Kubernetes Ingress Services by VMware® Avi™ Load Balancer

Elasticity, Security, Observability – Consolidated

KEY BENEFITS

- **Integrated solution**
  Consolidated services including comprehensive Load Balancer, container ingress, intrinsic security, WAF, GSLB, DNS, and IPAM in one platform
- **Operational simplicity**
  A single solution with central control and ease of troubleshooting
- Rich observability
  Real-time telemetry with application insights, end-to-end across all components
- **Plug and play with VMware Cloud Foundation (VCF)**
  Enterprise-grade, fully automated L4-L7 load balancing for Kubernetes workloads on VCF

KEY FEATURES

- **Traffic management and security** optimized for North-South traffic
- **Integration with Kubernetes** to automate deployment and management of container clusters
- **Multi-cluster, multi-site and multi-AZ** container cluster support across multiple geos and availability zones on a highly scalable platform

WHAT'S INCLUDED

A single platform that provides

- Container ingress
- L4-L7 load balancing
- On-demand application scaling
- Web application firewall (WAF)
- Global server load balancing (GSLB)

Kubernetes Ingress Needs a Scalable and Enterprise-Class Solution

Modern application architectures supporting AI applications and agentic workloads have made appliance-based load balancing solutions obsolete. Traditional appliance-based load balancers or open-source tools are not equipped to support enterprise-grade north-south ingress services, which require robust security, elastic autoscaling, integrated peripheral services and full-stack automation that are needed for a modern enterprise application architecture. This lack of a single solution results in separate products from multiple vendors to provide load balancing, ingress traffic management, DNS, IPAM and WAF services. IT faces more complex operations managing and troubleshooting multiple independent components with disparate analytics and no end-to-end visibility. These stitched solutions necessitate in depth scripting knowledge to provide only partial automation, if any at all, leading to compromises between feature, automation, and scale.

Challenges with Application Services for Kubernetes

Common application services, such as load balancing, network performance monitoring, and security, that are available in conventional applications often need to be implemented or approached differently in container- based applications. Here are some of such challenges in deploying container-based applications using multiple vendors:

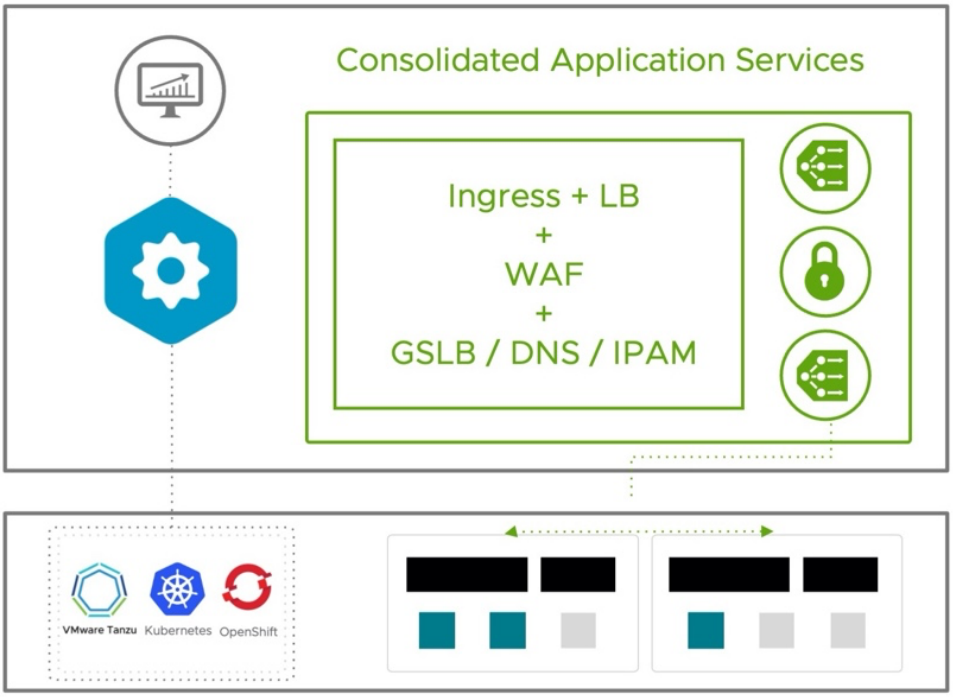


FIGURE 1: Avi Consolidated Kubernetes Services

vmware® by Broadcom

Multiple discrete solutions

Modern microservices architectures have made traditional appliance-based load balancers obsolete, as they cannot handle dynamic service-to-service traffic, autoscaling, or integrate natively with essential services like DNS, IPAM, and WAF. As a result, organizations are often forced to rely on multiple discrete solutions to fill these gaps, leading to increased operational complexity and inefficiency in managing distributed, cloud-native applications.

Complex operations

With multi -vendor solutions IT faces more complex operations in managing and troubleshooting multiple independent components from different vendors.

Lack of observability

End-to-end visibility is especially important with container-based applications. Application developers and operations teams alike need to be able to view the interactions between the peripheral services and the container services to identify erroneous interactions, security violations, and potential latencies.

Partial automation

Application and networking services need to be API-driven and programmable without the constraints of hardware appliances or multi-vendor solutions that limit their flexibility and portability across environments. Multi-vendor solutions also necessitate in depth scripting knowledge for different products to provide only partial automation, if any at all, leading to compromising between feature, automation, and scale. (see Figure 1).

Avi Overview

Avi offers a unified, software-defined platform for enterprise Kubernetes, combining load balancing, ingress, security, and observability. It provides advanced L4-L7 services like GSLB, DNS/IPAM, WAF, and analytics for consistent application delivery with a complete networking and security stack. Centralized policies and full automation simplify operations, enabling self-service and reducing manual tasks. Avi delivers real-time telemetry and deep analytics for end-to-end visibility across network, users, security, and application performance. With inbuilt automation and elastic autoscaling, Avi dynamically scales resources based on analytics, ensuring reliable, production-ready Kubernetes applications.

Kubernetes Ingress Services is based on a software-defined, distributed architecture with four major components:

Avi Controller: The Avi Controller is the central management component of the Avi architecture providing all control plane functionality of infrastructure orchestration, centralized management, and the analytics dashboard. In Kubernetes environments, the Avi Controller is in lock steps with Kubernetes master in a scalable manner. It can be deployed anywhere if connectivity and latency requirements are satisfied.

Avi Service Engines: In Kubernetes environments, the SEs are deployed external to the cluster and provide services such as LB, GSLB, analytics, DNS and WAF in the data plane.

Avi Kubernetes Operator: AKO which acts as an ingress controller is a pod running in Kubernetes clusters that provides communications with Kubernetes master. AKO remains in sync with the required Kubernetes objects and calls the Avi Controller APIs to deploy the Ingress Services via the Avi Service Engines. Avi also supports the Kubernetes Gateway API, a next-generation ingress standard that simplifies and enhances container ingress management by providing more expressive and extensible routing capabilities

Avi Multi-Kubernetes Operator: The AMKO facilitates multi-cluster application deployment extending application ingress controllers across multi-region and multiple Availability Zone deployments. AMKO calls Avi APIs for Avi Controller to create GSLB services on the leader cluster which synchronizes with all follower clusters.

Get Kubernetes Apps Production Ready with Consolidated Container Ingress Services

Avi's Consolidated Container Ingress Services enable production-ready Kubernetes applications by unifying critical capabilities—application resiliency, enterprise-grade security, DevOps automation, and deep analytics—into a single, integrated platform. (See Figure 2)

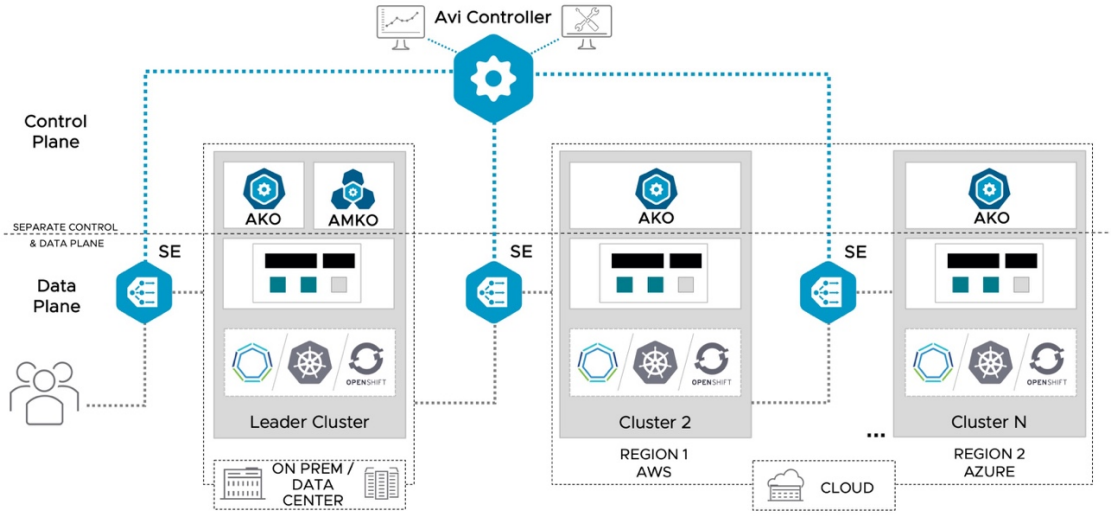


FIGURE 2: Application Services Architecture for Containers

Application Resiliency: Avi's elastic, software-defined architecture automatically scales ingress resources in response to real-time traffic patterns and analytics, ensuring that applications remain highly available and can rapidly recover from failures across clusters and regions. This dynamic scaling is managed by the Avi Controller, which orchestrates resources and maintains operational consistency regardless of the underlying environment. By deploying Service Engines externally to the Kubernetes clusters, Avi preserves cluster resources and reduces latency, further enhancing resiliency and performance.

Enterprise-Grade Security: Avi integrates robust security features such as a built-in Web Application Firewall (WAF), granular access controls, and automated policy enforcement to protect applications from threats and maintain compliance across all clusters. Security policies are centrally managed through the Avi Controller, ensuring consistent enforcement and simplifying governance in complex, distributed environments. This centralized approach allows organizations to quickly adapt to evolving security requirements while minimizing manual effort.

Comprehensive Analytics: Avi provides real-time telemetry and leverages machine learning-driven insights to deliver end-to-end visibility into application performance, user experience, and security events. The centralized analytics dashboard within the Avi Controller aggregates data from all Service Engines, enabling rapid troubleshooting, proactive optimization, and data-driven decision-making. These analytics help reduce mean time to resolution (MTTR) and ensure that applications consistently meet performance and reliability targets.

DevOps Readiness and consumption: Avi empowers DevOps teams with full lifecycle automation and self-service capabilities, allowing them to deploy, update, and manage ingress services seamlessly without manual intervention. The Avi Kubernetes Operator (AKO) runs as a pod within each Kubernetes cluster, translating Kubernetes objects into Avi configurations and automatically syncing with the Avi Controller. This automation streamlines CI/CD workflows, accelerates application delivery, and reduces operational overhead, enabling teams to focus on innovation rather than infrastructure management.

Avi supports the Kubernetes Gateway API, a next-generation ingress standard that simplifies and enhances container ingress management by providing more expressive and extensible routing capabilities. Its flexibility, extensibility, and robust feature set make it the clear path forward for organizations seeking to future-proof their Kubernetes environments while empowering DevOps teams to deliver faster, safer, and more scalable applications.

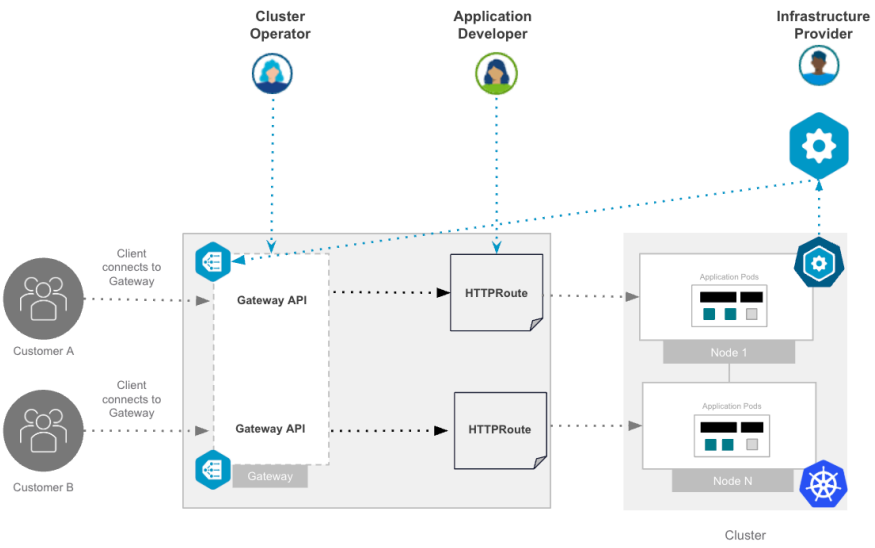


FIGURE 3: Avi Gateway API Architecture

Avi Plug and Play Integrations with VCF and Tanzu

Avi delivers enterprise-grade, fully automated L4-L7 load balancing for both VM and Kubernetes workloads including VCF with vSphere Supervisor and the Tanzu platform. Avi simplifies traffic management, enhances observability and security, and ensures consistent application delivery through a unified, software-defined architecture.

Enterprise-Grade Load Balancing for VCF with Avi

Avi is the only solution that provides comprehensive multi-tenant L4–L7 services for both Supervisor-based VM workloads and vSphere Kubernetes Service (VKS). Tight Supervisor integration lets Avi's Kubernetes Operator (AKO) auto-deploy inside the control plane; as the cluster comes online, AKO registers with the Avi Controller through REST APIs and begins configuring virtual services—no manual provisioning required.

Native support extends beyond pod networking and VMs launched through VMservice appear in the same dashboards and share the same policies, giving operators a single source of truth. AKO also runs in VKS clusters, acting as an Ingress or Gateway controller while exposing Avi's full feature set, including custom traffic-shaping rules and performance optimizations.

Because the data path sits on external Service Engines, cluster resources are preserved, latency is reduced, and WAF policies are enforced with precision. Customers report up to 30 % faster application rollouts and 20 % lower operational overhead, while mean-time-to-resolution drops thanks to Avi's real-time analytics and intent-based automation. Competing controllers lack Supervisor-native automation and unified VM/K8s visibility, making Avi the clear choice for modern, mixed-workload estates.

Unified Application Delivery for Tanzu with Avi

Modern enterprises deploying applications on Tanzu Platform for Cloud Foundry (formerly Tanzu Application Service) require resilient, scalable, and automated traffic management solutions. Avi uniquely addresses these needs by providing comprehensive Layer 4 and Layer 7 load balancing. Through native integrations and intelligent traffic management, Avi ensures consistent application availability, performance, and security across varying infrastructure layers.

In Cloud Foundry environments, Avi integrates natively via Ops Manager to provide high-performance Layer 7 load balancing for Gorouters. As Gorouter instances scale dynamically, Avi's architecture ensures that new instances are automatically added to the relevant virtual services, facilitating real-time elasticity without requiring manual intervention. Avi further enhances application security with built-in Web Application Firewall (WAF) capabilities and Access Control Lists (ACLs), giving platform teams fine-grained control over traffic policies and security postures. The integration with BOSH further automates the lifecycle of load balancing services, empowering developers and operators to focus on delivering value rather than managing infrastructure components.

Features at a Glance

Predictive Autoscaling

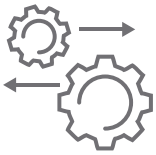
- Autoscaling of load balancers and apps based on traffic patterns
- Ability to deploy a load balancer with different capacity in real time
- Dynamically manages IPAM/DNS for discovered newly created/deleted/updated ingress controllers

Universality

- Centrally orchestrated services with load balancing, security, and analytics
- Multi-Infra: Traditional and modern apps in VMs/bare metal/containers
- Multi-Cluster: Inter/intra container cluster management and secure gateways
- Multi-Region: GSLB for multiple regions and geo-ware load balancing

Security

- Distributed Web Application Firewall (WAF) for application security
- Single sign-on (SSO) integration for enterprise-grade authentication and authorization
- Positive security model and application learning for automated acceptlist/denylist policies
- DDoS detection and mitigation for Layer4 and Layer7 attacks
- Transaction tracing & logging



Traffic Management

- Advanced Kubernetes ingress controller with integrated IPAM/DNS
- L4-7 load balancing with SSL/TLS offload
- North-south traffic management with content switching, redirection, caching, and compression
- CI/CD and application upgrades using Blue-Green or canary deployment

Observability

- Machine learning-based insights and app health analytics
- Application and infra performance metrics
- Health Monitoring of cluster connectivity and performance
- Real-time application and container performance monitoring with tracing

vmware® by Broadcom

Item No: vmw-bc-vrt-sol-brief-temp-uslet-word-2024 1/24