TECHNICAL WHITE PAPER: **October 2026**



Mendix on vSphere Kubernetes Service on VMware Cloud Foundation

Reference Architecture

Table of Contents

Executive Summary 3
vSphere Kubernetes Service 3
Mendix Platform 4
Solution Architecture 5
Component Responsibilities 6
Network Flows 7
Design Considerations 7
VKS Cluster Configuration 7
Kubernetes Packages 8
Solution Validation 8
Install Mendix on VKS 8
Step 1: Install Contour and a Gateway 9
Step 2: Register the cluster in the Mendix Portal 9
Step 3: Install the Mendix Operator and Agent 9
Step 4: Configure registry, plans and routing 10
Validate Mendix on VKS 10
Conclusion 11
Resources 11

Executive Summary

This paper describes a validated reference architecture for running Mendix applications on vSphere Kubernetes Service (VKS) on VMware Cloud Foundation (VCF). Mendix is a platform for agentic development and orchestration, used to build the workflows, AI agents and applications that enterprises run on. Mendix is part of Siemens Intelligence Center X (ICX). With Mendix on Kubernetes, the customer runs the applications and keeps their data in its own VKS clusters. Developers keep deploying from the Mendix Platform.

VKS provides the cluster, the load balancer, persistent storage and Gateway API routing through the Contour add-on. Mendix provides the Operator and the Agent, which build and run the apps. The customer adds a relational database, S3-compatible object storage and a container registry, chosen from products it already runs.

The design was validated on VKS 3.6.2 with Kubernetes 1.35 and Mendix Operator 2.28.1. A Mendix app was deployed from the Mendix Portal, stored its data in a relational database and object storage, and served users through a Gateway API route.

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.

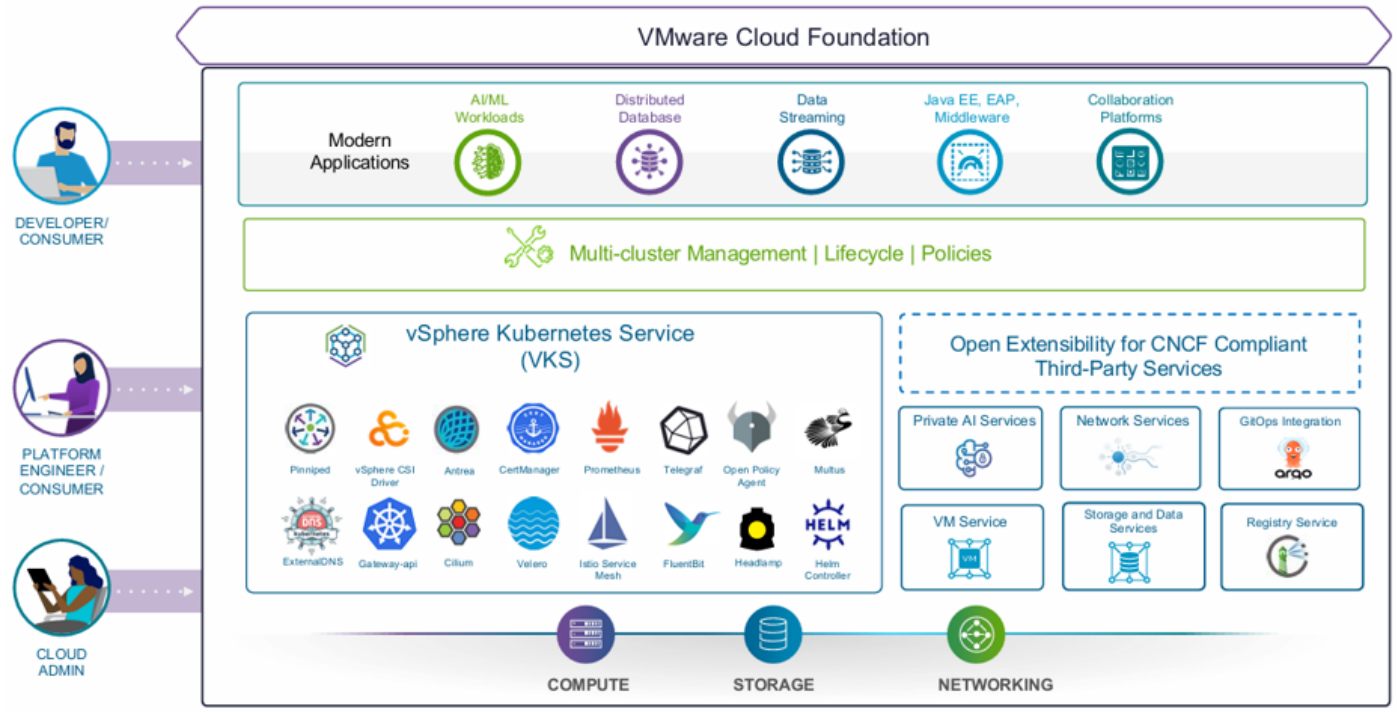


Figure 1: VKS on VCF Ecosystem

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.

Mendix Platform

Mendix is a platform for agentic development and orchestration. Teams use it to build the workflows, AI agents and applications their business runs on, with visual low-code modeling and AI-assisted development.

Mendix is part of Siemens Intelligence Center X (ICX). ICX combines Mendix with Siemens Graph Studio and AI Studio to turn IT, OT and engineering data into a system of action for industrial enterprises, with people and AI agents working in one governed environment.

Developers model an app in Mendix Studio Pro and deploy it from the Mendix Portal. Mendix on Kubernetes, also called Mendix for Private Cloud, runs those apps on a Kubernetes cluster that the customer controls.

Two Mendix components run inside the cluster:

- **Mendix Operator** builds a container image for each deployment and runs the app environments. It also creates a database, a storage bucket and a network route for each environment. See Introduction to Operators and Operator Flows.
- **Mendix Agent** connects the cluster to the Mendix Portal. The connection always starts from the cluster, so the Portal never needs a way into the customer network. See Creating a Mendix on Kubernetes Cluster.

This paper uses Connected mode, where teams manage environments in the Mendix Portal. Standalone mode manages the same resources with Kubernetes manifests and needs no Portal connection. Figure 2 shows how a deployment moves from the Mendix Platform into the VKS cluster.

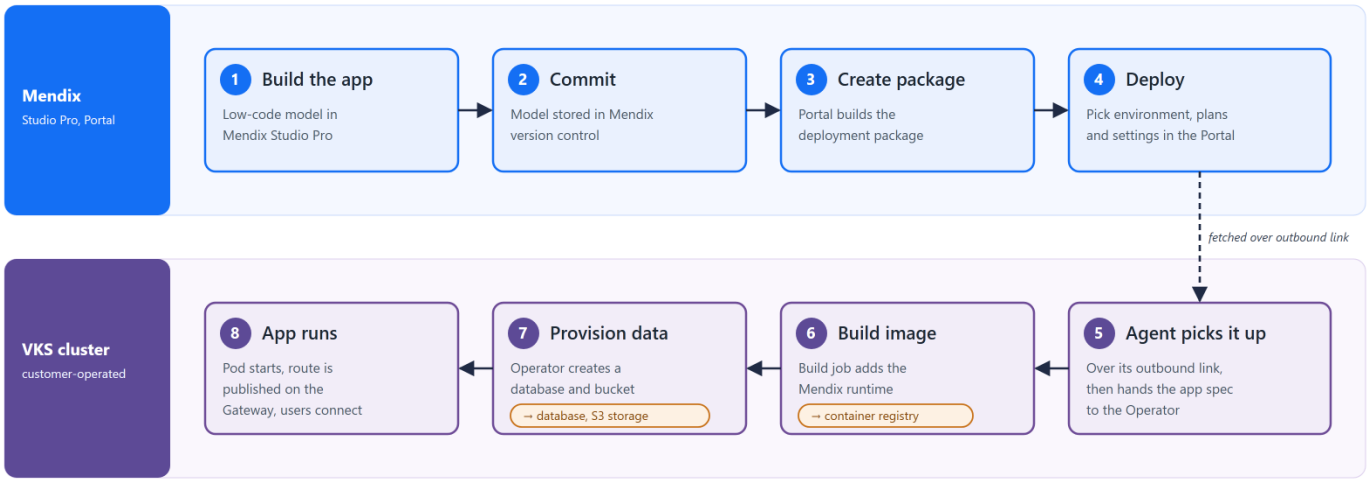


Figure 2: Mendix deployment workflow

Solution Architecture

Figure 3 shows the reference architecture. The colors show who supplies each component: Mendix, VMware, or the customer. The customer-supplied parts are generic building blocks. Any product that meets the requirements in Table 2 fits.

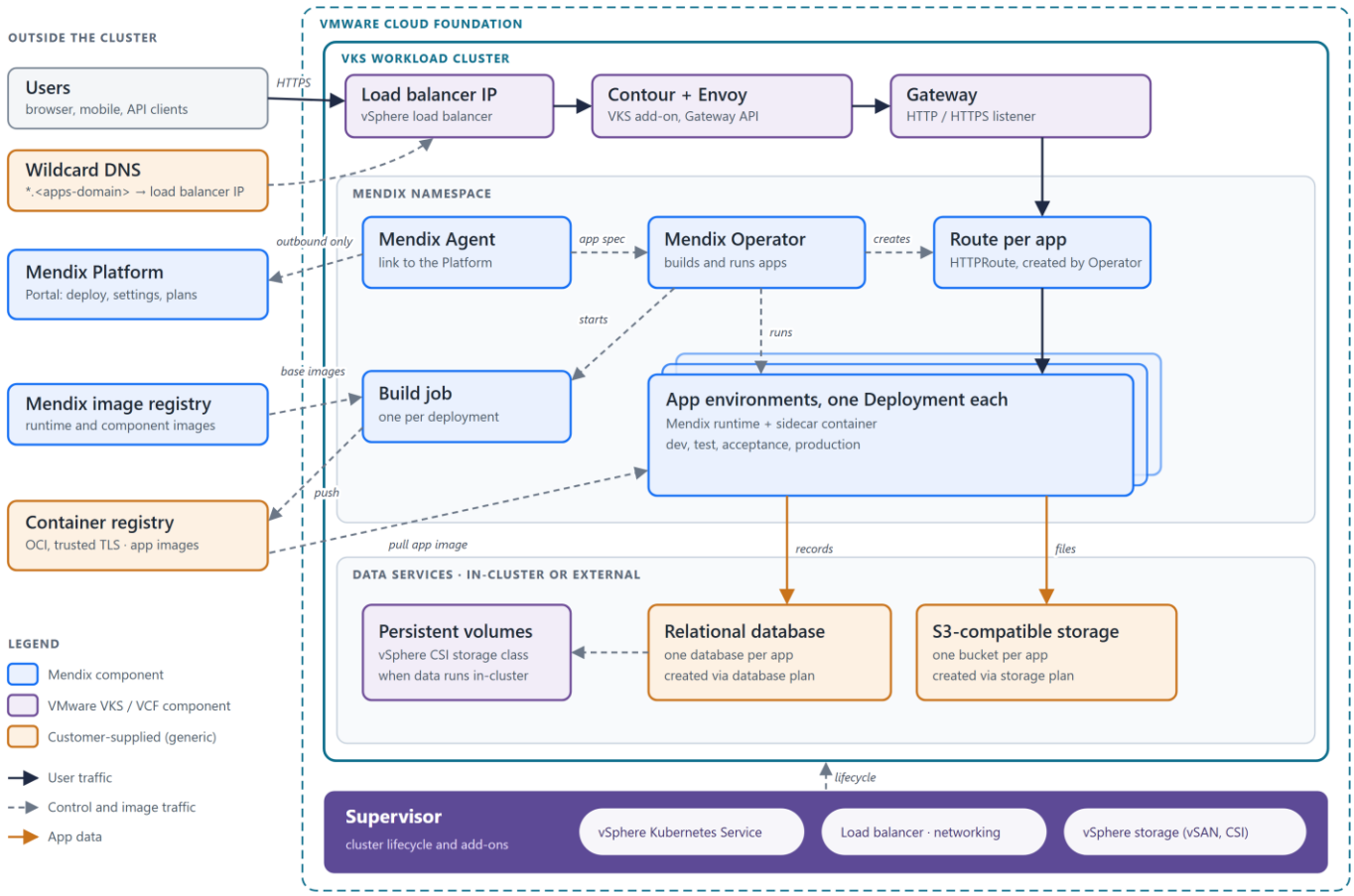


Figure 3: Mendix on VKS reference architecture

At its core, the Supervisor control plane provides essential services, such 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) via Cluster API. The specific version of VKS used in this white paper is 3.6.2.

Table 1: Installed Supervisor Components

Component	Version	Notes
vSphere Kubernetes Service	3.6.2	Lifecycle of CNCF-conformant Kubernetes clusters
vSphere Foundation Load Balancer		Layer 4 load balancer. Gives the Envoy service its external IP

Component Responsibilities

Table 2 lists each component, who supplies it, and which products fit. Product names in the last column are examples, not requirements.

Table 2: Component responsibilities

Component	Supplied by	Role	Requirements and options
Mendix Platform (Portal)	Mendix	Deploys apps. Holds environment settings and the database and storage plans	Software as a service, used in Connected mode. Deploying an app
Mendix Agent	Mendix	Links the cluster to the Portal	Outbound HTTPS from the cluster to Mendix
Mendix Operator	Mendix	Builds images, runs environments, creates the database, bucket and route per environment	Installed with the Mendix CLI `mxpc-cli`
Mendix runtime	Mendix	Runs the app. One Deployment per environment, with a sidecar container	Base images come from the Mendix image registry
VKS workload cluster	VMware	Runs the Mendix namespace and the app pods	Kubernetes version within the Mendix supported range
Load balancer	VMware	External IP for the ingress service	vSphere Foundation Load Balancer, or the load balancer of the Supervisor network
Contour + Envoy	VMware (VKS add-on)	Gateway API implementation for user traffic	Installed and upgraded through the Supervisor
Gateway API	VMware (VKS package)	Standard routing API. The Operator creates one HTTPRoute per environment	Gateway must accept routes from the Mendix namespace. Network ingress settings
Persistent volumes	VMware	Disks for data services that run in the cluster	vSphere CSI storage class, for example on vSAN
Container registry	Customer	Stores the app images that the build job pushes. Nodes pull from it	Any OCI registry with a trusted certificate. Registry configuration
Relational database	Customer	One database and user per environment	PostgreSQL 13 to 17 (17 tested) or another engine on the Mendix supported list. Admin role with CREATEDB and CREATEROLE is enough. Storage plans
S3-compatible object storage	Customer	One bucket per environment for uploaded files	Any S3-compatible service. Examples: an existing enterprise object store, Ceph Object Gateway, a commercial S3 product. Storage plans

Component	Supplied by	Role	Requirements and options
DNS and certificates	Customer	App URLs and HTTPS	Wildcard DNS record to the load balancer IP. Certificates from the enterprise CA or cert-manager (VKS add-on)

Network Flows

- **User traffic:** users resolve the app URL through wildcard DNS to the load balancer IP. Envoy receives the request, and the Gateway sends it to the HTTPRoute of that environment and on to the app pod.
- **Control traffic:** the Agent opens an outbound HTTPS connection to the Mendix Platform and keeps it open. Nothing connects into the cluster from Mendix.
- **Image traffic:** the build job pulls Mendix base images from the Mendix image registry and pushes the app image to the container registry. The nodes pull the app image from there.
- **App data:** app pods connect to the database and the object storage over the cluster or data center network. App data never passes through Mendix.

Design Considerations

- **Pod Security:** VKS clusters enforce the baseline Pod Security level by default, and the Mendix pods run under it. Under the stricter restricted level, Operator 2.28.1 pods only lack a seccomp profile setting. Label the Mendix namespace with warn and audit for restricted to track this.
- **TLS:** the validation used HTTP. For production, add an HTTPS listener with a wildcard certificate to the Gateway and switch on TLS in the Mendix ingress settings.
- **Outbound access:** Connected mode needs outbound access to Mendix, directly or through a proxy. Clusters without outbound access can use Standalone mode with a private copy of the Mendix images.
- **Data services:** run the database and object storage outside the cluster, or as managed services with backups. In-cluster instances suit test environments.
- **Environment types:** switch on development, test, acceptance and production (DTAP) mode for the namespace in the Portal. Without it, every environment runs in production mode.

VKS Cluster Configuration

The validation cluster had one control plane node and two worker nodes. Persistent volumes came from the default vSphere CSI storage class.

Each worker had about 4 vCPU and 28 GiB of memory.

Table 3: Kubernetes cluster configuration summary

Component	Configuration	Notes
Kubernetes control plane	1 replica	Lab size. Use 3 replicas for production
Kubernetes worker nodes	2 replicas About 4 vCPU, 28 GiB RAM each amd64	Size for the number of app environments
Kubernetes release	v1.35.2+vmware.1	Within the Mendix range of 1.19 to 1.36
OS image	VMware Photon OS 5	Kernel 6.1, containerd 2.2.1

Component	Configuration	Notes
Storage class	wcp-storage	Default class, vSphere CSI, ReadWriteOnce
Pod Security default	baseline enforced	Cluster-wide default

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 4: Kubernetes cluster packages

Package	Version	Notes
Gateway API	1.4.0	VKS core package
cert-manager	1.19.4	VKS add-on
Contour	1.33.2+vmware.1-vks.1	VKS add-on, installed for this solution
Mendix Operator	2.28.1	Mendix, installed with mxpc-cli
Mendix Agent	2.28.1	Mendix, Connected mode
Mendix runtime	11.15.0	Runtime version of the test app
Relational database	PostgreSQL 17	Customer-supplied. In-cluster for validation
S3-compatible object storage	n/a	Customer-supplied. In-cluster test instance for validation
Container registry	n/a	Customer-supplied. External OCI registry for validation

Solution Validation

Install Mendix on VKS

The steps below show the main commands. Values in angle brackets are placeholders. The commands follow the Mendix non-interactive installation procedure.

Step 1: Install Contour and a Gateway

Install the Contour add-on through the Supervisor. On vSphere the Envoy service defaults to NodePort, so set it to LoadBalancer to get an external IP.

contour-values.yaml

```
namespace: tanzu-system-ingress
contour:
  configFileContents:
    gateway:
      gatewayRef:
        name: contour
        namespace: tanzu-system-ingress
envoy:
  service:
    type: LoadBalancer
```

bash

```
vcf addon install create contour --cluster-name <cluster> -n <vsphere-namespace> \
  -v 1.33.2+vmware.1-vks.1 -f contour-values.yaml -y
kubectl get svc envoy -n tanzu-system-ingress    # EXTERNAL-IP is the load balancer IP
```

Create the GatewayClass and a Gateway. The Gateway must accept routes from other namespaces, so the Mendix routes can attach to it.

gateway.yaml (excerpt)

```
apiVersion: gateway.networking.k8s.io/v1
kind: Gateway
metadata:
  name: contour
  namespace: tanzu-system-ingress
spec:
  gatewayClassName: contour
  listeners:
    - name: http
      protocol: HTTP
      port: 80
      allowedRoutes:
        namespaces:
          from: All
```

Step 2: Register the cluster in the Mendix Portal

In the Mendix Portal, register the cluster and add a namespace in Connected mode. The Portal shows an install command that contains the namespace ID and a secret. Treat that command as a credential.

Step 3: Install the Mendix Operator and Agent

bash

```
kubectl create ns mendix
kubectl label ns mendix \
  pod-security.kubernetes.io/warn=restricted \
  pod-security.kubernetes.io/audit=restricted
./mxpc-cli base-install -n mendix -i <namespace-id> -s <secret> -t generic -m connected
```

Step 4: Configure registry, plans and routing

Configure the namespace with the installer screens, or apply a configuration file. The excerpt shows the routing and registry sections. The database and storage plans in the same file point at the customer database and object storage.

mx_config_cli.yaml (excerpt)

```
ingress:
  type: gateway-route
  enable_tls: false
  gateway_route:
    domain: <apps-domain>
    group: gateway.networking.k8s.io
    kind: Gateway
    name: contour
    namespace: tanzu-system-ingress
registry:
  type: generic
  generic_registry:
    auth_pull_url: <registry-host>
    auth_push_url: <registry-host>
    registry_name: <project>/<repository>
    enable_auth: true
    link_secret_to_service_account: true
```

bash

```
./mxpc-cli apply-config -i <namespace-id> -s <secret> -f mx_config_cli.yaml
```

The configuration file holds credentials. Store it with owner-only permissions and keep it out of version control.

Validate Mendix on VKS

Deploy an app from the Mendix Portal, then check the cluster side:

bash

```
kubectl get pods -n mendix                    # operator, agent and app pods Running
kubectl logs -n mendix deploy/mendix-agent | grep 'Websocket accepting cluster requests'
kubectl get httproute -n mendix               # one route per environment
curl -sI http://<env>.<apps-domain>/ | head -1    # HTTP/1.1 200 OK
```

In the Portal, the environment shows as running. The validation app ran on both the ephemeral plans and the database and storage plans.

Conclusion

Mendix runs on VKS with standard VKS building blocks. VKS provides the cluster, load balancing, persistent storage and Gateway API routing through the Contour add-on. Mendix provides the Operator and the Agent, which build and run the apps and connect the cluster to the Mendix Platform over an outbound link.

The customer adds a relational database, S3-compatible object storage and a container registry from products it already runs. Customers run Mendix apps on their VCF private cloud and keep app data inside their own environment, while developers keep the Mendix deployment workflow.

Resources

- Mendix
- Siemens Intelligence Center X
- Mendix on Kubernetes documentation
- Supported providers and versions
- Creating a Mendix on Kubernetes cluster: Agent and Operator installation
- Technical appendix: Introduction to Operators and Operator Flows
- Non-interactive installation with mxpc-cli
- Network ingress settings: Ingress and Gateway API
- Registry configuration
- Storage plans: database and object storage
- Deploying an app from the Mendix Portal
- Standalone mode: deploying with the command line



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; Oct-26