TECHNICAL WHITE PAPER: **June 2026**

Automic Automation on vSphere Kubernetes Service on VMware Cloud Foundation

Reference Architecture

Table of Contents

Executive Summary

As enterprise infrastructure continues to evolve towards private cloud and Kubernetes-based operating models, workload automation platforms must evolve from traditional application deployments into cloud-native services. This document details the successful validation and deployment of Automic Automation Kubernetes Edition (AAKE) version 26.0.0 on VMware Kubernetes Service (VKS). This validation proves that the AAKE operator and core engine framework are fully compatible with a native vSphere Supervisor Namespace environment, bridging the gap between traditional workload automation and modern containerized infrastructure.

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.

Benefits of using VKS

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.

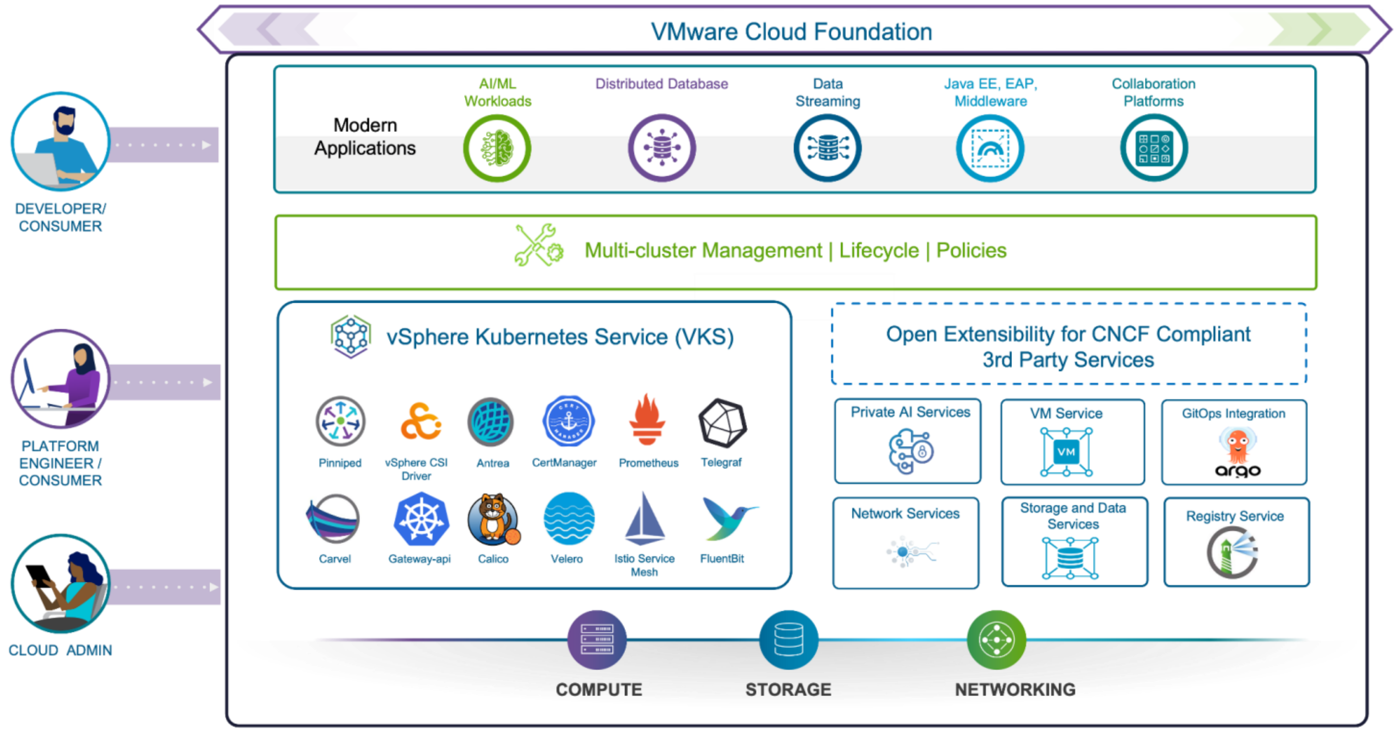

Figure 1: VKS on VCF Ecosystem

Automic Automation

Automic Automation v26 is an enterprise-wide automation control plane designed to govern the entire landscape, from legacy workloads to modern Kubernetes-native applications. Validating Automic Automation v26 on vSphere Kubernetes Service (VKS) ensures a secure, containerized foundation capable of hosting next-generation automation architecture. This includes supporting native Model Context Protocol (MCP) Client and Server capabilities to safely govern and orchestrate data flows between enterprise automation workflows and localized or private AI models. It empowers organizations to maximize the potential of every agent with orchestration across agents and tools, cross-ecosystem governance, and full transparency into performance. By natively integrating with vSphere Kubernetes Service (VKS), Automic enables platform engineers to treat workload automation as a first-class, cloud-managed service of business critical automation processes.

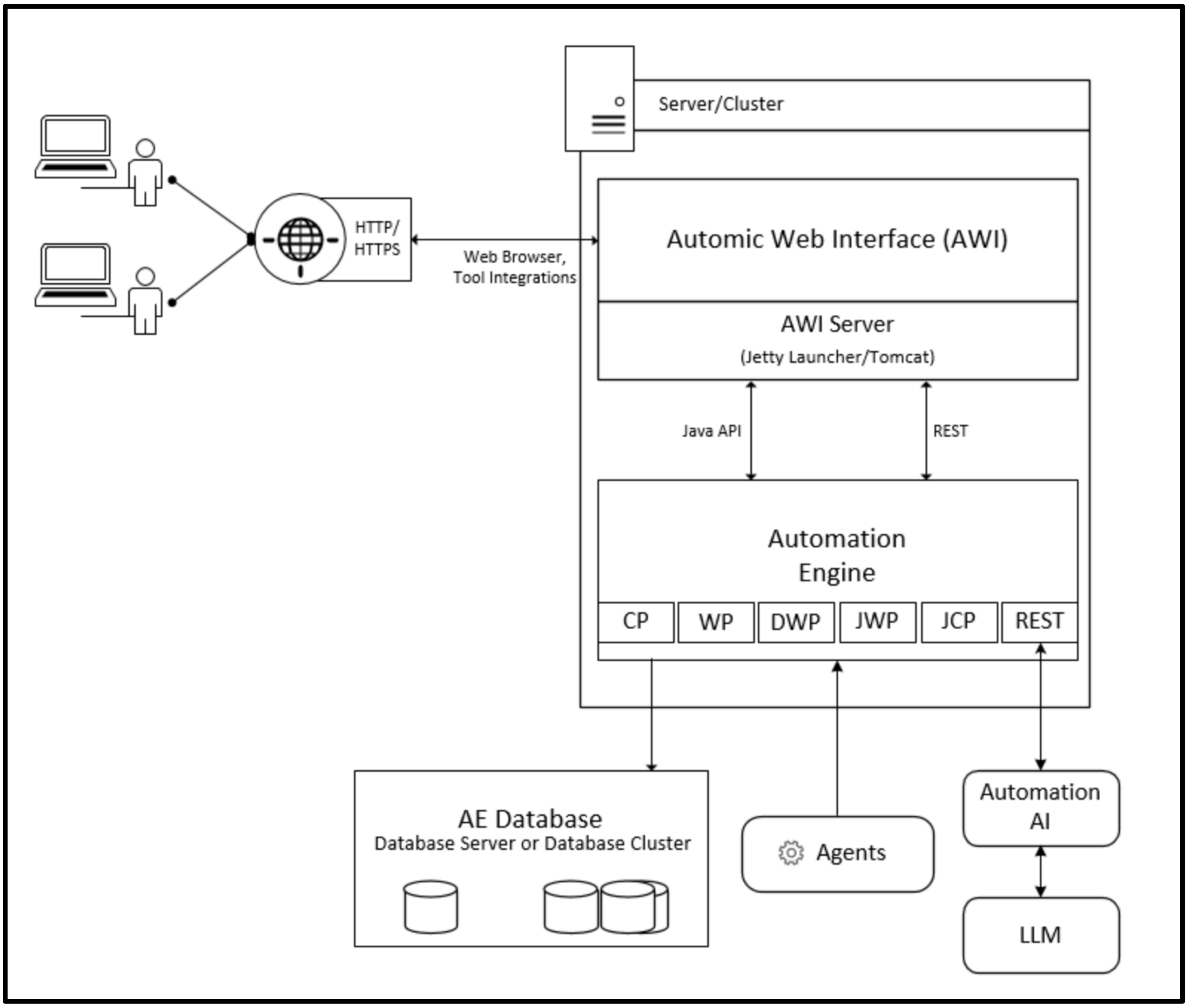

Figure 2: Automic Automation: The Intelligent Control Plane

Solution Architecture

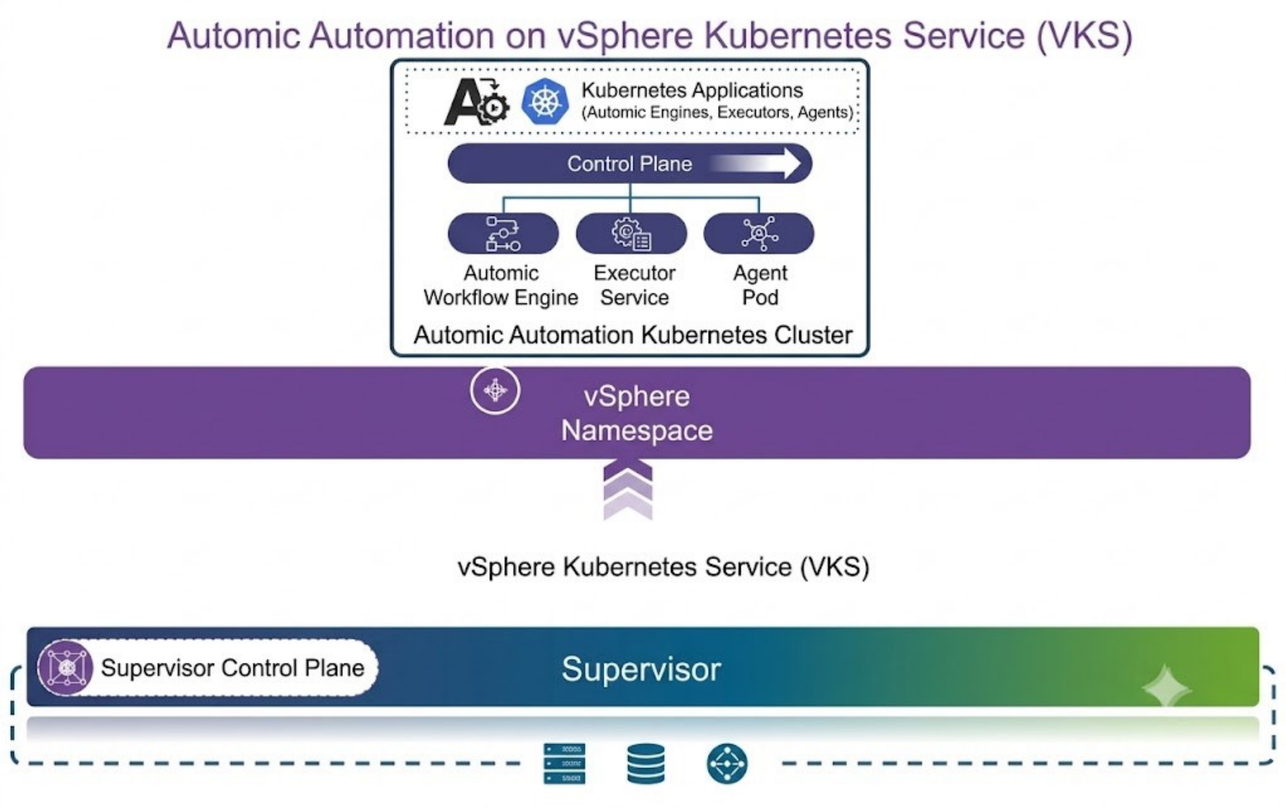

Figure 3: Automic Automation on VKS

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 3.5.

Table 1: Installed Supervisor Components

Component	Version	Notes
vSphere Kubernetes Service	3.5	Allows for CNCF K8s clusters

The standard delivery mechanism for Automic Automation on VKS is the VCF Automation Blueprint. This Blueprint orchestrates the deployment lifecycle, consuming certified Helm charts and configuration schemas from the supporting repository: https://github.com/coz-git-repo/automic. This repository serves as the single Source of Truth for the automation logic, ensuring consistency across all deployments.

VKS Cluster Configuration

The cluster comprised three worker nodes. For the control plane, this validation utilized a single node setup to demonstrate core functionality; however, it is recommended to deploy three control plane replicas for production environments to ensure High Availability (HA). Storage was provided by vSAN Express Storage Architecture (ESA). Transparently, the vSphere CSI driver exposed this policy to an adjacent Kubernetes storage class, *vsan-default-storage-policy*.

Worker node resources were configured using a *best-effort-large* profile, while control plane nodes utilized the guaranteed medium profile.

Here you can find the sizing guide.

Table 2: Kubernetes Cluster Configuration Summary

Component	Configuration	Notes
Kubernetes Control Plane	1 Replica (Validation) / 3 Replicas (Recommended for HA)	VM Class: “best-effort-medium”
Kubernetes Worker Nodes	3 Replicas — 4x vCPU 16GB RAM	VM Class: “best-effort-large” Storage Class: “vsan-default-storage-policy”
Kubernetes Release	v1.35.0---vmware.2-vkr.4	
OS Image	vmwarePhoton64Guest	Kernel version 6.12.60-18.ph5

The VKS cluster configuration is defined declaratively. The following snippet highlights the key configuration parameters; the complete cluster manifest is available in the accompanying repository at https://github.com/coz-git-repo/automic.

VKS Cluster Yaml

```
topology:
  class: builtin-generic-v3.3.0
  version: v1.35.0---vmware.2-vkr.4
  variables:
    - name: vmClass
      value: best-effort-medium
    - name: storageClass
      value: vsan-default-storage-policy
    - name: node
      value:
        labels:
          Environment: isv-lab
          Purpose: automic-aake
  workers:
    machineDeployments:
      - class: node-pool
        replicas: 3
        variables:
          overrides:
            - name: vmClass
              value: best-effort-large
```

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	2.4.4	VKS core package
Gateway API	1.3.0	VKS core package
Guest-cluster auth-service	1.4.4	VKS core package
Metrics Server	0.8.0	VKS core package
Pinniped	0.41.0	VKS core package
Secretgen Controller	0.19.2	VKS core package
vSphere CPI	1.34.0	VKS core package
vSphere CSI	3.6.0	VKS core package
Cert-Manager	1.18.2	VKS add-on
Automic Automation	26.0.0	USAGE/PURPOSE

Solution Validation

Install VMware Automic Automation on Kubernetes

Configuration Highlights

The following table summarizes the key configuration parameters optimized for the Automic Automation Kubernetes Edition (AAKE) deployment on VKS.

Table 4: Configuration Parameters

Parameter	Configuration	Purpose
AUTOMIC_GLOBAL_SYSTEM	AAKEV26	System identifier
spec.version	26.0.0	Automic version
repository	Internal Harbor Registry	Secured image management
storageClass	vsan-default-storage-policy	vSAN ESA integration
awiReplicas	1	Frontend scalability
wpReplicas	8	Workload processing
zdu.enabled	True	Zero Downtime Upgrade mode

Deployment & Reference Implementation

The installation is orchestrated through the VCF Automation Blueprint, which automates the deployment lifecycle. For transparency, validation, or troubleshooting, the underlying reference implementation steps are provided below.

Pre-Flight Requirements:

Before running the commands below, ensure your environment is ready:

1. **Verify Context:** Run `kubectl config current-context` and ensure you are pointing to `aake-vks-cluster-01`.
2. **Install Tools:** Ensure `kubectl` and `helm` are installed on your workstation.
3. Add Repositories:

```
helm repo add bitnami https://charts.bitnami.com/bitnami

helm repo update
```

4. **Prepare Assets (for Reference Implementation):** To facilitate manual verification or troubleshooting of the underlying automation logic, download the `automic-automation-6.0.0.tgz` file and clone the accompanying repository: https://github.com/coz-git-repo/automic.

Phase 1: Namespace & Secrets

Establish the 'automic' namespace and configure the Harbor registry pull secret.

```
kubectl create namespace automic

kubectl create secret docker-registry automic-image-pull-secret \

 --docker-server=lvn-sc-harbor01.showcase.tmm.broadcom.lab \

 --docker-username="robot$automic+automic-pull-robot" \

 --docker-password="<PASTE_TOKEN_HERE>" \

 -n automic
```

Phase 2: Database Provisioning

Deploy the PostgreSQL instance using the Bitnami Helm chart. Ensure the password is updated according to your organization's security standards.

```
helm install aake-db bitnami/postgresql --namespace automic \

 --set global.postgresql.auth.database=ae \

 --set global.postgresql.auth.username=automic \

 --set global.postgresql.auth.password=**** \

 --set primary.persistence.size=20Gi \
```

```
  --set primary.persistence.storageClass=vsan-default-storage-policy \
  --set primary.extendedConfiguration="vacuum_cost_limit = 10000"
```

Create the required database secret for the Automation Engine:

```
kubectl create secret generic ae-db --namespace automic \
  --from-literal=host=aake-db-postgresql.automic.svc.cluster.local \
  --from-literal=vendor=postgres --from-literal=port='5432' \
  --from-literal=user=automic --from-literal=db=ae \
  --from-literal=password=**** \
  --from-literal=data-tablespace-name=pg_default \
  --from-literal=index-tablespace-name=pg_default \
  --from-literal=additional-parameters="connect_timeout=10 client_encoding=UTF-8"
```

Phase 3: Operator Installation

Apply the AAKE Helm chart using the configuration file defined in the project repository.

```
helm install aake-v26 automic-automation-6.0.0.tgz \
  -f ./automic-automation/values.yaml -n automic
```

Phase 4: Validation

Execute helm and kubectl commands to confirm successful operator reconciliation and service readiness.

Monitor the installation logs to confirm successful reconciliation of the AAKE components:

```
kubectl logs -l app=install-operator -n automic --tail=50
```

Validate Automic Automation on Kubernetes

Deployment Verification

To confirm the environment is ready and meets the certification standards, the following diagnostic commands were executed against the 'aake-vks-cluster-01' context.

1. Cluster Context

```
kubectl config current-context aake-vks-cluster-01
```

2. Namespace Verification

```
kubectl get namespaces
NAME     STATUS   AGE
default   Active   ...
automic   Active   ...
```

3. Pod Status (namespace: automic)

```
kubectl get pods -n automic
NAME                          READY   STATUS    RESTARTS   AGE
aake-db-postgresql-0              1/1     Running   0        ...
awi-deployment-xxxx               1/1     Running   0        ...
jcp-deployment-xxxx               1/1     Running   0        ...
jwp-deployment-xxxx               1/1     Running   0        ...
wip-deployment-xxxx               1/1     Running   0        ...
install-operator-xxxx             1/1     Running   0        ...
```

4. Service Verification

```
kubectl get svc -n automic
NAME         TYPE       CLUSTER-IP     EXTERNAL-IP   PORT(S)
aake-db       ClusterIP   <Cluster-IP>    <none>      5432/TCP
awi          ClusterIP   <Cluster-IP>    <none>      8080/TCP
jcp          ClusterIP   <Cluster-IP>    <none>      8080/TCP
operator      ClusterIP   <Cluster-IP>    <none>      443/TCP
```

5. Storage (PV/PVC) Verification

```
kubectl get pvc -n automic

NAME          STATUS   VOLUME   CAPACITY   ACCESS MODES   STORAGECLASS   AGE

pvc-aake-data    Bound    ...     ...      RWO          vsan-default   ...
```

6. Node Status

```
kubectl get nodes -o wide

NAME         STATUS   ROLES          AGE   VERSION

cp-node-01    Ready   control-plane   ...   v1.35.0

worker-01     Ready   <none>          ...   v1.35.0

worker-02     Ready   <none>          ...   v1.35.0

worker-03     Ready   <none>          ...   v1.35.0
```

Lifecycle Management

Patching: Infrastructure (OS) patching is handled automatically by the VKS/Supervisor layer. Application-level upgrades are managed via Blueprint using Zero-Downtime Upgrade (ZDU) capabilities.

Scaling: Horizontal Pod Autoscaling (HPA) is natively configured within the Blueprint and managed by Kubernetes based on resource utilization.

Backups: Standard vSAN snapshots for persistent volumes are recommended as the primary strategy for application state protection and recovery.

Troubleshooting:

- **If pods aren't starting:** Run `kubectl describe pod <pod-name> -n automic` to see if there are image pull errors or resource constraints.
- **If Helm fails:** Ensure you are in the directory where the `.tgz` file is located and all Helm repositories are added.

Conclusion

This technical validation confirms that Automic Automation (AAKE) version 26.0.0 is fully operational and supported on VMware Kubernetes Service (VKS), demonstrating a breakthrough in the Automic operator and core engine framework's native integration within vSphere Supervisor Namespaces.

Resources

– Automic Learning Paths: https://academy.broadcom.com/learning-paths#automic-automation

– Automic Marketplace: https://marketplace.automic.com/marketplace/browse?search=&products=1565173278634

Item No: vmw-bc-wp-tech-uslet-word-2026; May-26