TECHNICAL WHITE PAPER: **October 2026**



SAS Viya on vSphere Kubernetes Service on VMware Cloud Foundation

Reference Architecture

Table of Contents

Executive Summary 3
VMware vSphere Kubernetes Service 3
SAS Viya 4
SAS Viya validation on VKS: Architecture 6
Kubernetes Workload Cluster Configuration 7
Kubernetes Packages 9
Solution Validation 9
Designing SAS Viya for the VKS Validation 9
Start with a supported Kubernetes foundation: 9
Place workloads intentionally: 9
Use more than one type of storage: 10
Treat networking and ingress as part of the platform: 10
Build security and operations into the design: 10
Size for the test, design for growth: 11
Availability is a layered decision: 11
Install SAS Viya on VKS 11
Validate SAS Viya on VKS 11
Conclusion 13
Reference 13
About the Authors 14

Executive Summary

Organizations increasingly rely on data and AI platforms to drive decisions, detect fraud, forecast outcomes, and personalize customer experiences. SAS Viya is a cloud-native Data and AI platform delivered as a collection of containers that run on Kubernetes, making the choice of Kubernetes foundation critical to performance, scalability, and operational efficiency.

VMware 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.

This reference architecture describes how to deploy and validate SAS Viya on VKS running on VCF. It covers the Kubernetes foundation, workload placement strategy, storage design, networking and ingress, security and operations, sizing, and availability. A validation environment was built using a VKS cluster with VMware® vSAN™ Express Storage Architecture (ESA) for persistent storage, and SAS Viya was deployed and tested with SAS Performance Suite workloads.

The validation confirmed that SAS Viya runs on VKS with the expected platform services and that the vSAN-backed storage delivers the throughput that SAS analytics workloads require. The architecture treats compute capacity and temporary storage as separate design decisions, combining high-throughput SASWORK storage with compute nodes sized for CPU concurrency. This reference architecture provides a validated starting point for organizations deploying SAS Viya on VKS.

VMware 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.

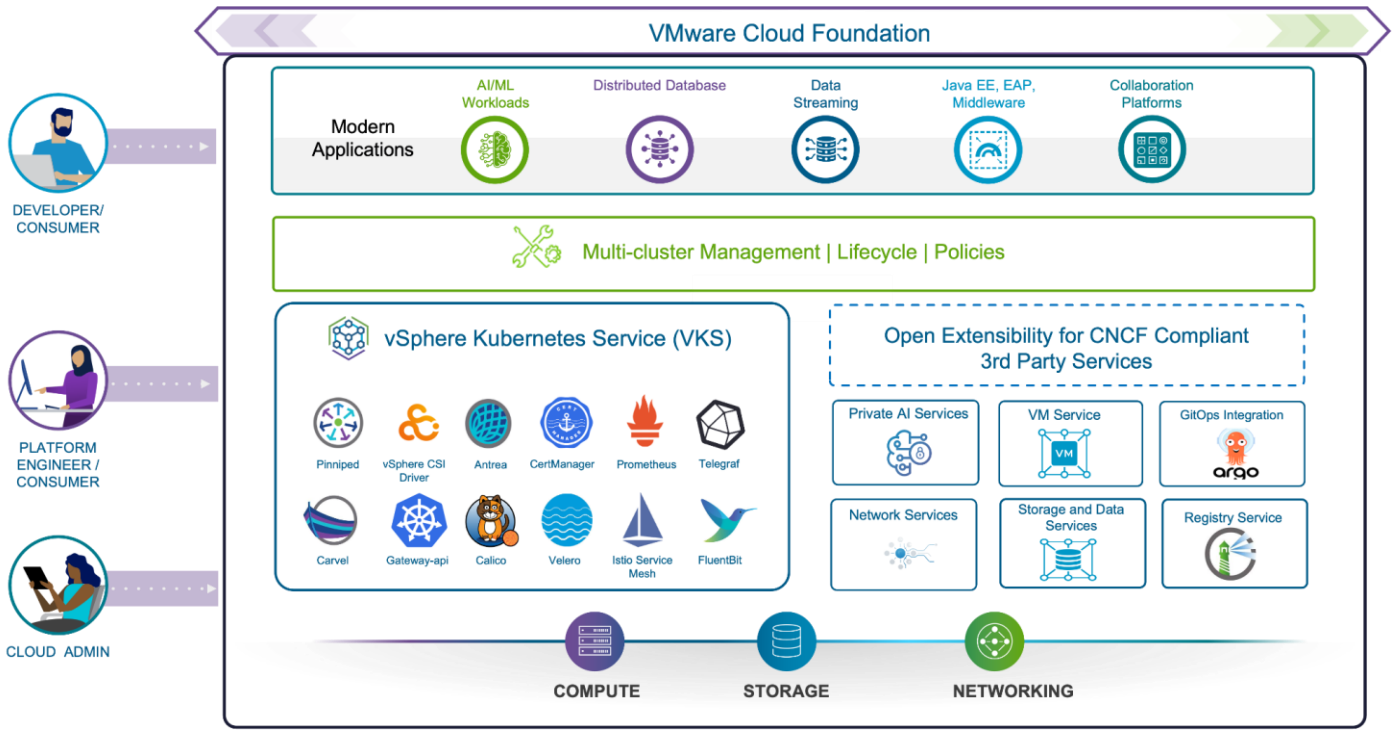


Figure 1: VKS on VCF Ecosystem

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.

SAS Viya

SAS Viya is an enterprise Data and AI platform that helps organizations turn data into better business decisions. It brings together data management, analytics, artificial intelligence, reporting, and decision automation in a single environment. Organizations use SAS Viya to identify opportunities, reduce risk, improve operational efficiency, detect fraud, forecast future outcomes, and deliver personalized customer

experiences. The platform helps both business users and data scientists collaborate using visual tools, AI assistance, and advanced analytics. SAS Viya also provides governance and transparency, ensuring that AI-driven decisions are explainable, compliant, and trustworthy. It can be deployed in the cloud, on-premises, or in hybrid environments, allowing organizations to scale analytics across the enterprise

SAS Viya is a cloud-native Data and AI platform that is designed to run on Kubernetes. Unlike traditional analytics software that is installed on individual servers, Viya is delivered as a collection of containers that run across a Kubernetes cluster. This architecture provides scalability, resilience, automated recovery, and the ability to efficiently use cloud resources.

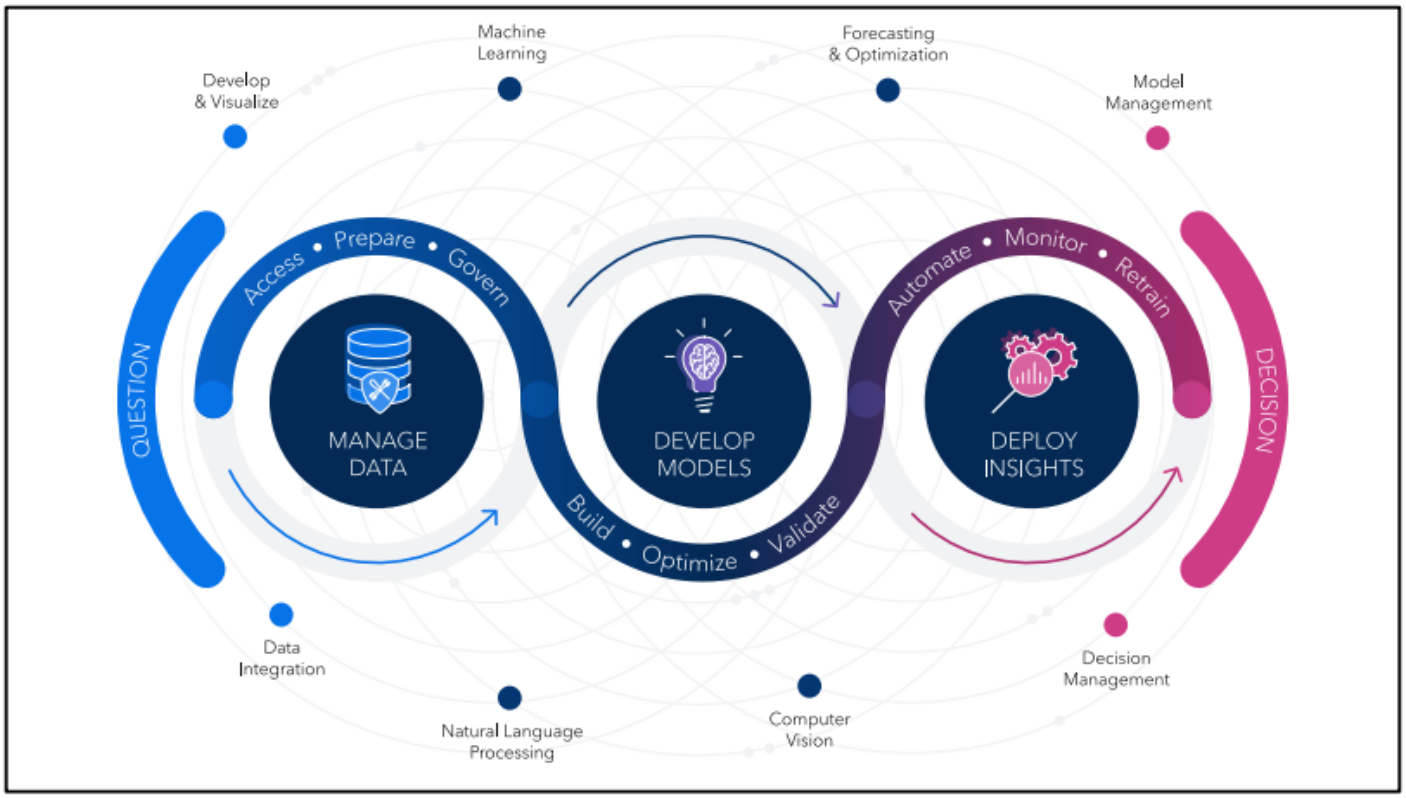


Figure 2: SAS Viya Platform Architecture

SAS Viya validation on VKS: Architecture

A SAS Viya deployment generally requires:

- A supported Kubernetes cluster
- A dedicated namespace for SAS Viya
- Persistent storage (RWO and RWX storage classes)
- DNS and TLS certificates
- Multiple worker nodes for compute workloads
- Shared storage for platform services
- Sufficient CPU, memory, and disk capacity based on workload size

Starting in May 2026, all SAS Viya supported versions can be deployed on Kubernetes distributions that are certified under the Cloud Native Computing Foundation (CNCF) Kubernetes Software Conformance program, provided that the environment complies with all system requirements described in the SAS Viya Operations guide in the Kubernetes Cluster Requirements page. For more information on Kubernetes Cluster Requirements visit Platform-Specific Requirements.

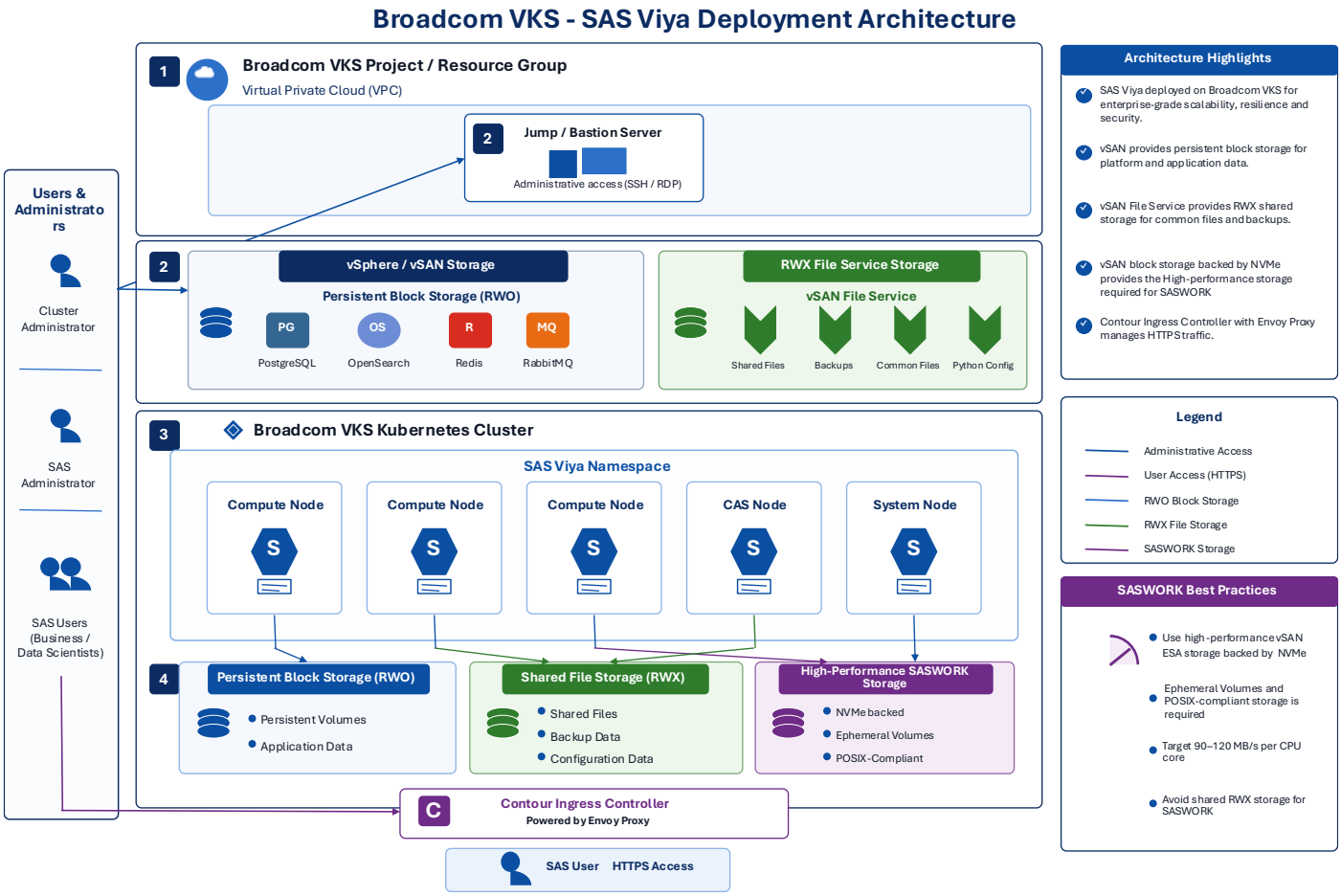


Figure 3: VKS Validation on SAS Viya

Defining a workload placement strategy is a key step in designing the Viya architecture. SAS recommends separate node pools for stateful, stateless, CAS (Cloud Analytic Services), and compute workloads. For this test, only CAS and compute node pools were used. Although both workloads can scale horizontally, in the lab used a VKS cluster with one control-plane node and four worker nodes: three labelled for SAS Compute and one for SAS CAS.

```
holuser@console:~/deployments/viya4$ kubectl get nodes -L workload.sas.com/class
NAME                                  STATUS   ROLES           AGE   VERSION            CLASS
sasviya-cvx4z-s244q                   Ready    control-plane   29d   v1.35.5+vmware.1
sasviya-node-pool-1-r8rz5-nbmz4-87kkv Ready    <none>          29d   v1.35.5+vmware.1   compute
sasviya-node-pool-1-r8rz5-nbmz4-cb4pm Ready    <none>          29d   v1.35.5+vmware.1   compute
sasviya-node-pool-1-r8rz5-nbmz4-ckkbq Ready    <none>          29d   v1.35.5+vmware.1   compute
sasviya-node-pool-1-r8rz5-nbmz4-kfswz Ready    <none>          29d   v1.35.5+vmware.1   cas
```

Figure 4: Workload Kubernetes Cluster nodes for SAS Viya deployment

Table 1: SAS Viya Version

Component	Version	Notes
SAS Viya	LTS 2026.03	

Kubernetes Workload Cluster Configuration

The cluster consisted of one control-plane node and four worker nodes. Storage was provided by vSAN ESA, using the default RAID 5 storage policy for application Read Write Once (RWO) and vSAN File Services for application ReadWriteMany (RWX) requirement. Transparently, the VMware vSphere CSI driver exposed this policy to an adjacent Kubernetes storage class, **vsan-esa-default-policy-raid5**. For more information on using storage classes see Using Storage Classes for Persistent Volumes

Worker node resources were configured using a ***best-effort-4xlarge*** profile, whilst control plane nodes utilized the ***guaranteed-large*** profile.

Table 2: Kubernetes Version and Cluster Configuration Summary

Component	Software Version or Configuration	Notes
VKS version	3.6	vSphere Kubernetes Service
Kubernetes	v1.35.5	vSphere Kubernetes Release (VKr)
OS Image	VMware Photon OS/Linux	KERNEL-VERSION 6.1.170-1.ph5
Kubernetes Control Plane	1 Replicas — 4x vCPU 16GB RAM vSAN RAID 5	VM Class: “guaranteed-large” Storage Class: “vsan-esa-default-policy-raid5” Note: For high availability in Production, you may want to use 3 Replicas and higher CPU / RAM configuration

Kubernetes Worker Nodes	4 Replicas — 16 vCPU 128GB RAM vSAN RAID 5 vSAN File Service	VM Class: "best-effort-4xlarge" Note: For Production use case you may want to use fully reserved VM Class type Example "guaranteed-4xlarge" Storage Class: "vsan-esa-default-policy-raid5" (Read Write Once (RWO) vSAN Block Storage) Storage Class: "vsan-fs-rwx" (ReadWriteMany (RWX) volume support by enabling vSAN File Service)
Kubernetes	v1.35.5	vSphere Kubernetes Release (VKr)
OS Image	VMware Photon OS/Linux	KERNEL-VERSION 6.1.170-1.ph5

At its core, the VKS 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, see vSphere Supervisor 9.0 Services and Standalone Components

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.

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/v1beta2
kind: Cluster
metadata:
  name: sasviya
  namespace: sasviya
spec:
  clusterNetwork:
    services:
      cidrBlocks: ["10.96.0.0/12"]
    pods:
      cidrBlocks: ["192.168.0.0/16"]
    serviceDomain: "cluster.local"
  topology:
    classRef:
      name: builtin-generic-v3.6.0
    version: v1.35
    controlPlane:
      replicas: 1
    workers:
      machineDeployments:
        - class: node-pool
          name: node-pool-1
          replicas: 4
          variables:
            overrides:
            - name: vmClass
              value: best-effort-4xlarge
            - name: volumes
              value:
```

```
            - capacity: 100Gi
              mountPath: /var/lib/containerd
              name: containerd
              storageClass: vsan-esa-default-policy-raid5
  variables:
    - name: vmClass
      value: guaranteed-large
    - name: storageClass
      value: vsan-esa-default-policy-raid5
    - name: vsphereOptions
      value:
        persistentVolumes:
          defaultStorageClass: vsan-esa-default-policy-raid5
    - name: kubernetes
      value:
        security:
          podSecurityStandard:
            enforce: baseline
```

Kubernetes Packages

VKS clusters (via the VKr release) include a default set of core packages as mentioned here. In this validation Contour was used.

Table 3: Kubernetes Cluster Packages installed

Package	Version	Notes
Contour	contour.kubernetes.vmware.com.1.33.1+vmware.1-vks.1	VKS core package

Solution Validation

Designing SAS Viya for the VKS Validation

A successful SAS Viya deployment begins well before the first container starts. The architecture must connect Kubernetes platform choices with the way analytics workloads consume CPU, memory, network, and storage. For the VKS validation, the design focused on a compact but deliberate topology that could isolate the two most performance-sensitive workload classes: SAS Compute and CAS.

Start with a supported Kubernetes foundation:

SAS Viya is designed to run as a collection of containers in a Kubernetes namespace. The target cluster therefore needs to satisfy the SAS Viya Kubernetes system requirements, provide enough capacity for the selected software offering, and expose the infrastructure services required by the platform. The broader design must also account for DNS, certificates, image access, identity services, data sources, administration tools, and other systems that interact with Viya.

For this validation, VKS supplied the CNCF-conformant Kubernetes runtime on VMware Cloud Foundation. A dedicated SAS Viya namespace kept the deployment logically separated, while VKS and Cluster API provided declarative cluster creation and lifecycle management.

Place workloads intentionally:

Kubernetes can schedule pods wherever resources are available, but that default behavior is not ideal for resource-intensive analytics. SAS Viya architecture therefore starts with a workload placement strategy. Node labels, selectors, taints, tolerations, and affinity rules can be used to keep demanding workloads predictable and to reduce contention between platform services and analytic engines.

The test topology used three worker nodes labelled for SAS Compute and one worker node labelled for SAS CAS. This was a pragmatic validation layout rather than a full production topology. In a production design, stateful, stateless, compute, CAS, and system or monitoring workloads should be evaluated separately, with node pools sized according to workload, availability, and operational requirements.

Use more than one type of storage:

SAS Viya does not have a single storage profile. The architecture must combine persistent, shared, and temporary storage according to the workload:

- RWO persistent volumes for stateful platform components such as PostgreSQL, Redis, RabbitMQ, Consul, and OpenSearch.
- RWX persistent volumes for shared files, backup content, common configuration, Python configuration, and other services that must be mounted from more than one node.
- High-performance ephemeral storage for SASWORK and CAS_DISK_CACHE, where throughput and latency directly influence analytic processing.

SAS Technical Architect Rob Collum explains how to address these storage requirements in A node-based storage architecture for the SAS Viya platform. The article groups SAS Viya storage into persistent RWO, shared persistent RWX, and temporary storage needs.
It demonstrates how OpenEBS can use node-local disks to provide these storage types, including high-performance scratch space.
This node-based design is a practical option rather than a universal recommendation and should align with each organization's storage standards and resilience requirements.

In the VKS cluster, the vSphere CSI driver exposed vSAN ESA (Express Storage Architecture) block storage through the vsan-esa-default-policy-raid5 storage class for RWO volumes. vSAN File Service supplied RWX storage. For temporary analytic I/O workload, the preferred design is local SSD or NVMe-backed storage, or a generic ephemeral volume backed by a suitably fast storage class. In this architecture vSAN ESA backed by NVMe was used for temporary analytic I/O workload. Object storage is useful for data repositories, but it is not natively POSIX-compliant and should not be treated as a direct SASWORK file system.

Treat networking and ingress as part of the platform:

SAS Viya uses Kubernetes ClusterIP services for internal communication and ingress for external HTTP and HTTPS access. The tested architecture used Contour, powered by Envoy, to route user traffic into the SAS Viya namespace. DNS must resolve the Viya host name to the load-balancing endpoint, and TLS certificates must be planned for the chosen encryption model. If external clients require the native CAS binary protocol, an additional LoadBalancer service is needed because that traffic does not pass through HTTP ingress.

Build security and operations into the design:

The architecture must define who administers the Kubernetes platform and what delegated access is available to SAS administrators. It must also identify the identity source, authentication method, UID and GID strategy, access to external systems, certificate ownership, and secrets-management approach. Logging, monitoring, and auditing need an explicit design because container logs are collected differently from traditional server-based SAS deployments. The SAS Viya Monitoring for Kubernetes framework can be considered alongside the organization's existing observability tooling.

Size for the test, design for growth:

CPU, memory, storage throughput, network bandwidth, node count, and pod placement all affect SAS Viya performance. The validation cluster was sized to execute the selected test workload and measure the behavior of VKS and vSAN. It should not be read as a universal production sizing. Production architecture should be based on expected users, concurrency, data volumes, analytic workload, growth, and availability objectives, with enough spare capacity for updates, pod movement, and component failure.

Availability is a layered decision:

High availability depends on the infrastructure, the Kubernetes cluster, and the SAS Viya services together. A single control-plane node and a limited number of workers can be acceptable for a controlled test, but a production implementation should define recovery time and recovery point objectives, control-plane redundancy, worker-node failure tolerance, storage resilience, backup and restore, and the replica strategy for SAS Viya services. Supporting components such as ingress and certificate management must be included in the same availability review.

Install SAS Viya on VKS

For a manual deployment, first download and extract the SAS Viya deployment assets, create a site directory, and configure the base kustomization with the required overlays, transformers, storage settings, ingress, certificates, and workload placement definitions. Use **kustomize build** to render the site manifests, review the generated YAML, and apply any documented prerequisite resources and CustomResourceDefinitions in the required order. Deploy the platform to its dedicated namespace with **kubectl apply**, monitor pods, jobs, events, services, ingress, and persistent volume claims until all components are ready, and then validate the deployment and save the final site configuration for updates and recovery. Detailed procedures and release-specific commands are available in the SAS Viya deployment documentation.

Validate SAS Viya on VKS

After deploying SAS Viya using the SAS deployment tooling, validation confirmed that all platform services started correctly, that persistent volumes were bound, and that ingress routing delivered traffic to the SAS Viya namespace. The SAS Viya web interface and command-line tools were used to verify that users could authenticate, submit jobs, and retrieve results.

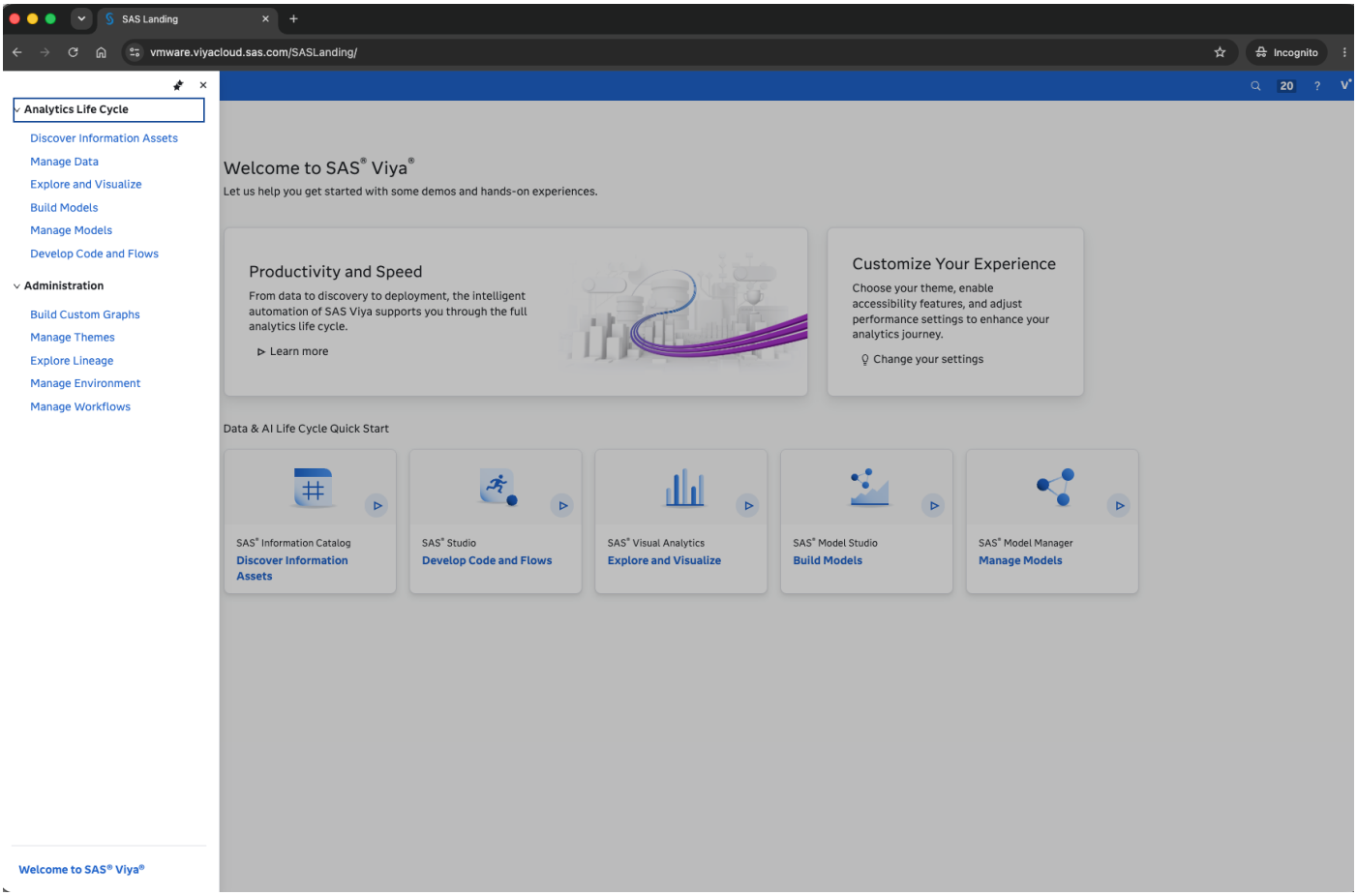


Figure 5: SAS Viya web interface showing the successful deployment

What the validation showed

The SAS Performance Suite workload was run on the VKS cluster to validate the architecture in practice. The suite contained measured steps covering DATA step processing, sorting, SQL, memory-intensive analysis, regression and logistic processing, mixed workloads, and separate SASWORK write and read tests. The purpose was to observe how the VKS compute allocation and vSAN storage behaved under this workload.

The validation confirmed that the compact VKS test topology completed the full test suite. The results exposed clear differences between CPU-bound and storage-bound work. The SASWORK steps demonstrated required storage throughput, confirming that vSAN ESA delivers the I/O performance that SAS analytics workloads require. The test reinforced the need to treat compute capacity and temporary storage as separate design decisions. For production deployment the VKS architecture should combine high-throughput SASWORK storage with compute nodes sized for the expected level of CPU concurrency. For detailed sizing recommendations for SAS Viya visit SAS Viya Sizing Recommendations.

A closer look at SASWORK

SASWORK is the temporary scratch space used by the SAS Programming Runtime Environment for:

- SORT operations
- SQL joins

- DATA step intermediate results
- UTILLOC spill files
- Temporary tables and work datasets

SASWORK must reside on fast, POSIX-compliant storage capable of sustaining the I/O patterns generated by SAS workloads. SAS documentation and training material note that SAS Compute, Batch, and Connect workloads depend on fast storage, and that SAS programs are typically more I/O bound than CPU bound.

For Kubernetes deployments, SAS identifies SASWORK as a prime candidate for:

- Generic ephemeral volumes
- Local SSD or NVMe-backed storage
- HostPath or local ephemeral storage

rather than slower shared storage tiers. In this validation vSAN ESA block storage was used for SASWORK. For all types of storage, SAS recommends sequential I/O bandwidth of 90–120 MB per second, per CPU core. The validation confirmed that the vSAN ESA-backed storage class provided the throughput needed for SASWORK operations, making it a viable storage option for SAS Viya deployments on VKS.

Conclusion

This reference architecture demonstrates that SAS Viya can be deployed and validated on VMware vSphere Kubernetes Service running on VMware Cloud Foundation. VKS provides the CNCF-certified Kubernetes runtime, and VCF provides the infrastructure services, including vSAN storage, that SAS Viya requires.

The validation confirmed that SAS Viya platform services start correctly, that ingress routing delivers traffic to the SAS Viya namespace, and that vSAN ESA storage delivers the throughput needed for SAS analytics workloads. The architecture's separation of compute capacity and temporary storage as independent design decisions allows organizations to tune each layer according to workload requirements.

Organizations adopting this reference architecture should adapt the cluster sizing, storage configuration, networking, and availability design to their specific workload, concurrency, and growth objectives. The validation cluster described here provides a starting point, not a universal production specification. Production deployments should define recovery objectives, control-plane redundancy, worker-node failure tolerance, and the replica strategy for SAS Viya services.

By combining SAS Viya's cloud-native analytics platform with VKS's integrated Kubernetes runtime on VCF, organizations can reduce operational complexity, leverage existing vSphere skills and tools, and deploy analytics workloads on a validated, supported Kubernetes foundation.

Reference

- VKS Product page
- SAS Viya Platform Operations page

About the Authors

Frederik Vandenberghe, Cloud Architect, in the Global Technology practice of SAS

Palani Murugan, Product Marketing Engineer, in the VCF Technical Marketing team of VMware Cloud Foundation Business Unit of VMware by Broadcom

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