presented to the VKS cluster by way of either NVIDIA vGPU or GPU passthrough, configured on the ESX hosts and surfaced to Kubernetes nodes through a GPU-enabled VM Class. The NVIDIA GPU Operator, installed by Helm, manages the driver, container toolkit and device plugin on the GPU node pool, causing those nodes to advertise the nvidia.com/gpu extended resource. H2O AI Engine Manager then requests that resource when a user creates a GPU-backed engine.

Table 1: Installed Supervisor Components

Component	Version	Notes
vSphere Kubernetes Service	3.6.2	Allows for CNCF K8s clusters

VKS Cluster Configuration

To prepare the VKS cluster with AI capabilities powered by GPU, there are two ways to equip GPU resources with VKS, passthrough GPU or vGPU.

GPU Preparation on ESX Hosts

You can follow this guide to configure either passthrough GPU or vGPU on ESX hosts: https://techdocs.broadcom.com/us/en/vmware-cis/private-ai/foundation-with-nvidia/9-0/private-ai-foundation-9-x/deploying-private-ai-foundation-with-nvidia/configure-nvidia-vgpu-or-gpu-passthrough-on-the-esx-hosts.html

VM Class Preparation for GPU Resources

Once passthrough GPU or vGPU is configured on the ESX hosts, you can follow this guide to create VM Classes that are configured with GPU resources: https://techdocs.broadcom.com/us/en/vmware-cis/private-ai/foundation-with-nvidia/9-0/private-ai-foundation-9-x/deploying-private-ai-foundation-with-nvidia/configure-vgpu-based-vm-classes-for-ai-workloads.html

The cluster consisted of one control plane node and three worker nodes. Storage was provided by vSAN Express Storage Architecture (ESA), using the default RAID 5 storage policy. Transparently, the vSphere CSI driver exposed this policy to an adjacent Kubernetes storage class, **vsan-esa-default-policy-raid5**.

Worker node resources were configured with two different VM Classes, with vGPU and without vGPU, while the control plane node used the guaranteed large profile.

Table 2: Kubernetes Cluster Configuration Summary

Component	Configuration	Notes
Kubernetes Control Plane	1 Replica — 4x vCPU 16GB RAM vSAN RAID 5	VM Class: “guaranteed large” Storage Class: “vsan-esa-default-policy-raid5”
Kubernetes Worker Nodes with GPU	1 Replica — 96x vCPU 384GB RAM vSAN RAID 5 2x NVIDIA RTX Pro 6000 vGPU 96c pciPassthru0.cfg.enable_uvm=1 pciPassthru1.cfg.enable_uvm=1	User Created VM Class with GPU: vks-96-384-vgpu-96c Storage Class: “vsan-esa-default-policy-raid5”
Kubernetes Worker Nodes without GPU	2 Replicas — 96x vCPU 384GB RAM vSAN RAID 5	User Created VM Class without GPU: vks-96-384 Storage Class: “vsan-esa-default-policy-raid5”
Kubernetes Release	v1.35.5	
OS Image	Ubuntu 24.04.4 LTS	Kernel version 6.17.0-29-generic

The VKS cluster configuration is defined declaratively in the YAML file below. As previously outlined, applying this manifest to the Supervisor triggers the creation of the cluster, which is then managed and reconciled by the Cluster API.

```
# vks-cluster.yaml

apiVersion: cluster.x-k8s.io/v1beta1
kind: Cluster
metadata:
  name: pais-vks
  namespace: supervisor-namespace
spec:
  clusterNetwork:
    pods:
      cidrBlocks:
        - 192.168.156.0/20
    services:
      cidrBlocks:
        - 10.96.0.0/12
    serviceDomain: cluster.local
  topology:
    class: builtin-generic-v3.6.0
    version: v1.35.5---vmware.1-vkr.1
    variables:
      - name: kubernetes
        value:
          certificateRotation:
            enabled: true
            renewalDaysBeforeExpiry: 90
      - name: vmClass
        value: guaranteed-large
      - name: storageClass
        value: vsan-esa-default-policy-raid5
    controlPlane:
      replicas: 1
      metadata:
        annotations:
          run.tanzu.vmware.com/resolve-os-image: os-name=ubuntu,os-version=24.04
    workers:
      machineDeployments:
        - class: node-pool
          name: gpu-pool
          replicas: 1
          variables:
            overrides:
            - name: vmClass
              value: vks-96-384-vgpu-96c
          metadata:
            annotations:
              run.tanzu.vmware.com/resolve-os-image: os-name=ubuntu,os-version=24.04
        - class: node-pool
          name: no-gpu-pool
          replicas: 2
          variables:
            overrides:
            - name: vmClass
              value: vks-96-384
          metadata:
            annotations:
              run.tanzu.vmware.com/resolve-os-image: os-name=ubuntu,os-version=24.04
```

```
  variables:
    - name: volumes
      value:
        - name: containerd
          mountPath: /var/lib/containerd
          storageClass: vsan-esa-default-policy-raid5
          capacity: 1024Gi
        - name: kubelet
          mountPath: /var/lib/kubelet
          storageClass: vsan-esa-default-policy-raid5
          capacity: 1024Gi
```

Kubernetes Packages

VKS clusters (via the VKr release) include a default set of core packages, such as Antrea for networking and Pinniped for authentication.

Table 3: Kubernetes Cluster Packages

Package	Version	Notes
Antrea	2.5.1	VKS core package
Gateway API	1.4.0	VKS core package
Guest-cluster auth-service	1.4.6	VKS core package
Metrics Server	0.8.1	VKS core package
Pinniped	0.42.0	VKS core package
Secretgen Controller	0.20.1	VKS core package
vSphere CPI	1.35.1	VKS core package
vSphere CSI	3.7.0	VKS core package
Cert-Manager	1.19.2	VKS add-on

H2O.AI Charts

For this validation, we pivoted to the latest GA version of H2O AI Cloud (HAIC) at the time. HAIC version used was 26.04.1 (change log) provided by Helm chart version 4.4.1.

Solution Validation

Install GPU Operator

There are two different approaches to install the GPU Operator depending on whether you are using Passthrough NVIDIA GPU or NVIDIA vGPU. Our validation presented in this document used NVIDIA vGPU.

Passthrough GPU

You can follow the steps below to install GPU Operator for Passthrough NVIDIA GPU in VKS:

Prerequisite: nodes Ready, a StorageClass exists, and outbound reachability confirmed to nvcr.io and helm.ngc.nvidia.com:

```
kubectl get nodes -o wide
curl -sI https://nvcr.io | head -1
curl -sI https://helm.ngc.nvidia.com | head -1
```

1. Add the Helm repo and create the namespace

```
helm repo add nvidia https://helm.ngc.nvidia.com/nvidia
helm repo update

kubectl create namespace gpu-operator

for level in enforce warn audit; do
  kubectl label namespace gpu-operator pod-security.kubernetes.io/${level}=privileged --overwrite
done
```

The PSA labels are required — the driver/toolkit pods need privileged mode and VKS's default Pod Security Admission will otherwise block them.

2. Install the operator

```
helm install --wait --generate-name \
  -n gpu-operator --create-namespace \
  nvidia/gpu-operator
```

No values file needed for passthrough — this is the plain OSS default install (no vGPU manager, no licensing config).

3. Verify

```
kubectl -n gpu-operator get pods

DRIVER_POD=$(kubectl -n gpu-operator get pod -l app.kubernetes.io/component=nvidia-driver -o jsonpath='{.items[0].metadata.name}')
kubectl -n gpu-operator exec "$DRIVER_POD" -c nvidia-driver-ctr -- nvidia-smi

kubectl get nodes -o json | jq '.items[] | {name: .metadata.name, gpu: .status.capacity."nvidia.com/gpu"}'
```

Pass criteria: all gpu-operator pods Running, nvidia-smi works inside the driver pod, GPU worker nodes advertise nvidia.com/gpu.

NVIDIA vGPU

You can follow the instructions below to install GPU Operator for NVIDIA vGPU in VKS:

- For Connected environment: https://techdocs.broadcom.com/us/en/vmware-cis/private-ai/foundation-with-nvidia/9-0/private-ai-foundation-9-x/deploying-ai-workloads-on-tkg-clusters/deploy-a-gpu-accelerated-tkg-cluster-with-kubectl-connected.html
- For Air-Gap environment: https://techdocs.broadcom.com/us/en/vmware-cis/private-ai/foundation-with-nvidia/9-0/private-ai-foundation-9-x/deploying-ai-workloads-on-tkg-clusters/deploy-a-gpu-accelerated-tkg-cluster-with-kubectl-disconnected.html

Install H2O AI Cloud on VKS

H2O AI Cloud is installed onto the VKS cluster with Helm. The procedure below follows the same structure used for the GPU Operator: each stage has an explicit pass criterion that must be met before proceeding. All commands are run against the VKS workload cluster context, not the Supervisor.

Prerequisites

Before installing the platform, confirm that the environment meets the following requirements: a default StorageClass backed by the vSphere CSI driver exists and can provision volumes; GPU worker nodes advertise the nvidia.com/gpu resource; a PostgreSQL instance version 11.0 or later is reachable, with a database and credentials created for the platform; S3-compatible object storage is reachable, with a bucket and credentials created; a wildcard DNS record resolves to the Contour ingress load balancer address and a matching wildcard TLS certificate is

available as a Kubernetes secret; an OIDC provider is reachable and a client has been registered for the platform; and credentials for the H2O.ai container registry and Helm repository have been issued by H2O.ai.

```
kubectl get storageclass

kubectl get nodes -o json | jq '.items[] | {name: .metadata.name,
gpu: .status.capacity."nvidia.com/gpu"}'

nslookup testwildcard.haic.example.com
```

Pass criteria: a default StorageClass exists, GPU nodes report nvidia.com/gpu capacity, and the wildcard DNS record resolves to the ingress address.

Create the Namespace and Registry Credentials

Create the target namespace and the image pull secret used to retrieve H2O AI Cloud images in the chart values.yaml file shown in the section below.

```
kubectl create namespace haic
kubectl label namespace haic pod-security.kubernetes.io/enforce=privileged --overwrite
```

Add the H2O Helm Repository

```
helm registry login registry-1.docker.io #enter password when prompted
helm pull oci://registry-1.docker.io/h2oairelease/h2oai-chart --version 4.4.1
```

Configure the Deployment Values

The platform is configured through a Helm values file. The values below show the settings that are specific to running on VKS: the vSphere CSI storage class, the Contour ingress and wildcard TLS configuration, and the scheduling constraints that place GPU-backed engines onto the GPU node pool.

```
$ head -n 30 values-broadcom-lab.yaml
global:
  cloudProtocol: https
  cloudHost: haic.10-139-10-91.nip.io

  cors:
    policy: "appstore"

  storageClassName: vsan-esa-default-policy-raid5

  createNamespaces: true

  imageRegistry: docker.io/h2oairelease
  imagePullSecret:
    registry:
      annotations:
        "helm.sh/hook": pre-install, pre-upgrade
        "helm.sh/hook-weight": "-5"
      data:
        registry: docker.io/h2oairelease
        username: xxxxxxxxx
        password: zzzzzzzzzzzzzzzzz
        email: aaaaaa@h2o.ai
  imagePullSecrets:
    - h2oai-registry
```

```
  securityContext:
    pod:
      preferGlobal: true
    container:
      preferGlobal: true
```

Install the Platform

```
helm upgrade --install h2oai h2oai-chart-4.4.1.tgz -f values-broadcom-lab.yaml -n haic
```

The install deploys the platform services, the AI App Store server and their dependencies. Depending on the number of optional components enabled and the speed of image pulls from the registry, the initial deployment can take a considerable time.

Verify the Deployment

```
kubectl -n haic get pods
kubectl -n haic get ingress
kubectl -n haic get pvc
helm -n haic status h2oai
```

Pass criteria: all pods in the haic namespace reach Running or Completed with no restart loops, every PersistentVolumeClaim is Bound, the Ingress resource has an external address assigned, and the Helm release reports status deployed.

Figure 5 shows the H2O AI Cloud deployments in the haic namespace, with every deployment reporting all replicas ready.

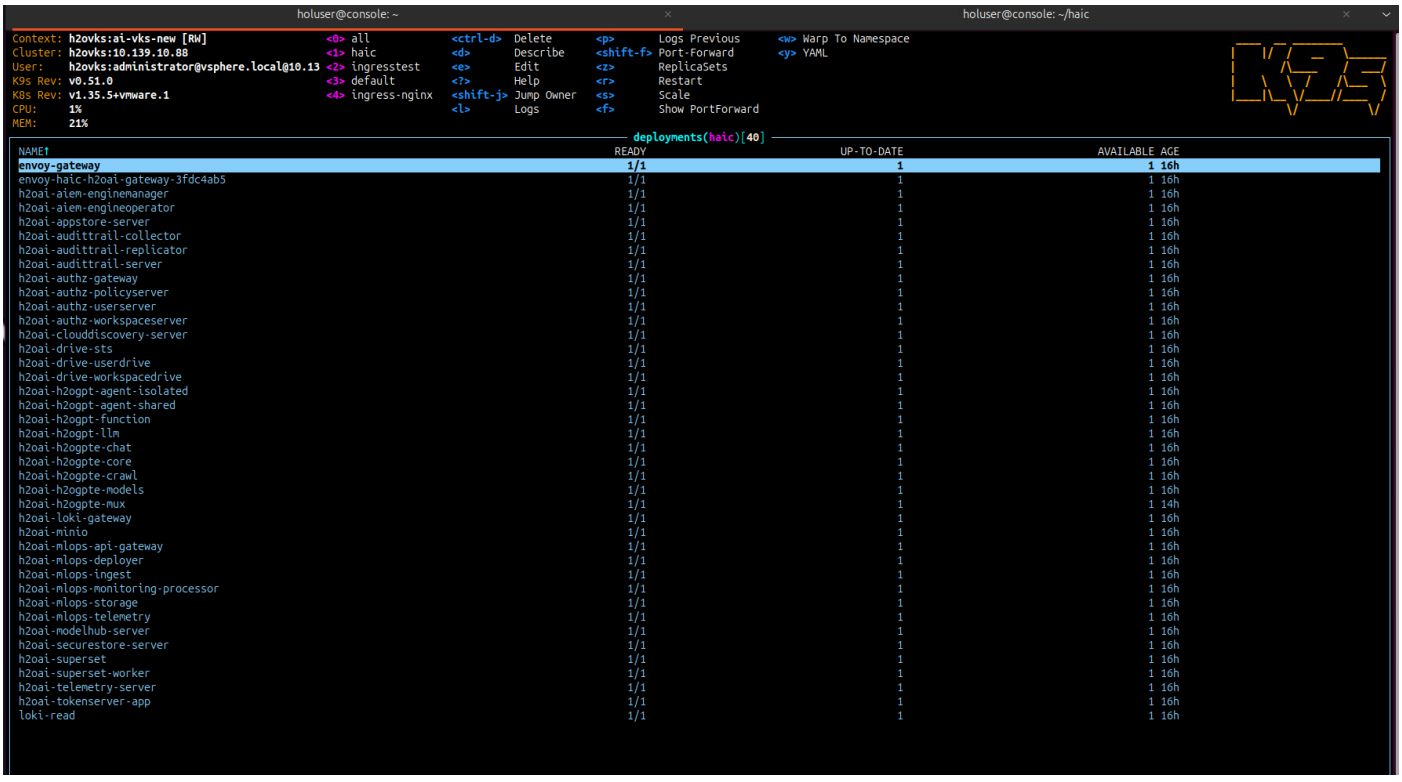


Figure 5: H2O AI Cloud Deployments in the haic Namespace

Figure 6 shows the stateful platform components, including Keycloak, PostgreSQL, RabbitMQ and Redis, running as StatefulSets with all replicas ready.

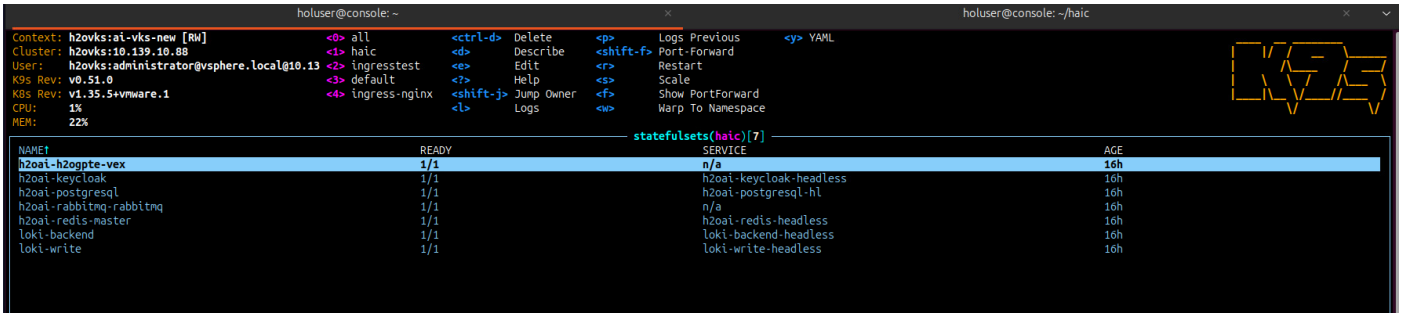


Figure 6: H2O AI Cloud StatefulSets in the haic Namespace

Figure 7 shows the ingress resources created for the platform services. Each service is exposed on its own subdomain of the platform base domain and has an external address assigned.

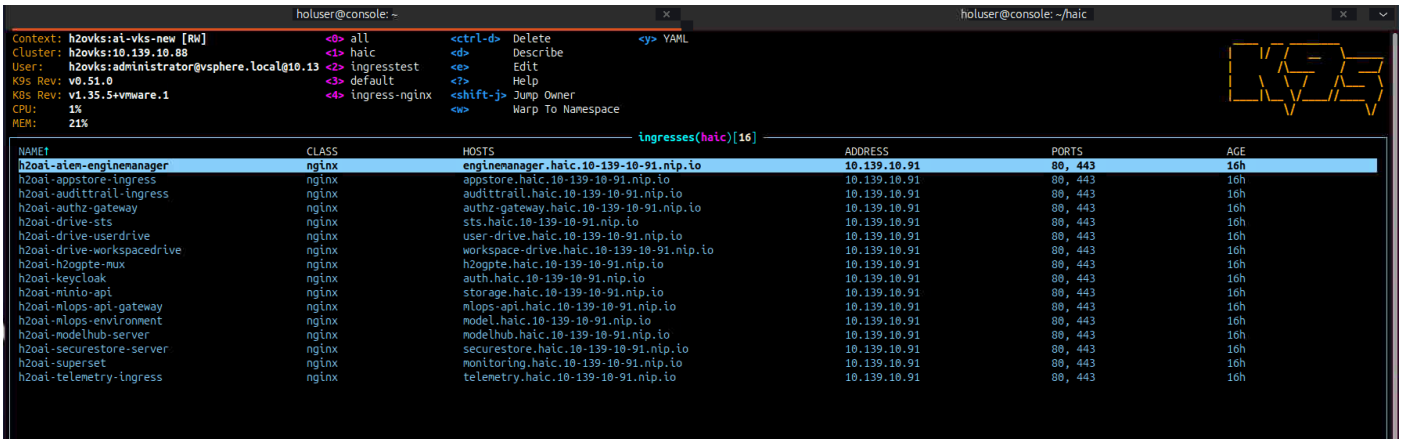


Figure 7: H2O AI Cloud Ingress Resources in the haic Namespace

Validate H2O AI Cloud on VKS

Validation proves that the platform is not merely installed but functional end to end: that a user can authenticate, obtain GPU-accelerated compute on demand, train a model, promote it to production, score against it, and exercise the generative AI capability. Each step below states what is executed and what constitutes a pass.

Platform Reachability and Authentication

Browse to the platform address and authenticate through the configured OIDC provider. The H2O AI Cloud home page should render with the Workspace selector, the app instances widget, the AI engines widget and the App Store navigation present.

Figure 8 shows the Workspace AI Engines page, reached after authenticating to the platform as a standard user.

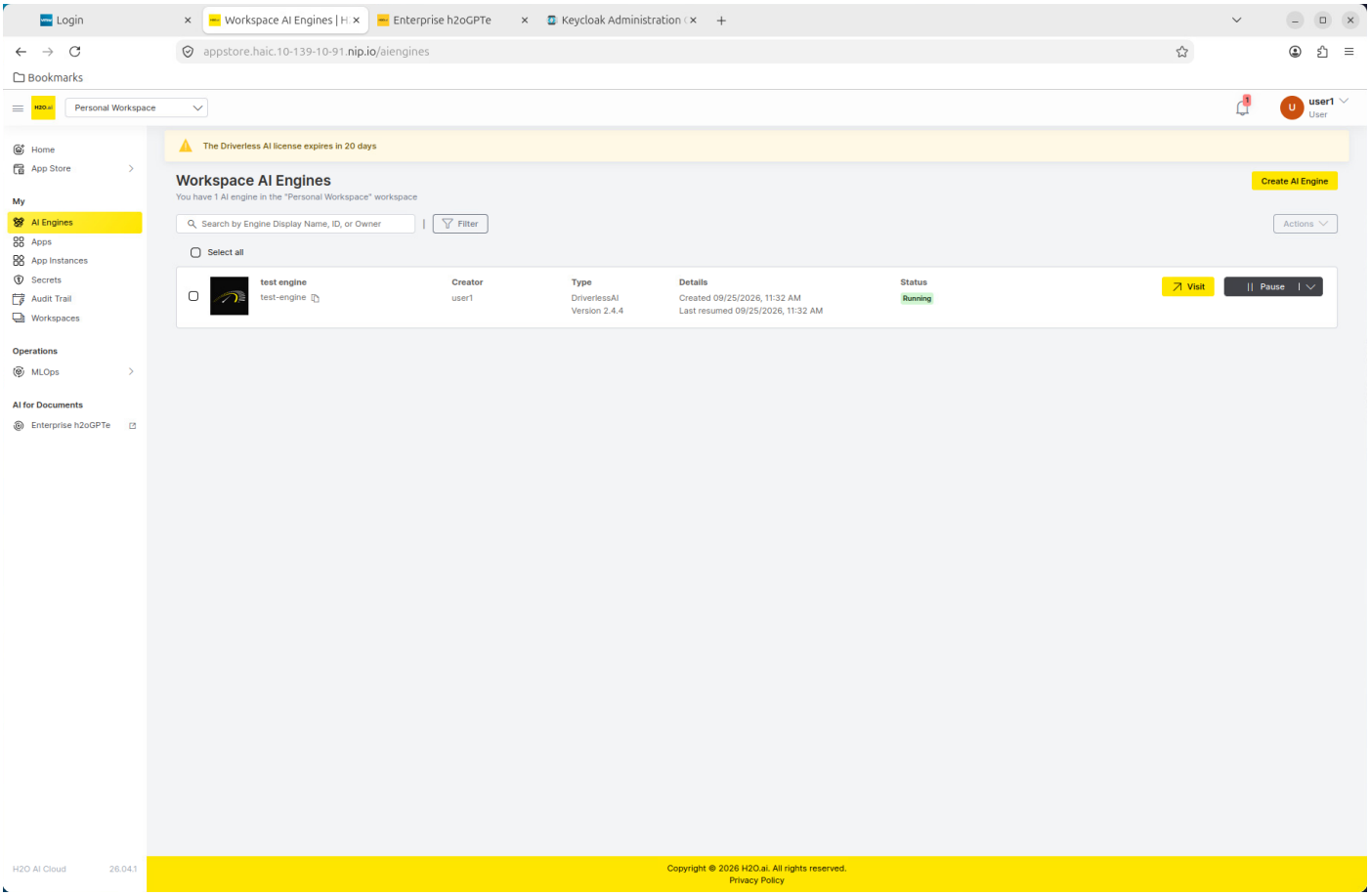


Figure 8: H2O AI Cloud Workspace AI Engines Page

AI Engine Provisioning

From AI Engines, create a Driverless AI engine. Confirm that the engine pod is initiated and Driverless AI can be accessed through the UI for model training.

Figure 9 shows the Driverless AI experiment setup page, opened from the running engine, with the iris.csv dataset selected for training.

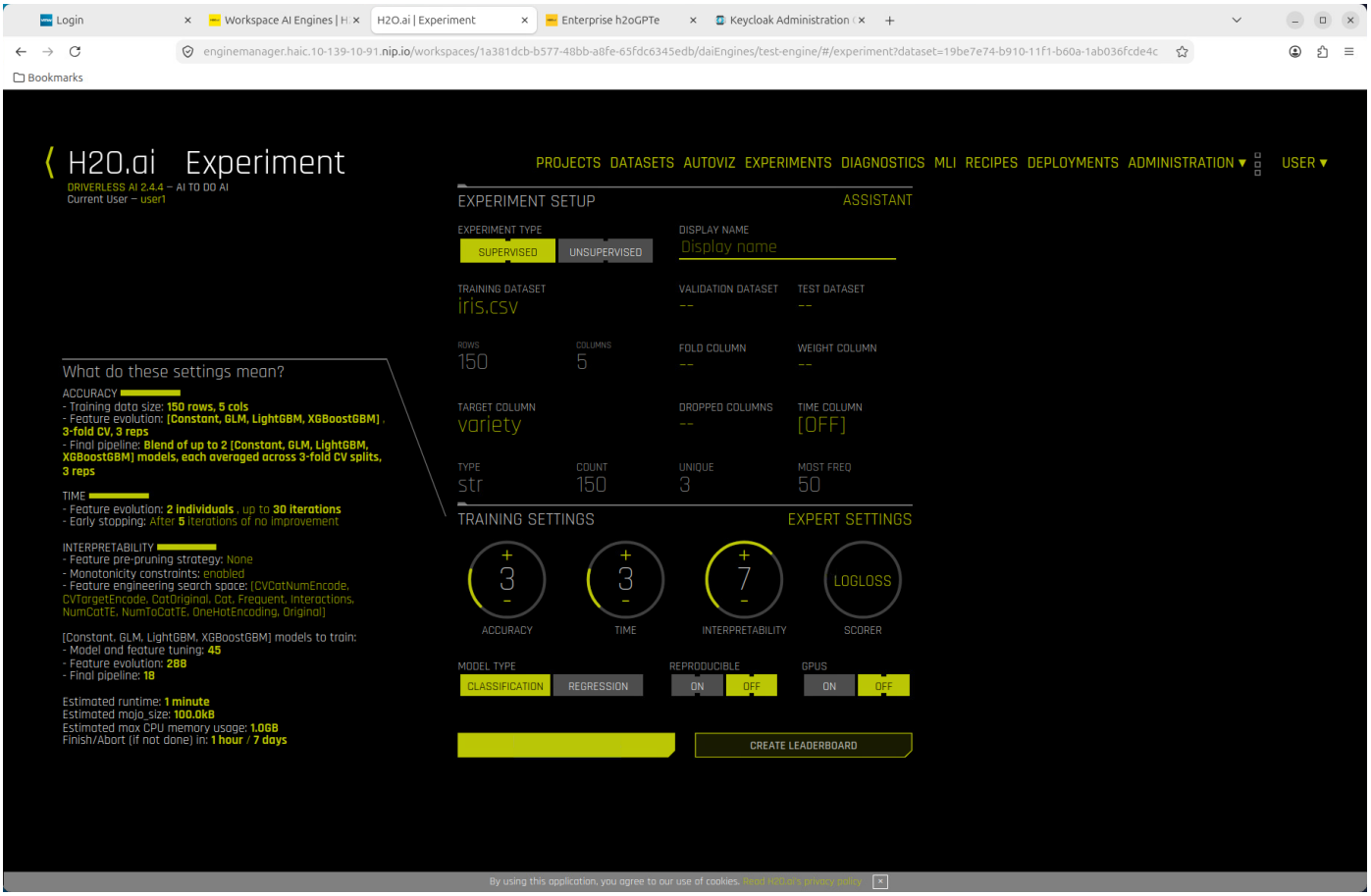


Figure 9: Driverless AI Experiment Setup with the Iris Dataset

Model Training

Upload a dataset and run an experiment on the GPU-backed engine to completion. A well-known public dataset is sufficient for a functional validation.

Figure 10 shows the experiment in progress, using the public Iris dataset (150 rows, 5 columns) with variety as the target column.

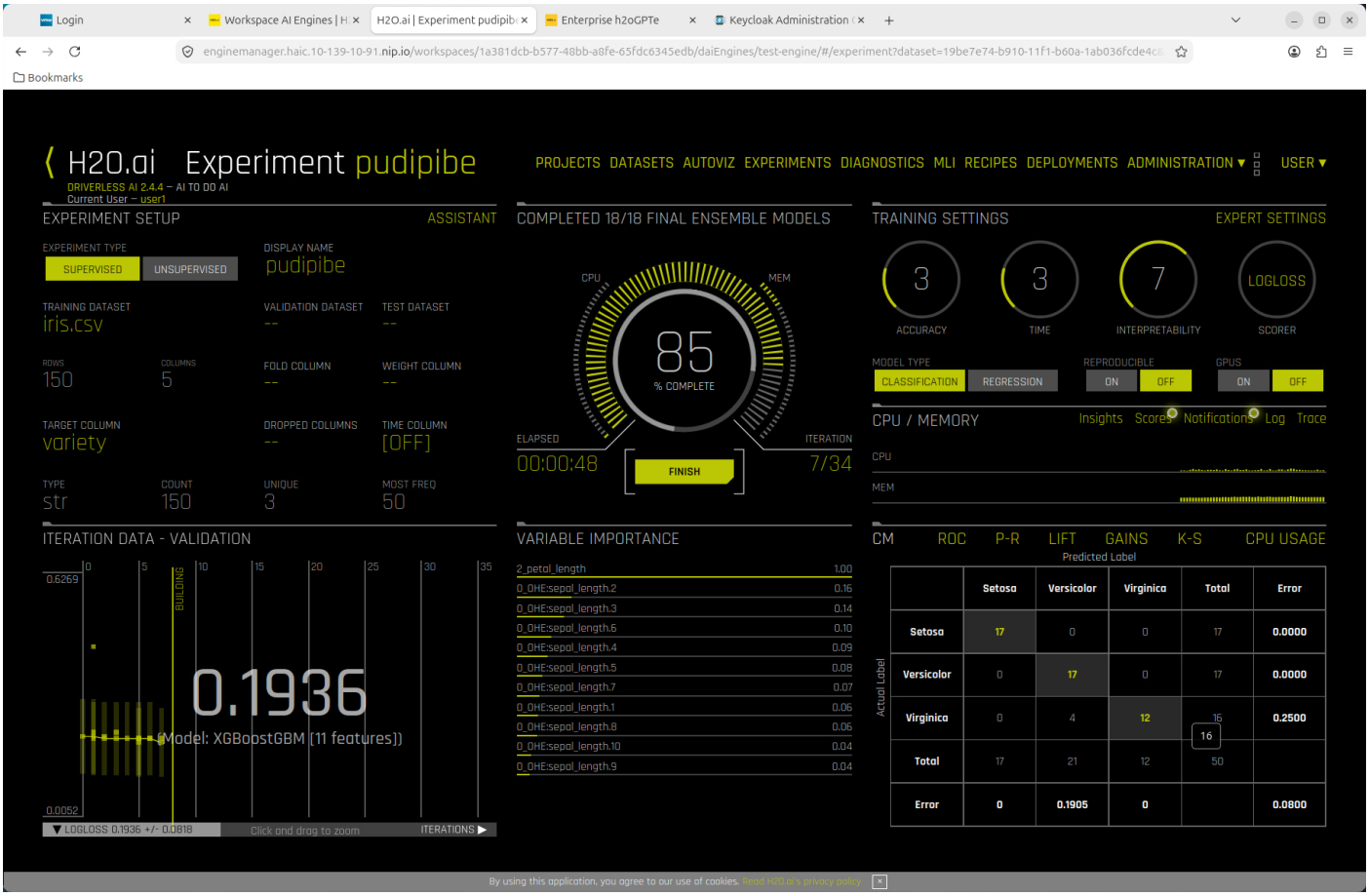


Figure 10: Driverless AI Experiment in Progress

Pass criteria: the experiment completes without error, GPU utilization is observed on the node during training, and the resulting model is visible in the platform.

Figure 11 shows the completed experiment, with the final model built and the model actions, including Deploy, available.

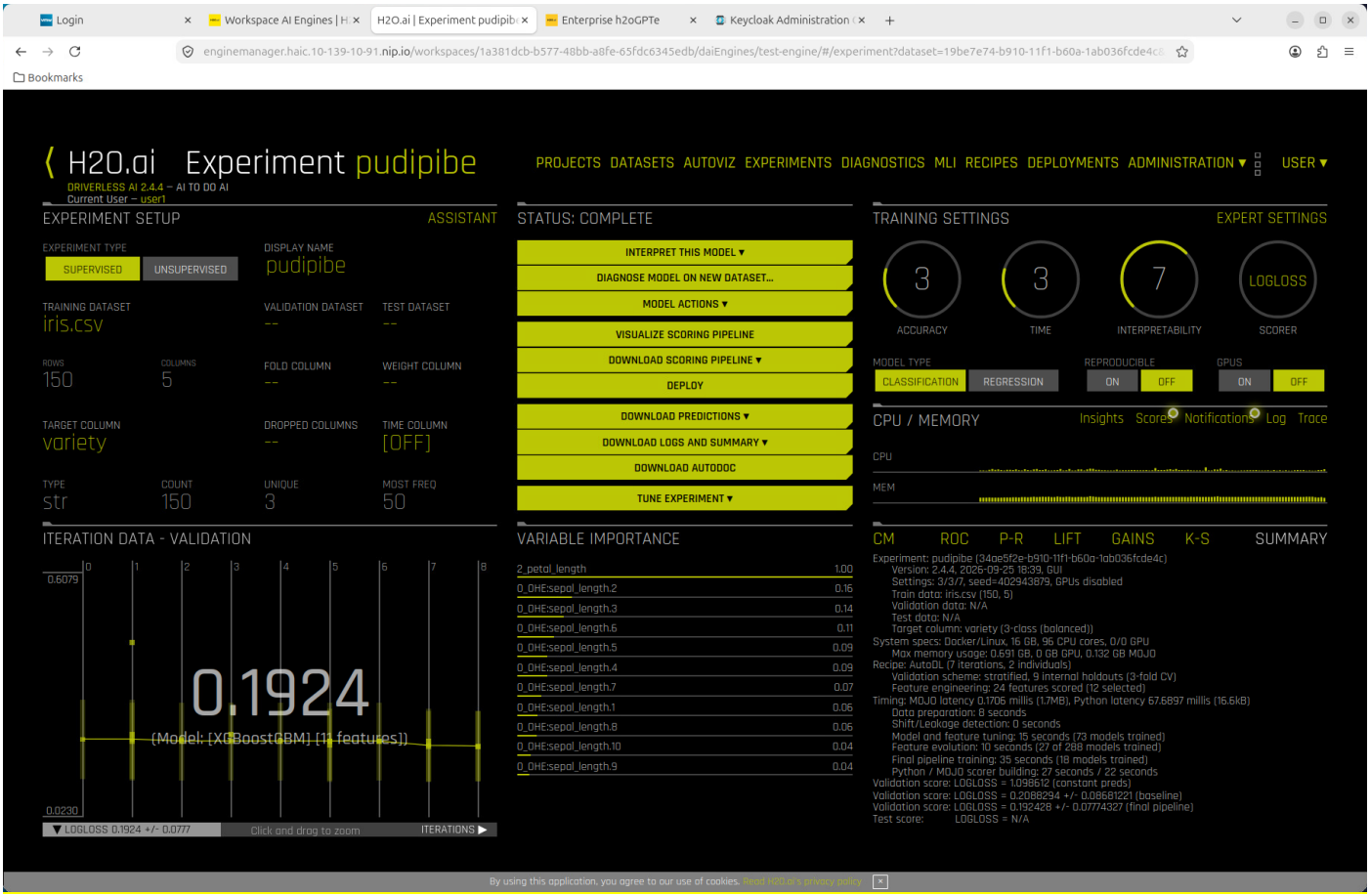


Figure 11: Completed Driverless AI Experiment Summary

Model Deployment and Scoring

Promote the trained model into H2O MLOps, deploy it as a scoring endpoint, and submit a scoring request against that endpoint.

Figure 12 shows the trained model registered in H2O MLOps, with the Deploy action available from the model menu.

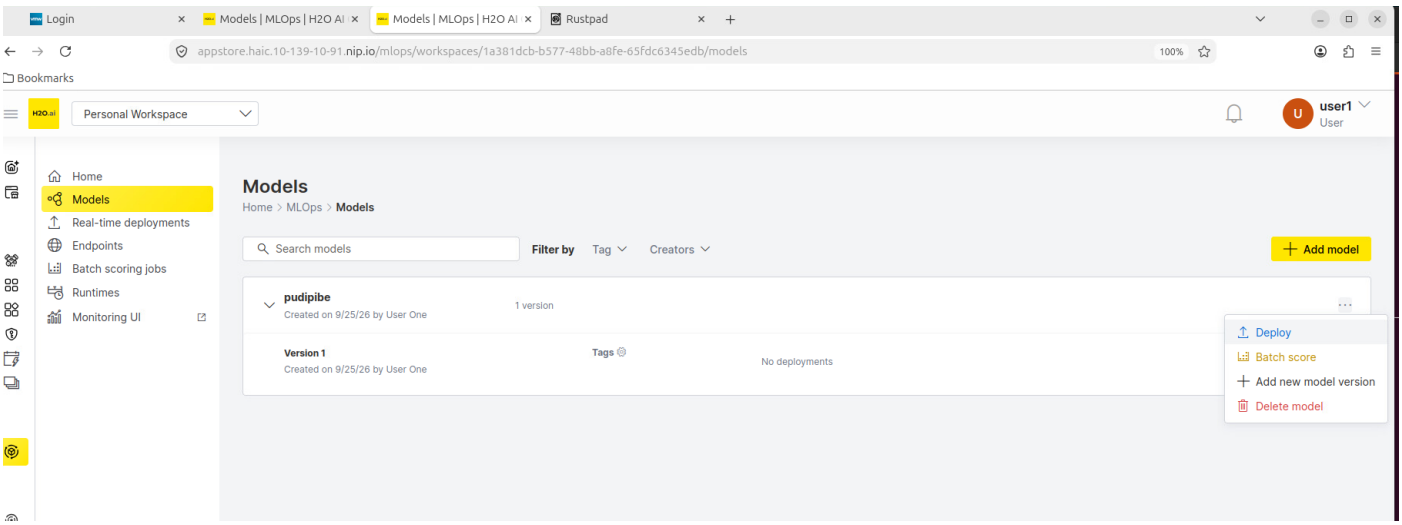


Figure 12: Trained Model Registered in H2O MLOps

Figure 13 shows a real-time deployment being created for the model, using the DAI MOJO Scorer runtime.

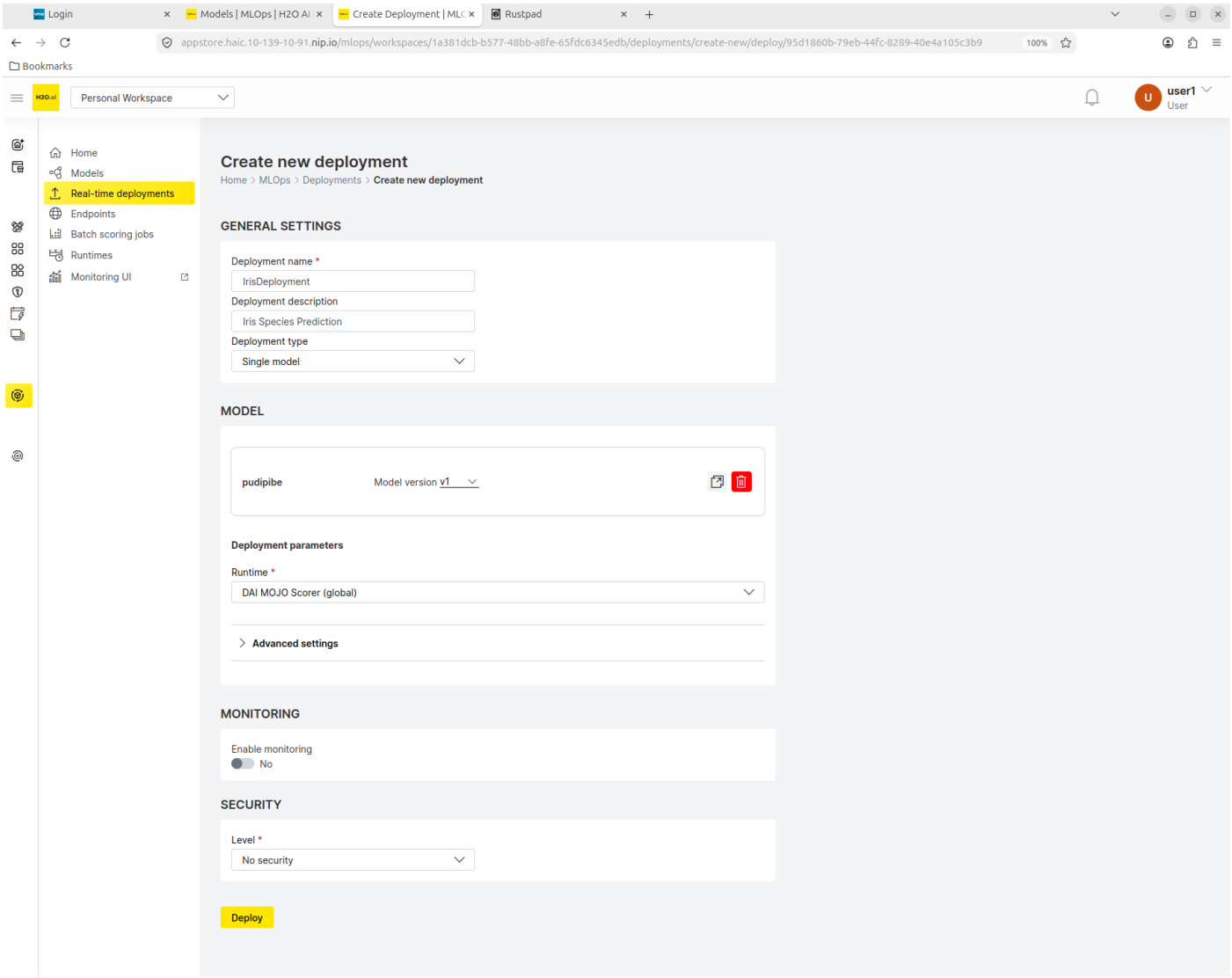


Figure 13: Creating a Real-Time Deployment in H2O MLOps

Figure 14 shows the resulting deployment in a Healthy state, together with the scoring endpoint URL used in the request below .

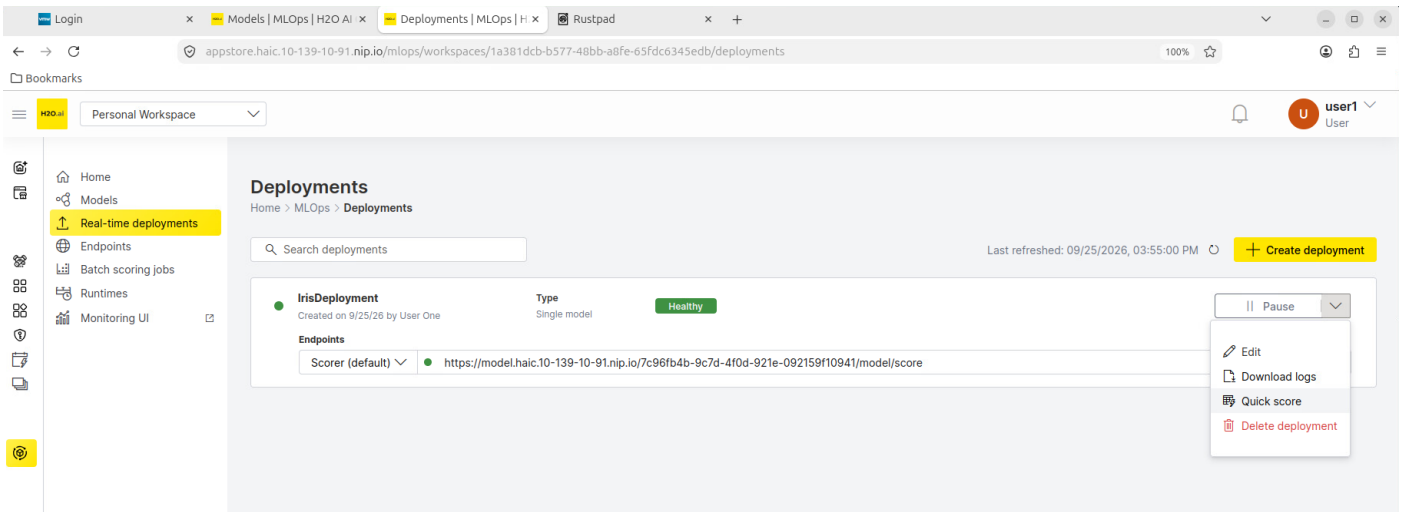


Figure 14: Healthy MLOps Deployment with Scoring Endpoint

```
$ curl -k -X POST -H "Content-Type: application/json" -d
'{"fields":["sepal.length","petal.length"],"rows":[["3.4","2.1"]]}'
https://model.haic.10-139-10-91.nip.io/7c96fb4b-9c7d-4f0d-921e-092159f10941/model/score


{"id":"34ae5f2e-b910-11f1-b60a-
1ab036fcde4c","fields":["variety.Setosa","variety.Versicolor","variety.Virginica"],"sco
re":[["0.8751154211855661","0.0893804921405984","0.035504086673835485"]]}
```

Pass criteria: the deployment reaches a healthy state in MLOps, the scorer pod is Running, and the scoring request returns a valid prediction.

AI Application Delivery

Launch an application from the AI App Store and confirm that its instance is exposed on its own subdomain of the platform base domain. This proves that the wildcard DNS and TLS configuration is correct, which is the configuration step most often missed in on-premises deployments.

Figure 15 shows the Data Science applications available in the AI App Store.

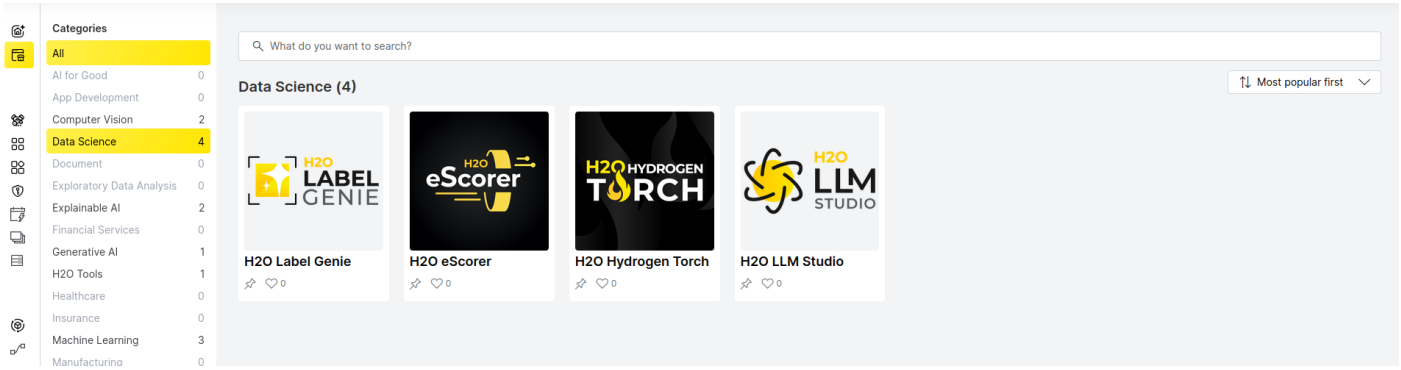


Figure 15: AI App Store – Data Science Applications

Figure 16 shows the details page for the H2O Hydrogen Torch application, from which an application instance can be launched.

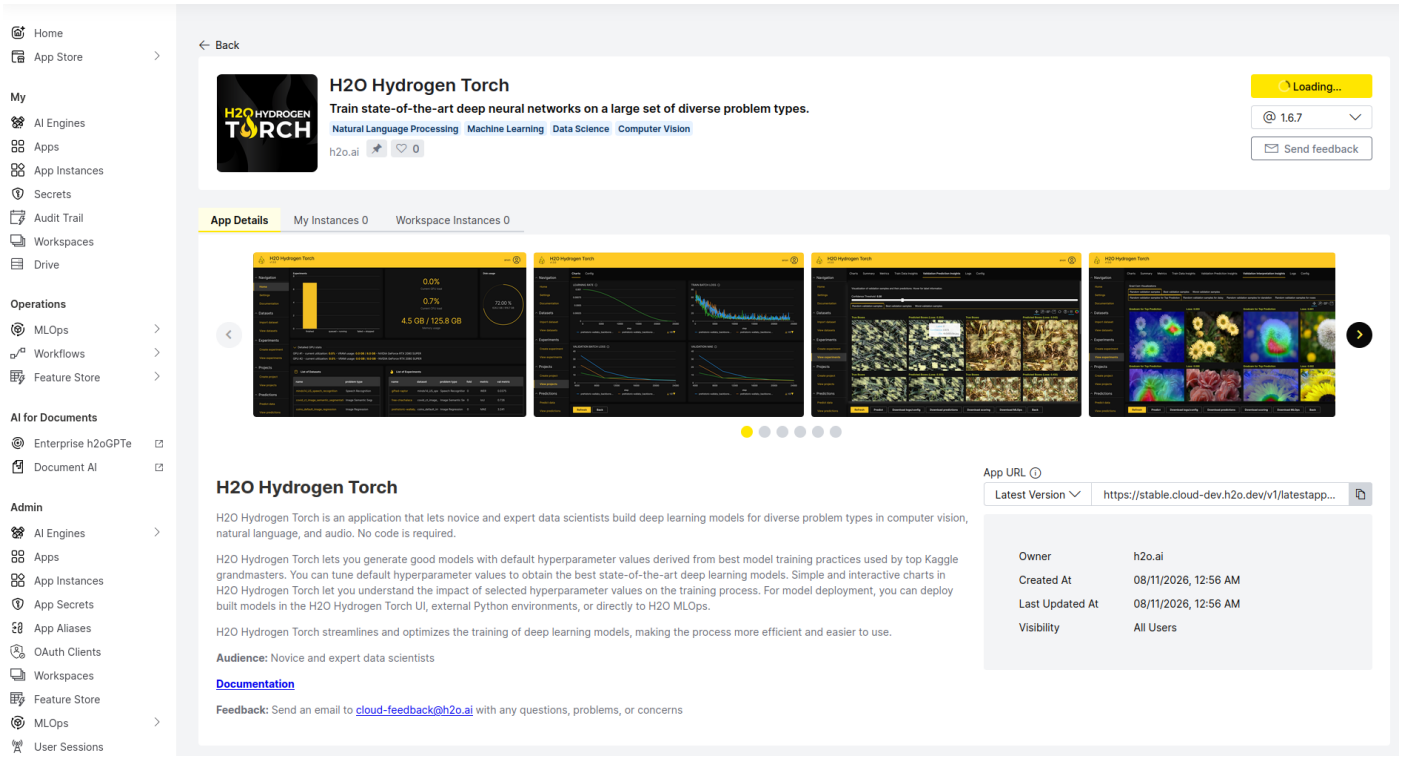


Figure 16: H2O Hydrogen Torch Application in the AI App Store

Generative AI

Where Enterprise h2oGPTe is included in the deployment, create a document collection, ingest documents into it, and submit a retrieval-augmented query. This exercises the GPU node pool under inference load rather than training load.

Pass criteria: document ingestion completes, the query returns a grounded response with citations to the ingested documents, and GPU utilization is observed on the serving node during inference.

Figure 17 shows a chat session with Enterprise h2oGPTe, served from its own subdomain of the platform base domain.

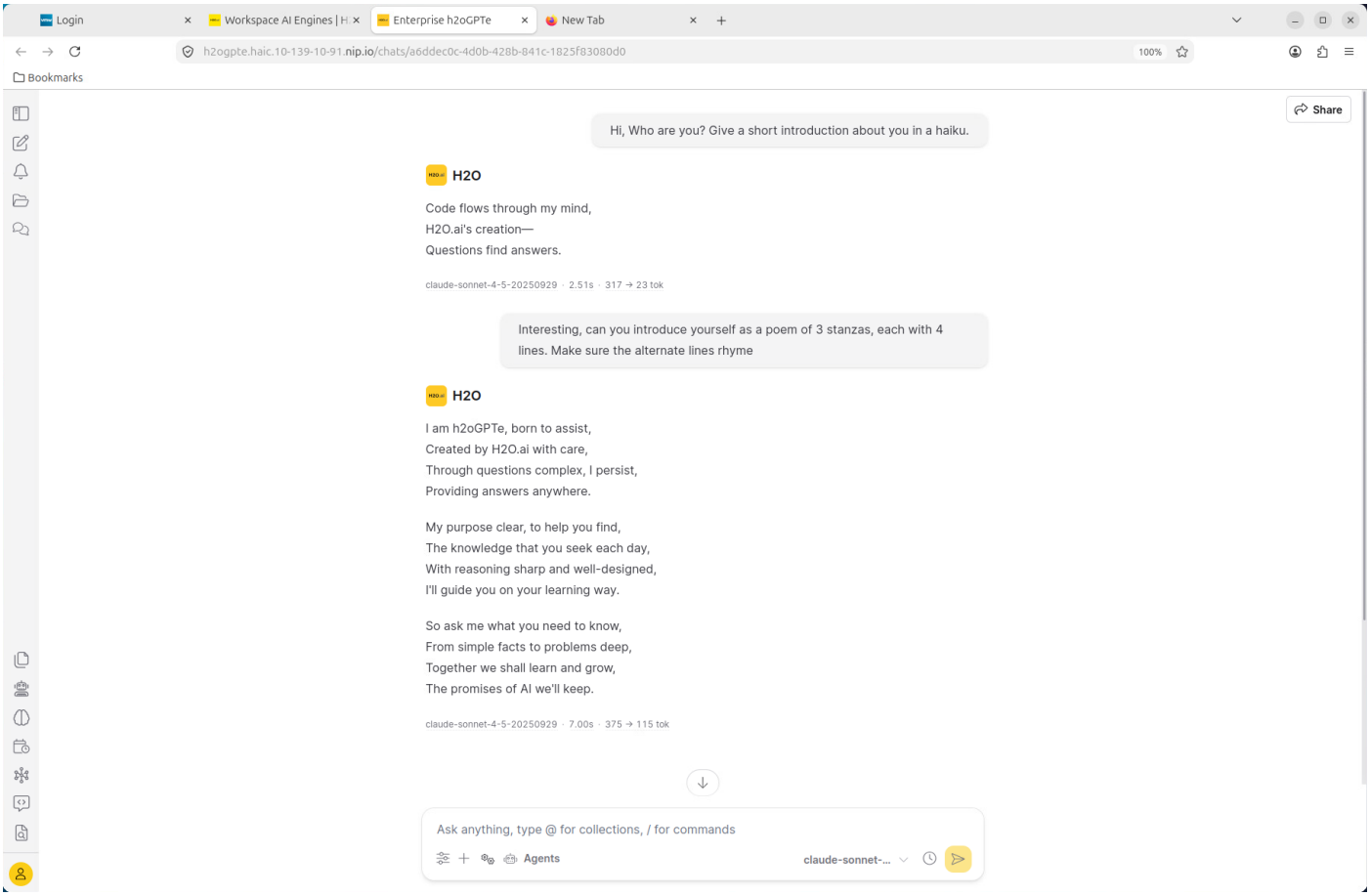


Figure 17: Enterprise h2oGPTe Chat Session

Figure 18 shows the chat configuration with the agent enabled, and a request to plot Apple and Nvidia closing stock prices submitted to the General Agent.

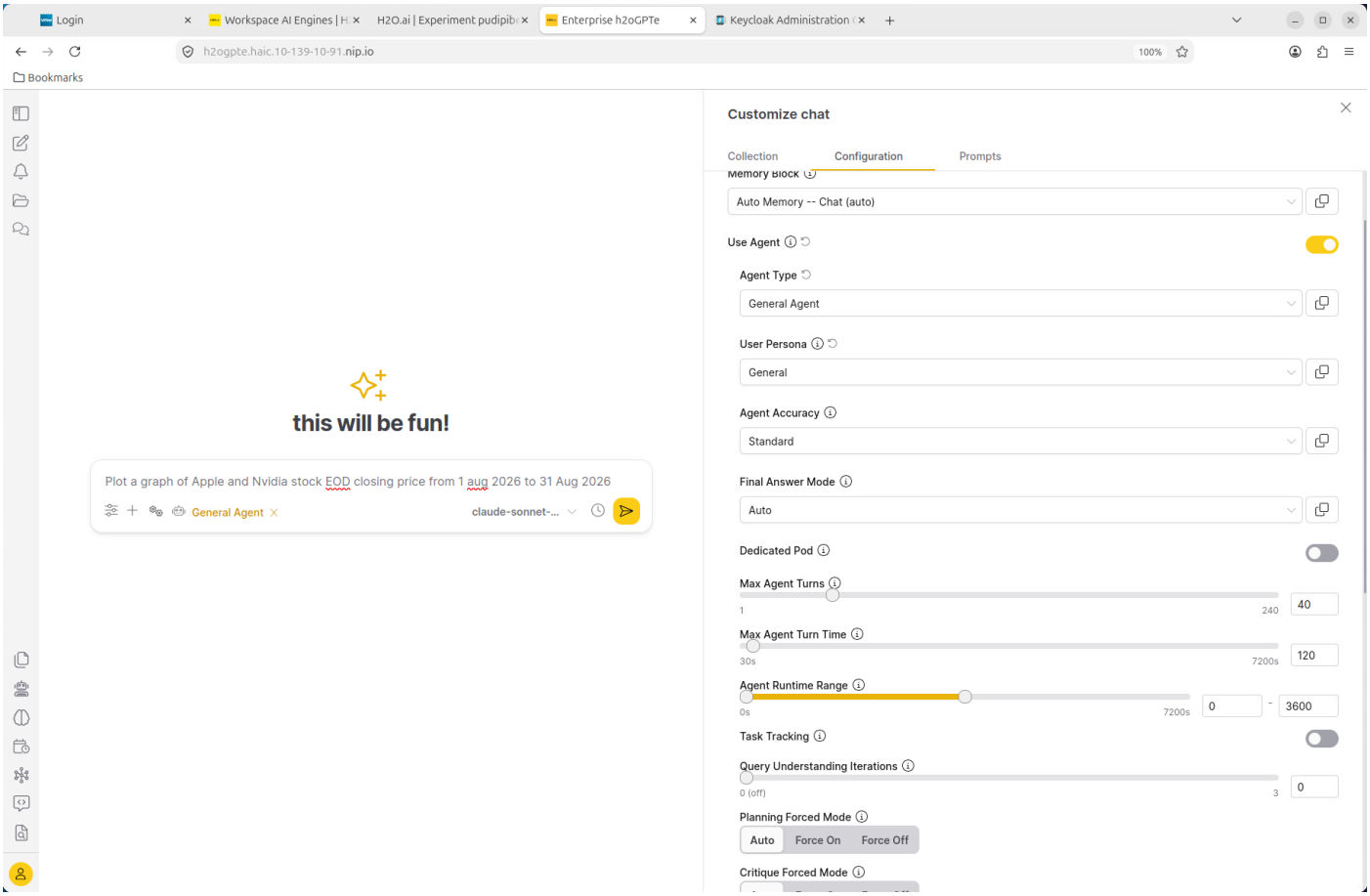


Figure 18: Enterprise h2oGPTe Agent Configuration

Figure 19 shows the agent's response, including a summary analysis and the chart it generated.

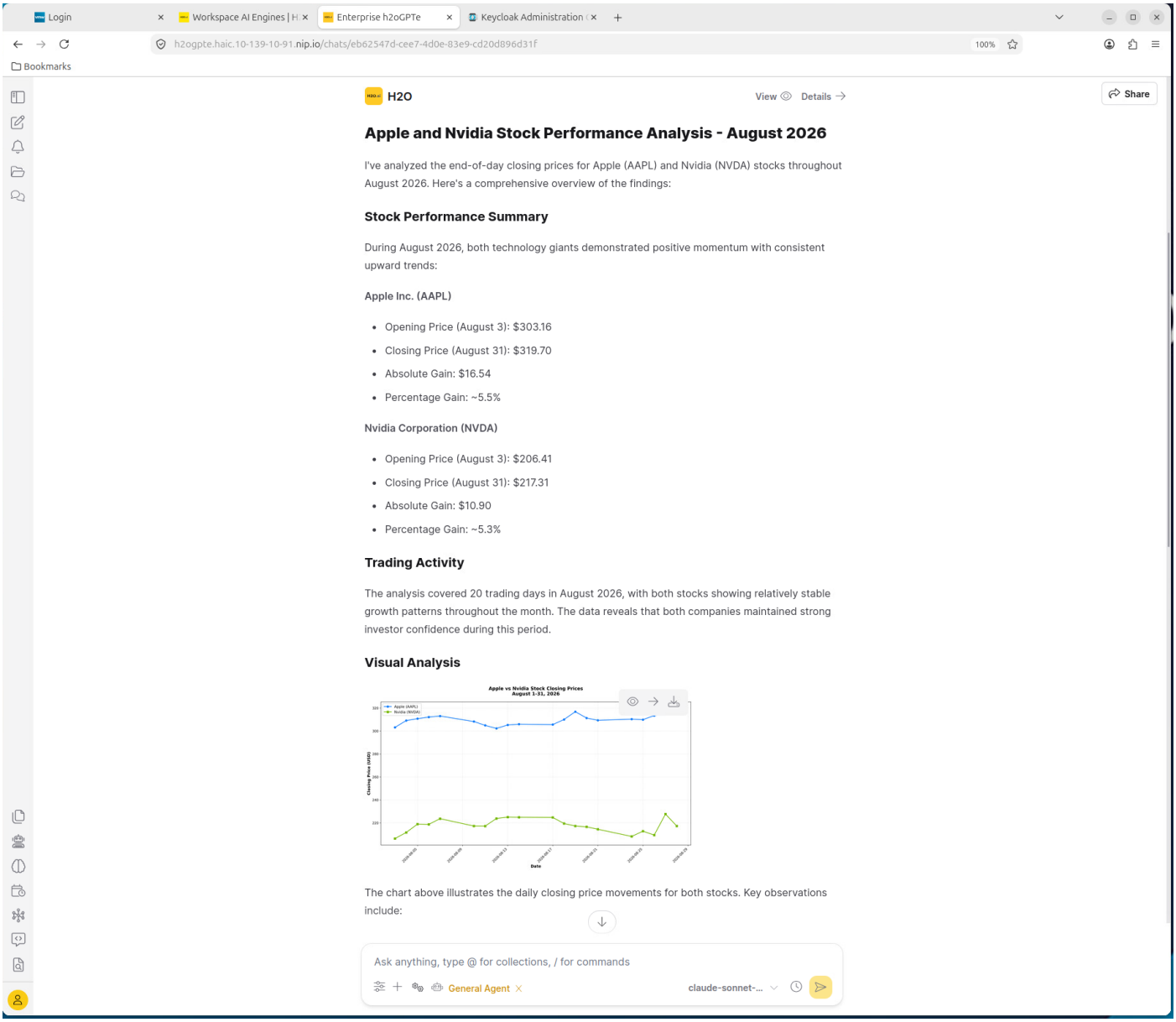


Figure 19: Enterprise h2oGPTe Agent Response with Generated Chart

Conclusion

This paper has described the deployment of H2O AI Cloud, the H2O.ai enterprise AI platform, onto vSphere Kubernetes Service running on VMware Cloud Foundation, and the validation performed to confirm that the combined stack functions end to end.

The result for a joint customer is straightforward. An organization that has already standardized on VCF does not need to build a parallel platform in order to adopt enterprise AI. The same vSphere and Platform administrators, the same lifecycle management, the same vSAN storage policies, the same network and security boundary and the same GPU hardware investment serve both the existing estate and the AI platform. Data scientists consume GPU capacity as a self-service resource through AI Engine Manager rather than through a ticket, while data, models and prompts remain inside infrastructure the organization controls — the requirement that drives most regulated enterprises toward private AI in the first place.

For Broadcom, this validation extends the set of production AI workloads that VKS is proven to run. For H2O.ai, it establishes VCF as a supported and tested deployment target for the enterprise platform. For customers of both companies, it removes the integration risk that would otherwise sit between the two products and gives both organizations a shared, documented configuration to support.

Next steps. The manifests, Helm values and supporting configuration referenced in this paper are published in the companion repository. Customers evaluating this architecture should begin by confirming GPU readiness on their ESX hosts and creating a GPU-enabled VM Class, then follow the procedure in the Solution Validation section.



Copyright © 2026 Broadcom. All rights reserved.
The term “Broadcom” refers to Broadcom Inc. and/or its subsidiaries. For more information, go to www.broadcom.com. All trademarks, trade names, service marks, and logos referenced herein belong to their respective companies. Broadcom reserves the right to make changes without further notice to any products or data herein to improve reliability, function, or design. Information furnished by Broadcom is believed to be accurate and reliable. However, Broadcom does not assume any liability arising out of the application or use of this information, nor the application or use of any product or circuit described herein, neither does it convey any license under its patent rights nor the rights of others.
Item No: vmw-bc-wp-tech-uslet-word-2026; Aug-26