AVI Networks®

Avi Vantage Reference Architecture for Amazon Web Services

Avi Technical Reference (v20.1)

Avi Vantage Reference Architecture for Amazon Web Services

view online

Introduction

About AWS

Amazon Web Services (AWS) is the largest public cloud services provider offering fully automated, elastic, and resilient services pertaining to cloud infrastructure and application provisioning.

The Amazon Elastic Compute Cloud (Amazon EC2) service provides on-demand, scalable compute services for hosting compute hosts. This typically resides in an Amazon Virtual Private Cloud (VPC) which provides isolation and a virtual network for the resources to operate in.

All AWS offerings can be customized and automated to facilitate minimal deployment and management intervention.

Enterprises adopt Amazon Web Services (AWS) to serve as a natural extension to their data centers and private clouds. These organizations are application-centric and adopt continuous delivery practices across multiple environments (on-premises and cloud) and diverse infrastructures (bare-metal servers, VMs, and containers).

Traditional appliance-based load balancers lack the ability to elegantly scale across multiple clouds and do not offer real-time visibility into end-user experience or application performance. These legacy solutions also lack native integration with AWS APIs and require manual configuration. Each instance needs to be managed separately and is not developer-friendly.

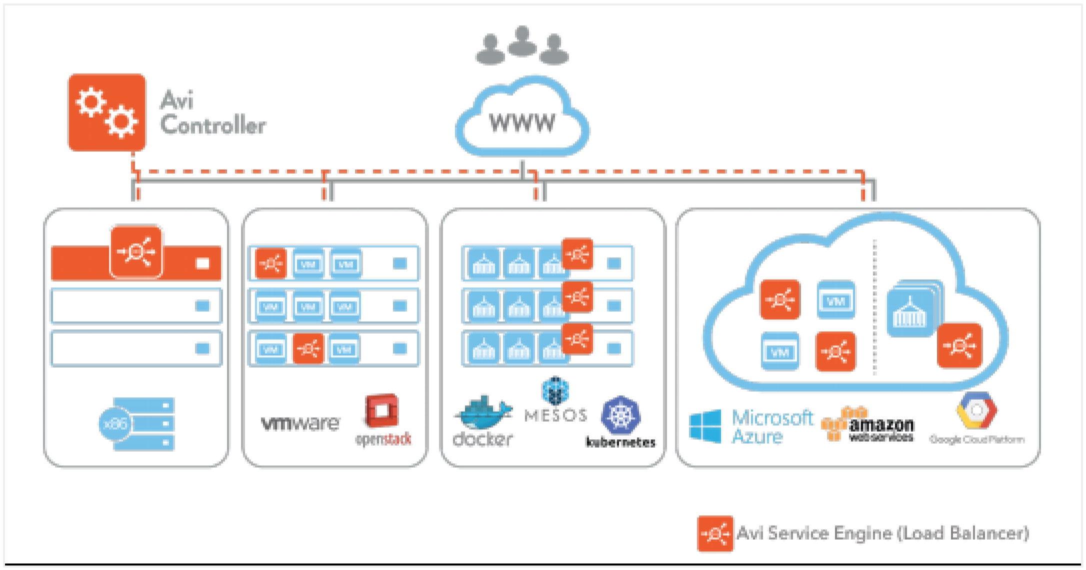
Cloud-native solutions like AWS?s Elastic Load Balancing (ELB) and Application Load Balancing (ALB) also lack enterprise-class load balancing capabilities, multi-cloud traffic management, and real-time application analytics.

Premium Load Balancing for AWS with Avi Networks

The Avi Vantage Platform is a next-generation, full-featured elastic application services fabric that is built on software-defined architectural principles.

Avi Vantage offers application services such as load balancing, security, application monitoring and analytics, and multi-cloud traffic management for workloads deployed in bare metal, virtualized, or container environments in a data center or a public cloud (Amazon Web Services, Google Cloud Platform, or Microsoft Azure).

A consistent feature set across diverse cloud environments enable IT teams to be agile without needing to constantly re-skill their IT personnel.



What Avi Vantage Provides for AWS

Enterprises modernize and maximize infrastructure utilization with AWS. The next phase of this modernization is to extend the app-centricity to the networking stack. Avi Networks delivers elastic application services that extend beyond load balancing to deliver real-time app and security insights, simplify troubleshooting, auto scale predictively, and enable developer self- service and automation.

Avi Networks provides an ELB-like experience for applications deployed in on-premises and multiple cloud infrastructures.

Full-featured load balancing

AWS ELB and ALB provide basic load balancing capabilities and lack enterprise-class features and advanced policy support. Avi Vantage delivers full-featured load balancing, including multiple load-balancing algorithms, advanced HTTP content switching capabilities, comprehensive persistence, customizable health monitoring, DNS services, and GSLB across multiple clouds. Avi Vantage provides these capabilities in an as-a-service experience similar to AWS ELB with native AWS API integration.

Automation

Avi Vantage is a 100% REST API-based solution, that offers Python SDK, Ansible playbooks, and CloudFormation templates for automating configuration and operations. Avi Vantage natively integrates with AWS APIs for spinning up EC2 instances, allocating Elastic IPs, Route53 integration, auto scaling, and availability zone awareness. Avi Vantage simplifies CI/CD ops by supporting blue-green deployments and canary upgrades.

Advanced Security

AWS ELB and ALB lack advanced security policies, SSL insights, and DDoS capabilities. Avi Vantage provides network ACLs, advanced HTTP security policies, SSL insights, DDoS detection and mitigation capabilities, along with micro-segmentation in container environments. Avi Vantage also provides Web Application Firewall (WAF) capabilities seamlessly integrated with the load balancing feature sets.

Visibility and Monitoring

With ALB and ELB, admins and developers do not have integrated real-time telemetry and must deploy third party tools and services for analytics. Avi Vantage delivers real-time insights into application health, end-user experience, log analytics, and security insights.

Multi-cloud load balancing

Inconsistent capabilities across clouds create challenges for network engineers to move workloads across multiple cloud infrastructures. This also forces enterprises to re-invest in training and education. Using native tools locks enterprises to the specific cloud, preventing workload mobility and increasing business risk. Avi Vantage enables dynamic workload mobility across clouds based on business metrics such as cost, performance, security, and compliance requirements, reducing risk and providing flexibility.

Reduced TCO

With AWS, the cost of load balancing (ELB, ALB), security (WAF), and visibility (third-party logging tools) adds up to a significantly higher investment. Avi Vantage reduces the total cost of ownership (TCO) while providing rich functionality.

Intended Audience

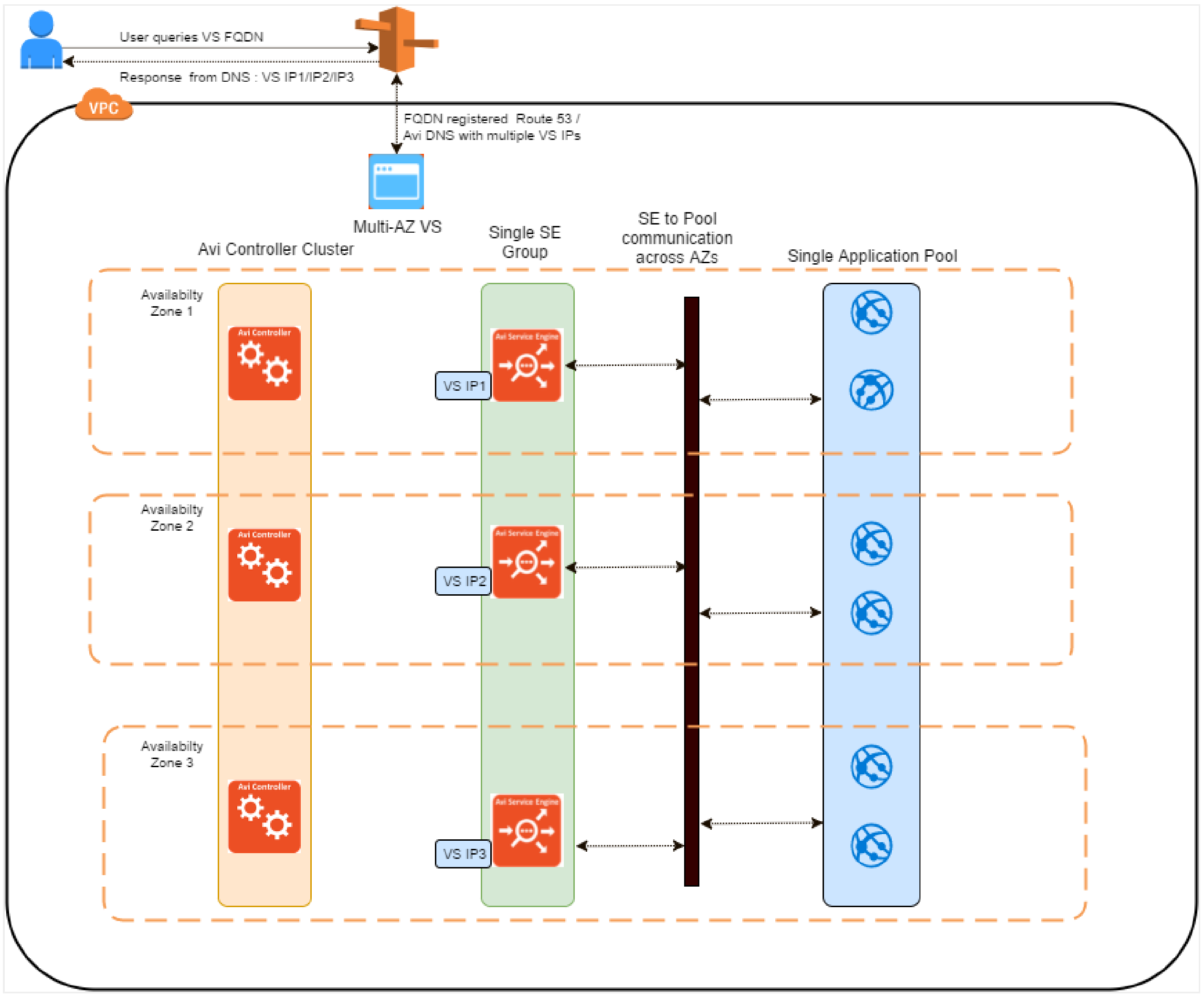
To understand this reference architecture, we assume familiarity with: * Basic AWS functionality, particularly Networking (VPC), Compute (EC2) and DNS (Route53). For complete information, refer to AWS Documentation. * Basic understanding of Load Balancing and ADC, refer to Avi Documentation.

Avi Vantage Reference Deployments in AWS

This section discusses few reference deployments where Avi Vantage is used to provide application delivery services to workloads on AWS.

Single VPC, Multiple Availability Zones (AZs)

The single VPC deployment has all workloads present within a single AWS VPC. This is a typical deployment when an application is hosted solely on AWS. Multiple Availability Zones (AZs) are leveraged for high availability and redundancy.



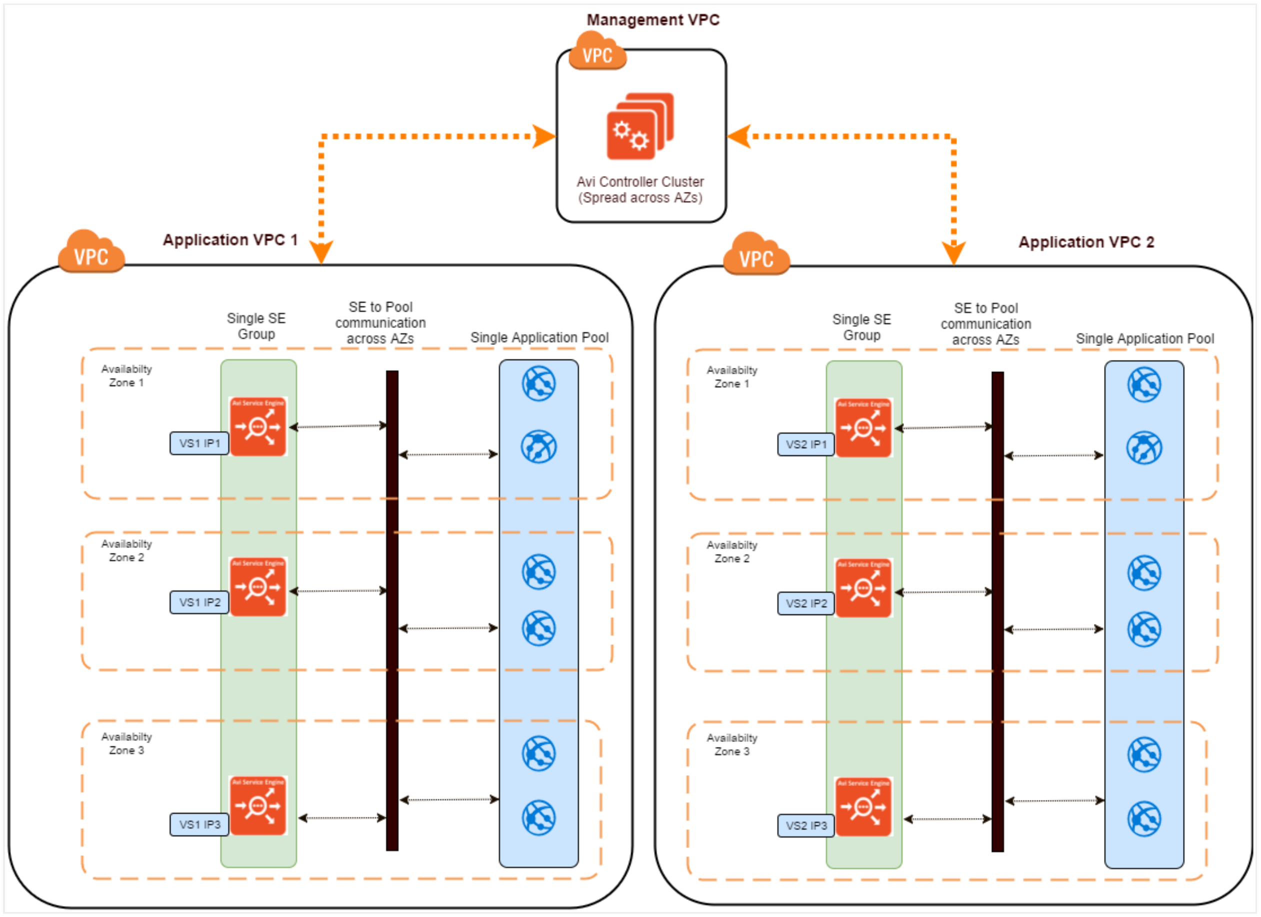
Consider the following points while deploying Avi Vantage in this architecture: * Spawn the Avi Controller cluster in the same VPC. As the Avi Controller cluster has three controller nodes, these nodes can be spawned in separate Availability Zones to ensure high availability. * The virtual service is configured with multiple VIPs, one per Availability Zone. As the virtual service is distributed across multiple AZs, a minimum of one Service Engine per AZ is sufficient to provide high availability. This can be achieved by updating Buffer Service Engines to zero when the default Elastic HA (N+M) HA mode is selected. * As there will be one VIP created per AZ, the virtual service will end up having multiple VIPs. To redirect client traffic to various SEs, Route 53 or Avi DNS needs to be configured. DNS Load balancing will ensure that all VIPs are advertised and utilized.

Centralized Management VPC, Dedicated Application VPCs

This deployment separates out the management function (Avi Controller) to a separate VPC. The workloads are present in other VPCs. One AWS cloud is configured within the Avi Controller cluster for each VPC where application delivery services are required.

This separation of management VPC is typically preferred when a central IT team is tasked with managing the infrastructure, but wants to provide multi-tenant, self-service functionality to its internal users. The AWS resource usage billing is also separated out in this architecture, to be at the per-VPC level.

Per-line-of-business VPCs is a primary use case for this approach.

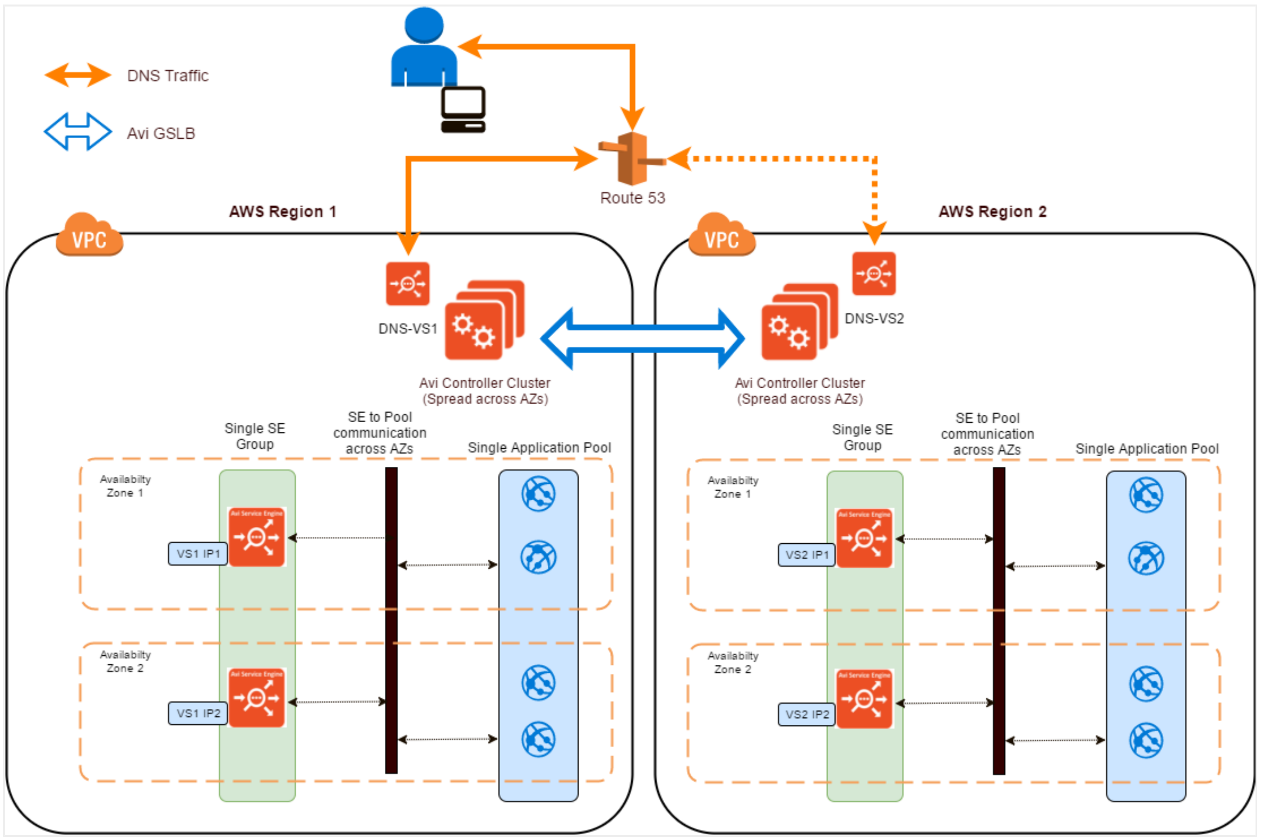


Consider the following points while deploying Avi Vantage in this architecture:

- Spawn the Avi Controller cluster in the management VPC. As the Avi Controller cluster has three controller nodes, these nodes can be spawned in separate AZs to ensure high availability.
- Configuring a separate AWS cloud in Avi Vantage, per application VPC. Each AWS cloud can be associated with a separate Avi tenant, so that only designated users can provision or manage virtual services for that VPC.
- Within each VPC, it is recommended to use multiple AZs for Service Engine (and correspondingly virtual service) HA.
- Route 53 or Avi DNS integration will be required to advertise and manage the virtual services with multiple VIPs.

AWS Multi-Region Deployment with GSLB

This deployment comprises of applications that are available in multiple AWS regions simultaneously, for redundancy as well as better latency and user experience for global end users.

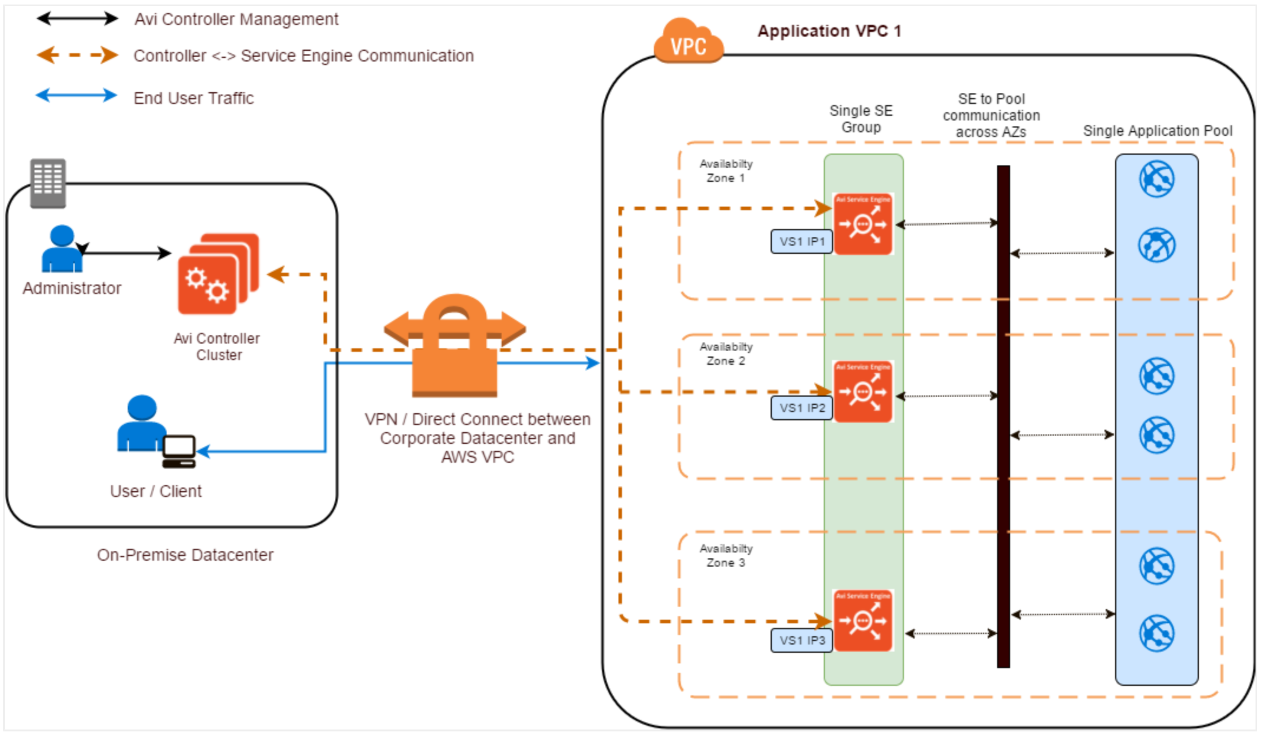


In this deployment, at least two VPCs, each in a different region are used. There is one Avi Controller cluster per VPC. As the applications are deployed in each VPC, the Avi Controller of the respective VPC manages the virtual service for the application.

To tie multiple virtual services across the VPCs or regions and make them available to the end-users transparently, Avi Vantage?s Global Server Load Balancing (GSLB) feature is used across all the controller clusters.

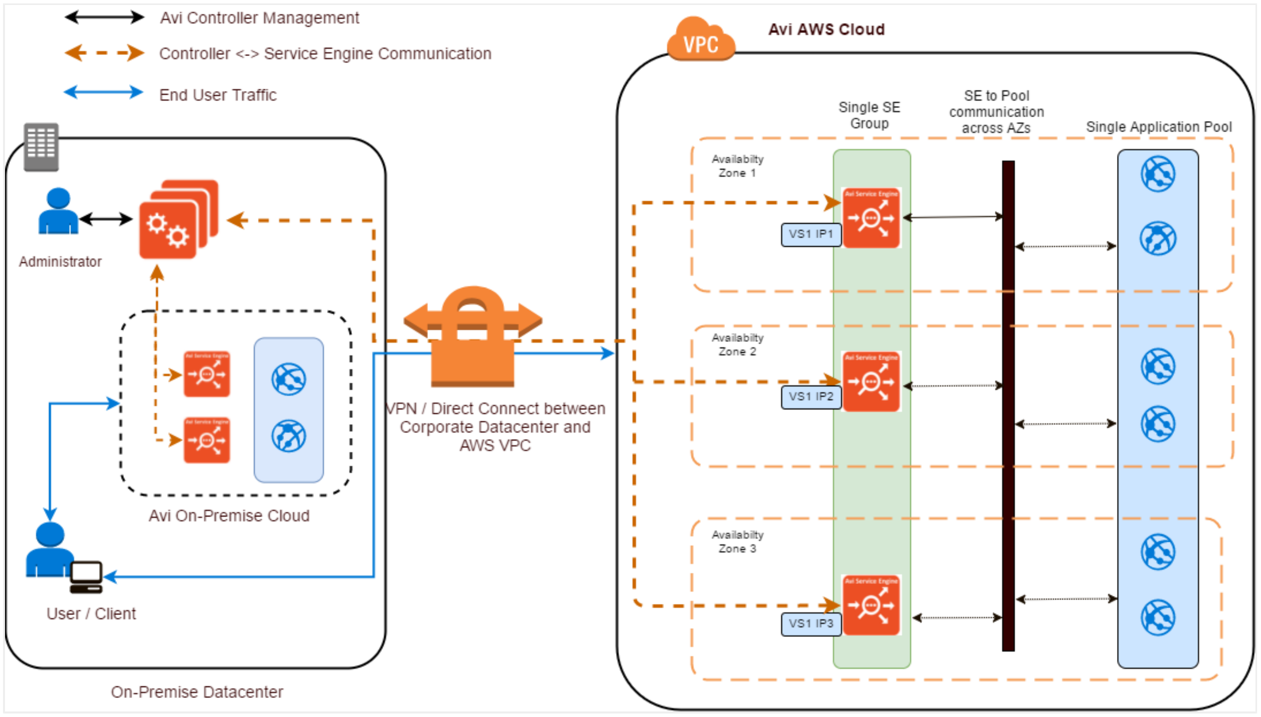
Hybrid ? Avi Controller On-Premises, Applications in AWS

This deployment comprises of an architecture where the Avi Controller is present on-premises in the customer datacenter. The application workloads could either be hosted exclusively in an AWS region, or present in both AWS and the on-premises datacenter.



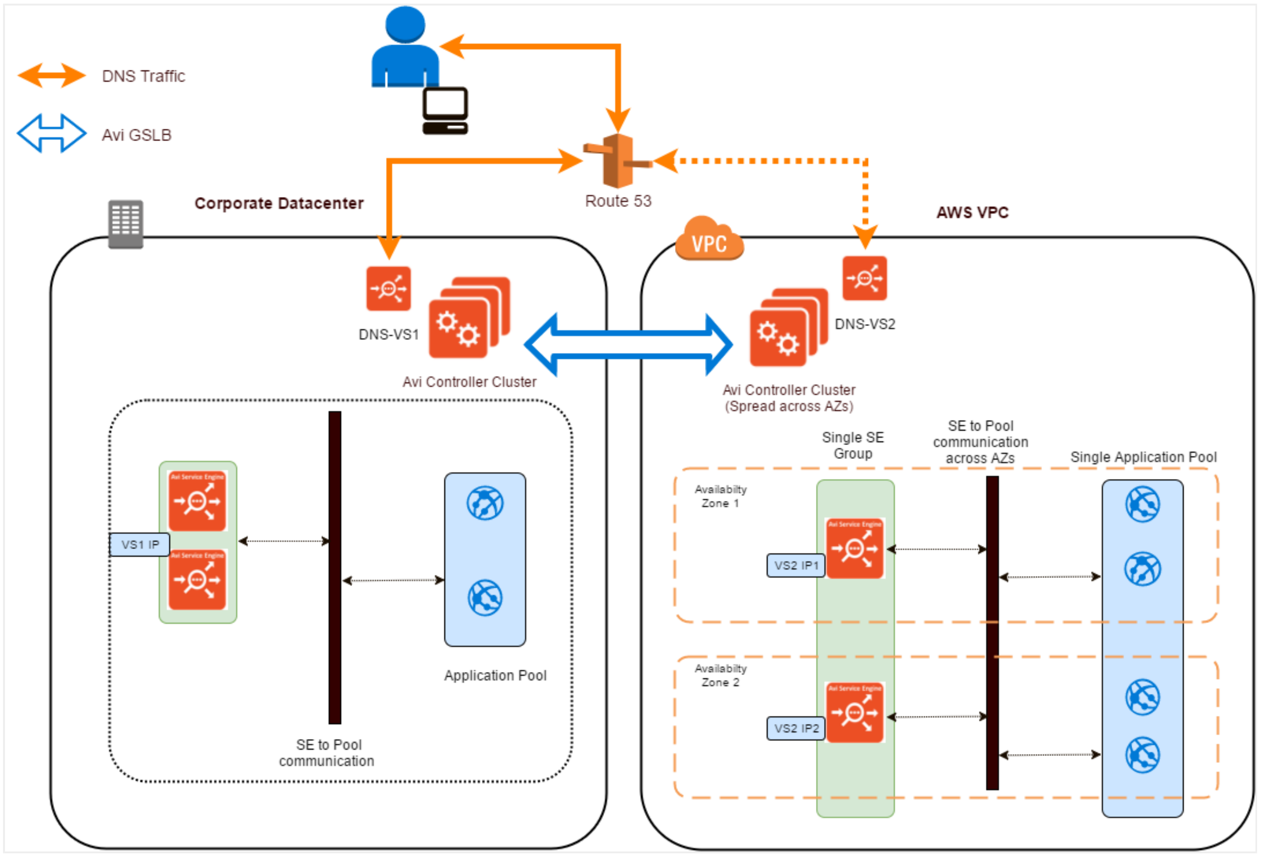
In the case where the application workloads are exclusively in AWS, an AWS cloud is configured in the Avi Controller. The Avi Controller spawns Service Engines in the AWS VPC provided.

This scenario requires connectivity between the datacenter and AWS VPC to allow management communication between Avi SEs and Avi Controllers.



The second use case has the application workloads hosted in the on-premises datacenter as well as AWS. The on-premises datacenter could be the primary application workload provider, with the AWS cloud being configured for a smaller percentage of traffic, or for scaling up in case of traffic rise. This provides a seamless way to increase application availability during high load, while making efficient use of cloud resources.

Hybrid ? Multi Cloud with GSLB



A combination of (c) and (d), this is a multi-cloud approach. In this scenario, the multiple Avi Controller clusters are all not on AWS. Instead, they could be distributed across public and private cloud ecosystems, such as AWS, Google Cloud Platform, Microsoft Azure, Openstack, VMware, etc.

GSLB is used to manage DNS queries for the virtual services and redirect requests to individual application instances as required.

Given Avi Vantage?s extensive support for multiple ecosystem, this enables end customers to have a truly diverse and resilient architecture for applications and be served seamlessly.

Avi Vantage Architecture

The Avi Vantage Platform provides enterprise-grade distributed ADC solutions for on-premises as well as public cloud infrastructure. Avi Vantage also provides inbuilt analytics that improves the end-user application experience as well as ease of operationalizing for network administrators.

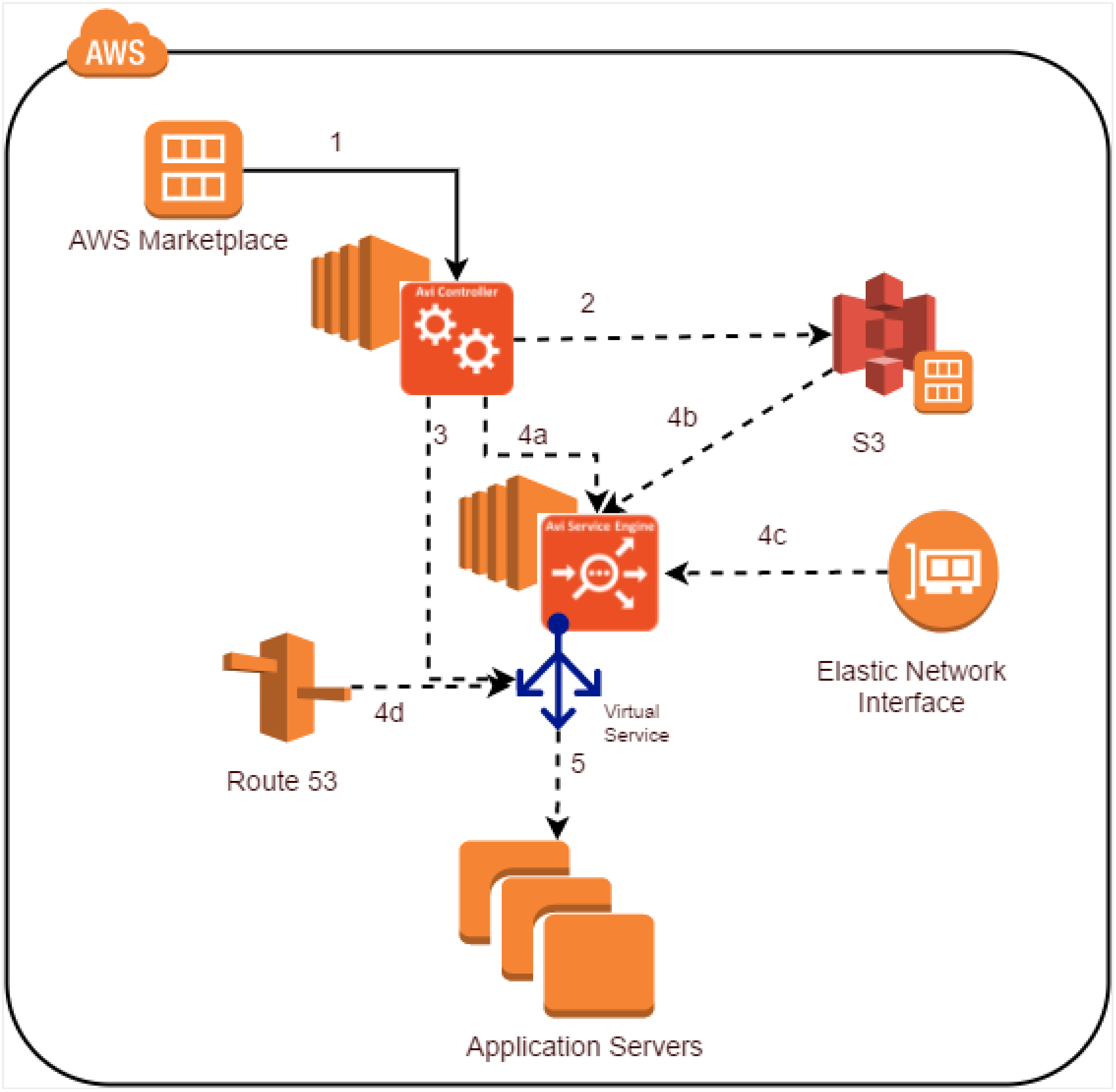
Avi Vantage is a complete software solution which runs on commodity x86 servers or as a Virtual Machine and is entirely enabled by REST APIs.

Avi Architecture Components

The Avi Vantage Platform is built on software-defined architectural principles which separates the control and data planes. The product components include:

- Avi Controller (control plane): Central policy and management plane that analyzes the real-time telemetry collected from Avi Service Engines and presents visual, actionable dashboards using an intuitive user interface built on RESTful APIs.
- Avi Service Engines (data plane): Distributed load balancers that are deployed closest to the applications across multiple cloud infrastructures. The Avi Service Engines collect and send real-time application telemetry to Avi Controller.

Interactions with AWS Services



The above diagram depicts the Avi Vantage deployment process briefly.

1. The Avi Controller is available in the AWS Marketplace. A new instance of the controller is deployed.

2. Once the Avi Controller is instantiated and is running, the controller UI is accessed, and an AWS Cloud is configured within the controller. This entails providing AWS credentials, VPC details, etc. At the end of this process, an Amazon Machine Images (AMI) is generated and uploaded to an Amazon Simple Storage Service (S3) bucket within the account. This AMI is used to deploy Service Engines as required.

3. In this step, a new virtual service is configured using the Avi UI. Details related to the VIP, backend pool members, FQDN for the virtual service are entered. This in turn triggers the following events:

4. Step 4a: The controller scans for information regarding any existing Service Engine with capacity to place the virtual service. Step 4b: As this is a new controller (or if all the Service Engines are loaded), a new Service Engine instance is launched in EC2 with the AMI uploaded by the controller in step 2.
   Step 4c: The Service Engine instance obtains a couple of ENIs ? one for management communication to the controller and other SEs, and another for data traffic between clients and the application servers.
   Step 4dd: Route 53 is updated with the FQDN, if specified, and the VIP is advertised.

5. The virtual service is marked up and starts load balancing operations for traffic from the client to the application servers.

The various AWS components that Avi Vantage interacts with are explained below:

i. Authentication and IAM Roles

The Avi Controller needs permissions to deploy and manage the Service Engines, and optionally interact with Route 53 on AWS. The following are the two methods used by Avi Vantage for authentication:

1. Access key ID and secret access key
2. IAM role that has been configured and used while creating the Avi Controller. For complete information on creating IAM roles, refer to iam role setup for installation into aws.

ii. EC2: Instances

Avi Vantage deploys the controller and service engines in the AWS EC2 environment. Typically, a VPC would have been provisioned for your infrastructure, including the application servers, etc. Avi Vantage will be deployed within this VPC and will utilize resources connected to this VPC, as described in the sections below.

iii. Controller resources and deployment

The Avi Vantage Controller is available in the AWS marketplace at the link here.

To get started, launch an Avi controller instance on AWS EC2.

The Avi Controller can be deployed on different instance sizes, depending on the scale expected. At a minimum, an m4. 2xlarge instance is recommended. Please refer to this link for controller sizing details.

Once the Avi Controller is launched and is online, perform the initial configuration for the cloud type you want to use (In this document, we assume that the backend servers are on AWS, and this guide will use the AWS cloud).

At this stage, once the AWS cloud credentials (or IAM roles have been provided), Avi Vantage reports back with cloud status as Ready after verification.

The system is now available to configure virtual services.

iv. SE resources and deployment

The Avi Service Engine (SE) is the load balancing entity on which the virtual service IP (VIP) is programmed. The SEs are automatically spawned by the Avi Controller when an application is configured. The entire lifecycle of the SE is automated.

The SE can be of varied instance type, depending on application needs ? from t2.micro to c4.8xlarge. Depending on application traffic and load requirements, the appropriate instance type can be configured.

In addition, multiple SEs can be spawned (scaled out) to handle more incoming traffic.

v. Networking: VPC, Subnets, ENIs, Elastic IP

The Avi Controller interacts with the following AWS networking elements:

- Virtual Private Cloud (VPC) : Avi Vantage is typically deployed in a VPC. All instantiations of Service Engines, virtual service creation, addition of application pool members happens in this context.
- Subnets: Avi Vantage accesses AWS subnets to obtain IP addresses for multiple purposes:
    - IP address for the Service Engines? management interface.
    - IP address for the virtual services.
    - IP address for the Service Engine data ENIs, to be able to communicate with the back-end application servers.
- ENIs: Avi Vantage automatically creates ENIs and assigns IP addresses from the appropriate subnet, and attaches these to the requisite Service Engine.
- Elastic IP: Avi Vantage can use an elastic IP with a public IP for a virtual service, if required. This is not a mandatory configuration. If it is not a public facing application, you can continue using the internal IP address accessible within the VPC.

vi. DNS - Route 53 and IPAM

Avi Vantage provides native support for DNS (to resolve virtual service IPs) and IPAM (for virtual service IP address management). Both these services are optional, but will likely be used in production deployments.

In addition to providing native support, Avi Vantage also interacts with various third party services depending on the configured cloud ecosystem.

In the case of AWS, IPAM for native AWS workloads is provided by default, without any configuration. As the controller and service engines are powered up, they interact with the AWS VPC and networking infrastructure to receive appropriate IP addresses.

For DNS, the options available for AWS are either Avi Vantage?s Native DNS or Route 53.

When Route 53 is used, Avi Vantage will request an optional FQDN for each virtual service being created. On providing a valid FQDN, Avi Vantage will program this record on AWS Route 53 DNS.

When the virtual service is deleted, all resources including the DNS record are removed.

For more details on configuring DNS with Avi Vantage on AWS, refer to Avi Vantage DNS with AWS Cloud

Starting release 17.1.2, Avi Vantage is multi-AZ capable. As a result, virtual service can be hosted on Service Engines in different availability zones. One virtual service IP (VIP) will be configured in each AZ, and the DNS record can point to the multiple VIPs for load balancing.

vii. AWS Auto Scale Groups, SNS and SQS

Starting Avi Vantage release 17.1.2, Avi pool configuration can reference an AWS Auto Scale group. When this is configured, Avi Vantage polls the AWS Auto Scale group for membership changes, and synchronizes the Avi pool configuration accordingly.

For example, when a new application instance is added to the AWS auto scale group, Avi Vantage will automatically add it to the Avi pool configuration without any user intervention. This new instance will then start servicing client requests. Conversely, when an instance is removed from the AWS auto scale group, the same event is synchronized with Avi Vantage?s pool configuration.

For more details on this feature, refer to Avi Integration with AWS Auto Scaling Groups.

Starting release 17.2.3, Avi Vantage can take over the management of an AWS Auto Scale group. In this case, the membership for the AWS Auto Scale group can be administered by Avi Vantage?s metrics, providing more granular control over member scale out and scale in.

Advantages of using Avi Vantage

A single Avi Controller cluster can manage hundreds of Service Engines. The Avi Controller has the intelligence to place virtual services on appropriate SEs. Due to Avi Vantage?s architecture, the distributed load balancers can achieve unique high-availability features.

1. Per-application load balancer:

a. As Avi Vantage enables the deployment of hundreds of distributed load balancers and it becomes administratively simpler to dedicate a load balancer per application.
b. A per-app load balancer reduces security concerns from neighboring applications.
c. A per-app load balancer also removes the noisy neighbor problem.
d. A per-app load balancer can be configured by enabling the Per Application knob within the Service Engine Group tab.

2. Auto-healing load balancers:

a. As Avi Controller can communicate with AWS to dynamically spin up and spin down Service Engines, it can automatically recover lost load balancing capacity. If one of the Service Engines fails or faces some issues, Avi Controller moves the virtual service to a buffer Service Engine and will then spin up a new Service Engine to recover the HA capability.
b. Auto-healing improves high availability and reduces application downtime.
c. Auto-healing provides ease of use and management for load balancing administrators.

3. Auto scaling load balancers:

Avi Controller provides automatic scaling up and down of load balancing capacity depending upon the virtual service requirements. Auto scaling is decided based on the Service Engine CPU usage. The following are few examples:
a) If Service Engine CPU usage goes above 80% (MAX_ALLOWED_CPU) then Avi Controller will spin up one more instance of Service Engine and perform the scale out operation.
b) If Service Engine CPU usage goes below 30% (MIN_ALLOWED_CPU) then Avi Controller will spin down one of the instances of Service Engine and perform the scale in operation.

Sizing and Performance

Controller Sizing

During the deployment, the Avi Controller?s system capacity can be specified based on the following system resource allocations:

- CPUs
- Memory (RAM)
- Disk

There is a minimum requirement for each of the above, and larger amount of resources will be required based on the virtual service scale and application load.

In a production environment, a cluster of the three similar-sized controllers is recommended.

For a non-production environment, a single controller may suffice.

In case of deployment in AWS, provisioning of the Avi Controller AMI is allowed only on instances with enough resources.

For more details on controller sizing, refer to Avi Controller Sizing.

Service Engine Sizing

i. SE Instance Size

When deploying an Avi SE on AWS, the instance type needs to be selected.

Avi Vantage supports many instance types and sizes. The main considerations are support for at least two ENIs, 1 vCPU, and 1 GB RAM. By default, a t2.micro instance is instantiated. This can be modified in the SE group properties.

- For general purpose requirements, t2.* instances may suffice. Note that t2.* instances are not supported in a dedicated VPC.
- For high performance requirements, Avi recommends using the C5.* and R4.* instances as these provide superior performance, and also support enhanced networking.

ii. Virtual Service Scale per SE

The number of virtual services that can be created on each instance type also varies.

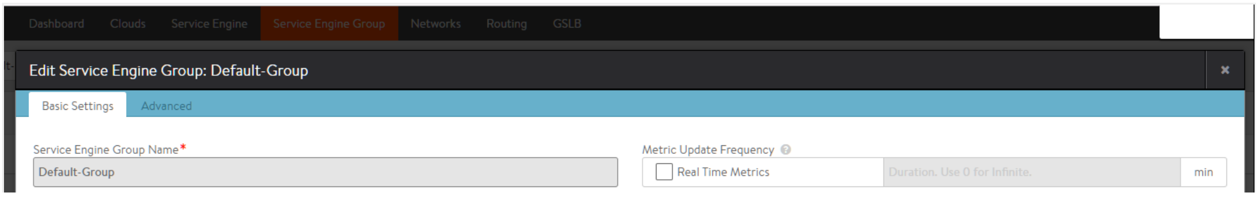
Prior to release 17.1, Avi Vantage supported one data NIC, and hence the number of virtual services that can be created was (n-1), where n is the number of IPv4 addresses supported on the NIC. One IPv4 address was reserved as the interface IP address. For example c4.large supports 10 IP addresses on each interface. Hence, a maximum of 10 ? 1 = 9 virtual services can be configured.

Starting release 17.1, Avi Vantage supports multiple data NICs, and the number of virtual services that can be created is equal to (n-1) * (m-1). Here, n is the number of IPv4 addresses supported on each NIC, and m is the number of NICs supported on the instance. For example c4.large supports 10 IP addresses on each interface. And supports 3 interfaces. Hence, a maximum of (10 ? 1) * (3 ? 1) = 18 virtual services can be configured.

For more details on AWS ENI and IPv4 limits, refer to the AWS documentation for Elastic Network Interfaces.

iii. Per-App Deployment

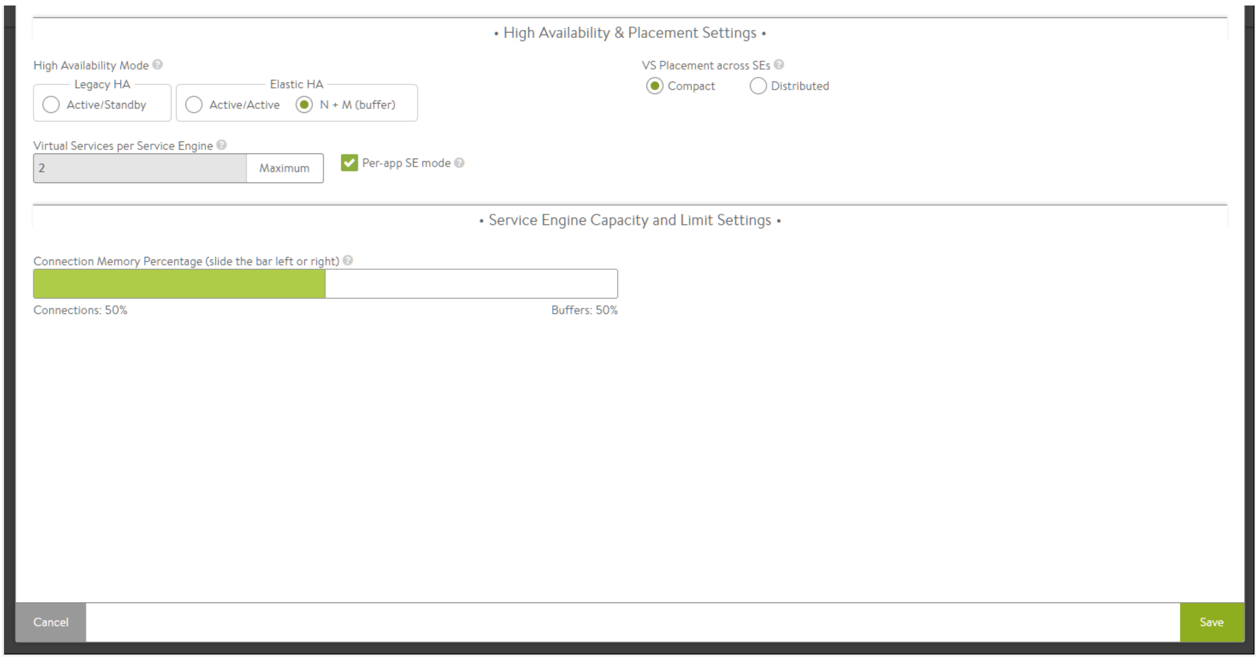
The per-app deployment can be turned on in the Service Engine Group properties using a check-box as shown below.



Edit Service Engine Group: Default-Group

Service Engine Group Name* Default-Group

Metric Update Frequency
Real Time Metrics
Duration. Use 0 for Infinite.


• High Availability & Placement Settings •

High Availability Mode
Legacy HA: ( ) Active/Standby
Elastic HA: ( ) Active/Active [✓ N + M (buffer)]

VS Placement across SEs: [✓ Compact] ( ) Distributed

Virtual Services per Service Engine: 2 Maximum

✓ Per-app SE mode

• Service Engine Capacity and Limit Settings •

Connection Memory Percentage (slide the bar left or right)
Connections: 50% Buffers: 50%

The per-app deployment mode is useful in cases where higher resilience is required for the applications. As the number of virtual services is limited to two per Service Engine, the failure domain for a given application is reduced. In this mode, each Service Engine is limited to hosting a maximum of two virtual services, and four pools per virtual service. There is no limit on the size of the Service Engine and this can be selected based on the sizing considerations.

Each license in this mode counts as a 0.25 core, providing cost optimization.

iv. Service Engine Performance

The performance of Avi Service Engine can be characterized based on the following parameters:

- SSL transactions per second (TPS)

  SSL TPS signifies the number of GET requests for a 128-byte page, that can be served by Avi Vantage in one second, over an HTTPS connection.

  As x86 chipsets increasingly provide advanced encryption offload capabilities, the Avi SE takes advantage of these x86 features to provide high SSL performance.

  AWS instances such as the C5.* are compute optimized with dedicated Intel V3 CPUs, providing TPS performance comparable to a bare-metal SE.

- L4, HTTP and HTTPS throughput

  This metric portrays the maximum traffic that can be serviced by the Service Engine, and is calculated by requesting multiple GET requests for a 128 KB document.

  The network performance is largely constrained by the hypervisor or QoS bandwidth limitations. In case of AWS as well, the instances which support Enhanced Networking, such as c4., *c5.*, r4., *and some m4.* instances, provide high throughput.

- Concurrent connections per second

This metric signifies the number of open, concurrent connections that the Avi SE can process. This is largely governed by the memory size.

Provisioning

This section covers Avi Vantage features in AWS, and references related to provisioning, monitoring, and troubleshooting various Avi Vantage elements.

Provisioning - CloudFormation

The Avi Controller can be deployed and configured using AWS CloudFormation templates. This are useful when the controller deployment needs to be done multiple times.

In addition to the controller deployment, the CloudFormation template can also contain the configuration data for the pools, virtual services and others, that need to be configured at initialization.

A sample CloudFormation template is available here.

Scaling - Auto Scale

Avi Vantage collects more than 700 data points per second regarding the performance and characteristics of various network and compute entities ? service engines, backend application servers, front-end (virtual service) CPU/traffic load, etc. Using these metrics, Avi Vantage can predict and fine-tune its functioning. One important use-case is knowing when to scale the system automatically, based on the current and predicted loads.

At first level, Avi Vantage can scale out the virtual service to another Service Engine based on these metrics. This will ensure that any increase in load to the virtual service is met with using the additional service engine instance(s) that have been created. This can be done manually or automatically based on CPU utilization thresholds, or by using user-defined ControlScripts tied to specific events.

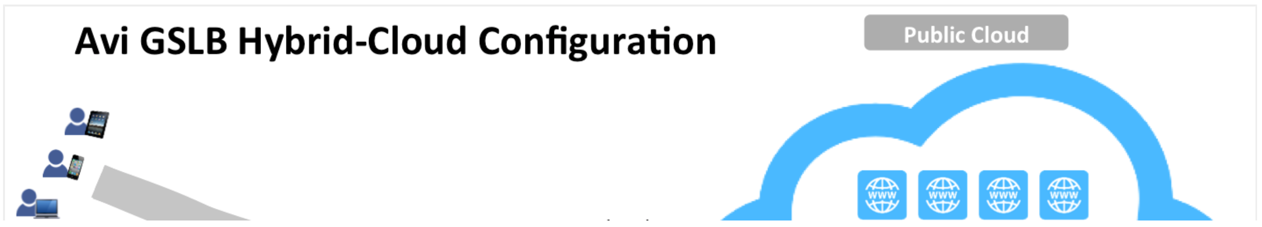
On AWS, service engine scale out is a matter of checking whether an existing service engine instance with sufficient capacity exists within the same SE group. If not, a new EC2 instance is spawned and assimilated into the group of SEs servicing the virtual service.

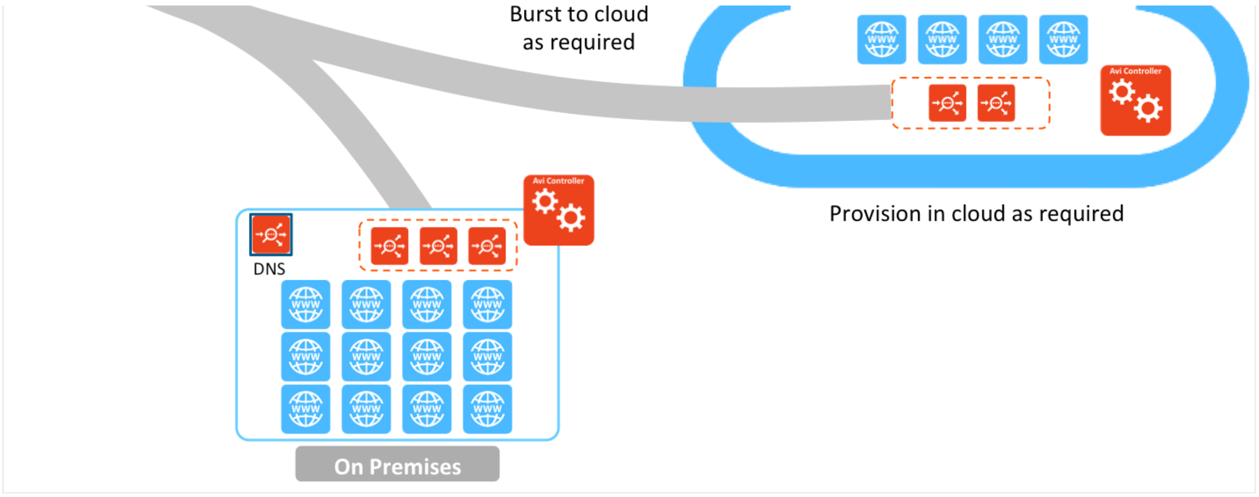
The second level of auto scaling is when the backend application pool servers need to be scaled out as well. If the load on an application has increased, chances are that only scaling the load balancer will not alleviate the load on the application servers.

Avi Vantage facilitates auto scaling of application server by using an user-defined event (or a set of events) to be the trigger to create another instance of the application in AWS. Once this is done, Avi Vantage adds the new instance into the virtual service?s server pool so that it can start serving the incoming requests.

Scaling - Cloud bursting

Cloud bursting is a mechanism to handle excess on-demand load for a temporary period. As an example, consider major US shopping events such as Black Friday, Cyber Monday, or financial deadlines such as those for filing tax returns. In these cases, the retailers? and the financial service providers? infrastructure, respectively, would encounter load that is tens to hundreds of times greater.





In this scenario, if the organization?s primary data center is on-premises, a secondary overflow cloud can be configured on public cloud such as AWS.

When the load reaches a certain threshold, Avi Controller can trigger scripts to * Provision and enable the application and infrastructure on AWS. * Re-configure DNS to prioritize a certain, user-defined amount of traffic to AWS using GSLB. * Continue to monitor the load.

Once the load reaches the normal levels, the cloud infrastructure can be stopped to save costs.

This scenario can be provisioned to be triggered manually, or automatically without requiring operator intervention.

Operations and Troubleshooting

The following are the options available for debugging and operationalizing Avi Vantage: * Events * Alerts * Services * Traffic Capture

Events

Events are used throughout Avi Vantage to provide a history of relevant changes that have occurred. Events are a permanent record, and can be used to generate alerts which can take action on the event. In the UI, events may be viewed within the context of specific objects, such as a virtual service, a pool, or a server. For more information, refer to Events Overview.

Alerts

In its most generic form, alerts are intended to inform administrators of significant events within Avi Vantage. In addition to system events? triggers, alerts may also be triggered based on one or more of the 200+ metrics that Avi Vantage is tracking. For more information, refer to Events Overview, Backup and Restore Avi Vantage Configuration.

Services

Avi Vantage provides alert actions using various mechanisms. You can configure remote syslog server, Splunk, email, SNMP TRAP etc., to log appropriate events and logs into an external audit system.

Traffic Capture

Most connection troubleshooting or traffic data may be done quickly using virtual services logs. However, some troubleshooting may require full visibility into the packet transmission. Avi Vantage provides packet capture, which runs TCPdump against a designated virtual service. The packet capture is done on all Service Engines that may be hosting the virtual service and is then collated into a completed capture. For more information, refer to [Traffic Capture.]

Resources

1. Avi Knowledge Base: https://avinetworks.com/docs/
2. Avi Vantage AWS Installation Guide: https://kb.avinetworks.com/docs/latest/installing-avi-vantage-in-amazon-web-services/
3. Avi SDK: https://avinetworks.com/docs/latest/avi-sdk-and-migration-tools-release-notes/
4. Avi Devops repository ? automation samples: https://github.com/avinetworks/devops/

Document Revision History

Revision	Avi Vantage Release	Modification
1	17.2.1	This feature was introduced.