TECHNICAL GUIDE:
April 2024

VMware® Avi™ Load Balancer

Design Guide and Best Practices for VMware vSphere Environments

vmware® by Broadcom

Table of Contents

About this Document

This technical document describes the process to integrate VMware Avi Load Balancer (Avi) with a vSphere Cloud and the associated automation workflows. It can also serve as a design guide for vSphere integration with Avi, providing recommendations on best practices as applicable.

Contact Details

Company: VMware

ANS BU Contact: Vijayendra Singh
Yuriy Andrushko

Solution Overview

- VMware vSphere provides best-in-class virtualization solutions for businesses of all sizes with emphasis on data governance and privacy.
- The vCenter assumes a significant role in efficiently managing ESXi hosts, forming the fundamental backbone of the VMware product suite. VMware offers a wide variety of software solutions that includes VCF for software-defined datacenter operations and networking, Tanzu for the cloud-native application platform and Avi for software-defined application delivery control.
- The solution outlined in this document is centered on the integration of the Avi Controller with vCenter. This controller employs APIs to interact with the vCenter server, facilitating the discovery of the underlying infrastructure. The Controllers are tasked with managing the lifecycle of Service Engines (SE), responsible for handling data plane traffic. The Avi Controllers serve as the control-plane and management console, empowering users to configure load balancing for their applications, while the Service Engines establish a distributed and elastic load balancing fabric.

Applicability

- vCenter 6.7, 7.0 and 8.0
- VMware Avi Load Balancer (Avi) – 20.1.4 and above

Prerequisites

As a general prerequisite for deployment of Avi into vSphere environment, this environment must adhere to the VMware recommendations, which can be referenced at https://www.VMware.com/content/dam/digitalmarketing/VMware/en/pdf/techpaper/performance/vSphere-ESXi-vCenter-server-80-performance-best-practices.pdf.

System Requirements

Avi has specific requirements regarding hardware resources assigned to virtual machines. It is essential to assess the compute, storage, and memory requirements for each component, including Controllers and Service Engines (SE). Broadly, augmenting hardware capacity significantly enhances the overall system capacity for both Avi SEs and Avi Controllers. For detailed information on sizing Controllers and Service Engines, please refer to the document below and size your infrastructure accordingly.

- Controller Sizing
- Service Engine Sizing

In a standard deployment, a redundant Controller cluster typically consists of three Controllers. The necessary number of SEs depends on factors such as the quantity of applications, traffic load handled by each SE, and the configured level of redundancy. Service Engines are grouped together based on their properties and the group of Virtual Services they handle.

Notes:

- While it is not mandatory, it is strongly recommended to reserve CPU and memory.
- Ensure uniform sizing across all components, including Controller nodes and SEs within an SE Group.

Controllers and Service Engines have various limits for Virtual Services, configurable parameters etc. Please refer to VMware Configurations Maximums tool to get detailed information about different maximums for a specific Avi deployment.

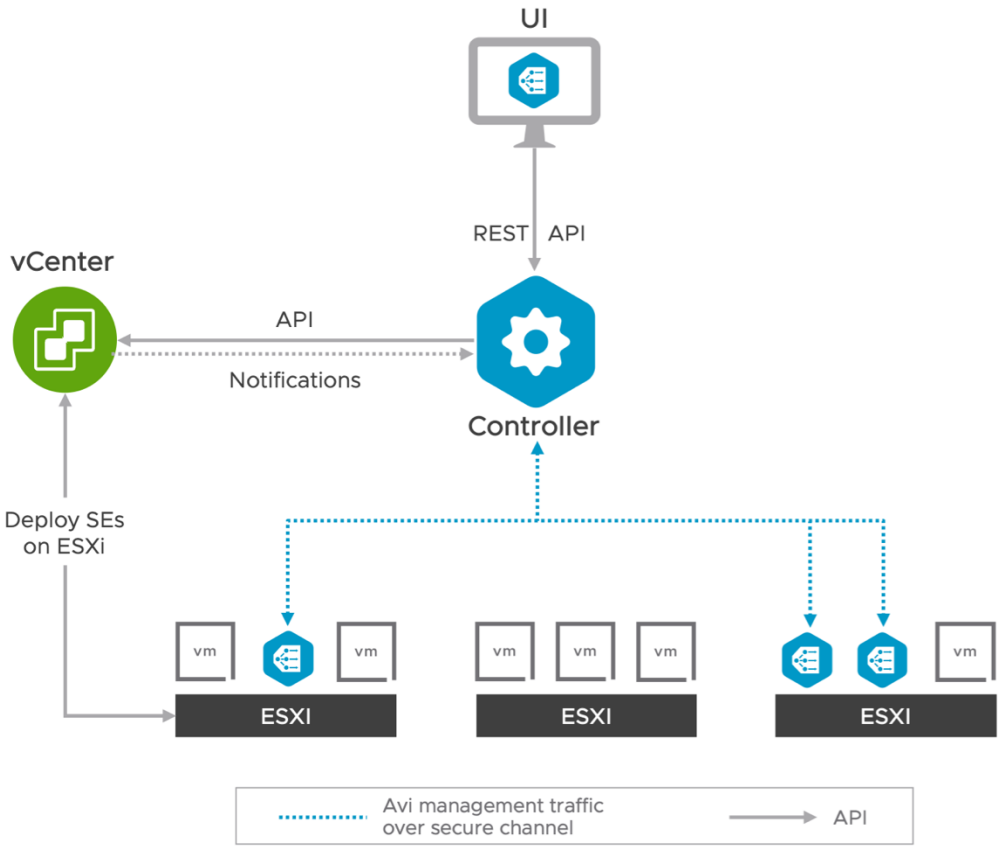
Licensing

VMware Avi Load Balancer is available in two editions:

- VMware Avi Load Balancer Enterprise edition, which provides a full-featured enterprise grade load balancing including multi-cloud integration, active-active high availability, Global Server Load Balancing (GSLB), Web Application Firewall (WAF), and so on.
- VMware Avi Load Balancer Enterprise with Cloud Services edition provides same set of features as Enterprise plus ability to employ centralized licensing, automatic WAF CRS updates, proactive and automatic support case management and many more.

Virtual Infrastructure Design

Avi components (Controllers and Service Engines) run as virtual machines (VMs) managed by vCenter. In a VMware cloud managed by vCenter, Avi operates as a fully distributed, virtualized system comprising the Avi Controller and Avi Service Engines, each running as a virtual machine. Below section describes the architecture and key components:



Avi Controller (Control Plane)

The Avi Controller is responsible for storing and managing all configurations and policies pertaining to services and management. It utilizes vCenter APIs to discover VMs, data centers, networks, and hosts. Leveraging this auto-discovered information, virtual services can be added through the web interface of the Avi controller. To deploy a virtual service, the Avi Controller autonomously selects an ESXi server, initiates an Avi SE, and connects it to the appropriate networks (port groups).

Avi Service Engines (Data Plane)

Each Avi Service Engine operates within its own virtual machine. These SEs deliver application delivery services for end-user traffic and capture real-time end-to-end metrics for communication between users and applications. The Service Engines push logs and metrics to the Controllers or they can be configured to send metrics / logs to external log servers such as Splunk.

Avi Ports and Protocols Used

The Avi solution is comprised of distributed Control (Controller Cluster) and Data (Service Engines) planes, necessitating communication between them. Depending on factors like the placement of Management interfaces within the same or different networks and the presence of firewalls, explicit allowances for management traffic between these components may be required. In fully orchestrated mode, also known as Write Access Mode, the Avi Controller Cluster will additionally need specific ports opened to engage with the Cloud endpoint, such as the vCenter server.

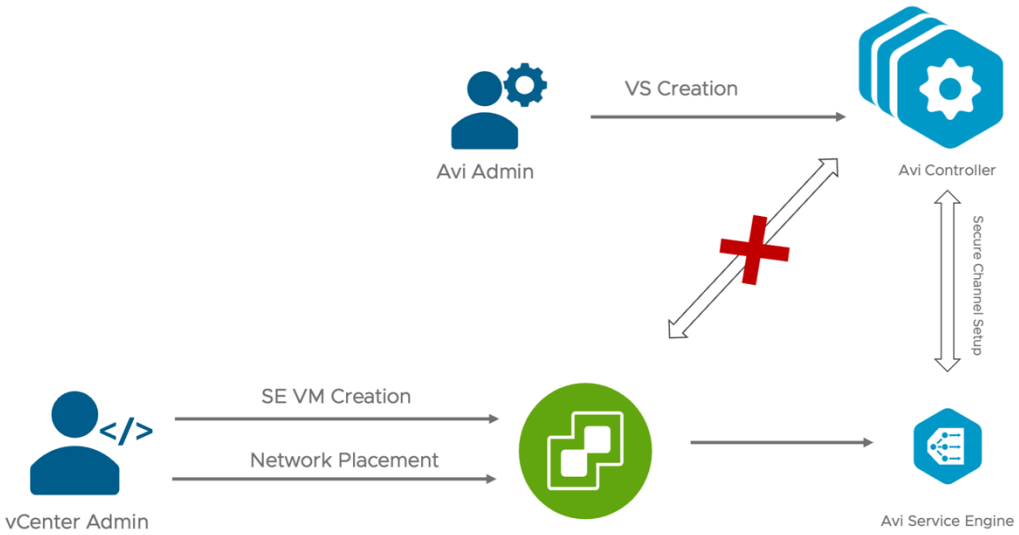
For a comprehensive list of utilized protocols and port numbers, please refer to VMware Ports and Protocols.

Orchestrator Access Modes

Avi can be deployed in a virtualized environment, and its communication with the virtualization orchestrator (like VMware vCenter) plays a crucial role in system operation and configuration. This communication level is termed the access mode. Avi can be deployed in two different modes in vCenter environment:

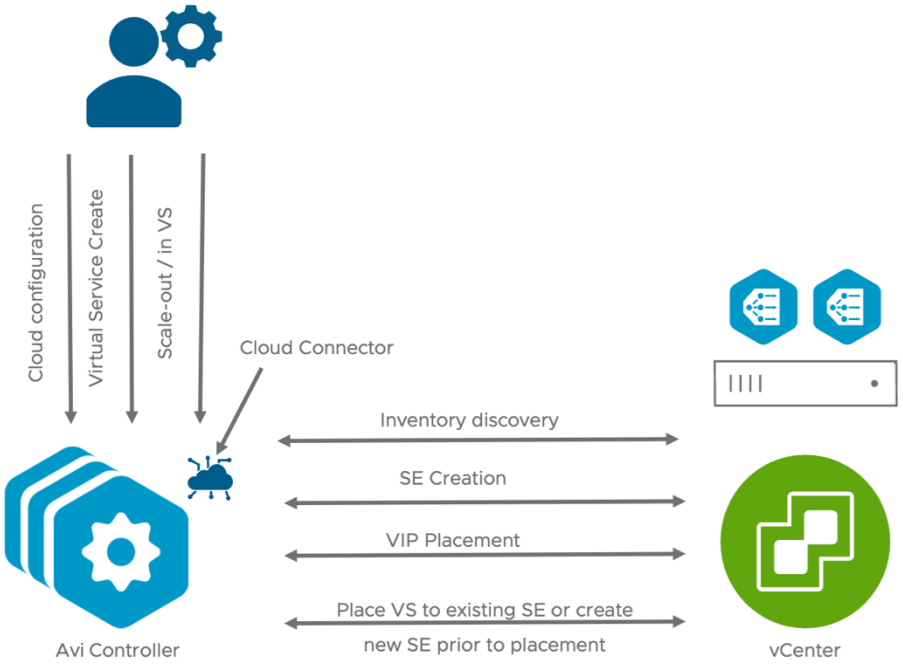
- No Orchestrator Mode
- Fully Orchestrated Mode (Write-Access)

No Orchestrator Mode



The Avi Controller operates without read or write access to vCenter in this mode. Consequently, tasks such as adding, removing, or modifying properties of a Service Engine requires manual intervention by an administrator. For example, to install a new SE through the vCenter, the administrator must upload the OVA and configure resource and networking properties. Similarly, creating a new virtual service may require admin access to vCenter to adjust the network settings for the new virtual server. Auto-discovery of servers and networks is not supported, and configurations must be manually set. In this mode, the Avi cloud setting is configured as "no-orchestrator," regardless of the cloud or virtualization environment.

Fully Orchestrated Mode



In this mode, Avi is given complete write access to the Cloud endpoint. It can autonomously manage the creation, modification, and removal of SEs and other resources to dynamically respond to evolving traffic requirements. This deployment mode is strongly recommended for cloud environments whenever possible. Fulfilling this automation requires the user credentials of the Cloud endpoint with specific access rights.

IP Address Requirements

The Avi Controller only needs a single management IP address. Administrative commands are set up on the Controller by accessing it through this IP address. This same IP address is also used by the Controller to communicate with Service Engines. It's essential that the management IP address for all Controllers in a cluster belongs to the same subnet.

For each Avi Service Engine, you need one management IP address for the management vNIC interface, an IP address for the data vNIC connected to the VIP segments, and another IP address for the vNIC connected to the pool network.

It's recommended to use DHCP for Avi SE management and data vNIC IP address allocation, rather than static assignment. This recommendation assumes the presence of supporting infrastructure and the necessary configuration.

When creating the load balancing application, the virtual service IP address is manually specified. The allocation of the VS VIP can be automated using native Avi IPAM or by integrating it with any other IPAM service.

Design Decisions for Virtual Infrastructure Provisioning

Below table consist of design decisions one should consider while planning the infrastructure to make sure high availability of both control plane and data plane of Avi.

Design Decision ID	Design Decision	Design Justification	Design Implication
Avi-VI	Place the ESXi hosts across multiple racks.	The management hosts shall be highly available.	Eliminates downtime associated with an entire rack failure.
	Create a management cluster with a minimum of four (4) ESXi hosts.	Having four (4) ESXi hosts in in the management cluster guarantees host-level redundancy during outages or maintenance operations.	Maintenance operations and outages can impact traffic forwarding.
	Choose vSAN-based or another high-speed SSD-backed datastore for placement of Avi components	Logs processing and retrieval is a disk-intensive task and thus will benefit from high-speed disk backing	Events and log display may show slowness in data retrieval
	Create separate vCenter user account with only needed permissions	Separate user account gives visibility and accountability of changes incurred by Avi. Limited permissions guardrails Avi instance from compromising Virtual Infrastructure.	Limited audit possibilities and higher impact in case of misconfigurations.
	Create vCenter content library	Content library allows reusing generated Service Engine images and speeds up deployment, especially in situations when Controller Cluster is residing in network other than vSphere Management network.	Service Engine deployment will be done directly against ESXi hosts and thus will mandate API/HTTPS access from Avi Controller Cluster to ESXi hosts.
	Enable AES-NI instructions setting in the BIOS for ESXi hosts if available.	AES-NI instruction set provides efficiency in SSL performance.	Most modern machines have AES-NI enabled by default, if not enabled by default, a reboot of ESXi hosts would be necessary to enable this setting.
	Anti-affinity 'VM/ Host' rule advised to be used to prevent collocation of Avi Controller VMs.	vSphere will take care of placing the Avi Controller VMs in a way that always ensures maximum HA for the Avi Controller cluster.	vSphere DRS can put multiple members of Controller Cluster or Service Engine Group onto same host thus increasing potential impact of host crash.
	A virtual machine group advised to be used for the Avi Controller VMs.	Ensures that the Avi Controller VMs can be managed as a group.	vSphere configuration is needed to add virtual machines to the allocated groups.

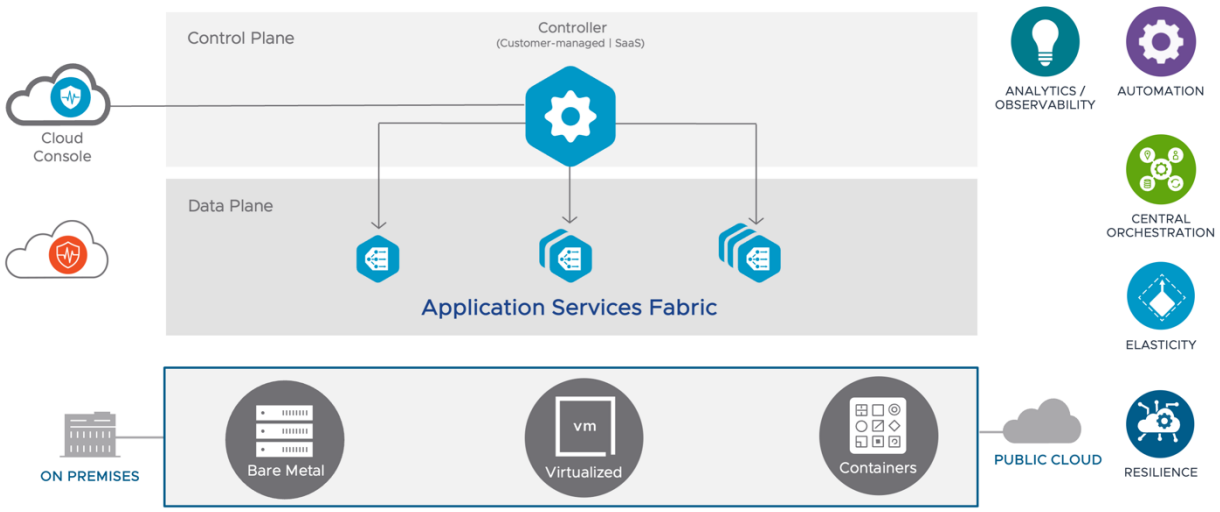
Design Decision ID	Design Decision	Design Justification	Design Implication
	In vSphere HA, for each Avi Controller and Avi Service Engine VMs, set the restart priority policy to high and host isolation response to disabled.	This ensures fast recovery for Avi.	Manual intervention would be required to bring Controller Cluster VMs back online.
	Place Controller Cluster Nodes on Same management network as vCenter server.	Ensures uninterruptible communication between components	Routed reachability needs to be provisioned and management ports needs to open if firewall resides between components.
	Provision DHCP services for consumption by Service Engine Management and Data interfaces.	Having DHCP enabled for data networks would make Avi SE configuration simple.	Separate static static/default route configuration and static addressing per affected VRF will be needed
	Allow outbound communication for Controller IPs toward VMware Cloud Services FQDN	Needed for Enterprise with Cloud Services Licensing tier	Enterprise with Cloud Services Licensing tier would not be possible
	Reserve four IP addresses in the management subnet to be used as the cluster IP and member nodes for the Avi Controller cluster.	A floating IP that will always be accessible regardless of a specific individual Avi cluster node allows easier access to Cluster nodes.	None
	NTP will be used for time synchronization for the Avi Controller.	Prevents time synchronization issues and provides consistent timestamps for log entries. Not required to provide connectivity to an external NTP server.	An operational NTP service must be available in the environment. Ensure that NTP traffic between the Avi Controllers, Avi SEs and the NTP servers is allowed on the required network ports and not firewalled.
	Configure system DNS server.	Allows name resolution for SE and Controller	None

Note:

In addition to the above design decisions, it is important to note that whenever DRS (Distributed Resource Scheduler) setting is set to Automatic, it is recommended to put Service Engine VMs in VM Override Group. Avi Controller chooses best ESXi node for SE deployment but some type of traffic e.g., UDP can show drops after vMotion triggered by DRS

Avi Controller Cluster Design

The Avi platform is built on software-defined principles, enabling a next generation architecture to deliver the flexibility and simplicity expected by IT and lines of business. Avi architecture separates the data and control planes to deliver application services beyond load balancing, such as application analytics, predictive autoscaling, micro-segmentation, and self-service for app owners in both on-premises or cloud environments. The platform provides a centrally managed, dynamic pool of load balancing resources on commodity x86 servers, VMs or containers, to deliver granular services close to individual applications. This allows network services to scale near infinity without the added complexity of managing hundreds of disparate appliances.



The Avi Controller cluster uses big data analytics to analyze the data and present actionable insights to administrators on intuitive dashboards on the Avi Admin Console.

Avi provides out-of-the-box integrations for on-premises or cloud deployments. These integrations with private cloud frameworks, SDN controllers, container orchestration platforms, virtualized environments and public clouds enable turnkey application services and automation.

Key Concepts and Terminology

This section provides the common Avi concepts that are used in the documentation and user interface. Avi Platform has three core components – Avi Service Engines, Avi Controller cluster, and the Avi Admin Console.

Control Plane

It computes runtime state based on configuration from the management plane. It receives input from users and various connected network end-points and pushes stateless configuration to forwarding engines. It consists of three Controllers joined into the Controller Cluster. Cluster Leader is usually responsible for the presentation of Admin Console.

Data Plane

Performs stateless forwarding or transformation of packets based on tables populated by the control plane. Data plane reports topology information to the control plane that maintains packet level statistics. It consists of all the Service Engines deployed by Controller Cluster.

Avi Controller

The Avi Controller is the single point of management and control that serves as the "brain" of the entire Avi system and for high availability is typically deployed as a three-node cluster. As its name implies, the Controller implements the control plane.

Cloud

Clouds are containers for the environment that Avi is installed or operating within. During initial setup of Avi, a default cloud, named "Default-Cloud," is pre-created. Additional clouds may be configured, containing Service Engines and virtual services, as per the requirements.

Service Engine Group

Service Engines are created within a group, which contains the definition of how the SEs should be sized, placed, and made highly available. Each cloud will have at least one SE group.

Virtual Services

Virtual services are the core of Avi load-balancing and proxy functionality. A virtual service advertises an IP address and ports to the external world and listens for client traffic.

Pools

Pools maintain the list of servers assigned to them and perform health monitoring, load balancing, persistence, and functions that involve Avi-to-server interaction. One or more pools could be attached to a virtual service.

Pool Groups

A pool group is a list of server pools, accompanied by logic to select a server pool from the list. A virtual service can be attached to a pool-group where it can refer to a server pool direcly or via rules / DataScripts / service port pool selector.

Health Monitors

Avi must validate whether servers are working correctly and are able to accommodate additional workloads before load balancing a client to a particular server. Health monitors perform this function by either actively sending a synthetic transaction to a server or by passively monitoring client experience with the server.

Static Route

A static route may also be set as the default gateway. Default gateways may also be defined within the settings of an SE, which will override the global static routes, and will be specific to the modified SE. If DHCP is not used and a default gateway needs to be defined, then it is recommended to define the gateway within the Static Routes tab, which will be applicable to all SEs.

VRF
Virtual Routing Framework (VRF) is a method of isolating traffic within a system. This is also referred to as a route domain within the load balancer community.

Gateway Monitor
Health monitoring of the first-hop gateway connected to Avi SEs. ICMP echo packets are used for the health monitoring. Gateway monitoring is also available for routers that are not directly connected.

GSLB
Global server loading balancing (GSLB) is the act of balancing an application's load across instances of the application that have been deployed to multiple locations (typically, multiple data centers and/or public clouds).

Tenant
A tenant is an isolated instance of Avi. Each Avi user account is associated with one or more tenants. The tenant associated with a user account defines the resources that user can access within Avi. When a user logs in, Avi restricts their access to only those resources that are in the same tenant.

Design Decisions for Avi Controller Provisioning
Below table consist of design decisions one should consider before deploying the controller to make sure the Controllers are highly available and as per the recommendations given by VMware:

Design Decision ID	Design Decision	Design Justification	Design Implication
Avi- CTLR-001	Initial setup should be done only on one Avi Controller VM out of the three deployed to create an Avi Controller cluster.	The Avi Controller cluster is created from an initialized Avi Controller which becomes the cluster leader. Follower Avi Controller nodes need to be uninitialized to join the cluster.	Avi Controller cluster creation would fail if more than one Avi Controller is initialized.
	Deploy each node in the Controller cluster with a minimum controller size mentioned in this doc.	Different Avi Controller sizes support different VS limits. Refer to configMax for these limits before planning the controller size.	Under sizing, Avi Controllers can lead to unstable control plane functionality.
	Use static IPs or DHCP with reservation, ensuring a permanent lease for Avi Controllers.	Controller cluster uses management IPs to form and maintain quorum for the control plane.	If Controller's IP changes, manual intervention is needed to re add the node to Cluster
	Choose tenancy settings for Service Engine management – managed in provider context or in Tenant context.	This Cluster-wide setting is one of the ways for application traffic delimitation and Service Engine access.	None
	Rotate passwords at least every 3 months.	Ensures security of the user accounts.	None

Design Decision ID	Design Decision	Design Justification	Design Implication
	Limit the use of the Avi local accounts for both interactive or API access and solution integration.	Dedicated service accounts, security groups, group membership, and security controls must be defined and managed in Active Directory.	Local accounts are not specific to user identity and do not offer complete auditing from an endpoint back to the user identity.
	Create user accounts on Avi Controller with desired Roles to limit the scope and privileges for accounts used for both interactive or API access and solution integrations.	The principle of least privilege is a critical aspect of access management and should be part of a comprehensive defense-in-depth security strategy.	Custom roles and security controls may need to be defined to limit the scope and privileges used for interactive access or solution integration.
	Choose one or more types of notification of choice for monitoring/ alerting. It is recommended to enable alerts on the following pre-defined `System` alerts: System-VS-Alert System-SSL-Alert System-SE-Alert System-Controller-Alert System-CC- Alert	Ensure good health through proactive alerting of the VMware Avi Load Balancer cluster.	None
	Configure Alert notification settings.	Allows user to get notifications for default and custom events in preferred way	Events and Alerts will be visible only in Avi UI
	Backup schedule tailored to customers SLA.	Backed up configuration will aid in rebuilding and recovering the Avi configuration from catastrophic failures.	Backup server should support SCP as the transport protocol.

vSphere Cloud Design

During the initial Cloud setup for fully orchestrated type, a vCenter account must be entered to allow communication between the Controller and the vCenter. The vCenter account must have the privilege to create new folders in the vCenter. This is required for Service Engine creation, which then allows virtual service placement. For detailed description of user role requirements, refer to Avi installation guide.

No orchestrator type of access does not require an additional user and will not communicate with vSphere server for automation of Service Engine lifecycle.

To make a thoughtful decision regarding settings for High Availability and scale-out, refer to official product documentation.

Design Decisions for vCenter Cloud Configuration

Below table consist of design decisions one should consider before configuring the cloud. This configuration defines the attributes for Controller to vCenter communication and configuration done under cloud is applicable to all SEs created under a particular cloud.

Design Decision ID	Design Decision	Design Justification	Design Implication
Avi- CLD-001	Create a new custom Cloud object or edit the Default-Cloud object	Custom cloud object gives flexibility to name Cloud object and provide custom tags for objects instantiated by it.	You cannot change the name of the Default-Cloud.
	Provide one or more VLAN segments as data networks for the vCenter Cloud connector.	Avi SE interfaces would be connected to vSphere-managed VLAN segments.	None
	Use FQDN (domain names) for vSphere server and other relevant components	Eases eventual IP changes	In case IP for vCenter server will need to be changed, Cloud object will need to be deleted and re-created
	Provide an object name prefix when creating the Cloud Connector on the Avi Controller.	Used for uniquely identifying Cloud Connector-created resources in vCenter Server object database.	None
	Provision DHCP services for used Management and Data networks	This will greatly ease configuration of associated interfaces.	Static configuration will be needed for each of Service Engine interfaces
	Use IPAM for Service Engine interfaces	As an alternative to DHCP for SE interface address, IPAM can be used for the same purpose. IPAM is a way to only assign IP addresses and it lacks assignment of Default Gateway, DNS, etc.	IPAM implies need for static routes if used for SE interface address assignment due to fact IPAM can't program default gateway.
	Use IPAM for VIPs	For VIPs IPAM provides an easy way of automatic administering IP assignment to speed up application provision and guarantee consistency.	Static IP assignment will be required for each configured VS VIP.
	Delegate a subdomain and create corresponding DNS profile for automatic registration of VIP-specific domain names.	Allows creation of human friendly FQDN for provisioned applications.	Either IP-based navigation to provisioned applications or additional configuration on external DNS server would be required.
	Choose whether to prefer static routes (two-arm) or connected networks (one-arm).	This cloud-wide setting controls the way frontend and backend are being attached to Service Engine Data Interfaces	Consider vSphere limit of 10 vNICs per VM (Service Engine) Two-armed topology gives the best Packets Per Second rates but consumes two interfaces for each Frontend-Backend pair.

Design Decision ID	Design Decision	Design Justification	Design Implication
			One-armed topology consumes single interface for each Frontend-Backend pair, which can be a bottleneck in PPS-intensive scenarios (L4 application).
	Create additional VRFs as needed from traffic separation perspective.	In case Tenancy routing mode is per-tenant, additional VRFs will be needed to comply with it.	Single shared VRF will be used for all applications
	Use pre-created VCenter Content Library.	Content library allows reusing generated Service Engine images and speeds up deployment, especially in situations when Controller Cluster is residing in network other than vSphere Management network.	Service Engine deployment will be done against ESXi hosts and thus will mandate API/HTTPS access from Avi Controller Cluster to ESXi hosts.
	Configure BGP for VIP announcements.	In case considerable number of applications are in place or there is a need to provide ECMP scale-out, BGP is needed	Layer 2 scale out is limited to 4 SEs with primary carrying more load than secondary SEs. BGP is needed for ECMP and scaling out beyond 4 SEs.
	Configure DNS resolution on SEs.	When Controller Cluster is deployed in air-gapped environment or name resolution should be different from Management plane, you can opt in to resolve DNS names from SE Data interface	Controller Cluster is resolving all DNS queries via its management connection. This configuration eliminates the dependency on controller for DNS resolution.
	Create multiple SE Groups as desired to isolate applications.	Allows efficient isolation of applications and allows for better capacity planning. Allows flexibility of life- cycle-management.	None
	Configure SE Group for Active/ Active HA mode.	Provides optimum resiliency, performance, and utilization.	Certain applications might not work in Active/Active mode. For instance, applications that require preserving client IP. In such cases, use the Legacy Active/ Standby HA mode.
	Set 'Placement across Avi SEs' setting to 'distributed'.	This allows for maximum fault tolerance and even utilization of capacity.	Might require more Avi SE VMs as compared to 'compact' placement mode.
	Enable CPU and Memory reservation on the SE Group.	Avi SEs are a critical infrastructure component providing load-balancing services to mission critical applications.	None

Design Decision ID	Design Decision	Design Justification	Design Implication
	For Service Engine Groups where only single application will reside (high-bandwidth applications), activate Per-app Service Engine Group setting	Allows reducing licensing needs for Application-dedicated SEGs	None
	Change number of Service Engines a group can have	Each group can host several applications, depending on number of SEs or SE-Group, VS scale and number of VS per SE. This setting guardrails that your group will not expand beyond expected limits.	None
	Change VS scale	This SEG-wide setting specifies how many VS replicas will exist. Setting it to more than one (1) effectively turns Active/Active mode of operation. Applies to every application VS placed onto this SEG. Will impact amount of provisioned Service Engines as every placed application VS will effectively create X VSes, where X is a scale number.	Every VS will exist in a single replica.
	Change maximum number of Virtual Service, an SE can have	This setting controls the maximum number of Virtual Services, each SE can host.	None
	Provide host Cluster and Datastore Cluster configuration	This allows you to explicitly set Hosts and Datastores where SEs can be spun up. Applicable in distributed design where specific group of hosts is serving Edge role.	None
	Assign object tags	Provides an ability for easier grouping and automation using vSphere tags	None

Performance Tuning

Avi's default configurations are best suited for most of the cases, but there are scenarios which may need tweaking configurations at each level such as: Cloud, Service Engine Group, Virtual Service etc.

The table given below helps to identify use-cases and recommends the configuration that will help with better user experience. The links to VMware documentations helps with configuration examples and details about the features discussed.

Use-Case	Recommendation	Link
High number of Pool Servers / Pool Groups.	In case of backend servers autoscaling (Tanzu / k8s / OpenShift etc.), use autoscale_polling_interval (cloud level property). It is recommended to increase polling interval in this use-case. It helps to decrease API load on controllers.	None
Low PPS with High Proxy Utilisation (WAF, high SSL transactions per second).	Use Service engines with only single dispatcher core and have high number of Proxy cores (>=3)	Link to follow
PPS heavy traffic profile with lesser proxy utilisation.	Use Service Engine with 2 Dispatcher with max_queue_per_vnic = 2, for a service engine >=4 vCPU.	Link to follow
Latency and Jitter Sensitive applications	Set se_dp_isolation mode to true and set number of non-datapath core value.	Link to follow
IP Routing traffic	Make sure to disable LRO for IP Routing traffic.	Link to follow
Virtual Service Need to scale to more TCP Connections per second.	Virtual Service with TCP proxy profile, should have knob "ignore_time_wait" enabled in TCP Proxy Profile. This will help with additional new connections per seconds.	Link to follow
Running 2vCPU Service Engine size.	Hybrid RSS mode is recommended for 2 vCPU SEs, to have better performance.	Link to follow
Time sensitive Application	It is recommended to disable GRO / LRO as receive offloads may add latency, waiting for incoming packets.	Link to follow
High number of SSL Open Connections	Open connections have memory implications. Increase the Service Engine Memory to accommodate additional SSL overhead.	Link to follow
For high-speed NICs (10+ Gbps) enabling RSS helps to achieve higher Packet-per-Second values.	RSS works only with number of Dispatcher equal to power of two (1,2,4,8). Amount of NIC queues should be higher than number of dispatchers – set this value to zero to perform automatic calculation. RSS does not work with IPv6 – if IPv6 address gets configured on any interface, feature is disabled on that interface. Enabling RSS is a disruptive action and requires SE reboot.	Link to follow
For SEs with four or more dedicated Dispatcher cores, enable GRO.	For high-bandwidth applications enabling GRO reduces CPU overhead related to packet assembly. GRO is incompatible with routing, so if SE is configured with routing options, GRO can't be enabled. Enabling GRO is non-disruptive.	Link to follow
Enabling of WAF for VS placed on SE requires at least 2vCPU and 4 GB of memory.	Compiling WAF signatures, packet buffering, as well as other WAF engine functions require increased amount of RAM. Some other functions, such as Bot classification will even require dedicated configuration memory in SEG settings	Link to follow