WHITE PAPER:
June 2025

Deliver Elastic Kubernetes Applications

A single platform for consolidated traffic management, security, and observability

vmware®
by Broadcom

Table of contents

Introduction

Infrastructure platforms have evolved from a hardware-defined internet era to a software-defined cloud era, accelerating to an AI-defined era for AI applications including agentic workloads. It has become increasingly critical for load balancing to adopt the cloud operating model which accelerates consumption with self-service and automation, elasticity with auto-scale, auto-healing and automated lifecycle management, as well as intelligence and rapid issue resolution with application latency analytics.

Kubernetes offers an excellent automated application deployment platform for container-based workloads. However, application services such as traffic management, load balancing within a cluster and across clusters/regions, monitoring/analytics, and application security are critical for modern application infrastructure. Enterprises require a scalable, real-world-tested, and robust services fabric to deploy microservices applications in Kubernetes clusters ready for production environments. This whitepaper provides an overview of the requirements for such application services and explains how the VMware Avi Load Balancer (Avi) provides a proven solution to deliver container-based workloads in production environments using Kubernetes clusters.

Challenges with Application Services for Kubernetes

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

Multiple discrete solutions

Modern application architectures based on microservices have made appliance-based load balancing solutions obsolete. Traditional hardware/virtual load balancers or open-source tools are not equipped to support the north-south ingress services, do not support application autoscaling, and lack the native integration with peripheral services such as DNS, IPAM and web application firewall (WAF).

Complex operations

With disparate 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. Multi-vendor solutions can 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

Application Services Requirements for Container-based Environments

Containerized application services require a robust suite of features and functionalities to ensure effective management, security, and performance within a container-based environment. These services are crucial for orchestrating, monitoring, and securing such applications across various platforms.

Traffic Management Local Load Balancing

Local load balancers or application delivery controllers (ADCs) need to provide application networking services such as load balancing, health monitoring, TLS/SSL offload, session persistence, content/URL switching, and content modification.

Traffic Management Global Load Balancing

Global load balancing directs clients to the appropriate site/region based on several criteria including availability, locality of the user to the site, site persistence, site load, etc.

Monitoring/Analytics

Enterprise applications deployed in production require constant monitoring and alerting based on application and network performance, health, and security.

Scalability

Infrastructure platforms are accelerating to support AI/GenAI applications including agentic workloads The scalability of the underlying load balancing solution becomes crucial to support such unpredictable workloads and ensure seamless user experiences.

Security

Enterprise-class secure applications require TLS/SSL cert management, microservice-based network security policies that control application access, DDoS protection/mitigation, and web application firewall (WAF).

Overview of VMware Avi Load Balancer

VMware Avi Load Balancer provides significant benefits for container ingress in enterprise Kubernetes environments by delivering a unified, software-defined platform that consolidates load balancing, ingress, security, and observability services.

Integrated solution

In addition to container ingress services Avi Kubernetes Ingress Services offers advanced L4-L7 services including global server load balancing (GSLB), DNS/IPAM, application security, WAF and analytics. It helps deliver applications consistently in multi-cloud environments with industry's only complete L2-L7 networking and security stack.

Operational simplicity

Centralized policies and full lifecycle automation eliminate manual tasks providing administrators with central control, self-service application delivery automation and operational consistency.

Rich observability

Real-time telemetry with application insights across all components through closed-loop analytics and deep machine learning provides holistic end-to-end insights, across the network, end users, security, and real-time application performance monitoring.

Cloud-native automation with elasticity

Elastic autoscaling based on closed-loop analytics and decision automation across on-premises data centers and public clouds, including VMware, OpenStack, AWS, Azure, and Google Cloud Platform.

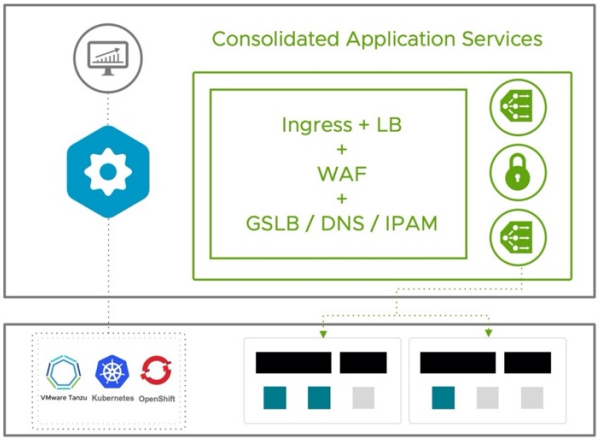


Fig 1: Avi Kubernetes Ingress Services

Avi 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 primaries 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 load balancing, GSLB, analytics, DNS and WAF in the data plane.

Avi Kubernetes Operator: Avi Kubernetes Operator (AKO) acts as an ingress controller that synchronizes Kubernetes ingress objects with the Avi Controller APIs, automating the deployment and configuration of L4-L7 load balancing services via Avi Service Engines. It runs as a pod inside each Kubernetes cluster. This integration ensures consistent, scalable, and secure ingress traffic management within individual clusters by translating Kubernetes resources into Avi virtual services and handling data plane traffic efficiently.

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

Get Kubernetes Apps Production Ready with Consolidated Container Ingress Services

Avi's Consolidated Container Ingress Services enable organizations to make their Kubernetes applications production-ready by seamlessly integrating key capabilities such as application resiliency, enterprise-grade security, DevOps automation, and comprehensive analytics into a single, unified platform. This solution ensures high availability through dynamic scaling and rapid failover, while protecting applications with built-in Web Application Firewall (WAF), access controls, and automated security policies. It accelerates DevOps workflows by providing full lifecycle automation and self-service capabilities, reducing manual effort and operational overhead. Additionally, Avi delivers real-time telemetry and actionable insights into application performance, user experience, and security events, enabling faster troubleshooting and optimization. Designed to simplify management across multi-cloud and multi-cluster Kubernetes environments, Avi empowers enterprises to securely deliver scalable, resilient, and highly available Kubernetes applications with confidence and efficiency.

Application Resiliency

Avi's elastic, software-defined architecture, combined with deep integration into Kubernetes environments and advanced Global Server Load Balancing (GSLB), significantly enhances application resiliency. GSLB intelligently distributes traffic across geographically diverse locations, ensuring high availability and disaster recovery by directing users to optimal data centers. Seamless integration with CI/CD pipelines via a 100% REST API enables automated provisioning and management of load balancing services. Avi's software-defined architecture supports elastic auto-scaling and healing, delivering high performance and low latency even during large-scale deployments and fluctuating loads.

GSLB

Avi delivers a robust GSLB capability for enterprise applications deployed across multiple data centers and private or public cloud regions. Avi's GSLB responds to DNS queries by returning the appropriate Virtual IP (VIP) address for an application, intelligently directing users to the optimal site or region. This selection leverages a combination of geo-location, site persistence, and site availability to ensure best performance and user experience.

For Kubernetes environments, the Avi Kubernetes Operator (AKO) acts as an ingress controller within each cluster, translating Kubernetes ingress and service objects into Avi configurations. To extend GSLB capabilities across multiple Kubernetes clusters, the Avi Multi-Cluster Kubernetes Operator (AMKO) works with AKO to map applications deployed in different clusters to a single GSLB service, enabling unified ingress management and cross-cluster high availability for modern, distributed applications.

AKO runs as a pod inside each Kubernetes cluster and acts as an ingress controller that synchronizes Kubernetes ingress objects with the Avi Controller APIs, automating the deployment and configuration of L4-L7 load balancing services via Avi Service Engines. This integration ensures consistent, scalable, and secure ingress traffic management within individual clusters by translating Kubernetes resources into Avi virtual services and handling data plane traffic efficiently.

To extend ingress services across multiple clusters and geographic regions, AMKO operates alongside AKO by automating Global Server Load Balancing (GSLB) configuration and DNS/IPAM management. AMKO runs as a pod in a designated leader cluster and coordinates GSLB services that tie together the virtual services created by AKO on all participating clusters. This enables intelligent global traffic distribution, failover, and disaster recovery for containerized applications deployed across multi-region or multi-cloud environments, ensuring high availability and fault tolerance at a global scale.

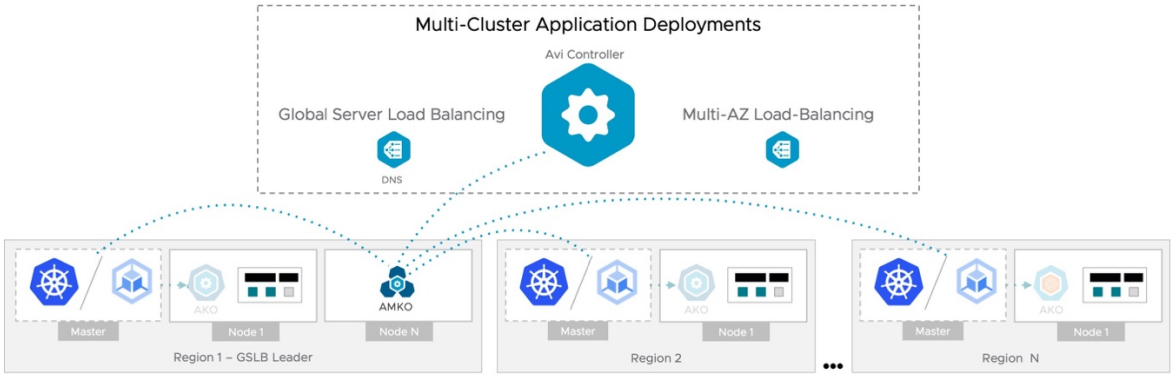


Fig 2: AMKO GSLB Service Architecture

Note: *Click here to download the Avi ingress design guide. This guide provides a brief description of all the Avi solution components and their design considerations.*

Scale and Performance

Avi ensures scale and performance for Kubernetes workloads by leveraging Avi's elastic, software-defined architecture and deep integration with Kubernetes environments. The AKO acts as the bridge between Kubernetes and the Avi Controller, translating Kubernetes ingress and service objects into Avi's advanced load balancing configurations. When application traffic increases or resource exhaustion is detected on Service Engines (SEs)—due to CPU, memory, or traffic patterns—the Avi Controller continuously monitors real-time telemetry and can automatically migrate virtual services to unused SEs or scale out services across multiple SEs. This allows multiple SEs to concurrently share the workload of a single virtual service, ensuring high availability and seamless performance even during traffic spikes

Enterprise-Grade Security for Kubernetes

Avi's Web Application Firewall (WAF) delivers enterprise-grade container security by providing a comprehensive, intelligent security stack purpose-built for modern, distributed applications—including those running in Kubernetes environments. The Avi WAF protects containerized applications from a wide range of threats, including OWASP Top 10 vulnerabilities such as SQL injection and cross-site scripting, as well as DDoS attacks, brute force attempts, and malicious bots. Leveraging a distributed application security fabric, Avi WAF enforces security through closed-loop analytics and application learning mode, enabling real-time visibility and adaptive policy creation based on observed traffic patterns.

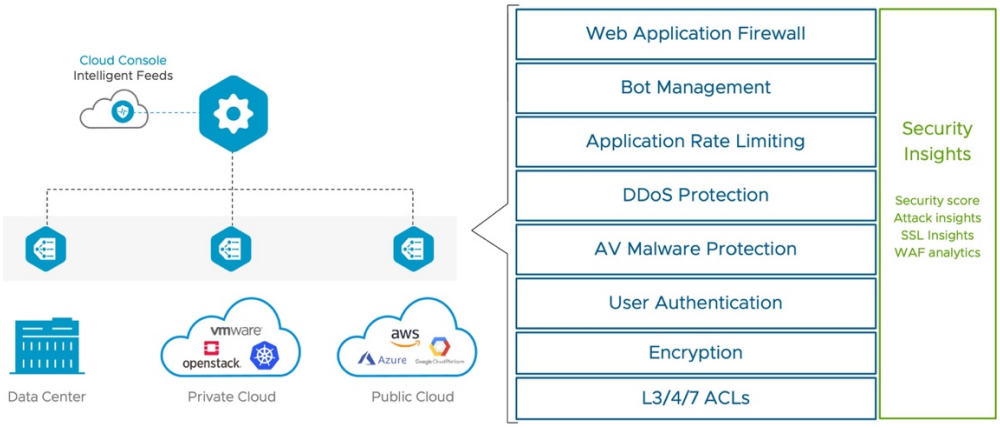


FIg 3: Avi WAF Comprehensive Security Stack and Insights

The solution supports compliance requirements such as PCI DSS, HIPAA, and GDPR, and offers point-and-click simplicity for policy management, allowing security teams to customize protections per application with ease. Avi WAF elastically scales on demand, automatically increasing capacity to handle surges in traffic or security attacks without impacting application performance. Integrated with Avi's analytics, administrators gain granular insights into security events and application behavior, enabling rapid response and precise tuning of security policies. This unified, elastic, and intelligent approach ensures robust, enterprise-grade protection for containerized workloads across any environment.

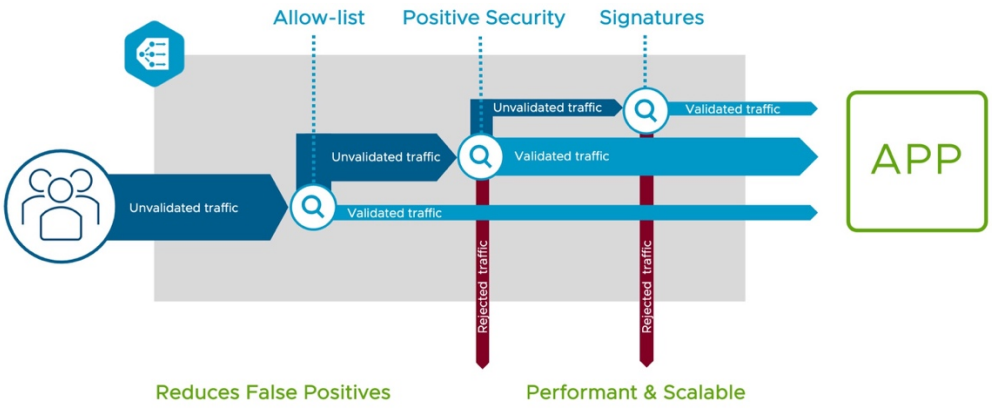


FIg 4: Avi WAF Security Pipeline Optimization

Application Insights with Real Time Monitoring and Analytics

In Kubernetes environments, maintaining application performance and diagnosing service issues require deep and real-time visibility into how workloads behave within the clusters. Traditional monitoring tools often fall short, offering only fragmented metrics or delayed insights. Avi bridges this gap with a centralized analytics engine designed specifically for Kubernetes-native observability. Avi simplifies how operators, SREs, and developers monitor, analyze, and respond to real-time traffic, errors, and application health—all without deploying separate observability stacks.

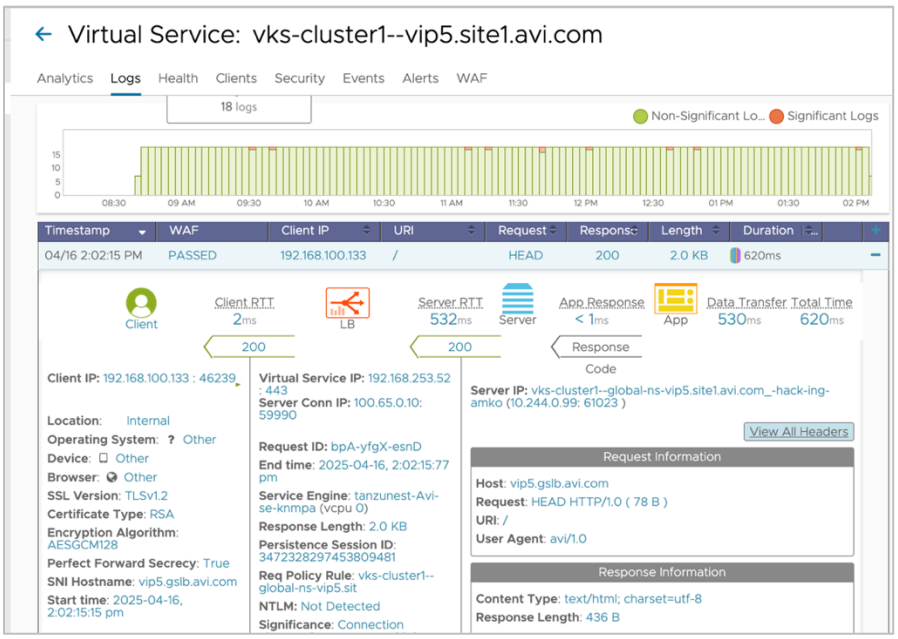


Fig 5: End User Analytics

Within a Kubernetes cluster, Avi SEs operate outside the cluster's resource pool but handle all ingress and east-west traffic managed through AKO. These SEs continuously collect telemetry at both Layer 4 and Layer 7 for every virtual service, backend pool, and transaction flowing through the cluster. This telemetry includes critical Kubernetes-specific indicators such as request throughput, error rates (e.g., 5xxs), connection counts, pod-level backend health, and service anomalies. These data points are streamed to the Avi Controller, which aggregates and correlates the information, powering intuitive dashboards accessible through UI and REST APIs.

One of Avi's core strengths is its ability to correlate high-volume Kubernetes traffic into actionable analytics. For example, users can set alerts when HTTP 5xx errors spike on a specific Kubernetes Ingress, or when response times breach defined thresholds over a rolling period. Alert logic is flexible and can be scoped per virtual service, making it ideal for dynamically scaling microservices. Actions triggered by alerts can notify SREs, create tickets, or even invoke remediation scripts via the Avi Controller—closing the gap between detection and response.

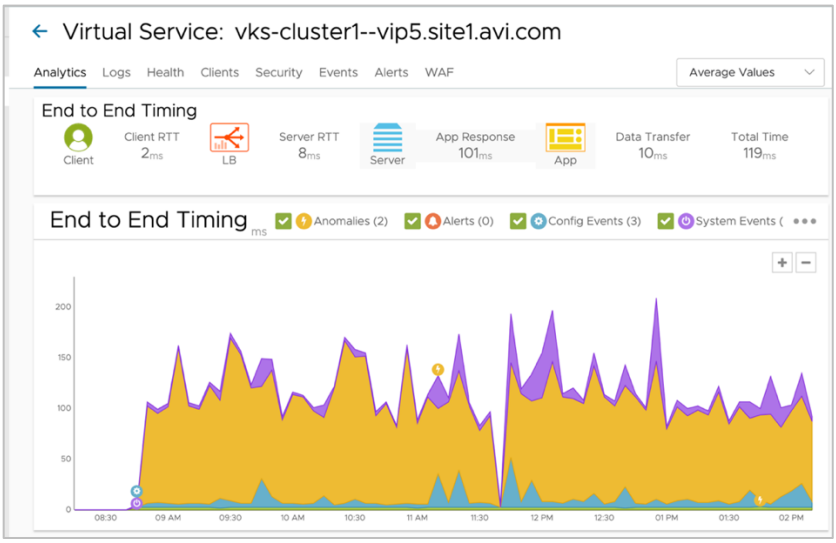


Fig 6: Avi Analytics Dashboard

The Avi analytics UI also offers Kubernetes-aware health insights by visualizing pool-level performance across backend pods. Operators can identify noisy or failing pods using server-level metrics like CPU and memory usage or connection scatter plots that expose uneven traffic distribution. Avi's embedded machine learning engine enhances Kubernetes troubleshooting by learning the normal behavior of cluster services and flagging anomalies such as traffic spikes, degraded backend performance, or unusual latency patterns—all without custom configuration.

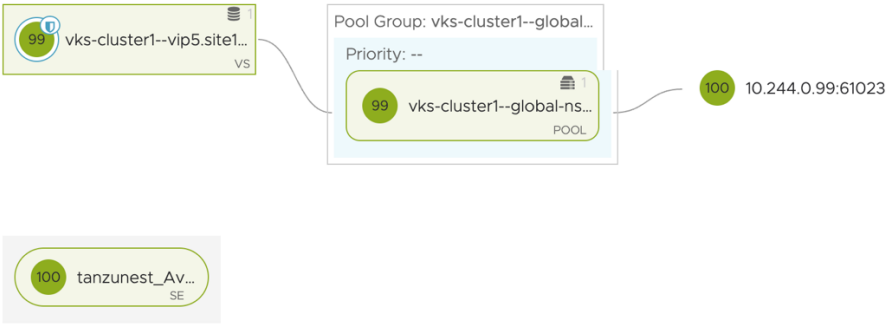


Fig 7: Application Health Scores

This Kubernetes-native observability layer empowers DevOps and platform teams to manage services proactively. Unlike traditional solutions that bolt on monitoring after deployment, Avi embeds visibility directly into the traffic flow and API ecosystem of Kubernetes. It aligns with GitOps and declarative paradigms while offering operational clarity through integrated logs, metrics, and security events.

Avi transforms Kubernetes observability from reactive to proactive—eliminating the need to stitch together tools for metrics, logging, alerting, and diagnostics. With Avi, platform operators gain a holistic, real-time understanding of service behavior within the cluster, making it dramatically easier to scale applications, respond to issues, and deliver consistent performance at enterprise scale.

DevOps Consumption of Load Balancing Services

Avi empowers DevOps teams to efficiently consume and automate load balancing services, leveraging Kubernetes Gateway API and seamless CI/CD pipeline integration. This tight integration eliminates manual bottlenecks, enhances agility, and ensures consistent and reliable application delivery. Below is a detailed breakdown of how Avi addresses these needs.

Gateway API

The Gateway API is a foundational shift in Kubernetes networking. 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.

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. It introduces new resource types, such as `Gateway`, `GatewayClass`, and `HTTPRoute`, to define and configure ingress, load balancing, and routing for applications. Unlike traditional ingress controllers, Gateway API allows greater customization, scalability, and separation of concerns by enabling cluster operators and application developers to work with distinct layers of network configuration. Its modular architecture provides a robust framework for multi-tenancy, traffic management, and advanced networking use cases.

Avi seamlessly integrates with Gateway API to deliver enterprise-grade application services for Kubernetes environments. Avi acts as the underlying implementation for Gateway API constructs, such as `Gateway` and `HTTPRoute`, providing features like L7 load balancing, TLS offloading, and observability. Avi's centralized control plane (Avi Controller) interacts with the Gateway API to automate traffic routing, handle dynamic service discovery, and support advanced networking policies, all while leveraging its software-defined distributed architecture for scalability and elasticity.

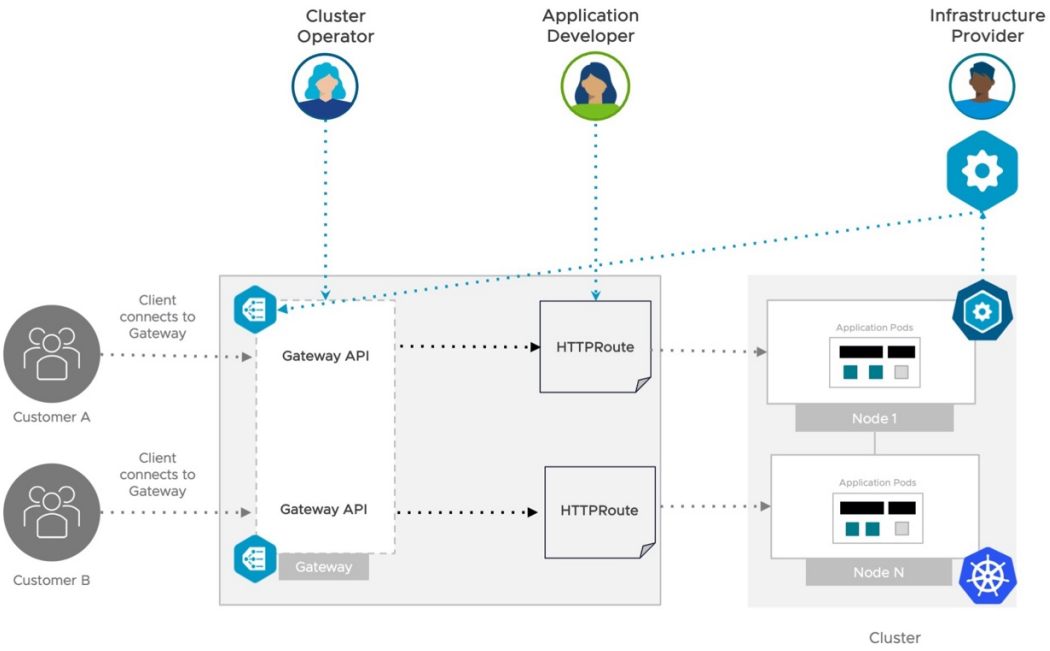


Fig 8: Avi Gateway API Architecture

Note: Customers using VMware Avi have the flexibility to choose between deploying Avi as a traditional ingress controller or leveraging the newer Kubernetes Gateway API for container ingress services. Both options integrate seamlessly with Avi's distributed architecture, enabling consolidated ingress, global server load balancing (GSLB), DNS/IPAM, and security services across multi-cluster and multi-cloud environments. This choice allows customers to select the ingress model best suited to their operational preferences and application requirements while benefiting from Avi's scalable, secure, and automated container ingress platform.

Continuous Integration and Delivery (CI/CD)

Avi offers a fully API-driven architecture, making it highly compatible with CI/CD pipelines for application delivery. This enables DevOps teams to automate the provisioning, configuration, and scaling of load balancers as part of their deployment workflows, reducing manual steps and accelerating delivery cycles

Applications can be upgraded using a Blue-Green or canary deployment pattern. Avi offers an out-of-the-box non-disruptive, graceful application upgrade capability. When a new application version is available, the Service Engines (SEs) can direct new users to the new application version while existing users continue to be serviced by the older version. Once all deployment criteria for the new application version are fulfilled, all traffic is directed towards the new application version. After a sufficient period, when all existing users have disconnected, the older version is safely deleted. The entire process can be controlled by the administrator, or Avi can provide a policy-based Blue-Green orchestration that automates the entire process. This approach ensures that application upgrades are efficient, reliable, and transparent to end users.

Avi Plug and Play integration 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 Plug and Play Load Balancing for VCF

Avi is the only solution currently available that provides comprehensive L4-L7 load balancing for both VM-based and Kubernetes workloads running under vSphere Supervisor. Its seamless integration with the Supervisor infrastructure ensures consistent, automated delivery services across VM and container ecosystems. Automation is deeply embedded into the architecture—AKO is auto-deployed within the Supervisor control plane, removing the need for manual provisioning and configurations. As the Supervisor initiates, AKO is automatically instantiated, registering with the Avi Controller to manage services via REST APIs. This enables immediate support for Kubernetes-native APIs and enhances operational efficiency through automation.

Avi supports native load balancing for Supervisor-based VM workloads, vSphere pods and for vSphere Kubernetes Service (VKS), offering unified visibility and control for all workloads. When AKO is used, it can operate as a standard Ingress or Gateway controller and enable Avi's complete L4–L7 feature set, including advanced customizations and performance optimizations.

Avi's advanced observability, analytics, and security toolsets help reduce mean time to resolution (MTTR) and improve user satisfaction. By offloading the data path from the Kubernetes clusters to external Avi Service Engines, performance is optimized, resource contention is minimized, and security policies can be enforced with precision. From a business standpoint, customers report 30% faster deployment times, 20% lower operational overhead, and measurable improvements in end-user experience.

Avi's integration with vSphere Supervisor stands apart in the market, offering unmatched automation, visibility, and security for modern workloads. Competing Ingress controllers provide limited functionality, visibility and lack Supervisor-native integration and automation, making Avi a clear differentiator for enterprises seeking a consolidated and automated platform for application delivery and security across VMs and Kubernetes.

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.

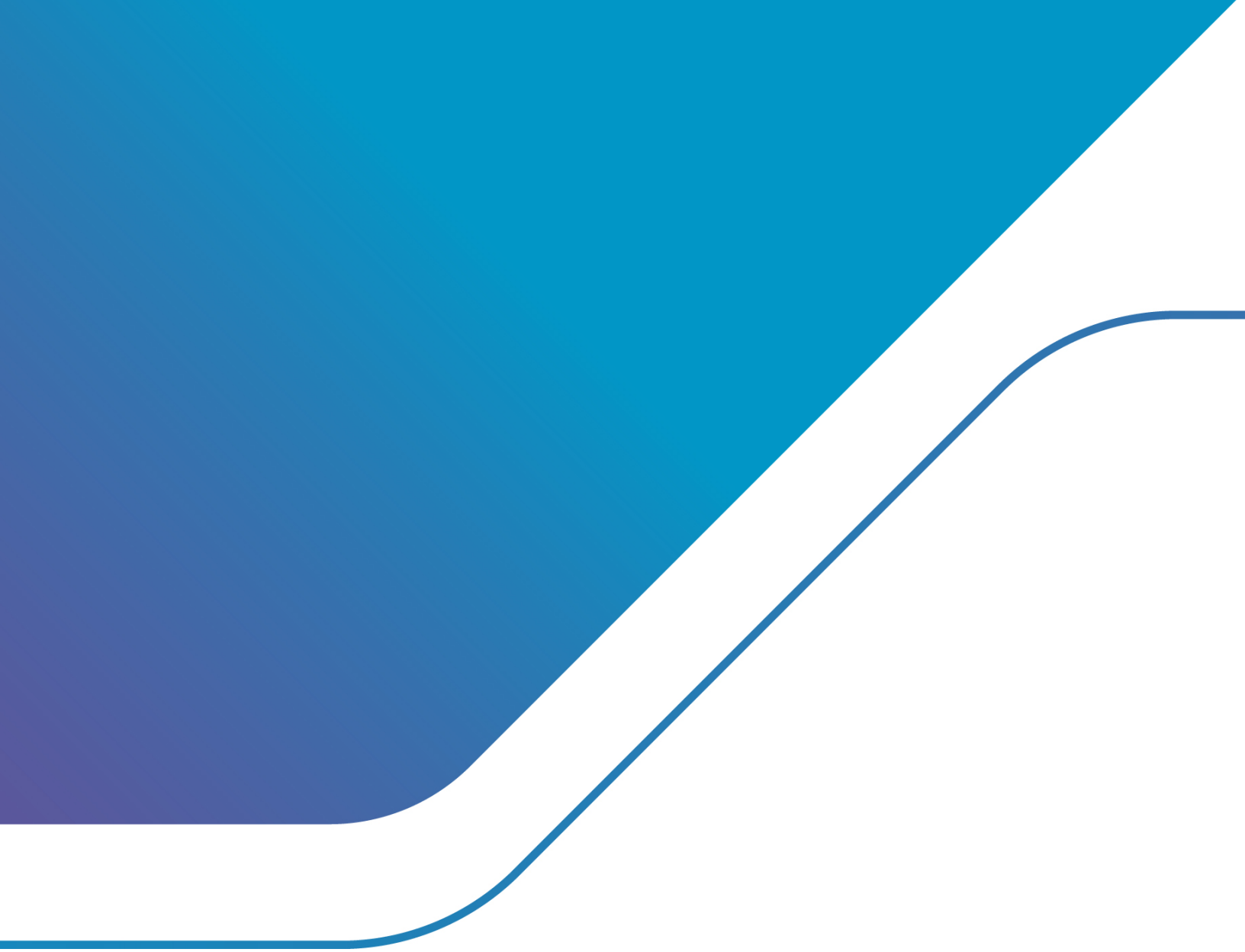
In Cloud Foundry environments, Avi integrates natively via Ops Manager to provide high-performance Layer 7 load balancing for Gorouters. As Gorouter instances scale dynamically, Avi's architecture ensures that new instances are automatically added to the relevant virtual services, facilitating real-time 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.

Avi simplifies load balancing operations through a unified architecture for both Cloud Foundry and Kubernetes. The software-based nature of Avi avoids the need for specialized hardware, reducing capital expenditure and operational complexity. Elastic scaling ensures that resources are allocated efficiently, aligning with real-time application demand. By streamlining deployment, enabling automation, and offering deep visibility into application traffic, Avi helps platform teams accelerate time-to-value while reducing total cost of ownership.

Unlike fragmented solutions that require separate tools for L4, L7, WAF and GSLB functionality, Avi offers a consolidated, feature-rich platform with native analytics, logging, and observability built-in. Whether supporting critical production workloads or enabling the migration of applications, Avi provides a consistent, reliable, and scalable load balancing solution that aligns with Tanzu's mission of modern application delivery.

Summary

Avi addresses the complexities of deploying modern application services in Kubernetes environments by providing a unified, software-defined platform. It offers comprehensive traffic management, security, and observability features with robust automation and scalability. Avi's integration with Kubernetes, Gateway API, CI/CD pipelines, VCF, and Tanzu simplifies operations, enhances resilience, and delivers enterprise-grade security, enabling organizations to deploy production-ready containerized applications efficiently.



Item No: vmw-bc-wp-tech-temp-uslet-word-2024 1/24